RAYTHEON INFORMATION SYSTEMS COMPANY

ADDENDUM NOTICE

EOS Core System (ECS) Project Contract No. NAS5-60000

September 16, 1999

Document No.: 611-CD-500-001

Title: Mission Operation Procedures for the ECS Project: Release 5A

The following addendum pages have been added to the subject document:

Insert

28-1 and 28-2

If you have any questions, please contact our Data Management Office at (301) 925-0509.

RAYTHEON INFORMATION SYSTEMS COMPANY

ERRATA NOTICE

EOS Core System (ECS) Project Contract No. NAS5-60000

September 16, 1999

Document No.: 611-CD-500-001

Title: Mission Operation Procedures for the ECS Project: Release 5A

The following pages have been replaced in the subject document:

Removed	Inserted
vii through l	vii through l
7-1 through 7-4	7-1 through 7-4

If you have any questions, please contact our Data Management Office at (301) 925-0509.

611-CD-500-001

EOSDIS Core System Project

Mission Operation Procedures for the ECS Project

This document has not yet been approved by the Government for general use or distribution.

Release 5A

July 1999

Raytheon Systems Company
Upper Marlboro, Maryland

Mission Operation Procedures for the ECS Project

Release 5A

July 1999

Prepared Under Contract NAS5-60000

RESPONSIBLE ENGINEER

Earnest Matthews /s/	7/13/99
Earnest T. Matthews, Systems Engineer EOSDIS Core System Project	Date

SUBMITTED BY

Karen Hoffman /s/	7/13/99
Karen K. Hoffman, I&O Support Manager EOSDIS Core System Project	Date

Raytheon Systems Company
Upper Marlboro, Maryland

611-CD-500-001

This page intentionally left blank.

Preface

This document is a contract deliverable with an approval code of 3. This technical paper is delivered to NASA for information only, but is subject to approval as meeting contractual requirements.

Any questions should be addressed to:

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

This page intentionally left blank.

Abstract

This document, The Mission Operation Procedures for Release 5A for ECS, provides DAAC procedures that assign and describe operators, engineers, operations support, administration and management staff actions required to configure, maintain and operate Release 5A ECS system. The DAAC portion of this document contains standard procedures that can be modified at the DAACs during subsequent training, operations exercises and procedure review activities to reflect desired uniqueness. The objectives of Release 5A system are to provide ECS components to support the MODIS mission; Version 0 Data Access; EOS-AM-1 and Landsat 7 Interface Testing; and EOS-AM- 1 Algorithm Integration and Test.

Keywords: operations, DAACs, SMC, EOC, mission support, operation procedures, TRMM, EOS, AM-1, Landsat 7, software, integration, test, SSI&T, Version 0

This page intentionally left blank.

Change Information Page

List of Effective Pages	
Page Number	**Issue**
Title	Original
iii through l	Errata 1
1-1 through 1-4	Original
2-1 through 2-6	Original
3-1 through 3-86	Original
4-1 through 4-28	Original
5-1 through 5-24	Original
6-1 through 6-14	Original
7-1 through 7-24	Original
8-1 through 8-24	Original
9-1 through 9-138	Original
10-1 through 10-10	Original
11-1 through 11-318	Original
12-1 through 12-32	Original
13-1 through 13-38	Original
14-1 through 14-40	Original
15-1 through 15-12	Original
16-1 through 16-50	Original
17-1 through 17-102	Original
18-1 through 18-24	Original
19-1 through 19-80	Original
20-1 through 20-6	Original
21-1 through 21-24	Original
22-1 through 22-30	Original
23-1 through 23-14	Original
24-1 through 24-6	Original
25-1 through 25-4	Original
26-1 through 26-10	Original
27-1 through 27-54	Original
28-1 and 28-2	Addendum 1
A-1 and A-2	Original
AB-1 through AB-12	Original

Document History			
Document Number	**Status/Issue**	**Publication Date**	**CCR Number**
611-CD-500-001	Original	July 1999	
611-CD-500-001	Errata 1/Addendum 1	September 1999	

This page intentionally left blank.

Contents

Preface

Abstract

1. Introduction

1.1 Identification 1-1

1.2 Scope 1-1

1.2.1 On-Site Procedures Tailoring Guide 1-1

1.3 Purpose 1-1

1.4 Status and Schedule 1-2

1.5 Organization 1-2

2. Related Documentation

2.1 Parent Documents 2-1

2.2 Applicable Documents 2-1

2.3 Information Documents 2-4

2.3.1 Information Documents Referenced 2-4

2.3.2 Information Documents Not Referenced 2-5

3. System Administration

3.1 System Startup and Shutdown 3-2

3.1.1 Startup 3-2

3.1.2 Shutdown 3-5

3.2 System Backup and Restore 3-8

3.2.1 Incremental Backup 3-8

3.2.2 Full Backup 3-12

3.2.3 File Restore 3-15
3.2.4 Complete System Restore 3-18
3.2.5 Tape Handling 3-22
3.3 System Log Maintenance 3-29
3.4 User Administration 3-31
3.4.1 Adding a User 3-31
3.4.2 Deleting a User 3-32
3.4.3 Changing a User Account Configuration 3-33
3.4.4 Changing User Access Privileges 3-34
3.4.5 Changing a User Password 3-34
3.4.6 Checking a File/Directory Access Privilege Status 3-35
3.4.7 Changing a File/Directory Access Privilege 3-37
3.4.8 Moving a User's Home Directory 3-40
3.5 Installing a New Workstation 3-41
3.5.1 Preparation 3-41
3.5.2 Installation 3-43
3.5.3 Testing and Verification 3-60
3.6 DCE Configuration 3-65
3.6.1 Configuring Initial Cell 3-65
3.6.2 Configuring DTS Servers 3-67
3.6.3 Configuring Additional CDS Servers 3-69
3.6.4 Configuring Security and CDS Client Systems 3-71
3.6.5 Configuring DTS Clerks 3-72
3.6.6 Configuring GDA Servers 3-73
3.6.7 Creating Security Server Replica 3-74
3.6.8 Unconfiguring DCE Client 3-75
3.7 ECS Assistant 3-76
3.7.1 Using ECS Assistant to Install Subsystem Software 3-77
3.7.2 Using ECS Assistant to Start Up/Shut Down Servers 3-78
3.7.3 Using ECS to Perform System Monitoring 3-81

4. Database Administration

4.1 Overview of Database Administration 4-1

4.2 SQL Server Environment....4-1
4.2.1 Naming Conventions, ...4-1
4.2.2 SQL Server Directory Structure 4-5
4.2.3 SQL Server Installation 4-6
4.3 Database Administrator Responsibilities....4-7
4.3.1 Startup of SQL Server....4-7
4.3.2 Shutdown of SQL Server 4-7
4.3.3 Showing SQL Server(s)....4-8
4.4 Allocation of Resources 4-9
4.4.1 Creating Logical Devices 4-9
4.4.2 Creating and Altering Databases....4-10
4.4.2.2 Example of Altering a Database....4-11
4.5 Loading a database you have created into a different database: 4-13
4.6 Monitoring Space Usage 4-14
4. 6.1 Thresholds 4-14
4.7 Creating Database Objects 4-14
4.7.1 Example of Creating a User Table....4-14
4.8 Creating and Managing Logins and Roles 4-15
4.8.1 Example of Creating a Login and Granting Database Access 4-15
4.9 Permissions....4-16
4.9.1 Example of Granting Privileges to a Specific User 4-16
4.10 Backup and Recovery....4-17
4.10.1 Automatic Backups....4-17
4.10.2 Manual Backups 4-20
4.10.3 Manual Recovery 4-21
4.10.4 The BulkCopy Utility 4-21
4.11 Database Performance and Tuning....4-23
4.12 Installation of the Applications....4-23
4.12.1 Installation of the Application Database....4-23
4.12.2 The AUTOSYS Application and other Configuration Issues....4-23
4.12.3 Spatial Query Server (SQS) 4-24
4.13 Passwords Security....4-25

5. Security Services

5.1 Monitoring Network Vulnerabilities.....5-1

5.2 Ensuring Password Integrity.....5-2

5.2.1 Detecting Weak Passwords.....5-2

5.2.2 Configuring AnlPasswd.....5-7

5.2.2.1 Installing anlpasswd.....5-9

5.3 Enforcing Strong Passwords (Secure Shell).....5-11

5.3.1 The SSH Encryption Mechanism.....5-12

5.3.2 How a User uses Secure Shell.....5-13

5.3.3 Configuration of Secure Shell.....5-13

5.3.4 Administration of Secure Shell.....5-17

5.4 Monitoring Requests for Network Services (TCP Wrappers).....5-17

5.5 INSTALLATION, CONFIGURATION, and TESTING for Wrappers.....5-17

5.6 Monitoring File and Directory Integrity (Tripwire).....5-20

5.6.1 Installation of Tripwire.....5-20

5.6.2 Updating the Tripwire Database.....5-21

5.6.3 Configuring the tw.config file.....5-22

5.7 Reporting Security Breaches.....5-23

5.8 Initiating Recovery from Security Breaches.....5-23

6. Network Administration

6.1 HP Open View Network Node Manager (NNM).....6-1

6.1.1 Starting Network Node Manager (NNM).....6-2

6.1.2 Creating Additional Objects.....6-4

6.1.3 Viewing the Current Network and System Configuration.....6-9

6.1.4 Viewing Network Address Information.....6-10

6.1.5 Viewing How Traffic is Routed on a Network.....6-11

6.1.6 Viewing the Services Available on a Node.....6-12

6.2 Diagnosing Network Problems.....6-13

7. System Monitoring

7.1 Checking the Health and Status of the Network 7-1
7.1.1 Starting NNM (Network Node Manager) 7-2
7.1.2 Verify That an Object Is Not Functioning.....7-5
7.1.3 Looking at Maps for Color Alerts 7-6
7.1.4 Looking at Maps for New Nodes 7-8
7.1.5 Creating Special Submaps for Monitoring Status 7-9
7.1.6 Checking for Event Notifications.....7-9
7.1.7 Rediscovering the Network.....7-10
7.1.8 Starting and Shutting Down Servers from HP OpenView 7-12
7.2 Tivoli Enterprise Console 7-14
7.3 Application Servers.....7-21
7.3.1 Locate Application/Program icons 7-22
7.3.2 Start an Application/Program.....7-23
7.3.3 Shutdown an Application 7-23
7.3.4. Remove Red Application/Program icons (dead executable symbols) from the map 7-23
7.3.5 Exit HP OpenView and stop background processes 7-23

8. Problem Management Procedures

8.1 The Problem Resolution Process — An Overview 8-2
8.1.1 ECS Internal Process 8-2
8.1.2 Interface with ESDIS.....8-4
8.2 Problem Management Procedures.....8-5
8.3 Using the Trouble Ticket System.....8-6
8.3.1 Accessing the Trouble Ticket System.....8-9
8.3.2 Submit a Trouble Ticket 8-13
8.3.3 Reviewing and Modifying Open Trouble Tickets.....8-15
8.3.4 Forwarding Trouble Tickets.....8-16
8.3.5 Adding Users to Remedy 8-16
8.3.6 Changing Privileges in Remedy 8-16
8.3.7 Modifying Remedy's Configuration.....8-18
8.3.8 Generating Trouble Ticket Reports.....8-18
8.3.9 Re-prioritization of Dated Trouble Ticket Logs 8-18

8.4 Using Hypertext Mark-up Language (HTML) Screens 8-19
8.4.1 ECS Trouble Ticketing HTML Submit Screen 8-19
8.4.2 ECS Trouble Ticketing HTML Success Screen 8-20
8.4.3 ECS Trouble Ticketing HTML List Screen 8-20
8.4.4 ECS Trouble Ticketing HTML Detailed Screen..... 8-20
8.4.5 ECS Trouble Ticketing HTML Help Screen 8-21
8.5 Emergency Fixes 8-22

9. Configuration Management Procedures

9.1 Configuration Identification Procedure 9-2
9.1.1 Purpose 9-2
9.1.2 Applicable to 9-2
9.1.3 References 9-2
9.1.4 Procedures 9-3
9.2 Configuration Change Control Procedures 9-4
9.2.1 Purpose 9-4
9.2.2 Applicable to 9-4
9.2.3 References 9-4
9.2.4 Procedures 9-4
9.3 Configuration Status Accounting Procedures 9-22
9.3.1 Purpose 9-22
9.3.2 Applicable to 9-22
9.3.3 References 9-22
9.3.4 Procedures 9-22
9.4 Configuration Audits..... 9-23
9.4.1 Purpose 9-23
9.4.2 Applicable to 9-24
9.4.3 References 9-24
9.4.4 Procedures 9-24
9.5 Data Management 9-25
9.5.1 Purpose 9-25
9.5.2 Applicable to 9-26
9.5.3 References 9-26

9.5.4 Procedures 9-26
9.6 Archiving Procedures for the SW CM Manager (ClearCase) 9-30
9.6.1 Purpose 9-30
9.6.2 Applicable to 9-30
9.6.3 References 9-30
9.6.4 Procedures 9-30
9.7 SW Transfer and Installation 9-33
9.7.1 Purpose 9-33
9.7.2 Applicable to 9-33
9.7.3 References 9-34
9.7.4 Procedures 9-34
9.8 Change Request Manager 9-44
9.8.1 The Configuration Change Request (CCR) 9-45
9.8.2 Accessing Change Request Manager 9-49
9.8.3 View a CCR 9-49
9.8.4 Submit a CCR 9-49
9.8.5 Change State of CCR 9-52
9.8.6 Modify CCR 9-58
9.8.7 Print CCR 9-58
9.8.8 Required Operating Environment 9-59
9.8.9 Reports 9-60
9.9 Use of the Baseline Manager 9-63
9.9.1 Overview 9-63
9.9.2 Procedures 9-89
9.9.3 Setting Tunable Parameters 9-118
9.9.4 Special Constraints 9-120
9.9.5 Command Line Interface 9-120
9.9.6 Outputs 9-121
9.9.7 Event and Error Messages 9-122
9.9.8 Reports 9-123

10. Metadata Administration

10.1 ESDT Descriptor Files 10-1

10.1.1 Steps in Generating a Descriptor File 10-2
10.2 Preparation of Earth Science Data Types 10-3
10.3 Tools used in Generating a Descriptor File 10-6
10.3.1 Data Dictionary and Valids Checking 10-7
10.3.2 PSA Registry 10-7
10.3.3. Metadata Works 10-8
10.4 Metadata Population 10-8
10.4.1 Collection-Level Metadata 10-8
10.4.2 Granule-Level Metadata 10-8
10.4.3 Product-Specific Metadata 10-9
10.5 Testing and Validation 10-9

11. SSIT Operational Procedures

11.1 Science Software Integration and Test (SSI&T) Preparation and Setup 11-2
11.1.1 Key Operator Roles 11-2
11.1.2 COTS Software Tools 11-2
11.1.3 General Process 11-2
11.1.4 Preparation and Setup 11-8
11.1.5 SSIT Software Operating Instructions: 11-10
11.1.6 Updating the Leap Seconds and the Earth Motions files 11-11
11.1.7 Script Name: update_leapsec.sh 11-12
11.2 Science Software Integration and Test (SSIT) Manager 11-14
11.2.1 SSIT Manager Overview 11-14
11.2.2 SSIT Manager GUI 11-16
11.2.3 SSIT Manager Tools 11-20
11.2.4 Using the SSIT Manager: 11-20
11.3 Delivered Algorithm Package (DAP) - Acquiring, Unpacking, Subscription 11-21
11.3.1 Acquiring the Delivered Algorithm Package (DAP) 11-21
11.3.2 Unpacking a DAP 11-24
11.4 Science Software Configuration Management 11-25
11.4.1 ClearCase Overview 11-26
11.4.2 Creating a View in ClearCase 11-26
11.4.3 Setting a View in ClearCase 11-29

11.4.4 Creating a New Directory 11-30
11.4.5 Importing files into ClearCase 11-31
11.4.6 Checking Out a File From ClearCase 11-35
11.4.7 Checking a Modified Element into ClearCase 11-36
11.5 Standards Checking of Science Software 11-37
11.5.1 Checking FORTRAN 77 ESDIS Standards Compliance 11-37
11.5.2 Checking for ESDIS Standards Compliance in Fortran 90 11-39
11.5.3 Checking for ESDIS Standards Compliance in C 11-41
11.5.4 Checking for ESDIS Standards Compliance in Ada 11-42
11.5.5 Prohibited Function Checker 11-45
11.5.6 Checking Process Control Files 11-49
11.5.7 Extracting Prologs 11-51
11.6 Compiling and Linking Science Software 11-54
11.6.1 Updating the Process Control File 11-55
11.6.2 Setting up a SDP Toolkit Environment 11-57
11.6.3 Compiling Status Message Facility (SMF) Files 11-63
11.6.4 Building Science Software with the SCF Version of the SDP Toolkit 11-64
11.6.5 Building Science Software with the DAAC Version of the SDP Toolkit 11-67
11.7 Running a PGE in a Simulated SCF Environment 11-69
11.7.1 Setting Up the Environment for Running the PGE 11-69
11.7.2 Running and Profiling the PGE 11-70
11.8 File Comparison and Data Visualization 11-73
11.8.1 Using the GUI HDF File Comparison GUI 11-73
11.8.2 Using the hdiff HDF File Comparison Tool 11-73
11.8.3 Using the ASCII File Comparison Tool 11-74
11.8.4 Using the Binary File Difference Assistant 11-75
11.8.5 Data Visualization 11-77
11.9 Science Software Integration and Test (SSI&T) Release RELEASE 4. 11-89
11.9.1 RELEASE 4 Architecture: Overview 11-90
11.9.2 Implications for SSI&T Functions 11-95
11.10 Using ECS Assistant to View ECS Science Data Server Database 11-96
11.10.1 Starting ECS monitor GUI: 11-97
11.10.2 Using ECS Assistant to View Database 11-99
11.10.3 Using Browser to View ECS Science PDPS/IOS Database 11-101

11.11 Installing ESDTs and Inserting Granules on the Science Data Server 11-102
11.11.1 Required Servers for Installing ESDT's 11-102
11.11.2. Installing/Removing (ESDT/DLL) using the Science Data Server Operator GUI 11-102
11.11.3 Granule Insert into Science Data Server 11-105
11.11.4 Acquire a Granule from Science Data Server 11-106
11.11.5 Add ESDTs through the ECS Assistant GUI 11-108
11.11.6 Using ECS Assistant to Manage ESDTs 11-109
11.12 Production Rules 11-114
11.12.1 Purpose and Scope 11-114
11.12.2 Overview of Production Rules 11-115
11.12.3 Production Rule Definitions 11-116
11.12.4 Combinations of Production Rules 11-127
11.12.5 Production Rules Technical Information Sources 11-128
11.12.6 Production_Rules_Syntax 11-128
11.12.7 Production Requests 11-129
11.13 PGE Registration and Test Data Preparation 11-191
11.13.1 PGE ODL Preparation 11-191
11.13.2 ESDT ODL Preparation 11-193
11.13.3 Update PDPS/SSIT Database with PGE Science Metadata 11-195
11.13.4 Operational Metadata 11-198
11.13.5 SSIT Operational Metadata Update GUI 11-200
11.13.6 Test Data Preparation and Insertion of Data Granules 11-201
11.13.7Creating a Target MCF (.met) for a Dynamic/Static Granule 11-202
11.13.8 Inserting Static Data Granules into the Data Server 11-204
11.13.9 Inserting Dynamic Data Granules to the Science Data Server 11-206
11.13.10 Science Server Archive Package (SSAP) 11-208
11.13.11 PGE Checkout 11-211
11.13.12 Placing the Science Software Executable (SSEP) on the Data Server 11-212
11.14 PGE Planning Processing and Product Retrieval 11-217
11.14.1 Using the Production Request Editor 11-217
11.14.2 Invoking the Production Request 11-218
11.14.3 Defining a New Production Request 11-220
11.14.3 Processing 11-224

11.14.4 Using the Production Planning Workbench 11-226
11.14.5 Monitoring Production in PDPS Subsystem 11-233
11.14.6 Using the Q/A Monitor..... 11-234
11.15 Postprocessing and General Investigation 11-236
11.15.1 Examining PGE Log Files 11-236
11.15.2 The Production History 11-241
11.16 Examining PDPS-Related Scripts and Message Files 11-242
11.16.1 Examining AutoSys JIL Scripts..... 11-243
11.16.2 Examining Application Log Files (ALOG) 11-243
11.17 DPREP..... 11-245
11.17.1 Introduction..... 11-245
11.17.2 SSI&T Activity for DPREP 11-245
11.17.3 DPREP Processes and Procedures..... 11-247
11.17.4 HowToRunDPREP 11-251
11.18 Learn more about SSI & T 11-260
11.18.1 References: 11-260
11.18.2 Server Node Names Convention:..... 11-263
11.18.3 A Handy Alias file to use while conducting SSI&T: 11-264
11.18.4 HOWTO_SSIT HELPFUL NOTES 11-265

12. Resource Planning

12.1 Resource Planning Process..... 12-1
12.2 Resource Editor..... 12-3
12.2.1 Launch the Resource Editor 12-3
12.2.2 Synchronize Resource Listings 12-5
12.2.3 Determine Actual Processing Resources 12-6
12.2.4 Add a Resource 12-11
12.2.5 Modify a Resource..... 12-15
12.2.6 Delete a Resource..... 12-15
12.3 Resource Scheduler 12-16
12.3.1 Launch the Resource Scheduler..... 12-16
12.3.2 Create a Resource Reservation Request 12-18
12.3.3 Edit a Resource Reservation Request 12-21

12.3.4 Validate or Reject a Resource Reservation Request 12-22
12.3.5 Approve a Resource Reservation Request..... 12-23
12.3.6 Commit Resource Reservation Requests 12-25
12.3.7 Review the Resource Timeline..... 12-25
12.3.8 Delete a Resource Reservation Request 12-28
12.4 Shut Down Resource Planning Applications 12-29

13. Production Planning

13.1 Production Request Editor 13-1
13.1.1 Launch the Production Request Editor..... 13-2
13.1.2 Create New Production Request 13-3
13.1.3 Edit/Modify Production Request 13-11
13.1.4 Delete Production Request 13-12
13.1.5 Review Data Processing Requests..... 13-13
13.1.6 Delete DPRs 13-16
13.1.7 Re-Generate Granules Affected by Loss of Files from the Archive 13-16
13.2 Production Planning Workbench..... 13-20
13.2.1 Launch the Production Strategies GUI..... 13-21
13.2.2 Define or Modify a Production Strategy..... 13-23
13.2.3 Launch the Planning Workbench and Planning Timeline GUIs..... 13-27
13.2.4 Create New Production Plan 13-29
13.2.5 Review Plan Timeline..... 13-32
13.3 Submitting or Withdrawing a Subscription Using the Subscription Editor 13-35
13.3.1 Submit or Withdraw a Subscription Using the Subscription Editor 13-35

14. Production Processing

14.1 Launch the AutoSys GUI Control Panel and Configure AutoSys Runtime Options... 14-2
14.1.1 Launching the AutoSys GUI Control Panel..... 14-2
14.1.2 Configuring AutoSys/AutoXpert Runtime Options..... 14-4
14.1.3 Configure Hardware Groups 14-6
14.2 Review Hardware Status 14-8
14.2.1 Review Hardware Status 14-8
14.2.2 Select Hardware Status View Options 14-10

14.3 Review DPR Dependencies 14-11
14.4 Review DPR Production Timeline 14-13
14.5 Review Alarms 14-15
14.5.1 Review Alarms 14-15
14.5.2 Select Alarms for Alarm Manager Display 14-18
14.6 Review Job Activities 14-20
14.6.1 Specify Job Selection Criteria 14-20
14.6.2 Review Job Activities Using the AutoSys Job Activity Console 14-23
14.7 Modify Job Status 14-24
14.7.1 Determine the Ownership of an AutoSys Job 14-25
14.7.2 Modify Job Status 14-26
14.7.3 Cancel a Sent Event 14-30
14.7.4 Perform Job Management Functions 14-31
14.8 Review AutoSys Activity and Job Dependency Logs 14-32
14.8.1 Review Activity Log 14-32
14.8.2 Review Job Dependency Log 14-34
14.9 Define and Run Monitors/Browsers 14-35
14.9.1 Define Monitors/Browsers 14-35
14.9.2 Run Monitor/Browser from the Monitor/Browser GUI 14-37
14.9.3 Run Monitor/Browser from the Command Shell 14-38
14.10 Modify Configuration Files 14-39
14.10.1 Change AutoSys Event Processor Database Maintenance Time 14-39
14.10.2 Modify the Maximum Number of Jobs in AutoSys 14-40

15. Quality Assurance

15.1 QA Monitor 15-1
15.1.1 DAAC Product QA with the QA Monitor 15-5
15.1.2 Updating QA Metadata 15-7
15.1.3 Production History 15-10
15.2 QA Metadata Update Tool 15-10

16. Ingest

16.1 Ingest Custom Software Items 16-1
16.2 ECS Ingest Tool 16-2
16.2.1 Starting the Ingest GUI 16-2
16.2.2 Viewing the Ingest History Log 16-3
16.2.3 Ingest History Log Reports 16-5
16.2.4 Monitoring/Controlling Ingest Requests 16-7
16.2.5 Suspending Ingest Requests 16-9
16.2.6 Resuming Ingest Requests 16-10
16.2.7 Ingest Operator Tools 16-10
16.2.8 Physical Media Ingest 16-16
16.3 Interactive Ingest Tool 16-22
16.3.1 Creating a Data Availability Notice (DAN) 16-22
16.3.2 Submitting an Ingest Request 16-25
16.3.3 Monitoring On-Going Request Status 16-26
16.4 Ingest Polling Process 16-27
16.4.1 Polling Ingest With Product Delivery Record 16-28
16.4.2 Polling Ingest Without Product Delivery Record 16-29
16.4.3 Ingest Information Notifications 16-29
16.4.4 Ingest Archive Verification 16-30
16.4.5 Deleting Files from the Ingest Polling Directory 16-31
16.5 Recovery from a Data Ingest Failure 16-32
16.5.1 Troubleshooting a Data Ingest Failure 16-33
16.5.2 Recovering from a Faulty DAN 16-34
16.5.3 Recovering from Exceeding the Volume Threshold 16-34
16.5.4 Recovering from Exceeding the Maximum Number of Concurrent Requests 16-35
16.5.5 Recovering from Insufficient Disk Space 16-36
16.5.6 Recovering from Exceeding the Expiration Date/Time Period 16-36
16.5.7 Recovering from File Transfer (ftp) Error 16-36
16.5.8 Recovering from Processing Errors 16-37
16.6 Document Ingest 16-38
16.6.1 Document Inserts From Within ECS 16-38
16.6.2 Scanning Documents 16-39
16.6.3 Accessing Scanned Documents 16-40
16.7 ECS Science Data Server Operator Tool 16-41

16.7.1 Starting the Science Data Server Operator GUI 16-41
16.7.2 View the Science Data Server Data Type Information 16-42
16.7.3 Add Science Data Server Data Types 16-43
16.7.4 View Science Data Server Descriptor Information 16-46
16.7.5 Monitor Science Data Server Request Processing 16-46
16.7.6 Change Science Data Server Polling Rate 16-48

17. Archive Management

17.1 Custom Software Items 17-1
17.2 COTS Software 17-2
17.2.1 AMASS 17-4
17.2.2 Distributive AML Server (DAS) 17-9
17.2.3 Automated Cartridge System Library Software (ACSLS) 17-9
17.3 Archive Configuration 17-10
17.4 Graphical User Interface (GUI) 17-13
17.4.1 Launching DSS GUIs Using UNIX Commands 17-14
17.4.2 Storage Management Control GUI 17-16
17.4.3 AMASS GUI 17-18
17.5 Command Line Functions 17-20
17.5.1 Formatting a Volume 17-23
17.6 Storing New Data in Archive Repository 17-24
17.6.1 Monitor Insertion of Data into the Archive 17-26
17.6.2 Monitor Insertion of Data into the Archive using AMASS 17-27
17.6.3 Monitor Archive Status Using ACSLS 17-27
17.6.4 Monitor Insertion of Data into the Archive using DAS 17-28
17.6.5 Fault Notification 17-28
17.6.6 Recovery from Failure to Store or Retrieve Data 17-30
17.6.7 Deleting Files From the Archive using the AMASS GUI 17-33
17.7 Archive Backup 17-33
17.7.1 AMASS Database Backup 17-34
17.7.2 ACSLS Database Backup and Restore 17-36
17.7.3 Backing Up Archive Data 17-39

17.8 Media Quality Control 17-45
17.8.1 Handling a Cartridge 17-46
17.8.2 Inspecting a Cartridge 17-47
17.8.3 Storing Environment for Cartridges 17-47
17.8.4 Operating Environment for Cartridges 17-48
17.8.5 Cleaning Cartridges 17-48
17.9 Archive Data Restoration after Failure 17-49
17.9.1 Data Recovery Process Overview 17-50
17.9.2 Data Recovery Process 17-52
17.9.3 Results of File and Granule Recovery 17-55
17.9.4 Manual Use of Backup Data for Recovery 17-56
17.9.5 Manual Data Recovery From Damaged Cartridge 17-58
17.9.6 SDSRV Procedures in Support of Data Recovery 17-62
17.10 Operations of Archive Media Interfaces 17-67
17.10.1 STK Media Interface 17-68
17.10.2 EMASS Media Interface Units 17-74
17.10.3 EMASS I/O UNIT 17-87
17.11 Archiving Application Log Files to AMASS 17-92
17.11.1 Configuration 17-93
17.11.2 ALOG File Archiving Procedure 17-94
17.11.3 ALOG File Retrieval from Archive 17-98

18. Data Distribution

18.1 Data Distribution Custom Software Items 18-2
18.2 ECS Data Distribution Operator Tool 18-2
18.2.1 Starting the Data Distribution Operator GUI 18-3
18.2.2 Monitoring/Controlling Data Distribution Requests 18-4
18.2.3 Configuring Data Distribution Polling and Error Retry Rate 18-6
18.2.4 Changing the Priority of Data Distribution Requests 18-6
18.2.5 Suspending Data Distribution Requests 18-8
18.2.6 Resuming Processing on a Suspended Data Distribution Request 18-9
18.2.7 Canceling a Data Distribution Request 18-10
18.3 Physical Media Operations 18-12

18.3.1 Loading and Unloading Tapes 18-12
18.3.2 Correcting Tape Faults 18-14
18.3.3 Labeling Tape Cartridges 18-14
18.4 Product Shipment 18-14
18.4.1 Performing Quality Control of Hard Media 18-15
18.4.2 Packaging Hard Media Products for Shipment 18-16
18.4.3 Mark Hard Media for Shipment 18-16
18.5 Recovery from a Data Distribution Failure 18-18
18.5.1 Troubleshooting a Data Distribution Failure 18-19
18.5.2 Recovering from Ddist Refresh Failure 18-19
18.5.3 Recovering from a Ddist Cancel Failure 18-19
18.5.4 Recovering from Ddist Set Priority Failure 18-20
18.5.5 Recovering from Ddist Suspend/Suspend All Failure 18-21
18.5.6 Recovering from Ddist Resume/Resume All Failure 18-21
18.5.7 Recovering from Ddist Mark Shipped Failure 18-21
18.5.8 Checking Database Connections 18-22
18.5.9 Display Data Distribution Error Logs 18-22
18.5.10 Missing E-mail Notification Preamble 18-23

19. User Services

19.1 ECS User Account Management 19-1
19.1.1 Retrieve User Account/Validate a User 19-2
19.1.2 Create a User Account 19-4
19.1.3 Account Creation from URL Registration 19-17
19.1.4 Edit/Modify an Existing Account 19-20
19.1.5 Deleting an ECS Account 19-36
19.1.6 Canceling an ECS Account 19-39
19.1.7 Changing an ECS User's Password 19-42
19.2 Processing an Order 19-44
19.2.1 Create a User Contact Log Record 19-45
19.2.2 Retrieve User Information 19-50
19.2.3 Locate Data Via Search and Order Tool 19-53
19.2.4 Request Price Estimate 19-53

19.2.5 Specify Order Details 19-53
19.2.6 Update User Contact Log 19-54
19.3 Canceling an Order 19-57
19.3.1 ECS Order Tracking 19-57
19.3.2 Canceling the Order 19-59
19.4 Fulfilling a Subscription 19-59
19.4.1 Fulfilling a Need for a One-Time Subscription 19-61
19.4.2 Fulfilling a Need for an Open-Ended Subscription 19-64
19.4.3 Returning a List of Subscriptions 19-67
19.4.4 Canceling a Subscription 19-69
19.5 Data Acquisition Request (DAR) Tool 19-71
19.5.1 Purpose of the DAR Tool 19-71
19.5.2 The DAR Tool User Interface 19-71
19.5.3 Create/Edit Request 19-72
19.5.4 Spatial Requirements 19-73
19.5.5 Temporal Requirements 19-73
19.5.6 Optional Screens 19-73
19.5.7 Resource Estimate 19-74
19.5.8 Modifying a DAR 19-78

20. Library Administration

20.1 Systems Operation Support Baselined (CDRL/DID) Document Maintenance 20-1
20.1.1 Library Administration Overview 20-1
20.1.2 Authoring Documents 20-1
20.1.3 Formatting Documents 20-1
20.2.3 Importing Documents 20-2
20.1.4 Exporting Documents 20-2
20.1.5 Metadata Maintenance 20-2
20.2 On-Site (DAAC) Baselined (CDRL/DID) Document Maintenance 20-2
20.2.1 Authoring Documents 20-2
20.2.2 Formatting Documents 20-2
20.2.3 Importing Documents 20-2
20.2.4 Exporting Documents 20-2

20.2.5 Metadata Maintenance 20-2
20.3 Maintenance of Baselined ECS Documents in Baseline Manager Tool 20-3
20.4 Document Metadata Insertion Subscription 20-3
20.5 Document Repository and Maintenance 20-3
20.5.1 Data Management Organization (DM0) 20-3
20.5.2 System Monitoring Center (SMC) 20-3
20.5.3 Configuration Management and Data Management 20-4
20.6 Retrieval of HTTP Formatted Documents 20-4
20.7 Systems Operation Support COTS Document & Software Maintenance 20-4
20.7.1 COTS Library Database 20-4
20.7.2 Document Access Control 20-5
20.7.3 COTS Library Reports 20-5
20.7.4 On-Site COTS Document & Software Maintenance 20-5

21. COTS Hardware Maintenance

21.1 COTS Hardware Maintenance - General 21-1
21.1.1 Corrective Maintenance 21-2
21.1.2 Preventive Maintenance 21-2
21.1.3 Configuration Management 21-2
21.1.4 COTS Hardware Maintenance Safety 21-2
21.1.5 Y2K Compliance 21-3
21.2 COTS Hardware Maintenance - Contract Information 21-3
21.2.1 COTS Hardware Maintenance Contract Database 21-3
21.3 Hardware Repairs - Standard 21-4
21.3.1 Hardware Problem Reporting 21-4
21.3.3 Hardware Corrective Maintenance Actions 21-5
21.3.4 Contract On-Site Hardware Maintenance 21-6
21.3.5 Return-to-Depot Support 21-10
21.4 Maintenance Spares 21-11
21.4.1 Use of Maintenance Spares 21-12
21.4.2 Return of Failed LRUs 21-13
21.5 Non-standard Hardware Support 21-13

21.5.1 Escalation of COTS Hardware Support Problem 21-14
21.5.2 Uneconomically Reparable Equipment 21-14

22. COTS Software Maintenance

22.1 Introduction 22-1
22.1.1 COTS Software Maintenance 22-2
22.1.2 Management of COTS Software Maintenance Contracts 22-2
22.1.3 Management of COTS Software Licenses 22-3
22.1.4 COTS Software Installation and Upgrades 22-3
22.1.5 Obtaining COTS Software Support 22-4
22.1.6 COTS Software Problem Reporting 22-5
22.2 Custom Software Maintenance 22-6
22.2.1 Implementation of Modifications 22-8
22.2.2 Test Plans and Procedures 22-9
22.2.3 Custom Software Installation 22-11
22.2.4 Obtaining Software Support 22-20
22.2.5 Science Software 22-22

23. Property Management

23.1 Receipt of Equipment and Software from Vendor 23-1
23.2 Receipt of Equipment and Software from the ILS Property Administrator 23-3
23.3 Equipment Tagging 23-3
23.4 Property Records and Reporting 23-4
23.4.1 Maintaining Property Records 23-5
23.4.2 Reporting Loss, Theft, Damage or Destruction 23-5
23.5 Equipment Relocation 23-5
23.5.1 Intra-site Relocation 23-5
23.5.2 Inter-site Relocation 23-6
23.5.3 External Transfers 23-6
23.6 Inventories and Audits 23-6
23.7 Storage 23-7
23.7.1 Segregation Requirements 23-7

23.7.2 Stock Rotation 23-7
23.7.3 Physical Security 23-7
23.8 Packing and Shipping 23-7
23.9 Electrostatic Discharge (ESD) Program 23-8

24. Installation Planning

24.1 Responsibilities 24-1
24.2 Process Description 24-1
24.3 Maintenance of Facility and Hardware Diagrams 24-2
24.4 Maintenance of LAN Cable Management Scheme 24-3
24.5 Maintenance of SCSI Cable Management Scheme 24-3

25. COTS Training

25.1 Requesting COTS Training 25-1
25.2 Coordinating COTS Training 25-2
25.3 Canceling/Rescheduling COTS Training 25-3
25.4 Maintenance of COTS Training Records 25-3
25.5 Contractor COTS Training Funds Accounting 25-3

26. Earth Science On-Line Directory Administration

26.1 Accessing ESOD 26-4
26.2 ESOD Moderation 26-4
26.3 ESOD Administration 26-6
26.3.1 Create a Moderation Group 26-6
26.3.2 Update a Moderation Group 26-7
26.3.3 Delete a Moderation Group 26-9

27. Inventory Logistical Management (ILM)

27. 1 XRP-II (Inventory Logistical Management {ILM}) 27-1

27.2 ILM Operator Functions 27-1
27.3 General Information 27-2
27.3.1 Using XRP 27-2
27.3.2 ILM System 27-2
27.4 Quick Start Using ILM 27-3
27.4.1 Invoking ILM from the Command Line Interface 27-3
27.5 ECS Management System Main Menu 27-4
27.6 Commands and Functionality 27-6
27.6.1 Cursor Motion 27-7
27.6.2 Next or Prior 27-7
27.6.3 View Command 27-7
27.6.4 Find Command 27-7
27.6.5 Go Command 27-7
27.6.6 Select Command 27-8
27.6.7 Selection Criteria 27-8
27.6.8 Sort Command 27-9
27.6.9 Note Command 27-9
27.6.10 Add/Insert 27-10
27.6.11 Modify Command 27-10
27.6.12 Delete Command 27-10
27.6.13 Write, Execute Command 27-10
27.6.14 Items Command 27-10
27.6.15 Help Command 27-10
27.6.16 More Command 27-10
27.6.17 Quit Command 27-10
27.6.18 Zoom Command 27-10
27.6.19 Left, Right, Justify Commands 27-11
27.6.20 Tag, Untag Command 27-11
27.6.21 Report Command 27-11
27.7 EIN Menu 27-12
27.7.1 EIN Structure Manager 27-13
27.7.2 Item Page of the Structure Manager 27-14
27.7.3 Checking the relationship status between Parent and its Children 27-15
27.7.4 EIN Manager 27-16

27.8 ILM Report Menu 27-19
27.8.1 EIN Structure Reports 27-19
27.9 Install/Receipt Report 27-22
27.10 Installation Summary Reports 27-24
27.11 Maintenance Menu 27-25
27.11.1 Work Order Entry Screens 27-26
27.11.2 Item Page for Work Order Entry Screens 27-29
27.12 Work Order Modification Screens 27-32
27.13 Delay Times Page for Work Order Modification Screens 27-34
27.13.1 Chargeable Hours Page for Work Order Modification Screens 27-36
27.13.2 Items Page for Work Order Modification Screens 27-36
27.14 Preventative Maintenance Items Screens 27-40
27.14.1 Generate Preventative Maintenance Orders 27-41
27.15 Maintenance Work Order Reports Screens 27-43
27.16.1 Work Order Status Reports Screens 27-45
27.17 Maintenance Codes Screens 27-48
27.17.1 Maintenance Contracts Screens 27-48
27.17.2 Authorized Employees Screens 27-49
27.18 ILM Master Menu 27-50
27.18.1 Employee Manager Screens 27-51
27.18.2 Hardware/Software Codes Screens 27-53
27.18.3 OEM Part Numbers 27-54

28. Procedure for Removing Archived Files from the PDR Server Staging Areas

28.1 Overview 28-1
28.1.1 PDR Server MODAPS Area 28-1
28.1.2 PDR Server DAAC Areas 28.1
28.2 PDR Server File Deletion Procedure

Figures

3.7-1 ECS Assistant GUI 3-79
3.7-2 Subsystem Manager GUI 3-80
3.7-3 Server Monitor GUI 3-84
3.7-4 cdsping GUI 3-85
4.13-1 Technical Directive 4-26
4.13-2 Technical Directive 4-27
8.3.2-1 Trouble Ticket E-mail Template 8-15
9.2.4-1 Configuration Change Request (CCR) Form 9-5
9.2.4-2 ECS CCR Impact Analysis 9-9
9.2.4-3 ECS M&O CCR Impact Summary 9-10
9.2.4-4 Work-Flow Diagram for SEO CM Administrator 9-14
9.2.4-5 Work-Flow Diagram for Site-level CM Administrator 9-17
9.2.4-6 Deviation/Waiver Form 9-21
9.5.4-1 CDRL Item Submission Criteria Flow 9-28
9.7.4-1 SW Transfer Functional Flow 9-35
9.7.4-2 SW Installation Functional Flow 9-36
9.7.4-3 Detailed Points of View 9-38
9.7.4-4 Detailed Points of View 9-39
9.7.4-5 Detailed Points of View 9-40
9.9-1 Hierarchy of Control Items for Baseline Management 9-65
9.9-2 Design (CIL/ CAL) View 9-67
9.9-3 Operational (Logical) View 9-68
9.9-4 Common Nomenclature for Baseline Manager Design View Items 9-70
9.9-5 Common Nomenclature for Baseline Manager Operational View 9-72
9.9-6 Common Nomenclature for Baseline Manager Operational View - DETAILS 9-76
9.9-7 Bill of Materials Report 9-125
9.9-8 Indented Bill of Materials 9-126

9.9-9 Summarized Bill Report 9-127
9.9-10 Multilevel Where-Used Display 9-128
9.9-11 Multilevel Where-Used Report..... 9-129
9.9-12 Configuration Items List - One..... 9-130
9.9-13 Configuration Items List - Two 9-131
9.9-14 Configured Articles List..... 9-132
9.9-15 Version Description Report..... 9-133
9.9-16 Site Baseline Report 9-134
9.9-17 Change History Report 9-135
9.9-18 BOM Comparison Report..... 9-136
10.2-1 The ESDT Generation Process 10-4
10.1-2 Steps in ESDT Development 10-6
10.3-1 Tools and Supporting Information Flows..... 10-7
11.1.3.1-1 SSI&T Process Flow Diagrams 1 11-4
11.1.3.1-2 SSI&T Process Flow Diagrams 2 11-5
11.2-1 SSIT Manager Window 11-17
11.2.3-1 SSIT Manager Window - Tools Menu 11-20
11.4.2-1 ClearCase File Browser Screen (Main Screen) 11-28
11.5.5-1 Prohibited Function Checker 11-47
11.5.5-2 Invoking the Prohibited Function Checker 11-48
11.5.6-1 Process Control File Checker GUI 11-49
11.5.7-1 Prolog Extractor Sample Run. 11-54
11.9.1-1 ECS Communications /Interetworking Subsystems..... 11-95
11.10-1 Database Login GUI 11-97
11.10-2 DB Viewer GUI..... 11-98
11.10-3 Server Monitor GUI 11-100
11.10-4 cdsping GUI 11-101
11.11.1 Science Dataserver Operator GUI. 11-104
11.11-6 ESDT Manager GUI..... 11-111

11.12.7.1-1 Production Request Editor Flow 11-130
11.12.8.8-1 Example of the Basic Temporal Production Rule 11-138
11.12.8.12-1 Example of the Advanced Temporal Production Rule 11-146
11.12.8.14-1 Example of the Alternate Input Production Rule..... 11-149
11.12.8.15-1 Example of the Minimum/Maximum Number of Granules Production Rule . 11-157
11.12.8.17-1 Example of the Optional DPRs Production Rule 11-162
11.12.8.19-1 Example of the Intermittent Activation Production Rule 11-165
11.12.8.20-1 Example of the Metadata Checks and Query Production Rules..... 11-167
11.12.8.24-1 Example of the Spatial Query Production Rule 11-174
11.12.8.27-1 Example of the Tiling Production Rule..... 11-178
11.12.8.29-1 Example of Closest Granule Production Rule 11-183
11.12.8.31-1 Example of the Orbital Processing Production Rule..... 11-186
11.13.5-1 SSIT Database Operational Metadata Update GUI – SELECT view..... 11-200
11.14.3-1 Production Request Editor Introductory GUI 11-220
11.14.3-2 Production Request Editor GUI(Planning)..... 11-223
11.14.3-3 PR List GUI. 11-225
11.14.4-1 Planning Workbench GUI..... 11-229
11.17.1-1 DPREP File Processes 11-246
14.1-1 AutoSys Hardware Group File Example 14-7
14.8-1 Sample Activity Log..... 14-33
14.8-2 Sample Job Dependency Log..... 14-34
14.9-1 Sample Browser Screen 14-35
15-1 QA Metadata Update Process..... 15-1
15-2 QA Monitor GUI - QRU Data Tab 15-5
15-3 QA Monitor - Visualize Data 15-7
15-4 QA Monitor Granule Parameters Window..... 15-10
17-2 FSMS Software Configuration 17-3
17.3-1 Archive Hardware and Software Configuration for STK..... 17-11
17.3-2 RAID Configuration 8/19/97 17-12

17.3-3 Archive Hardware and Software Configuration for EMASS 17-13
17.10-1 Cartridge Insertion Order into CAP 17-70
17.10-2 EIF Keypad Configuration 17-75
17.10-3 EMASS I/O Unit Layout 17-87
20.1.1-1 EDHS Home Page, ECS Baseline Information, and System Monitoring Center.... 20-1
21.5.2-1 DAAC Hardware Problem Reporting (A) 21-15
21.5.2-2 On-Site Maintenance Support (B) 21-16
21.5.2-3 On-Site Maintenance Support (B-1)..... 21-17
21.5.2-4 On-Site Maintenance Support or Site Depot Support (B-2)..... 21-18
21.5.2-5 On-site Maintenance Support or Site Depot Support (B-3) 21-19
21.5.2-6 Equipment Returned to Depot for Service (C)..... 21-20
21.5.2-7 Equipment Advance Replacement (D)..... 21-21
21.5.2-8 Replacement Item Provided by EDF Centrally Stocked Spares Inventory (E)..... 21-22
21.5.2-9 Time & Material Support (F)..... 21-23
22.2-1 Typical CCR Telecon Agenda 22-9
22.2.5-1 Licensing..... 22-23
22.2.5-2 COTS Software Patches and Upgrades (1 of 2) 22-24
22.2.5-3 COTS Software Patches and Upgrades (2 of 2) 22-25
22.2.5-4 COTS License Relocation/Installation (1 of 2) 22-26
22.2.5-5 COTS License Relocation/Installation (2 of 2) 22-27
22.2.5-6 COTS Software Problem Reporting (1 of 2) 22-28
22.2.5-7 COTS Software Problem Reporting (2 of 2) 22-29
23.2-1 ECS Property Tags (actual size) 23-4
23.9-1 Receipt of Equipment and Software from Vendor..... 23-9
23.9-2 Procedure for Completion of the Inventory Worksheet..... 23-10
23.9-3 Procedure for Completion of the Non Conforming Product Report (NCR)..... 23-11
23.9-4 Equipment Relocation (1 of 3) 23-12
23.9-5 Equipment Relocation (2 of 3) 23-13
23.9-6 Equipment Relocation (3 of 3) 23-14

24.5-1 Installation Process (1 of 3)……24-4
24.5-2 Installation Process (2 of 3)……24-5
24.5-3 Installation Process (3 of 3)……24-6
27.5-1 ECS Management System Main Menu……27-5
27.7-1 EIN Menu …… 27-12
27.7.1-1 EIN Structure Manager Screen…… 27-13
27.7.2-1 EIN Structure Manager Items page screen …… 27-14
27.7.4-1 EIN MANAGER Screen…… 27-16
27.8-1 Report Menu …… 27-19
27.8.1-1 EIN Structure Reports Screen…… 27-20
27.9-1 Install/Receipt Report Screen…… 27-22
27.10-1 Installation Summary Reports Screen …… 27-24
27.11-1 Maintenance Menu Screen …… 27-26
27.11.1-1 Work Order Entry Screen …… 27-27
27.11.2-1 Item Page for Work Order Entry Screen …… 27-29
27.12-1 Work Order Modification Screen …… 27-33
27.13-1 Delay Times Page for Work Order Modification …… 27-35
27.13.-1.Chargeable Hours Page for Work Order Modification …… 27-36
27.13.2-1 Items Page for Work Order Modification…… 27-37
27.14-1 Preventative Maintenance Items Screen…… 27-40
27.16.1 Work Order Status Reports Screen …… 27-45
27.17-1 Maintenance Codes Screen…… 27-48
27.17.1-1 Maintenance Contracts Screen …… 27-48
27.17.2-1 Authorized Employees Screen…… 27-49
27.18-1 ILM Master Menu …… 27-50
27.18.1-1 Employee Manager Screen…… 27-51
27.18.1.2-1 Inventory Location Manager Screen …… 27-52
27.18.2-1 Hardware/Software Codes Screen …… 27-53
27.18.3-1 OEM Part Numbers Screen …… 27-54

Tables

3.1 System Administration - Activity Checklist 3-1
3.1-1 Startup/Shutdown - Activity Checklist.....3-2
3.1-2 Cold System Startup - Machine Boot Sequence 3-3
3.1-3 Warm System Startup - Quick-Step Procedures.....3-4
3.1-4 Normal System Shutdown - Quick-Step Procedures.....3-6
3.1-5 Emergency System Shutdown - Quick-Step Procedures 3-8
3.1-6 Server System Shutdown - Quick-Step Procedures 3-8
3.2-1 Incremental Backup - Activity Checklist.....3-9
3.2-2 Perform Incremental Backup - Quick-Step Procedures 3-12
3.2-3 Full Backup - Activity Checklist.....3-13
3.2-4 Perform Full Backup - Quick-Step Procedures.....3-15
3.2-5 File Restore - Activity Checklist 3-16
3.2-6 Restore a File - Quick-Step Procedures 3-18
3.2-7 Complete System Restore - Activity Checklist 3-19
3.2-8 Restore a Partition - Quick-Step Procedures.....3-22
3.2-9 Tape Handling - Activity Checklist 3-22
3.2-10 Labeling Tapes - Activity Checklist 3-23
3.2-11 Label Tapes - Quick-Step Procedures.....3-25
3.2-12 Indexing Tapes - Activity Checklist 3-25
3.2-12 Index Tapes - Quick-Step Procedures.....3-28
3.3-1 System Log Maintenance - Quick-Step Procedures 3-30
3.4-1 Adding a User - Activity Checklist.....3-31
3.4-2 Deleting a User - Activity Checklist 3-32
3.4-3 Change a User Account Configuration - Activity Checklist.....3-33
3.4-7 Changing User Access Privileges - Activity Checklist 3-34
3.4-9 Changing a User Password - Activity Checklist.....3-35
3.4-11 Checking a File/Directory Access Privilege Status - Activity Checklist.....3-35

3.4-12 Check a File/Directory Access Privilege Status - Quick-Step Procedures 3-37
3.4-13 Changing a File/Directory Access Privilege - Activity Checklist..... 3-37
3.4-14 Change a File/Directory Access Privilege - Quick-Step Procedures 3-40
3.4-15 Moving a User's Home Directory - Activity Checklist 3-40
3.5-1 Installing a New Workstation - Activity Checklist..... 3-41
3.5-2 Hardware Preparation - Activity Checklist..... 3-42
3.5-3 Prepare for Network Configuration - Quick-Step Procedures..... 3-43
3.5-4 Report to Inventory - Quick-Step Procedures 3-44
3.5-5 Install the Solaris 2.5.1 Operating System - Quick-Step Procedures 3-47
3.5-6 Install the HP-UX 10.01 and 10.10 Operating System - Quick-Step Procedures 3-50
3.5-7 Install the IRIX 6.2 Operating System - Quick-Step Procedures..... 3-53
3.5-8 Configure the NCD - Quick-Step Procedures 3-57
3.5-9 Install Custom Software - Quick-Step Procedures..... 3-59
3.5-10 Install COTS Software - Quick-Step Procedures 3-60
3.5-11 Reboot - Quick-Step Procedures..... 3-61
3.5-12 Reboot the NCD - Quick-Step Procedures 3-62
3.5-13 Log In - Quick-Step Procedures..... 3-63
3.5-14 Test Environment - Activity Checklist..... 3-64
3.5-15 Test Environment - Quick-Step Procedures 3-65
3.6-1 Configuring Initial Cell - Quick-Step Procedures..... 3-67
3.6-2 Configuring DTS Servers - Quick-Step Procedures 3-68
3.6-3 Configuring Additional CDS Servers - Quick-Step Procedures..... 3-71
3.6-4 Configuring Security and CDS Client Systems - Quick-Step Procedures 3-72
3.6-5 Configuring DTS Clerks - Quick-Step Procedures 3-73
3.6-6 Configuring GDA Servers - Quick-Step Procedures 3-74
3.6-7 Creating a Security Server Replica - Quick-Step Procedures..... 3-75
3.6-8 Unconfiguring DCE Client - Quick-Step Procedures..... 3-76
3.7-1 ECSAssistant - Activity Checklist 3-77
3.7-2 Assumptions - Checklist..... 3-77

4.2-1 SQL Server General Definitions....4-5
4.2.2-1 SYBASE Directory Structure....4-6
4.2.3-1 SQL Server Parameters and Options 4-7
4.10-1 Backup and Recovery Definitions 4-17
4.10-2 Automatic Backup Components 4-20
6.1-1 Network Administration - Activity Checklist....6-2
6.1-2 Starting NNM (Network Node Manager) - Quick-Step Procedures 6-4
6.1-3 Adding a Network Object - Quick-Step Procedures 6-5
6.1-4 Adding a Segment Object- Quick-Step Procedures....6-6
6.1-5 Adding a Node Object- Quick-Step Procedures....6-8
6.1-6 Adding an IP Interface Object- Quick-Step Procedures....6-9
6.1-7 Viewing the Current Network and System Configuration - Quick-Step Procedures....6-10
6.1-8 Viewing Network Address Information- Quick-Step Procedures....6-11
6.1-9 Viewing How Traffic is Routed on a Network- Quick-Step Procedures....6-12
6.1-10 Viewing the Services Available on a Node- Quick-Step Procedures 6-13
7.1-1 Monitoring - Activity Checklist 7-1
7.1-2 Starting/Monitoring NNM - Quick-Step Procedures 7-4
7.1-3 Verify - Quick-Step Procedures 7-6
7.1-4 Operational Status Colors 7-7
7.1-5 Color Alerts- Quick-Step Procedures....7-8
7.1-6 Event Notifications - Quick-Step Procedures....7-10
7.1-7 Rediscovery - Quick-Step Procedures 7-11
7.2-1 Disk Event Configuration 7-15
7.2-2 Security Event Configuration....7-16
7.2-3 Network Event Configuration....7-17
7.2-4 System Event Configuration 7-19
7.2-5 Printer Event Configuration....7-20
8.3-1 Trouble Ticket System - Activity Checklist 8-9
8.3.1-1 RelB-Trouble Ticket Field Description 8-11

8.3.6-1 Table of Access Control Groupings 8-17
8.4.1-1 Trouble Ticket HTML Submit Screen Field Description 8-19
8.4.3-1 Trouble Ticket HTML List Screen Field Description 8-20
8.4.4-1 Trouble Ticket HTML Detailed Screen Field Description 8-21
8.5-1 Example of Emergency Change Procedure 8-23
9.7.4-1 Data Activity for Workflow at the SMC 9-41
9.7.4-2A Data Activity for Workflow at the DAAC 9-42
9.7.4-2B Data Activity for Workflow at the DAAC 9-43
9.7.4-2C Data Activity for Workflow at the DAAC 9-44
9.8.4-1 Submit Record Field Descriptions 9-50
9.8.4-2 Submit a CCR - Quick-Step Procedures 9-52
9.8.5 Change State of a CCR - Quick-Step Procedures 9-53
9.8.5-1 Assign-Eval Field Descriptions 9-53
9.8.5-2 Assign-Implement Field Descriptions 9-55
9.8.5-3 Assign-Verify Field Descriptions 9-56
9.8.5-4 Verify State Field Descriptions 9-57
9.8.5-5 Close State Field Descriptions 9-57
9.8.6 Modify a CCR - Quick-Step Procedures 9-58
9.8.7 Print a CCR - Quick-Step Procedures 9-59
9.8.9 Reports 9-60
9.9-1 Common ECS Operator Functions Performed with XRP-II 9-78
9.9-2 Screen Allocations to Tasks Groups 9-80
9.9-3 All Control Items Field Description 9-91
9.9-4 Hardware Items Only Field Description 9-94
9.9-5 Items Only Field Description 9-95
9.9-6 Document Items Only Field Description 9-96
9.9-7 Host Items Only Field Description 9-97
9.9-8 Partition Items Only Field Description 9-98
9.9-9 Engineering Change Entry Field Description 9-100

9.9-10 Engineering Change Approval Field Description..... 9-101
9.9-11 Configuration Items List - One Menu Screen Field Description..... 9-105
9.9-12 Change History Field Description 9-106
9.9-13 BOM Comparison Reports Field Description..... 9-107
9.9-14 Vendor Master Maintenance Field Description..... 9-108
9.9-15 Control Item Interdependency Maintenance Field Description..... 9-109
9.9-16 Control Item Deployment Maintenance Field Description..... 9-110
9.9-17 Responsible Engineer Field Description..... 9-111
9.9-18 Serial Numbers Maintenance Field Description 9-111
9.9-19 System Parameters Maintenance Field Description..... 9-112
9.9-20 Transaction Log Field Description 9-113
9.9-21 XRP II GUI/CHUI Field Description 9-113
9.9-22 Site MasterMaintenance Field Description..... 9-115
9.9-23 Machine Network Maintenance Field Description 9-115
9.9-24 Commodity Code Maintenance Field Description..... 9-116
9.9-25 Export Release Record Field Description..... 9-117
9.9-26 Export Site-Unique Changes Field Description..... 9-117
9.9-27 Export SMC Data Field Description..... 9-118
9.9-28 Tunable Parameters..... 9-119
9.9.29 Command Line Interfaces..... 9-121
9.9-30 Outputs 9-122
9.9-31 Reports..... 9-124
11.6.2-1 SDP Toolkits used by the SGI science processors..... 11-59
11.9.3.1 Major SSI&T Functions within VERSION 2.0 11-96
11.12.3.2-1 Basic Temporal..... 11-118
11.12.3.4-1 Alternate Inputs 11-119
11.12.3.6-1 Tiling..... 11-120
11.12.3.7-1 Subsetting & Subsampling..... 11-121
11.12.3.7-1 Level 0 to Level 1A..... 11-122

11.12.3.9-1 Intermittent Execution 11-123
11.12.3.10-1 Metadata-based Conditional Activation 11-125
11.12.8.5-1 Extract of PGE Metadata ODL File Template Showing Alternate Inputs 11-154
12.1-1 Resource Planning - Activity Checklist 12-2
12.2-1 Launch the Resource Editor - Quick-Step Procedures 12-5
12.2-2 Synchronize Resource Listings - Quick-Step Procedures 12-6
12.2-3 Determine Actual Processing Resources - Quick-Step Procedures 12-11
12.2-4 Add a Resource - Quick-Step Procedures 12-12
12.2-5 Modify a Resource - Quick-Step Procedures 12-15
12.2-6 Delete a Resource - Quick-Step Procedures 12-16
12.3-1 Launch the Resource Scheduler - Quick-Step Procedures 12-18
12.3-2 Create a Resource Reservation - Quick-Step Procedures 12-19
12.3-3 Frequency Qualifiers 12-21
12.3-4 Edit a Resource Reservation Request - Quick-Step Procedures 12-22
12.3-5 Validate or Reject a Resource Reservation Request - Quick-Step Procedures 12-23
12.3-6 Approve a Resource Reservation Request - Quick-Step Procedures 12-25
12.3-7 Commit Resource Reservation Requests - Quick-Step Procedures 12-25
12.3-8 Review the Resource Timeline - Quick-Step Procedures 12-28
12.3-9 Delete a Resource Reservation Request - Quick-Step Procedures 12-29
12.4-1 Shut Down Resource Planning Applications - Quick-Step Procedures 12-31
13.1-1 Production Request - Activity Checklist 13-1
13.1-2 Launch the Production Request Editor - Quick-Step Procedures 13-3
13.1-3 Create New Production Request - Quick-Step Procedures** 13-10
13.1-4 Edit/Modify Production Request - Quick-Step Procedures 13-12
13.1-5 Delete Production Request - Quick-Step Procedures 13-13
13.1-6 Review Data Processing Requests - Quick-Step Procedures 13-15
13.1-7 Delete DPRs - Quick-Step Procedures 13-16
13.1-8 Re-Generate Granules Affected by Loss of Files from the Archive - Quick-Step Procedures 13-20

13.2-1 Production Planning - Activity Checklist 13-21

13.2-2 Launch the Production Strategies GUI - Quick-Step Procedures 13-22

13.2-3 Define a Production Strategy - Quick-Step Procedures 13-26

13.2-4 Launch the Planning Workbench - Quick-Step Procedures 13-29

13.2-5 Create New Production Plan - Quick-Step Procedures 13-32

13.2-6 Review Plan Timeline - Quick-Step Procedures 13-34

13.3-1 Subscription - Activity Checklist 13-35

13.3-2 Submit/withdraw Subscription - Quick-Step Procedures 13-38

14.1-1 Production Planning - Activity Checklist 14-1

14.1-2 Launch AutoSys GUI Control Panel - Quick-Step Procedures 14-4

14.1-3 Runtime Options Table 14-5

14.1-4 Configure AutoSys Runtime Options - Quick-Step Procedures 14-6

14.1-5 Configure AutoSys Hardware Groups - Quick-Step Procedures 14-8

14.2-1 Review Hardware Status Using AutoSys HostScape - Quick-Step Procedures..... 14-10

14.2-2 Change the Hardware Status View in AutoSys HostScape -
Quick-Step Procedures 14-11

14.3-1 Review DPR Dependencies in AutoSys JobScape - Quick-Step Procedures 14-13

14.4-1 Review the DPR Production Timeline in AutoSys TimeScape -
Quick-Step Procedures 14-14

14.5-1 Review Alarms Using the AutoSys Alarm Manager - Quick-Step Procedures..... 14-16

14.5-2 AutoSys Alarms 14-17

14.5-3 Select Types of Alarms to be Displayed on the AutoSys Alarm Manager -
Quick-Step Procedures 14-20

14.6-2 Review Job Activities Using the AutoSys Job Activity Console -
Quick-Step Procedures 14-24

14.7-1 Determine the Ownership of an AutoSys Job - Quick-Step Procedures..... 14-26

14.7-2 Modify Job Status – Quick-Step Procedures 14-28

14.7-3 UNIX Signals 14-29

14.7-4 Cancel a Sent Event - Quick-Step Procedures 14-31

14.7-5 Perform Job Management Functions - Quick-Step Procedures 14-32

14.8-1 Review Activity Log - Quick-Step Procedures 14-33

14.8-2 Review Job Dependency Log - Quick-Step Procedures 14-35

14.9-1 Define Monitors/Browsers - Quick-Step Procedures..... 14-37

14.9-2 Run Monitor/Browser from the Monitor/Browser GUI - Quick-Step Procedures.... 14-38

14.9-3 Run Monitor/Browser from the Command Shell - Quick-Step Procedures..... 14-38

14.10-1 Change AutoSys Event Processor Database Maintenance Time - Quick-Step Procedures 14-39

14.10-2 Modify the Maximum Number of Jobs in AutoSys - Quick-Step Procedures 14-41

15.1-1 Common Operator Functions Performed with the QA Monitor 15-2

15.1-2 QA Monitor GUI Fields..... 15-2

15.1-3 QA Monitor GUI Pushbuttons 15-3

15.1-4 QA Metadata Attributes..... 15-8

16.2-1 ECS Ingest Tool - Activity Checklist 16-2

16.2-2 Starting Ingest Operator GUI - Quick-Steps 16-3

16.2-3 View Ingest History Log - Quick-Steps..... 16-5

16.2-4 Ingest History Log Reports - Quick-Steps..... 16-7

16.2-5 Monitoring/Controlling Ingest Requests - Quick-Steps..... 16-9

16.2-6 Suspending Ingest Requests - Quick-Steps..... 16-9

16.2-7 Resuming Ingest Requests - Quick-Steps..... 16-10

16.2-8 Ingest Operator Tools - Activity Checklist 16-11

16.2-9 Modify External Data Provider/ Interactive User Information - Quick-Steps 16-13

16.2-10 Modify System Parameters - Quick-Step Procedures 16-15

16.2-11 File Transfer- Quick-Step Procedures 16-16

16.2-12 Physical Media Ingest Types..... 16-17

16.2-13 Physical Media Ingest - Activity Checklist..... 16-17

16.2-14 Performing Media Ingest from 8mm Tape - Quick Steps Procedures 16-19

16.2-15 Performing Media Ingest from D3 Tape - Quick Steps Procedures..... 16-21

16.3-1 Interactive Ingest - Activity Checklist..... 16-22

16.3-2 Creating Data Availability Notice (DAN) - Quick-Step Procedures..... 16-24

16.3-3 Submitting an Ingest Request - Quick-Step Procedure 16-26
16.3-4 Monitoring On-Going Request Status - Quick-Step Procedures 16-27
16.4-1 Ingest Polling - Activity Checklist 16-28
16.4-2 Ingest Information Notification - Quick-Step Procedures 16-30
16.4-3 Ingest Archive Verification - Quick-Step Procedures 16-31
16.4-4 Deleting Files From the Ingest Polling Directory - Quick-Steps 16-32
16.5-1 Recovery from a Data Ingest Failure - Activity Checklist 16-33
16.6-1 Scanning Documents - Quick-Step Procedures 16-40
16.6-2 Opening Tiffs File - Quick-Step Procedures 16-40
16.7-1 ECS Science Data Server Operator Tool - Activity Checklist 16-41
16.7-2 Starting Science Data Server Operator GUI - Quick-Steps 16-42
16.7-3 View Science Data Server Data Type Information - Quick-Steps 16-43
16.7-4 Add Science Data Server Data Types - Quick-Steps 16-45
16.7-5 View Science Data Server Descriptor Information - Quick-Steps 16-46
16.7-6 Monitor Science Data Server Request Processing - Quick-Steps 16-48
16.7-7 Change Science Data Server Polling Rate - Quick-Steps 16-49
17.2-1 COTS Software Activity Checklist 17-4
17.2-2 Rebooting the AMASS Database 17-8
17.2-3 FSMS Media Types 17-10
17.4-1 Graphical User Interface- Activity Checklist 17-14
17.4-2 Launching DSS GUIs Using UNIX Commands 17-16
17.4-3 Common ECS Operator Functions Performed with This Tool 17-17
17.4-4 Starting The AMASS GUI 17-19
17.5-1 Common ECS Operator Functions Performed with AMASS 17-21
17.5-2 Command Line Functions - Activity Checklist 17-23
17.5-3 Formatting a Tape Volume 17-24
17.5-4 Formatting an Optical Disk Volume 17-24
17.6-1 AMASS Monitoring Commands 17-27
17.6-2 ACSLS Monitoring Commands 17-28

17.6-3 DAS Monitoring Commands 17-28
17.6-4 Fault Notification-Quick-Step Procedure 17-29
17.6-5 Recovery from Failure to Store or Retrieve Data Quick-Step 17-31
17.7-1 Archive Data Backup- Activity Checklist 17-34
17.7-2 AMASS Backup Command-Quick-Step Procedure 17-35
17.7-3 AMASS Backup Command-Quick-Step Procedure 17-36
17.7-3 Checkpoint and redo log files..... 17-37
17.7-4 Create ACSLS Back Up - Quick-Step Procedures 17-38
17.7-5 Create ACSLS Restore - Quick-Step Procedures 17-39
17.7-6 Generate List of Data to be Backed Up - Quick-Step Procedures 17-41
17.7-7 Create Local Back Up - Quick-Step Procedures 17-42
17.7-8 Creating AMASS Off-Site Backup Tapes 17-44
17.7-9 Create Off-Site Back Up - Quick-Step Procedures 17-45
17.8-1 Media Quality Control - Activity Checklist 17-46
17.8-2 Cleaning Cartridges 17-49
17.9-1 Archive Data Recovery/Restoration - Activity Checklist 17-50
17.9-2 Manual Data Recovery from Local Backup Tapes 17-57
17.9-3 Manual Data Recovery from Off-Site Backup Tapes 17-58
17.9-4 D3 Cartridge Data Recovery - Quick-Step Procedures 17-62
17.10-1 Operations of Archive Media Interfaces - Activity Checklist 17-67
17.10-2 Manual Insertion using CAP as an Access Door 17-69
17.10-3 Manual Insertion Using bulkinlet 17-71
17.10-4 Automatic Insertion of STK Media 17-72
17.10-5 Commanding Ejection of STK Media 17-73
17.10-6 Manually Removing STK Media From the LSM 17-74
17.10-7 Inserting All Media Types into EIF - Quick-Step Procedure 17-77
17.10-8 Automatically Loading EMASS Archive Media - Quick-Step Procedure 17-79
17.10-9 Manually Loading EMASS Archive Media - Quick-Step Procedure 17-81
17.10-10 Ejecting EMASS Archive Media Using AMASS - Quick-Step Procedure 17-82

17.10-11 Ejecting EMASS Archive Media Using DAS - Quick-Step Procedure..... 17-82
17.10-12 Parameter Explanation 17-83
17.10-13 Media Types 17-83
17.10-14 Removing All Ejected Media Types from EIF - Quick-Step Procedure..... 17-85
17.10-15 Ejecting EMASS Archive Media Using DAS - Quick-Step Procedure..... 17-86
17.10-16 Removing Problem Media - Quick-Step Procedure..... 17-87
17.10-17 Inserting All Media Types into I/O Unit - Quick-Step Procedure..... 17-89
17.10-18 Removing All Ejected Media Types from I/O Unit - Quick-Step Procedure 17-90
17.10.-19 Removing Problem Media from I/O Unit- Quick-Step Procedure..... 17-92
17.11-1 Archiving Application Log Files - Activity Checklist..... 17-93
17.11-2 ALOG file archiving procedure – Quick-Steps..... 17-95
17.11-3 Verify/Create Subsystem specific subdirectory – Quick-Steps 17-96
17.11-4 Copy and write ALOG file to Archive– Quick-Steps 17-98
17.11-5 ALOG file retrieval from Archive – Quick-Step Procedures..... 17-100
18.1-1 Distribution Media Types..... 18-1
18.2-1 ECS Data Distribution Operator Tool - Activity Checklist 18-3
18.2-2 Starting Data Distribution Operator GUI - Quick-Steps..... 18-4
18.2-3 Monitoring/Controlling Data Distribution Requests - Quick-Steps 18-5
18.2-4 Configuring Data Distribution Polling and Error Retry Rate - Quick-Steps 18-6
18.2-5 Changing the Priority of Data Distribution Requests - Quick-Steps 18-8
18.2-6 Suspend a Request - Quick-Steps..... 18-9
18.2-7 Resuming Processing on a Suspended Data Distribution Request - Quick-Steps 18-10
18.2-8 Canceling a Data Distribution Request - Quick-Steps 18-11
18.3-1 Media Operations - Activity Checklist..... 18-12
18.3-2 Loading and Unloading Tapes - Quick-Step Procedures 18-14
18.4-1 Product Shipment - Activity Checklist..... 18-15
18.4-2 Performing Quality Control of Hard Media - Quick-Steps..... 18-16
18.4-3 Mark Hard Media for Shipment - Quick-Steps 18-17
18.5-1 Recovery from a Data Distribution Failure - Activity Checklist..... 18-18

19.1-1 ECS User Account Management - Activity Checklist 19-2
19.1-2 Validate a User - Quick-Step Procedures 19-3
19.1-3 Create an ECS User Account - Activity Checklist 19-4
19.1-4 Creating an ECS User Account Quick-Step Procedures 19-15
19.1-5 Completion of URL Registration - Quick-Step Procedures 19-20
19.1-6 Edit/Modify and Existing Account - Activity Checklist 19-21
19.1-7 Edit/Modify an Account - Quick-Step Procedures 19-35
19.1-8 Deleting an ECS Account - Quick-Step Procedures 19-39
19.1-9 Canceling an ECS Account - Quick-Step Procedures 19-41
19.1-10 Changing an ECS User's Password - Quick-Step Procedures 19-44
19.2-1 Processing an Order - Activity Checklist 19-45
19.2-2 Creating a User Contact Log - Quick-Step Procedures 19-50
19.2-3 Retrieve User Account - Quick-Step Procedures 19-52
19.2-4 Locate Data Via EOS Data Gateway - Activity Checklist 19-53
19.2-5 Update User Contact Log Record - Quick-Step Procedures 19-56
21.3-1 DAAC Hardware Problem Reporting Procedures 21-5
21.3-2 Hardware Corrective Maintenance Actions 21-5
21.3.4-1 Obtaining On-Site Hardware Maintenance 21-7
21.3.5-1 Procedure for Obtaining Return to Depot Service 21-10
21.3.5-2 Procedure Equipment Advance Replacement 21-11
21.4.1-1 Centrally Stocked Spares 21-12
21.4.2-1 Return of Failed LRU 21-13
21.5-1 Procedure for Time and Material Support 21-14
22.1-1 COTS Maintenance - Activity Outline 22-2
22.2-1 Custom Software Maintenance - Activity Checklist 22-7
22.2-2 ECS Software Oriented Tables State Table 22-14
22.2-3 Valid State Transitions 22-15
22.2-4 Valid State Assignment Given Current Promotion Level 22-16
22.2-5 Science Software Oriented State Table 22-17

22.2-6 Science Software Oriented Valid State Transitions 22-17
22.2-7 Science Software Oriented Promotion Table..... 22-18
22.2-8 Detailed Steps of SW Installation..... 22-19
23.1- 2 Procedure for Completion of the Inventory Worksheet.....23-2
23.1-4 Receiving Process Checklist.....23-3
23.2-1 ILS Property Administrator.....23-3
24.1-1 Installation Planning Activity Outline.....24-2
25.1-1 COTS Training - Activity Checklist.....25-1
26-1 Common ECS Operator Functions Performed with the ESOD 26-3
26-2 ESOD Administration - Activity Checklist.....26-4
26-3 ESOD Moderation Quick-Step Procedures 26-6
26-4 Create a Moderation Group Quick-Step Procedures.....26-7
26-5 Update a Moderation Group Quick-Step Procedures.....26-9
26-6 Delete a Moderation Group Quick-Step Procedures..... 26-10
27.4.1-1 Procedures to Log into ILM 27-4
27.5-1 XRP-II menu Function Key bottom commands 27-5
27.5-2 ILM Main Menu Functions 27-6
27.7-1 EIN Menu options..... 27-13
27.7.1-1 EIN Structure Manager Field Descriptions..... 27-14
27.7.2-1 Items Page Field Descriptions..... 27-15
27.7.3-1 Procedure to Check relationship between Parent and its Children 27-15
27.7.4-1 EIN Manager Field Description..... 27-17
27.7.4-3 Procedure to Inquiry for EIN..... 27-18
27.8-1 ILM Reports Menu options 27-19
27.8.1-2 Procedures to generate EIN structure reports 27-21
27.9-1 Install/Receipt Report Field Descriptions..... 27-22
27.9-2 Procedures to generate Install/Receipt reports 27-23
27.10-1 Installation Summary Reports Field Descriptions..... 27-24
27.10.-2 Procedures to generate installation summary reports 27-25

27.11-1 Maintenance Menu options..... 27-26
27.11.1-1 Work Order Entry Field Descriptions 27-28
27.11.2-1 Item Page for Work Order Entry Field Descriptions 27-30
27.11.2-2 Procedures to enter new work order 27-31
27.12-1 Work Order Modification Field Descriptions..... 27-33
27.13-1 Delay Time Page for Work Order Modification Field Descriptions..... 27-35
27.13.1-1 Chargeable Hours Page for Work Order Modification Field Descriptions 27-36
27.13.2-1 Items Page for Work Order Modification Field Descriptions 27-37
27.13.2-2 Procedures to Modify Work Order..... 27-38
27.14-1 Preventative Maintenance Items Field Descriptions..... 27-40
27.14.1-1 Generate Maintenance Orders Field Descriptions 27-41
27.14.1-2 Procedures to generate PM orders 27-42
27.15-1 Maintenance Work Order Reports Field Descriptions..... 27-43
27.15-2 Procedures to generate Maintenance Work Order Reports 27-44
27.16.1 Work Order Status Reports Field Descriptions 27-45
27.16.2 Procedures to generate Maintenance Work Order Reports..... 27-46
27.17.1-1 Maintenance Codes Field Descriptions..... 27-49
27.17.1-2 Maintenance Contracts Field Descriptions..... 27-49
27.17.2-1 Authorized Employees Field Descriptions..... 27-50
27.18-1 ILM Master Menu options 27-50
27.18.1-1 Employee Manager Field Descriptions..... 27-51
27.18.1.2-1 Inventory Location Manager Field Descriptions..... 27-52
27.18.2-1 Hardware/Software Codes Field Descriptions..... 27-53
27.18.3-1 OEM Part Numbers Field Descriptions 27-54

Appendix A. Additional Material

Abbreviations & Acronyms

1. Introduction

This document, Mission Operation Procedures for the Release 5A system for ECS, provides Maintenance and Operations (M&O) procedures to configure, maintain and operate the Release 5A ECS system.

1.1 Identification

This document meets the milestone specified as Contract Data Requirements List (CDRL) Item 117, DID 611/OP3 under contract NAS5-6000. It reflects the ECS as delivered at Release 5A.

1.2 Scope

The scope of this document is directed to DAAC M&O activities to support the Release 5A ECS system. Both procedures and instructions are identified. Operations procedures are defined as the step-by-step commands or on-line procedures needed to perform a function. The Operations Instructions are the off-line procedures or directives for performing administrative, operations, management or operations support activities, e.g., Configuration Management, Problem Management, Performance Reporting, etc.

1.2.1 On-Site Procedures Tailoring Guide

Each DAAC may modify these procedures and instructions to accommodate site-specific M&O requirements. Such documentation should be edition dated in Word format with a master copy forwarded to SEO System Engineering. Specifics on authoring, formatting, importing, exporting and maintenance see Chapter 20, Library Administration.

1.3 Purpose

The purpose of this document is to identify the procedures and instructions to operate and maintain Release B systems. In addition, DAAC staff responsibilities are identified. The DAAC M&O staff is comprised of operators, engineers, as well as operations support, administration and management staff personnel.

This document will also be used as a training aid for M&O staff who are located at the sites. The operations procedures and operations instructions were derived from, and are intended to be consistent with, the system functions and capabilities specified in the ECS design specifications and the operations activities described in the ECS Operations Concept Document.

1.4 Status and Schedule

This document is to be delivered at CSR minus two weeks. Updates will be made to reflect subsequent system releases. Changes will be submitted through established configuration management procedures, such as document change notices or published revisions.

1.5 Organization

The contents subsequent to this first section are presented as follows:

- Section 2 **Related Documentation.** Lists documents that drive, support or expand on the material in this manual.
- Section 3 **System Administration.** Identifies the operations procedures and/or operations instructions for system administration activities, such as backup and restore, log maintenance, user account administration, and workstation installation.
- Section 4 **Database Administration.** Identifies the operations procedures and/or operations instructions for database administration activities, such as product installation, disk storage management, login and privileges administration, database validation, backup and recovery, database configuration, tuning and performance monitoring.
- Section 5 **Security Services.** Identifies the operations procedures and/or operations instructions for security services activities, such as user authentication and authorization, data access control, network services monitoring, password protection, file modification monitoring.
- Section 6 **Network Administration.** Identifies the operations procedures and/or operations instructions for network administration activities, such as network and system configuration monitoring, network services monitoring.
- Section 7 **System Monitoring.** Identifies the operations procedures and/or operations instructions for network system monitoring, such as problem monitoring and resolution.
- Section 8 **Problem Management.** Identifies the operations procedures and/or operations instructions for submitting trouble tickets and for processing and resolving trouble ticket submissions.
- Section 9 **Configuration Management.** Identifies the operations procedures and/or operations instructions for configuration management activities, such as CCB support, configuration item identification, configuration change requests (CCRs) submission and processing, configuration status accounting, configuration audits, data management, operational database maintenance, software transfer and installation.

- Section 10 **Metadata Administration.** Identifies the operations procedures and/or operations instructions for metadata administration activities, such as establishing collections, populating the database, specifying Earth Science Data Type (ESDT) services.

- Section 11 **Science Software Integration & Test (SSI&T) Operational Procedures.** Identifies the operations procedures and/or operations instructions to support SSI&T activities.

- Section 12 **Resource Planning.** Identifies the operations procedures and/or operations instructions for resource planning activities for non-routine ground events.

- Section 13 **Production Planning.** Identifies the operations procedures and/or operations instructions for production planning activities for job and resource prioritization and scheduling.

- Section 14 **Production Processing.** Identifies the operations procedures and/or operations instructions to support data processing activities.

- Section 15 **Quality Assurance.** Identifies the operations procedures and/or operations instructions to perform DAAC manual non-science quality assurance activities, such as visualization of science data products and updating quality assurance metadata.

- Section 16 **Ingest.** Identifies the operations procedures and/or operations instructions to support data acquisition.

- Section 17 **Archive.** Identifies the operations procedures and/or operations instructions for archiving activities, such as archive repository maintenance, fault monitoring and notification, temporary data storage.

- Section 18 **Data Distribution.** Identifies the operations procedures and/or operations instructions to support data distribution activities, such as media operations and product shipment.

- Section 19 **User Services.** Identifies the operations procedures and/or operations instructions to support user services activities to address user requests.

- Section 20 **Library Administration.** Identifies the operations procedures and/or operations instructions to support librarian administration activities, such as database administration of the Document Data Server Subsystem, change package preparation and distribution, master document control and maintenance.

- Section 21 **COTS Hardware Maintenance.** Identifies the operations procedures and/or operations instructions for preventive and corrective maintenance activities of COTS hardware for the ECS project.

- Section 22 **Software Maintenance.** Identifies the operations procedures and/or operations instructions to support maintenance activities for both COTS software, custom software, and science software.

- Section 23 **Property Management.** Identifies the operations procedures and/or operations instructions for the receipt, control, and accountability of ECS property at ECS sites during Release 5A.

- Section 24 **Installation Planning.** Identifies the operations procedures and/or operations instructions to support installation planning activities for conducting site surveys, ensuring that site preparations/coordination are completed on schedule, facilitating receipt and installation of the hardware.

- Section 25 **COTS Training.** Identifies the operations procedures and/or operations instructions to support COTS training activities, such as training request processing, training coordination, training scheduling, training record maintenance.

- Section 26 **On-line Advertising Service Administration.** Identifies the operations procedures and/or operations instructions for the Earth Science On-line Directory (ESOD) on-line advertising service activities, such as group administration and request processing.

- Section 27 **ILM (Integrated Logistics Maintenance Management).** To be added later

- Appendix A **Miscellany.** Contains additional information related to previous sections.

- **Abbreviations and Acronyms.** Identifies abbreviations and acronyms used throughout this document.

2. Related Documentation

2.1 Parent Documents

The parent documents are the documents from which this Mission Operation Procedures' scope and content are derived.

423-41-01	Goddard Space Flight Center, EOSDIS Core System (ECS) Statement of Work
423-41-03	Goddard Space Flight Center, EOSDIS Core System (ECS) Contract Data Requirements Document

2.2 Applicable Documents

The following documents are referenced within this Mission Operation Procedures, or are directly applicable, or contain policies or other directive matters that are binding upon the content of this volume.

209-CD-001	Interface Control Document Between EOSDIS Core System (ECS) and the NASA Science Internet for the ECS Project
209-CD-002	Interface Control Document Between EOSDIS Core System (ECS) and ASTER Ground Data System for the ECS Project
209-CD-006	Interface Control Document Between the EOSDIS Core System (ECS) and the National Oceanic and Atmospheric Administration (NOAA) Affiliated Data Center (ADC) for the ECS Project
209-CD-007	Interface Control Document Between the EOSDIS Core System (ECS) and TRMM Science Data and Information System (TSDIS) for the ECS Project
209-CD-008	Interface Control Document Between the EOSDIS Core System (ECS) and the Goddard Space Flight Center (GSFC) Distributed Active Archive Center (DAAC) for the ECS Project
209-CD-009	Interface Control Document Between the EOSDIS Core System (ECS) and the Marshall Space Flight Center (MSFC) Distributed Active Archive Center (DAAC) for the ECS Project
209-CD-010	Interface Control Document Between the EOSDIS Core System (ECS) and the Langley Research Center (LaRC) Distributed Active Archive Center (DAAC) for the ECS Project

209-CD-013	Interface Control Document Between the EOSDIS Core System (ECS) and the Landsat 7 System for the ECS Project
305-CD-002	Science Data Processing Segment (SDPS) Design Specification for the ECS Project
305-CD-003	Communications and System Management (CSMS) Design Specification for the ECS Project
305-CD-004	Overview of Release A SDPS and CSMS System Design Specification for the ECS Project
305-CD-005	Release A SDPS Client Subsystem Design Specification for the ECS Project
305-CD-006	Release A SDPS Interoperability Subsystem Design Specification for the ECS Project
305-CD-007	Release A SDPS Data Management Subsystem Design Specification for the ECS Project
305-CD-008	Release A SDPS Data Server Subsystem Design Specification for the ECS Project
305-CD-009	Release A SDPS Ingest Subsystem Deign Specification for the ECS Project
305-CD-010	Release A SDPS Planning Subsystem Design Specification for the ECS Project
305-CD-011	Release A SDPS Data Processing Subsystem Design Specification for the ECS Project
305-CD-012	Release A CSMS Communications Subsystem Design Specification for the ECS Project
305-CD-013	Release A CSMS Systems Management Subsystem Design Specification for the ECS Project
305-CD-014	Release A GSFC DAAC Design Specification for the ECS Project
305-CD-015	Release A LaRC DAAC Design Specification for the ECS Project
305-CD-016	Release A MSFC DAAC Design Specification for the ECS Project
305-CD-017	Release A EDC DAAC Design Specification for the ECS Project
305-CD-019	Release A System Monitoring and Coordination Center Design Specification for the ECS Project
305-CD-040	Flight Operations Segment (FOS) Design Specification for the ECS Project (Segment Level Design) Overview

305-CD-041	Flight Operations Segment (FOS) Planning and Scheduling Design Specification for the ECS Project
305-CD-042	Flight Operations Segment (FOS) Command Management Design Specification for the ECS Project
305-CD-043	Flight Operations Segment (FOS) Resource Management Design Specification for the ECS Project
305-CD-044	Flight Operations Segment (FOS) Telemetry Design Specification for the ECS Project
305-CD-045	Flight Operations Segment (FOS) Command Design Specification for the ECS Project
305-CD-046	Flight Operations Segment (FOS) Real-Time Contact Management Design Specification for the ECS Project
305-CD-047	Flight Operations Segment (FOS) Analysis Design Specification for the ECS Project
305-CD-048	Flight Operations Segment (FOS) User Interface Design Specification for the ECS Project
305-CD-049	Flight Operations Segment (FOS) Data Management Design Specification for the ECS Project
305-CD-050	Flight Operations Segment (FOS) Planning and Scheduling Program Design Language (PDL) for the ECS Project
305-CD-051	Flight Operations Segment (FOS) Command Management Program Design Language (PDL) for the ECS Project
305-CD-052	Flight Operations Segment (FOS) Resource Management Program Design Language (PDL) for the ECS Project
305-CD-053	Flight Operations Segment (FOS) Telemetry Program Design Language (PDL) for the ECS Project
305-CD-054	Flight Operations Segment (FOS) Real-Time Contact Management Program Design Language (PDL) for the ECS Project
305-CD-055	Flight Operations Segment (FOS) Analysis Program Design Language (PDL) for the ECS Project
305-CD-056	Flight Operations Segment (FOS) User Interface Program Design Language (PDL) for the ECS Project
305-CD-057	Flight Operations Segment (FOS) Data Management Program Design Language (PDL) for the ECS Project

305-CD-058	Flight Operations Segment (FOS) Command Program Design Language (PDL) for the ECS Project
313-CD-001	EOSDIS Core System (ECS) Internal Interface Control Document for the Flight Operations Segment for the ECS Project
313-CD-500	Release 5A ECS Internal Interface Control Document for the ECS Project
601-CD-001	Maintenance and Operations Management Plan for the ECS Project
604-CD-001	Operations Concept for the ECS Project: Part 1-- ECS Overview
604-CD-003	Operations Concept for the ECS Project: Part 2A -- ECS Release A
605-CD-001	Operations Scenarios for the ECS Project: ECS Release A
609-CD-500	Release 5A Operations Tools Manual for the ECS Project
505-41-33	Goddard Space Flight Center, Interface Control Document Between EOSDIS Core System (ECS) and Science Computing Facilities (SCF), 1/96

2.3 Information Documents

2.3.1 Information Documents Referenced

The following documents are referenced herein and, amplify or clarify the information presented in this document. These documents are not binding on the content of the ECS Mission Operation Procedures for the ECS Project.

501-CD-001	Performance Assurance Implementation Plan (PAIP) for the ECS Project
532-CD-001	Release A Environmental Control Plan for the ECS Project
602-CD-001	Property Management Plan for the ECS Project
613-CD-002	Release A COTS Maintenance Plan for the ECS Project
613-CD-003	Release B COTS Maintenance Plan for the ECS Project
616-CD-001	Release A Integrated Logistics Support Plan for the ECS Project
622-CD-001	Training Plan for the ECS Project
CM-1-016-1	Software Development Using ClearCase, ECS Work Instruction
CM-1-023-1	Software Build Process Using ClearCase, ECS Work Instruction
423-41-02	Goddard Space Flight Center, Functional and Performance Requirements Specifications for the EOSDIS Core System (ECS)

2.3.2 Information Documents Not Referenced

The following documents, although not referenced herein and/or not directly applicable, do amplify or clarify the information presented in this document. These documents are not binding on the content of the Mission Operation Procedures for the ECS Project.

607-CD-001	ECS Maintenance and Operations Positions Descriptions for the ECS Project

This page intentionally left blank.

3. System Administration

This section covers the procedures necessary for the System Administrator (SA) and/or Operator (OPR) to manage and operate the system.

Detailed procedures for tasks performed by the System Administrator and/or Operator are provided in the sections that follow. The procedures assume that the administrator and/or operator is authorized and has proper access privileges to perform the tasks (i.e., root) and that the SA and/or OPR has been properly trained in all aspects of the system.

Each procedure outlined will have an **Activity Checklist** table that will provide an overview of the task to be completed. The outline of the **Activity Checklist** is as follows:

Column one - ***Order*** shows the order in which tasks should be accomplished.

Column two - ***Role*** lists the Role/Manager/Operator responsible for performing the task.

Column three - ***Task*** provides a brief explanation of the task.

Column four - ***Section*** provides the Procedure (P) section number or Instruction (I) section number where details for performing the task can be found.

Column five - ***Complete?*** is used as a checklist to keep track of which task steps have been completed.

The following is the **Activity Checklist** table that provides an overview of the overall system administration processes and who performs them

Table 3.1. System Administration - Activity Checklist

Order	Role	Task	Section	Complete?
1	OPR	Startup/Shutdown	(I) 3.1	
2	OPR	System Backup and Restore	(I) 3.2	
3	SA	System Log Maintenance	(I) 3.3	
4	SA	User Administration	(I) 3.4	
5	SA	Installing a New Workstation	(I) 3.5	
6	SA	DCE Configuration	(I) 3.6	
7	SA/ OPR	Installing CUSTOM Software/System Monitoring	(I)3.7	

For each procedure outlined in this section, there is a corresponding **QUICK STEP** procedure immediately following in this chapter. The **QUICK STEP** procedures are designed for persons who have ***prior training or are experienced system administrators with prior system administration experience.*** The ***QUICK STEP*** procedures should be used by ***experienced persons ONLY***.

3.1 System Startup and Shutdown

The Startup and Shutdown processes begin when it has been determined by the DAAC Operations Supervisor or his designee that it is necessary to stop or start the system. The least impacting method is determined and users are appropriately notified.

When determining the least impacting way to perform the startup or shutdown, the OPR, along with the Operations Supervisor takes into consideration whether only specific server software packages would need to be started/stopped or an entire system startup/shutdown is required.

Once these steps have been taken, the shutdown or startup is performed.

The **Activity Checklist** table that follows provides an overview of the startup and shutdown processes.

Table 3.1-1. Startup/Shutdown - Activity Checklist

Order	Role	Task	Section	Complete?
1	OPS Sup	Determine that Startup/Shutdown is necessary.	(I) 3.1	
2	OPR	Determine the Least Impacting Way to Perform the Startup/Shutdown.	(I) 3.1	
3	OPR	Notify Those Effected by the Startup/Shutdown.	(I) 3.1	
4	OPR	Perform the Startup/Shutdown	(P) 3.1	

3.1.1 Startup

Startup means that power to the system is restored and the system is being taken to a fully useable and operational state.

3.1.1.1 Cold - By Subsystem

A cold startup means that power to the system has been previously powered off and the system(s) is being restarted from this cold state. The System Startup process begins after a previously completed shutdown, either scheduled or emergency. The System Startup is done in sequential order by subsystem. This startup sequence is predetermined by the SA.

This procedure assumes that the OPR has been properly trained to startup all aspects of the system and that the system is currently powered off (due to a normal or emergency shutdown).

The procedure assumes that the Startup has been scheduled well in advance, all planning involved has been concluded well in advance and all other Distributed Active Archive Centers (DAACs) have been notified of the system returning to an on-line state.

This section explains how to perform a cold system startup by subsystem. The sequence of the execution of the steps below are VERY IMPORTANT. To begin a cold system startup, execute the procedure steps that follow:

1 The **sequence** of booting the machines is **IMPORTANT**:

- ***Remember to power on peripherals before powering on each CPU***
- ***Monitor each system Boot Up activity on that system's monitor***
- *****The DNS and NIS servers must be booted FIRST***
- ***Once each system has booted without error, proceed to the next machine***
- Boot the machines per Table 3.1-2

2 Continue booting the remaining machines

Table 3.1-2 presents the cold system startup machine boot sequence, however, the machine names are to be added once identified at each DAAC, per their specific baseline by the SA.

Table 3.1-2. Cold System Startup - Machine Boot Sequence

Step	What to Enter or Select	Action to Take (Server)	Machine Name
1	(No entry)	NIS Master	
2	(No entry)	DCE Servers	
3	(No entry)	DNS Master	
4	(No entry)	Mail Hub Server(s)	
5	(No entry)	Automount	
6	(No entry)	Clearcase Server(s)	
7	(No entry)	CSS Server(s)	
8	(No entry)	MSS: *Tivoli Server* *Sybase SQL Servers*	
9	(No entry)	DSS	
10	(No entry)	Ingest	
11	(No entry)	PDPS	
12	(No entry)	CIDM	
13	(No entry)		

3.1.1.2 Warm - By Subsystem Startup

A warm startup means the system has been previously powered on, but the system(s) is not fully operational, either the system has had some service performed (i.e. single user mode) or is being rebooted to correct some minor malfunction. The System Startup is done in sequential order by subsystem. This startup sequence is predetermined by the software dependencies.

The order of the re-boot is contingent on software dependencies per site.

If the NIS/DCE servers service has been interrupted, the users will automatically be transferred to a backup server. Once the faulty server(s) has been repaired, re-establish connection with the primary NIS/DCE server by rebooting the Backup Server; the users would then be transferred back to the primary NIS/DCE server.

Table 3.1-3 presents the **QUICK STEP** procedure required to perform a warm system startup.

Table 3.1-3. Warm System Startup - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	(No entry)	determine software dependencies
2	(No entry)	reboot independent server(s)
3	(No entry)	reboot dependent server(s)

Note - in addition to warm system startup/reboot sequences, ECS servers which use the Sybase SQL server may need to be bounced whenever the SQL server is bounced. At present, this is certainly the case for all STMGT servers. That is, if the Sybase SQL server is stopped and restarted for any reason, all STMGT servers need to be stopped and restarted, once the Sybase SQL server has come back on-line.

3.1.1.3 Additional tasking - Updating leapsec.dat and utcpole.dat files

In addition to starting system servers there are essential tasks that System Administrators must perform on a regular basis.

In order to ensure proper operation of Program Generated Executives (PGEs), two files must be updated weekly with data transferred from the U.S. Naval Observatory. These files are ${PGSHOME}/database/common/TD/leapsec.dat and ${PGSHOME}/database/common/CSC/utcpole.dat. The update of these files is accomplished by executing leapsec_update.sh and utcpole_update.sh in the /tools/admin/exec directory with root privileges. It has not been determined yet if these tasks will be accomplished manually or via cron job scripting .

3.1.2 Shutdown

Shutdown means that the system is being removed from a fully useable and operational state and possibly, power to the system will be terminated. The types of shutdown would vary depending upon circumstances (i.e. shutdown to single user mode; shutdown to power off; etc.)

3.1.2.1 Normal - By Subsystem

The Normal System Shutdown process is performed at the discretion of the SA usually for a scheduled repair. The system shutdown is **normally performed in reverse order of the system startup.**

The procedures assume that the Shutdown has been scheduled well in advance, all planning involved has been concluded well in advance and the OPR/SA has been properly trained to shutdown all aspects of the system.

This section explains how to perform a normal system shutdown by subsystem**.**

3.1.2.1.1 Shutdown a Machine

The OPR must be logged in as root to perform a shutdown. To begin a normal system shutdown, execute the procedure steps that follow:

1 Login to the server as root.

2 Enter root password.

3 Type **wall** and press **Return**. Use **wall -a** on Sun machines to cross NFS mounts.

4 Type **This machine is being shutdown for *reason*. The anticipated length of down time is *xxx*. Please save your work and log off now. The machine will be coming down in *xxx* minutes. We are sorry for the inconvenience.** then press **Return.** Press **Control** and **D** keys **simultaneously**. Include your name and closest telephone number.

5 Wait at least five minutes.

6 Type **shutdown -g600 -i0 -y** UNIX prompt and press **Return**. (600 = Number of Seconds)

7 When the system is at the prompt it is safe to ***Power off*** all peripherals first, and then the CPU.

The servers should be shutdown in the reverse order of the startup:

1 Determine which machines are dependent on a server first:

- ***Once each system has stopped without error, power off peripherals***
- ***Proceed to the following machine***

- ***Follow steps 1-7 for Task: 3.1.2.1.1 Shutdown Machine*** *for* ***each*** *machine*

2 **Do NOT** turn power off on the **HP Systems**, instead bring them down to the halted mode, unless power is going to be lost for the entire facility.

3 The **NIS/DCE server** must be the **last system** to be shutdown.

Table 3.1-4 presents the **QUICK STEP** procedure required to perform a normal system shutdown.

Table 3.1-4. Normal System Shutdown - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	(No entry)	**determine subsystems and server dependencies**
2	(No entry)	**Login to the server as root**
3	**wall**	**Press Return**
4	**This machine is being shutdown for *reason*. The anticipated length of down time is *xxx*. Please save your work and log off now. The machine will be coming down in *xxx* minutes. We are sorry for the inconvenience.**	**Press Control and D keys simultaneously**
5	(No entry)	**Wait at least five minutes**
6	**shutdown -g600 -i0 -y** **- OR -** **shutdown now -i0 -y**	**Press Return**
7	(No entry)	**Power off all peripherals and the CPU.**
8	(No entry)	**Repeat steps 2 through 7 above for all servers Table 3.1.2**

3.1.2.2 Emergency - By Subsystem

The Emergency System Shutdown process begins after it is determined that the system may fail during emergency situations (i.e., storms, power outages) by the System Administrator (SA). The Emergency System Shutdown is done in sequential order by subsystem. This shutdown sequence is predetermined by the SA.

The NIS/DCE servers must be the last system to shutdown.

Detailed procedures for tasks performed by the OPR/SA are provided in the sections that follow.

This section explains how to perform an emergency system shutdown by subsystem. The OPR must be logged in as root to perform a shutdown. To begin an emergency system shutdown, execute the procedure steps that follow:

1 Login to the server as root.

2 Enter root password.

3 Type **sync** at the UNIX prompt and hit **Return**.

Sync executes the sync system primitive. If the system is to be stopped, sync must be called to insure file system integrity. It will flush all previously unwritten system buffers out to disk, thus assuring that all file modifications up to that point will be saved.

4 Type **sync** again at the UNIX prompt and hit **Return**.

5 Type **halt** at the UNIX prompt and hit **Return**.

6 Once the halt has completed, turn the power switch on all the peripherals and the CPU off.

The servers should be shutdown in the following order:

1 Shutdown all client workstations.

2 ***Follow steps 1-7 for Task: 3.1.2.1.1 Shutdown Machine*** *for* ***each*** *machine*

3 Do **NOT** turn power off on the **HP Systems**, instead bring them down to the halted mode, unless power is going to be lost for the entire facility.

4 The **NIS/DCE servers** must be the **last systems** to shutdown.

In case of ***EXTREME emergency*** where time does not allow you to execute the above procedures, execute the following procedure steps for ***Sun machines ONLY.***

1 Login to the server as root.

2 Enter root password.

3 Hit the L1 or Stop key and the a key simultaneously.

4 Once returned to an **ok** or > prompt, turn the power switches on the CPU and all peripherals to off.

NOTE: The use of L1a does not ensure file system integrity. There is a very high risk of losing data when using this process.

Table 3.1-5 presents the **QUICK STEP** procedures required to perform a Emergency System Shutdown.

Table 3.1-5. Emergency System Shutdown - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	(No entry)	determine subsystems and server dependencies
2	(No entry)	Login to server as root
3	(No entry)	Type sync at prompt and press enter
4	(No entry)	Type sync at prompt and press enter
5	(No entry)	Type halt at prompt and press enter
6	(No entry)	Turn power switches on CPU and all peripherals to off.
7	(No entry)	Repeat steps 2 through 5 above for all servers

3.1.2.3 Server - By Server Software

The System Shutdown by Server Software process is performed by the OPR. The system shutdown is normally performed in reverse order of the system startup.

The procedures assume that the Shutdown has been scheduled well in advance, all planning involved has been concluded well in advance and the SA has been properly trained to shutdown all aspects of the system.

Table 3.1-6 presents the **QUICK STEP** procedure required to perform a normal system shutdown.

Table 3.1-6. Server System Shutdown - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	(No entry)	determine software dependencies
2	(No entry)	shutdown dependent server(s)
3	(No entry)	shutdown independent server(s)

3.2 System Backup and Restore

System Backup and Restore is the process of copying, either the entire or partial system, the information from the machine for safe keeping for a specific time period. Restore is the process of returning the data to the machine to allow operation to continue from a specific point in time. The OPR must be in the admin list to use Networker. This is not root privilege.

3.2.1 Incremental Backup

Non-scheduled incremental backups can be requested at any time by submitting a request through **REMEDY** for ***Incremental Backup*** to the **OPS supervisor**. The supervisor schedules the request with the OPR who performs the incremental backup. Afterwards, the OPR notifies the requester and supervisor that the incremental backup is complete.

The **Activity Checklist** table that follows provides an overview of the incremental backup processes.

Table 3.2-1. Incremental Backup - Activity Checklist

Order	Role	Task	Section	Complete ?
1	Requester	Submit Request for **Incremental** Backup **using REMEDY** to OPS Supervisor.	(I) 3.2.1	
2	OPS Super	Schedule Incremental Backup with OPR	(I) 3.2.1	
3	OPR	Perform Incremental Backup.	(P) 3.2.1	
4	OPR	Notify Requester and OPS Super when Incremental Backup is Complete.	(I) 3.2.1	

Detailed procedures for tasks performed by the OPR are provided in the sections that follow.

The procedures assume that the requester's request for an incremental backup has already been approved by DAAC Management. Incremental backups can be requested at any time by submitting a request through **REMEDY** for ***Full Backup*** to the **OPS supervisor**. In order to perform the procedure, the OPR must have obtained the following information from the requester:

a. **Name of machine to be backed up**

b. **Files/directories to be backed up (optional)**

To perform an incremental backup for the requester, execute the procedure steps that follow:

Note 1: If you run out of tapes at any time during this procedure, execute procedure 3.2.5.2 Labeling Tapes and then return to this procedure.

1 Log into the **machine to be backed up** by typing: **telnet *BackedUpSystemName* -OR- rsh *BackedUpSystemName***, then press **Return**.

2 If a **Login**: prompt appears, log in as yourself by typing: ***YourUserID***, then press **Return**.

- A password prompt is displayed.

3 Enter ***YourPassword***, then press **Return.**

- Remember that ***YourPassword*** is case sensitive.
- You are authenticated as yourself and returned to the UNIX prompt.

4 Log in as root by typing: **su**, then press **Return**.

- A password prompt is displayed.

5 Enter the ***RootPassword***, then press **Return**.

- Remember that the ***RootPassword*** is case sensitive.
- You are authenticated as root and returned to the UNIX prompt.

6 Set display to current terminal by typing: **setenv DISPLAY *IPNumber*:0.0** ***-OR -*** **setenv DISPLAY *BackedUpSystemName*:0.0** then press **Return**.

7 Execute the NetWorker Administrative program by entering: **nwadmin &**, then press **Return**.

- A window opens for the NetWorker Administrative program.
- You are now able to perform an incremental backup.

8 Click Clients.

- Click Client Setup
- Click Host Being Backed Up
- Highlight the group to be Backed Up

9 Go to the **Customize** menu, select **Schedules**.

- The **Schedules** window opens.

10 Look at the button for today. If there is an **i** next to the date on this button, go to step 12.

- The **i** stands for incremental.
- The **f** stands for full.

Whichever is on the button for today is what kind of backup that will be done, unless it is overridden.

11 Click and hold the button for today, select **Overrides** from the resulting menu, select **Incremental** from the next resulting menu.

12 Click the **Apply** button.

13 Close the **Schedules** window by clicking in the upper left corner of the **Schedules** window and selecting **Close** from the resulting menu.

14 Click the **Group Control** button.

- The **Group Control** window opens.

15 Click the **Start** button.

- A **Notice** window opens.

16 Click the **OK** button.

- The **Notice** window closes.
- The regularly scheduled backup will still run (even though we are now doing a backup).

17 Close the **Group Control** window by clicking in the upper left corner of the **Group Control** window and selecting **Close** from the resulting menu.

- Status updates appear in the **nwadmin** window.
- When the backup is complete, a **Finished** message will appear.

18 If the button for today in step 9 had an **i** on it, go to step 22.

19 Go to the **Customize** menu, select **Schedules**.

- The **Schedules** window opens.

20 Click and hold the button for today, select **Overrides** from the resulting menu, select **Full** from the next resulting menu.

21 Click the **Apply** button.

22 Close the **Schedules** window by clicking in the upper left corner of the **Schedules** window and selecting **Close** from the resulting menu.

23 Select **Exit** from the **File** menu to quit the NetWorker Administrative program.

- The **nwadmin** window closes.

24 At the UNIX prompt for the **machine to be backed up**, type **exit**, then press **Return**.

- **Root** is logged out.

25 Type **exit** again, then press **Return**.

- You are logged out and disconnected from the **machine to be backed up**.

Table 3.2-2 presents the **QUICK STEP** procedure required to perform an incremental backup.

Table 3.2-2. Perform Incremental Backup - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	**telnet** *BackedUpSystemName* **-or-** **rsh** *BackedUpSystemName*	**press Return**
2	*YourUserID* **or-** (No entry)	**press Return -or-** (No action)
3	*YourPassword*	**press Return**
4	**su**	**press Return**
5	*RootPassword*	**press Return**
6	**setenv DISPLAY** *IPNumber*:**0.0** **-or-** **setenv DISPLAY** *BackedUpSystemName*:**0.0**	**press Return**
7	**nwadmin**	**press Return**
8	Click Client Click **Client Setup** Click Host Being Backed Up **-** Highlight the Group to be Backed Up	
9	**Customize → Schedules**	**if i on today's button then go to step 10.** **Otherwise, click and hold today's button.**
10	**Overrides → Incremental**	**click Apply button**
11	(No entry)	**close Schedules window**
12	(No entry)	**click Group Control button**
13	(No entry)	**click Start button**
14	(No entry)	**click OK button**
15	(No entry)	**close Group Control window**
16	(No entry)	**if there was an i on today's button in step 8, go to step 18.**
17	**Customize → Schedules**	**click and hold today's button**
18	**Overrides → Full**	**click Apply button**
19	(No entry)	**close Schedules window**
20	**File → Exit**	(No action)
21	**exit**	**press Return**
22	**exit**	**press Return**

3.2.2 Full Backup

Non-scheduled full backups can be requested at any time by submitting a request through **REMEDY** for ***Full Backup*** to the OPS supervisor. The supervisor schedules the request with the OPR who performs the full backup. Afterwards, the OPR notifies the requester and supervisor that the full backup is complete.

The **Activity Checklist** table that follows provides an overview of the full backup processes.

Table 3.2-3. Full Backup - Activity Checklist

Order	Role	Task	Section	Complete?
1	Requester	Submit Request for **Full** Backup **using REMEDY** to OPS Supervisor.	(I) 3.2.2	
2	OPS Super	Schedule Full Backup with **OPR**	(I) 3.2.2	
3	OPR	Perform Full Backup.	(P) 3.2.2	
4	OPR	Notify Requester and OPS Super when Full Backup is Complete.	(I) 3.2.2	

Detailed procedures for tasks performed by the OPR are provided in the sections that follow.

The procedures assume that the requester's application for a full backup has already been approved by DAAC Management. In order to perform the procedure, the OPR must have obtained the following information from the requester:

a. **Name of machine to be backed up**

b. **Files/directories to be backed up** (optional)

To perform a full backup for the requester, execute the procedure steps that follow:

Note 1: If you run out of tapes at any time during this procedure, execute procedure 3.2.5.2 Labeling Tapes and then return to this procedure.

1 Log into the **machine to be backed up** by typing: **telnet *BackedUpSystemName* -OR- rsh *BackedUpSystemName***, then press **Return**.

2 If a **Login**: prompt appears, log in as yourself by typing: ***YourUserID***, then press **Return**.

- A password prompt is displayed.

3 Enter ***YourPassword***, then press **Return.**

- Remember that ***YourPassword*** is case sensitive.
- You are authenticated as yourself and returned to the UNIX prompt.

4 Log in as root by typing: **su**, then press **Return**.

- A password prompt is displayed.

5 Enter the ***RootPassword***, then press **Return**.

- Remember that the ***RootPassword*** is case sensitive.
- You are authenticated as root and returned to the UNIX prompt.

6 Set display to current terminal by typing: **setenv DISPLAY *IPNumber*:0.0 *-OR -* setenv DISPLAY *BackedUpSystemName*:0.0** then press **Return**.

7 Execute the NetWorker Backup program by entering: **nwbackup &**, then press **Return**.

- A **NetWorker Backup** window opens.
- You are now able to perform a full backup.

8 Click Clients.

- Click Client Setup
- Click Host Being Backed Up
- Highlight the group to be Backed Up

9 If no list of **files/directories to be backed up** was provided, i.e. the whole machine is to be backed up, then type **/** in the **Selection** field and click the **Mark** button.

- / is designated for backup and has a check next to it.

10 If names of **files/directories to be backed up** were provided then select the **files/directories to be backed up** in the directory display and click the **Mark** button.

- Drag scroll bar with mouse to scroll the list up and down.
- Double click on directory name to list its contents.
- To move up a directory level, type the path in the **Selection** field.
- Clicking the **Mark** button designates the file for backup and puts a check next to it.

11 Click the **Start** button.

- A **Backup Options** window opens.

12 Click the **OK** button.

- The **Backup Options** window closes.
- The **Backup Status** window opens providing updates on the backup's progress.

13 After the **Backup Completion Time** message appears in the **Backup Status** window, click the **Cancel** button.

- The **Backup Status** window closes.
- The backup is complete.

14 Select **Exit** from the **File** menu to quit the NetWorker Backup program.

- The **NetWorker Backup** window closes.

15 At the UNIX prompt for the **machine to be backed up**, type **exit**, then press **Return**.

- **Root** is logged out.

16 Type **exit** again, then press **Return**.

- You are logged out and disconnected from the **machine to be backed up**.

To perform a full backup, execute the steps provided in the following table.

Table 3.2-4 presents the **QUICK STEP** procedure required to perform a full backup.

Table 3.2-4. Perform Full Backup - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	**telnet** *BackedUpSystemName* **-or-** **rsh** *BackedUpSystemName*	**press Return**
2	*YourUserID* **or-** (No entry)	**press Return -or-** (No action)
3	*YourPassword*	**press Return**
4	**su**	**press Return**
5	*RootPassword*	**press Return**
6	**setenv DISPLAY** *IPNumber*:**0.0** **-or-** **setenv DISPLAY** *BackUpSystemName*:**0.0**	**press Return**
7	**nwbackup &**	**press Return**
8	Click Client Click **Client Setup** Click Host Being Backed Up **-** Highlight the Group to be Backed Up	
9	**to back up whole machine, place / in selection field**	**click Mark button**
10	**to back up certain files/directories, indicate the files/directories**	**click Mark button**
11	(No entry)	**click Start button**
12	(No entry)	**click OK button**
13	(No entry)	**click Cancel button**
14	**File** → **Exit**	(No action)
15	**exit**	**press Return**
16	**exit**	**press Return**

3.2.3 File Restore

SINGLE OR MULTIPLE FILES RESTORE

From time to time, individual files or groups of files (but not all files) will have to be restored from an Incremental or Full backup tape(s) due to Operator error or system failure. This can be accomplished using the following file restoration procedure.

The File Restore process begins when the requester submits a request to the Operator. The Operator restores the file(s) and notifies the requester when complete.

The Activity Checklist table that follows provides an overview of the file restore process.

Table 3.2-5 File Restore - Activity Checklist

Order	Role	Task	Section	Complete?
1	Requester	Submit Request for File Restore to Operator	(I) 3.2.3	
2	Operator	Restore file(s).	(P) 3.2.3	
3	Operator	Inform Requester of completion.	(I) 3.2.3	
4	Operator	Complete System Restore/Partition Restore	(P) 3.2.4	

Detailed procedures for tasks performed by the Operator are provided in the sections that follow.

The procedures assume that the requester's application for a file restore has already been approved by the Ops Supervisor. In order to perform the procedure, the Operator must have obtained the following information from the requester:

a. **Name of machine to be restored**

b. **Name of file(s) to be restored**

c. **Date from which to restore**

d. **User ID of the owner of the file(s) to be restored**

e. **Choice of action to take when conflicts occur. Choices are:**

- **Rename current file**
- **Keep current file**
- **Write over current file with recovered file**

Table 3.2-6 represents the steps required to restore a file in a condensed manner. If you are already familiar with the procedure, you may prefer to use the quick-step table. If you are new to the system, or have not performed these tasks recently, you should use the detailed procedures presented below.

To restore a file for the requester, execute the procedure steps that follow:

1 Log into the **machine to be restored** by typing: **telnet machine to be restored** or **rsh, the name of the *machine to be Restored***, then press **Return**.

2 If a **Login:** prompt appears, log in by typing: ***Your User ID***, then press **Return**.

- A password prompt is displayed.

3 Enter ***Your Password***, then press **Return.**

- Remember that your password is case sensitive.
- You are authenticated as yourself and returned to the Unix prompt..

4 Set display to current terminal by typing: **setenv DISPLAY *IPNumber*:0.0** on the current machine or **setenv DISPLAY the name of the *Machine to be Restored*:0.0**, then press **Return**.

NOTE: Before executing the NetWorker Recover insure, that you are in the /data1/COTS/networker directory.

5 Execute the **NetWorker Recovery** program by entering: **nwrecover &**, then press **Return**.

- A window opens for the Networker Recovery program.
- You are now able to perform the file restoration.

6 Select **file(s) to be restored** and click the **Mark** button.

- Drag scroll bar with mouse to scroll the list up and down.
- Double click on directory name to list its contents.
- Clicking the Mark button designates the file for restore and places a check in the box adjacent to the file or directory name.

7 Go to the **Change** menu, select **Browse Time**.

- The Change Browse Time window opens.

8 Select the **date from which to restore**.

- **NetWorker** will automatically go to that day's or a previous day's backup which contains the file.

9 Click the **Start** button.

- The Conflict Resolution window opens.

10 Answer **Do you want to be consulted for conflicts** by clicking the **yes** button, then click the **OK** button.

- If prompted with a conflict, choices of action will be: **rename current file**, **keep current file**, or **write over current file with recovered file**. Select the requesters **choice of action to take when conflicts occur**.
- The Recover Status window opens providing information about the file restore.
- If all the required tapes are not in the drive, a notice will appear. Click the **OK** button in the notice window.
- If prompted for tapes, click cancel in the **Recover Status** window and execute procedure 3.2.5-2 Index Tapes.

11 When a recovery complete message appears, click the **Cancel** button.

12 Go to the **File** menu, select **Exit**.

- The **NetWorker Recovery** program quits.

13 Type **exit**, then press **Return**.

- The **owner of the file(s) to be restored** is logged out.

14 Type **exit** one last time, then press **Return**.

- You are logged out and disconnected from the **machine to be restored.**

To restore a file, execute the steps provided in the following table.

Table 3.2-6. Restore a File - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	**telnet** to the machine to be restored -**or- rsh**, the *Machine to be Restored*	**press Return**
2	*Your User ID* **-or-** (No entry)	**press Return -or-** (No action)
3	*Your Password*	**press Return**
4	**setenv DISPLAY** *IPNumber*:**0.0 on the current machine -or** **setenv DISPLAY** the *Machine to be Restored*:**0.0**	**press Return**
5	**nwrecover &**	**press Return**
6	**file(s) to be restored**	**click the Mark button**
7	**Change → Browse Time**	(No action)
8	**date from which to restore**	**click the Start button**
9	**yes**	**click OK button**
10	(No entry)	**choose what action to take when notified of conflicts**
11	(No entry)	**click Cancel button**
12	**File → Exit**	(No action)
13	**exit**	**press Return**
14	**exit**	**press Return**
15	**exit**	**press Return**

3.2.4 Complete System Restore

The Complete System Restore process begins when the requester has determined that a complete system restore is the only way to resolve the problem and has approval from the Operations Supervisor. Once notified of the request, the Operator performs restores of all partitions on the system. Afterwards, the Operator documents and logs all actions in the operator's log book and notifies the requester and Ops Supervisor that the system restore is complete.

The Activity Checklist table that follows provides an overview of the complete system restore process.

Table 3.2-7. Complete System Restore - Activity Checklist

Order	Role	Task	Section	Complete?
1	Requester	Trouble Shoot and Determine that a Complete System Restore is necessary.	(I) 3.2.4	
2	Operator	Restore all Partitions on the System	(P) 3.2.4	
3	Operator	Document and Log in operator's log book, and Inform Requester and Ops Supervisor of completion.	(I) 3.2.4	

Detailed procedures for tasks performed by the Operator are provided in the sections that follow. The procedures assume that the requester's application for a complete system restore has already been approved by Ops Supervisor. In order to perform the procedures, the Operator must have obtained the following information about the requester:

a. **Name of system to be restored**

b. **Date from which to restore**

A complete system restore involves restoring all partitions on that system.

Table 3.2-8 presents the steps required to restore a partition in a condensed manner. If you are already familiar with the procedure, you may prefer to use the quick-step table. If you are new to the system, or have not performed these tasks recently, you should use the detailed procedures presented below.

To restore a partition for the requester, execute the procedures steps that follow:

1 Log into the backup server by typing: **telnet machine to be restored,** then press **Return**.

2 If a **Login:** prompt appears, log in as yourself by typing: ***Your User ID***, then press **Return**.

- A password prompt is displayed.

3 Enter ***Your Password***, then press **Return.**

- Remember that ***Your Password*** is case sensitive.
- You are authenticated as yourself and returned to the UNIX prompt.

4 Set display to current terminal by typing: **setenv DISPLAY *IPNumber*:0.0** on the current machine or **setenv DISPLAY on the machine to be restored,** then press **Return**.

5 Execute the **NetWorker Administrative** program by entering: **nwadmin &,** then press **Return**.

- A window opens for the NetWorker Administrative program.
- You are now able to perform restores of partitions.

6 Go to the **Save Set** menu, select **Recover Set.**

- The **Save Set Recover** window opens.

7 Select the **Name of system to be restored** (referred to as **System** in the rest of this procedure) in the **Client** field's menu.

- The **Save Set** listing updates. This is a listing of partitions on the **System**.
- At this time, note the partitions listed for the **System**. To do a complete system restore, this procedure needs to be performed for each partition listed.

8 Select the **Save Set**/partition from the listing.

- The **Instance** listing updates.

9 Select the appropriate **Instance**.

- An Instance is a particular Networker client backup. A listing of Instances is a report detailed with the Networker client backups that have occurred.
- Select an Instance based upon the Date from which to restore(referred to as Date in the rest of this procedure) and an appropriate level:

*__NOTE__ 1: To determine a base **Date**, you must consider the time of day that backups occur. For example, if the backups occur at 02:00 each morning, then a system corrupted at noon on June 6th would require a restoration of the June 6th backup. If the Backups are full or incremental, perform the following actions: Select the most recent full backup that occurred on or prior to the **Date** and perform a partition restore. If the date of this full backup is not the same as the **Date**, perform a partition restore using each incremental backup, in chronological order, between this full backup and the day after the **Date**.

If the backups are of different numerical levels, follow these steps:

1) Select the most recent level **0/full backup** prior to or on the **Date** and perform a restore of the partition.
2) If a level **0/full backup** did not occur on the **Date**, select the most recent backup of the next highest level occurring after this level **0** and prior to or on the **Date**.
3) Perform a restore of the partition.
4) Continue to select the most recent backup of the next highest level occurring between the last used **Instance** and the day after the **Date** until reaching an instance on the **Date**.

- You can double click an **Instance** to see which tape is required.

10 Click the **Recover** button.

- The Save Set Recover Status window opens.
- Clicking the Volumes button will show which tapes are required.

11 Click the **Options** button.

- The **Save Set Recover** Options window opens.

12 Set **Duplicate file resolution** to **Overwrite existing file** by clicking its radio button.

13 Make sure that the **Always prompt** checkbox is not checked.

14 Click the **OK** button.

- The Save Set Recover Options window closes.

15 Click the **Start** button in the **Save Set Recover Status** window.

- Status messages appear in the Status box.
- If prompted for tapes, click the Cancel button in the **Save Set Recover Status** window and follow steps **1-18** of procedure **3.2.5.2** Index tapes(or steps 1-19 of procedures **3.2.5.2** Index Tapes Quick Steps)
- A **recovery complete** message appears when recovery is complete.

16 Click the **Cancel** button after the **recovery complete** message appears.

- The Save Set Recover Status window closes.

17 If additional partition restores are required, go to step **8**. Otherwise, select **Exit** from the **File** menu to quit the NetWorker Administrative program.

18 At the UNIX prompt for the backup server, type **exit**, then press **Return**.

19 Type **exit** again, then press **Return**.

To restore a partition, execute the steps provided in the following table.

Table 3.2-8. Restore a Partition - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	**telnet** ***to the host which requires partition restoration***	**press Return**
2	*Your User ID*	**press Return**
3	*Your Password*	**press Return**
4	**setenv DISPLAY** *IPNumber*:**0.0 on the current machine -or-** **setenv DISPLAY** *BackupServerName*:**0.0**	**press Return**
5	**nwadmin &**	**press Return**
6	**Save Set → Recover Set**	(no action)
7	**Client/System**	(no action)
8	**Save Set/partition**	(no action)
9	**Instance**	**click Recover button**
10	(No entry)	**click Options button**
11	**Overwrite existing file**	**deselect Always prompt**
12	(No entry)	**click OK button**
13	(No entry)	**click Start button**
14	(No entry)	**click Cancel button**
15	(No entry)	**go to step 8 -or-** **select File → Exit**
16	**exit**	**press Return**
17	**exit**	**press Return**

3.2.5 Tape Handling

The following procedures demonstrate how to label tapes, index tapes, and clean tape drives. Each of these procedures contains detailed steps that explain how to complete the procedure properly. Each tape handling procedure is significant in maintaining a working backup system. DAAC scheduled backups depend on proper maintenance of tape media and tape drives. Listed are complete explanations of the procedures and their relevance to the Computer Operator position.

The Activity Checklist table that follows provides an overview of the tape handling process.

Table 3.2-9. Tape Handling - Activity Checklist

Order	Role	Task	Section	Complete?
1	Operator	Labeling Tapes	(I) 3.2.5.1	
2	Operator	Indexing Tapes	(P) 3.2.5.2	
3	Operator	Tape Drive Cleaning	(P) 3.2.5.3	

3.2.5.1 Labeling Tapes

The Tape Labeling process begins when the Operator is performing procedures 3.2.1 Incremental Backup or 3.2.2 Full Backup (or their associated Quick Steps) and runs out of tapes. The tape(s) must be installed in the jukebox and labeled. NetWorker uses tape labels for identification. The label that NetWorker creates is on the tape media itself, rather than a sticker on the outside of the tape cassette. An index is kept by NetWorker associating tape labels with particular backups/data. When you select files to be recovered using the NetWorker Recovery window or view saved sets on a backup volume using the Volume Management window in NetWorker, you are viewing this index. After labeling the required tape(s), the Operator resumes procedure 3.2.1 or 3.2.2.

Table 3.2-10. Labeling Tapes - Activity Checklist

Order	Role	Task	Section	Complete?
1	Operator	Install Required Tape(s) in Jukebox	(P) 3.2.5.1.1	
2	Operator	Label 8mm Tapes	(P) 3.2.5.1.2	
3	Operator	Label DLT Tapes	(P) 3.2.5.1.3	

Detailed procedures for tasks performed by the Operator are provided in the sections that follow.

3.2.5.1.1 Install Required Tape(s) in Jukebox

The procedures assume that the Operator was previously executing procedure 3.2.1 or 3.2.2. In order to perform the procedures, the Operator must have obtained the following:

a. **Blank tape(s)**

All tapes are stored in the storage cabinet located in the control room. There are five tapes in each box, and every box of tapes has a unique number. To begin finding tapes for recycling to be labeled and installed in the Juke box, the lowest numbers of a tape or a box of tapes should be used. Do not recycle any tape or box of tapes that the numbers are higher or current.

3.2.5.1.2 8mm or D3 Tapes Labeling Process

Table 3.2.5-2 presents the steps required to label tapes in a condensed manner. If you are already familiar with the procedure, you may prefer to use the quick-step table. If you are new to the system, or have not performed this task recently, you should use the detailed procedure presented below.

To label tapes, execute the procedure steps that follow:

1 Log into the **backup server** by typing: **telnet** ***appropriate server (i.e. gsfcsrvr7),*** then press **Return**.

2 If a **Login** prompt appears, log in by typing: ***Your User ID***, then press **Return**.

- A password prompt is displayed.

3 Enter ***Your Password***, then press **Return**.

- Remember that ***Your Password*** is case sensitive.
- You are authenticated as yourself and returned to the UNIX prompt.

4 Set display to current terminal by typing: **setenv DISPLAY *IPNumber*:0.0** or **setenv DISPLAY *BackupServerName*:0.0**, then press **Return**.

5 Execute the **NetWorker Administrative** program windows by entering: **nwadmin &**, then press **Return**.

- The **NetWorker Administrative** program windows displayed
- Remove all non-blank tapes from the cartridge.
- ***Dismount the drive(s) that will be used for tapes Labeling***

6 Insert the blank tape(s) in the jukebox's cartridge, then install the cartridge in the jukebox.

7 Click the **Label** button from the menu bar.

- The **Jukebox Labeling** window opens.

8 Enter **tape one** in the non-removable **Slot field**.

- **Slot 1** is at the top of the cartridge, and it is a non-removable slot. **Slot 2** through **Slot 11** are removable slots. **Slot 11** contains a cleaning tape. Do not enter any tape in Slot 11 for labeling. The default setting is **1** through **10** for only the tapes that will be labeled and used for backup.
- It is ok to have empty slots.

9 Click the **OK** button.

- A status message appears and updates.
- Labeling a full cartridge of tapes takes about one and half hours. Nine Minutes per tape.

10 When the status in the **Jukebox Labeling** window reads finished, click the **Cancel** button.

- The **Jukebox Labeling** window closes.

11 Go to the **File** menu, select **Exit**.

12 Put a sticker on the outside of each tape cassette.

- This is done in order for you to identify each tape.

13 Resume procedure 3.2.1 or 3.2.2.

To label tapes, execute the steps provided in the following table.

3.2.5.1.3 Label DLT Tapes

The dlt tape labeling process is the same as the 8mm tape labeling scenarios, except for some few things that are additionally different. Steps to follow in labeling the dlt tapes as follow: Repeat

Table **3.2.5.1** step 1 through step 9. Change default to **SPRDLT,** then click OK button. The system will display the beginning number of the tape label. Once again repeat step 11 through step 17 of Table **3.2.5.1**, tape labeling process to end the task.

Table 3.2-11. Label Tapes - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	**telnet** *to* ***appropriate server (i.e. gsfcsrvr7)***	**press Return**
2	*Your User ID*	**press Return**
3	*Your Password*	**press Return**
4	No Entry	**press Return**
5	**nwadmin &**	**press Return**
6	(No entry)	**put blank tape(s) in cartridge and install cartridge in jukebox**
7	(No entry)	**click Label button**
8	**2**	(No action)
9	**11**	**click OK button**
10	(No entry)	**click Cancel button**
11	**File → Exit**	(No action)
12	**exit**	**press Return**
13	**exit**	**press Return**
14	(No entry)	**put a sticker on the outside of each tape**
15	(No entry)	**resume previous procedure**

3.2.5.2 Indexing Tapes

The Indexing Tapes process begins when the Operator has finished performing procedures 3.2.5.1, (**Tape Labeling**). If the tape(s) is/are not ***indexed/inventoried,*** Networker will not be aware of it/them. After indexing the required tape(s), the Operator resumes procedure 3.2.1 or 3.2.2.

The Activity Checklist table that follows provides an overview of the indexing tapes process.

Table 3.2-12. Indexing Tapes - Activity Checklist

Order	Role	Task	Section	Complete?
1	Operator	Pull Required Tape(s) from Tape Storage.	(I) 3.2.5.2.1	
2	Operator	Index Tapes	(P) 3.2.5.2.2	

3.2.5.2.1 Pull Required Tape(s) from Tape Storage

In order to perform the procedure, the Operator must have obtained the following:

a. **The required tape(s)**

3.2.5-2 Index Tapes

Detailed procedures for tasks performed by the Operator are provided in the sections that follow.

The procedures assume that the Operator has previously executed procedure 3.2.5.1, **Tape Labeling**.

Table **3.2-5-2** presents the steps required to index tapes in a condensed manner. If you are already familiar with the procedure, you may prefer to use the quick-step table. If you are new to the system, or have not performed this task recently, you should use the detailed procedure presented below.

To index tapes, execute the procedure steps that follow:

NOTE: You may proceed to step **8** if you are still logged into the backupserver.

1 Log into the **backup server** by typing: **telnet to *appropriate server (i.e. gsfcsrvr7),*** then press **Return**.

2 If a **Login** prompt appears, log in as yourself by typing: ***Your User ID***, then press **Return**.

- A password prompt is displayed.

3 Enter ***Your Password***, then press **Return.**

- Remember that your password is case sensitive.
- You are authenticated as yourself and returned to the Unix prompt.

4 Set display to current terminal by typing: **setenv DISPLAY *IPNumber*:0.0** or **setenv DISPLAY *Backup Server Name*:0.0**, then press **Return**.

5 Execute the **Networker Administrative** program by entering: **nwadmin &,** then press **Return**.

- The Networker Administrative program windows is displayed.
- You are now able to index tapes.
- Click the Mount button to show what tapes **Networker** is currently aware of. The **Jukebox Mounting** windows opens. Once you have finished with this window, click the **Cancel** button.

6 Put the **required tape(s)** in the jukebox's cartridge, install the cartridge in the jukebox.

- For instructions, refer to the jukebox's documentation.

7 Go to the **Media** menu, select **Inventory**.

- The **Jukebox Inventory** window opens.

8 Enter **1** in the **First Slot** field, enter **10** in the **Last Slot** field.

- Slot 1 is the non-removable slot within the jukebox.
- Slot 2 is the first top of the removable cartridge and 11 at the bottom, and it contains a cleaning tape. A default setting 1 through 10.
- It is OK to have empty slots or slots with tapes which have already been indexed.

9 Click the **OK** button.

- A checking volume message appears and updates.
- Performing an inventory on a full cartridge takes about forty minutes.

10 When the status in the **Jukebox Inventory** window says finished, click the **Cancel** button.

- The **Jukebox Inventory** window closes.

11 Click the **Mount** button to verify that the indexing worked.

- The Jukebox Mounting window opens.
- The required tape(s) should be shown. If not, repeat from step 8.

12 Click the **Cancel** button.

- The **Jukebox Mounting** window closes.

13 Go to the **File** menu, select **Exit**.

14 At the UNIX prompt for the ***backup server***, type **exit**, then press **Return**.

15 Type **exit** again, then press **Return**.

16 Resume procedure 3.2.3 at step 12, procedure 3.2.3 at quick-step 11 - action part, procedure 3.2.4 at step 12, or procedure 3.2.4 at quick-step 11 - action part.

To index tapes, execute the steps provided in the table.

Table 3.2-12. Index Tapes - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	**telnet** ***appropriate server (i.e. gsfcsrvr7)***	**press Return**
2	*Your UserID*	**press Return**
3	*Your Password*	**press Return**
4	**setenv DISPLAY** *IPNumber*:**0.0 -or-** **setenv DISPLAY** *BackupServerName*:**0.0**	**press Return**
5	**nwadmin**	**press Return**
6	(No entry)	**put required tape(s) in cartridge and install cartridge in jukebox**
7	**Media → Inventory**	(No action)
8	**2**	(No action)
9	**1 1**	**click OK button**
10	(No entry)	**click Cancel button**
11	(No entry)	**click Mount button**
12	(No entry)	**verify indexing**
13	(No entry)	**click Cancel button**
14	**File → Exit**	(No action)
15	**exit**	**press Return**
16	**exit**	**press Return**
17	(No entry)	**resume previous procedure**

3.2.5.3 Tape Cleaning Process

The system will at times prompt for drive(s) cleaning. This happens usually during non processing periods, however during the course of the tape backup process period, the drive(s) may send a request for cleaning. Manual cleaning should be performed each time tapes are installed in the Juke Box. Maintaining clean drives can help prevent backup interruption which may occur due to unclean tape drive heads. If the system is prompted for drive(s) cleaning: Follow the details steps below:

(1) Log into the Backup Server by typing: **telnet** ***appropriate server (i.e. gsfcsrvr7)***, then press **Return.**

- The system prompts for your user's ID

(2) Type your user's ID, then press **Return.**

- A **password** prompt displayed.

(3) Type your **password,** then press **Return.** Remember, your password is case sensitive.

(4) Execute Net Worker by typing: **nwadmin &**, then press Return.

- A NetWorker administrative program windows displayed.

(5) High light the desirable drive(s) that the system has prompted for cleaning.

(6) Click dismount from the menu bar and wait a few minutes for the drive to be dismounted completely. Repeat step 6 on the second drive until the both are dismounted.

To open the Exabyte door turn the key in the door counter clockwise. The last tape at the bottom of the cartridge is the cleaning tape. Remove it from the slot field and insert it gently into each drive below. Wait until the tape has been ejected and the flashing lights on the drive are off before removing the tape from the drive. Insure that the cleaning tape is still usable before each use. Cleaning tapes will expire after several uses. After each use mark the appropriate box on the surface of the tape to maintain a list of usage. Insert the cleaning tape back into the last slot and lock the Exabyte door.

3.3 System Log Maintenance

The System Log Maintenance process is performed through Tivoli by the System Administrator (SA). The System Administrator will setup and execute the jobs to be run in Tivoli in various formats (i.e., reoccurring day and time, maximum amount of disk space, etc.). This section assumes that task jobs have already been created and discusses how to edit the job for System Log Maintenance.

The procedures assume that system log maintenance has been scheduled well in advance, all planning involved has been concluded well in advance, and the SA has been properly trained to perform all aspects of system log maintenance, and that the SA is a Tivoli administrator.

This section explains how to edit a task in Tivoli for System Log Maintenance. To begin, execute the procedure steps that follow:

1 Log into the **Tivoli server** by typing: **telnet *TivoliServerName* or rsh *TivoliServerName*** at the UNIX prompt, then press **Return**.

2 If a **Login**: prompt appears, log in as yourself by typing: ***YourUserID***, then press **Return**.

- A password prompt is displayed.

3 Enter ***YourPassword***, then press **Return**.

- Remember that ***YourPassword*** is case sensitive.
- You are authenticated as yourself and returned to the UNIX prompt.

4 Enter **setenv DISPLAY *IPNumber*:0.0** *-OR -* **setenv DISPLAY *MachineName*:0.0** then press **Return**.

5 Enter **tivoli,** then press **Return**.

- The **TME Desktop Administrator** window appears.

6 Double-click on the **Scheduler** icon.

- The **Scheduler Browser** window appears.

7 Highlight the job you wish to edit by clicking the job.

8 Select **Edit** from the menu at the bottom of the screen.

- The **Edit Scheduled Job** window appears.

9 Enter all desired changes.

10 Select **Update & Close** from the menu at the bottom of the window.

- You are logged out of and disconnected from the **TME Desktop Administrator**.

11 Type **exit**, then press **Return**.

- You are logged out of the **Tivoli** server.

Table 3.3-1 presents the **QUICK STEP** procedures required to perform System Log Maintenance.

Table 3.3-1. System Log Maintenance - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	*YourUserID*	press Return
2	*YourPassword*	press Return
3	setenv DISPLAY *IPNumber*:0.0 -or- setenv DISPLAY *MachineName*:0.0	press Return
4	tivoli	press Return
5	(No entry)	double-click on Scheduler Icon
6	job	single-click
7	(No entry)	single-click on Edit button
8	make changes to job	
9	(No entry)	single-click on Update & Close button
10	exit	press Return

3.4 User Administration

Note: User Administration procedures will be implemented through command-line and/or script entries. The Tivoli Management Environment (TME) will not be used for User Administration procedures.

3.4.1 Adding a User

The Adding a User process begins when the requester fills out a "User Registration Request Form" (located in Appendix A), and submits it to the site supervisor. The "User Registration Request Form" includes information regarding the user (User's Name, Group, Organization, etc.), as well as the user's explanation of why an account on the system is needed. The requester's supervisor reviews the request, and if it is determined that it is appropriate for the requester to have UNIX and DCE accounts, forwards the request to the Operations Supervisor (OPS Super). The OPS Super reviews the request and forwards it to the System Administrator (SA). The SA verifies that all required information is contained on the form. If it is, the SA implements the request. (Incomplete forms are returned to the requester's supervisor for additional information.) After the user is registered, the SA provides the user with a password to use for logging onto their accounts. The SA also sends an e-mail message to the user's supervisor and the OPS Super, informing them that the user's accounts were created.

The **Activity Checklist** table that follows provides an overview of the adding a user process.

Table 3.4-1. Adding a User - Activity Checklist

Order	Role	Task	Section	Complete?
1	Requester	Complete User Registration Form and forward to the Supervisor.	(I) 3.4.1	
2	Super	Approve/Deny Request. If Approve, Forward Request to OPS Super.	(I) 3.4.1	
3	OPS Super	Review Request and Forward to SA.	(I) 3.4.1	
4	SA	Review User Registration Form for Completeness.	(I) 3.4.1	
5	SA	Add User.	(P) 3.4.1	
6	SA	Phone/e-mail User with Password. Notify Supervisor and OPS Super that user was added.	(I) 3.4.1	

Depending upon the script utilized, in order to add a new user the SA should obtain information such as the following about the requester:

a. **Real name of the new user**

b. **User name of the new user**

c. **Office number of the new user**

d. **Office phone number of the new user**

e. **Home phone number of the new user**

f. **Organization**

g. **Group affiliation(s)**

h. **Role(s) of the new user**

The SA creates a new user account with command-line/script entries. As an example, The Goddard Space Flight Center DAAC uses a script, *Newuser*, to add new users to the system. The script, which is available to other DAACs, walks the System Administrator through data input of user information, checks for the same user in other systems, creates a User ID, synchronizes password files and creates home directories for new users.

3.4.2 Deleting a User

The Deleting a User process begins when the requester has determined that no useful files remain in the user's home directory and submits a request to delete the user's account to the user's supervisor. The supervisor approves or denies the request. Once approved, the request is forwarded to the OPS Super. The OPS Supervisor reviews the request and forwards it to the SA, who deletes the user's account. When the user has been deleted, the SA notifies the requester, supervisor and OPS Super.

The Activity Checklist table that follows provides an overview of the deleting a user account process.

Table 3.4-2. Deleting a User - Activity Checklist

Order	Role	Task	Section	Complete?
1	Requester	Determine that No Useful Files Remain in the User's Home Directory and Submit Request to user's Supervisor.	(I) 3.4.2	
2	OPS Super	Approve/Deny Request. If Approve, Forward Request to OPS Super.	(I) 3.4.2	
3	OPS Super	Review Request and Forward to SA.	(I) 3.4.2	
4	SA	Delete User.	(P) 3.4.2	
5	SA	Notify Requester, Supervisor and OPS Super that user was deleted.	(I) 3.4.2	

The process assumes that the requester's application for deleting a user has already been approved by DAAC Management. In order to perform the procedure, the SA must have obtained the following information from the requester:

a. **UNIX login of the user to be deleted**

a. **Role(s) of the user to be deleted**

The SA deletes a user with command-line/script entries. As an example, The Goddard Space Flight Center DAAC uses a script, *Lockdown,* to lock, unlock and delete user accounts. This script, which is available to other DAACs, walks the System Administrator through the steps necessary to delete a user account. It assists the System Administrator in locating the correct user account for deletion, deletes the user account and all associated file references. It also enables the System Administrator to lock or unlock accounts.

3.4.3 Changing a User Account Configuration

The Changing a User Account Configuration process begins when the requester submits a request to the OPS Supervisor detailing what to change about the account configuration and the reason for the change. The OPS Supervisor reviews the request and forwards it to SA who changes the user's account configuration. When the changes are complete the SA notifies the requester and OPS Supervisor.

The Activity Checklist table that follows provides an overview of the changing a user account configuration process.

Table 3.4-3. Change a User Account Configuration - Activity Checklist

Order	Role	Task	Section	Complete ?
1	Requester	Submit Request to OPS Supervisor.	(I) 3.4.3	
2	OPS Super	Review and Forward to SA.	(I) 3.4.3	
3	SA	Change User Account Configuration.	(P) 3.4.3	
4	SA	Inform Requester and Supervisor of completion.	(I) 3.4.3	

The process assumes that the requester's application for changing a user account configuration has already been approved by the OPS Supervisor. In order to perform the procedure, the SA must have obtained the following information from the requester:

a. **What to change and new settings.**

Can be any of:

New Real User Name

New Login ID

New Office Number

New Office Phone Number

New Home Phone Number

New UNIX Group

New DCE Group

New DCE Organization

New Login Shell

b. **Current UNIX Login of the User**

The SA changes the appropriate configuration items manually in the users home directory.

3.4.4 Changing User Access Privileges

The Changing User Access Privileges process begins when the requester submits a request to the supervisor. The supervisor approves or denies the request. Once approved, the request is forwarded to the OPS Super. The OPS Super reviews the request and forwards it to the SA who changes the user's access privileges. When the changes are complete the SA notifies the requester, supervisor and OPS Super.

The Activity Checklist table that follows provides an overview of the changing user access privileges process.

Table 3.4-7. Changing User Access Privileges - Activity Checklist

Order	Role	Task	Section	Complete?
1	Requester	Submit Request to the Supervisor.	(I) 3.4.4	
2	Super	Approve/Deny Request. If Approve, Forward Request to OPS Super.	(I) 3.4.4	
3	OPS Super	Review Request and Forward to SA.	(I) 3.4.4	
4	SA	Change User Access Privileges.	(P) 3.4.4	
5	SA	Inform Requester, Supervisor and DAAC Mgr of completion.	(I) 3.4.4	

Detailed procedures for tasks performed by the SA are provided in the sections that follow.

The procedures assume that the requester's application for changing user access privileges has already been approved by DAAC Management and that the SA is an administrator. In order to perform the procedure, the SA must have obtained the following information about the requester:

a. **Role(s) to which the user is to be added**

b. **Role(s) from which the user is to be removed**

c. **UNIX login of the user**

To change user access privileges for the requester, execute the procedure steps that follow:

3.4.5 Changing a User Password

The Changing a User Password process begins when the requester submits a request to the SA. The SA verifies that the requester is who s/he claims to be. Once verified, the SA changes the user's password. When the change is complete the SA notifies the requester.

The **Activity Checklist** table that follows provides an overview of the changing a user password process.

Table 3.4-9. Changing a User Password - Activity Checklist

Order	Role	Task	Section	Complete?
1	Requester	Submit Request to SA.	(I) 3.4.5	
2	SA	Verify that the Requester is Who S/he Claims to Be.	(I) 3.4.5	
3	SA	Change Password.	(P) 3.4.5	
4	SA	Inform Requester of completion.	(I) 3.4.5	

Detailed procedures for tasks performed by the SA are provided in the sections that follow.

The procedures assume that the requester's application for changing a user password has already been approved by DAAC Management and that the SA is a Tivoli administrator. In order to perform the procedure, the SA must have obtained the following information about the requester:

a. **UNIX login of the user**

b. **New password for the user**

To change a user password for the requester, execute the procedure steps that follow:

3.4.6 Checking a File/Directory Access Privilege Status

The Checking a File/Directory Access Privilege Status process begins when the requester submits a request to the SA. The SA checks the file/directory access privilege status and reports the status back to the requester.

The **Activity Checklist** Table 3.4-11 that follows provides an overview of the checking a file/directory access privilege status process.

Table 3.4-11. Checking a File/Directory Access Privilege Status - Activity Checklist

Order	Role	Task	Section	Complete ?
1	Requester	Submit a Request to the SA.	(I) 3.4.6	
2	SA	Check a File/Directory Access Privilege Status.	(P) 3.4.6	
3	SA	Inform Requester of completion and Report the File/Directory Access Privilege Status.	(I) 3.4.6	

Detailed procedures for tasks performed by the SA are provided in the sections that follow. In order to perform the procedure, the SA must have obtained the following information about the requester:

a. **full path of the file/directory on which privilege status is needed**

Table 3.4-12 contains a table which presents the steps required to check a file/directory access privilege status in a condensed manner. If you are already familiar with the procedure, you may prefer to use the quick-step table. If you are new to the system, or have not performed this task recently, you should use the detailed procedure presented below.

To check a file/directory access privilege status for the requester, execute the procedure steps that follow:

1 At a UNIX prompt, type **cd *Path***, press **Return**.

- The ***Path*** is the full path up to but not including the file/directory on which access privilege status is needed. For example, if the requester wants access privileges status on directory /home/jdoe then type **cd /home** and press **Return**.

2 Type **ls -la | grep *FileOrDirectoryName***, press **Return**.

This command will return information like this:

```
drwxr-xr-x   19 jdoe user      4096 Jun 28 09:51 jdoe
-r-xr--r--    1  jdoe user      80   Jun 22 11:22 junk
```

What this output means, from left to right, is:

The file type and access permissions:

The *first character* indicates what type of file it is:

d means that the file is a directory.
- means that the file is an ordinary file.
l means that the file is a symbolic link.

The *next three characters* indicate the owner privileges, in the order: **r** = read **w** = write **x** = execute. **-** is a place holder. ***Example:*** the owner (jdoe) of the file ***junk*** does not have *write* permissions, so a - appears rather than a w.

The *next three characters* indicate the group privileges, in the order: **r** = read **w** = write **x** = execute. **-** is a place holder. ***Example:*** the group (user) of the directory ***jdoe*** does not have write permissions, so a - appears rather than a w as the sixth character in the line.

The *next three characters* indicate the privileges that everyone else/other has, in the order: **r** = read **w** = write **x** = execute. **-** is a place holder.
Example: other in the case of the directory ***jdoe*** does not have write permissions, so a - appears rather than a w as the ninth character in the line.

There are 19 links to the file/directory ***jdoe***.
The owner of the file/directory is jdoe.
The file/directory's group is user.
The file/directory is 4096 bytes large.
The last time the file/directory was modified is Jun 28 at 09:51.
The name of the file/directory is jdoe.

3 Create a report of the file/directory's access privilege status by using the information produced by step 2 and by filling out this template:

full path of the file/directory: _____

owner: _____

group: _____

owner/user privileges: _____ **read** _____ **write** _____ **execute**

group privileges: _____ **read** _____ **write** _____ **execute**

everyone else/other privileges: _____ **read** _____ **write** _____ **execute**

To check a file/directory access privilege status, execute the steps provided in the following table.

Table 3.4-12. Check a File/Directory Access Privilege Status - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	**cd *Path***	**press Return**
2	**ls -la | grep *FileOrDirectoryName***	**press Return**
3	(No entry)	**generate a file/directory access privilege status report**

3.4.7 Changing a File/Directory Access Privilege

The Changing a File/Directory Access Privilege process begins when the requester submits a request to the supervisor to have file/directory access privileges changed. The supervisor approves/denies the request. When approved, the request is forwarded to the OPS Supervisor who reviews the request and forwards it to the SA. The SA changes the file/directory access privileges and then notifies the requester, supervisor and OPS Supervisor of completion.

The **Activity Checklist** table that follows provides an overview of the changing a file/directory access privilege process.

Table 3.4-13. Changing a File/Directory Access Privilege - Activity Checklist

Order	Role	Task	Section	Complete ?
1	Requester	Submit Request to the Supervisor.	(I) 3.4.7	
2	Super	Approve/Deny Request. If Approve, Forward Request to OPS Supervisor.	(I) 3.4.7	
3	OPS Super	Review Request and Forward to SA.	(I) 3.4.7	
4	SA	Change a File/Directory Access Privilege.	(P) 3.4.7	
5	SA	Inform Requester, Supervisor and OPS Supervisor of completion.	(I) 3.4.7	

Detailed procedures for tasks performed by the SA are provided in the sections that follow.

The procedures assume that the requester's application for changing a file/directory access privilege has already been approved by the supervisor. In order to perform the procedure, the SA must have obtained the following information about the requester:

a. **full path of the file/directory on which access privileges will be changed**

b. **new access privileges to set on the file/directory. Can be any of:**

New owner

New group

New user/owner privileges (read, write and/or execute)

New group privileges (read, write and/or execute)

New other privileges (read, write and/or execute)

To change a file/directory access privilege for the requester, execute the procedure steps that follow:

1 At the UNIX prompt, type **su**, press **Return**.

2 At the **Password** prompt, type ***RootPassword***, press **Return**.

- Remember that ***RootPassword*** is case sensitive.
- You are authenticated as root.

3 Type **cd *Path***, press **Return**.

- ↑ The ***Path*** is the full path up to but not including the file/directory on which access privileges will be changed. For example, if the requester wants access privileges changed on directory /home/jdoe then type **cd /home** and press **Return**.

4 If there is a **New owner** then type **chown *NewOwner FileOrDirectoryName***, press **Return**.

- ↑ The ***FileOrDirectoryName*** is the name of the file/directory on which access privileges will be changed minus the path. For example, if the requester wants access privileges changed on directory /home/jdoe then type: (You must be /home)
chown *NewOwner* jdoe and press **Return**.

5 If there is a **New group** then type **chgrp *NewGroup FileOrDirectoryName***, press **Return**.

- ↑ The ***FileOrDirectoryName*** is the name of the file/directory on which access privileges will be changed minus the path. For example, if the requester wants access privileges changed on directory /home/jdoe then type:
chgrp *NewGroup* jdoe and press **Return**.

6 If there are **New user/owner privileges** then type **chmod u=*NewUserPrivileges FileOrDirectoryName***, press **Return**.

- ↑ The ***FileOrDirectoryName*** is the name of the file/directory on which access privileges will be changed minus the path. For example, if the requester wants access privileges changed on directory /home/jdoe then type:
  chmod u=*NewUserPrivileges* jdoe and press **Return**.

- ↑ The ***NewUserPrivileges*** are **r** = read **w** = write **x** = execute. To give the user/owner read, write and execute privileges, type:
  chmod u=rwx *FileOrDirectoryName* and press **Return**.

7 If there are **New group privileges** then type **chmod g=*NewGroupPrivileges FileOrDirectoryName***, press **Return**.

- ↑ The ***FileOrDirectoryName*** is the name of the file/directory on which access privileges will be changed minus the path. ***Example:*** if the requester wants access privileges changed on directory /home/jdoe then type: (You must be in /home)
  chmod g=*NewGroupPrivileges* jdoe and press **Return**.

- ↑ The ***NewGroupPrivileges*** are **r** = read **w** = write **x** = execute.
  Example: to give the group read and execute privileges, type:
  chmod g=rx *FileOrDirectoryName* and press **Return**.

8 If there are **New other privileges** then **type:**
chmod o=*NewOtherPrivileges FileOrDirectoryName*, and press **Return**.

- ↑ The ***FileOrDirectoryName*** is the name of the file/directory on which access privileges will be changed minus the path. For example, if the requester wants access privileges changed on directory /home/jdoe then type:
  chmod o=*NewOtherPrivileges* jdoe, and press **Return**.

- ↑ The ***NewOtherPrivileges*** are r for read, w for write and x for execute. For example, to give other read privileges, type:
  chmod o=r *FileOrDirectoryName* and press **Return**.

9 Type **exit**, press **Return**.

- Root is logged out.

To change a file/directory access privilege, execute the steps provided in the following table.

Table 3.4-14 contains a table which presents the steps required change a file/directory access privilege.

Table 3.4-14. Change a File/Directory Access Privilege - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	**su**	**press Return**
2	***RootPassword***	**press Return**
3	**cd *Path***	**press Return**
4	**chown *NewOwner FileOrDirectoryName***	**press Return**
5	**chgrp *NewGroup FileOrDirectoryName***	**press Return**
6	**chmod u=*NewUserPrivileges FileOrDirectoryName***	**press Return**
7	**chmod g=*NewGroupPrivileges FileOrDirectoryName***	**press Return**
8	**chmod o=*NewOtherPrivileges FileOrDirectoryName***	**press Return**
9	**exit**	**press Return**

3.4.8 Moving a User's Home Directory

The Moving a User's Home Directory process begins when the requester submits a request to the OPS Supervisor. The OPS Supervisor approves or denies the request. Once approved, the request is forwarded to the SA who moves the user's home directory. When the changes are complete the SA notifies the requester and OPS Supervisor.

The Activity Checklist table that follows provides an overview of moving a user's home directory process.

Table 3.4-15. Moving a User's Home Directory - Activity Checklist

Order	Role	Task	Section	Complete ?
1	Requester	Submit Request to OPS Supervisor.	(I) 3.4.8	
2	OPS Super	Approve/Deny Request in Accordance with Policy. Forward to SA if approved.	(I) 3.4.8	
3	SA	Move a User's Home Directory.	(P) 3.4.8	
4	SA	Inform Requester and OPS Super of completion.	(I) 3.4.8	

Detailed procedures for tasks performed by the SA are provided in the sections that follow.

The procedures assume that the requester's application for moving a user's home directory has already been approved by DAAC Management and that the SA is an administrator. In order to perform the procedure, the SA must have obtained the following information about the requester:

a. **UNIX login of the user**

b. **New location for home directory**

To move a user's home directory for the requester, execute the procedure steps that follow:

3.5 Installing a New Workstation

The Installing a New Workstation process has three stages - preparation, installation and testing, and verification. The preparation stage begins with the System Administrator (SA) performing procedure 3.5.1.1 Hardware Preparation. Once the hardware is prepared, the SA continues on and prepares for network configuration, procedure 3.5.1.2.

The next stage is installation which begins by reporting the hardware to inventory, procedure 3.5.2.1.1. The SA continues this stage by installing the operating system, procedure 3.5.2.2. Installation is then completed by installing custom software, procedure 3.5.2.3.1, and installing COTS software, procedure 3.5.2.3.2.

The final stage is testing and verification. The SA begins this stage by rebooting the machine, procedure 3.5.3.1. The testing and verification is completed by logging in, procedure 3.5.3.2, and testing the environment, procedure 3.5.3.3.

The Activity Checklist table that follows provides an overview of the New Workstation Installation process.

Table 3.5-1. Installing a New Workstation - Activity Checklist

Order	Role	Task	Section	Complete?
1	SA	Prepare Hardware	(P) 3.5.1.1	
2	SA	Prepare for Network Configuration	(P) 3.5.1.2	
3	SA	Report Hardware to Inventory	(P) 3.5.2.1.1	
4	SA	Install Operating System	(P) 3.5.2.2	
5	SA	Install Custom Software	(P) 3.5.2.3.1	
6	SA	Install COTS Software	(P) 3.5.2.3.2	
7	SA	Reboot	(P) 3.5.3.1	
8	SA	Log In	(P) 3.5.3.2	
9	SA	Test Environment	(P) 3.5.3.3	

3.5.1 Preparation

3.5.1.1 Hardware Preparation

The Hardware Preparation process begins when the requester submits a request to the SA. The SA then determines if the requested hardware is on hand or must be ordered. Once the hardware is available along with all the necessary attachments, the SA will schedule the installation. After the Hardware Preparation is complete, the SA proceeds to procedure 3.5.1.2, Network Configuration.

The Activity Checklist table that follows provides an overview of the Hardware Preparation process.

Table 3.5-2. Hardware Preparation - Activity Checklist

Order	Role	Task	Section	Complete?
1	Requester	Make request known to Operation Supervisor	(I) 3.5.1.1	
2	OPS Super	Submit a request to the DAAC Manager for approval	(I) 3.5.1.1	
3	DAAC Manager	Submit approved request to the SA	(I) 3.5.1.1	
4	SA	Determine if the requested hardware is on hand or must be ordered. Network is in place and IP Addresses have been assigned to hardware.	(I) 3.5.1.1	
5	SA	Schedule the Installation.	(I) 3.5.1.1	

Detailed procedures for tasks performed by the SA are provided in the sections that follow.

The procedures assume that the hardware installation has been scheduled well in advance, all planning involved has been concluded well in advance and the SA has been properly trained to install the hardware. The SA must obtain the following information from the requester:

a. **type of hardware desired (HP, Sun, SGI or NCD)**

b. **location of installation**

Refer to Section 3.3 of the Release A Installation Plan (800-TP-005-001) for detailed instruction on how to install hardware.

3.5.1.2 Network Configuration

The Network Configuration process begins after section 3.5.1.1 Hardware Preparation has been completed in the Installing a New Workstation process. Once complete, the SA proceeds to procedure 3.5.2.1.1 Reporting to Inventory.

Detailed procedures for tasks performed by the SA are provided in the sections that follow.

The procedures assume that procedure 3.5.1.1 Hardware Preparation has been completed and that the SA has been properly trained in network configuration.

The following steps are required to prepare for network configuration:

1 Determine the name of the hardware.

- For example, if the hardware is a NCD, the name will be ncd# where # is sequential from the inventory list. The standard naming convention follows: (ex. g0css02)
- x = DAAC site (i.e., g = goodard, l = langley, e = edc, n = nsidc)
- 0 = B0 machine
- c = characters for subsystem identification
- s = characters for subsystem identification
- s = machine type (i.e., s = sun, g = sgi, h = hp)
- 0x = x indicates the number of machines in that subsystem suite (?) (i.e., 01, 02, etc)

NOTE: If the hardware is a NCD, the SA needs to determine the name of the NCD Login Host. The NCD Login Host will be the name of the X-server this NCD will use.

2 Submit a request to the Resource Manager for the IP address and the DNS entry.

To prepare for network configuration, execute the steps provided in the following table.

Table 3.5-3 presents the **QUICK STEP** procedures required to prepare for network configuration.

Table 3.5-3. Prepare for Network Configuration - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	(No entry)	**determine name of workstation**
2	(No entry)	**determine NCD Login Host**
3	(No entry)	**submit request to Resource Manager for IP address and DNS entry**

3.5.2 Installation

3.5.2.1 Hardware

3.5.2.1.1 Reporting to Inventory

The Reporting to Inventory process begins after the SA has completed Section 3.5.1.1 Hardware Preparation and 3.5.1.2 Network Configuration. After the Reporting to Inventory is complete, the SA proceeds to procedure 3.5.2.2 Operating System Installation.

Detailed procedures for tasks performed by the SA are provided in the sections that follow.

The procedures assume that procedures 3.5.1.1 and 3.5.1.2 have been completed.

The following steps are required to report to inventory:

1 Locate the Inventory Control Number on each hardware component and record them.

- The Inventory Control Number is on a small bright sticker on the front of each hardware component.

2 Submit the Inventory Control Numbers and location of the machine to the Inventory Controller.

To report to inventory, execute the steps provided in the following table.

Table 3.5-4 presents the **QUICK STEP** procedures required to report to inventory.

Table 3.5-4. Report to Inventory - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	(No entry)	**locate and record the Inventory Control Numbers**
2	(No entry)	**report the Inventory Control Numbers and location of the machine to the Inventory Controller**

3.5.2.2 Operating System Installation - By Operating System Type

3.5.2.2.1 Solaris 2.5.1 Operating System Installation

Solaris 2.5.1 is also known as Sun OS 5.5.1. The Solaris 2.5.1 Operating System Installation process begins when procedure 3.5.2.1 Installation of Hardware has been completed in the Installing a New Workstation process. Once complete, the SA proceeds to procedure 3.5.2.3.1 Installation of Custom Software.

Detailed procedures for tasks performed by the SA are provided in the sections that follow.

The procedure assumes that procedures 3.5.1 Preparation and 3.5.2.1 Installation of Hardware have been completed. The procedure also assumes that the workstation is powered off.

This section explains how to install the Solaris 2.5.1 operating system, including network configuration and patch installation. If you would like a listing of the patches installed, please see document number 420-TD-012-001 Release A Sun Solaris Operating System Patch List attached at the end of this document. To install the Solaris 2.5.1 operating system, execute the procedure steps that follow:

1 Get the download disk.

2 Check that the download disk is set to be target 2.

- Facing the front of the download disk, the target number is found on the back, to the right on the disk.
- You can change the target number by hitting the pins above and below it.

3 Plug the download disk into the Sun.

4 Power on the download disk.

- Facing the front of the disk, the power switch is found on the back, to the left on the disk.

5 Power on the monitor, power on the Sun.

- The Sun's power switch is located on the back of the Sun. When facing the front of the Sun, the power switch is on the right.

6 At the > prompt, type **probe-scsi**, press **Return**.

- Verify that target 2 exists by finding it in the listing that appears.

7 Type **boot disk2 -swr**, press **Return**.

- The Sun boots up.
- s is for single user, w is for writeable and r is for reconfigure (required because you added a drive).

8 Type ***RootPassword***, press **Return**.

- ***RootPassword*** is the root password for the download disk.
- Remember that the ***RootPassword*** is case sensitive.
- You are authenticated as root and returned the UNIX prompt.

9 Type **/download/setup**, press **Return**.

- Status messages will be displayed.

10 When prompted for the Sun's name, type ***SunsName***, press **Return**.

11 When prompted for the Sun's IP address, type ***SunsIP***, press **Return**.

- The Sun's network and hostname are configured.

12 When you are returned to a UNIX prompt, type **/etc/halt**, press **Return**.

13 At the > prompt, power off the download disk.

14 Disconnect the download disk from the Sun.

15 At the > prompt, type **boot -r**, press **Return**.

- The Sun boots up.
- r is for reconfigure (required because you removed a drive).

16 At the **login:** prompt, type **root**, press **Return**.

17 Type ***RootPassword***, press **Return**.

- ***RootPassword*** is the root password for the download disk. (The Sun uses the download disk's root password until a new one is set.)
- Remember that the ***RootPassword*** is case sensitive.
- You are authenticated as root and returned the UNIX prompt.

18 Type **passwd root**, press **Return**.

19 At the **New password:** prompt, type ***RootPassword***, press **Return**.

- ***RootPassword*** is the root password for the Sun.
- Remember that the ***RootPassword*** is case sensitive.

20 At the **Re-enter new password:** prompt, type ***RootPassword***, press **Return**.

- ***RootPassword*** is the root password for the Sun.

- This step confirms that the root password has been entered correctly.
- Remember that the ***RootPassword*** is case sensitive.
- The root password for this Sun is set. Inform all authorized personnel of ***RootPassword***.

21 Type **exit**, press **Return**.

- Root is logged out of the Sun.

22 Inform the backup administrator of the new machine.

To install the Solaris 2.5.1 operating system, execute the steps provided in the following table.

Table 3.5-5 presents the **QUICK STEP** procedures required to install the Solaris 2.5.1 operating system.

Table 3.5-5. Install the Solaris 2.5.1 Operating System - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	(No entry)	**get the download disk**
2	(No entry)	**check that download disk is set to target 2**
3	(No entry)	**plug download disk into Sun**
4	(No entry)	**power on download disk**
5	(No entry)	**power on monitor**
6	(No entry)	**power on Sun**
7	probe-scsi	**press Return**
8	(No entry)	**verify that target 2 exists**
9	boot disk2 -swr	**press Return**
10	*RootPassword of download disk*	**press Return**
11	/download/setup	**press Return**
12	*SunsName*	**press Return**
14	*SunsIP*	**press Return**
15	/etc/halt	**press Return**
16	(No entry)	**power off download disk**
17	(No entry)	**disconnect download disk from Sun**
18	boot -r	**press Return**
19	root	**press Return**
20	*RootPassword of download disk*	**press Return**
21	passwd root	**press Return**
22	*RootPassword for the Sun*	**press Return**
23	*RootPassword for the Sun*	**press Return**
24	exit	**press Return**
25	(No entry)	**inform all authorized personnel of *RootPassword for the Sun***
26	(No entry)	**inform backup administrator of new Sun**

3.5.2.2.2 HP-UX 10.01 AND 10.10 Operating System Installation

The HP-UX 10.01 and 10.10 Operating System Installation process begins when procedure 3.5.2.1 Installation of Hardware has been completed in the Installing a New Workstation process. Once complete, the SA proceeds to procedure 3.5.2.3.1 Installation of Custom Software.

Detailed procedures for tasks performed by the SA are provided in the sections that follow.

The procedure assumes that procedures 3.5.1 Preparation and 3.5.2.1 Installation of Hardware have been completed. The procedure also assumes that the workstation is powered off.

This section explains how to install the HP-UX 10.01 and 10.10 operating system, including network configuration and patch installation. If you would like a listing of the patches installed, please see document number 420-TD-014-001 Release A HP Operating System Patch List attached

at the end of this document. To install the HP-UX 10.01 and 10.10 operating system, execute the procedure steps that follow:

1 Get the download disk.

2 Check that it is set to be target 2.
- The target number is found on the back of the disk.
- You can change the target number by hitting the buttons above and below it.

3 Plug the download disk into the HP.

4 Power on the download disk.
- The power switch is located on the back of the drive.

5 Power on the monitor, power on the HP.
- The power switch is located on the right side of the HP, towards the front.
- The HP starts booting up.

6 At the **Selecting a system to boot. To stop selection process press and hold the ESCAPE key** message, press and hold **Escape**.
- You have 10 seconds to press **Escape** before the boot process proceeds.
- The boot process will stop and a menu of boot commands will appear.

7 Select boot scsi.2.0 by typing **b *DeviceSelectionForscsi.2.0* isl**, press **Return**.
- For example, if the Device Selection for scsi.2.0 in the menu is P1 then type **b P1 isl** and then press **Return.**
- **isl** will cause the HP to boot in interactive mode.

8 At the **ISL>** prompt, type **hpux -is boot disk(scsi.2;0)/hp-ux**, press **Return**.
- **-is** causes the HP to boot in single user mode.
- You will be returned to the UNIX prompt.

9 Type **/download/setup**, press **Return**.
- Status messages will be displayed.

10 When prompted for the HP's name, type ***HPsName***, press **Return**.

11 When prompted for the HP's IP address, type ***HPsIP***, press **Return**.
- The HP's network and hostname are configured.

12 When you are returned to a UNIX prompt, type **/etc/shutdown -h -y now**, press **Return**.
- The HP shuts down and comes to a halt.

13 Once the HP has halted, power off the download disk, power off the monitor.

14 Power off the HP.

15 Disconnect the download disk from the HP.

16 Power on the monitor.

17 Power on the HP.

- The HP starts booting up.

18 At the **Selecting a system to boot. To stop selection process press and hold the ESCAPE key** message, press and hold **Escape**.

- You have 10 seconds to press **Escape** before the boot process proceeds.
- The boot process will stop and a menu of boot commands will appear.

19 Select boot scsi.6.0 by typing **b *DeviceSelectionForscsi.6.0***, press **Return**.

- For example, if the Device Selection for scsi.6.0 in the menu is P1 then type **b P1** and press **Return**.

20 At the **login:** prompt, type **root**, press **Return**.

21 Type ***RootPassword***, press **Return**.

- ***RootPassword*** is the root password for the download disk. (The HP uses the download disk's root password until a new one is set.)
- Remember that the ***RootPassword*** is case sensitive.
- You are authenticated as root and returned the UNIX prompt.

22 Type **passwd root**, press **Return**.

23 At the **New password:** prompt, type ***RootPassword***, press **Return**.

- ***RootPassword*** is the root password for the HP.
- Remember that the ***RootPassword*** is case sensitive.

24 At the **Re-enter new password:** prompt, type ***RootPassword***, press **Return**.

- ***RootPassword*** is the root password for the HP.
- This step confirms that the root password has been entered correctly.
- Remember that the ***RootPassword*** is case sensitive.
- The root password for this HP is set. Inform all authorized personnel of ***RootPassword***.

25 Type **exit**, press **Return**.

- Root is logged out of the HP.

26 Inform the backup administrator of the new machine.

To install the HP-UX 10.01 and 10.10 operating system, execute the steps provided in the following table.

Table 3.5-6 presents the **QUICK STEP** procedure required to install the HP-UX 10.01 and 10.10 operating system.

Table 3.5-6. Install the HP-UX 10.01 and 10.10 Operating System - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	(No entry)	**get the download disk**
2	(No entry)	**check that download disk is set to target 2**
3	(No entry)	**plug download disk into HP**
4	(No entry)	**power on download disk**
5	(No entry)	**power on monitor**
6	(No entry)	**power on HP**
7	(No entry)	**press and hold Escape**
8	b *DeviceSelectionForscsi.2.0* isl	**press Return**
9	hpux -is boot disk(scsi.2;0)/hp-ux	**press Return**
10	/download/setup	**press Return**
11	*HPsName*	**press Return**
12	*HPsIP*	**press Return**
14	/etc/shutdown -h -y now	**press Return**
15	(No entry)	**power off download disk**
16	(No entry)	**power off the monitor**
17	(No entry)	**power off the HP**
18	(No entry)	**disconnect download disk from HP**
19	(No entry)	**power on the monitor**
20	(No entry)	**power on the HP**
21	(No entry)	**press and hold Escape**
22	b *DeviceSelectionForscsi.6.0*	**press Return**
23	root	**press Return**
24	*RootPassword of download disk*	**press Return**
25	passwd root	**press Return**
26	*RootPassword for the HP*	**press Return**
27	*RootPassword for the HP*	**press Return**
28	exit	**press Return**
29	(No entry)	**inform all authorized personnel of *RootPassword for the HP***
30	(No entry)	**inform backup administrator of new HP**

3.5.2.2.3 IRIX 6.2 Operating Systems Installation

The IRIX 6.2 Operating Systems Installation process begins when procedure 3.5.2.1 Installation of Hardware has been completed in the Installing a New Workstation process. Once complete, the SA proceeds to procedure 3.5.2.3.1 Installation of Custom Software.

Detailed procedures for tasks performed by the SA are provided in the sections that follow.

The procedure assumes that procedures 3.5.1 Preparation and 3.5.2.1 Installation of Hardware have been completed. The procedure also assumes that the workstation is powered off.

Table 3.5-7 presents the steps required to install the IRIX 6.2 operating systems in a condensed manner. If you are already familiar with the procedure, you may prefer to use the quick-step table. If you are new to the system, or have not performed this task recently, you should use the detailed procedure presented below.

This section explains how to install the IRIX 6.2 operating systems, including network configuration and patch installation.

To install the IRIX 6.2 operating system, execute the procedure steps that follow:

1 Get the download disk.

2 Check that it is set to be target 2.
- The target number is found on the bottom of the disk.
- You can change the target number by hitting the buttons above and below it.

3 Plug the download disk into the SGI, if available; otherwise use CD-ROM skip to Step 5

4 Power on the download disk.
- The power switch is located on the back of the drive.

5 Power on the monitor; power on the SGI.
- The power switch is located on the front of the SGI, towards the left.

6 At the **Starting up the system**... message, click the **Stop for Maintenance** button.
- You have only a few seconds to click the **Stop for Maintenance** button before the boot process proceeds.
- The boot process will stop and a **System Maintenance** menu will appear.

7 Select **5 Enter Command Monitor**.
- You will be returned to the Command Monitor prompt which is >>.

8 At the >> prompt, type **hinv**, press **Return**.
- Verify that target 2 exists by finding it in the listing that appears. It will appear as **SCSI Disk: scsi(0)disk(2)**.

9 Type **boot -f dksc(0,2,0)sash**, press **Return**.
- The SGI boots from the download disk into the stand alone shell.
- You will be returned to a UNIX prompt.

10 Type **/download/setup**, press **Return**.
- Status messages will be displayed.

11 When prompted for the SGI's name, type ***SGIsName***, press **Return**.

12 When prompted for the SGI's IP address, type ***SGIsIP***, press **Return**.

- The SGI's network and hostname are configured.

13 When you are returned to a UNIX prompt, type **/etc/shutdown -y -g0**, press **Return**.

- The SGI shuts down.
- You will be returned to a >> prompt, a **System Maintenance** menu or a message saying that **this system can be powered off**.

14 Power off the download disk, power off the monitor.

15 Power off the SGI.

16 Disconnect the download disk from the SGI.

17 Power on the monitor.

18 Power on the SGI.

- The SGI starts booting up.

19 At the **login:** prompt, type **root**, press **Return**.

20 Type ***RootPassword***, press **Return**.

- ***RootPassword*** is the root password for the download disk. (The SGI uses the download disk's root password until a new one is set.)
- Remember that the ***RootPassword*** is case sensitive.
- You are authenticated as root and returned the UNIX prompt.

21 Type **passwd root**, press **Return**.

22 At the **New password:** prompt, type ***RootPassword***, press **Return**.

- ***RootPassword*** is the root password for the SGI.
- Remember that the ***RootPassword*** is case sensitive.

23 At the **Re-enter new password:** prompt, type ***RootPassword***, press **Return**.

- ***RootPassword*** is the root password for the SGI.
- This step confirms that the root password has been entered correctly.
- Remember that the ***RootPassword*** is case sensitive.
- The root password for this SGI is set. Inform all authorized personnel of ***RootPassword***.

24 Type **exit**, press **Return**.

- Root is logged out of the SGI.

25 Inform the backup administrator of the new machine.

To install the IRIX 6.2 operating system, execute the steps provided in the following table.

Table 3.5-7. Install the IRIX 6.2 Operating System - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	(No entry)	**get the download disk**
2	(No entry)	**check that download disk is set to target 2**
3	(No entry)	**plug download disk into SGI**
4	(No entry)	**power on download disk**
5	(No entry)	**power on monitor**
6	(No entry)	**power on SGI**
7	(No entry)	**click the Stop for Maintenance button**
8	5 Enter Command Monitor	**(No action)**
9	hinv	**press Return**
10	(No entry)	**verify that SCSI Disk: scsi(0)disk(2) appears in the listing**
11	boot -f dksc(0,2,0)sash	**press Return**
12	/download/setup	**press Return**
13	*SGIsName*	**press Return**
14	*SGIsIP*	**press Return**
15	/etc/shutdown -y -g0	**press Return**
16	(No entry)	**power off download disk**
17	(No entry)	**power off monitor**
18	(No entry)	**power off SGI**
19	(No entry)	**disconnect download disk from SGI**
20	(No entry)	**power on monitor**
21	(No entry)	**power on SGI**
22	root	**press Return**
23	*RootPassword of download disk*	**press Return**
24	passwd root	**press Return**
25	*RootPassword for the SGI*	**press Return**
26	*RootPassword for the SGI*	**press Return**
27	exit	**press Return**
28	(No entry)	**inform all authorized personnel of *RootPassword for the SGI***
29	(No entry)	**inform backup administrator of new SGI**

3.5.2.2.4 NCD Operating System Installation

The NCD Operating System Installation process begins when procedure 3.5.2.1 Installation of Hardware has been completed in the Installing a New Workstation process. Once complete, the SA proceeds to procedure 3.5.3 Testing and Verification.

Detailed procedures for tasks performed by the SA are provided in the sections that follow.

The procedure assumes that procedures 3.5.1 Preparation and 3.5.2.1 Installation of Hardware have been completed. The procedure also assumes that the workstation is powered off.

Installing the NCD operating system consists of configuring the NCD.

Table 3.5-8 presents the steps required to configure the NCD in a condensed manner. If you are already familiar with the procedure, you may prefer to use the quick-step table. If you are new to the system, or have not performed this task recently, you should use the detailed procedure presented below.

This section explains how to configure the NCD, including putting the necessary start-up files in place on the server.

To configure the NCD, execute the procedure steps that follow:

1 Turn on the NCD and monitor. The monitor power button is on the lower front of the monitor. The NCD power switch is on the back, on the right.

- The message **Boot Monitor V*x.x.x*** will appear.

2 Press the **Escape** key twice.

- You have only a few seconds to press the **Escape** key.
- The boot process stops and a boot monitor prompt, >, appears.
- If you do not see a > prompt then press the **Escape** key a few more times.

3 Press the **Setup** key.

- The **Main** menu will appear.

4 Go to the **Keyboard** menu by pressing the **Right Arrow** key.

- The **Keyboard** menu appears.

5 Select **N-101** by pressing the **Down Arrow** key.

- You may need to press the **Down Arrow** key a few times before **N-101** is selected.

6 Go to the **Monitor** menu by pressing the **Right Arrow** key.

- The **Keyboard** menu disappears.
- The **Monitor Resolution** menu appears.

7 Select **1600x1200 65 Hz** by pressing the **Down Arrow** key.

- ↑ You may need to press the **Down Arrow** key a few times before **1600x1200 65 Hz** is selected.

8 Press the **Shift** and **T** keys.

- This tests the new monitor resolution setting.

9 Use the **+** and **-** keys on the front of the monitor under the **ADJUST** label to adjust the screen.

10 Press the **STORE** key on the front of the monitor.

- The monitor stores the screen adjustments.

11 Press the **Escape** key.

- The monitor resolution test ends.
- You are returned to the **Main** menu.

12 Go to the **Network** menu by pressing the **Right Arrow** key twice.

- The **Monitor Resolution** menu disappears.
- The **Network** menu appears.

13 Select **NVRAM** for the **Get IP Addresses From** option.

- You can use the **Space Bar** to move between the available options.

14 Press the **Down Arrow** key.

15 Type the ***NCDIPaddress*** for the **Terminal IP Address** option, press the **Down Arrow** key.

- The ***NCDIPaddress*** is in dotted decimal notation, for example, 155.157.21.34.

16 Type the ***StartupFileServerIPaddress*** for the **First Boot Host IP Address** option, press the **Down Arrow** key.

- ↑ The ***StartupFileServerIPaddress*** is in dotted decimal notation, for example, 155.157.44.22.
- The ***StartupFileServer*** is the machine where the NCD startup files are stored.

17 Press the **Down Arrow** key twice.

18 Type the ***NCDGatewayIPaddress*** for the **Gateway IP address** option, press the **Down Arrow** key.

- ↑ The ***NCDGatewayIPaddress*** is in dotted decimal notation, for example, 155.157.44.22.
- ↑ The ***NCDGatewayIPaddress*** is the same as the ***NCDIPaddress*** except the last number/octet is 1. For example, if the ***NCDIPaddress*** is 155.157.21.34, the ***NCDGatewayIPaddress*** is 155.157.21.1.

19 Press the **Down Arrow** key, type the ***BroadcastIPaddress*** for the **Broadcast IP Address** option.

- ↑ The ***BroadcastIPaddress*** is in dotted decimal notation, for example, 155.157.44.22.
- ↑ The ***BroadcastIPaddress*** is the same as the ***NCDIPaddress*** except the last number/octet is 255. For example, if the ***NCDIPaddress*** is 155.157.21.34, the ***BroadcastIPaddress*** is 155.157.21.255.

20 Press the **Right Arrow** key.

- The **Network** menu disappears.
- The **Boot** menu appears.

21 Type **Xncdhmx_s** for the **Boot File** option, press the **Down Arrow** key.

22 Press the **Down Arrow** key, type **/data/ncd/** for the **NFS Boot Directory** option, press the **Down Arrow** key.

23 Press the **Down Arrow** key, type **/usr/lib/X11/ncd/configs/** for the **UNIX Config Directory** option, press the **Down Arrow** key.

24 Press the **Down Arrow** key, press the **d** key.

- The **TFTP Order** option is set to **Disabled**.

25 Press the **Down Arrow** key, press the **1** key.

- The **NFS Order** option is set to **1**.

26 Press the **Down Arrow** key, press the **d** key.

- The **MOP Order** option is set to **Disabled**.

27 Press the **Down Arrow** key, press the **d** key.

- The **LOCAL Order** option is set to **Disabled**.

28 Press the **Right Arrow** key.

- The **Boot** menu disappears.
- The **Done** menu appears.
- **Reboot** is selected.

29 Press the **Return** key.

- The NCD reboots.
- Status messages appear.

30 Log into the ***StartupFileServer*** by typing: **telnet** ***StartupFileServer*** or **rsh** ***StartupFileServer*** at a UNIX prompt, then press **Return**.

31 If a **Login:** prompt appears, log in as yourself by typing: ***YourUserID***, then press **Return**.

- A password prompt is displayed.

32 Enter ***YourPassword***, then press **Return.**

- Remember that ***YourPassword*** is case sensitive.
- You are authenticated as yourself and returned to the UNIX prompt.

33 Log in as root by typing: **su**, then press **Return**.

- A password prompt is displayed.

34 Enter the ***RootPassword,*** then press **Return**.

- Remember that the ***RootPassword*** is case sensitive.
- You are authenticated as root and returned to the UNIX prompt.

35 Type **cd /usr/lib/X11/ncd/configs**, press **Return**.

36 Type **./i**, press **Return**.

- **i** is a script which builds a NCD startup file.

37 Type the last two numbers/octets of the ***NCDIPaddress*** when the script prompts you for the **IP address**, press **Return**.

- ↑ For example, if the ***NCDIPaddress*** is 155.157.21.34 then type **21.34** and then press **Return**.

38 Type the ***NCDLoginHost*** when the script prompts you for the **Login Host**, press **Return**.

- The ***NCDLoginHost*** is the name of one of the X-servers.

39 When the script prompts you for the **NCD Number**, type the ***NCDname*** minus the ncd part.

- For example, if the ***NCDname*** is ncd2 then the **NCD Number** is 2.
- Some status messages appear telling you what the script is doing.
- The script exits.

40 Type **exit**, then press **Return**.

- **Root** is logged out

41 Type **exit** again, then press **Return**.

- You are logged out and disconnected from the ***StartupFileServer*.**

To configure the NCD, execute the steps provided in the following table.

Table 3.5-8. Configure the NCD - Quick-Step Procedures (1 of 2)

Step	What to Enter or Select	Action to Take
1	(No entry)	power on the monitor and NCD
2	(No entry)	press Escape twice
3	(No entry)	press Setup key
4	(No entry)	press Right Arrow key
5	N-101	press Right Arrow key
6	1600x1200 65 Hz	press Shift and T keys
7	(No entry)	use + and - keys under ADJUST on front of monitor to adjust the screen
8	(No entry)	press the STORE key on the front of the monitor
9	(No entry)	press Escape
10	(No entry)	press Right Arrow key twice
11	NVRAM	press Down Arrow key

Table 3.5-8. Configure the NCD - Quick-Step Procedures (2 of 2)

12	*NCDIPaddress*	**press Down Arrow key**
13	*StartupFileServerIPaddress*	**press Down Arrow key three times**
14	*NCDGatewayIPaddress*	**press Down Arrow key twice**
15	*BroadcastIPaddress*	**press Right Arrow key**
16	Xncdhmx_s	**press Down Arrow key twice**
17	/data/ncd/	**press Down Arrow key twice**
18	/usr/lib/X11/ncd/configs/	**press Down Arrow key twice**
19	(No entry)	**press d key**
20	(No entry)	**press Down Arrow key**
21	(No entry)	**press 1 key**
22	(No entry)	**press Down Arrow key**
23	(No entry)	**press d key**
24	(No entry)	**press Down Arrow key**
25	(No entry)	**press d key**
26	(No entry)	**press Right Arrow key**
27	(No entry)	**press Return**
28	telnet *StartupFileServer* -or- rsh *StartupFileServer*	**press Return**
29	*YourUserID*	**press Return**
30	*YourPassword*	**press Return**
31	su	**press Return**
32	*RootPassword*	**press Return**
33	cd /usr/lib/X11/ncd/configs	**press Return**
34	./i	**press Return**
35	last two numbers/octets of *NCDIPaddress*	**press Return**
36	*NCDLoginHost*	**press Return**
37	*NCDname* minus the ncd part	**press Return**
38	exit	**press Return**
39	exit	**press Return**

3.5.2.3 Software

3.5.2.3.1 Custom

The Installation of Custom Software process begins when procedure 3.5.2.2 Operating System Installation has been completed in the Installing a New Workstation process. Once complete, the SA proceeds to procedure 3.5.2.3.2 Installation of COTS Software.

Detailed procedures for tasks performed by the SA are provided below. The procedure assumes that procedures 3.5.1 Preparation, 3.5.2.1 Installation of Hardware and 3.5.2.2 Operating System Installation have been completed.

Table 3.5-9 presents the steps required to install custom software in a condensed manner. If you are already familiar with the procedure, you may prefer to use the quick-step table. If you are new to the system, or have not performed this task recently, you should use the detailed procedure presented below.

To install custom software for the requester, execute the procedure step that follows:

1 execute procedures in Section 22.2.3 Custom Software Installation in this document.

To install custom software, execute the steps provided in the following table.

Table 3.5-9. Install Custom Software - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	(No entry)	**execute procedures in Section 22.2.3 of this document**

3.5.2.3.2 COTS

The COTS Software Installation process begins after the SA has completed Section 3.5.2.3.1 Custom Software Installation. After the COTS software installation is complete, the SA proceeds to procedure 3.5.3 Testing and Verification.

Detailed procedures for tasks performed by the SA are provided below. The procedure assumes that procedures 3.5.1 Preparation, 3.5.2.1 Installation of Hardware, 3.5.2.2 Operating System Installation and 3.5.2.3.1 Custom Software Installation have been completed.

Table 3.5-10 contains a table which presents the steps required to install COTS software in a condensed manner. If you are already familiar with the procedure, you may prefer to use the quick-step table. If you are new to the system, or have not performed this task recently, you should use the detailed procedure presented below.

To install COTS software for the requester, execute the procedure steps that follow:

1 Refer to the Release A Hardware and Software Mapping Baseline (attached at the end of this document) for your site.

- For LaRC, refer to document number 420-TD-007-001.
- For SMC, refer to document number 420-TD-008-001.
- For GSFC, refer to document number 420-TD-006-001.

2 In the Release A Hardware and Software Mapping Baseline for your site, look up which COTS packages need to be installed on the new workstation using the **Subsystem** and hardware type (the **Target Operating System** column in the document) of the new machine.

To install COTS software, execute the steps provided in the following table.

Table 3.5-10. Install COTS Software - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	(No entry)	**refer to the Release A Hardware and Software Mapping Baseline for your site**
2	(No entry)	**look up which COTS packages to install using the Subsystem and Target Operating System of the new workstation**

3.5.3 Testing and Verification

3.5.3.1 Reboot

The Reboot process begins when procedure 3.5.2.3.2 COTS - By Package has been completed in the Installing a New Workstation process. Once complete, the SA proceeds to procedure 3.5.3.2 Logging In.

Detailed procedures for tasks performed by the SA are provided in the sections that follow.

The procedures assume that procedures 3.5.1 Preparation and 3.5.2 Installation have been completed.

Table 3.5-11 presents the steps required to reboot in a condensed manner. If you are already familiar with the procedure, you may prefer to use the quick-step table. If you are new to the system, or have not performed this task recently, you should use the detailed procedure presented below.

3.5.3.1.1 SGI, HP and Sun

To reboot, execute the procedure steps that follow:

1 At the UNIX prompt for the workstation, type **su**, press **Return**.

2 At the **Password** prompt, type ***RootPassword***, press **Return**.
- Remember that ***RootPassword*** is case sensitive.
- You are authenticated as root.

3 Type **who**, press **Return**.
- A list of users currently logged into the workstation appears.

4 If users other than root and you are logged in:
type **wall**,
press **Return**,
type **The system is going down in 5 minutes for *Reason*. Please save your work and log off. We apologize for the inconvenience.**,
press **Return**,
press **Control-D**,

wait 5 minutes before proceeding to step 5.

5 Type **/etc/reboot**, then press **Return**.

- The workstation reboots.
- Watch the status messages that appear for any errors.
- ↑ If you are returned to a **Login** prompt and saw no errors during the reboot, the reboot was successful.
- ↑ If the reboot was unsuccessful, use the error messages and system logs to figure out what is incorrect in the workstation installation. The system logs are: /var/adm/messages for Solaris 2.5.1/5.4, /var/adm/SYSLOG for IRIX 5.3 and 6.2, and /usr/adm/syslog and rc.log for HP-UX 10.01 and 10.10.

To reboot, execute the steps provided in the following table.

Table 3.5-11. Reboot - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	su	**press Return**
2	*RootPassword*	**press Return**
3	who	**press Return**
4	wall	**press Return**
5	The system is going down in 5 minutes for *Reason*. Please save your work and log off. We apologize for the inconvenience.	**press Return**
6	(No entry)	**press Control-D**
7	(No entry)	**wait 5 minutes**
8	/etc/reboot	**press Return**
9	(No entry)	**watch for errors in the boot messages**

3.5.3.1.2 NCD

The Reboot the NCD process begins when procedure 3.5.2.3.2 COTS - By Package has been completed in the Installing a New Workstation process. Once complete, the SA proceeds to procedure 3.5.3.2 Logging In.

Detailed procedures for tasks performed by the SA are provided in the sections that follow.

The procedures assume that procedures 3.5.1 Preparation and 3.5.2 Installation have been completed.

To reboot the NCD, execute the procedure steps that follow:

1 Press the **Setup** key.

- The **NCD User Services: Console** window will appear.

2 Go to the **Console** menu, select **Reboot**.

- The **Reboot** window opens asking if it is **OK to reboot the terminal**.

3 Click the **OK** button.

- The NCD reboots.
- Watch the status messages that appear.
- Once the NCD successfully reboots, a login screen appears.
- ↑ If the NCD does not successfully reboot then use the information in the status messages to determine what went wrong in procedure 3.5.2.2.4 NCD Operating System Installation.

Table 3.5-12 presents the **QUICK STEP** procedure required to reboot the NCD.

Table 3.5-12. Reboot the NCD - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	(No entry)	**press Setup key**
2	Console → Reboot	**click OK button**
3	(No entry)	**watch for errors in the boot messages**

3.5.3.2 Logging In

The Logging In process begins when procedure 3.5.3.1 Reboot has been completed in the Installing a New Workstation process. Once complete, the SA proceeds to procedure 3.5.3.3 Test Environment.

Detailed procedures for tasks performed by the SA are provided in the sections that follow.

The procedures assume that procedures 3.5.1 Preparation, 3.5.2 Installation and 3.5.3.1 Reboot have been completed. The procedures also assume that the workstation is currently at a **Login** prompt.

Table 3.5-13 presents the steps required to log in in a condensed manner. If you are already familiar with the procedure, you may prefer to use the quick-step table. If you are new to the system, or have not performed this task recently, you should use the detailed procedure presented below.

To log in, execute the procedure steps that follow:

1 At the **Login** prompt for the workstation, type ***YourUserID***, press **Return**.

2 At the **Password** prompt, type ***YourPassword***, press **Return**.

- Remember that ***YourPassword*** is case sensitive.
- You are logged in and authenticated as yourself.
- You are returned to a UNIX prompt.

- ↑ If you are not logged in and returned to a UNIX prompt, logging in was unsuccessful. Follow these steps:

  a Execute this procedure one more time.
  If logging in is unsuccessful again, there is a problem with the workstation installation. Continue to step b.

  b Type **root** at the **Login** prompt, press **Return**.

  c Type ***RootPassword*** at the **Password** prompt, press **Return**.
  Remember that ***RootPassword*** is case sensitive.
  You are authenticated as root and returned to a UNIX prompt.

  d Check that automount is running by typing **ps -ef | grep auto** or **ps -aux | grep auto**, press **Return**.
  If automount is running then you will see output similar to this:

  ```
  yourID   10173  0.2  0.4  648  408 pts/38   S 15:35:51  0:00 grep auto
  root      140     0.0  0.8  1796 1004 ?       S  Jun 25    2:40 /usr/lib/autofs/
                                                  automountd -D ARCH=sun5
  ```

  If automount is not running then run it by typing:
  /usr/lib/autofs/automountd -D ARCH=sun5 for Solaris 2.5.1/5.4,
  /usr/etc/automount -D ARCH=sgi for IRIX 5.3 and 6.2,
  /usr/etc/automount -D ARCH=hp for HP-UX 10.01 and 10.10,
  press **Return**.
  Try logging in again by typing **su -** ***YourUserID***, press **Return**, type***YourPassword***, press **Return**. Type **whoami** and press **Return** to confirm that you successfully logged in as yourself and type **cd**, press **Return**, type **pwd**, press **Return** to confirm that you are in your home directory. If these commands return ***YourUserID*** and your home directory, you have successfully logged in. If you have not successfully logged in, proceed to step e.

  e The workstation probably did not successfully bind to a NIS server. Verify that
  the NIS server is up and on the network. Once it is, execute procedure 3.5.3.1
  Reboot and then execute this procedure again.

To log in, execute the steps provided in the following table.

Table 3.5-13. Log In - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	*YourUserID*	press Return
2	*YourPassword*	press Return

3.5.3.3 Test Environment

The Test Environment process begins when procedure 3.5.3.2 Logging In has been completed in the Installing a New Workstation process. Once the test environment procedure is complete, the Installing a New Workstation process is complete and the SA notifies the requester, the supervisor and the DAAC Manager.

Table 3.5-14. Test Environment - Activity Checklist

Order	Role	Task	Section	Complete ?
1	SA	Test Environment.	(P) 3.5.3.3.1	
2	SA	Inform Requester, Supervisor and DAAC Manager of completion.	(I) 3.5.3.3.1	

Detailed procedures for tasks performed by the SA are provided in the sections that follow.

The procedures assume that procedures 3.5.1 Preparation, 3.5.2 Installation, 3.5.3.1 Reboot and 3.5.3.2 Logging In have been completed.

Table 3.5-15 presents the steps required to test the environment in a condensed manner. If you are already familiar with the procedure, you may prefer to use the quick-step table. If you are new to the system, or have not performed this task recently, you should use the detailed procedure presented below.

To test the environment, execute the procedure steps that follow:

1 At the UNIX prompt, type **ps -ef | more** or **ps -aux | more**.

- A screen full of information about the currently running processes is displayed.

2 Look for the processes associated with the custom and COTS software which you installed in the process listing.

- To move to the next page full of information, press the **Space** bar.
- If a process is missing in the listing, go back to the installation of that software package to determine what went wrong.

3 Type **cd** ~/***YourUserID***, press **Return**.

4 Type **pwd**, press **Return**, use the output to verify that you are in your home directory.

- ↑ This verifies that automount is running and working correctly for the NIS map auto.home. You may follow steps similar to steps 3 and 4 for the other NIS maps.
- This also verifies that the new workstation was able to contact a NIS server.

To test the environment, execute the steps provided in the following table.

Table 3.5-15. Test Environment - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	**ps -ef -or- ps -aux**	**press Return**
2	(No entry)	**find processes associated with the installed custom and COTS software packages in the listing**
3	(No entry)	**press Space bar for the next page of process information**
4	**cd ~/*YourUserID***	**press Return**
5	**pwd**	**press Return**
6	(No entry)	**use the output to verify that you are in your home directory**

3.6 DCE Configuration

3.6.1 Configuring Initial Cell

The Configuring the Initial Cell consists of configuring the Master Security Server, initial CDS Server and DTS Servers (Time & Time Provider servers). This section describes how to configure the Master Security server and initial CDS server. Section 3.6.2 explains setting up the DTS server(s).

Detailed procedures for tasks performed by the SA are provided in the sections that follow.

The procedures assume that all planning involved has been concluded well in advance and the SA has been properly trained to perform the configuration.

Table 3.6-1 presents the steps required to configure the initial cell in a condensed manner. If you are already familiar with the procedure, you may prefer to use the quick-step table. If you are new to the system, or have not performed this task recently, you should use the detailed procedure presented below.

This section explains how to configure the initial cell. To begin configuring the initial cell, execute the procedures steps that follow:

NOTE: When planning a DCE cell, note that you must configure a CDS client on any Security server system that is not running a CDS server. You must also configure a Time client on any system that is not running a Time server. Be sure to configure these clients only after you have configured all servers.

1 Log in as root on the system you wish to configure

2 Make sure that /etc is in your command search path:

#export PATH=/etc:$PATH (Bourne/Korn shell)

% setenv PATH /etc:$PATH (C shell)

3 Type in the following appropriate command:

The sun command is dcesetup config client

The IBM command is mkdce -a cell_admin

-s edf-bb.gsfc.nasa.gov -c edf-bb.gsfc.nasa.gov cds_cl

The HP OSF DCE command is dce_config

Example:

<baltic /home/reginald>dce_config (HP OSF DCE command)

4 From the DCE Main Menu, select configure and start DCE deamons (selection 1).

5 From the DCE Configuration Menu, select Initial Cell Configuration (selection 1).

6 From the Initial Cell Configuration Menu, select Security Server (selection 1).

7 If this is your very first cell configuration, or if you have previously run REMOVE, answer **n** to the following question. If you are re-configuring a cell, answer **y**.

Do you wish to first remove all remnants of previous DCE configurations for all components (y/n)?

You should do so only if you plan on re-configuring all existing DCE components now: (n)

8 Enter the name of your cell (without /.../).
dce_config will prompt you with a warning. If rpcd was recently running with the TCP protocol sequence, then wait until 4 minutes have elapsed since rpcd was stopped before continuing from this prompt:

9 Enter keyseed for initial database master key:

1 0 dce_config prompts you to choose the Cell Administrators' principal name and password. The default principle name for the Cell Administrator is cell_admin:

1 1 dce_config prompts you for the starting point for UNIX user and group ID's that will be generated by the DCE Security Service. This step prevents the DCE Security Service from generating IDs that are already in use by your system. Type <RETURN> to choose the default value, or enter a value of our choice:

1 2 Enter the starting point to be used for UNIX ID'S that are automatically generated by the Security Service when a principal is added using "rgy_edit": (N +100) <RETURN>

This system is now configured as the Master Security server. You must now create a CDS server, either on this system or on another system.

If the CDS server for this cell will be on another system, repeat the steps 1-4 above on that system, then continue with Step 13.

If the CDS server is on the same system as the Security server, continue with Step 14 below.

13 From the DCE Configuration Menu, select Initial Cell Configuration (selection 1)

14 From the Initial Cell Configuration menu, choose Initial CDS Server (selection 2)
This routine starts up cdsadv and cdsd, initializes the namespace, and sets ACLs for all new namespace entries.

15 dce_config asks if your cell resides on multiple LAN's. If your cell does reside on multiple LAN's, dce_config asks for the name of the LAN. The name you provide is arbitrary, and is used by dce_config to store cell profile information.

Table 3.6-1. Configuring Initial Cell - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	(No entry)	**Login as root**
2	(No entry)	**Make sure that /etc is in your command search path**
3	(No entry)	**type appropriate setup command**
4	(No entry)	**select 1 from DCE Main Menu**
5	(No entry)	**select 1 from the DCE Configuration Menu**
6	(No entry)	**select 1 from the Cell Configuration Menu**
7	(No entry)	**If reconfiguring, answer yes.** **If not reconfiguring, answer no.**
8	(No entry)	**Enter cell name**
9	(No entry)	**Enter keyseed.**
10	(No entry)	**Enter principal name and password.**
11	(No entry)	**Press <return> for default or enter a value.**
12	(No entry)	**Press <return> or enter value**
13	(No entry)	**select 1 from DCE Configuration Menu**
14	(No entry)	**select 2 from Initial Cell Configuration Menu**
15	(No entry)	**Enter arbitrary LAN name.**

3.6.2 Configuring DTS Servers

The Configuring the DTS Servers process begins after the Master CDS server has been configured. Refer to section 3.6.1 Configuring Initial Cell for Master CDS Server Configuration procedures before continuing with this section.

Detailed procedures for tasks performed by the SA are provided in the sections that follow.

The procedures assume that all planning involved has been concluded well in advance and the SA has been properly trained to perform the configuration.

Table 3.6-2 presents the steps required to configure DTS servers in a condensed manner. If you are already familiar with the procedure, you may prefer to use the quick-step table. If you are new to the system, or have not performed this task recently, you should use the detailed procedure presented below.

This section explains how to configure the DTS daemon. To begin configuring the DTS daemon, execute the procedures steps that follow:

NOTE: To configure a DTS server on a system not already configured as a Security or Directory server, repeat steps 1-4 of Section 3.6.1.1 (Configuring the Initial Cell) on that system, and then continue with the steps below. To configure a DTS server on a system already configured as a Security or Directory server, continue with step 1 below.

1 From the DCE Configuration menu, select Additional Server Configuration (selection 2).

2 From the Additional Server Configuration menu, select DTS (selection 2).

3 From the DTS Configuration menu, select DTS Local Server for servers on the LAN (selection 1). Otherwise, select DTS global server (selection 2). Either selection starts the dts daemon (dtsd) and dtstimed.

4 Configure a DTS time provider on one of the time servers in a cell. Select the DTS Time Provider (selection 4).

The DTS null time provider configures a system to trust its own clock as an accurate source of time. The DTS ntp provider obtains an accurate source of time from some other system outside the cell. See the OSF DCE Administration Guide for more information on time providers.

5 From the DTS Time Provider Menu, select Null Time Provider (selection 1) or NTP Time Provider (selection 2).

If you select the NTP time provider the following prompt appears: Enter the host name where the NTP server is running:

6 Enter the host name.

Table 3.6-2. Configuring DTS Servers - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	(No entry)	**select 2 from DCE Configuration Menu**
2	(No entry)	**select 2 from Additional Server Configuration Menu**
3	(No entry)	**select 1 for servers on the same LAN or select 2 from the DTS Configuration Menu**
4	(No entry)	**select 4 to configure DTS time provider**
5	(No entry)	**select 1 or 2 from DTS Time Provider Menu**
6	(No entry)	**enter the host name where the NTP server is running**

3.6.3 Configuring Additional CDS Servers

The Configuring Additional CDS Servers process begins after the DTS servers have been configured. Refer to section 3.6.2 procedures before continuing with this section.

Detailed procedures for tasks performed by the SA are provided in the sections that follow.

The procedures assume that all planning involved has been concluded well in advance and the SA has been properly trained to perform the configuration.

Table 3.6-3 presents the steps required to configure additional CDS servers in a condensed manner. If you are already familiar with the procedure, you may prefer to use the quick-step table. If you are new to the system, or have not performed this task recently, you should use the detailed procedure presented below.

This section explains how to configure additional CDS servers. To begin configuring the additional CDS servers, execute the procedures steps that follow:

1 From the DCE Configuration menu, select Additional Server Configuration (selection 2).

2 Enter the name of your cell.

3 Enter the host name of your cell's security server**.**

Make sure the contents of the pe_site file is identical on both the server and the client. The dce_config script checks this, but prompts for your confirmation. Identical pe_site files are normally generated automatically, but you should confirm this yourself, particularly during your initial set-up. If the pe_site files are not identical, you should start this procedure again.

Ensure the /opt/dcelocal/etc/security/pe_site file matches that on the server...

4 Press <RETURN> to continue, CTRL-C to exit: <RETURN>

5 Enter the cell administrator's principal name and password.

6 From the Additional Server Configuration menu, select Additional CDS Server(s) (selection 1).

7 Enter the name of the cell CDS server. If the cell has more than one CDS server, choose one.

8 dce_config asks if more directories should be replicated. If you answer **yes**, continue to step 9.

9 Enter a list of directories to be replicated, separated by spaces and terminated by <RETURN>.

Notes on Configuring Additional CDS Servers

Immediately after configuring an additional CDS server, you should skulk the root directory using the set directory /.: to skulk command as cell_admin in cdscp. This will initiate the propagation of

a consistent copy of the changed root directory information to all the CDS servers, and will prevent problems which might arise from use of inconsistent information before this propagation. The use of several CDS servers may increase the time required to complete the propagation of this information.

Configuration of additional CDS servers can occasionally fail if namespace information is not correctly propagated. Typical failures observe from this cause are:

ERROR: Error during creation of clearinghouse /.:/nodename_ch.

Message from cdscp:

Failure in routine: cp_create_clh; code = 282109010

Requested operation would result in lost connectivity to root directory (dce / cds)

ERROR: Error during creation of clearinghouse / .:/nodename_ch.

Message from cdscp:

Failure in routine: cp_create_clh; code = 282108908

Unable to communicate with any CDS server (dce / cds)

If this happens, the server daemon cdsd has been successfully launched, but its clearinghouse has not bee properly created. the clearinghouse is in an intermediate state and cannot be used or deleted, although the rest of the cell namespace an other servers are unaffected. To recover, skulk the root directory, and then use the create clearinghouse /.:/nodename_ch command as cell_admin in cdscp on the new CDS server node to manually complete the configuration of the new server and its clearinghouse. Then skulk the root directory again.

In rare circumstances, you may see the following error when configuring a CDS client or additional server:

ERROR: cdscp error during "define cached server" command.

Message from cdscp:

Failure in routine: cp_define_cached_server; code = 282111142

Cached Server clearinghouse already exists (dce / cds)

This error is benign and results from the system trying to repeat an operation that has already been done. This error may be ignored.

Table 3.6-3. Configuring Additional CDS Servers - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	(No entry)	select 2 from the DCE Configuration menu
2	(No entry)	Enter the name of your cell
3	(No entry)	Enter the host name of your cell's security server
4	(No entry)	press <return> to continue or CTRL-C to exit
5	(No entry)	Enter the cell administrator's principal and password
6	(No entry)	select 1 from the Additional Server Configuration menu
7	(No entry)	Enter the name of the cell CDS server
8	(No entry)	answer yes and continue to step 9 or answer no to end process
9	(No entry)	Enter a list of directories to be replicated

3.6.4 Configuring Security and CDS Client Systems

Before configuring clients, configure server systems as described in the Initial Cell Configuration, Configuring DTS Servers, and Configuring Additional CDS Servers sections. Then use this procedure to configure client systems.

Detailed procedures for tasks performed by the SA are provided in the sections that follow.

The procedures assume that all planning involved has been concluded well in advance and the SA has been properly trained to perform the configuration.

Table 3.6-4 presents the steps required to configure security and CDS client systems in a condensed manner. If you are already familiar with the procedure, you may prefer to use the quick-step table. If you are new to the system, or have not performed this task recently, you should use the detailed procedure presented below.

This section explains how to configure security and CDS client systems. To begin configuring the security and CDS client systems, execute the procedures steps that follow:

You must configure a CDS client on any Security server system that is not running a CDS server. To configure a client system, you need to know the name of the Security server and the initial CDS server for the cell.

Note that this procedure does not create a DTS clerk (client). This is described in section 3.6.5.

1 Start dce_config on the system that you wish to configure with DCE client(s).

2 From the DCE Main menu, select Configure (selection 1).

3 From the DCE Configuration menu, select DCE Client (selection 3).

4 Enter the name of your cell.

5 Enter the name of the security server for the cell. Then press <RETURN> to continue, CTRL-C to exit: <RETURN>

6 Enter the cell administrator's principal name and password.

7 Enter the name of a CDS server in this cell. If there is more than one, first enter the name of the server to be cached, if necessary. Then continue with the next server(s).

You will be asked whether or not this node is to be configured as a DFS client. Answer **no**.

8 You will now be asked to create a LAN profile so clients and servers can be divided into profile groups for higher performance in a multi-lan cell. Answer **no**.

9 You will now be asked if this machine should be configured as a DTS Clerk, DTS Local Server or DTS Global Server (the default is DTS Clerk). Type **local**.

Table 3.6-4. Configuring Security and CDS Client Systems - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	(No entry)	start dce_config
2	(No entry)	select 1 from DCE Main menu
3	(No entry)	select 3 from DCE Configuration menu
4	(No entry)	Enter the name of your cell
5	(No entry)	Enter the name of the security server for the cell
6	(No entry)	Enter the cell administrator's principal name and password
7	(No entry)	Enter the name of a CDS server in this cell
8	(No entry)	answer no for the node to be configured as a DFS client
9	(No entry)	type local for DTS Local Server

3.6.5 Configuring DTS Clerks

Detailed procedures for tasks performed by the SA are provided in the sections that follow.

The procedures assume that all planning involved has been concluded well in advance and the SA has been properly trained to perform the configuration.

Table 3.6-5 presents the steps required to configure DTS clerks in a condensed manner. If you are already familiar with the procedures, you may prefer to use the quick-step tables. If you are new to the system, or have not performed this task recently, you should use the detailed procedures presented below.

This section explains how to configure DTS clerks. To begin configuring the DTS clerks, execute the procedures steps that follow:

You must configure a DTS clerk (client) on any system not running a DTS server. A DTS clerk is not stared automatically via the dce_config DCE Client menu option; you must explicitly start a DTS clerk from the DTS menu under Additional Server Configuration.

1 Start dce_config on the system that you wish to configure with a DTS clerk.

2 From the DCE Main menu, select Configure (selection 1).

3 From the DCE Configuration menu, select Additional Server Configuration (selection 2).

4 From the Additional Server Configuration menu, select DTS (selection 2).

5 From the DTS Configuration menu, select DTS Clerk (selection 3).

This node is now a DTS clerk.

Table 3.6-5. Configuring DTS Clerks - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	(No entry)	**start dce_config**
2	(No entry)	**select 1 from DCE Main menu**
3	(No entry)	**select 2 from DCE Configuration menu**
4	(No entry)	**select 2 from Additional Server Configuration menu**
5	(No entry)	**select 3 from DTS Configuration menu**

3.6.6 Configuring GDA Servers

The DCE Global Directory Agent (GDA) facilitates communication between DCE cells. This section describes how to start the GDA server. A GDA server can only be configured on an exiting client system or CDS server system.

Detailed procedures for tasks performed by the SA are provided in the sections that follow.

The procedures assume that all planning involved has been concluded well in advance and the SA has been properly trained to perform the configuration.

Table 3.6-6 presents the steps required to configure GDA servers in a condensed manner. If you are already familiar with the procedures, you may prefer to use the quick-step tables. If you are new to the system, or have not performed this task recently, you should use the detailed procedures presented below.

This section explains how to configure GDA servers. To begin configuring the GDA servers, execute the procedures steps that follow:

1 From the DCE Main menu, select Configure (selection 1).

2 From the DCE Configuration menu, select Additional Server Configuration (selection 2).

3 From the Additional Server Configuration menu, select GDA Server (selection 7).

The system configures the GDA server and starts the GDA server daemon, gdad.

Table 3.6-6. Configuring GDA Servers - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	(No entry)	**select 1 from DCE Main menu**
2	(No entry)	**select 2 from DCE Configuration menu**
3	(No entry)	**select 7 from Additional Server Configuration menu**

3.6.7 Creating Security Server Replica

A new feature of HP DCE/9000 is Security Server Replication, which provides for improved cell performance and reliability. These steps will allow you to create a security replica via dce_config.

Detailed procedures for tasks performed by the SA are provided in the sections that follow.

The procedures assume that all planning involved has been concluded well in advance and the SA has been properly trained to perform the configuration.

Table 3.6-7 presents the steps required to create a security server replica in a condensed manner. If you are already familiar with the procedures, you may prefer to use the quick-step tables. If you are new to the system, or have not performed this task recently, you should use the detailed procedures presented below.

This section explains how to create a security server replica. To begin creating a security server replica, execute the procedures steps that follow:

1 From the DCE Main menu, select Configure (selection 1).

2 From the DCE Configuration menu, select Additional Server Configuration (selection 2).

3 From the Additional Server Configuration menu, select Replica Security Server (selection 8).

4 Enter the keyseed for the initial datatbase master key.

The default name for the replica is subsys/dce/sec/$HOSTNAME. If you wish to change the name of the security replica that is created by dce_config, change the value of SEC_REPLICA, either in the file /opt/dcelocal/etcdce_com_env or in the shell environment from which dce_config is run. Note that you must do this before running dce_config.

Table 3.6-7. Creating a Security Server Replica - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	(No entry)	**select 1 from DCE Main menu**
2	(No entry)	**select 2 from DCE Configuration menu**
3	(No entry)	**select 8 from Additional Server Configuration menu**
4	(No entry)	**Enter keyseed for initial database master key**

3.6.8 Unconfiguring DCE Client

The UNCONFIGURE option removes the target machine from the cell Security database and the CDS namespace. DCE daemons must be running on the system executing the UNCONFIGURE option If daemons have been stopped, use the START option on the DCE Main Menu to restart them before using UNCONFIGURE. A successfully configured client system can be unconfigured locally. If there were any errors in configuring the client system as a security or directory service client, then the client must be unconfigured from some other system in the cell. Do not use the UNCONFIGURE option on a system that is used as a Security server or a CDS server. The UNCONFIGURE option removes the system from a cell. If the system is used as a Security server or a CDS server, UNCONFIGURE will break the cell.

Detailed procedures for tasks performed by the SA are provided in the sections that follow.

The procedures assume that all planning involved has been concluded well in advance and the SA has been properly trained to perform the configuration.

Table 3.6-8 present the steps required to unconfigure a DCE client in a condensed manner. If you are already familiar with the procedures, you may prefer to use the quick-step tables. If you are new to the system, or have not performed this task recently, you should use the detailed procedures presented below.

This section explains how to Unconfigure a DCE Client. To begin Unconfiguring DCE Client, execute the procedures steps that follow:

1 From the DCE Main menu, select Unconfigure (selection 4).

2 Enter the host name of the node to be unconfigured.

3 Unconfiguring a node will remove the node's ability to operate in the cell. A reconfiguration of the node will be required.

4 You will be asked if you wish to continue. Type **y** to continue.

5 Enter the Cell Administrator's principal name and password.

WARNING: A dce_config REMOVE will need to be performed from node baltic before reconfiguring it.

6 From the DCE Main menu, select Remove (selection).

Remove will remove the node's ability to operate in the cell. A reconfiguration of the node will be required. If this is not a server node, then this node should be unconfigured before a REMOVE is done.

7 You will again be asked if you wish to continue. Type **y** to continue.

8 From the DCE Configuration menu, select Exit (selection 99).

You have now exited from dce_config.

Table 3.6-8. Unconfiguring DCE Client - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	(No entry)	**select 4 from DCE Main menu**
2	(No entry)	**enter host name of node to be configured**
3	(No entry)	**type y to continue**
4	(No entry)	**enter Cell Administrator's principal name and password**
5	(No entry)	**select 5 from DCE Main menu**
6	(No entry)	**type y to continue**
7	(No entry)	**select 99 from DCE Configuration menu**

3.7 ECS Assistant

This section covers the procedures necessary the System Administrator (SA) and/or Operator (OPR) to install new subsystem application software and monitor the system. This tool assists in the coordination and installation of software across multiple heterogeneous host machines.

ECS Assistant, is an easy-to-use GUI tool, has been developed to facilitate ECS SSI&T activities. Currently, the ECS Assistant is comprised of three major scripts: EcCoAssist, EcCoModemgr, and EcCoEsdtmgr. These scripts provide users with a Graphical User Interface to perform functions such as subsystem server startup and shutdown, ESDT management, and database review when using the ECS system. During the course of performing their tasks, SSI&T operators can use ECS Assistant to performs several functions through its GUI, which are listed in the **Activity Checklist** table.

Table 3.7-1. ECSAssistant - Activity Checklist

Order	Role	Task	Section	Complete?
1	SA	Install Subsystem Application Software	(I) 3.7.1	
2	SA/OPR	Start up and shut down servers for each subsystem	(P) 3.7.2	
3	SA/OPR	Graphically monitor the server up/down status	(P) 3. 7.3	
4	SA/OPR	Open and view the detailed log files for each server used	(P) 3. 7.4	
5	SA/OPR	Review various databases used in the ECS system	(I) 3. 7.6	
**		**Refer to Drop 4 Build Plan - Appendix D for**		
		Additional Information, if necessary		

In the following sections, we will address aspects of how to use the ECS Assistant in our SSI&T activities. This section explains how to use ECS Assistant to facilitate and manage the subsystems and their servers, including:

- server start up and shut down
- monitor servers for each subsystem, including using the ECS Monitor and ECS logfile viewer.

Table 3.7-2 outlines the assumptions that are required for all of the procedures defined in this section to operate properly.

Table 3.7-2. Assumptions - Checklist

Order	Role	Task	Section	Complete ?
1	SA	ECS Assistant has been properly installed.	ALL	
2	SA/OPR	Required environment variables have been set properly.	ALL	

3.7.1 Using ECS Assistant to Install Subsystem Software

Refer to the Drop 4 Build Plan - Appendix D

3.7.1.1 DCE CELLS

TBD

3.7.1.2 Making Config Parameter Files

Refer to the Drop 4 Build Plan - Appendix D

3.7.2 Using ECS Assistant to Start Up/Shut Down Servers

This procedure describes routings for using the ECS Assistant GUI to start up and shut down subsystem servers. The procedure described here will apply to all the servers from different subsystems.

To run the ECS Assistant, execute the procedure steps that follow:

Subsystem Server Start Up / Shut Down

1 Log into one of the host machines.

2 At the UNIX prompt on the host from which the ECS Assistant is to be run, type:
setenv DISPLAY *hostname*:0.0, press **Return**.

- 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.
- To verify the setting, type **echo $DISPLAY**, press **Return**.

3 At the UNIX prompt on the host from which the ECS Assistant is to be run, type:
setenv ECS_HOME /usr/ecs, press **Return**.

- To verify the setting, type **echo $ECS_HOME**, press **Return**.

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

- The ***ViewName*** is the ClearCase view to be used while the ECS Assistant is running in this session. For example, type **cleartool jdoe**, press **Return**.
- 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 **Return**. Then type:
EcCoAssist &, press **Return**.

- **/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 Figure 3.7-1.

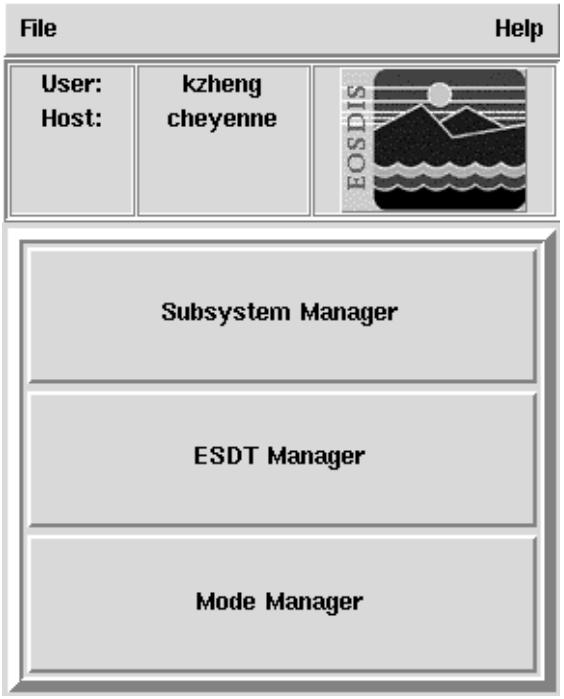


Figure 3.7-1. ECS Assistant GUI

6 At the ECS Assistant GUI, click the **Subsystem Manager** pushbutton.

- This will invoke the Subsystem Manager GUI, as indicated in Figure 3.7-2.

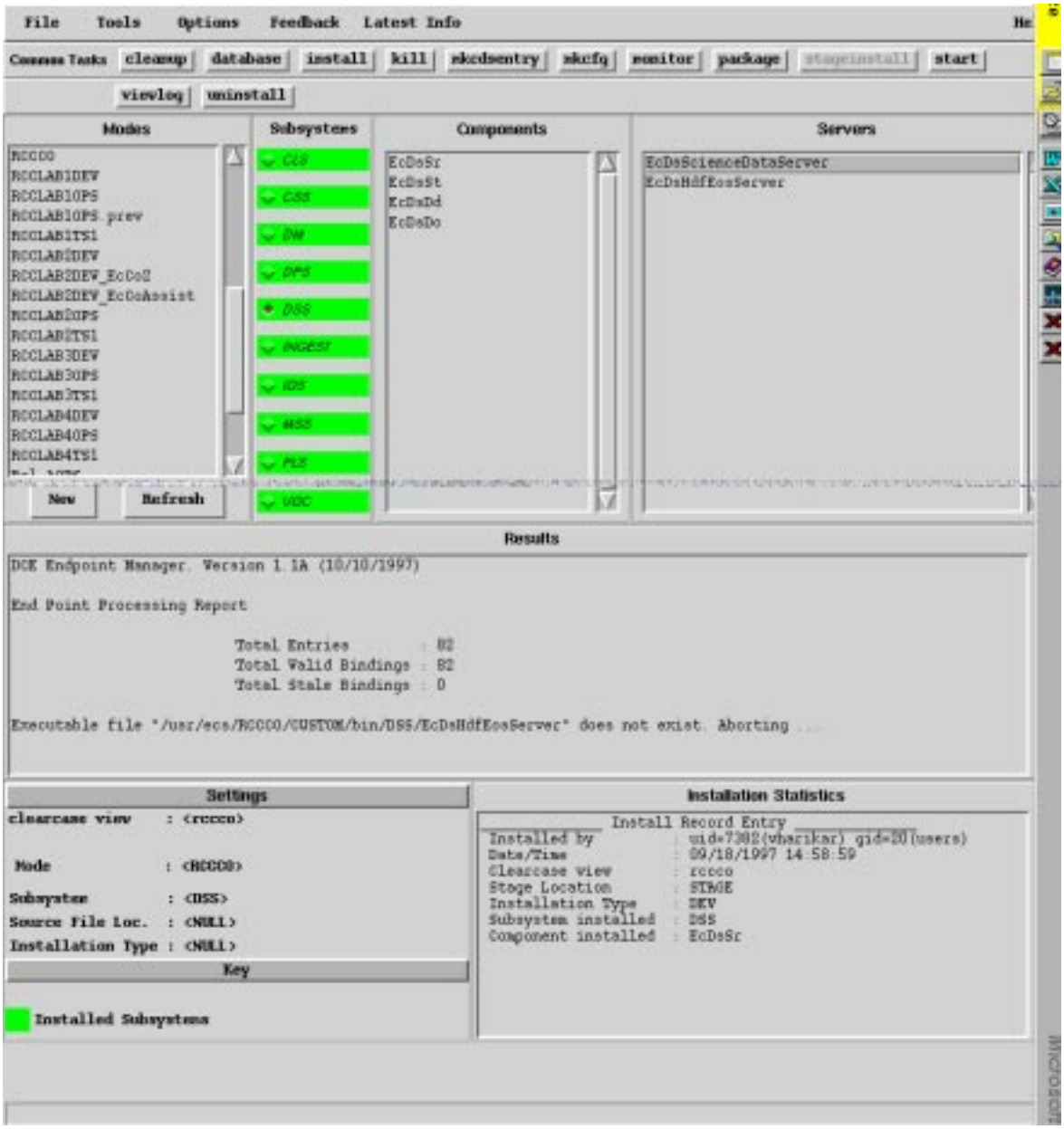


Figure 3.7-2. Subsystem Manager GUI

7 Select a mode by clicking a mode in the mode listing.

- Once the mode is selected, the color of the subsystem name list is changed.

8 Select a subsystem with the **Subsystem** radio button.

- The component list for the selected subsystem will appear in the component window.

9 Select a component by clicking the component name under the component window.

- The selected component will be highlighted.
- The server list corresponding to that component will appear in the server window.

10 Select a server by clicking the server name from the server list under the servers window.

- The server selected is highlighted.

11 To start a server up or shut it down:

- Click the **start** button in the common tasks bar. This will start up the selected server.
- Click the **kill** button in the common tasks bar. This will shut down the selected server.

12 Repeat steps 7-11 to start up or shut down other servers.

13 To exit the Subsystem Manager GUI, select **File..Exit** in the menu bar of the Subsystem Manager GUI.

- This will terminate the Subsystem Manager GUI**.**

3.7.3 Using ECS to Perform System Monitoring

ECS Assistant provides two ways to monitor server status. The first one is by performing tail -f to log files which record the important activity history performed on the servers. The other way is by using a database table to display server up/down status' dynamically. These monitoring methods are described in the following sections.

3.7.3.1 Using ECS Assistant to Open / View Log Files for a Selected Server

Log files are used extensively in the ECS system to record a history of activity performed on the system. They provide useful information about server activities. ECS Assistant provides an easy way to access and view these log files. In the Subsystem Manager GUI, there is one button called **viewlog** in the Common Tasks bar. Click this button to invoke a log file GUI, as shown in Figure 2. You can review the log files for a particular server by choosing the server name from the Menu for the Subsystem to which it belongs. You can also view all of the log files for a component by choosing it in the Components menu. Menu entries are dimmed if no log files are present. The following example shows how to use this GUI to open log files for a particular server.

To run the Log Viewer, execute the procedure steps that follow:

1 Click the **viewlog** button in the Subsystem Manager GUI.

- This will invoke the log viewer GUI.

2 To open and view log files for a particular server, select a server from the Subsystem pull down menu, then click the server name.

- This will open all the log files corresponding to that server.
- The log file name is indicated in the title bar for each log file GUI.

3 The log file GUI provides the following options for users to view log file contents. Follow the guidance in the GUI to select the proper options:

- **Foreground color** for changing the foreground color.
- **Background** color for changing the background color.
- **Font size** for changing font sizes.
- **View entire file** for displaying the entire file.
- **Continuous update (tail -f)** for displaying the updated log file continuously.
- **Search for** for performing word searches in the log file.
- **Print** for printing the log file.

4 To view log files for other servers, repeat steps 1-3.

5 Exit the log file by pressing **EXIT**.

3.7.3.2 Using ECS Assistant to Monitor Server Status

ECS Assistant provides another convenient way to monitor the status of the servers by listing their up/down condition. The status flag for a server is up or down indicating whether or not that server is running.

To start up the ECS monitor GUI, execute the procedure steps that follow:

1 At the ECS **Subsystem Manager** GUI, select a mode by clicking a mode in the mode list.

- The mode should be the one to be used for SSI&T.
- Once the mode is selected, the color of the subsystem name list is changed.

2 Select a subsystem by clicking the radio button next to the subsystem name under the subsystem component window.

- The selected subsystem radio button will be highlighted.
- The components corresponding to that the subsystem will be displayed in the component window.

3 Select a component by clicking its name under the component window.

- All the servers for the selected component will be displayed in the server window.

4 If desired, click the **monitor** button from the common tasks window.

- This will invoke the ECS Monitor GUI window as shown in Figure 3.7-3.
- The status UP/DOWN indicates whether the server is running.

5 To see which host each server is running on, click the **cdsping all servers** button.

- This will invoke the ECS Monitor (cdsping) GUI as indicated in Figure 3.7-4.
- The host name for each running server is listed

6 Both ECS monitor GUI and ECS Monitor (cdsping) GUI can be updated by clicking the **update** button in the GUI.

- This will cause the list to update to the current status.

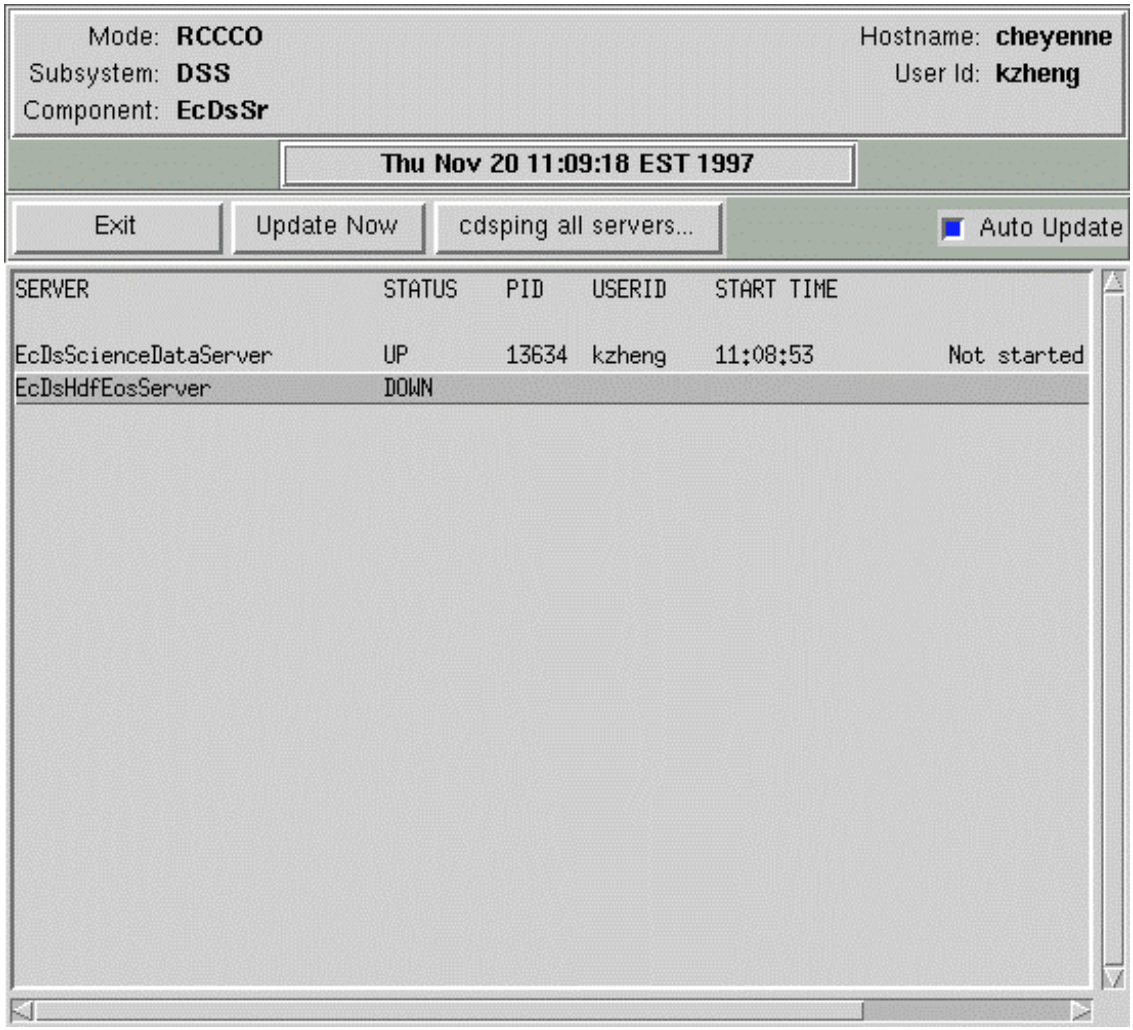


Figure 3.7-3. Server Monitor GUI

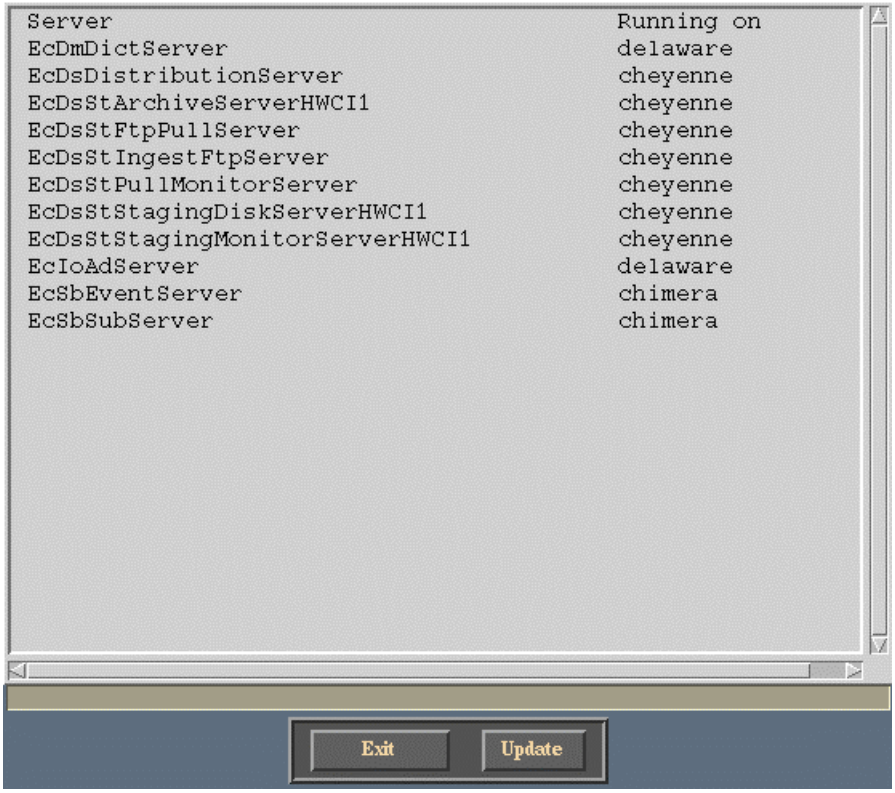


Figure 3.7-4. cdsping GUI

7 To monitor other servers, repeat steps 2-4.

8 To exit, click the **EXIT** button.

- This will end the monitor GUI.

This page intentionally left blank.

4. Database Administration

4.1 Overview of Database Administration

The Database Administrator or DBA, is the individual or group responsible for the installation, configuration, update, maintenance, and overall integrity, performance and reliability of the SQL Server database. In general, the DBA is concerned with the availability of the server, the definition and management of resources allocated to the server, the definition and management of databases and objects resident on the server, and the relationship between the server and the operating system.

Each section in this chapter provides necessary background information, followed by step-by-step instructions and actual scripts where applicable.

4.2 SQL Server Environment

4.2.1 Naming Conventions,

As one of the most important, yet least applied concepts, naming conventions are presented in this chapter by examples according to the following rules.

Rule1: Regardless of the length of the name, it should indicate the function and/or content of the object

Rule_2: Only easily understandable abbreviations should be used

Rule_3: Parts of names are separated by underscores "_", only one optional suffix is permitted (appended to the name by a . ".")

Rule_4: The full path of the object is considered to be part of the name

The names of the databases and tables themselves may or may not follow the above rules, these rules are specifically for the DBA to work with SQL Server objects, and files in the UNIX environment.

All **COTS** software is installed in the /usr/ecs/OPS/COTS directory.

All **SYBASE** software is located in the Sybase home directory (**$SYBASE**).

All backups are located in **$SYBASE**/sybase_dumps directory, which may or may not be on a separate physical disk.

Note

It is strongly recommended that backups be stored on a separate physical disk.

The database dumps are kept for a period of 2 days and also stored on a disk by Networker everyday. The database dumps are named as follows:

dbname.dat_YYMMDDHHMM.Z

where MMDDHHMM is the “sortable” eight digit month, day, hour, and minute. For example, on the date this chapter was written, a backup directory called **backups_for_99021100024.Z**

All SQL script files have the extension **.sql** as a suffix. Their names reference the objects they create or functions they perform, and are all located in **$SYBASE**/scripts.

SQL statement must follow precise syntactical and structural rules, and may include only SQL keywords, identifiers (names of databases, tables, or other database objects), operators, and constants. The characters that can be used for each part of a SQL statement vary from installation to installation and are determined in part by definitions in the default character set that version of the server uses.

For example, the characters allowed for the SQL language, such as SQL keywords, special characters, and Transact-SQL extensions, are more limited than the characters allowed for identifiers. The set of characters which may be used for data is much larger and includes all the characters that can be used for the SQL language or for identifiers.

The sections that follow describe the sets of characters that can be used for each part of a statement. The section on identifiers also describes naming conventions for database objects.

4.2.1.1 SQL Data Characters

The set of SQL data characters is the larger set from which both SQL language characters and identifier characters are taken. Any character in SQL Server's character set, including both single-byte and multibyte characters, may be used for data values.

4.2.1.2 SQL Language Characters

SQL keywords, Transact-SQL extensions, and special characters such as the comparison operators > and <, can be represented only by 7-bit ASCII values A- Z, a -z, 0-9, and the following ASCII characters:

4.2.1.3 Identifiers

Conventions for naming database objects apply throughout SQL Server software and documentation. Identifiers can be up to 30 bytes in length, whether or not multibyte characters are used. The first character of an identifier must be declared as an alphabetic character in the character set definition in use on Server.

The @ sign or _ (underscore character) can also be used. The @ sign as the first character of an identifier indicates a local variable.

Temporary table names must either begin with # (the pound sign) if they are created outside tempdb or be preceded by "tempdb..".

Table names for temporary tables that exist outside tempdb should not exceed 13 bytes in length, including the number sign, since SQL Server gives them an internal numeric suffix.

After the first character, identifiers can include characters declared as alphabetic, numeric, or the character $, #, @, _, ¥ (yen), or £ (pound sterling). However, you cannot use two @@ symbols together at the beginning of a named object, as in "@@myobject." This naming convention is reserved for global variables, which are system-defined variables that SQL Server updates on an ongoing basis.

The case sensitivity of SQL Server is set when the server is installed and can be changed by a System Administrator. To see the setting for your server, execute this command: sp_helpsort

4.2.1.4 Delimited Identifiers

Delimited identifiers are object names enclosed in double quotes. Using delimited identifiers allows you to avoid certain restrictions on object names. You can use double quotes to delimit table, view, and column names; you cannot use them for other database objects.

Delimited identifiers can be reserved words, can begin with non-alphabetic characters, and can include characters that would not otherwise be allowed. They cannot exceed 28 bytes.

Before creating or referencing a delimited identifier, you must execute:

set quoted_identifier on

The names of database objects need not be unique in a database.

However, column names and index names must be unique within a table, and other object names must be unique for each owner within a database. Database names must be unique on SQL Server.

If you try to create a column using a name that is not unique in the table or to create another database object such as a table, a view, or a stored procedure, with a name that you have already used in the same database, SQL Server responds with an error message.

You can uniquely identify a table or column by adding other names that qualify it, that is, the database name, the owner's name, and, for a column, the table name or view name. Each of these qualifiers is separated from the next by a period:

database.owner.table_name.column_name

database.owner.view_name.column_name

The same naming syntax applies to other database objects. You can refer to any object in a similar fashion:

If the quoted_identifier option of the set command is on, you can use double quotes around individual parts of a qualified object name.

Use a separate pair of quotes for each qualifier that requires quotes.

For example, use:

database.owner."table_name"."column_name"

rather than:

database.owner."table_name.column_name"

The full naming syntax is not always allowed in create statements because you cannot create a view, procedure, rule, default, or trigger in a database other than the one you are currently in. The naming conventions are indicated in the syntax as:

[[database.]owner.]object_name or: [owner.]object_name

The default value for owner is the current user, and the default value for database is the current database. When you reference an object in SQL statements, other than create statements, without qualifying it with the database name and owner name, SQL Server first looks at all the objects you own, and then at the objects owned by the Database Owner, whose name in the database is "dbo." As long as SQL Server is given enough information to identify an object, you need not type every element of its name. Intermediate elements can be omitted and their positions indicated by periods:

database..table_name

You must include the starting element, in this case, database, particularly if you are using this syntax when creating tables. If you omit the starting element, you could, for example, create a table named ..mytable. This naming convention prevents you from performing certain actions on such a table, such as cursor updates.

When qualifying a column name and a table name in the same statement, be sure to use the same naming abbreviations for each; they are evaluated as strings and must match or an error is returned.

4.2.1.5 Identifying Remote Servers

You can execute stored procedures on a remote SQL Server, with the results from the stored procedure printed on the terminal that called the procedure. The syntax for identifying a remote server and the stored procedure is:

[execute] server.[database].[owner].procedure_name

You can omit the execute keyword when the remote procedure call is the first statement in a batch. If other SQL statements precede the remote procedure call, you must use execute or exec. You must give the server name and the stored procedure name. If you omit the database name, SQL Server looks for procedure_name in your default database. If you give the database name, you must also give the procedure owner's name, unless you own the procedure or the procedure is owned by the Database Owner.

If the server name in interfaces is in uppercase letters, you must use it in uppercase letters in the remote procedure call.

In all cases throughout this chapter, when actual examples are provided, those which reference UNIX commands will be preceded by a “%”, and those that reference SQL statements will be preceded by a number and a “>” (e.g. 1>sp_help tablename).

The terms described in the following table will be used throughout this chapter.

Table 4.2-1. SQL Server General Definitions

Term	Definition
SQL Server	The server in the Sybase client/server architecture. SQL Server manages multiple databases and multiple users, keeps track of the actual location of data on disks, maintains mapping of logical data description to physical data storage, and maintains data and procedure caches in memory.
Client	SYBASE Open Client software located in the /tools/sybOCv(TBD) directory for SUN and HP platforms SYBASE Open Client software located in the /tools/sybOCv(TBD) directory for SGI platform
Backup Server	Similar to the dataserver, it uses a separate UNIX process to off load the cycles associated with DUMP and LOAD commands
backups	The set of UNIX files containing full database dumps, transaction log dumps, and dbcc output
dbcc	Database Consistency Checker - a utility program designed to check the logical and physical consistency of a database
sybase root directory	/usr/ecs/OPS/COTS/sybase, this is the home directory for all SYBASE software and related products and is referenced both in UNIX and in the rest of this document as **$SYBASE**
interfaces file	Lists the names and access paths for all servers and backup servers. This file is located in the **$SYBASE**
sa	System Administrator login, this is the superuser of the SQL Server
scripts	UNIX script programs located in **$SYBASE**/scripts and related subdirectories {$ecs_Home}/{mode}/custom/dbms/{subsystem}
showserver	A utility invoked at the UNIX command prompt to display active servers, located in **$SYBASE**/install.
SQL scripts	SQL and command statements located in **$SYBASE**/scripts and related subdirectories and /{$ecs_Home}/{mode}/custom/dbms/{subsystem}
Server Name	The name of the database server for a specific application in different modes **EX. -** PDPS application database server in OPS mode EX.- Pdps_TS1 in TS1 mode EX. -pdps _TS2 in TS2 mode
Port Numbers	The port number to be utilized by the above listed servers.
Release Directory	$SYBASE
SQL	Structured Query Language

4.2.2 SQL Server Directory Structure

The **sybase** directory structure is described in the following table. Subdirectories under the **scripts** can contain template files with easy to modify examples of SQL and SQL command syntax.

Table 4.2.2-1. SYBASE Directory Structure

Directory	Contains
$SYBASE/bin	Utilities necessary to load, run, and access the server
$SYBASE /install	Files used to start and initialize dataservers, backupserver and to record server messages (errorlogs)
$SYBASE /lib	db-lib, ct-lib, and xa-lib client library files used by applications to gain access to the server (local to server) *__Applications use automounted libraries__.
$SYBASE /scripts	Root directory for all script files executed on the server
$SYBASE /sybase_dumps	Root directory that contains all backup subdirectories, it is recommended, but not required, that this directory be on a separate physical disk. Dumps both database and transaction logs. **Backups are stored on disk in the backup subdirectories.
backup subdirectories $SYBASE /sybase_dumps/dumps $SYBASE /sybase_dumps/trans $SYBASE /sybase_dumps/dumps/logs $SYBASE /sybase_dumps/trans/logs $SYBASE /sybase_dumps/Week1 $SYBASE /sybase_dumps/Week2 $SYBASE /sybase_dumps/Week1/logs $SYBASE /sybase_dumps/Week2/logs	A cron job is run at night to move data from the current (week1) directory to the previous (week2) directory. Then, a dump of the databases and transaction logs is executed and is stored in the current directory. All logs are written to the log directory. Files are saved using the following naming convention:: dbname.dat.YYMMDDHHMM.Z - full database dumps dbname.tran.YYMMDDHHMM.Z - full transaction log dumpsdbname_backup.log. dbname_ERR.log.MMDDHHMM - Error log filesdbname_dbcc.log. MMDDHHMM
**xxdmh02 serves as a remote Backup Server	**xx are the 2 letter codes to identify a DAAC site (i.e., g0 = Goddard)

4.2.3 SQL Server Installation

SYBASE SQL Server Version (TDB) has been installed and configured by the ECS Installation Staff. Shared memory and disk resources have been allocated and configured by the System Administrator, and both the client and server portions have been set up by the DBA prior to shipment. The following table describes parameters and options used during installation.

Table 4.2.3-1. SQL Server Parameters and Options

Parameters Name	Brief Explanation/Settings
Retry Count	5 seconds
Retry Delay	5 seconds
Master device	28 Mb raw partition
Master Device Location	
Backup Server Name	SYB_BACKUP
sybsystemprocs	$SYBASE/devices/(MachineName)_sybprocs.dat, 19 Mb and on it's own device
Errorlog	$SYBASE/install/mode.errorlog (**mode** indicates the application)
Current default language	us_english
Current default character set	iso_8859-1 (Latin-1)
Current sort order	Binary ordering, for the ISO 8859/1 or Latin-1 character set (iso_1).
Internal auditing	On
sybsecurity database size	175 Mb - Varies – depends on disk allocations
sybsecurity device	sybsecurity, positioned on a 175 Mb raw partition

The installation script files are located in the **$SYBASE**/install directory. SQL Server installation is performed by an authorized user with the **sybinit** utility also located in the **$SYBASE**/install directory. See your UNIX System Administrator and the SYBASE SQL Server Installation Guide.

4.3 Database Administrator Responsibilities

The following subsections detail the most common functions that a DBA will perform.

4.3.1 Startup of SQL Server

Use **startserver** to start an SQL Server and/or a Backup Server. This command can only be issued by the **Sybase** user.

Syntax: % **startserver** [-f runserverfile]

The "runserverfile" is contained in the **$SYBASE**/install directory.

Note

SQS server should be started after the SQL Server

4.3.2 Shutdown of SQL Server

Use **shutdown** to bring the server to a halt. This command can only be issued by the Sybase System Administrator (sa).

Syntax:1> **shutdown [backup_server_name]]** [with] [wait] [with nowait]

2> **go**

The "with wait" is the default option. This option brings SQL Server down gracefully.

The “with nowait” option shuts down the SQL Server immediately without waiting for currently executing statements to finish.

If you do not give a server name, shutdown shuts down the SQL Server you are using.

When you issue a shutdown command, SQL Server:

1. Disables logins, except for System Administrators
2. Performs a checkpoint in each database, flushing pages that have changed from memory to disk
3. Waits for currently executing SQL statements or procedures to finish

In this way shutdown minimizes the amount of work that automatic recovery must do when you restart SQL Server.

To see the names of the Backup Servers that are accessible from your SQL Server, execute

sp_helpserver. Use the value in the name column in the shutdown command. You can only shut down a Backup Server that is:

Listed in sysservers on your SQL Server, and

Listed in your local interfaces file.

Note 1

It recommended that "with wait" option be used. This allows executing statements to finish.

Also it is recommended that you perform a checkpoint of all database prior to shutdown.

Note 2

SQS server should be started after the SQL Server

4.3.3 Showing SQL Server(s)

Use **showserver** to determine whether the SQL Server(s) and/or Backup Server(s) are running.

Syntax: % **showserver**

The “**showserver**” is contained in the **$SYBASE**/install directory

Example: UNIX processes running the various servers:

UID PID PPID C STIME TTY TIME COMD

sybase 671 669 80 Apr 17 ? 80:05 /usr/ecs/OPS/COTS/Sybase/bin/dataserver -d /dev/rdsk/c1t0d0s1 -g0sps06_srvr

sybase 665 663 80 Apr 17 ? 50:02 /usr/ecs/OPS/COTS/sybase/bin/backupserver -g0sps06_backup -e/usr/ecs/OPS

4.4 Allocation of Resources

SQL Server can make reasonable default decisions about many aspects of storage management, such as where databases, tables, and indexes are placed and how much space is allocated for each one. However, the System Administrator has ultimate control over the allocation of disk resources to SQL Server and the physical placement of databases, tables, and indexes on those resources.

4.4.1 Creating Logical Devices

A logical device is created when the UNIX System Administrator determines that new disk space is available for use by SYBASE software, databases, transaction logs, and/or backups. Either raw disk partitions or UNIX filesystem partitions can be used to create a logical device. The creation of a logical device is a mapping of physical space to a logical name and virtual device number (**vdevno**) contained in the SQL Server **master** database. The **disk init** command is used to initialize this space. After the disk initialization is complete, the space described by the physical address is available to SQL Server for storage, and a row is added to the **sysdevices** table in the **master** database.

A System Administrator initializes new database devices with the disk init command.

Disk Init does the following: Maps the specified physical disk device or operating system file to a database device name

Lists the new device in master..sysdevices

Prepares the device for database storage

Note

Before you run disk init, see the SQL Server installation and configuration guide for your platform for information about choosing a database device and preparing it for use with SQL Server. You may want to repartition the disks on your computer to provide maximum performance for your Sybase databases.

Disk init divides the database devices into allocation units of 256 2K pages, a total of 1/2MB. In each 256-page allocation unit, the disk init command initializes the first page as the allocation page, which will contain information about the database (if any) that resides on the allocation unit.

Note

After you run the disk init command, be sure to use dump database to dump the master database. This makes recovery easier and safer in case master is damaged. If you add a device and fail to back up master, you may be able to recover the changes with disk reinit.

Syntax: disk init

name = "device_name" ,

physname = "physicalname" ,

vdevno = virtual_device_number ,

size = number_of_blocks

[, vstart = virtual_address ,

cntrltype = controller_number]

4.4.1.1 Example of Creating a Logical Device

A raw partition on a RAID device has been made available to SQL Server by the UNIX System Administrator. Essentially, the actual name of the raw device **c2t0d1s3** has had it's ownership changed to **sybase** and it's group changed to **user**.

1 . In **$SYBASE**/scripts/create.devices, DBA makes a script file from the template.

Syntax: % cd /usr/ecs/OPS/COTS/sybase/scripts/create.devices

% cp template.sql data_dev1.sql

2 . Appropriate items are modified so that the script file resembles the following:

1> disk init

2> name = "data_dev1",

3> physname = "/dev/rdsk/c2t0d1s3",

4> vdevno = 3,

5> size = 128000

6> go

7> sp_helpdevice data_dev1

8> go

3 . DBA runs the script from the UNIX command prompt:

Syntax: % isql -Usa -S**servername** -idata_dev1.sql -odata_dev1.out

4 . DBA checks the data_dev1.out file for success

4.4.2 Creating and Altering Databases

A user database is created by the DBA with a script containing the **create database** command. A database is created on one or more physical devices. Specifying the device is optional - but highly

recommended. When indicating the device, you use the logical name you specified as part of a **disk init** (described above). Unlike the **disk init** command, the size of the database data and log components is specified in MB instead of 2K pages.

4.4.2.1 Example of Creating a Database

The logical device **data_dev1** has been created (as above) along with another device called **tx_log1** (for transaction logging).

1. In **$SYBASE**/scripts/create.databases directory, DBA makes a script file from the template.

Syntax: % cd /usr/ecs/OPS/COTS/sybase/scripts/create.databases

% cp template.sql userdb.sql

2 . Appropriate items are modified so that the script file resembles the following:

1> create database UserDB on data_dev1 = 100 log on tx_log1 = 50 [with override]

2> go

3> sp_helpdb UserDB

4> go

3 DBA runs the script from the UNIX command prompt:

Syntax: %isql -Usa -S**servername** -iuserdb.sql -ouserdb.out

4 DBA checks the userdb.out file for success

4.4.2.2 Example of Altering a Database

The user database **UserDB** has run out of space and it has been determined that it should be increased by 50MB.

1 In **$SYBASE**/scripts/create.databases, DBA creates a script file containing the ALTER DATABASE command (named alter_userdb.sql)

Syntax: Alter database UserDB on data_dev3 = 50

2 DBA runs the script from the UNIX command prompt:

Syntax: % isql -Usa -S**servername** -ialter_userdb.sql -oalter_userdb.out

3 DBA checks the alter_userdb.out file for success

4.4.2.3 Data Placement - Segmentation

Segments are named subsets of the database devices available to a particular SQL Server database. Segment names are used in **create table** and **create index** commands to place tables or indexes

on specific database devices. Using segments allows the DBA to better control the size of database objects and may improve performance by spreading i/o more evenly across devices.

Once the database device exists and is available, the segment can be defined with the system stored procedure **sp_addsegment**.

Syntax: sp_addsegment segname, dbname, devname

After the segment has been defined in the current database, the **create table** or **create index** commands use the optional clause "on segment_name" to place the object on a particular segment.

Syntax: create table table_name (column_name datatype ...) [on segment_name]

create [clustered | nonclustered] index index_name on table_name (columns)

Use **sp_helpdb** database_name to display the segments defined for that database.

Use **sp_helpsegment** segment_name to list the objects on the segment and show the mapped devices.

4.4.2.3.1 Example of Creating a Segment

The DBA receives a request to create a segment for the storage of the DATA_INFO table indexes in the pdps_db_ops database, on a separate physical disk. Two devices **data_dev1** and **data_dev2** have already been created and are located on different physical disks.

1 . In **$SYBASE**/scripts/create.segments directory, DBA makes a script file from the template.

Syntax: % cd /usr/ecs/OPS/COTS/sybase/scripts/create.segments

% cp template.sql segments_dev1.sql

2 . The script file is modified so that it resembles the following:

1> sp_addsegment seg1_dev1, pdps, data_dev1

2> sp_addsegment seg1_dev2, pdps, data_dev2

3> go

3 . DBA runs the script from the UNIX command prompt:

Syntax: %isql -Usa -S**servername** -ipdps_db_ops_segments.sql \

-opdps_db_ops_segments.out

4 . DBA checks the pdps_segments.out file for success

5 . When the table and indexes are created according to the instructions in Section 4.4.6, the "on seg1_dev1" must be appended to the DATA_INFO.sql **create table** statement, and the "on seg1_dev2" must be appended to the DATA_INFO_indexes.sql CREATE

INDEX statement.

Syntax: **create index** DATA_INFO_IDX on DATA_INFO (DI_ID) on SEG1_DEV2

4.5 Loading a database you have created into a different database:

Occasionally, you may want to create an exact copy of a database of you system. First, dump the existing database. Then create a database to load with this dump. The database does not have to be the same size as the original. The only requirement is that the destination database must be at least as large as the dumped database and have the same beginning fragments as the original database. This information can be obtained from saved database creation scripts, or by running the following command:

select segmap,'Size in MB'=size/512 from sysusages where dbid= db_id("database_name")

Example:

suppose your database was created with the following statement:

create database dbname on datadevice1 = 1000,

log on Logdevice1 = 200

go

alter device dbname on datadevice2 = 500 running:

select segmap,'Size in MB'=size/512 from sysusages

where dbid= db_id("dbname")

would return:segmap Size in MB

3 1000

4 200

3 500

You could create a 3GB database as follows and load your database into it (using "for load" option will shorten database load time):

create database newdatabase on datadevice3 = 1000 log on logdevice3 = 200

for load

go

alter database newdatabase on datadevice 3=500 for load go

alter database newdatabase on datadevice4=300 for load go

alter database newdatabase on datadevice5=1000 for load go

load database newdatabase from dbname_dump go

4.6 Monitoring Space Usage

4. 6.1 Thresholds

Thresholds are defined on segments to provide a free space value at which a procedure is executed to provide a warning or to take remedial action.

Use **sp_addthreshold** to define your own thresholds:

sp_addthreshold database_name, segment_name, free_space, procedure_name

where free_space is the number of free pages at which the threshold procedure executes; procedure_name is the stored procedure which the threshold manager executes when the number of free pages falls below the free_space value. Please see the section on Auditing later in this chapter for an example of Thresholds.

Example of Threshold Commands mentioned above:

Sp_addthreshold CustomerDB, "default", 10230, CustDefaultSegWarn

4.7 Creating Database Objects

For special cases, creation (and modification) scripts are stored in **$SYBASE**/scripts/scriptname. There should be a template for each type of object to be created.

4.7.1 Example of Creating a User Table

The DBA has received a request to create a new table in the pdps_db_ops database called **PGE_Statistics** which has three column, pge_id, pge_statistic_type, and pge_statistic.

1 . In the **$SYBASE**/scripts/create.db_objects directory, DBA creates a script file from the proper template.

Syntax: % cd /usr/ecs/OPS/COTS/sybase/scripts/create.db_objects

% cp table_template.sql PGE_Statistics_table.sql

2 . Appropriate items are modified so that the script file resembles the following:

```
1> create table PGE_Statistics (
2> pge_id                int,
3> pge_statistic_type   int,
4> pge_statistic         float )
5> go
6> sp_help PGE_Statistic
```

7> go

3 . DBA runs the script from the UNIX command prompt:

Syntax: %isql -Usa -S**servername** -iPGE_Statistics_table.sql \

-oPGE_Statistics_table.out

4 . DBA checks the PGE_Statistics_table.out file for success

Other objects are created in like manner but are not included here due to space considerations.

4.8 Creating and Managing Logins and Roles

Earlier versions of SQL Server administrative responsibilities needed to be executed by and individual logged in –literally- as sa. Now specific user logins can be assigned components of administrative responsibility, enabling you to track and audit administrative activities.

The three roles are sa_role (systems administrator) for administrative tasks, sso_role(site security officer) for security tasks, and oper_rol (operator) for backup and recovery tasks.

In order to connect to a SQL Server a login must be created by the System Administrator or a system security officer. Login details are stored in the syslogins table in the **master** database.

The system stored procedure **sp_addlogin** adds new login names to the server but does not grant access to any user database.

Syntax: **sp_addlogin** login_name, password, [,default database ,language, fullname]

In order to gain access to a database, the System Administrator, system security officer, of the specific database owner must "add" the user with the **sp_adduser** system stored procedure.

Syntax: 1> **sp_adduser**login_name [username, group_name]
2> go

4.8.1 Example of Creating a Login and Granting Database Access

The DBA has received a request to authorize John Q. Public to the pdps_db_ops database.

***It is a good practice to have a default_db, when you create a user account.**

1 . In the **$SYBASE**/scripts/create.users directory, DBA creates a script file containing the sp_addlogin command (named public.sql)

Syntax: % cd /usr/ecs/OPS/COTS/sybase/scripts/create.users

% cp template.sql public.sql

2 . DBA modifies appropriate fields so that the script resembles the following:

1> sp_addlogin jpublic,jpublic, default_db

2> go

3> use pdps (OPS mode) 4> go

5> sp_adduser jpublic

6> go

7> sp_helpuser

8> go

3 . DBA runs the script from the UNIX command prompt:

Syntax: % isql -Usa -S**servername** - public.sql -opublic.out

4 . DBA checks the public.out file for success

4.9 Permissions

Permissions are used to control access within a database. The DBA uses the **grant** and **revoke** statements to accomplish this. There are two types of permissions within a database, **Object** and **Command**. In general, **Object** privileges control select, insert, update, delete, and execute permissions on tables, views, and stored procedures. **Command** permissions control access to the **create** statements for databases, defaults, procedures, rules, tables, and views.

The syntax for the **grant** and **revoke** statements are quite similar:

grant {all [privileges] | command_list }

to { public | name_list | role_name }

revoke {all [privileges] | command_list }

from { public | name_list | role_name }

4.9.1 Example of Granting Privileges to a Specific User

The DBA receives a request that John Q. Public should be able to read the DATA_INFO table and read and update the SUBSCRIPTION_NOTIFICATION TABLE.

Syntax: 1> **grant** select on DATA_INFO to jpublic

2> **grant** select, update on SUBSCRIPTION_NOTIFICATION to jpublic

3> go

Note: It is recommended that the DBA store these command in a ".sql" file in the **$SYBASE**/scripts/create.db_objects directory, along with their results.

4.10 Backup and Recovery

Table 4.10-1. Backup and Recovery Definitions

Term	Definition
Backup Script Components	Located in the **$SYBASE** directory, they include: sybasedump, dmpdb_trns, copy_daily_dumps_to_week1, copy_daily_dumps_to_week2
Backup files	Defined in Table 4.2-2, the location of these files has been determined during server setup
Backup Statements	Generated from the sql in sybasedump these include calls to dbcc, Dump Database, and Dump Transaction commands
Backup Subdirectory	The only directory level underneath of the Backup Directory, defined in Table 4.2-2.
Backup Summary	An extraction of the successful Dump messages along with any errors generated by the Backup Statements stored in the Backup Subdirectory.

4.10.1 Automatic Backups

The following are the list of all procedures and scripts files that are currently being used for Sybase backups. There are cron jobs running at all sybase **servers** that have SQL server installed. All dump files are currently written to LOCAL machine. The site DBA is responsible for configuring the backup dump to the REMOTE sybase directory.

To check if the crontab is up and running, enter:

> crontab -l

- Example of the output:
- 019 * * 1-6 /usr/ecs/OPS/CUSTOM/dbms/COM/DBAdmin/EcCoDbSyb_DumpDb
- 012 * * 1-6 /usr/ecs/OPS/CUSTOM/dbms/COM/DBAdmin/EcCoDbSyb_DumpTran
- 021 * * 1-5 /usr/ecs/OPS/CUSTOM/dbms/COM/DBAdmin/EcCoDbSyb_CkErrorLog

NOTE:

If the crontab is not running enter:

> crontab /usr/ecs/OPS/COTS/sybase/run_sybcron

The following files will be installed by EcCoAssist to the /usr/ecs/OPS/CUSTOM/dbms/COM/DBAdmin directory:

EcCoDbSyb_README

EcCoDbSyb_DumpDb

EcCoDbSyb_DrumpTran

EcCoDbSyb_DbStat

EcCoDbSyb_SedFile

EcCoDbSyb_DboMail

EcCoDbSyb_SetupKsh

EcCoDbSyb_CkErrorLog

EcCoDbSyb_tran_log.awk

SCRIPTS	**DESCRIPTIONS**
EcCoDbSyb_SetupKsh	This file contains the SYBASE and DSQUERY (server) environment setup. This file is call by EcCoDbSyb_DumpDb, EcCoDbSyb_DrumpTran, and EcCoDbSyb_CkErrorLog scripts.
EcCoDbSyb_DumpDb	This script contains the code to dump the databases. First, it checks for any DBCC error on the master database, if there is any error on the master, the script sends an email to the DBA and exit the program. If the master database dump was successfull, then the rest of the databases are dumped. Each database has a DBCC check, if there is any error on the database then the database is NOT dumped and an email is send to the DBA. At the end, an status email is send, providing all the database names that were succefully dumped
EcCoDbSyb_DumpTran	This script contains the code to dump the transaction logs. This dumps the transaction logs for each database, it check the error log file, if the error Msg is 4207 or 4221 it will do a dump of the database firt, then it will do the trasaction dump. If there is any other error Msg then the transaction dump will fail and email will be send. At the end, an status of the transaction log dumps is email to the DBA
EcCoDbSyb_SedFile	This file contains all the database that don't need to be dump (i.e., temp, model, etc.)
EcCoDbSyb_DboMail	This file contains the email list of all the DBA's.
EcCoDbSyb_DbStat	This script updates the index table of a database. This script is called from EcCoDbSyb_DumpDb after each successfully database dump.
EcCoDbSyb_CkErrorLog	This script checks for specific database error messages from the Sybase Error Log File every hour and emails the error messages to the DBA's in the EcCoDbSyb_DboMailfile.

EcCoDbSyb_tran_log.awk	This script matches the current hour with the hour the error messages were enerated in the Error Log File. If errors found, the messages are saved in a mailfile and sent to DBA's.

THE FOLLOWING FILES MUST BE MODIFIED BEFORE RUNNING ANY OF THE ABOVE SCRIPTS:

EcCoDbSyb_SetupKsh	Make user you have the SYBASE files under /usr/ecs/OPS/COTS/sybase
EcCoDbSyb_SedFile	Add any other database that might not need to be backed up. The databases that are listed in this file do not need to be backed up.
EcCoDbSyb_DboMail	Add/delete the email of the DBA and any other email that might need to be added/deleted. All the errors and status will be send to them.
run_sybcron	The following is an example on the crontab file that should be run by a sybase user. The first one will run the EcCoDbSyb_DumpDb script that dumps the databases at midnight from Monday to Saturday. The second one, EcCoDbSyb_DumpTran script that dumps the transaction logs will run tree times a day, 10AM, 1PM and 4PM from Monday to Saturday. The Third one, EcCoDbSyb_CkErrorLog that check the SYBASE error log file will run every hour from Monday to Saturday.

0 0 * * 1-6 /usr/ecs/OPS/CUSTOM/dbms/COM/DBAdmin/EcCoDbSyb_DumpDb

0 10,13,16 * * 1-6 /usr/ecs/OPS/CUSTOM/dbms/COM/DBAdmin/EcCoDbSyb_DumpTran

0 * ** 1-6 /usr/ecs/OPS/CUSTOM/dbms/COM/DBAdmin/EcCoDbSyb_CkErrorLog

NOTE: Make sure there is an OPS mode directory with all script files.

All these scripts reside in “/usr/ecs/OPS/CUSTOM/dbms/COM/DBAdmin” directory. The assigned site DBA will be responsible for maintaining, modifying and applying necessary changes that are applicable to their site as for (security, and backup schedule).

SQL Server backups are performed nightly by a **cron** job which runs the **run_sybcron** program located in the **$SYBASE**/ directory. The following table of definitions will be used throughout the rest of this section.

Table 4.10-2. Automatic Backup Components

Component Name	Function(s)
run_sybcron	File added with the crontab -e command, contains several executable cron commands. **Example:** 00 19 * * 1-6 /data1/COTS/sybase/sybasedump
EcCoDbSyb_DumpDb	Controlling script that performs the following functions: run isql to create the Backup Statements run isql to execute the Backup Statements record the results of the Backup Statements in Backup Files copy the Backup Files to the Backup Subdirectory create the Backup Summary "greps" successful Dump statements along with any errors generated, sends e-mail to the DBA and writes them to the backup_summary file
sp	SQL Server password file - contains password for backup role

No intervention in the Automatic Backup Process is required by the DBA, though periodic checks of the Backup Subdirectories are recommended.

4.10.2 Manual Backups

Manual backups can be performed at any time by the System Administrator and are recommended for the following situations:

1. Any change to the **master** database - this includes new logins, devices, and databases
2. Any major change to user databases - a large ingest or deletion of data, definition of indexes
3. Other mission-critical activities - as defined by the DAAC Operations Supervisor.

Both the **dump database** and **dump transaction** command processing are off-loaded to the backup server, and will not affect normal operations of the database. These commands are performed by the System Administrator on appropriate databases as follows:

Syntax:

```
1> dump database master to
"/usr/ecs/OPS/COTS/sybase/sybase_dumps/dumps/dbname.dat.MMDDHHMM."

2> go
```

After dumping the database, compress the dump file by executing:

%compress /usr/ecs/OPS/COTS/sybase/sybase_dumps/dumps/dbname.dat.MMDDHHMM.

Syntax:

```
1> dump transaction pdps_db_ops to
   "/usr/ecs/OPS/COTS/sybase/sybase_dumps/trans/pdps_OPS.tran.YYMMDDH
   HMM""
2> go
```

4.10.3 Manual Recovery

Manual recovery of a user database is performed by the System Administrator by the use of the **load database** and **load transaction** commands. For issues concerning the **master** database, please consult your System Administrator's Guide for assistance. It is recommended that any user database to be recovered be dropped and created with the **for load** option., The **databasename**.sql along with any **alter.databasename**.sql scripts can be , combined into one script which will re-create the user database with the **for load** option. This will insure the success of the **load database** and **load transaction** commands.

4.10.4 The BulkCopy Utility

The **bcp** utility is located in the **$SYBASE**/bin directory and is designed to copy data to and from SQL Server databases to operating system files.

4.10.4.1 Requirements for Using bcp

In general, you must supply the following information for transferring data to and from SQL Server:

a. Name of the database and table

b. Name of the operating system file

c. Direction of the transfer (in or out)

In order to use **bcp**, you must have a SQL Server account and the appropriate permissions on the database tables and operating system files that you will use. To copy data **in**to a table, you must have **insert** permission on that table. To copy data **out** to an operating system file, you must have select permission on the following tables:

a. The table being copied

b. sysobjects

c. syscolumns

d. sysindexes

bcp Syntax

bcp [[database_name].owner.]table_name {in | out} datafile [-e errfile] [-n] [-c]

[-t field_terminator] [-r row_terminator] [-U username] [-S server]

4.10.4.2 Example of User Database Recovery

The database **UserDB** was created using the following script excerpt: (stored in home/scripts/create.databases/userdb.sql)

create database UserDB on data_dev1 = 100 log on tx_log1 = 50 [with override]

and was modified using the following script excerpt: (home/scripts/create.databases/alteruserdb.sql)

Alter database **UserDB** on data_dev1=50

For the purposes of this example, the full database backup and transaction log dumps were successful and located in /usr/ecs/OPS/COTS/UserDB.dat and UserDB_tx.dat

1 . In the **$SYBASE**/scripts/create.databases directory, DBA makes a script file from the template.

Syntax: % cd /usr/ecs/OPS/COTS/sybase/scripts/create.databases

% cp template.sql userdb_for_load.sql

2 . Appropriate items are modified so that the script file resembles the following:

1> create database UserDB on data_dev2=100 log on tx_log2=50 **for load**

2> go

3> alter database UserDB on data_dev3=50

4> go

3 . DBA saves the script in $SYBASE/scripts/create.databases/userdb_for_load.sql

4 DBA runs the script from the UNIX command prompt.

Syntax: %isql -Usa -S**servername** -iuserdb_for_load.sql -ouserdb_for_load.out

5 DBA checks the userdb_for_load.out file for success

6 DBA loads the database from the full backup.

Syntax: 1> load database UserDB from

2> "/usr/ecs/OPS/COTS/sybase/sybase_dumps/week1/dbname.dat.MMDD HHMM"

3> go

7 DBA loads the transaction file from the transaction file dump.

Syntax: 1> load transaction UserDB from

2> "/usr/ecs/OPS/COTS/sybase/sybase_dumps/week1/dbname.tran.MMDDHHMM"

3> go

4.11 Database Performance and Tuning

Once your application is up and running, the DBA monitors its performance, and may want to customize and fine-tune it. Use the following software tools provided by SQL Server:

a. Setting query processing options with the **set** command

b. Setting database options with **sp_dboption**

c. Monitoring SQL Server activity with **sp_monitor**

d. Using **update statistics** to ensure that SQL Server makes the best use of existing indexes

e. Changing system variables using **sp_configure** and the **reconfigure** command

f. Placing objects on segments to spread i/o, improve throughput, etc. as described in Section 4.4.4

For a complete discussion of issues related to SQL Server performance and tuning, refer to your SYBASE SQL Server Performance and Tuning.

4.12 Installation of the Applications

DBA should have physical devices configured before installing either autosys or remedy. Both applications use Sybase as their database.

4.12.1 Installation of the Application Database

The installation of the application databases has been automated using ECS Assistant. The application databases are created using the DbBuild script which can only be invoked through ECS Assistant or the Command Line. Scripts that ECS Assistant invokes are:

DbBuild - Create new empty database and loads with initial data

DbPatch - Upgrade to new schema while retaining existing data.

4.12.2 The AUTOSYS Application and other Configuration Issues

The AUTOSYS application works in tandem with PDPS/DPSs to schedule the jobs that run on Science Processor. Autosys installation is performed in /usr/ecs/OPS/COTS by the auto install program located in the autosys/install directory. The results of the installation are stored in an

autosys_install.scr file located in the AUTOSYS home directory (/use/ecs/OPS/COTS/autosys). For pdps to run properly with AUTOSYS, the following activities are completed:

a. A user is defined named **autosys**

b. **autosys** user is added to the pdps database (OPS mode)

c. The autosys server is added to the sysservers table with **sp_addserver**

d. The server is added to the sysservers table on the AUTOSYS server with **sp_addserver**

4.12.3 Spatial Query Server (SQS)

SQS is a multi-threaded, Sybase Open Query database engine, which is required by the Science Data Subsystem (SDSRV). This product allows definition of spatial data types, spatial operators, and spatial indexing. SQS communicates with Sybase SQL Server to process SDSRV requests to push and pull metadata. SDSRV database server resides on an SGI machine. SQS also, reside on the same machine as SDSRV Sybase SQL Server.

- Named X1acg01 - where X is the DAAC specific identifying character.
- pathname - /usr/ecs/OPS/COTS/sqs222/bin/sqsserver
- Should have one dedicated CPU per instance running. Defaults to one instance now, but may require additional instances later for performance reasons.
- Requires one entry in the Sybase "interfaces" file per instance of the SQS server to be run.
- Consult startup scripts in /etc/init.d/sybase and /etc/init.d/sqs_222

SQS requires a Sybase login with SA or sa_role and associated password to start. SQS environment variables requirements:

- SYBASE = Location of the Sybase home directory. Example: /tools/sybOCv(TBD)
- PATH = Must include in this order - /usr/bin; /usr/sbin;$SYBASE/bin
- DSQUERY = Name of SQL Server to which to connect. From the $SYBASE/intefaces file. Examples - g0acg01 _srvr
- DSLISTEN = Name of SQS server to use. Example - g0acg01_ srvr
- SQSUSER = Name of the user (SA or sa_role) for system connection.
- SQSPASSWORD = Password for the system connection login

The SQS startup script requires the following information:

SQSHOME = location of sqsserver binaries.

The following is a list of options that can be imbedded in the startup script, these options are beneficial, but they are not required.

4.12.3.1 SQS STARTUP OPTIONS:

- -e path of the SQS server logfile. Example /usr/ecs/OPS/COTS/sqs222/sqs/bin/sqs_222.log
- -u number of concurrent SQS connections. Recommend minimum of 125. Example -u 125
- Usually started with a delay, after the SQL Server is started. This delay be sufficient for the SQL server to recover and come-up.
- $SQSHOME/bin/sqsserver -e $SQSHOME/sqs_222.log -u $USER &

SQS has dependencies on Sybase, such as:

- Sybase must be running prior to starting SQS
- SQS user id that starts SQS, which is different from the application user ID must have admin privileges
- SQS opens a connection to Sybase's because it writes to the Sybase System tables
- SQS server thread runs under the userid sa. In order to avoid confusion when monitoring this thread, it is best to:
- create a separate login and userid specifically to monitor SQS
- grant sa_role authority to the userid created to monitor SQS

EXAMPLE: 1> **sp_adduser** sqs_mon

2> **grant** sa_role to sqs_mon

3> go

4.13 Passwords Security

Security has become a sensitive issue throughout the IT Industry. The ECS program is also concern about security and the risks associated with security. As a result the following directive is issued to all DAACs.

All System Administrators and Database Administrators at the sites are responsible for easonable security measures when installing ECS custom software. This means:

1. Changing the permissions of online secure files to the minimum level required .
2. Backing up secure file(s) to removable media (floppy or tape) and removal of secure files immediately after installation is complete.
3. The media should then be kept in a secure location.

The following file is affect as result of this requirement on the ECS program.

A. /usr/ecs/<MODE>/CUSTOM/dbms/<SUBSYSTEM>/Ec<server>SybaseLogins.sql

B. Set permissions to 711 (user read, write, execute, group and other read only)

Figures 4.13-1 and 4.13-2 are the Technical Directives issued by the Director of Systems Engineering.

Raytheon Systems

ECS Project

Engineering Technical Directive DRAFT

No. 98-XXX

Subject: Ftp account password distribution

January 15, 1999

The following directive is issued to all DAACS and labs.

Issue: **Ftp accounts are required for ECS custom software to function. Accounts and passwords have been distributed together via public means (such as electronic mail) that presents an unacceptable security risk.**

Fix: **When communicating sensitive information such as accounts and passwords, do not put them together in the same document.**

Testing: **N/A**

Implementation: For distribution of account information when it is not hand carried:

1. Send the account information in one document.
2. Send the password information in a different document in inverted order (the first password entry is matched to the last account entry).

NOTE: It is not required but it is recommended to encrypt ftp account information using the UNIX crypt command when sending via electronic means. If crypt is used, the password used should be contained in a third document.

Point of Contact:

Approved By: M. McBride
Director, Systems Engineering

Reference CCR: 98-XXXX

--------**End of Directive**------

Figure 4.13-1. Technical Directive

The following directive is issued to all DAACS and labs.

Issue: System verification testing has revealed that passwords are being written in clear text to log files.

Implementation:

1. This practice presents an unacceptable security risk to ECS.
2. Names of system accounts may be written to logs. However, logs that are so written must have file permissions on the log file changed to 770 with the cm<MODE> account being the owner and cm<MODE> group being the group. If logs are retained on the host, they must be encrypted using the UNIX crypt command and a locally agreed upon password. One such password per location is acceptable but the password may not be kept in clear text on any system.

Point of Contact: Byron Peters, Chair, TSIWG, email – bpeters@eos.hitc.com, phone 301/883-4077

Approved By: M. McBride
Director, Systems Engineering

Reference CCR: 98-XXXX

--------**End of Directive**-------

Figure 4.13-2. Technical Directive

This page intentionally left blank.

5. Security Services

ECS security architecture must meet the requirements for data integrity, availability, and confidentiality. ECS Security Services meets these requirements by incorporating a variety of mechanisms to establish and verify user accounts, issue and verify passwords, audit user activity, and verify and protect data transfer. Security logs will be monitored and security reports generated by the System Administrator as required. Several FREEWARE products provide tools for authentication and network and system monitoring: SATAN, Crack, anlpasswd, TCP Wrappers, Secure Shell, and Tripwire. The Open Software Foundation's Distributed Computing Environment (OSF/DCE) employs Kerberos for authenticating user requests for network services. (DCE administration tools a are discussed in Section 3 of this document.) The FREEWARE product, SATAN, monitors networks and finds system security vulnerabilities. Three FREEWARE products — Crack, anlpasswd and Secure Shell — provide additional password checking for local system and network access. To monitor and limit access to network services, ECS Security Services use TCP Wrappers. Tripwire monitors for intruders changes to files by detecting and flagging any changes. Security Services also supports detection of, reporting, and recovery from security breaches.

The following section defines step-by-step procedures for M&O personnel to run the Security Services tools. The procedures assume that the requester's application for a Security process has already been approved by DAAC Management. It is recommended that access to these tools be controlled through the **root access only**.

5.1 Monitoring Network Vulnerabilities

The Security Administrator Tool for Analyzing Networks (SATAN) is a testing and reporting tool that collects a variety of information about networked hosts. SATAN gathers information about specified hosts and networks by examining network services (for example, finger, NFS, NIS, ftp). SATAN also gathers general network information (network topology, network services run, types of hardware and software being used on the network). The data is used to point out system vulnerabilities. Data can be reported in a summary format. Problems are described briefly and pointers provided to patches or workarounds.

Periodically, the System Administrator will run SATAN as **root**. The procedures are provided below.

1 Make sure that **DISPLAY** is set on your workstation.

2 From **/usr/local/solaris/satan-1.1.1** type **./satan**.

3 From the **SATAN Control Panel**, select **SATAN Configuration Management**. Set all variables or use the default values.

4 Go back to the SATAN Control Panel.

5 From the **SATAN Control Panel**, select **SATAN Data Management**. Create the SATAN database if it does not exist. When you create the database for the first time, you will see a warning message concerning password disclosures. Take no action and continue. The database is stored as **satan-data** in the directory

/satan-1.1.1/results.

6 You will be notified when the SATAN finishes creating the database and scans the system (network or cluster) for vulnerabilities.

7 From this screen, you can click on "Continue with Reporting and Analysis" or you can return to the **SATAN Control Panel**, to make this selection. Select the reports that you want to review.

5.2 Ensuring Password Integrity

One aspect of system security is discretionary access control based on user passwords. Passwords should be so unique that they are virtually impenetrable to unauthorized users. Two products provide utilities to create effective password practices. "Crack" detects weak passwords that could be easily bypassed in a "batch" mode. Anlpassword enforces strong password rules as the user is changing their password.

Crack and anlpasswd provide comprehensive dictionary, which can shared. These "source" dictionaries provide lists of words, which if used, would create vulnerable passwords. You can add other dictionaries, for example, acronym lists, to eliminate commonly used terms from being used as passwords.

Both products are installed in a secure location, that has, **root access only**. Such precautions are particularly apt when running Crack, which gives the administrator access to everyone's password that he/she penetrates.

5.2.1 Detecting Weak Passwords

Running Crack against a system’s password file will enable a system administrator to assess how vulnerable the file is to unauthorized users and how well authorized users select secure passwords. Crack is designed to find standard Unix eight-character DES-encrypted passwords by standard guessing techniques.

Crack takes as its input a series of password files and source dictionaries. It merges the dictionaries, turns the password files into a sorted list, and generates lists of possible passwords from the merged dictionary or from information gleaned about users from the password file. It does not attempt to remedy the problem of allowing users to have guessable passwords, and it should NOT be used in place of getting a really good, secure password program replacement.

The instructions provided in the following sections are general in nature because how you configure Crack and how you run it depends on the platforms on which it resides and on the local security requirements established for your site. M&O personnel should be familiar with these tasks to:

1. Configure the Crack shellscript and config.h files based on the README file and on requirements established for your site. See Section “Configuring Crack” below.
2. Run Crack based on requirements established for your site. See “Running Crack”, below.
3. Customize the dictionaries. See Section “Creating Dictionaries”, below.

5.2.1.1 Configuring Crack

Although Crack should already be configured for your system, the instructions are provided should you have to reconstruct the makefile as a result of file corruption. Crack has two configuration files: the Crack shellscript, which contains all the installation-specific configuration data, and the file Sources/conf.h, which contains configuration options specific to various binary platforms.

1 In the Crack shellscript, edit the CRACK_HOME variable to the correct value. This variable should be set to an absolute path name through which the directory containing Crack may be accessed on ALL machines on which Crack will be run. (Path names relative to username are acceptable as long as you have some sort of csh.)

There is a similar variable, CRACK_OUT, which specifies where Crack should put its output files — by default, this is the same as $CRACK_HOME.

2 Edit the file Sources/conf.h and establish which switches to enable. Each #define has a small note explaining its purpose. Portability of certain library functions, should not be a problem.

3 If using Crack -network (see Section Options, below), generate a Scripts/ **network.conf** file. This file contains a list of hostnames to rsh to, what their binary type is (useful when running a network Crack on several different architectures), an estimate of their relative power (take your slowest machine as unary, and measure all others relative to it), and a list of per-host flags to add to those specified on the Crack command line, when calling that host. There is an example of such a file provided in the Scripts directory.

4 To specify a more precise figure as to the relative power of your machines, play with the **command make** tests in the source code directory. This can provide you with the number of fcrypt()s that your machine can do per second, which is a number that you can plug into your **network.conf** as a measure of your machines' power (after rounding the value to an integer).

5.2.1.2 Running Crack

Crack is a self-installing program. Once the necessary configuration options for the Crack shellscript and config.h have been set, the executables are created via **make** by running the Crack shellscript.

Notes for Yellow Pages (NIS) Users:

To get Crack running from a YP password file, the simplest way is to generate a passwd format file by running:-

ypcat passwd > passwd.yp

and then running Crack on this file.

To launch Crack:

1 From /usr/local/solaris **/crack**, at the command line type: **./Crack**

2 For the single platform version:

./Crack [options] [bindir] /etc/passwd [...other passwd files]

3 For the network version:

./Crack -network [options] /etc/passwd [...other passwd files]

For a brief overview of the [options] available, see Section "Options", below. Section "Crack Support Scripts" briefly describes several very useful scripts.

5.2.1.3 Creating Dictionaries

Crack works by making many individual passes over the password entries that you supply to it. Each pass generates password guesses based upon a sequence of rules, supplied to the program by the user. The rules are specified in a simplistic language in the files gecos.rules and dicts.rules, located in the Scripts directory (see Section "Crack Support Scripts", below).

Rules in Scripts/gecos.rules are applied to data generated by Crack from the pw_gecos and pw_gecos entries of the user's password entry. The entire set of rules in gecos.rules is applied to each of these words, which creates many more permutations and combinations, all of which are tested. After a pass has been made over the data based on gecos information, Crack makes further passes over the password data using successive rules from the Scripts/dicts.rules by loading the whole of Dicts/bigdict file into memory, with the rule being applied to each word from that file. This generates a resident dictionary, which is sorted and uniqued so as to prevent wasting time on repetition. After each pass is completed, the memory used by the resident dictionary is freed up, and re-used when the next dictionary is loaded.

Crack creates the Dicts/bigdict dictionary by merging, sorting, and uniq'ing the source dictionaries, which are to be found in the directory DictSrc and which may also be named in the Crack shellscript, via the $STDDICT variable. (The default value of $STDDICT is /usr/dict/words.)

The file DictSrc/bad_pws.dat is a dictionary which is meant to provide many of those common but non-dictionary passwords, such as 12345678 or qwerty.

To create your own dictionary:

1 Copy your dictionary into the DictSrc directory (use compress on it if you wish to save space; Crack will unpack it while generating the big dictionary).

2 Delete the contents of the Dicts directory by running Scripts/spotless. Your new dictionary will be merged in on the next run.

5.2.1.4 Options

-f Runs Crack in foreground mode, i.e., the password cracker is not backgrounded, and messages appear on stdout and stderr as you would expect. This option is only really useful for very small password files, or when you want to put a wrapper script around Crack.

Foreground mode is disabled if you try running Crack-network -f on the command line, because of the insensibility of rshing to several machines in turn, waiting for each one to finish before calling the next. For more information, read the section about Network Cracking without NFS/RFS in the README.NETWORK file.

-v Sets verbose mode, whereby Crack will print every guess it is trying on a per-user basis. This is a very quick way of flooding your filestore, but useful if you think something is going wrong.

-m Sends mail to any user whose password you crack by invoking Scripts/nastygram with their username as an argument. The reason for using the script is so that a degree of flexibility in the format of the mail message is supplied; i.e., you don't have to recompile code in order to change the message.

-nvalue Sets the process to be nice()ed to value, so, for example, the switch -n19 sets the Crack process to run at the lowest priority.

-network Throws Crack into network mode, in which it reads the Scripts/network.conf file, splits its input into chunks which are sized according to the power of the target machine, and calls rsh to run Crack on that machine. Options for Crack running on the target machine may be supplied on the command line (for example, verbose or recover mode), or in the network.conf file if they pertain to specific hosts (e.g., nice() values).

-r<pointfile>

This is only for use when running in recover mode. When a running Crack starts pass 2, it periodically saves its state in a pointfile, with a name of the form Runtime/P.* This file can be used to recover where you were should a host crash. Simply invoke Crack in exactly the same manner as the last time, with the addition of the **-r** switch (for example, **-rRuntime/Pfred12345**). Crack will startup and read the file, and jump to roughly where it left off. If you are cracking a very large password file, this can save a lot of time after a crash.

5.2.1.5 Crack Support Scripts

The Scripts directory contains a small number of support and utility scripts, some of which are designed to help Crack users check their progress. The most useful scripts are briefly described below.

Scripts/shadmrg

This is a small script for merging /etc/passwd and /etc/shadow on System V style shadow password systems. It produces the merged data to stdout, and will need redirecting into a file before Crack can work on it.

Scripts/plaster

This is a simple frontend to the Runtime/D* diefiles that each copy of the password cracker generates. Invoking Scripts/plaster will kill off all copies of the password cracker you are running, over the network or otherwise. Diefiles contain debugging information about the job, and are generated so that all the jobs on the entire network can be called quickly by invoking Scripts/plaster. Diefiles delete themselves after they have been run.

Scripts/status

This script rshes to each machine mentioned in the Scripts/network.conf file, and provides some information about processes and uptime on that machine. This is useful when you want to find out just how well your password crackers are getting on during a Crack -network.

Scripts/{clean,spotless}

These are really just frontends to a makefile. Invoking Scripts/clean tidies up the Crack home directory, and removes probably unwanted files, but leaves the pre-processed dictionary bigdict intact. Scripts/spotless does the same as Scripts/clean but obliterates bigdict and old output files, too, and compresses the feedback files into one.

Scripts/nastygram

This is the shellscript that is invoked by the password cracker to send mail to users who have guessable passwords, if the **-m** option is used. Edit it to suit your system.

Scripts/guess2fbk

This script takes your out* files as arguments and reformats the 'Guessed' lines into a slightly messy feedback file, suitable for storing with the others.

An occasion where this might be useful is when your cracker has guessed many peoples' passwords, and then died for some reason (a crash?) before writing out the guesses to a feedback file. Running Scripts/guess2fbk out* >> Runtime/F.new will save the work that has been done.

5.2.1.6 Checking the Log

Crack loads dictionaries directly into memory, sorts and uniques them, before attempting to use each of the words as a guess for each users' password. If Crack correctly guesses a password, it marks the user as done and does not waste further time on trying to break that user's password.

Once Crack has finished a dictionary pass, it sweeps the list of users looking for the passwords it has cracked. It stores the cracked passwords in both plain text and encrypted forms in a feedback file in the directory **Runtime**. Feedback files have names of the form **Runtime/F***. The purpose of this is so that when it is next invoked, Crack can recognize passwords that it has successfully cracked previously, and filter them from the input to the password cracker. This provides an instant list of crackable users who have not changed their passwords since the last time Crack was run. This list appears in a file with name **out*** in the **$CRACK_OUT** directory, or on **stdout**, if foreground mode (**-f**) is invoked (see Section “Options”, above).

Similarly, when a Crack run terminates normally, it writes out to the feedback file all encrypted passwords that it has NOT succeeded in cracking. Crack will then ignore all of these passwords next time you run it.

Obviously, this is not desirable if you frequently change your dictionaries or rules, and so there is a script provided, **Scripts/mrgfbk**, which sorts your feedback files, merges them into one, and optionally removes all traces of "uncrackable" passwords, so that your next Crack run can have a go at passwords it has not succeeded in breaking before.

mrgfbk is invoked automatically if you run **Scripts/spotless** (see Section “Crack Support Scripts”, above).

5.2.2 Configuring AnlPasswd

anlpasswd was written by Argonne National Laboratory. There is no install script and installation is by hand. anlpasswd consists of a setuid C program that is used to call the anlpasswd Perl script. The Perl script uses several standard include files that come with Perl and several additional files that are included with anlpasswd. Additionally, a dictionary file is used to match attempted passwords against possible bad passwords that are in the dictionary file.

It is assumed that Perl 5.003 is properly installed in /tools/bin/perl5 for each platform that anlpasswd is to be used on. The binary ypstuff most likely be placed in a NFS shared directory (/tools/bin). The actual Perl program that does the work should be placed in /usr/local/anlpasswd

and chmod to 600. This can't be placed in a NFS directory since /tools/bin/ usually isn't "root equivalent" on all machines and this script should be set to root read only.The Perl includes and dictonary file should also be NFS mounted and placed in /tools/lib/anlpasswd.

The remainder of the installation will need to be completed on each individual machine (moving passwd and yppasswd, creating a SUID program, creating the links, etc.)

The final directory structure should be:

/tools/bin/perl5

/usr/local/anlpasswd/anlpasswd (chmod 600)

passwd link to anlpasswd

yppasswd link to anlpasswd

/usr/bin/passwd.orig original passwd program (chmod 644)

yppasswd.orig original yppasswd program (chmod 644)

passwd suidwrap program (chmod 4111)

yppasswd link to /usr/bin/passwd

/tools/lib/anlpasswd/badpats this directory is the same on all

bigdict machines, text files only

encrypt_passwd

im_prompt2.pl

/tools/bin/ypstuff this binary is arch dependent and is located in {source install dir}/bin.ARCH/ypstuff

The following should be done on each machine (as root):

This assumes that the /tools/bin and /tools/lib directories are already setup as directed above.

```
cd /usr/bin
mv passwd passwd.orig
chmod 644 passwd.orig
mv yppasswd yppasswd.orig
chmod 644 yppasswd.orig
cd /usr/local
mkdir anlpasswd
cd anlpasswd
cp {source install dir}/anlpasswd .
chmod 600 anlpasswd
ln -s anlpasswd passwd
ln -s anlpasswd yppasswd
cd /usr/bin
cp {source install dir}/bin.{ARCH}/suidwrap passwd
```

```
chmod 4111 passwd

ln -s passwd yppasswd
```

Below is the original installation instructions:

5.2.2.1 Installing anlpasswd

1. Copy and modify the anlpasswd Perl script. This is located in the "anlpasswd/perl" directory of the distribution, and is called logically enough) "anlpasswd". The configuration section of the

code is located near the beginning of the script, and is labeled "Configs". Here are the lines you need to be concerned with changing:

```
@legal_shells = ('/bin/sh, /bin/csh');
```

This is an array containing the valid shells available on your system. Note that this may not necessarily be the same as the information listed in /etc/shells;

for example, on some machines, /bin/csh and /bin/sh don't have to be listed in /etc/shells. This is not the case with anlpasswd; you should set @legal_shells to contain a list of all valid login shells. If you add new login shells to your systems, you need to update this array.

It is very important that the shells listed here are available on all of the machines on your YP network; otherwise, a user may change his/her shell to one that doesn't exist on one of your machines, and therefore will be unable to log in to that machine.

```
unshift(@INC, "/tools/lib/anlpasswd");
```

These are additional locations for Perl to look for the include files that came with the anlpasswd distribution. These are currently set for our local configuration. If you decide to put the Perl libraries in this distribution (im_prompt2.pl, encrypt_passwd) in locations other than the main Perl include directory, you should add those locations to the include file search path (@INC) as shown above.

Otherwise, you can just delete these lines.

```
# $bigdict = large list of words

$dictdir = "/tools/lib/anlpasswd";    # location of dictionaries

$bigdict = "$dictdir/bigdict";        # large list of words

$ypstuffdir = "/tools/bin";           # location of ypstuff executable

$BADPATS = "$dictdir/badpats";        # location of added bad patterns
```

These are the locations of other files that anlpasswd needs. Change these to reflect the location of your dictionaries, the location of the main dictionary (bigdict.sorted, in this case), and the location of the "ypstuff" program. Again, if you're using this package over a networked system, the

dictionaries must be located on some filesystem cross-mounted on all of your machines; otherwise this won't work.

There is a badpats file in the perl directory to use as an example.

Again location ofthe file is up to your customization, but we put it in with the big dictionary.

Finally, edit the definition of the @dictlist array to contain the names of the additional dictionaries (if any) you want to use.

2. Decide where you want to put the anlpasswd script, and copy it there. If you are installing this on a network of machines, this location must be on a filesystem cross-mounted on all machines.

Since Perl disallows running setuid Perl scripts, the anlpasswd script is not executed directly. Instead, a setuid C wrapper is used to call the anlpasswd script. The Perl script should not be

executable or setuid, and should not be located in anyone's path. You should probably "chmod 600 anlpasswd" to be safe.

3. Make a link to anlpasswd called "passwd", i.e.

ln -s anlpasswd passwd

If you are using YP, make another link for yppasswd:

ln -s anlpasswd yppasswd.

4. Decide where you want the passwd executable to reside. This is a C wrapper running setuid to root which calls the anlpasswd script. This will probably be in /bin or /usr/local/bin.

You should keep a copy of the original passwd program around somewhere, in case something breaks, but it should not be executable. A good idea would be to do the following:

cd /usr/bin (or wherever the original passwd program was kept)

mv passwd passwd.orig

chmod 644 passwd.orig

5. Modify anlpasswd/c-routines/suidwrap.c to suit your local configuration. The PASSWD_ACTUAL constant contains the location of the "passwd" link to anlpasswd. Similarly

the YPPASSWD_ACTUAL constant contains the location of the "yppasswd" link. You should change these to reflect the locations you chose in steps 2 and 3. The current settings assume the

yppasswd and passwd executables can be located in /bin or /usr/local/bin. You should modify these to reflect the location(s) you chose in step 4.

6. Run "make" in anlpasswd.ARCH/c-routines. This will compile the suid wrapper and the ypstuff executable. In the sgi directory, manally run the lines in the makefile that are commented out.

Copy "suidwrap" to the location you chose in step 4, and rename this copy as "passwd". (Be sure you saved your original passwd program in a safe place). Change this to be executable by all users, and set it to run suid to root (you must be logged in as root to do this). Make a link to this file called "yppasswd".

i.e., if passwd_exec_dir is the location you chose in step 4,

```
cp suidwrap passwd_exec_dir/passwd
  cd passwd_exec_dir
  chmod 4111 passwd
ln -s passwd yppasswd
```

Unless the location you have chosen for the passwd executable is on a partition cross-mounted on all machines, you will have to repeat this procedure on every machine (or architecture) on your network. (Don't confuse the executable C wrapper with the Perl anlpasswd script; there should only be one copy of the Perl script, on a partition accessible by all the machines on your network.)

7. Copy anlpasswd.ARCH/c-routines/ypstuff to the location you chose for it in step 1.

8. Put the large dictionary file in the location you chose in step 1. There is a C program and instructinos to do this in dictionary-create.

That should be all that is needed to get this program up and runnning. If there are any problems or inaccuracies in this documentation, or have any improvements or bug fixes, please send email to "support@mcs.anl.gov"

5.3 Enforcing Strong Passwords (Secure Shell)

The security risks involved in using “R” commands such as rlogin, rsh, rexec and rcp are well known but their ease of use has made their use tempting in all but the most secure of environments. Ssh is an easy-to-use, drop in replacement for these commands developed by Tatu Ylonen. Ssh is a “user” level application. No changes to the host kernel are required. The UNIX server is available as freeware on the Internet but ECS will have a support agreement in place to minimize risk to the program in the event of a product defect.

There are four programs that users use that should be put in /tools/bin for each architecture:

- ssh - replaces rsh, rlogin and rexec for interactive sessions
- scp - replaces rcp for interactive file transfer
- ssh-add - add access to a specific ssh host
- ssh-keygen - generates keys for the local host based on a passcode (long password)

There are three system programs that should be local to each system:

- sshd - the ssh daemon started in /etc/rc2.d NOT in inetd.conf
- ssh-agent - the key management agent started in /etc/rc2.d
- make-ssh-known-hosts - perl script to generate keys for known ssh systems

Several files are generated on installation and when running and are installed locally:

- /etc/ssh_config - system-wide configuration for the ssh client
- /etc/ssh_host_key - contains the long number used for one of the keys
- /etc/ssh_host_key.pub - contains the key known to the public
- /etc/ssh_random_seed - base number used in generating keys
- /etc/sshd.pid - the process number of the sshd currently running
- /etc/sshd_config - defines the local security policy

The amount of disk space that the programs and the configurations files takes is less than 6 MB. NOTE: To compile the source requires about 30MB of disk space.

5.3.1 The SSH Encryption Mechanism[1]

Each host has a host-specific RSA key (normally 1024 bits) used to identify the host. Additionally, when the daemon starts, it generates a server RSA key (normally 768 bits). This key is normally regenerated every hour if it has been used, and is never stored on disk.

Whenever a client connects the daemon, the daemon sends its host and server public keys to the client. The client compares the host key against its own database to verify that it has not changed. The client then generates a 256 bit random number. It encrypts this random number using both the host key and the server key, and sends the encrypted number to the server. Both sides then start to use this random number as a session key which is used to encrypt all further communications in the session. The rest of the session is encrypted using a conventional cipher. Currently, IDEA, DES, 3DES, ARCFOUR, and TSS (a fast home-grown algorithm) are supported. IDEA is used by default. The client selects the encryption algorithm to use from those offered by the server.

Next, the server and the client enter an authentication dialog. The client tries to authenticate itself using .rhosts authentication, .rhosts authentication combined with RSA host authentication, RSA challenge- response authentication, TIS challenge response authentication, or password based authentication.

---

[1] From the *sshd* man page

Rhosts authentication is normally disabled because it is fundamentally insecure, but can be enabled in the server configuration file if desired. System security is not improved unless rshd(8), rlogind(8), rexecd(8), and rexd (8) are disabled (thus completely disabling rlogin(1) and rsh(1) into that machine).

If the client successfully authenticates itself, a dialog for preparing the session is entered. At this time the client may request things like allocating a pseudo-tty, forwarding X11 connections, forwarding TCP/IP connections, or forwarding the authentication agent connection over the secure channel.

5.3.2 How a User uses Secure Shell

(TBD)

5.3.3 Configuration of Secure Shell

A user can set up one's account manually or by using a script called sshsetup. The only thing you need to know before executing the script or running these commands manually is picking a good

passphrase (since you can and *should* use spaces and multiple words with numbers and misspellings and special characters). Note that passwords are NOT echoed back to the screen. PLEASE DO NOT USE THE PASSWORDS/PASSPHRASES USED HERE OR IN ANY OTHER DOCUMENTATION!

Using the script sshsetup should look like:

```
% sshsetup
```

Use a passphrase of at least 10 characters which should include numbers or special characters and MAY include spaces

```
Initializing random number generator...

Generating p:  .++ (distance 6)

Generating q:  ........++ (distance 110)

Computing the keys...

Testing the keys...

Key generation complete.

Enter file in which to save the key ($HOME/.ssh/identity): [RETURN]

Enter passphrase (empty for no passphrase): litt1e 1amp jumb3d

Enter same passphrase again: litt1e 1amp jumb3d

Your identification has been saved in /home/JohnDoe/.ssh/identity.

Your public key is:
```

1024 37 [lots of numbers] JohnDoe@theagency.nasa.gov

Your public key has been saved in /home/JohnDoe/.ssh/identity.pub

And now you are ready!

NOTE: If you have accounts in the miniDAAC, EDF and the VATC, do sshsetup in EACH environment.

To do the same thing manually requires the following commands:

% ssh-keygen

Initializing random number generator...

Generating p: .++ (distance 6)

Generating q: .....++ (distance 110)

Computing the keys...

Testing the keys...

Key generation complete.

Enter file in which to save the key ($HOME/.ssh/identity): [RETURN]

Enter passphrase (empty for no passphrase): litt1e 1amp jumb3d

Enter same passphrase again: litt1e 1amp jumb3d

Your identification has been saved in /home/JohnDoe/.ssh/identity.

Your public key is:

1024 37 [lots of numbers] JohnDoe@theagency.nasa.gov

Your public key has been saved in /home/JohnDoe/.ssh/identity.pub

% cd ~/.ssh

% cp identity.pub authorized_keys

% cd

% chmod go-w . .ssh .ssh/authorized_keys

Note that the last four commands *are* automatically done in the sshsetup script.

** Changing your passphrase **

To change your passphrase, use the following command:

% ssh-keygen -p

Enter file in which the key is ($HOME/.ssh/identity): [RETURN]

Enter old passphrase: litt1e 1amp jumb3d

Key has comment 'JohnDoe@theagency.nasa.gov'

Enter new passphrase (empty for no passphrase): br0wn cow 3ats grass

Enter same passphrase again: br0wn cow 3ats grass

Your identification has been saved with the new passphrase.

** Using ssh to login to hosts **

*** The simplest way ***

To login, use the command:

% slogin defiant

Enter passphrase for RSA key 'JohnDoe@nevermor': br0wn cow 3ats grass

Last login: Sun Feb 22 06:50:59 1998 from echuser.east.hitc.com

No mail.

%

NOTE: The first time you login to a host the following message will pop up asking if you want to continue. In response, type yes and [enter]:

Host key not found from the list of known hosts.

Are you sure you want to continue connecting (yes/no)? yes

Host 't1acg01' added to the list of known hosts.

To transfer a file, use the command:

% scp hostone:/etc/info info

Enter passphrase for RSA key 'bpeters@nevermor': br0wn cow 3ats grass

stty: : Function not implemented

This will copy the file /etc/info from hostone to your local host. Note that your passphrase is needed to initiate the transfer. The "stty: : function not implemented" is a nuisance information message but does not impact the transfer.

IMPORTANT NOTE: The default directory on the *target* host is always the users HOME directory.

Also, one may send/receive file recursively using "-r" such as:

% scp -r ~/files/* hostone:~/files

will send what is in the home directory files subdirectory to the target host hostone in the home files directory.

To execute a command remotely, use the command:

```
% ssh whoisonfirst "ps -ef"

Enter passphrase for RSA key 'bpeters@nevermor':  br0wn cow 3ats grass
```

*** A layer of convienence ***

If you are already a user of "r" commands, you probably know about the .rhost file. Ssh will allow a user to setup the .rhost equivalent called .shost in one's home directory. .Rhost and .shost contain the names of the hosts to which one normally connects. The nice thing about using it is one need not enter one's passphrase. Unlike "r" commands, however, ssh commands use long strings of numbers to gain access which makes it quite difficult for an intruder to impersonate a legitimate user. One word of caution, however. If you leave your terminal while logged on, a passerby could logon to any host in your .rhost/.shosts file and potentially cause malicious damage to you and your colleagues work. Be aware!

NOTE: ssh checks the mode of .shost, so change permission on .shost by typing:

```
% chmod 600 /home/JohnDoe/.shost
```

where you must substitute your own home directory for /home/JohnDoe.

*** Multiple connections ***

If you open multiple connections, it is more convienient to keep your keys in system memory. To do this requires executing two commands:

```
% ssh-agent tcsh

% ssh-add

Need passphrase for /home/JohnDoe/.ssh/identity (bpeters@nevermor).

Enter passphrase: br0wn cow 3ats grass

Identity added: /home/JohnDoe/.ssh/identity (bpeters@nevermor)
```

You may replace "tcsh" above with your favorite shell (sh, ksh, csh).

Now, one may make connections (slogin, scp, ssh) to hosts that are running ssh without using a passphrase.

** Other notes **

IMPORTANT: Ssh will automatically "tunnel" X sessions without user involvement even through multiple hops. However, it is important to NOT change the DISPLAY parameter or X will not work!

In an effort to reduce the probability of intrusion, ECS does not use the default TCP/IP port for ssh. So if you use ssh from facility other than the DAACs or the Landover facility, call the ECS help desk for the port so that your configuration can be changed.

5.3.4 Administration of Secure Shell

(TBD) See Audit File var/log/ssh

etc/ssh/config

5.4 Monitoring Requests for Network Services (TCP Wrappers)

With TCP Wrapper, you can monitor and filter incoming requests for network services, such as FTP.

TCP Wrapper provides small daemon wrapper programs that can be installed without any changes to existing software or to existing configuration files. The wrappers report the name of the client host and the name of the requested service; the wrappers do not exchange information with the client or server applications, and impose no overhead on the actual conversation between the client and server applications. The usual approach is to run one single daemon process that waits for all kinds of incoming network connections. Whenever a connection is established, this daemon runs the appropriate server program and goes back to sleep, waiting for other connections.

M&O personnel will monitor requests for these network services:

Client	Server	Application
ftp	ftpd	file transfer
finger	fingerd	show users

You will monitor the **log** file. To view log, at the command line type

nowait /var/log/wrappers

Standard Unix commands can be added, such as vi, emacs, or lp -d:

more/var/log/ wrappers | lp -d [printer name]

The log file provides information concerning who tried to access the network service. TCP Wrapper blocks any request made by unauthorized users. TCP Wrapper can be configured to send a message to any administrator whose request is rejected.

5.5 INSTALLATION, CONFIGURATION, and TESTING for Wrappers

Become root on the server that you would like to install wrappers

Note: This documentation is written for sun servers; however, you can replace

the package "wrappers_sun.7.4.tar" for other OS and proceed using the steps below.

(1) Go to any servers with root priveledge

cd /tools/admin/install_pkgs

ftp wrappers_sun.7.4.tar to the server that you would like to install

(put it in /tmp so it does not interfere with others)

(2) cd /etc

cp /etc/inetd.conf /etc/inetd.conf.orig

tar xvf /tmp/wrappers_sun.7.4.tar

(3) Create /etc/hosts.allow if it does not exist. Using the template shown below.

/etc/hosts.allow

==========

```
# Added for HIPPI network.  Do Not Delete!
ALL: 192.168.1.:   BANNERS /etc/wrappers/banners : ALLOW
ALL: 198.118.:   BANNERS /etc/wrappers/banners : ALLOW
ALL: 198.118.192.:   BANNERS /etc/wrappers/banners : DENY
ALL: 192.150.28.: BANNERS /etc/wrappers/banners : ALLOW
ALL: 38.177.222.: BANNERS /etc/wrappers/banners : ALLOW
ALL: 128.183.:   BANNERS /etc/wrappers/banners : ALLOW
ALL: ALL:       BANNERS /etc/wrappers/banners : DENY
```

(4) Insert the following lines into /etc/inetd.conf (usually right after the current ftp and telnet declaration entries):

```
ftp     stream  tcp     nowait  root    /etc/wrappers/tcpd      in.ftpd
telnet  stream  tcp     nowait  root    /etc/wrappers/tcpd          in.telnetd
```

and then comment the existing ftp and telnet entries there.

This is what it should look like:

```
#ftp    stream  tcp     nowait  root    /usr/sbin/in.ftpd       in.ftpd
#telnet stream  tcp     nowait  root    /usr/sbin/in.telnetd    in.telnetd
ftp     stream  tcp     nowait  root    /etc/wrappers/tcpd      in.ftpd
telnet  stream  tcp     nowait  root    /etc/wrappers/tcpd          in.telnetd
```

(5) Make sure these lines are commented out from /etc/inetd.conf (as shown):

```
##############################################################################
#
#  This is the original inetd.conf configuration
#
##############################################################################
#shell  stream  tcp     nowait  root    /usr/sbin/in.rshd       in.rshd
#login  stream  tcp     nowait  root    /usr/sbin/in.rlogind    in.rlogind
#exec   stream  tcp     nowait  root    /usr/sbin/in.rexecd     in.rexecd
#comsat dgram   udp     wait    root    /usr/sbin/in.comsat     in.comsat
#talk   dgram   udp     wait    root    /usr/sbin/in.talkd      in.talkd
#
```

(6) Insert this block of lines to /etc/inetd.conf (usually right after the block in step 5):

```
##############################################################################
#
#  This is the setup associated with the configuration of TCP WRAPPERS
#
##############################################################################
shell   stream  tcp     nowait  root    /etc/wrappers/tcpd      in.rshd
login   stream  tcp     nowait  root    /etc/wrappers/tcpd      in.rlogind
exec    stream  tcp     nowait  root    /etc/wrappers/tcpd      in.rexecd
comsat  dgram   udp     wait    root    /usr/sbin/in.comsat     in.comsat
talk    dgram   udp     wait    root    /etc/wrappers/tcpd      in.talkd
```

(6) ps -ef |grep inetd;

kill -HUP "inetd process number"

(7) Test to make sure that you can login from the existing host and also from another host on a different subnet. If it all goes well, that concluded the installation. Otherwise, go back to step (1) and doublecheck your steps.

(8) chmod 755 /etc/hosts.allow

touch /var/log/wrappers

chmod 755 /var/log/wrappers

rm /tmp/wrappers_sun.7.4.tar

rm /etc/inetd.conf.orig

5.6 Monitoring File and Directory Integrity (Tripwire)

Tripwire is a tool that aids in the detection of unauthorized modification of files resident on Unix systems. Tripwire is automatically invoked at system startup. This utility checks file and directory integrity by comparing a designated set of files and directories against information stored in a previously generated database. Tripwire flags and logs any differences, including added or deleted entries. When run against system files regularly, Tripwire spots any changes in critical system files, records these changes into its database, and notifies system administrators of corrupted or tampered files so that they can take damage control measures quickly and effectively. With Tripwire, system administrators can conclude with a high degree of certainty that a given set of files remain free of unauthorized modifications if Tripwire reports no changes.

Note: Since system files should not change and users' files change constantly, Tripwire should be used to **monitor only system files**. The list of system files you want to monitor is stored in **./configs/tw.conf**.

Tripwire is configured to mail the system administrator any output that it generates. However, some files on your system may change during normal operation, and this necessitates updating the Tripwire database.

5.6.1 Installation of Tripwire

Note: This is written for Solaris; however, Tripwire install for other OS (e.g. HP and SGI) is the same.

(1) Go to any servers with root privilege.

cd /tools/admin/install_pkgs

ftp tripwire_sun.1.2.tar to the server that you would like to install

(put it in /tmp so it does not interfere with others)

(2) cd /etc

cp /etc/inet/inetd.conf /etc/inet/inetd.conf.orig

tar xvf /tmp/tripwire_sun.1.2.tar

(3) To setup:

/etc/tripwire-1.2/src/tripwire -init

This will create a database file resided in

/etc/tripwire-1.2/src/databases/tw.db_"whatever_the_hostname"

(4) To test: touch /etc/intruder

/etc/tripwire-1.2/src/tripwire -v > /tmp/tw.report

This should reports the present of /etc/intruder. It works !!! Yeah !!!

(5) rm /etc/intruder

rm /tmp/tw.report

rm /tmp/tripwire_sun.1.2.tar

5.6.2 Updating the Tripwire Database

You can update your Tripwire database in two ways. The first method is interactive, where Tripwire prompts the user whether each changed entry should be updated to reflect the current state of the file, while the second method is a command-line driven mode where specific files/entries are specified at run-time.

5.6.2.1 Updating Tripwire Database in Interactive mode

Running Tripwire in Interactive mode is similar to the Integrity Checking mode. However, when a file or directory is encountered that has been added, deleted, or changed from what was recorded in the database, Tripwire asks the user whether the database entry should be updated.

For example, if Tripwire is run in Interactive mode and a file's timestamp changed, Tripwire will print out what it expected the file to look like, what it actually found, and then prompt the user whether the file should be updated. For example,

```
/etc/hosts equiv

    st_mtime: Wed May  5 15:30:37 1993      Wed May  5 15:24:09 1993

    st_ctime: Wed May  5 15:30:37 1993      Wed May  5 15:24:09 1993

---> File:  /etc/hosts equiv

---> Update entry?  [YN(y)nh?] y
```

You could answer yes or no, where a capital 'Y' or 'N' tells Tripwire use your answer for the rest of the files. (The 'h' and '?' choices give you help and descriptions of the various inode fields.)

While this mode may be the most convenient way of keeping your database up-to-date, it requires that the user be "at the keyboard." A more conventional command-line driven interface exists, and is described next.

5.6.2.2 Updating Tripwire Database in Database Update Mode

Tripwire supports incremental updates of its database on a per-file/directory or tw.config entry basis. Tripwire stores information in the database so it can associate any file in the database with the tw.config entry that generated it when the database was created.

Therefore, if a single file has changed, you can:

tripwire -update /etc/newly.installed.file

Or, if an entire set of files that made up an entry in the tw.config file changed, you can:

tripwire -update /usr/local/bin/Local_Package_Dir

In either case, Tripwire regenerates the database entries for every specified file. A backup of the old database is created in the **./databases** directory.

Tripwire can handle arbitrary numbers of arguments in Database Update mode.

The script **twdb_check.pl** script is an interim mechanism to ensure database consistency. Namely, when new entries are added to the tw.config file, database entries may no longer be associated with the proper entry number. The twdb_check.pl script analyzes the database, and remaps each database entry with its proper tw.config entry.

5.6.3 Configuring the tw.config file

Edit your **tw.config** file in the **./configs** directory, or whatever filename you defined for the Tripwire configuration file, and add all the directories that contain files that you want monitored. The format of the configuration file is described in its header and in the "man" page. Pay especially close attention to the select-flags and omit-lists, which can significantly reduce the amount of uninteresting output generated by Tripwire. For example, you will probably want to omit files like mount tables that are constantly changed by the operating system.

Run Tripwire with **tripwire -initialize**. This will create a file called **tw.db_[hostname]** in the directory you specified to hold your databases (where [hostname] will be replaced with your machine hostname).

Tripwire will detect changes made to files from this point on. You ***must*** be certain that the system on which you generate the initial database is clean; however, Tripwire cannot detect unauthorized modifications that have already been made. One way to do this would be to take the machine to single-user mode, reinstall all system binaries, and run Tripwire in initialization mode before returning to multi-user operation.

This database must be moved someplace where it cannot be modified. Because data from Tripwire is only as trustworthy as its database, choose this with care. It is recommended to place all the system databases on a read-only disk (you need to be able to change the disk to writeable during initialization and updates, however), or exporting it via read-only NFS from a "secure-server." (This pathname is hardcoded into Tripwire. Any time you change the pathname to the database repository, you must recompile Tripwire. This prevents a malicious intruder from spoofing Tripwire into giving a false "okay" message.)

We also recommend that you make a hardcopy printout of the database contents right away. In the event that you become suspicious of the integrity of the database, you will be able to manually compare information against this hardcopy.

Once you have your database set up, you can run Tripwire in Integrity Checking mode by typing **tripwire** on the command line from the directory in which Tripwire has been installed.

5.7 Reporting Security Breaches

Reporting of Security breaches shall be in accordance with EOSDIS Security Policy and Guides (EOSDIS-IVV-0821.2-9/30/97). Appendix B and C.

5.8 Initiating Recovery from Security Breaches

Recovery from Security breaches shall be in accordance with EOSDIS Security Policy and Guides (EOSDIS-IVV-0821.2-9/30/97). Appendix B and C.

This page intentionally left blank.

6. Network Administration

This section covers the procedures necessary for the management operations which monitor and control the system network capabilities.

Detailed procedures for tasks performed by the Network Administrator are provided in the sections that follow. The procedures assume that the administrator is authorized and has proper access privileges to perform the tasks (i.e., root).

The **$OV_BIN** directory must be in your path. If the **$OV_BIN** directory is not in your path, then each command referenced must be prefixed with the **$OV_BIN/** directory path to execute any of the commands specified.

Each procedure outlined will have an **Activity Checklist** table that will provide an overview of the task to be completed. The outline of the **Activity Checklist** is as follows:

Column one - ***Order*** shows the order in which tasks should be accomplished.

Column two - ***Role*** lists the Role/Manager/Operator responsible for performing the task.

Column three - ***Task*** provides a brief explanation of the task.

Column four - ***Section*** provides the Procedure (P) section number or Instruction (I) section number where details for performing the task can be found.

Column five - ***Complete?*** is used as a checklist to keep track of which task steps have been completed.

For each procedure outlined in this section, there is a corresponding **QUICK STEP** procedures immediately following on the procedure in this chapter. The **QUICK STEP** procedures are designed for persons who are ***experienced administrators with prior network administration experience***.

6.1 HP Open View Network Node Manager (NNM)

HP Open View Network Node Manager (NNM) is a multivendor network management application for use in managing TCP/IP networks and network devices that support the Simple Network Management Protocol (SNMP). NNM is an HP Open View SNMP-based application running under the HP Open View Windows (OVW) graphical user interface.

The NNM product is a configuration, performance, and fault management application for multivendor TCP/IP (Transmission Control Protocol/Internet Protocol) networks. NNM enables you to:

- Automatically discover the devices on the TCP/IP network and monitor the status of those devices.

- Automatically draw the Internet Protocol (IP) topology maps based on discovered information. A map is a graphical and hierarchical representation of your network and its systems. Discovered devices are placed in appropriate segments, networks, or Internet based on the topology of the IP network.
- Manage any vendor devices that support the Simple Network Management Protocol (SNMP). NNM can manage standard Management Information Base (MIB) objects, as well as Enterprise-specific MIB objects.
- Include new Enterprise-specific MIBs into the NNM MIB. Once you have loaded the new MIB module on the management station, you can manage any of the MIB objects defined in that MIB module.

The **Activity Checklist** table that follows provides a basic overview of the NNM functions..

Table 6.1-1. Network Administration - Activity Checklist

Order	Role	Task	Section	Complete ?
1	Net Admin.	Start Network Node Manager	(P) 6.1.1	
2	Net Admin.	Add a Network Object	(P) 6.1.2.1	
3	Net Admin.	Add a Segment Object	(P) 6.1.2.2	
4	Net Admin.	Add a Node Object	(P) 6.1.2.3	
5	Net Admin.	Add an IP Interface Object	(P) 6.1.2.4	
6	Net Admin.	View the Current Network and System Configuration	(P) 6.1.3	
7	Net Admin.	View Network Address Information	(P) 6.1.4	
8	Net Admin.	View How Traffic is Routed on a Network	(P) 6.1.5	
9	Net Admin.	View the Services Available on a Node	(P) 6.1.6	

Detailed procedures for these tasks are provided in the sections that follow.

6.1.1 Starting Network Node Manager (NNM)

HP Open View Network Node Manager is a set of applications that are integrated with HP Open View Windows (OVW). To start NNM, HP OpenView Windows must be activated first. Once activated, OVW will automatically start NNM. HP OpenView windows will also automatically start the applications that are installed and registered.

Prerequisites for this Task

The network management processes that work with OVW and NNM ***must be running***. The network management processes consist of the following HP OpenView background processes:

- **ovwdb** - OpenView Object Database deamon
- **trapd** - OpenView Event Specification and Event Configuration
- **ovtopmd** - OpenView IP Topology deamon
- **ovactiond** - OpenView deamon to execute shell commands upon receipt of an event
- **snmpCollect** - Simple Network Management Protocol
- **netmon** - Network Monitor
- Check to see if these processes are running using **/usr/ov/bin/ovstatus** command.

These procedures explain how to start the HP OpenView Windows graphical user interface:

1 Type **ovstart** at the command prompt and **press Enter**.

- If the network management processes are not running, start them by executing the **ovstart** command.

2 Type **ovstatus | more** at the command prompt and **press Enter**.

- This command will check the status of the processes. *If any errors are reported refer to the troubleshooting section of *Using HP Network Mode Manager*.

3 Type **ovw** at the command prompt and **press Enter**.

- This command will start HP OpenView Windows.

To exit NNM and all other integrated applications, you must exit OVW. You can exit OVW in **either** of the following ways:

3 Select **File: Exit** from the menu bar of any submap window

- OR -

4 **Click** on the **Close** button on all open submap windows until a black submap window is displayed. When the black submap window is displayed, click on the **Close** button.

- The open map is saved, and all the submap windows and dialog boxes of the map are closed. OVW, all NNM applications, and all other integrated applications are closed.

Table 6.1-2 presents the Quick-Step Procedures to start Network Node Manager.

Table 6.1-2. Starting NNM (Network Node Manager) - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	ovstart $OV_BIN/ovstart if network management processes are not running	press Enter
2	ovstatus	press Enter
3	x11start	press Enter
4	ovw $OV_BIN/ovw if $OV_BIN directory is not in the path	press Enter
	To exit Network Node Manager: (2 methods)	
5	Exit	File → Exit
6	Close button on all open submap windows	single-click on Close button

6.1.2 Creating Additional Objects

To complete the distribution of resources over the map to better match how your network is organized, you must first expand the lower levels of the network map by creating additional network, segment, and node objects. The following sections show how to add network, segment, node, and interface objects to the network map so that IP Map will manage them.

6.1.2.1 Adding a Network Object

Prerequisites for this Task

The map must be opened with read-write access. The first instance of opening the map starts up in read-write status as indicated on the lower left of the map window.

1-4 Follow prerequisites steps (1-4 ***ONLY***) for Task: ***Starting Network Node Manager***

5 Select **Edit: Add object**

6 From the **symbol palette, choose the desired symbol type for the network object by selecting the desired subclass and use button 2 to drag the symbol to the submap**. The **Add object** dialog box appears.

7 Enter a selection name for the object in the **Selection Name** field of the **Add Object** dialog box.

8 In the **Object Attributes** list, select **IP Map** and click **Set Object Attributes**. The **IP Map Set Attributes** dialog box for a network object appears.

9 Enter a **Network Name.**

10 Enter a **Network Address**.

11 Optionally, **Network Subnet Mask** can be entered.

12 Click **Verify** to check for valid entries.

13 Click on **OK** to close the **Set Attributes** dialog box.

14 Click on **OK** in the **Add Object** dialog box to complete the operation.

Table 6.1-3 presents the **QUICK STEP** procedures to Add a Network Object.

Table 6.1-3. Adding a Network Object - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1 - 4	Follow prerequisites steps (1-4 ***ONLY***) for Task: ***Starting Network Node Manager***	press Enter
5	Edit:	select with mouse
6	Select the desired symbol type	select with mouse
7	Selection Name	(No action)
8	Select IP Map	click Set Object Attributes
9	Enter Network Name	press Enter
10	Enter Network Address	press Enter
11	Enter Network Subnet Mask (Optional)	press Enter
12	Select Verify	select with mouse
13	Select OK to close the Set Attributes	select with mouse
14	Select OK in the Add Object	select with mouse

6.1.2.2 Adding a Segment Object

If the segment which the node is on can be identified, IP Map places the segment on that node. If it cannot make the identification, IP Map places the segment on the default segment submap. The default segment submap is the submap created by IP Map when OVW was first started. If that submap has been deleted, the default segment submap becomes the oldest segment submap. IP Map discovers new nodes on segments attached to SNMP, IP addressable bridges and multi-port repeaters (hubs).

Prerequisites for this Task

The map must be opened with read-write access.

1-4 Follow prerequisites steps (1-4 ***ONLY***) for Task: ***Starting Network Node Manager***

5 Select **Edit: Add Object**.

6 From the **symbol palette, choose the desired symbol type for the segment object by selecting the desired class, then the desired subclass, and drag the symbol to the submap**. The **Add Object** dialog box appears.

7 Enter a selection name for the object in the **Selection Name** field of the **Add Object** dialog box.

8 In the **Object Attributes** list, select **IP Map** and click **Set Object Attributes**. The **IP Map Set Attributes** dialog box for a segment object appears. A figure of the dialog box follows this procedure.

9 Enter a **name for the segment**. It must be unique to other segment names in the submap.

10 Click **Verify** to check for valid entries.

11 Click on **OK** to close the **Set Attributes** dialog box.

12 Click on **OK** in the **Add Object** dialog box to complete the operation.

Table 6.1-4 presents the Quick-Step Procedures to Add a Segment Object to the network map.

Table 6.1-4. Adding a Segment Object- Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1 - 4	Follow prerequisites steps (1-4 ***ONLY***) for Task: ***Starting Network Node Manager***	press Enter
5	Select Edit: Add Object	select with mouse
6	Desired symbol type for the segment object by	select desired class, then the desired subclass, and drag the symbol to the submap
7	Enter Selection Name	press Enter
8	Select IP Map	click on Set Object Attributes
9	Enter a name for the segment (must be unique)	press Enter
10	Select Verify	select with mouse
11	Select OK to close the Set Attributes	select with mouse
12	Select OK in the Add Object dialog box	select with mouse

6.1.2.3 Adding a Node Object

An Object can be added that represents a node or a network device to Segment submap by placing one of the supported symbols on a Segment submap. Double-clicking on the node symbol opens a Node submap. IP Map discovers and manages the symbols that represent the Computer, Connector, and Net Device classes in a Node submap.

Prerequisites for this Task

The map must be opened with read-write access.

1-4 Follow prerequisites steps (1-4 ***ONLY***) for Task: ***Starting Network Node Manager***

5 Select **Edit: Add Object**

6 From the **symbol palette, choose the desired symbol type for the node object by selecting the desired class, then the desired subclass, and drag the symbol to the submap**. The Add Object dialog box appears.

7 Enter a selection name for the object in the **Selection Name** field of the **Add Object** dialog box.

8 In the **Object Attributes** list, select **IP Map**, and click **Set Object Attributes**. The **IP Map Set Attributes** dialog box for a node object appears.

9 Enter the hostname of the node.

10 Enter the IP address of the node.

11 Click **Verify** to check for valid entries.

12 Click on **OK** to close the **Set Attributes** dialog box.

13 Click on **OK** in the **Add Object** dialog box to complete the operation.

Table 6.1-5 presents the Quick-Step procedures to Add a Node Object to the network map.

Table 6.1-5. Adding a Node Object- Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1 - 4	Follow prerequisites steps (1-4 ***ONLY***) for Task: ***Starting Network Node Manager***	**press Enter**
5	**Select Edit: Add Object**	**select with mouse**
6	**Desired symbol type for the segment object**	**select desired class, then the desired subclass, and drag the symbol to the submap**
7	**Enter selection name**	**press Enter**
8	**Select IP Map**	**click Set Object Attributes**
9	**Enter hostname of the node**	**press Enter**
10	**Enter the IP address of the node**	**press Enter**
11	**Select Verify**	**select with mouse**
12	**Select OK to close the Set Attributes**	**select with mouse**
13	**Select OK in the Add Object to complete the operation**	**select with mouse**

6.1.2.4 Adding an IP Interface Object

IP interface can be added to a Node submap by placing an IP Interface symbol on a Node submap. This is done by entering the IP address of the interface.

Prerequisites for this Task

The map must be opened with read-write access.

1-4 Follow prerequisites steps (1-4 ***ONLY***) for Task: ***Starting Network Node Manager***

5 Select **Edit: Add Object**.

6 From the **symbol palette, select the IP Interface symbol in the Cards class**. The Add Object dialog box appears.

7 Enter a **selection name for the object in the Selection Name field of the Add Object dialog box**.

8 In the Object Attributes list, **select IP Map, and click Set Object Attributes**. The IP Map Set Attributes dialog box for an IP Interface object appears.

9 Enter the **IP Address for the IP interface**. The subnet mask is added for you.

10 Click **Verify** to check for valid entries.

11 Click on **OK** to close the Set Attributes dialog box.

12 Click on **OK** in the Add Object dialog box to complete the operation.

Table 6.1-6 presents the Quick-Step procedure to Add an IP Interface Object to the Network Map.

Table 6.1-6. Adding an IP Interface Object- Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1 - 4	Follow prerequisites steps (1-4 ***ONLY***) for Task: ***Starting Network Node Manager***	press Enter
5	Select Edit: Add Object	select with mouse
6	Select the IP Interface	select with mouse
7	Enter selection name for the object	press Enter
8	Select IP Map	select Set Object Attributes
9	Enter IP Address for the IP interface	press Enter
10	Select Verify	select with mouse
11	Select OK to close the Set Attributes dialog box	select with mouse
12	Select OK in the Add Object dialog box	select with mouse

6.1.3 Viewing the Current Network and System Configuration

NNM provides quick access to information about your network and system configurations. This section points you to the menu items available to accessing this information.

Prerequisites for this Task

One or more nodes must be selected from the map. If you select more than one node, you receive a dialog box for each selected node.

1-4 Follow prerequisites steps (1-4 ***ONLY***) for Task: ***Starting Network Node Manager***

5 Select the **object** for which you want a description.

6 Select Monitor: **Description - Selected Objects**. The **Object Description** dialog box appears.

7 In the **Object Description** dialog box, select **IP Map** and select **View/Modify Object Attributes**. The **Set Attributes** dialog box appears.

Table 6.1-7 presents the **QUICK STEP** procedures to View the Current Network and System Configuration.

Table 6.1-7. Viewing the Current Network and System Configuration - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1 - 4	Follow prerequisites steps (1-4 ***ONLY***) for Task: ***Starting Network Node Manager***	**press Enter**
5	**Select the object for which you want a description**	**select with mouse**
6	**Select Monitor: Description - Selected Object**	**select with mouse**
7	**Select IP Map**	**Select with mouse**
8	**Select View/Modify Object Attributes**	**Select with mouse**

6.1.4 Viewing Network Address Information

This task is useful for determining the addresses associated with a node, without looking through configuration files. The information you see is real-time data taken from the node versus static information taken from a database.

Prerequisites for this Task

- The node must support SNMP.
- One or more SNMP nodes must be selected from the map. If you select more than one node, you receive a dialog box for each selected node.

1-4 Follow prerequisites steps (1-4 ***ONLY***) for Task: ***Starting Network Node Manager***

5 Select a **node** on the map.

6 Use the **Monitor: Network Configuration - Addresses….** Operation to view the following information about each interface on the node:

- interface index
- interface name
- IP address
- network mask
- network address
- link-level address (physical address)

For example:

Index	Interface	IP address	Network Mask	Network Address	Link Address
4	lan1	126.1.0.2	255.255.0.0	126.1.0.0	0x01002033333
3	lan0	126.1.0.3	255.255.0.0	126.1.0.0	0x01000202222
2	lo0	222.0.0.1	255.0.0.0	222.0.0.0	<none>

Table 6.1-8 presents the **QUICK STEP** procedures to view network address information.

Table 6.1-8. Viewing Network Address Information-Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1 - 4	Follow prerequisites steps (1-4 ***ONLY***) for Task: ***Starting Network Node Manager***	**press Enter**
5	**Select a node on the map**	**select with mouse**
6	**Use the Monitor: Network Configuration - Addresses**	**select with mouse**

6.1.5 Viewing How Traffic is Routed on a Network

This task lists the routing table information for a remote SNMP node. It can be useful in determining more efficient routes on the network, assessing the need for explicit routes and diagnosing connectivity problems. The information you see is real-time data taken from the node versus static information taken from a database.

When rerouting traffic on the network, there are two ways of performing the task:

- For temporary change, the command route command is used.
- For permanent change, netlinkrc needs to be edited and the system needs to be rebooted.

Prerequisites for this Task

- The node must support SNMP.
- One or more SNMP node must be selected from the map. If you select more than one node, you receive a dialog box for each selected node.

1-4 Follow prerequisites steps (1-4 ***ONLY***) for Task: ***Starting Network Node Manager***

5 Select a **node** on the map.

6 Use the **Monitor: Network Configuration - Routing Table..** operation to view the following information about each destination node with which the selected node communications:

- destination name (default is a route that the system uses when it cannot find a specific route)
- name of the gateway (router) between the selected node and the destination
- type of route (for example, directly connected to a LAN, through a remote gateway, or route currently not available)
- network subnet mask associated with the route
- name of the interface that is used to reach the destination

Table 6.1-9 presents the Quick-Step procuedure for Viewing Traffic.

Table 6.1-9. Viewing How Traffic is Routed on a Network-Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1 - 4	Follow prerequisites steps (1-4 ***ONLY***) for Task: ***Starting Network Node Manager***	**press Enter**
5	**Select a node on the map**	**select with mouse**
6	**Select Monitor: Network Configuration - Routing Table**	**select with mouse**

6.1.6 Viewing the Services Available on a Node

This task lists the IP networking services for which a remote SNMP node is listening. It is useful for determining what configured services a node is currently running. The information you see is real-time data taken from the node versus static information taken from the database.

Prerequisites for this Task

- The node must support SNMP.
- One or more SNMP nodes must be selected from the map. If you select more than one node, you receive a dialog box for each selected node.

1-4 Follow prerequisites steps (1-4 ***ONLY***) for Task: ***Starting Network Node Manager***

5 Select a **node** on the map.

6 Use the **Monitor: Network Configuration - Services**.. operation, to view the following information about the selected node:

- service protocol: either TCP (Transmission Control Protocol) or UDP (User Datagram Protocol).
- Port to which the service is bound.
- Service for which the node is listening (for example, telnet, nfs). If no service is listed, the service is either unavailable or unknown.

Table 6.1-10 presents the Quick-Step procedures to View Available Services on a node.

Table 6.1-10. Viewing the Services Available on a Node-Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1 - 4	Follow prerequisites steps (1-4 ***ONLY***) for Task: ***Starting Network Node Manager***	**press Enter**
5	**Select a node on the map**	**select with mouse**
6	**Select Monitor: Network Configuration - Services**	**select with mouse**

6.2 Diagnosing Network Problems

A fault within network is defined as something that causes those systems "to fail to meet their operational objectives." Three elements are involved in managing network faults: detection of the fault, isolation of the fault to a particular component, and correction of the fault. Fault management, therefore, may include the maintenance of error logs, error detection processes, and diagnostic testing procedures. For many managers, the term network management is synonymous with fault management.

Performance and fault management are difficult to separate. High performance usually implies a low incident of faults. Performance management, however, goes beyond minimizing faults; it is responsible for gathering statistics on the operation of the network, maintaining and analyzing logs of the state of the system, and optimizing network operation.

Sniffer Network Analyzer: Ethernet Monitor is used to make sure the network is working at its peak performance and will diagnose any possible fault within the network. The Ethernet Monitor is a network monitoring program. The monitor provides an accurate picture of network activity at any moment or a historical record on network activity over a period of time. This information helps you find traffic overloads, plan for network expansion, detect intruders, establish performance baselines, and distribute traffic more efficiently among servers and subnets.

The monitor's report capabilities let you communicate this information to others, complete with graphs and tables. The alarm capabilities alert you to problems with the network or with individual stations.

This list summarizes the monitor's capabilities:

- Monitors up to 1,024 network stations
- Generates visible and audible alarms for the entire network or for individual stations
- Complies a historical alarm log
- Provides real-time traffic and historical information for individual stations and for the entire network

- Sorts statistics to show only the items that interest you
- Creates customized management reports
- Automatically prints selected information at set time intervals

Note: The Ethernet Monitor only monitors frames on the Ethernet network segment to which the Network Interface Card is attached.

7. System Monitoring

7.1 Checking the Health and Status of the Network

Once a network has been discovered by **HP OpenView IP discovery and layout,** monitoring the state of the network can begin. Monitoring includes tasks, such as, checking the map for color alerts which indicate problems, creating submaps needing special monitoring, and checking for network changes.

Objects that have an abnormal condition can be identified without having to look at every object on the network map. A color alert on a symbol indicates that some part of that object may have problems. To isolate a fault somewhere on the network, follow the color alerts to increasingly more specific submaps until the specific object that is not functioning is reached. Follow color alerts by opening child submaps of objects that contain a color alert.

In addition, HP **OpenView** has the capability of monitoring the status of each of the application servers and modes. Through the use of this tool the user is capable of determining status of both hardware and software components.

Each procedure outlined will have an **Activity Checklist** table that will provide an overview of the task to be completed. The outline of the **Activity Checklist** is as follows:

Column one - ***Order*** shows the order in which tasks should be accomplished.

Column two - ***Role*** lists the Role/Manager/Operator responsible for performing the task.

Column three - ***Task*** provides a brief explanation of the task.

Column four - ***Section*** provides the Procedure (P) section number or Instruction (I) section number where details for performing the task can be found.

Column five - ***Complete?*** is used as a checklist to keep track of which task steps have been completed.

Table 7.1-1. Monitoring - Activity Checklist (1 of 2)

Order	Role	Task	Section	Complete?
1	Fault Manager	Starting NNM (Network Node Manger) and the Graphical Interface	(P) 7.1.1	
2	Fault Manager	Verify that an object is not functioning	(P) 7.1.2	
3	Fault Manager	Looking at Maps for Color Alerts	(P) 7.1.3	
4	Fault Manager	Looking at Maps for New Nodes	(P) 7.1.4	
5	Fault Manager	Create Special Submaps for Monitoring Status	(P) 7.1.5	
6	Fault Manager	Checking for Event Notifications	(P) 7.1.6	
7	Fault Manager	Rediscovering Network	(P) 7.1.7	

Table 7.1-1. Monitoring - Activity Checklist (2 of 2)

Order	Role	Task	Section	Complete?
8	Fault Manager	Configure HPOV Monitoring Settings	(P) 7.1.1.1	
9	Fault Manager	Setup Data Collection	(P) 7.1.1.2	
10	Fault Manager	Add Vendor MIBs	(P) 7.1.1.3	
11	Fault Manager	Community Name Setup	(P) 7.1.1.3.1	
12	Fault Manager	Setup Map	(P) 7.1.1.4	
13	Fault Manager	Tivoli Enterprise Console	(I) 7.2	
14	Fault Manager	Application Servers	(P) 7.3	
15	Fault Manager	Locate Application Servers	(P) 7.3.1	
16	Fault Manager	Start Application Servers	(P) 7.3.2	
17	Fault Manager	Shutdown Application Servers	(P) 7.3.3	
18	Fault Manager	Remove Red Applications	(P) 7.3.4	
19	Fault Manager	Exit Application Servers	(P) 7.3.5	

Detailed procedures for tasks performed by the Fault Manger are provided in the sections that follow.

If you are already familiar with the procedures, you may prefer to use the quick-step tables. If you are new to the system, or have not performed this task recently, you should use the detailed procedures presented in Sections 7.1.1 through 7.1.5

For each procedure outlined in this section, there is a corresponding **QUICK STEP** procedures immediately following the procedure in this chapter. The **QUICK STEP** procedures are designed for persons who are ***experienced network administrators with prior monitoring/network administration experience***.

7.1.1 Starting NNM (Network Node Manager)

HP **OpenView** Network Node Manager is a set of applications that are integrated with HP **OpenView** Windows (OVW). To Start NNM, HP Openview Windows must be activated first. Once activated, OVW will automatically start NNM. HP Openview windows will also automatically start the applications that are installed and registered.

Prerequisites for this Task

The network management processes that work with OVW and NNM must be running. The network management processes consist of the following HP OpenView background processes: **ovwdb, trapd, ovtopmd, ovactiond, snmpCollect, and netmon**. You can check to see if these processes are running with **/usr/ov/bin/ovstatus** command.

1. Check for required processes
2. The ***EcMsCmEcsd*** daemon must be started first on the **MSS Server** to ensure proper tracking of system icons.

   -- Login to the MSS Server.
   -- Determine if the "EcMsCmEcsd" process is running

-- (/bin/ps -ef | grep EcMsCmEcsd).

-- **If not**, start it by doing the following:

-- Run /usr/ecs/SHARED/CUSTOM/utilities/EcMsCmStartEcsd SHARED

3. The ***Deputy Agent*** must be then be started on the **MSS Server**.

   -- Login to the MSS Server.

   -- Determine if the Deputy Agent is running (/bin/ps -ef | grep EcMsAgDeputy)

   -- **If not**, start it by doing the following:

   -- Run /usr/ecs/SHARED/CUSTOM/utilities/EcMsAgStartDeputy SHARED. The script determines if the Deputy Agent is already running (only one copy of Deputy Agent should be running at a time). To also ensure proper tracking of system icons, the EcMsAgDeputy must be shutdown prior to shutting down the EcMsCMEcsd daemon.

4. The ***Subagent*** must be running on the host(s) where you want to ***manage applications***. If not, start it by doing the followings:

   -- Run /usr/ecs/SHARED/CUSTOM/utilities/EcMsAgStartSubagent SHARED.
   (only one copy of Deputy Agent should be running at a time).

   -- If an ECS host crashes, the SubAgent cannot relay an event to OpenView, and the operator will have to manually remove any spurrious icons from the display.

This procedure explains how to start the HP OpenView Windows graphical user interface.

1 Set and initiate the appropriate environment variables:

- setenv **DISPLAY** *hostname:0.0*
- setenv **ECS_HOME** /usr/ecs
- setenv **SHLIB_PATH** /opt/OV/lib
- source **/opt/OV/bin/ov.envvars.csh**
- source **/etc/Tivoli/setup_env.csh**
- ↑ If the map is opened as a "read-only" map ("read-only" or "read-write will be indicated in the lower left corner of the GUI), you will need to use the "File->Refresh Map" menu item to refresh the map whenever there are changes other than symbol color changes.

2 Type **$OV_BIN/ovstart** at the command prompt and **press enter**.

- ↑ If the network management processes are not running, you can start them by executing the **$OV_BIN/ovstart** command.

3 Type **ovstatus** at the command prompt and **press enter**.

- This command will check the status of the processes.

Start the HP OpenView Windows NNM Graphical Interface

4 cd /data1/OPS/COTS/OV/bin and press enter
-- Type **$OV_BIN/ovw -map *mapname* &** at the command prompt and **press enter**. This displays a map with two icons (IP Network and Services).
-- If HPOV is started without specifying a map, then the view is an expanded version of your current node.

-- If the map is opened as a "read-only" map ("read-only" or "read-write will be indicated in the lower left corner of the GUI), you will need to use the "File --→Refresh Map" menu item to refresh the map whenever there are changes other than symbol color changes.

-- Double click on either the Internet or Services symbol to expand either symbol.

To exit NNM and all other integrated applications, you must exit OVW. You can exit OVW in one of the following ways:

6 Select **File: Exit** from the menu bar of any submap window or go to step number 7;

7 **Click** on the **Close** button on all open submap windows until a black submap window is displayed. When the black submap window is displayed, click on the **Close** button.

- ↑ The open map is saved, and all the submap windows and dialog boxes of the map are closed. OVW, all NNM applications, and all other integrated applications exit.

IMPORTANT: Do not use the **Quick-Step** version of a procedure unless you are already very familiar with the procedure. To start **NNM**, execute the steps provided in the table.

Table 7.1-2. Starting/Monitoring NNM - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	Login to appropriate machine	press Return
2	setenv DISPLAY IPNumber:0.0 -- or – setenv DISPLAY hostname:0.0	press Return
	cd /data1/OPS/COTS/OV/bin	press Return
3	$OV_BIN/ovstart (HPOV not running)	press Return
4	ovw&	press Return
5	Select Monitor – or – Diagnose	press Return
5	Select device Configuration –or—Network Connectivity	press Return
6	Select System information –or – Demand Poll	press Return
7	$OV_BIN/ovw -map mapname&	press Return

7.1.1.1 Configure Monitoring Settings

This section explains how to configure the appropriate elements to be managed and monitored by HP Openview. To ensure that the appropriate elements are being included as an elements to be monitored and managed by HPOV:

1. Bring up the IP Internet submap by double clicking on the IP Internet icon.

2. Ensure that all icons representing managed objects (managed hosts and network devices, as listed in the site hardware baseline mapping) are being managed. An unmanaged icon will normally be wheat-colored. A managed icon will usually be blue, green, yellow, or red.
   -- A view of the color scheme currently in use by HPOV can be obtained by selecting:
      -- Help->Display Legend.

3. To *manage objects that are* <u>*currently unmanaged*</u> by HPOV,
   -- select the ***object*** by left-clicking on the object
      -- then select "Map--> Manage Objects" menu option.
      The color of the icon representing this object should change to indicate that the object is now being managed by HPOV. Once the object becomes managed, additional icons may appear on The map as HPOV discovers other devices that are connected to that object.

4. Repeat step 3 (above) until all <u>*unmanaged*</u> objects at the site are managed.

5. **Note:** HPOV may discover some icons to be part of a subnet, and group them together under a subnet icon. In this case, the object icons may be found by "*drilling down*" to the next level (*double- clicking* on the subnet icon). If an object that should be managed by HPOV cannot be found, look for it by using the -- Locate->Selected Objects List... menu item.
   -- This will bring up a "Locate by Selection Name" window.

   The object (if it has been discovered by HPOV) can be found by typing all or part of the object name into the box labeled "Regular Expression (or substring):" on this window. When the name (or a part thereof) is entered and the "Apply" button is selected, HPOV will search for all instances of this object and will list them in the "Located and Highlighted:" box. If an object still cannot be found, it has not been discovered by HPOV. The most likely reasons for this would be if it is connected to the network via another device that has also not been discovered or it has been down since HPOV started its discovery process.

6. Ensure that all icons not representing managed objects are unmanaged.

   Note: HPOV can manage additional network devices; however, the process of managing the device will cause additional network traffic that may not be desired, especially if the device is not owned by ECS.

7. To unmanage an object that is currently being managed by HPOV, select the icon by left-clicking on it and then select the "Map-> Unmanage Objects" menu option. The color of the icon should change to wheat fairly quickly to indicate its unmanaged state.

7.1.2 Verify That an Object Is Not Functioning

This section explains how to verify that an object is not functioning and assumes that HP Openview is running. If not, then see Section 7.1.1. To verify that an object is not functioning, any of the following procedures can be executed.

1 Select the **Configuration** pull down menu

2 Select **System Information**

- OR -

1 Select the **Fault** pull down menu
2 Select **Network Connectivity: Poll Node**

- OR -

1 Select the **Fault** pull down menu
2 Select **Ping**

If these operations do not produce any responses or they time out, then the node is probably down or otherwise unreachable over the network. See Section 7.1.5 Checking for Event Notifications to verify event status of the node. If a Fault has occurred see Section 8 on Problem Management and Section 21 COTS Hardware Maintenance.

IMPORTANT: Do not use the quick step version of a procedure unless you are already very familiar with the procedure.

To verify that an object is not working, execute the steps provided in the table.

Table 7.1-3. Verify - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	Configuration	Use the pull down menu
2	System Information	Pull down menu

7.1.3 Looking at Maps for Color Alerts

Looking at maps for color alerts, HP ***OpenView*** uses colored symbols to indicate the health and operational status of the various objects within the network. Operational status refers to the actual operation of the node on the sub-map. Operational status is changed by Network Node Manager(NNM) automatically when a change in the operational condition of the node is detected.

Table 7.1-4. Operational Status Colors

SYMBOL COLOR	STATUS
Blue	UNKNOWN. A device or a network that has been discovered by NNM. This can happen when a non-IP addressable device has been added to the map manually and NNM has not yet discovered it. The status is set to unknown. It cannot be determined.
Green	NORMAL. The device or network is up and functioning normally. The object is in its normal operational state.
Cyan	WARNING. There is a potential fault or problem
Yellow	MINOR or MARGINAL. There is something wrong with this object, but the device is still responding to NNM.
Orange	MAJOR. The object is in a state of having a major fault or problem.
Red	CRITICAL or DOWN. The device is not responding to NNM.

Looking at maps for color alerts

To check your network for color alerts, you must first have the map for the network open. To open a map, use the following procedure:

1. With HP ***OpenView*** NNM running, follow menu path **Map->Maps->Open/List**

- The **Available Maps** dialog box is displayed.

2. Select the name of the map you want to open and click on "**Open Map**".

- A confirmation box is displayed

3. Click "**OK".**

- Any open map and its sub-map windows and dialogs close.
- The **Home Sub-map (Root)** of the selected map is displayed.

If you do not know the compound status scheme of the open map, follow the menu path **File->Describe/Modify Map** to obtain the **Map Description** dialog box and display/set the compound status scheme for default. Suppose there is a fault in an interface card in one of the workstations on your network. Use the following procedure to trace it using color alerts.

Looking at Maps for Color Alerts

1. Double click on the yellow **IP Internet** symbol

- Note: The symbol will be yellow because the *critical failure* of the card in the workstation is propagated up to the level of the **IP Internet** symbol as a *minor* problem at that level.
- The **Internet sub-map** opens and displays the IP network(s). One network symbol is yellow. This indicates a marginal problem with the network.

2. Double click on the yellow **IP network** symbols**.**
   - A network sub-map opens and displays the segment(s) attached to the gateway(s). The segment symbol is yellow. This indicates a problem somewhere on the segment.
3. Double click on the yellow **segment** symbol.
   - A **segment sub-map** opens and displays the nodes attached to that segment. Of all the nodes in the segment, the workstation node is red. The problem is isolated to that workstation.
4. Double click on the red **workstation** symbol.
   - A **Node sub-map** opens and displays its interface symbol. It is red.
   - You have isolated the fault to a single card of a single node on your network.

IMPORTANT: Do not use the quick step version of a procedure unless you are already very familiar with the procedure.

To look at Maps for Color Alerts, execute the steps provided in the table.

Table 7.1-5. Color Alerts- Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	Internet Symbol	Double click
2	IP Network	Double click
3	Yellow segment symbol	Double click
4	Red workstation symbol	Double click

7.1.4 Looking at Maps for New Nodes

This section assumes that HP Openview is running. If not, then see section 7.1.1 **IP Map** will automatically discover the **IP-addressable nodes** for the open map. The purpose of this section is to identify new objects that have been discovered and added to the open map. Because discovery is automated, additional symbols of objects on the network map will be seen. **IP Map** must be enabled for the map. This is the default. **IP Map** places new symbols directly on the submap if autolayout is enabled. IP Map places new symbols in the **New Object Holding Area** if autolayout is disabled for the submap.

1 To check the default **Segment submap** for any new nodes that may have been discovered, open the default **Segment submap** from the segment symbol in the Network submap.

- View the submap for any new symbols.

2 To easily see new symbols in the submap, disable autolayout for the submap. When autolayout is disabled, a **New Object Holding Area** appears at the bottom of the submap.

- All newly added symbols are placed in the **New Object Holding Area**.

7.1.5 Creating Special Submaps for Monitoring Status

This section assumes that HP Openview is running on the desktop. If not, then see section 7.1.1 Submaps can be created that are logically organized instead of physically organized. This will help to create logical submaps for specialized monitoring.

IMPORTANT: See section 4-5 of the **HP Openview Network Node Manager User's Guide** to use this feature.

7.1.6 Checking for Event Notifications

This section assumes that HP Openview is running on the desktop. If not, then see Section 7.1.1 Anytime a change occurs on the network an event is generated. Through the **Network Node Manager's** internal processes, the event is sent to a predefined category in the **Events Browser** window. The **Events Categories** window provides a notification of when new events occur. This window has a button corresponding to each of the event categories. When the button that corresponds to a specific event category is clicked, a window listing the events for that specific category appears. These windows are **Event Browser** windows. When a button in the **Event Categories** window changes color, it is an indication that an event occurred on the network which relates to that category. The color of the button indicates the highest severity event in the category. The default categories included in the **Event Categories** window are:

Error Events. This indicates inconsistent or unexpected behavior.
Threshold Events. This indicates that a threshold was exceeded.
Status Events. This indicates an object or interface status changed to up or down, or an object or interface started or stopped responding to ICMP echo requests.
Configuration Events. This indicates a node's configuration changed.
Application Alert Events. This indicates an HP OpenView Window application generated an alarm or alert.
All Events. This list all the above events and other events in one dialog box.

In the following example the **Threshold Events** button is red, which indicates that a critical threshold was exceeded somewhere on the network.

1 Click on the **Threshold Events** button in the **Event Categories** window. The **Threshold Events Browser** dialog box appears with a chronological listing of the threshold events that have occurred, with the most recent events at the bottom of the list.

- Each event listed includes the severity, time the event occurred, node on which the event occurred, and a brief event message.

2 To view the node that generated the event shown in this example, select the event from the list and click on **Action → Highlight Source on Map.**

- A map will appear with the **busynode** node highlighted. At this point, select the highlighted node by clicking on it, and invoke appropriate operations from the menu bar to further diagnose and correct the situation which caused the threshold to be exceeded.

3 To delete the event, select the event and click on **Action → Delete → Selected Event.**

- This will delete only the selected event.
- For more information about event notification, click on the **help** button in the dialog box for the event being viewed or select **View SNMP Events** from the **Help: Index → Task**

IMPORTANT: Do not use the quick step version of a procedure unless you are already very familiar with the procedure.

To check for **Event Notifications**, execute the steps provided in the table.

Table 7.1-6. Event Notifications - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	Threshold Events	press Return
2	Action	press Return
3	Highlight source on Map	press Return

7.1.7 Rediscovering the Network

This section assumes that HP Openview is running on the desktop. If not, then see section 7.1.1 Occasionally you may edit your maps beyond recognition and want to start from scratch.

1 Exit all Openview sessions (if running) **cd /usr/OV/bin** and then enter **ovstop** at the command line

- This will stop all HP OpenView processes

2 Remove the Openview database **(do a backup first)**

- **cd $OV_DB/openview**
- **rm -rf $OV_DB/openview/***

3 Remove all of the current events**.**

- **rm $OV_LOG/xnmevents.***
- **rm $OV_LOG/trapd.log***
- **rm $OV_LOG/netmon.trace***

4 Clear the SNMP cache**.**

- cd /usr/OV/bin
- **xnmsnmpconf -clearCache**

5 Re register OVW fields**.**

- **ovstart ovwdb**
- **ovw -fields**

6 Restart NNM**.**

- **ovstart**
- **ovw &**

IMPORTANT: Do not use the quick step version of a procedure unless you are already very familiar with the procedure.

Table 7.1-7. Rediscovery - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	Stop all OVW sessions	select exit from pull down menu
2	ovstop	press enter
3	cd $OV_BIN/openview	press enter
4	rm -rf $OV_DB/openview/*	press enter
5	rm $OV_LOG/xnmevents.*	press enter
6	rm $OV_LOG/trapd.log*	press enter
7	rm $OV_LOG/netmon.trace*	press enter
8	xnmsnmpconf -clearCache	press enter
9	ovstart ovwdb	press enter
10	ovw -fields	press enter
11	ovstart	press enter
12	ovw &	press enter

7.1.8 Starting and Shutting Down Servers from HP OpenView

Starting and Shutting Down Servers from HP OpenView

One of the key features of HP OpenView is its capability to start and shutdown system servers using the graphical user interface. This can be accomplished either for all servers in a specific mode or individually by subsystem. Using this procedure will save significant effort over command-line initiation of servers.

1 Start the HP OpenView application.

2 After the **About OVW** box is displayed, the OVW **Root** window will appear. From the menu bar click on the **Map; Maps; Open/List** selections to bring up the **Available Maps** selection window.

3 Select the desired map to bring up the **Root** map that displays the **Services** and **IP Internet** icons. Double click on the **Services** icon to display the **Service** submap.

4 From the **Service** submap select either the **Mode** icon to bring up the Mode submap to start or shutdown all servers for a given mode (TS1, TS2, OPS, etc.), or select a specific subsystem icon to start or shutdown servers for a specific subsystem.

5 Right-click and hold on the desired Mode icon to display the pop-up menu. Continue to hold and scroll down to either the **Start executable** or **Shutdown executable** option. Release the right mouse button and the servers for the selected mode will start or shutdown.

6 To start or shutdown specific servers in a subsystem (CLS, MSS, IDG, etc.), double-click on the desired subsystem icon in the **Service** submap.

7 From the expanded submap right-click and hold on the desired server application to display the pop-up menu, scroll down to the **Start executable** or **Stop executable** option, release the button and the server will start or shutdown.

In order to assist the DAACs with possible recovery issues, the following guides are provided:

1 A core dump of a SubAgent **will not** cause any other servers on a monitored host to core dump. It can, however, result in green icons displayed for servers that died while the SubAgent was down or blue icons for servers for which only partial events were received/sent. The following steps can help recover the state of the OpenView maps for the ECS monitored servers.

2 These incorrect objects can be removed manually by selecting Edit->Delete->From All Submaps after highlighting the erroneous object. It is important to "drill-down" to the lowest level and delete the objects from the lowest level and then from higher levels. This prevents "orphaned" objects from cluttering the OpenView object database ie) program level objects without application level parent objects. If you delete the application level icon first, only the application object is removed. The program and process level object remains in the database and can cause erroneous results in the future.

3 To recycle the custom server monitoring maps in OpenView, select the following from the OpenView menu bar: Misc->ECS Application->Remove all ECS objects. You may need to select this option twice to delete all objects as OpenView occasionally misses objects when deleting a large number of objects. After the deletion, you will need to wait for a rediscovery cycle to complete before all currently running servers and all installed applications are displayed in OpenView. By default, the rediscovery cycle occurs every 20 minutes and is set the SubAgent configuration file as the option MsAg_DiscoverInterval_sec. The object database can become corrupt even when only hardware is being monitored by OpenView without the additional events generated by the ECS custom monitoring. The next step gives instructions on completely cleaning the object database. HP refers to this as a "forced rediscovery".

HP gives guidelines for cleaning the HP OpenView object database, which is the only way to recover the state of the object database if it becomes corrupt. The steps are as follows:

- `kill -15 <EcMsAgDeputy pid>`
- `kill -15 <EcMsCmEcsd pid>`
- `Exit all ovw sessions`
- `Stop all NNM processes using the command 'ovstop' as root`
- `source /opt/OV/bin/ov.envvars.csh (or .sh as dictated by your current shell)`
- `cd $OV_DB/openview`
- `rm -rf defmap mapdb ovwdb topo`
- `cd $OV_LOG`
- `rm xnmevents.* trapd.*`
- `xnmsnmpconf -clearCache`
- `ovstart ovwdb`
- `ovw -fields`
- `ovstart`
- `ovw (run as root to recreate the directory structure)`
- `restart EcMsCmEcsd`
- `restart EcMsAgDeputy`

You will need to wait for a rediscovery cycle to occur after clearing the database for all running servers and installed applications to be displayed again.

7.2 Tivoli Enterprise Console

The Tivoli Enterprise Console (TEC) provides centralized processing and management of distributed events, the ability to allow shared or partitioned administrator responsibilities based on enterprise-defined areas of responsibility, and a flexible interface to view and respond to events based on the events severity, source, location, or other characteristics. The following tables document the Tivoli event configuration.

Table 7.2-1. Disk Event Configuration

Resource	Response Level	Trigger When	Threshold	Response
Inodes Free	Warning	Less than	200	Change icon.
	Severe	Less than	150	Send Tivoli notice. Change icon.
	Critical	Less than	100	Send Tivoli notice. Change icon. Popup alarm.
Inodes Used	Warning	Greater than	X	Change icon.
	Severe	Greater than	X	Send Tivoli notice. Change icon.
	Critical	Greater than	X	Send Tivoli notice. Change icon. Popup alarm.
% Inodes Used	Warning	Greater than	80	Change icon.
	Severe	Greater than	90	Send Tivoli notice. Change icon.
	Critical	Greater than	95	Send Tivoli notice. Change icon. Popup alarm.
Space Free	Warning	Less than	200 MB	Change icon.
	Severe	Less than	100 MB	Send Tivoli notice. Change icon.
	Critical	Less than	50 MB	Send Tivoli notice. Change icon. Popup alarm.
Space Used	Warning	Greater than	X	Change icon.
	Severe	Greater than	X	Send Tivoli notice. Change icon.
	Critical	Greater than	X	Send Tivoli notice. Change icon. Popup alarm.
% Space Used	Warning	Greater than	80	Change icon.
	Severe	Greater than	90	Send Tivoli notice. Change icon.
	Critical	Greater than	95	Send Tivoli notice. Change icon. Popup alarm.
Tivoli DB Free Space	Warning	Less than	20 MB	Change icon.
	Severe	Less than	10 MB	Send Tivoli notice. Change icon.
	Critical	Less than	5 MB	Send Tivoli notice. Change icon. Popup alarm.

Table 7.2-2 Security Event Configuration (1 of 2)

Resource	Response Level	Trigger When	Threshold	Response
Check File Permission:				
/etc/passwd	Critical	Changes from	-rw-r--r--	Send Tivoli notice. Change icon. Popup alarm.
Compare Files:				
Daemon Status:				
amd	Critical	Becomes unavailable		Send Tivoli notice. Change icon. Popup alarm.
biod	Critical	Becomes unavailable		Send Tivoli notice. Change icon. Popup alarm.
cron	Critical	Becomes unavailable		Send Tivoli notice. Change icon. Popup alarm.
inetd	Critical	Becomes unavailable		Send Tivoli notice. Change icon. Popup alarm.
lockd	Critical	Becomes unavailable		Send Tivoli notice. Change icon. Popup alarm.
lpd	Critical	Becomes unavailable		Send Tivoli notice. Change icon. Popup alarm.
mountd	Critical	Becomes unavailable		Send Tivoli notice. Change icon. Popup alarm.
nfsd	Critical	Becomes unavailable		Send Tivoli notice. Change icon. Popup alarm.
portmap	Critical	Becomes unavailable		Send Tivoli notice. Change icon. Popup alarm.
snmpd	Critical	Becomes unavailable		Send Tivoli notice. Change icon. Popup alarm.

Table 7.2-2 Security Event Configuration (2 of 2)

Resource	Response Level	Trigger When	Threshold	Response
statd	Critical	Becomes unavailable		Send Tivoli notice. Change icon. Popup alarm.
File Checksum:				
/etc/passwd	Warning	Not equal to	value	Change icon.
File Size:				
/var/adm/messages	Warning	Greater than	200 Kbytes	Change icon.
Occurrences in File:				
/var/adm/messages	Warning	Greater than	value	Change icon.
Process Instances:				
tivoli	Warning	Greater than	3	Change icon.
HP OpenView	Warning	Greater than	3	Change icon.
User Logins by User:				
root	Warning	Greater than	1	Change icon.
Users Logged in	Warning	Greater than	20	Change icon.
	Severe	Greater than	25	Send Tivoli notice. Change icon.
	Critical	Greater than	30	Send Tivoli notice. Change icon. Popup alarm.

Table 7.2-3. Network Event Configuration (1 of 3)

Resource	Response Level	Trigger When	Threshold	Response
Client RPC timeouts	Warning	% increase of	10	Change icon.
	Severe	% increase of	25	Send Tivoli notice. Change icon.
	Critical	% increase of	50	Send Tivoli notice. Change icon. Popup alarm.
Server Status				
Network Collisions	Warning	% increase of	5	Change icon.
	Severe	% increase of	10	Send Tivoli notice. Change icon.
	Critical	% increase of	25	Send Tivoli notice. Change icon. Popup alarm.

Table 7.2-3. Network Event Configuration (2 of 3)

Resource	Response Level	Trigger When	Threshold	Response
Network Collisions/packets	Warning	% increase of	5	Change icon.
	Severe	% increase of	10	Send Tivoli notice. Change icon.
	Critical	% increase of	25	Send Tivoli notice. Change icon. Popup alarm.
NFS bad calls	Warning	% increase of	10	Change icon.
	Severe	% increase of	25	Send Tivoli notice. Change icon.
	Critical	% increase of	50	Send Tivoli notice. Change icon. Popup alarm.
Input packet errors	Warning	% increase of	10	Change icon.
	Severe	% increase of	25	Send Tivoli notice. Change icon.
	Critical	% increase of	50	Send Tivoli notice. Change icon. Popup alarm.
Input packets	Warning	% increase of	10	Change icon.
	Severe	% increase of	25	Send Tivoli notice. Change icon.
	Critical	% increase of	50	Send Tivoli notice. Change icon. Popup alarm.
Output packet errors	Warning	% increase of	10	Change icon.
	Severe	% increase of	25	Send Tivoli notice. Change icon.
	Critical	% increase of	50	Send Tivoli notice. Change icon. Popup alarm.
Output packets	Warning	% increase of	10	Change icon.
	Severe	% increase of	25	Send Tivoli notice. Change icon.
	Critical	% increase of	50	Send Tivoli notice. Change icon. Popup alarm.

Table 7.2-3. Network Event Configuration (3 of 3)

Resource	Response Level	Trigger When	Threshold	Response
Remote oserv status	Critical	Becomes unavailable		Send Tivoli notice. Change icon. Popup alarm.
RPC bad calls	Warning	% increase of	10	Change icon.
	Severe	% increase of	25	Send Tivoli notice. Change icon.
	Critical	% increase of	50	Send Tivoli notice. Change icon. Popup alarm.

Table 7.2-4. System Event Configuration (1 of 2)

Resource	Response Level	Trigger When	Threshold	Response
Available swap space	Warning	Less than	20 MB	Change icon.
	Severe	Less than	15 MB	Send Tivoli notice. Change icon.
	Critical	Less than	10 MB	Send Tivoli notice. Change icon. Popup alarm.
Host status:				
cyclops	Critical	Becomes unavailable		Send Tivoli notice. Change icon. Popup alarm.
Lingering terminated processes	Warning	Greater than	10	Change icon.
	Severe	Greater than	20	Send Tivoli notice. Change icon.
	Critical	Greater than	30	Send Tivoli notice. Change icon. Popup alarm.
Load average	Warning	Greater than	10	Change icon.
	Severe	Greater than	20	Send Tivoli notice. Change icon.

Table 7.2-4. System Event Configuration (2 of 2)

Resource	Response Level	Trigger When	Threshold	Response
	Critical	Greater than	30	Send Tivoli notice. Change icon. Popup alarm.
Mail queue length	Warning	Greater than	20	Change icon.
	Severe	Greater than	40	Send Tivoli notice. Change icon.
	Critical	Greater than	50	Send Tivoli notice. Change icon. Popup alarm.
Page-outs	Warning	% increase of	50	Change icon.
	Severe	% increase of	80	Send Tivoli notice. Change icon.
	Critical	% increase of	90	Send Tivoli notice. Change icon. Popup alarm.

Table 7.2-5 Printer Event Configuration (1 of 2)

Resource	Response Level	Trigger When	Threshold	Response
Daemon status:				
lpsched	Critical	Becomes unavailable		Send Tivoli notice. Change icon. Popup alarm.
Jobs in print queue:				
print_queue	Warning	Greater than	10	Change icon.
Status of print queue:				
print_queue	Critical	Becomes unavailable		Send Tivoli notice. Change icon. Popup alarm.

Table 7.2-5 Printer Event Configuration (2 of 2)

Resource	Response Level	Trigger When	Threshold	Response
Total size queued:				
print_queue	Warning	Greater than	10 M	Change icon.
	Severe	Greater than	20 M	Send Tivoli notice. Change icon.
	Critical	Greater than	30 M	Send Tivoli notice. Change icon. Popup alarm.

X - system dependent parameters.

7.3 Application Servers

HP OpenView has the capability to monitor the status of each application server and its mode. Through the use of this capability, both Tivoli and Remedy can be invoked from the option bar. These procedures assume that HP OpenView is active and available for service, otherwise follow steps 1-7 of 7.1.1 to start the server.

This procedure explains how to maneuver the HP OpenView Windows graphical user interface for application servers: .

1 Type **/opt/OV/bin/ovw -map** XXXX **& press enter**.

- This command will display the appropriate IP and application server map:
- **XXXX is the name of a previously created map
- This command will display both the hardware map and the services map

2 Double-click **Services** icon

- This command will display the service map, which displays each *application* and *mode* as icons. The colors of the icon indicate the state of the application server – (Select Help – select *Legend* - will display of the color legend)
- This screen can be closed by double-clicking the 1st icon on the icon bar located above the display and below list of choices.
- This command will display the status of each application server

3 Double-click **mode** icon

- This command will display the mode map which has an icon for each mode (Shared, TS1, TS2, and OPS)
- This command will display the status of the modes.

Double-click **on one of the modes** icon

- This command will display the executable module icon for the specific mode selected.
- This command will display the status of the executable modules.

Double-click **mode** icon

- This command will display the mode map which has an icon for each mode (Shared, TS1, TS2, and OPS)
- This command will display the status of the modes.

-- OR --

4 Double-click **on one of the application server** icons

- This command will display the executable application servers map. Each icon represents an executable module.
- This command will display the status of the executable modules.

Double-click **on one of the application servers** icon

- This command will display the executable module icon for the specific mode selected.
- This command will display the status of the executable modules.

Double-click **application server** icon

- This command will display the mode map which has an icon for each mode (Shared, TS1, TS2, and OPS)
- This command will display the status of the modes.

7.3.1 Locate Application/Program icons

1. From the HPOV menu bar, select: Locate->Objects->By Selection Name (or CTRL+N)

2. Type Application/Program ID desired (i.e., 7000004).
   -- Hit "Apply" button and then locate the desired object. The object name has a naming convention of "applID_hostname_instID_mode" for an application object or "progID_applID_hostname_instID_mode" for a program object. You should select the one with an instID of -1, which indicates that the object represents the installed Application/Program.

3. Choose "Open" , then "Select" to highlight the desired icon.

4. Check the label of the installed application/program icon (wheat in color) to verify whether it is for the desired mode.

7.3.2 Start an Application/Program

1. Right-click on installed application/program icon (wheat in color).
2. Select "Start executable".
3. A start executable request will be sent to the Subagent on the selected managed host.
4. A new green application/program icon will be populated onto the map if the application/program has been started correctly.
5. Verify that the desired server programs are started by double-clicking the application icon to go to the program level submap.

7.3.3 Shutdown an Application

1. Right-click on running application/program icon (green in color).
2. Select "Shutdown executable".
3. A shut down executable request will be sent to the Subagent on the selected managed host.
4. The green application icon will be removed from the map if the application has been shut down properly.

7.3.4. Remove Red Application/Program icons (dead executable symbols) from the map

1. Make sure you really want to remove all red Application/Program icons from the map.
2. Select the "Misc. -> ECS Application -> Remove Died Executable Symbols" menu item from the HPOV menu bar.
   -- The word "Sync" will appear at the lower left corner of the OV GUI.
   -- After the "Sync" label disappears, all red ECS Application/Program icons will be removed from the map.

7.3.5 Exit HP OpenView and stop background processes

1. Select Map->Exit (CTRL +E).
   -- Stop any background processes you have started by entering ctrl-c or a "kill" command.

This page intentionally left blank.

8. Problem Management Procedures

ECS Problem Management is administered through system-level and site-level control board reviews. These control boards oversee the analysis, recommendations, and actions taken to resolve ECS system/site problems concerning hardware, software, documentation, and procedures. The M&O and the site-level maintenance organization resolve routine maintenance issues at the system-level and site-level, respectively, using the Trouble Ticket System for tracking maintenance changes. Trouble Tickets (TT) may evolve into Non Conformance Reports (NCR), as required, which may then be utilized to generate Configuration Change Requests (CCR) to effect changes to the approved baseline. To ensure controlled change, NCRs are tracked using the DDTS in the EDF and CCRs manually by Systems Engineering.

The Trouble Ticket System is the first vehicle used to record and report problems with the operational system. Trouble Tickets can be generated by operations, maintenance, development, and customer personnel as well as users. The Trouble Ticket System is an automated database that tracks the resolution activities associated with each trouble ticket. Documentation that is related to the problem, and is not in electronic form, or is in electronic form at the DAAC, is distributed by the local CM Administrator (CMA), and is listed as an attachment to the trouble ticket.

The CM Administrator at each site serves as Trouble Ticket System administrator. Trouble Tickets initially generated at a site, the resolution of which require changes to the system level baseline, are forwarded to the SMC, where they are reviewed and *translated* into NCRs. Additionally, Trouble Tickets, NCRs and CCRs that are generated at the sites, which are repaired locally, and result in site-unique extensions to the system level baseline, are forwarded to the SMC for tracking across the ECS baseline. The SMC CMA is responsible for tracking ECS level TTs after they have been received from the sites, and for propagating system problem resolutions for site visibility.

CMAs also support the activities of the local Trouble Ticket Review Board. This includes generating status reports, and implementing resolutions, instructions, and changes as directed by the Board. User Services Representatives monitor trouble tickets to notify users concerning problem resolution and status. Maintenance engineers at respective levels will record all activities in the trouble ticket. This information can be used to determine critical maintenance concerns related to frequency of occurrence, criticality level, and the volume of problems experienced. The maintainability analysis will guide critical changes, volume and type of support components to be utilized, and will focus further ECS release development.

This section provides an overview of the Trouble Ticketing process and defines the M&O procedures for processing and resolving trouble ticket submissions. In addition, this section provides instructions for diagnosing network problems.

8.1 The Problem Resolution Process — An Overview

8.1.1 ECS Internal Process

Any ECS user may submit Trouble Tickets using the local Trouble Ticketing System (TTS) at any site. TT submission triggers an internal review by the site's review board. Primary objectives of the internal review are to quickly identify and correct problems that fall within the site's capability to maintain, review and validate the priority of the problem, and to elevate to the system level those problems that either exceed capability to repair, or that require a change to the system level baseline.

Problems passed from the sites to the ECS system level, are passed by transferring the trouble ticket from the local TTS to the SMC TTS. Here they are reviewed by the M & O Trouble Ticket Review Board (TTRB), which hosts daily teleconferences, known as the TT Telecon.

The permanent membership of the Trouble Ticket Review Board is as follows:

a. Chair: ECS M&O Manager or designee

b. Each DAAC: one member representative

c. SMC: one member representative

d. ECS Integrated Logistics Support: one member representative

e. Secretary: SMC CM Administrator

f. Sustaining engineering Lead or designee

The roles and responsibilities of the Trouble Ticket Review Board are to:

a. Verify and assign priority to each TT.

b. Assign each TT to an individual or organization for action.

c. Adjust the prioritization and assignment of each TT as required.

d. Review the status of all TTs periodically.

e. Collect and report TT statistics and metrics.

f. Direct the preparation on NCRs, as required.

g. Refer TTs from SMC Remedy to ECS DDTS as NCRs

The TTRB performs a preliminary review of each trouble ticket to confirm the priority assigned by the site, the completeness of information and data relevant to the problem, and whether it requires a change to the system level operational baseline.

The TTRB may forward TTs to the appropriate organizations or individuals for further analysis and technical investigation, resolution proposal, and NCR preparation, when required. The CM Administrator at the SMC provides administrative support to the TTRB by publishing minutes of teleconference meetings, tracking TTs and action items, and by translating TTs to NCRs.

The TTRB has the authority to direct resolutions to trouble ticket problems that do not change, or in any way affect, the ECS operational baseline and baseline documentation. An NCR is required when the Technical Investigation (TI) determines that the operational baseline must be changed in order to correct the problem identified in the trouble ticket. When complete, the TTRB reviews the NCR for completeness, as well as cost and schedule impact. The TTRB then forwards the NCR to the M&O CCB.

The TTRB is not a voting board; the membership is appointed for the purpose of providing timely, direct technical support to the Chair, who has the decision making responsibility and authority.

The M&O CCB has the authority and responsibility to approve Class II changes to the operational baseline. Specific responsibilities include:

a. Review, approve and schedule; review and backlog; or reject each NCR's proposed resolution, or cost and schedule input form the Responsible Engineer (RE).

b. Approve the schedule for the deployment of configuration changes in the form of a 'drop' to the SMC.

c. Approve the content of each block.

d. Manage and adjust the schedule and contents of each block in accordance with ECS SDPS Program priorities and the progress of NCR work-off.

e. Review the status of all backlogged NCRs on a periodic basis. Schedule NCRs for a future block as appropriate

f. Collect and report on NCR statistics.

The permanent membership of the M&O CCB is as follows:

a. Chair: ECS M&O Manager of designee.

b. Each DAAC: one representative.

c. SMC: one representative.

d. System Engineering Department: one representative

e. Development Department: one representative.

f. The Patch IPT (or Test Department): one representative.

g. ECS Integrated Logistics Support Department: one representative.

h. Secretary: TBD

The M&O CCB reviews each NCR as received from the TTRB for technical merit, completeness, priority and complexity. It may direct additional technical investigation and analysis, or assign immediately for resolution. The Responsible Engineer prepares a proposal which will include:

a. A short, narrative description of the change that identifies the object or objects, configuration files, COTS SW, documentation, or scripts that require change. The nature of the change must also be included in the narrative.

b. A cost estimate where:

   1. LOW requires fewer than 5 staff-days of effort

   2. MEDIUM requires 6 to 20 Staff-days of effort

   3. HIGH requires more than 20 staff-days of effort

c. The date by which the change could be ready for integration and deployment, given the RE's knowledge of tasking, priorities, development activities, etc.

The M&O CCB is not a voting board. The CCB Chair has the singular authority for decision making. Other board members are appointed to provide direct and timely technical support to the CCB's decision making process.

In accordance with ECS Configuration Management procedures, the RE, as directed by the CCB, makes the necessary changes to the system baseline, including the ECS baseline documentation changes to support the resolution. These changes are then integrated into an M&O block by the Patch IPT.

The Patch IPT integrates and tests each block to verify that the system still meets functional and performance requirements, that the integrated block corrects the NCR(s), and that new features execute properly. The status of each block passing through integration and test is reported to the CCB regularly by the Patch IPT.

Upon satisfactory completion of the test program, the M&O CCB may authorize deployment of the block to the DAACs. The SMC pulls the block from the EDF and saves it to a locked-down directory. The SMC then notifies the DAACs that the block is available for transfer. Each DAAC desiring the block notifies the SMC of the transfer path, and the SMC pushes the block as requested. DAACs may then install the block and keep the SMC apprised of the state of each mode.

8.1.2 Interface with ESDIS

The EOSDIS Sciences Systems Configuration Management Board CCB must approve all ECS Class I CCRs before work is authorized to commence. (See the *Sciences Systems Configuration Management Board (PCMB) Configuration Management Plan (level 3)* for discussion of change classification.)

Should the RE recommend, and the M&O CCB concur, that a CCR is an enhancement rather than a maintenance change, the M&O CCB will place the CCR in backlog after reviewing and approving the technical solution, cost and schedule impact. The CCR is then sent to the PCMB for approval. When approved by the PCMB and returned to the M&O CCB, the CCR is assigned to an RE for implementation.

When the PCMB or the ESDIS CCB forwards a new CCR as an enhancement to the M&O CCB, the CCR is assigned to an RE by the CCB. The CCB approved technical, cost and schedule assessments of the RE are sent by the M&O CCB back to the PCMB.

8.2 Problem Management Procedures

The Trouble Ticket System is comprised of the Remedy Action Request System, a Commercial Off-The-Shelf (COTS) product that provides a distributed trouble ticketing service which provides a common environment and means of classifying, tracking, and reporting problem occurrence and resolution to both ECS users and operations personnel. The trouble ticketing service:

- Provides a GUI for operations personnel to access all Trouble Ticket services.
- Provides a common Trouble Ticket entry format.
- Stores Trouble Tickets.
- Retrieves Trouble Tickets via ad hoc queries.
- Allows operations personnel to forward problems from one DAAC to another.
- Produces stock and common reports.
- Provides an interface to user's and operator's e-mail to provide automatic notification.
- Offers an application programming interface through which applications can submit trouble tickets.
- Provides summary information to the SMC from each DAAC to allow trend reports regarding trouble tickets.
- Defines a consistent "life cycle" for trouble tickets.
- Allows each DAAC a degree of customization through definition of further re-prioritization and action rules.

In addition to the functionality provided by Remedy's Action Request System, the Trouble Ticketing Service utilizes a set of custom HTML pages ("screens") to provide registered users with the ability to submit new trouble tickets and query the current status of any of their previous entries. Access to the Trouble Ticketing System through this technique provides users an easy method for reporting problems in an environment with which most are already familiar. Additionally, as another means of trouble ticket entry, the Trouble Ticket System provides a text e-mail template through which automated entry of trouble tickets is possible. Support staff members are able to enter Trouble Tickets through the TTS interface for problems received via other methods (for example, phone calls).

The Remedy Action Request System also functions as the User Contact Log. Remedy's Action Request System is configured to have a separate schema that contains the entries that User Services personnel enter for each contact that they receive from a user. The User Contact Log allows a trouble ticket to be initiated from a log entry: with the push of a button — the Trouble Ticket will be populated with information from the contact log.

Users submit trouble tickets to the User Services Desk. External users submit trouble tickets through the Internet [using a series of hypertext mark-up language (HTML) screens. Site personnel submit trouble tickets via the Trouble Ticket System (Remedy).

User Services personnel process trouble tickets through the Trouble Ticket System for problem resolution in accordance with local policy. Trouble tickets are first evaluated to determine the severity of the problem and assignment of on-site responsibility. Every trouble ticket is logged into the database for record keeping purposes. Trouble tickets that can be resolved locally are assigned and tracked at the local center. The Operations Supervisor reviews each trouble ticket for priority verification and problem description; and assigns it to an appropriate Maintenance Engineer for resolution.

Matters that require external or higher level assistance; and problems, the repair of which require changes to the system baseline; are escalated to M&O via the Trouble Ticket Review Board (TTRB) Telecon for discussion and disposition. The telecon is held to coordinate trouble ticket activities within the M&O organization as well as with development, customer, and user organizations.

The SMC CM Administrator is responsible for preparing and disseminating the agenda and the minutes for the TTRB. Electronic dissemination via Internet e-mail is preferred. Agenda items may be supplemented or replaced by hardcopy or softcopy reports. Material from this meeting is distributed within each ECS organization and to customer and user organizations as required. A typical agenda might include:

- Review and prioritize each trouble ticket opened at each center.
- Review and re-prioritize older trouble tickets (as required).
- Assign trouble ticket work-off responsibility to one organization.
- Review distribution of trouble tickets by organization, priority and age.
- Discuss trouble ticket issues with development organizations.

8.3 Using the Trouble Ticket System

1. User or Operator discovers a problem with ECS (hardware, software, documentation, procedure) and documents this problem for later resolution. The submitter forwards a trouble ticket to User Services by: calling up the Trouble Ticket System via the Internet; going on-line with the Trouble Ticket System; phoning User Services; or sending an e-mail message to the Trouble Ticket System.

2. The trouble ticket is logged into the system. TTS automatically assigns "New" status to the trouble ticket and notifies the Operations Supervisor for assignment and prioritization. TTS notifies the Operations Supervisor via email, or through Remedy's notification tool, or both. The status of each trouble ticket, as it progresses through the resolution process, is recorded by the CMA.

3. The trouble ticket log is prioritized according to a triage system of maintenance priorities which determines priorities in relation to the effect of a problem on mission success, and is differentiated by scope of impact, frequency of occurrence, and the availability of an adequate work-around. The *Performance Assurance Requirements* document, NASA 420-05-03, identifies problem categories which correspond to the triage system of maintenance priorities:

Priority 1: System/Service cannot perform critical function or imposes major safety hazard. (TTS Priority: High)

A High priority trouble ticket is a persistent problem for which no workaround exits or a trouble for which no workaround can be accommodated by DAAC operators given a detailed workaround procedure is documented but the procedure is inadequate based upon the complexity of the procedure, the abilities of an adequately trained and experienced operator, or both; and the consequence of the occurrence causes:

- the unrecoverable loss of data,
- the system to be unable to ingest, process or distribute data
- the system to be unable to support user searches for available data, or

  the operator to be unable to startup, shutdown or determine the status of

  system components.

Priority 2: System/Service substantially impaired. (TTS Priority: Medium)

A Medium priority Trouble ticket is a trouble for which a workaround exits but the frequency of occurrence is expected to be more than approximately once per day, the occurrence cannot be anticipated, and the impact is such that system performance is degraded to a point that there is reasonable risk that 24 hours of work cannot be accommodated within a 24 hour period.

Priority 3: System/Service slightly impaired. (TTS Priority: Low)

Causes minor or no substantial impact to development, operations, services, or data processing functions. Support may be degraded, but mission can still be accomplished.

The Trouble Ticket System (TTS) has coded these three priorities as HIGH, MEDIUM, and LOW. All trouble ticket submittals are required to designate a priority level for the

problem. However, the formal priority is assigned by the Operations Supervisor and maintained by the CM Administrator. M&O applies these additional priorities:

Priority 4: Nuisance Problem: Problems that do not impair the capability of the ECS, but rather could simplify the it's use, such as the arrangement of video screens, color, and so on.

Priority 5: Enhancement: Problems, the resolution of which result in enhanced system capability, or are out-of scope (Class I problem).

4. All affected Operations Supervisors at the sites (SMC, DAACs, EOC, EDF) are notified by e-mail of the problem and solicited for inputs to problem assessment (impact) and resolution.
5. The Trouble Ticket database is updated by the Operations Supervisor, and the trouble ticket may be modified to reflect this new information/coordination activity.
6. The Operations Supervisor assigns the problem to a Problem Investigator for further follow-up
7. The Problem Investigator coordinates input from SEO, developers, vendors, and external organizations to effect the local resolution. The Problem Investigator presents significant issues at the TTRB Telecon.
8. The Problem Investigator updates the trouble ticket database.
9. The Problem Investigator forwards any information regarding proposed/implemented fixes to the established notification list.
10. In those cases where the problem resolution will result in a change to the operational baseline, or where the local site wishes to elevate the TT to M&O for advice or for resolution, the site CMA forwards the TT to the SMC
11. The proposed resolution is then presented to the Trouble Ticket Review Board (and Government Failure Review Board for Priority 1 and Priority 2 problems) for review, ratification, or revision.
12. Changes that do not affect controlled configuration items may be approved and implemented by the Failure Review Board/Trouble Ticket Review Board and closed.
13. All system level changes are proposed in Non Conformance Reports (NCRs). The SMC is the custodian of a translation utility which, when invoked, opens an NCR in the Change Request Manager, and closes the trouble ticket in TTS. Emergency fixes (Priority 1) can be made locally with the approval of the local CCB, and then reported to M&O after the crisis is resolved. The M&O CCB may approve, reject, defer or revise the NCR.
14. The off-site problem resolution process is monitored by the M&O Trouble Ticket Review Board, which may also revise the proposed solution because of any system-level effect(s).
15. The NCR may be escalated to higher level CCBs for system and/or external elements that may be involved in the resolution process.

Table 8.3-1. Trouble Ticket System - Activity Checklist

Order	Role	Task	Section
1	ECS users	Access the Trouble Ticket System	8.3.1, 8.4.1
2	ECS users	Submit Trouble Ticket	8.3.2
3	Maintenance Engineer	Modify Open Trouble Ticket	8.3.3
4	Operations Supervisor, Maintenance Engineer	Forward Trouble Ticket	8.3.4
5	Database Administrator	Add Users to TTS	8.3.5
6	CMA	Modify TTS User Privileges	8.3.6
7	CMA	Modify TTS' Configuration	8.3.7
8	Maintenance Engineer, CMA	Generate Reports	8.3.8
9	Maintenance Engineer	Maintain Escalation Time Table	8.3.9

8.3.1 Accessing the Trouble Ticket System

The Trouble Ticket System may be accessed through either HTML or Remedy. The Trouble Ticket HTML is used by both User Services and the end user to submit trouble tickets without going through Remedy. It is accessed by clicking on the Trouble Ticket icon. Through HTML, the user can submit, obtain a list, and view details of trouble tickets. Complete and detailed instuctions for Remedy may be found in the current DID 609-CD and *Remedy's Action Request System Users's guide.*

Through Remedy, the User clicks on the User Tool icon, which opens the RelB-Trouble Tickets schema to submit, query, or work a Trouble Ticket. The Main Remedy Trouble Ticket screen is used to select the appropriate schema for submitting, modifying, or displaying a trouble ticket. The Main Page data fields are identified in Table 8.3.1-1.

The Remedy Action Request System provides a distributed Trouble Ticketing Service that furnishes DAACs a common environment and the means of classifying, tracking, and reporting problem occurrences and resolutions to both ECS users and operations personnel. The Trouble Ticketing Service:

· provides a GUI for operations personnel to access all Trouble Ticket services

· provide a common Trouble Ticket entry format

· stores Trouble Tickets

· retrieves Trouble Tickets via ad-hoc queries

· allows operations personnel to forward problems from one DAAC to another

· generates reports and statistics

· interfaces with user's and operator's e-mail to provide automatic notification

· offers an application programming interface through which applications can submit Trouble Tickets

· provides summary information to the SMC from each DAAC to allow trend reports regarding Trouble Tickets

· enables operations personnel to forward a copy of a "closed" trouble ticket to the SMC for insertion into the ECS Closed Trouble Ticket Database

· defines a consistent "life-cycle" for Trouble Tickets

· allows each DAAC a degree of customization through definition of further escalation and action rules.

Several time-saving features are available through Remedy: the Admin Tool, GUI Import tool, the Hardware Information schema, and the GUI Notification tool. Brief descriptions are provided in Sections 8.3.1.1 through 8.3.1.4.

Table 8.3.1-1. RelB-Trouble Ticket Field Description (1 of 2)

Field Name	Data Type	Size	Entry	Description
Ticket-Id	Character	15	System generated	Ticket number which is set and maintained by the system
Ticket Status	Selection	4	Required	Status of the Trouble Ticket
Assigned-Priority	Selection	4	Required	Priority of Trouble Ticket assigned at the site (HIGH, MEDIUM, LOW)
Short Description	Character	128	Required	Short Description of the problem
Submitter Impact	Selection	4	Optional	Impact of the problem to the submitter
Long-Description	Character	255	Optional	Long Description of the problem
Resolution Log (End User Sees)	Diary	Unlim	Optional	General steps in the resolution of the problem
Detailed Resolution Log	Diary	Unlim	Optional	Detailed steps in the resolution of the problem
Submitter ID	Character	30	Required	User Id of the Submitter.
Submitter Name	Character	30	Optional	Full Name of the Submitter
Submitter Phone	Character	30	Optional	Phone number of the Submitter
Submitter e-mail	Character	64	Optional	E-mail address of the Submitter
Submitter Home DAAC	Character	60	Optional	Home DAAC of the Submitter
History	Diary	Unlim	Optional	Upon submission or modification, the person assigned to the ticket and the ticket status will be indicated in the History field. Due to a limitation in Remedy, this information will only be written when the Assigned-to and Status fields are modified
Assigned-To	Character	30	Optional	Person that Trouble Ticket has been assigned to
Last-modified-by	Character	30	System generated	Person that last modified the Trouble Ticket
Create-date	Date/Time	4	System generated	Date Trouble Ticket was created at the present site
Last-Modified-date	Date/Time	4	System generated	Date the Trouble Ticket was last modified
Related CCR	Character	60	Optional	ID of a related CCR
Key Words	Character	255	Optional	Key words to help identify this Trouble Ticket
Problem Type	Character	30	Optional	Type of problem addressed by this Trouble Ticket
Closing Code	Character	60	Optional	Origin of the problem that this Trouble Ticket resulted from

Table 8.3.1-1. RelB-Trouble Ticket Field Description (2 of 2)

Field Name	Data Type	Size	Entry	Description
Closed-by	Character	60	Optional	Person that closed this Trouble Ticket
Close-date	Date/Time	4	Optional	Date this Trouble Ticket was closed
Software Resource	Character	60	Optional	Software Resource that the problem came from
Hardware Resource	Character	60	Optional	Hardware Resource that this problem came from
Duplicate Master Id	Character	25	Optional	The Master Ticket-ID of this Trouble Ticket
Forward-to	Character	60	Optional	Site that this Trouble Ticket was last forwarded to
Forwarded-from	Character	60	Optional	Site that forwarded this Trouble Ticket
Forwarded-by	Character	60	Optional	Contact person at the forwarding site
Forward-date	Date/Time	4	Optional	Date Trouble Ticket was forwarded
Unique-Identifier	Character	20	Optional	Unique identifier which is established at the origination site This identifier should NEVER be changed once set
Forwarded-to-1	Character	60	Optional	First site to have been forwarded this Trouble Ticket
Forwarded-to-2	Character	60	Optional	Second site to have been forwarded this Trouble Ticket
Forwarded-to-3	Character	60	Optional	Third site to have been forwarded this Trouble Ticket
Forwarded-to-4	Character	60	Optional	Fourth site to have been forwarded this Trouble Ticket
Associated Contact Log Id	Character	30	Optional	ID number of the Associated Contact Log

8.3.1.1 Remedy's GUI Admin Tool

The Admin Tool is used to notify or set fields as soon as the trouble ticket reaches a particular state or to escalate the problem once a trouble ticket is in a particular state too long. This tool is accessed in two ways:

1. By clicking on Admin Tool to open correct filter, escalation, or active link. (Problem escalation is discussed in Section 8.3.9.)
2. By clicking on User Tool icon and opening RelB-TT-Times schema to review/modify a Trouble Ticket.

For more information on the Admin Tool, refer to the Remedy Administration Manual.

8.3.1.2 Remedy's GUI Import Tool

The GUI Import tool is used to import existing entries rather than retyping information manually. It also enables the user to import entries into a schema from a file generated by the Admin tool. This tool is accessed by clicking on the Remedy Import Tool icon. For more information on the Import tool, refer to the Remedy User Guide.

8.3.1.3 Remedy's Hardware Information Schema

If detailed hardware information needs to be provided beyond what can be entered on the Trouble Tickets schema. The User Tool, Hardware Information schema, provides the vehicle to add a description of a hardware problem that corresponds to a trouble ticket. Through this schema, the user can enter detailed information about failed hardware components (e.g., part and serial numbers) and the actions taken to correct the problem. This schema is accessed by clicking on the User Tool icon and opening RelB-Hardware Information schema, or via Hardware Information link from Trouble Tickets schema.

8.3.1.4 Remedy's GUI Notification Tool

The GUI Notification Tool is used as an alternative to email notification to notify the user of a Remedy event. This tool is accessed by clicking on the Remedy Notification Tool icon. It allows properties and options to be modified via pull-down menus. Examples of GUI notification include a beep, a pop-up window, a flashing message. In addition, both an email and a GUI notification can be sent if the site so desires.

8.3.2 Submit a Trouble Ticket

When a problem is either found by or reported to User Services, follow the procedure applicable to your system, to create and log trouble tickets. Trouble tickets can be submitted via HTML or via Remedy's user tool – RelB-Trouble Tickets schema. Remedy's Contact Log schema is used to classify, track, and report contacts of ECS users and operators and also to submit a trouble ticket from a log entry. E-mail is another method of submitting a trouble ticket. The template is available from your System Administrator.

1. For HTML submission:
   a) Access HTML Trouble Ticketing Main page.
   b) Select Submit link which opens the Submit page.
   c) Fill out the impact, short description, and detailed description fields.
   d) Select Submit.
2. For submission through Remedy (See the Remedy User Guide, Chapter 3, “Submitting an Action Request” for the general steps):
   a) Access Remedy User Tool (See the Remedy User Guide, Chapter 2, “Getting Started with the User Tool”, page 2-3, section on “Starting the User Tool”).

b) Access RelB-Trouble Ticket schema (See the Remedy User Guide, Chapter 2, "Getting Started with the User Tool", page 2-18, section on "Using Schemas").

c) Select Open Submit from the File menu.

d) Fill out at least Short Description and User Identification.

e) Fill out any other pertinent fields.

f) Select Apply.

3. For submission from a Remedy Contact Log entry:

a) Click on User Tool icon and open RelB-Contact Log schema.

b) Fill out Contact Log ID and Contact Information. If the contact is a registered Remedy user, the contact information is filled out automatically.

c) Fill in Short Description (limit is 128 characters).

d) Click on **Create TT** button.

4. For submission via E-mail:

a) Obtain Template from your System Administrator.

b) Address the message to arsystem@_____._____._____.

c) Copy template into message area. DO NOT INCLUDE AS AN ATTACHMENT. DO NOT ALTER TEMPLATE. The template is presented in Figure 8.3.2-1. The # sign indicates comments, which are not read by Remedy. **Enter data as indicated in Figure 8.3.2-1.** Send message.

```
#
#  File exported Wed Feb 28 19:01:27 1996
#
Schema: RelB-Trouble Tickets
Server: remedy server name
Login:
Password:

       Short Description !        8!:
        Submitter Impact !536870922!: Low
# Values: Low, Medium, High
        Long-Description !        9!:
            Submitter ID !        2!:
          Submitter Name !536870917!:
         Submitter Phone !536870918!:
         Submitter e-mail !536870921!:
     Submitter Home DAAC !536870919!:
```

Field ID internal to Remedy

Default value

Select one

Enter data after colon

Figure 8.3.2-1. Trouble Ticket E-mail Template

8.3.3 Reviewing and Modifying Open Trouble Tickets

Trouble tickets may need to be modified based on better understanding of the nature of problems defined and revised resolutions from the Maintenance Engineer investigations, Sustaining Engineering inputs, Developer inputs, Trouble Ticket Review Board decisions, Change Control Board decisions, and/or Failure Review Board decisions. The results will be factored into revisions and/ or additions to the Trouble Ticket log.

1. For HTML Review and Modification of Trouble Tickets:

   a) Access HTML Trouble Ticketing Main (see Section 8.4). Trouble Tickets can be ***submitted, queried or modified***.

   b) Select List link which opens the List page and shows each Trouble Ticket's Identification, Short Description, and Status.

   c) Select the Trouble Ticket Id to get a more detailed description of that particular Trouble Ticket.

2. For Reviewing and Modifying Trouble Tickets through Remedy (See the Remedy User Guide, Chapter 4, "Reviewing and Modifying Action Request" for the general steps):

   a) Access Remedy User Tool (See the Remedy User Guide, Chapter 2, "Getting Started with the User Tool", page 2-3, section on "Starting the User Tool").

   b) Access Release B-Trouble Ticket schema (See the Remedy User Guide, Chapter 2, "Getting Started with the User Tool", page 2-18, section on "Using Schemas").

c) Select List from the Query menu.

d) From the List pick the Trouble Ticket(s) that you would like to Review/Modify.

e) Select Modify Individual from the Query menu of the List window review and modify Trouble Ticket.

8.3.4 Forwarding Trouble Tickets

Trouble ticket administrative reports are forwarded for local and system-wide usage. The trouble ticket contains all forwarding information; once forwarded, it goes to the RelB-TT-ForwardToSite holding area (transparent to the user). The RelB-TT-Sites schema is used to indicate the site name and email address to be used in forwarding. To forward a trouble ticket:

1. Click on User Tool icon and open RelB-Trouble Ticket schema.
2. Set the status to "Forwarded".
3. Select a value for the Forward-to field from its picklist.
4. Select the Forward button.
5. Select Apply.

If necessary, a site name and email address can be modified, added, or deleted to update the picklist of Release B sites by authorized Remedy users: DAACs, SMC, NSI, EBnet. To modify picklist:

1. Click on User Tool icon and open RelB-TT-Sites schema.
2. Modify data.

8.3.5 Adding Users to Remedy

The database administrator uses the RelB-User schema to grant access to the Remedy tool. See Remedy Administrator's Guide for OSF/ Motif, Chapter 3, "Setting Up Users and Groups", page 3-11, section on "Adding Users". Users who change jobs can be deleted.

8.3.6 Changing Privileges in Remedy

This procedure is used by the CM Administrator to control privileges of those who have been granted access. For more information, refer to the Remedy Administrator's Guide.

NOTE: No group should be modified without proper configuration change approval.

To change Privileges in Remedy:

1. See Remedy Administrator's Guide, Chapter 3, "Setting Up Users and Groups", page 3-2, section on "Understanding Access Control".
2. See Remedy Administrator's Guide, Chapter 3, "Setting Up Users and Groups", page 3-4, section on "Access Control Groups".

3. Groups have already been created to accommodate all privileges needed by Remedy Users for Release B. These groups are identified in see Table 8.3.6-1.

Table 8.3.6-1. Table of Access Control Groupings

Groups	Description	Access Type
Operator	Submits trouble ticket internally.	Change
User Services	Submits trouble ticket internally for user.	Change
Operations Supervisor	Assigns problem priority and resolution responsibility. Can forward trouble ticket to another site.	Change
Resource Manager	Assigns problem priority and resolution responsibility. Can forward trouble ticket to another site.	Change
Resolution Technician	Attempts to resolve problem.	Change
Trouble Ticket Review Board Chair Person	Reviews proposed solutions	Change
Administrator	Adds groups and users. Changes permissions. Sets escalation times. Sets menu items. Etc.	Change
Sub-Administrator	Same functions as Administrator but only with certain Schemas.	Change
Browser	Read only permission.	Read
Customize	Can use all features of the customize facility.	Change
Submitter	Place holder for anyone that submits a trouble ticket.	NA
Assignee	Place holder for anyone that is assigned a trouble ticket.	NA
Public	Read only permission. Guest users are automatically put in this group.	Read
NotifyNewEscal	Everyone that will be notified on an escalation due to trouble ticket being in “New” status.	Read
NotifyAssignedEscal	Everyone that will be notified on an escalation due to trouble ticket being in “Assigned” status.	Read
NotifySolPropEscal	Everyone that will be notified on an escalation due to TT being in “Solution Proposed” status.	Read
NotifyImpSolEscal	Everyone that will be notified on an escalation due to trouble ticket being in “Implement Solution” status.	Read
NotifySolImpEscal	Everyone that will be notified on an escalation due to TT being in “Solution Implemented” status.	Read

8.3.7 Modifying Remedy's Configuration

RelB-Trouble Ticket schemas' pulldown menus can be customized. Customization is achieved through the User Tool by modifying the RelB-Menu-Closing Codes, RelB-Menu-Hardware Resources, RelB-Menu-Software Resources, RelB-Menu-Key Words, RelB-Menu-Problem Type, Sites schemas.

To modify the Remedy environmental variables, refer to the Remedy User's Guide and Remedy Administrator's Guide as indicated.

1. See Remedy User's Guide, Chapter 7, "Customizing the Environment."
2. See Remedy Administrator's Guide, Chapter 1, "Using the Administrator Tool."

   NOTE: No administrative configuration should be made without proper configuration change approval.

8.3.8 Generating Trouble Ticket Reports

A set of predefined reports will be placed in a public directory that should be downloaded to your personal configuration directory (see the Remedy User Guide, Chapter 2, "Getting Started with the User Tool," page 2-31, section "Sharing Macros, User Commands and Custom Reports," sub-section "Copying Files"). These reports are trouble ticket administrative reports generated for local and system-wide usage. See Remedy User's Guide, Chapter 5, "Reports."

8.3.9 Re-prioritization of Dated Trouble Ticket Logs

Remedy provides automated prioritization of trouble tickets based on delinquency status of out-dated trouble ticket logs. The File Tickler System automatically assigns higher priority to promote timely resolution.

1. Access Remedy User Tool (See the Remedy User Guide, Chapter 2, "Getting Started with the User Tool," page 2-3, section on "Starting the User Tool").
2. Access RelB-Times schema (See the Remedy User Guide, Chapter 2, "Getting Started with the User Tool", page 2-18, section on "Using Schemas").
3. Select List from the Query menu.
4. From the List pick the Time(s) that you would like to Review/Modify.
5. Select Modify Individual from the Query menu of the List window to review and/or modify the Time (in seconds).

8.4 Using Hypertext Mark-up Language (HTML) Screens

The hypertext mark-up language (HTML) Trouble Ticket Main Screen ("ECS Trouble Ticketing: Menu") provides an introduction on how to use the Trouble Ticketing HTML, and is used by registered ECS users to go to either the Submit page or List page.

Selecting **Submit a Trouble Ticket** will bring up the Trouble Ticketing Submit screen.

Selecting **List the [username] Trouble Tickets** will bring up the Trouble Ticketing List screen.

Help on the Trouble Ticket HTML screens is available by clicking on the Trouble Ticket Help icon at the bottom of the screen .

8.4.1 ECS Trouble Ticketing HTML Submit Screen

The HTML Trouble Ticket Submit screen is used by registered ECS users to submit a Trouble Ticket.

Table 8.4.1-1 below provides a description of the Trouble Ticket HTML Submit Screen fields.

Table 8.4.1-1. Trouble Ticket HTML Submit Screen Field Description

Field Name	Data Type	Size	Entry	Description
ID	character	30	System generated	Submitter Id
Name	character	30	System generated	Submitter Name
E-mail address	character	64	System generated	Submitter E-mail Address
Phone	character	30	System generated	Submitter Phone Number
Home DAAC	character	60	System generated	Submitter Home DAAC
Impact	selection	4	Required	Impact to Submitter
Short description	character	125	Required	Short description of problem
Detailed problem description	character	245	Optional	Long description of problem

When the information is completed, the user can submit the Trouble Ticket by clicking on the **Submit** button on the lower half of the screen. The Problem Information Fields can be cleared by clicking on the **Reset** button. The user also has the choice of returning to the Trouble Ticketing Homepage or going to the Trouble Ticket Help screen by clicking on the respective icons at the bottom of the page.

8.4.2 ECS Trouble Ticketing HTML Success Screen

The HTML Trouble Ticket Success screen is used by registered ECS users to ensure successful submission and report Trouble Ticket Id.

From this screen, the user is provided with the following information/options:

- Confirmation that the trouble ticket was successfully submitted, the trouble ticket identification number, and who submitted the trouble ticket.
- Notification that an E-mail message has been sent to the user indicating that a Trouble Ticket has been submitted and when it was closed. Selecting this Trouble Ticket will open the Trouble Ticket Detailed Screen.
- Instructions telling the user how to check the progress of Trouble Ticket resolution.

The user also has the choice of returning to the Trouble Ticketing Homepage or going to the Trouble Ticket Help screen by clicking on the respective icons at the bottom of the page.

8.4.3 ECS Trouble Ticketing HTML List Screen

The HTML Trouble Ticket List screen is used by registered ECS users to List Trouble Tickets for a user and links the listed Trouble Ticket Number to the Trouble Ticket Detailed Screen.

Table 8.4.3-1 below provides a description of the Trouble Ticket HTML List Screen fields.

Table 8.4.3-1. Trouble Ticket HTML List Screen Field Description

Field Name	Data Type	Size	Entry	Description
Trouble Ticket Number	character	15	System generated	Trouble Ticket Id
Problem Short Description	character	125	System generated	Short Description of Problem
Status	character	20	System generated	Status of Trouble Ticket

The user also has the choice of returning to the Trouble Ticketing Homepage or going to the Trouble Ticket Help screen by clicking on the respective icons at the bottom of the page.

8.4.4 ECS Trouble Ticketing HTML Detailed Screen

The HTML Trouble Ticket Detailed screen is used by registered ECS users to see a more detailed output of a Trouble Ticket.

Table 8.4.4-1 below provides a description of the Trouble Ticket HTML Detailed Screen fields.

Table 8.4.4-1. Trouble Ticket HTML Detailed Screen Field Description

Field Name	Data Type	Size	Entry	Description
ID	character	30	System generated	Submitter Id
Name	character	30	System generated	Submitter Name
E-mail address	character	64	System generated	Submitter E-mail Address
Phone	character	30	System generated	Submitter Phone Number
Home DAAC	character	60	System generated	Submitter Home DAAC
Status	selection	4	System generated	Status of Trouble Ticket
Impact	selection	4	System generated	Impact to Submitter (low, medium, high)
Short description	character	125	System generated	Short description of problem
Detailed problem description	character	245	System generated	Long description of problem
Log	character	unlim.	System generated	Diary of problem resolution

The user also has the choice of returning to the Trouble Ticketing Homepage or going to the Trouble Ticket Help screen by clicking on the respective icons at the bottom of the page.

8.4.5 ECS Trouble Ticketing HTML Help Screen

The HTML Trouble Ticket Help screen is used by registered ECS users to get help with the HTML screens.

This screen provides general information on the following:

- Index -- links that scroll the screen to the Introduction, Submit Page, and List Page sections listed below.
- Introduction – provides information about the Trouble Ticket Help page
- Menu Page – describes the Trouble Ticketing Menu page.
- Submit Page – describes the Trouble Ticket Submit page.
- Success Page – describes the Trouble Ticket Success page.
- List Page – describes the Trouble Ticket List page.
- Detailed Page - describes the Trouble Ticket Detailed page.

8.5 Emergency Fixes

Emergencies may be in real time with the understanding that the Trouble Ticket System must be brought up-to-date as soon as possible after implementing the repair. The example presented below, involves a hardware failure. The problem needs to be resolved quickly to bring a system back into operation. The resolution requires emergency replacement of a component that is of a later version than is contained in the original equipment. The scenario is summarized in Table 8.5-1.

Scenario — An Example of an Emergency Change Procedure

It is 7:00 on a Saturday evening. The DAAC operator detects a problem with the automated tape library (ATL) and reports the problem to the Trouble Ticket System. The trouble ticket is routed to the System Administrator, who confirms that the system will not operate and notifies the site Maintenance Engineer. After running further diagnostics, the Maintenance Engineer reports the problem and symptoms to the OEM's maintenance desk. The original equipment manufacturer (OEM) maintenance representative arrives and concludes that a controller card has failed. The only card the OEM has immediately available is of a later version and no spares are available on site. It will be Monday at the earliest before a replacement board of the same revision level can be located. The site maintenance engineer reports this to the operations Crew Chief (i.e., shift leader) for a decision.

The DAAC cannot afford to have the ATL down until Monday. The Crew Chief calls the DAAC manager at home, apprises him of the situation, and obtains approval to replace the board with the later version if tests conclude that it works properly. The OEM's maintenance representative installs the board. The site's sustaining engineer tests the new controller board, finds that it works properly, and brings the ATL back on-line. The sustaining engineer updates the trouble ticket to document the configuration change and the authority for the change, and forwards it to the site CMA. The site Maintenance Engineer updates the property record with the model, version, and serial number of the new board.

The site CM Administrator reviews the trouble ticket, and presents it to the local CCB for approval. The CMA then updates the Baseline Manager with the new configuration and TT number authorizing the change. At this point, the site is operational at variance from the system baseline, i.e., site unique, and is at risk of loosing maintenance support from M&O.

The site CMA forwards the trouble ticket to the SMC, presented to the TTRB for priority review and is solved or translated into an NCR. The M&O TTRB reviews all emergency TTs to assess whether there may be impacts to the ECS and/or applicability to other sites. The M&O CCB monitors all open NCR promotion and approves them for closure.

In the event that it is later discovered that the new version controller board has adverse impacts when operating in the ECS configuration, a board of the original version will have to be obtained to replace the newer version. In such cases, the action will be recorded on a new trouble ticket, citing the previous CCR.

Table 8.5-1 summarizes emergency procedures that might be taken during an after-hours, over-the-weekend emergency hardware failure.

Table 8.5-1. Example of Emergency Change Procedure

Operator/User	System
Operator prepares trouble ticket to report ATL controller failure.	Trouble ticket recorded.
System Administrator and Maintenance Engineer confirm ATL controller failure, call ATL maintenance vendor, report call and time in Trouble ticket.	Diagnosis and vendor call recorded in trouble ticket.
Maintenance vendor isolates failure to the controller card. The later version card is the only card available.	
Crew Chief notified of situation and decision needed to bring ATL up to full operating capability. Approves use of the newer version card, records decision in the trouble ticket, forwards trouble ticket to Sustaining Engineer.	
Maintenance vendor installs card, tests using hardware diagnostics. Crew Chief authorizes controller to be brought back on-line.	
Maintenance Engineer records card installation by model/version into the trouble ticket.	Trouble ticket action recorded.
Sustaining Engineer reads trouble ticket and prepares for discussion at 8:30 am meeting. Updates the TT.	Install action recorded in TT. TT routed to the CMA.
CMA updates site baseline, forwards TT to the CCB. When CCB approves the action, CMA forwards to SMC.	Site ATL baseline updated in Baseline Manager.
M&O reviews emergency NCR, checks for applicability to other sites, opens new NCR if other sites require change.	

This page intentionally left blank.

9. Configuration Management Procedures

The procedures that have been prepared are applicable to all of ECS including all hardware, software, and firmware components of systems or subsystems developed or acquired by the ECS contract and/or delegated to configuration management control by the operational site-level organizations. The procedures are applicable to all items maintained by the ECS Sustaining Engineering Organization in support of ECS mission-specific projects and multiple mission-specific institutional facilities. The procedures are not applicable to those entities controlled by higher level ESDIS Project Office CM Plans. The procedures also apply to ESDIS authorized replacements of or augmentations to fungible assets at extant facilities. CM procedures already in place may be used by the contractor subject to direction from the Change Control Board (CCB) chair person.

Some major features of the approach being offered here include:

- Customers participate in establishing the procedures;
- M&O does not constitute a CCB, but rather performs a support role for ESDIS and its designated on-site CCBs;
- Prioritization, automated tools and procedures are used for handling change requests;
- Diverse/Strategic representation at hierarchical CCBs facilitate a path for speedy escalation/resolution of problems/issues;
- Local organizations have the needed autonomy to accomplish their mission with the minimum necessary outside intervention to promote timely resolution of local problems and enable timely production of data products;
- Proper use and deployment of CM database assets to support all CCBs allows management monitoring, control, and analysis of activities;
- Coordination with the Failure Review Board allows coordinated response to problems and filtering of prioritized issues; and
- Common CM tools will be used in all elements of the ECS Project during operations.

The procedures are organized into nine major sections that address the flow-down of procedures from the ECS system-level to the site-level with references to site-tailoring of procedures where applicable. The topics include (Section 9.1) configuration identification, (Section 9.2) change control processes, (Section 9.3) configuration status accounting, (Section 9.4) configuration audits, (Section 9.5) data management, and the use of standardized CM tools known as (Sections 9.6 and 9.7) Software CM Manager (ClearCase), (Section 9.8) Change Request Manager (Distributed Defect Tracking System), and (Section 9.9) Baseline Manager (XRP-II).

9.1 Configuration Identification Procedure

9.1.1 Purpose

The purpose of configuration identification during maintenance and operations is to incrementally establish and maintain the definitive basis of control and status accounting for the ECS control items. To accomplish configuration identification for both hardware and software, the configuration management (CM) administrator (CMA) shall ensure the maintenance of each ECS configuration controlled item in an operational baseline by executing the following tasks:

a. Assign identifiers to configuration items (CIs) and their component parts and associated configuration documentation, including revision and version number where appropriate. Assign serial and lot numbers, as necessary, to establish the CI effectivity of each configuration of each item of hardware and software.

b. Follow ECS developer guidelines as referenced below in section 9.1.3.

c. Follow vendor nomenclature for COTS items.

d. Apply operation and maintenance (O&M) version name extensions to ECS modified item nomenclature following the rules in Section 9.1.4.

e. Follow author-designated version control and nomenclature for documents and follow guidelines from the ECS Librarian (cf. Section 20, *Library Administration*)

f. Support the ECS Librarian's efforts to maintain linkage of the ECS documentation to ECS configuration items in the Baseline Manager tool. Ensure that the marking and labeling of items and documentation with their applicable identifiers enables correlation between the item, configuration documentation, and other associated data.

g. Maintain a release system for configuration changes (cf. Section 9.2, *Configuration Change Control Procedures).*

h. Maintain views of operational baselines using the Baseline Manager tool (cf. Section 9.9).

i. Ensure that applicable identifiers are embedded in the source and object code.

9.1.2 Applicable to

All ECS CM Administrators and support personnel.

9.1.3 References

ESDIS CM Plan, June 1996

MO&DSD CM Plan, Sept. 1995

ECS M&O CM Plan, Nov. 1995

Software Build Process, CM-1-023 (25 March 1996)

Software Naming Conventions, SD-1-015 (14 July 1995)

Directory and File Name Guidelines and Standards for Release A, 15 April 1996

Document Numbering, DM-1-002 (25 November 1994)

DoD Mil-Std-973, April 1992

9.1.4 Procedures

9.1.4.1 Extended Configuration Identification

The extended configuration identification for ECS is of the standard form:

Control Item.Release.Organization.#_Dev.#_M&O.#_center

where:

- *Control Item* is the ECS Project designation of the CI at RRR turnover. The CI naming convention has been explained in CM-1-023 *SW Build Process* and further elaborated in SD-1-015 *SW Naming Conventions*.
- *Release* is the major release, A, B, C, or D.
- *Organization* is the organization that established the configuration. Legal values are DEV for development, M&O for Maintenance and Operations system-wide, or center (e.g., SMC, EOC, EDC, GSFC, etc.) for center unique.
- *#_Dev* is a numeric identifier applied by the development organization to the major release and/or a minor release.
- *#_M&O* is a numeric identifier applied by the M&O/SEO organization. This field is used by the SEO organization to establish the system M&O baseline.
- *#_Center* is a numeric identifier applied by each center. This field is used by the operational centers to establish the site specific baseline.

For example, at the TRMM Development Release RRR, the ECS CCB establishes the initial operational baseline. Assume this baseline is identified as CI.A.DEV.3. CI.A.DEV.3 is delivered to the ESDIS CCB. After ESDIS CCB acceptance, the M&O organization will configure and build its first, system-wide baseline, CI.A.M&O.3.0. If it is assumed that some M&O tailoring is applied, the baseline released to the operational centers is CI.A.M&O.3.1. Each center then configures a center specific baseline. The RRR baseline for EDC, GSFC, LaRC, and NSIDC as well as the SMC and EOC is built from CI.A.M&O.3.1 and are identified as CI.A.EDC.3.1.1, CI.A.GSFC.3.1.1, CI.A.LaRC.3.1.1, CI.A.NSIDC.3.1.1, CI.A.SMC.3.1.1 and CI.A.EOC.3.1.1.

9.1.4.2 Other Procedures as Applicable

The CM Administrator will author other configuration identification procedures applicable to the local site environment to carry out the objectives listed in the Section 9.1.1 Purpose.

9.2 Configuration Change Control Procedures

9.2.1 Purpose

The ESDIS CCB chartered Change Control Boards (CCBs) shall apply configuration control measures to all the ECS configuration items and the associated documentation prior to the time it is baselined for operations. The CCBs shall apply configuration control measures to

a. Ensure effective control of all CIs and their approved documentation;

b. Provide effective means, as applicable, for (1) proposing engineering changes to CIs, (2) requesting deviations and waivers pertaining to such items, (3) preparing notices of revision, and (4) preparing Specification Change Notices; and

c. Ensure the implementation of approved changes.

9.2.2 Applicable to

All ESDIS chartered ECS CCBs.

9.2.3 References

ESDIS CM Plan, June 1996

MO&DSD CM Plan, Sept 1995

ECS M&O CM Plan, Nov 1995

ECS Development Facility Change Control Process CM-1-007, March 1995

9.2.4 Procedures

9.2.4.1 Configuration Change Request Preparation

The Configuration Change Request (CCR) form in Figure 9.2.4-1 has been developed as a medium for the drafting of CCRs throughout the ECS Maintenance and Operations environment for changes processed by the ESDIS Change Control Board (CCB) and its ECS site-level chartered CCBs at the SMC, EOC, and DAACs (GSFC, LaRC, ASF, EDC, JPL, NSIDC, and ORNL). There are numbered items on the form that correspond exactly to the data entry required to be performed by the respective Configuration Management Administrator who maintains CCR records for the CCB on the distributed implementation of the Change Request Manager tool. Each CCB will have unique CCR identification sequence numbers. Each CCB can forward CCRs and reports from the Change Request Manager to SMC where SEO processes system-level CCRs for ESDIS CCB. The ESDIS CM Plan will determine the charter of the respective CCBs and thus the scope of CCR issues to be addressed by the site CCBs.

Earth Observing System Data and Information System (EOSDIS)
Core System (ECS)
Configuration Change Request (CCR)

1. **Configuration Control Board (CCB)**
ESDIS:___ ECS:___ SMC:___ DAAC: GSFC___, LaRC___, ASF___, EDC___, JPL___, NSIDC___, ORNL___ EOC:___

2. CCR No.

3. Submitted Date:

4. Revision

5. Priority
Emergency ☐
Urgent ☐
Routine ☐

6. Change Class

7. Status

8. CCR Title:

9. Originator: Org: e-mail: phone:

10. Approval:_____ _____
signature date

11. Reason for Change

(indicate attachment ___)

12. Description of Change

(indicate attachment ___)

13. Impact Analysis:
Cost: ☐ None ☐ Small (Not exceeding $100,000) ☐ Medium ($100,000 to $500,000) ☐ Large (Over $500,000)
Evaluation Engineer: Org: e-mail: phone:
Impact Evaluators: ESDIS___; ECS Dev___; SEO___; SMC___; DAACs: GSFC___, LaRC___, ASF___, EDC___, JPL___, NSIDC___, ORNL___; EOC___;
Others_____ (indicate attachment ___)

14. Comments: (Indicate Sites/ Organizations Affected)

(indicate attachment ___)

15. Board Action: ☐ Approved ☐ Withdrawn ☐ Disapproved ☐ Deferred Until_____
date
Further Action Required: ☐ ECP ☐ Waiver ☐ Deviation ☐ Tech Direction ☐ Contract Mod. ☐ DCN Other:_____

16. CCB Approval
Chair:_____ _____
signature date

17. CCR Implemented
CM Admin. signature:_____ date:_____

Figure 9.2.4-1. Configuration Change Request (CCR) Form

The following enumerated text corresponds to the numbered items on the CCR form:

(1) **Change Control Board (CCB) --** The designated CCB is checked-off for changes processed by the ESDIS Change Control Board (CCB) and its ECS site-level chartered CCBs at the SMC, EOC, and DAACs (GSFC, LaRC, ASF, EDC, JPL, NSIDC, and ORNL).

(2) **CCR Number --** The unique serialized CCR number is applied at each site.

(3) **Submitted Date --** The date that the CCR was prepared is documented.

(4) **Revision --** The current revision is designated for tracking changed versions of original CCR.

(5) **Priority --** The priority level of the CCR is assigned. Emergency CCRs may have already been implemented on a temporary basis by the Trouble Ticket Review Board (TTRB) with concurrence from the CCB Chair who later receives the CCR to document /implement the permanent change. Urgent items will be reviewed by the next CCB meeting. Routine items will be reviewed as soon as the schedule permits.

(6) **Change Class --** Change Classes are either I or II. Class I will be handled by ESDIS-only because of cost, schedule, and/or mission impacts that may require requirements changes. Class II items do not affect mission requirements, but may have cost and/or schedule implications which affect maintenance, operations, procedures, documentation, site-tailored items, COTS implementation, site installations of core system changes, science SW changes, etc.

(7) **Status --** The following table is a summary of the CCR states maintained by the Distributed Defect Tracking System (DDTS) application SW database tool that implements the Change Request Manager. Note that the hard copy form will not be updated chronically, but will be kept in the master suspense file of the CM Administrator until closed-out with a stamp (item #7 & 15) and appropriate signatures (viz., items 16, & 17).

State Table Composition in DDTS format

State Code	Available States	State Assigned
S	Submit	Submitted
N	New	New
A	Assign-Eval	Assign-Eval
O	Assign-Implement	Assigned-Implement
R	Implement	Implemented
T	Assign-Verify	Assigned-Verify
V	Verify	Verified
C	Close	Closed
D	Duplicate	Duplicate
F	Forward	Forwarded
P	Defer	Deferred

Explanations of DDTS' State Table Content:

Uppercase character in the 1st column is the character stored in the change request record to indicate what state a change request is in. DDTS uses this character to go into the table and extract the descriptive name for display in reports.

Names in the second column is the state in the present tense. They are shown on the DDTS list of states that are available for selection during input. It's also used by some of DDTS' query and report code. It facilitates querying based on descriptive names opposed to a single letter.

Names in the third column is the state in the past tense. They are shown on the DDTS change request record. It's also used by some of DDTS query and report code. It facilitates querying based on descriptive names opposed to a single letter.

Definitions (a state is the stage that a proposed change has reached in its life cycle.):

New - the initial state for all newly entered change requests.

Assign-Eval- state entered when the change request is being assigned to an engineer for evaluation/analysis.

Assign-Implement- state entered when the change request is being assigned to an engineer for development.

Implement-state entered when the proposed change has been developed.

Assign-Verify-state entered when the developed change is being assigned to an engineer for verification testing.

Verify- state entered when a developed change has been tested and verified that it functions properly.

Close- state entered when all activity specified in the change request has been completed or that the approval authority has decided to close it prior to completion of all activity.

Duplicate-state entered when a change request is determine to be duplicate of an existing change request. Duplicate change request identifies change request being duplicated.

Forward-state entered when a change request needs to be forwarded to another DDTS defined project (In DDTS terminology, a project is a grouping of change requests. For example, a change request from a site project can be forwarded to an ECS project.).

Defer- state entered when activity on a change proposal has to be postponed.

(8) **CCR Title --** The CCR title is supplied by the originator.

(9) **Originator --** The originator name, organization, e-mail address, and phone number is given.

(10) **Approval --** The CCR is approved by the designated management authority which is assigned by the CCB. This sponsorship requirement acts as a primary filter to eliminate from consideration those CCRs that cannot be implemented or which have no ECS site management support.

(11) **Reason for Change --** The reason for the change is narrated on the form and/or the designated attachment.

(12) **Description of Change --** The proposed implementation of the change is narrated along with any known impacts, resources, and expenses to be incurred.

(13) **Impact Analysis --** Impact analysis is documented in the form of figure 9.2.4-3. The impact analysis is collected by the CCB Chair appointed Evaluation Engineer in coordination with the CM Administrator who maintains the CM records and assembles the review package for the CCB. The Evaluation Engineer documents the list of Impact Evaluators and derives and/ or verifies cost, technical, and schedule impact of the proposed change based on all inputs received. The results of the coordinated CCR Impact Analysis inputs are presented in the CCR Impact Summary form shown in figure 9.2.4-3 part of the CCR review package.

CCR Impact Analysis

Responder Request Number: ____
Responder:_____
Responder Point of Contact:
address:_____
_____
phone:_____
e-mail:_____
CCB Schedule Date:_____
CCR Number:_____
CCR Log Date:_____
CCR Originator:_____
CCR Originator Point of Contact:
address:_____
_____
phone:_____
e-mail:_____

Evaluation Engineer:_____
Evaluation Engineer Point of Contact:
address:_____
_____
phone:_____
e-mail:_____
Requested Return Date:_____

Rough Order of Magnitude (ROM) Impact Analysis

Basis of Estimate:

Technical Assumptions and Comments:

Cost Impact:
None [ ]
Small [ ] < $ 100, 000
Medium [ ] $100,000 < x < $500,000
Large [ ] > $500,000

Schedule Impact:

Technical Assessment: (Your impact analysis should consider the implementation approach; interfaces affected; HW or SW changes required; documentation changes required-- change from/to pages; suggested alternatives, if any; and impact to security features. If your system is not impacted, please provide that information to the CM Administrator.)

Comments:

Signed:_____
(Responder)

Date:_____

Figure 9.2.4-2. ECS CCR Impact Analysis

CCR Impact Summary

Evaluation Engineer:_____
Evaluation Engineer Point of Contact:
address:_____
_____
phone:_____
e-mail:_____
CCR Board Date:_____

Resources Summarized:

Technical Summary:

ROM Summary (BOE, Cost, and Schedule):

Recommendation:

Signed:_____
(Evaluator)

Date:_____

Figure 9.2.4-3. ECS M&O CCR Impact Summary

(14) **Comments** -- Comments are added to the CCR to summarize sites and/ or organizations affected by the CCR. Additional comments may address proposed CCB dispositions and recommendations to be indicated by resolutions in item #15.

(15) **Board Action --** CCB actions and follow-up actions that will be facilitated and tracked by the CM Administrator are indicated. Possible CCB dispositions are given as approved, withdrawn, disapproved, and deferred (pending follow-up activities by the indicated schedule date). Further actions are indicated as

Engineering Change Proposal (ECP)-- changed scope of contract requirements

Waiver-- declaration that certain contract requirements no longer apply

Deviation-- change of contract terms or substitution of terms or deliverable requirements

Technical Direction-- order by Contracting Officer's Technical Representative (COTR) to perform certain tasks within the scope of the contract

Contract Modification-- changes to the terms of a contract

Document Change Notice (DCN)-- notification of changes to published documents

Others-- Engineering Change Notice, Change Order, Escalate to higher CCB authority, etc.

(16) **CCB Approval --** CCB approval signature authority by CCB Chair or designate.

(17) **CCR Implemented --** This signature and close-out stamp (item #7) are executed by the CM Administrator witnessing the completion of the CCR implementation process which is tracked in the Change Request Manager automated tool DDTS and updated in Baseline Manager (XRP-II) for affected version control status changes.

9.2.4.2 Change Control Board Process (System and Site-level CCBs)

The ECS M&O organization will provide administrative and technical support services for the CCB at each site. Each site's CCB is controlled by the host site organization and provides the authority and direction for the ECS contractor to modify the operational baseline. The ESDIS CCB has chartered an ECS Review Board to coordinate ECS system-level changes and problem management via the Sustaining Engineering Organization (SEO) contractor and on-site Review Boards that also act as site CCBs. This is illustrated using the CM Administrator's workflow for the SEO support of the ECS Review Board in Figure 9.2.4-4 and the On-Site CM Administrator's workflow for SEO support of the on-site CCB in Figure 9.2.4-5. The problem management process was discussed in detail in Section 8 of this document. Both diagrams illustrate the flow of CCRs through the respective CCBs with inputs from the review boards and evaluators that determine the disposition of proposed changes. Details of this process are given below:

System-level Change Control Procedures

(The enumeration corresponds to the diagram of figure 9.2.4-4)

(1) Configuration Change Requests are received by the SEO CM Administrator from all sources with regard to the operational ECS Core System as described in Section 9.2.4.1. These changes

designated as from other sources could involve system enhancements, procedures, interfaces (both external and internal), documentation changes, etc. that are not the subject of contemporaneous problem reports which would be first deliberated by the Trouble Ticket Review Board (TTRB) and/ or Failure Review Board as explained below.

(2) Proposed common baseline changes will be proposed based on Trouble Ticket (TT) resolutions obtained from the respective review boards (see Section 8 for details). The respective TT would be closed via a corresponding CCR to either ratify, i.e., to make permanent the prior temporary/emergency action taken by the TTRB or to consider normal priority (scheduled) changes for incorporation into future change releases.

(3) The SEO CM Administrator is responsible for logging the CCR into the Change Request Manager (DDTS tool) as described in Section 9.8.

(4) The CCB chair assigns an evaluator and the SEO CM Administrator coordinates impact assessment.

(5) Class I change requests (proposed changes that affect controlled milestones, schedules, budget, cost and requirements) are forwarded to the ESDIS CCB for consideration with recommendations from the ECS Review Board.

(6) Class II change requests (proposed changes affecting documentation, hardware [alternative use of], software [correction of errors], and COTS substitution without a Class I impact) are considered by ECS Review Board deliberations.

(7) Notice of proposed changes is distributed to affected parties and review board members to obtain and coordinate impact assessment and optimize the approach to implement proposed changes.

(8) The results of ECS Review Board deliberations are factored into review board resolutions which determine whether, when, or where the system changes will be implemented.

(9) Approved changes are processed by the SEO CM Administrator to the support activities, i.e., site CCBs, support personnel (SEO), vendors, etc. who are provided with change orders, schedule, and implementation instructions.

(10) Disapproved changes are processed by the SEO CM Administrator via official notifications, memo to the file, and update of the Change Request Manager (CRM).

(11) The SEO CM Administrator tracks implementation and closure of CCRs via directions to implementing organizations and their acknowledgements using the CRM tracking and statusing features (see section 9.8).

(12) New versions and/or maintenance updates are annotated in Baseline Manager at the SMC and at the affected sites by following the procedures for configuration identification, activation dates, deactivations dates, and issuing version description documents.

(13) Simultaneously, the SW Change Manager (ClearCase) is updated with directory trees, installation files, and software as required by SW maintenance.

(14) Status of this activity to implement changes and assigned responsibilities is tracked through closure in the CRM at SMC and at the sites.

(15) The databases are synchronized by manual checking between applications Baseline Manager vs. CRM vs. SW Change Manager) and automated verification by the SW CM Manager for purposes of SW distribution and maintenance.

(16) The TT Review Board is empowered to make emergency fixes without common baseline changes and update these changes directly to Baseline Manager with documentation to follow via the CCR submitted to the appropriate CCB. Proposed common baseline changes must be submitted by CCR to the ECS Review Board.

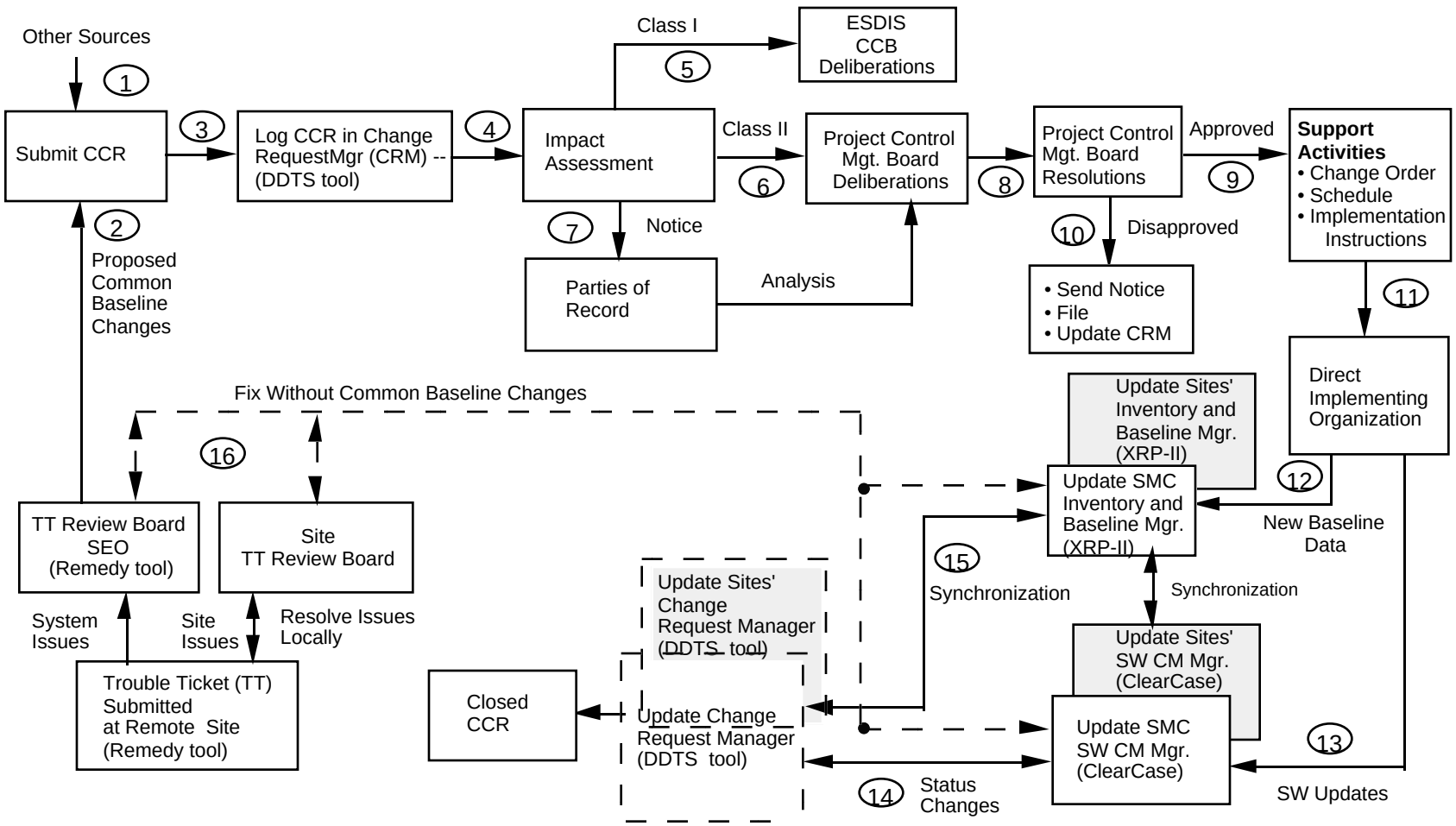


Figure 9.2.4-4. Work-Flow Diagram for SEO CM Administrator

Site-level Change Control Procedures

(The enumeration corresponds to the diagram of Figure 9.2.4-5)

(1) Configuration Change Requests are received by the Site CM Administrator from all sources with regard to the **site unique extensions** to the operational ECS Core System as described in section 9.2.4.1. These changes designated as from other sources could involve system enhancements, procedures, interfaces (both external and internal), documentation changes, etc. that are not the subject of contemporaneous problem reports which would be first deliberated by the Site/ SEO Trouble Ticket Review Board (TTRB) and/or Failure Review Board as explained below.

(2) Proposed site baseline changes will be proposed based on Trouble Ticket (TT) resolutions obtained from the respective review boards (see section 8 for details). The respective TT would be closed via a corresponding CCR to either ratify, i.e., to make permanent the prior temporary/ emergency action taken by the TTRB or to consider normal priority (scheduled) changes for incorporation into future change releases.

(3) The Site CM Administrator is responsible for logging the CCR into the Change Request Manager (DDTS tool) as described in Section 9.8.

(4) The CCB chair assigns an evaluator and the Site CM Administrator coordinates impact assessment.

(5) Class I/System Issues change requests (proposed changes that affect controlled milestones, schedules, budget, cost and requirements) are forwarded to the ECS Review Board for consideration with recommendations from the Site CCB. Class I issues are further forwarded with recommendations by the ECS Review Board to the ESDIS CCB for consideration.

(6) Class II change requests (proposed changes affecting documentation, hardware [alternative use of], software [correction of errors], and COTS substitution without a Class I impact) are considered by Site CCB deliberations.

(7) Notice of proposed changes is distributed to affected parties and review board members to obtain and coordinate impact assessment and optimize the approach to implement proposed changes.

(8) The results of Site CCB deliberations are factored into CCB resolutions which determine whether, when, or where the system changes will be implemented.

(9) Approved changes are processed by the Site CM Administrator to the support activities, i.e., other CCBs, support personnel (SEO), vendors, etc. who are provided with change orders, schedule, and implementation instructions.

(10) Disapproved changes are processed by the Site CM Administrator via official notifications, memo to the file, and update of the Change Request Manager (CRM).

(11) The Site CM Administrator tracks implementation and closure of CCRs via directions to implementing organizations and their acknowledgements using the CRM tracking and statusing features (see Section 9.8).

(12) New versions and/ or maintenance updates are annotated in Baseline Manager at the affected sites and the SMC by following the procedures for configuration identification, activation dates, deactivations dates, and issuing version description documents.

(13) Simultaneously, the SW Change Manager (ClearCase) is updated with directory trees, installation files, and software as required by SW maintenance.

(14) Status of this activity to implement changes and assigned responsibilities is tracked through closure in the CRM at the sites.

(15) The databases are synchronized by manual checking between applications Baseline Manager vs. CRM vs. SW Change Manager) and automated verification by the SW CM Manager for purposes of SW distribution and maintenance.

(16) The on-site TT Review Board is empowered to make emergency fixes without common baseline changes and update these changes directly to Baseline Manager with documentation to follow via the CCR submitted to the appropriate CCB. Proposed common baseline changes must be submitted by CCR to the ECS Review Board.

Each site's CCB accepts initial release or updates from the ESDIS CCB. Similarly, the Distributed Active Archive Center (DAAC) CCBs will accept product generation software from an ESDIS authority. Local tailoring and installation decisions are determined by the site CCB.

In the case of Evaluation Package (EP) and Prototype deliveries, the ECS CCB as directed by ESDIS will provide a configured, documented, executable with supporting files. Again, installation decisions are determined by the site CCB.

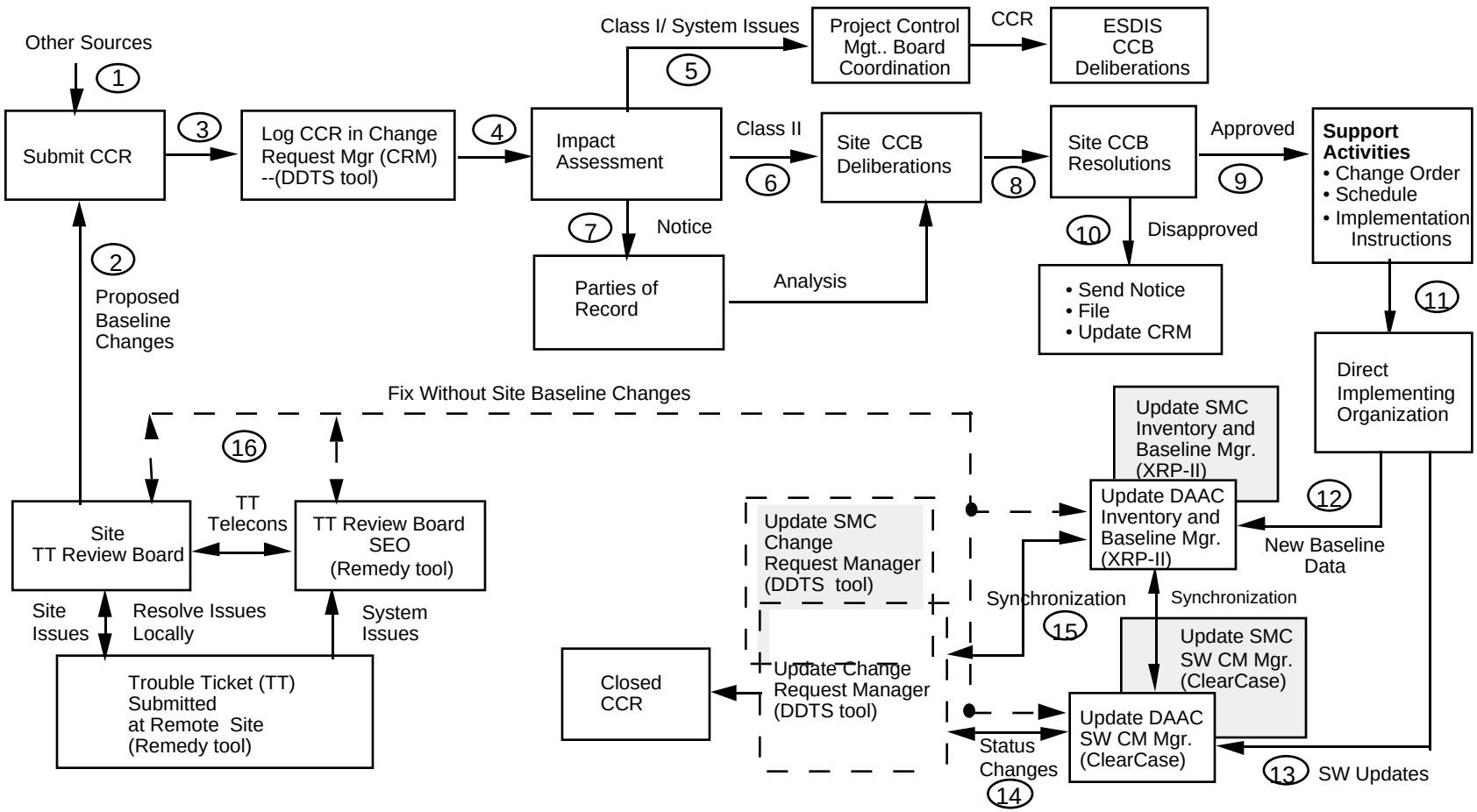


Figure 9.2.4-5. Work-Flow Diagram for Site-level CM Administrator

Each Science Computing Facility (SCF) is assumed to have a configuration control function. For commonality with other sites, it is assumed that this function will be performed by a CCB. A major difference, though, is the ECS contractor does not have an active role in the support of this CCB.

The SCF CCB will provide two types of configuration control:

(1) Configuration control of software and databases that are to be executed in another site's environment. In this mode, it operates very much as the ECS CCB does to establish a product baseline.

(2) Configuration control of SCF resources that are made available to the EOSDIS community. In this mode, its functions are the same as a DAAC CCB.

The ECS M&O CM function at each DAAC will accept science SW and data items from the SCF CCB. These items will be incorporated into the DAAC's operational baseline as directed by the DAAC CCB.

The EOC CCB will control the operational configuration of the required EOC operational baseline. ECS M&O CM will provide services as directed by the CCB.

The ECS Review Board will be charged with the responsibility for centralized coordination and control of ECS CM activities to ensure:

- ECS integrity and quality of service
- Successful coordination with both internal and external networks, systems, and on-site facilities
- Timely ESDIS CCB visibility into and oversight of ECS operations
- Convenient user administrative services

9.2.4.3 Configuration Control - Deviation and Waivers

1. Prior to completion of software development or to purchase of equipment or software, a Deviation from the specification may be granted by the ESDIS CCB.

2. Subsequent to the completion of development or delivery of equipment or software, a Waiver of specific requirements may be granted in order to accept defined nonconforming items. The waiver is traceable to a nonconformance report. A waiver is limited: additional deliveries of like items must conform to approved requirements .

3. Departures from expected configurations that are not at variance with customer-approved requirements are handled as Nonconformance Reports.

4. A request for a Deviation or Waiver consists of a Deviation/Waiver Form shown in figure 9.2.4-6 attached to a CCR form .

5. Instructions for completing the form are listed below. Additional pages should be attached as necessary.

Dev or Wai: Check the applicable box in accordance with the definitions given on page 1 of this Instruction.

Waiver Number: Assigned by ECS CM Administrator

Title: Enter a brief descriptive title. The title should be a statement, e.g., "Accept x in lieu of y".

Reason: Describe the reason for the deviation or waiver. This may also include the history of the problem and consequences of not implementing the deviation/waiver.

Existing Requirement: Specify and describe the baseline from which the deviation or waiver departs.

Departure: Give instructions for the deviation or waiver with reference to the requirement.

Implementation Scope: Include as appropriate, configuration item number, model no., supplier, subcontractor, series, serial numbers, order no., location, release number, quantity, time period or other criteria delimiting the deviation or waiver.

Documents Affected: List current release number(s) of affected document(s).

6. The Deviation or Waiver CCR is prepared in accordance with the instructions in section 9.2.4.1 with the following exceptions:

   Change Class: All deviations and waivers are change class I because they depart from approved requirements and must be approved by the ESDIS CCB.

   Description (Title): The title of the CCR will be the deviation or waiver title.

   Proposed Solution: Enter "See Attached Deviation/Waiver form ."

7. Waiver and Deviation CCRs are submitted to the ECS CCB. Upon ECS CCB approval, the CM Administrator forwards the CCR to ESDIS CCB for authorization to implement the deviation or waiver.

8. When the deviation or waiver is authorized, the ECS CM Administrator immediately distributes the authorization information to the appropriate implementors and issues a document change order (DCO) to the Document Management Organization.

9. The SEO Librarian copies the implementation instructions into the List of Deviations and Waivers at the front of the document and inserts the Deviation/Waiver number and effectivity at the point of applicability within the document.

10. Approved deviations/waivers are published via Document Change Notices (DCNs). Adding the deviation/waiver information to the document makes its status general knowledge. However, the deviation or waiver is in effect as soon as it is authorized by the customer .

11. A change in scope, effectivity or closeout, or any other change in a deviation or waiver requires a new CCR. The closeout or change is applied to the document via a Document Change Order in the same procedure as given in paragraphs 7 through 10.

ECS Deviation / Waiver

Deviation/Waiver No.	Deviation ☐ Waiver ☐	Date:
NCR No.		
Title:		
Reason for Deviation / Waiver:		
Existing Requirement: (attach pages as needed)		
Departure: (attach pages as needed)		
Implementation Scope: (Identify CI, model, supplier, subcontractor, series, order no., location, release, time period, etc. as applicable)		

Document No.	Page/Paragraph Reference	Document Title

CM05MR95 ***ECS***

Figure 9.2.4-6. Deviation/Waiver Form

9.3 Configuration Status Accounting Procedures

9.3.1 Purpose

Operational phase configuration status accounting (CSA) consists of recording and reporting information about the configuration status of the ECS Project's documentation, hardware and software throughout the Project life cycle. Periodic and ad hoc reports keep ESDIS informed of configuration status as the operational mission evolves. Reports to support reviews and audits will be extracted as needed starting from the RRR.

The Baseline Manager tool described in section 9.9 records and tracks as-built products designated as ECS control items (i.e., custom, COTS, science, toolkits, etc. SW and HW items along with their associated documentation and records) and historical versions of ECS operational configurations. Baseline Manager, which is updated with the acceptance tested version of the ECS baseline at RRR, records and reports M&O document change status and histories, mission milestone baselines, and change status.

CSA entails maintaining version histories of delivered and maintained products as well as histories of operational baselines and changes made to each baseline. Additionally, CSA tracks the status of proposed changes from initial CCR submission to ultimate disposition and/or implementation. CSA also maintains historical records of CCRs.

9.3.2 Applicable to

All ESDIS chartered CCBs.

9.3.3 References

ESDIS CM Plan

ECS M&O CM Plan

9.3.4 Procedures

The following are topical items subject to periodic or ad hoc reporting on behalf of the respective CCB or a system-level summary of information that will be reported by the SEO CM Administrator representing the operational baseline for all the ECS sites.

(a) **New CCRs and Revisions.** This is a standard Change Request Manager report (cf. Section 9.8). This report will be issued monthly and summarized annually.

(b) **CCB Review.** Distribute CCR copies for review (and Impact Analysis forms if applicable). Print the agenda and distribute prior to the meeting.

(c) **Open Action Items.** Open action items should be statused regularly between meetings.

(d) **CCB Meeting.** Record the CCB's disposition of each CCR.

(e) **Record Action Items.** Record actions, assignments, and due dates.

(f) **SEO Librarian Maintained Document Changes.** When all authorized document changes have been accomplished prepare DCN, post the final version on the ECS Document Data Server and distribute hardcopy as required.

(g) **Minutes Distribution.** Distribute minutes to the standard distribution and inform actionees of assigned action items.

(h) **CCR Implementation Status.**

- After CCB disposition, regularly status open CCRs until closure.
- Class I events include: CCR to ECS Review Board for review/appoval; Technical Review Board; and ESDIS Disposition
- Further events are as follows for M&O implementation status: Consent Obtained; Item Received; Installed; Document Completed; etc.
- CCR CLOSED: A Class I CCR is not closed until the ESDIS contract officer's authorization is received or the reference CCR has been withdrawn.
- Class II document change CCRs may be closed with the CM Administrator's issuance of the DCN.
- Other non-document change CCRs may be closed when the originator verifies to the CM Administrator that all specified changes have been implemented.

9.4 Configuration Audits

9.4.1 Purpose

SEO will support Functional Configuration Audit /Physical Configuration Audit (FCA/PCA) by IATO at RRR. SEO will also support audits by ESDIS and our own Quality Office functions. Internal CM self-audits will be conducted by the SEO. Self-audits evaluate the Project's compliance with the EOS Configuration Management Plan and the ESDIS CMP. The CM self-audits will verify:

- That CM policies, procedures, and practices are being followed.
- That approved changes to documentation, and to software and hardware products are properly implemented.
- That the as-built documentation of each CI agrees with the as-deployed configuration or that adequate records of differences are available at all times.

A post-audit report is written outlining the specific items audited, audit findings, and corrective actions to be taken. All action items are tracked to closure.

In addition, SEO supports formal audits scheduled and conducted by ESDIS. These audits are conducted to validate that each ECS CI is in conformance with its functional and performance requirements defined in the technical documentation. The audits validate that:

- The as-built configuration compares directly with the documented configuration identification represented by the detailed CI specifications.
- Test results verify that each ECS product meets its specified performance requirements to the extent determinable by testing.
- The as-built configuration being shipped compares with the final tested configuration. Any differences between the audited configuration and the final tested configuration are documented.
- When not verified by test, the compatibility of ECS products with interfacing products or equipment is established by comparison of documentation with the interface specifications which apply.
- COTS products are included in audits as integral parts of the ECS baseline.

9.4.2 Applicable to

All ESDIS chartered CCBs.

9.4.3 References

ESDIS CM Plan

ECS M&O CM Plan

9.4.4 Procedures

The audits willl be standardized for a limited set of issues that drive the process for which the audit is taken, viz., FCA/PCA, Security Issues, General Accounting, Test Readiness Review, or Operational Certifications. The documented basis for the audit process will be maintained in the Baseline Manager CM tool (cf. Section 9.9). Alternatively, the Version Description Document (VDD) will be used to document auditable changes to configured articles that are issued at the ECS configuration item (CI) level. The use of the VDD is discussed in the ECS M&O CM Plan. The VDD will contain the prioritized current status summary of any Trouble Tickets/Discrepancy Reports against the CI that is being issued per the change request.

Some general guidelines and/or items that must be tailored for the specific size and scope of configuration audit to be conducted include:

(a) Audit Plan;

(b) Conference Agenda;

(c) Location to collect and analyze data; conduct meetings;

(d) Applicable specifications, drawings, manuals, schedules, design and test data;

(e) Test Results Analysis;

(f) Meeting minutes including resulting audit action items;

(g) Tools and inspection equipment necessary for evaluation and verification;

(h) Unencumbered access to the areas and facilities of incoming inspection, fabrication, production, and testing;

(i) Personnel from each engineering, production, and quality department to be available for discussion of their respective areas;

(j) Copies of inspection reports, process sheets, data sheets, and other documentation deemed necessary by the Government FCA/ PCA teams; and

(k) Isolation of the item(s) and detailed parts to be reviewed.

9.5 Data Management

9.5.1 Purpose

The term "data management" pertains to two activities. The first pertains to the activities of the Data Management Organization (DMO) within the ECS project. The second, broader in scope, includes all organizing and use of documents and data across the ECS project. This procedure describes the DMO activities fully. In addition, certain activities that are material to the data management of the project are described in this procedure although they are performed by organizations other than the DMO.

This procedure describes:

- the policies and procedures that apply to data handling and document distribution.
- requirements for data management for the ECS project.
- the role of the data base administrator .
- the use of data control and data handling systems (Document Data Handling Subsystem (Section 20).
- the activities pertaining to establishing and maintaining system libraries and records.
- management of data required during operations readiness review, missions operations, and certification and to assist the on-going development of the system for design, implementation, and test.
- how data will be collected, maintained, and made available to the development team and for distribution to the NASA.
- the data management functions of controlling document masters, preparing change pages, and keeping auditable change records.
- the plan for controlling the data base structure, controlling the interfaces to the data base, establishing the data base security, and evaluating data base performance.

This procedure does not cover the handling or use of notes, test data, software, financial information, or draft documents that would generally be characterized as working papers or other non-controlled and non-deliverable information. This information is handled by the contractor Integrated Management Information System (IMIS).

9.5.2 Applicable to

All ECS Sites CM Administrators and the SEO Librarian

9.5.3 References

ECS Data Mangement Plan

ECS Documentation Management and Control	DM-1-001
Documentation Numbering	DM-1-002
CDRL and Required Document Generation, Review, Release, and Maintenance	DM-1-004
Electronic CDRL Preparation and Delivery	DM-1-005
CDRL Document Format	DM-1-006
Document Delivery and Dissemination	DM-1-007
ESDIS/ECS Review of Contract Required Data	DM-1-008
ECS Documentation Storage	DM-1-009
Document Archiving	DM-1-010
Technical and White Paper Generation, Review, Release, and Maintenance	DM-1-013
Electronic Storage and Distribution of ECS Documentation	DM-1-014

9.5.4 Procedures

The following text describes standard data management procedures and methods to be used by the ECS project. Occasionally special circumstances may arise which call for exceptions and flexibility to the customary procedures.

9.5.4.1 Information Preparation, Submittal, & Cataloguing

9.5.4.1.1 Creation/Preparation

The originator / author (usually engineering) will create all source material (text, graphics files, etc.) per CDRL/DID preparation instructions and be accountable for the accuracy of its content. Publishing will assist the author by providing Word Processing and Graphics support such as templates and fonts. (Publishing will do the final formatting later.) The DMO will provide the appropriate CDRL numbering and DID instructions.

9.5.4.1.2 Submission

The originator / author will submit all source material (text, graphics files, etc.) to the DMO electronically including necessary metadata descriptors. The latter include reference to source documents and dates.

The Data Management Office verifies delivery schedule with the appropriate task manager prior to a scheduled release or CDRL delivery date. The DMO notifies the responsible organization at the Program Office Weekly Status Report of upcoming CDRL items and their delivery dates, any special preparation instructions, formats, title pages, signature approvals, and any other pertinent information. Copies of the ECS Master Schedule are provided at the Weekly Status Review.

9.5.4.1.3 Identification and Numbering

Data Identification. Upon receipt of information, the SEO Librarian assigns a DMO identification number to each pertinent type of program data created for the ECS program. If appropriate both SEO and NASA numbering are assigned at this time.

9.5.4.1.4 Logging/Cataloguing

The DMO verifies proper submission of information into the system including valid numbering, manually making changes if necessary, and provide supplementary cross-references as part of the data base catalogue. DMO also updates the status log for those submissions that were pre-scheduled.

9.5.4.2 Information Review, Signoff, Release, and Change/Revision

9.5.4.2.1 Document/Test Data Review, Release, and Change Procedures

(ESDIS/ECS Review) of Contract Required Data (DM-1-008)

9.5.4.2.2 Review/Release

Prior to its release, each CDRL document will be reviewed and signed off by each of the participating functions involved (e.g., engineering, product assurance, systems engineering) as depicted in Figure 9.5.4-1. Only when all the proper signatures have been obtained will the document be released and distributed.

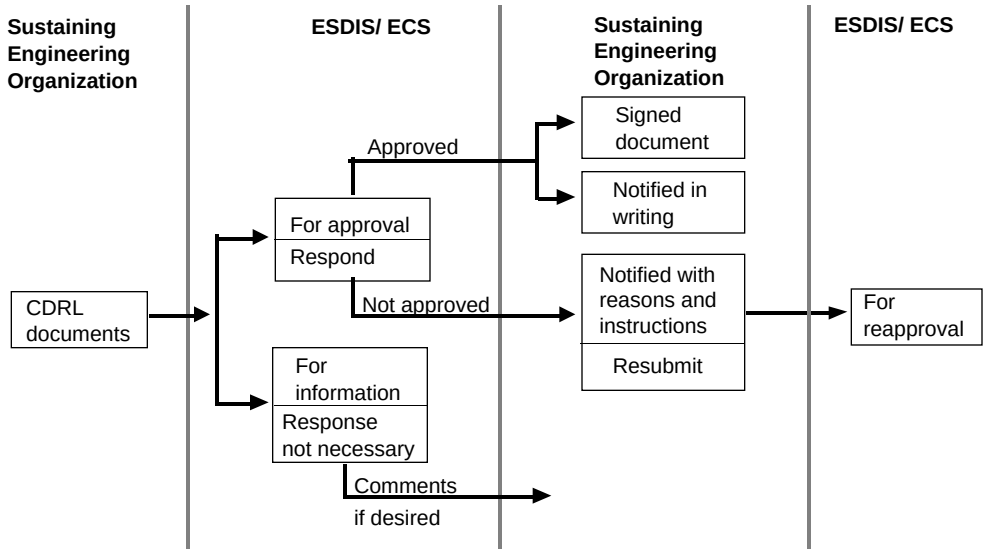


Figure 9.5.4-1. CDRL Item Submission Criteria Flow

The DMO will verify that review and signoff has been completed and annotate the status log for the respective information accordingly.

Data Control Procedures:

1. Data Item Scope, Content, and Format
2. Data Item Submittal (Schedule and Distribution)
3. Data Item Use of Common Information
4. Data Not Duplicated
5. Data Item Quality Control and Assurance
6. Data Item Approval
7. Internal Data
8. Data Items Rework
9. Recorded Information

9.5.4.2.3 Changes, Revision, and Document Maintenance

Changes to controlled documentation come under control of the appropriate Change Control Board (CCB) and details concerning the procedure are to be found in the Configuration Management Plan. In summary, once the author (usually engineering) and the CCB agree on a change, the DMO is mandated to reflect the change in the data and documents. Changes to other data come under control of the appropriate task manager.

Once information is released, no changes will be permitted except upon the instruction of the appropriate authority (CCB or task manager). Changes to specifications are submitted to a formal change review board, which repeats the review/approval process (described above).

The DMO receives the document with approval signatures and reviews for compliance with the SOW, DID, and other contractual requirements. If found to be deficient, the document is returned to the author for corrections.

Revision and resubmission of any CDRL item will be subject to the same submission criteria as applied to the initial release of the document. The DMO will store a file copy and include information about updates in the next status report. The DMO will maintain a detailed data catalog and provide, as requested, specific indices that include document revisions and issue dates.

9.5.4.3 Information Distribution and Submission to ESDIS/ECS

9.5.4.3.1 Data/Document Distribution/Submittal to ESDIS/ECS

ESDIS/ECS will receive preliminary versions of data for comments prior to final release. After data in final form is received and cataloged, the DMO will reproduce the data, if necessary, and make distribution in accordance with CDRL/DID instructions and 500-TIP-2601 delivery interchange standard. The DMO will maintain special distribution lists (sorts) for any requirement that may occur. For documents transferred electronically, the Document Data Server Subsystem will encode and transmit the documents to ESDIS/ECS and will prepare and deliver a backup disk or tape for each document in the appropriate format.

All contractual data submitted to ESDIS/ECS, both engineering and non-engineering, will have a standard transmittal sheet attached. This sheet will contain key information about the data being submitted, such as data number, description, submission criteria, format.

9.5.4.3.2 Categories of CDRL Data Submitted to ESDIS/ ECS

All CDRL data submitted to ESDIS/ECS will be classified as being *for information* or *for approval.*

1) For Information. – Routine documentation which will be evaluated by ESDIS/ECS to determine current program status, progress, and future planning requirements. Examples of *for information* documents include status reports and programs management directives. *For information* documents shall be sent to ESDIS/ECS as soon as approved and issued by the SEO. ESDIS/ECS may elect to provide comments, although a formal ESDIS/ECS response is not required.

2) For Approval. – Documentation that requires written approval from ESDIS/ECS before its acceptance, distribution, and intended use. Examples of *for approval* documents include all documentation that is required to come before a CCB. ESDIS/ECS will approve the document or ask for resubmission at ESDIS/ECS Program Office. Provisions will be made for ESDIS/ECS signature on the cover of documents submitted for approval. If the document is approved, ESDIS/ECS will sign and return the document and notify SEO in writing of the approval. If the

document is not approved, within a mutually agreed time, ESDIS/ECS will notify SEO of those parts of the document which cannot be approved, together with the reasons and instructions concerning resubmission of the document. If the ESDIS/ECS evaluation reveals inadequacies, ESDIS/ECS will inform SEO of the parts of the document that require alterations, recommend actions, and give resubmission instructions according to the character of the inadequacies. SEO will resubmit the document to ESDIS/ECS for approval after receipt of ESDIS/ECS's notification.

9.5.4.3.3 Documentation Distribution

The DMO will prepare approved CDRL(s) transmittal letters. The DMO will reproduce copies of the letter and data package as required by the CDRL, ship it to those on the approved distribution list, and provide internal distribution as appropriate.

9.6 Archiving Procedures for the SW CM Manager (ClearCase)

9.6.1 Purpose

These instructions establish Configuration Management procedures for the backup of ClearCase Version Object Base (VOB) data, Views, and data delivered to the customer.

9.6.2 Applicable to

All ECS CM Administrators, SW Maintenance Engineers, and Sustaining Engineers

9.6.3 References

Configuration Management Plan for the ECS	101-102-MG1-001
CCB Processes	CM-1-004
Configuration Control of Formal Track Software	CM-1-010
Configuration Control of EP Software	CM-1-016
Configuration Control of Prototype Software	CM-1-018
Software Development Library	CM-1-019
EDF Configuration Control	SE-1-002

9.6.4 Procedures

DEFINITION

ECS Development Facility (EDF) - the software development environment including data, hardware, software, networks, facilities, and procedures used to support ECS software development and testing.

Software - for the purpose of this instruction, software includes all ECS-developed application software, COTS software, build and environmental instructions, and databases used in the execution of these products.

Segment-level - for the purpose of this instruction, segment-level includes all software development undertaken in the EDF by ECS segments from the initiation of software coding through completion of segment integration and test (I&T).

System-level - for the purpose of this instruction, system-level includes all ECS integration and test activities beginning with installation of segment software in the system I&T files.

Software Development File - a repository for a collection of material pertinent to the development or support of software.

Thread - a set of components (software, hardware, and data) and operational procedures that implement a scenario, portion of a scenario, or multiple scenarios.

Build - An assemblage of threads to produce a gradual buildup of system capabilities.

VOB - Version Object Base. Secure, permanent, mountable file system. Repository for storage of version-controlled data.

View - A unique workspace management or "sandbox management" that provides developers with transparent, file-level access to any version of any element through the use of dynamically-evaluated, user-specified version selection rules.

GENERAL

1. System Administrators maintain a backup for all ECS systems. This procedure documents an additional backup procedure for only ClearCase-specific data. This process is in parallel with System Administrator procedures. ClearCase, the ECS level software tracking tool, contains within itself a number of repositories called Version Object Bases (VOBs). These VOBs maintain all versions of all software elements developed for the ECS project. Additionally, there are disk areas known as Views, which contain file(s) that may be newly created or modified but not yet given or returned to ClearCase control.

2. It is critical that these repositories, VOBs, and Views get backed up synchronously on a daily basis, and that the backup tapes be maintained in such a way as to guarantee their restoration to the system in the event of a catastrophic failure. A View may contain files that were checked out of a ClearCase VOB, modified, but not yet checked back in when the failure occurred. This CM archive procedure uses synchronization techniques to ensure that no data is lost. The System Administrator's procedure does not include these techniques.

3. Current system and network configurations prohibit the installation of the View storage area on the developers'/ maintainers' personal workstations. As a result, Configuration Management has assumed the responsibility of backing up the View storage areas from the ClearCase server. However, once the developers/maintainers are provided with workstations that permit the installation of a private View storage area, this responsibility will be shifted to the developer/maintainer.

4. All ECS software is delivered electronically and via magnetic media. Copies of the exact magnetic media that were delivered are made and stored similarly to in-house generated backup tapes.

5. Backups of VOBs and Views are accumulative rather than incremental. Therefore, each tape has all data contained on previous archive tapes plus new files created, modified, etc.

6. A tape set consists of 31 archive tapes. The tapes are numbered from 1 to 31, a tape for each day of the month.

7. All software CM personnel will be familiar with the archive procedures.

8. All tape backup and archives are performed on Triton.

INSTRUCTION

1. A Software Configuration Management person will be assigned the responsibility for tape backup and archives. Daily backups are created every day at 03:30 a.m. via a UNIX cron job. Every workday morning the assigned Software CM person will perform tape archive procedures.

2. The daily backup tapes are obtained from and maintained in a fireproof safe which is kept locked. Key CM personnel hold the combination to the safe. The most current tape (the tape corresponding with the day of the month) is removed from the Triton tape drive and is always placed in the slot farthest to the left in the location marked "Current Backup". The tape corresponding to the day of the month for the next day is inserted into the tape drive.

3. On the first business day of each week the backup tape from the last business day of the previous week is given to System Administrator personnel for transportation to an off-site storage facility. Periodically the tapes stored off-site are returned to CM for reuse.

4. All tapes will be labeled with the type of data contained. In the case of delivered software, this will include, at a minimum, the title and version of the delivery, the tool used to generate the tape, and the date made.

   Example of a routine weekly backup tape label:

   Weekly Clear Case Backup Tape

   VOBs and Views

   For week ending: 01/06/94

   Tar formatted.

   Person Performing Backup:

   Example of a Deployed Software Tape:

   194-904-DV3-002

   Label: PGS TK2 Test Drivers Date: April 29, 1994

Test_drivers.tar 2gbyte format

Command to copy file to disk: dd if=## of=file.tar
Command to untar tar file: tar xvf file.tar
where ## = your tape device and file.tar = Name you give tar file

5. During weekends and holiday periods the tape for the next workday will be placed in the tape drive on the last workday prior to the weekend/holiday. This tape will be written to on all days prior to the next workday. In this case should the system crash over the weekend/holiday period, there will be a more up-to-date backup. For example, on Friday, Monday's tape would be placed in the drive. This tape will be written to on Saturday, Sunday, and Monday due to the automated cron job. Should the system crash on Saturday after 03:30 a.m., there would be a backup from Saturday available rather than Friday. On Monday this tape would be removed and replaced with the tape for Tuesday.

6. This process does not archive personal view storage areas resident on personal development workstations (see General 3., above). Therefore, when personal development workstations are assigned, files checked out of ClearCase and under modification are not backed up. The System Administrator can back up these workstations. The user can request through the Help Desk that his development workstation be scheduled for System Administration backup. Alternately, personal backup procedures can be initiated.

7. At the time the tar file for software delivery is created, a second delivery tape will be generated in accordance with delivery procedures. This archive copy will be given to System Administrator personnel for transportation to the off-site storage facility. Incremental delivery tapes will be created, documented, and stored in accordance with this procedure. Delivery archive tapes will be stored for the life of this contract.

9.7 SW Transfer and Installation

9.7.1 Purpose

This procedure involves transferring a Sustaining Engineer Organization (SEO)-developed software maintenance change package from the SMC to a remote site (a DAAC) and later installing the ECS custom software on a selected host computer under a configuration management controlled process. The procedure begins when the SMC Configuration Management Administrator receives the software maintenance change from the SEO and directs transfer to a designated DAAC drop-off point (SEO on-site SW library). At the DAAC, the installation actions are executed by the site sustaining engineering SW Maintainer under direction from the DAAC Configuration Change Board (CCB).

9.7.2 Applicable to

All ECS sites' Sustaining Engineers, System Administrators, CM Administrators, and Maintenance Engineers.

9.7.3 References

- Software Distribution Scenario, 604-CD-002-001, 305-CD-029-001 (modified)
- COTS Software Upgrade Scenario, 604-CD-002-001, 305-CD-029-001 (adapted)
- M&O CM Plan, 102-CD-002-001
- Developed SW Maintenance Plan, 614-CD-001-002

9.7.4 Procedures

9.7.4.1 Overview

This procedure details the transfer of SW changes under CM control from the ESDIS CCB using SMC resources to distribute maintenance changes to the sites (nominally, to a DAAC). The flowcharts of figure 9.7.4-1 (transfer) and figure 9.7.4-2 (installation) detail the sequence of events involved in the delivering, testing, and installing of SW maintenance releases.

Assumptions:

- The SMC storage software will be ClearCase.
- The baseline records will be maintained in the Baseline Manager (XRP-II tool)
- The SMC transfer of software will be via tar tape or File Transfer Protocol (FTP) for Release A; will be upgraded to Tivoli Courier (Release B)
- The transfer-storage point will be the SEO on-site SW library.
- The Software Maintenance Change package is relatively small and requires no special build/test procedures.
- Resource Planning, Mode Management, and other issues are not addressed in this scenario.

Summary of Procedures:

- CM Process defined Changes to be incorporated by SEO into ECS Operational Baseline
- SW received at SMC from SEO CM Administrator
- Baseline changed via Baseline Manager (XRP-II tool)
- Packaged via ClearCase
- Transferred via tar tape or File Transfer Protocol (FTP) for Release A; will be upgraded to Tivoli Courier (Release B)
- DAAC CCB Approves the Installation of SW Change Package into DAAC Operational Baseline
- SW Change Package Installed at DAAC on selected host computer

Software Transfer & Install
Functional Flow I (Transfer)

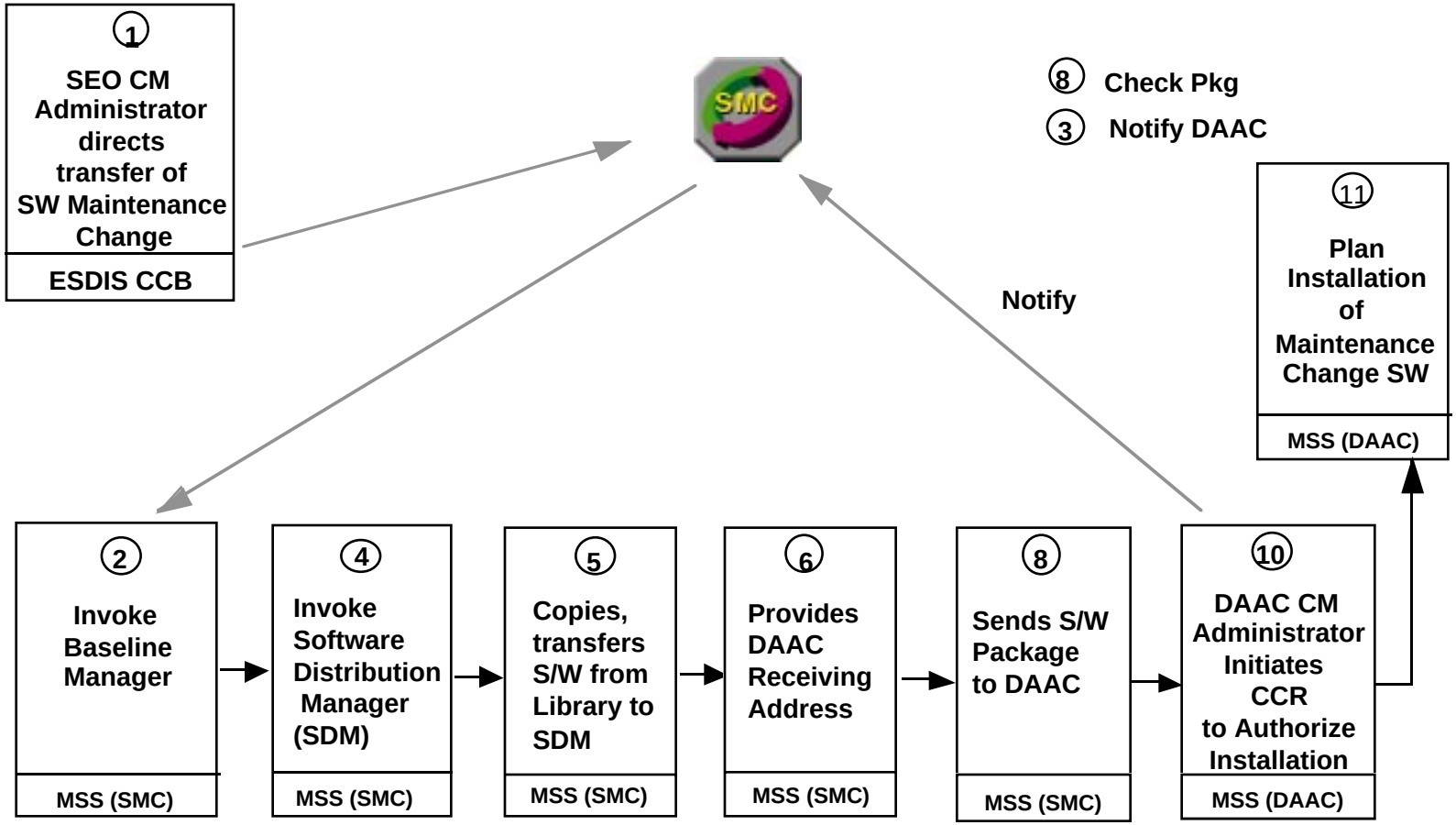


Figure 9.7.4-1. SW Transfer Functional Flow

Software Transfer & Install
Functional Flow II (Install)

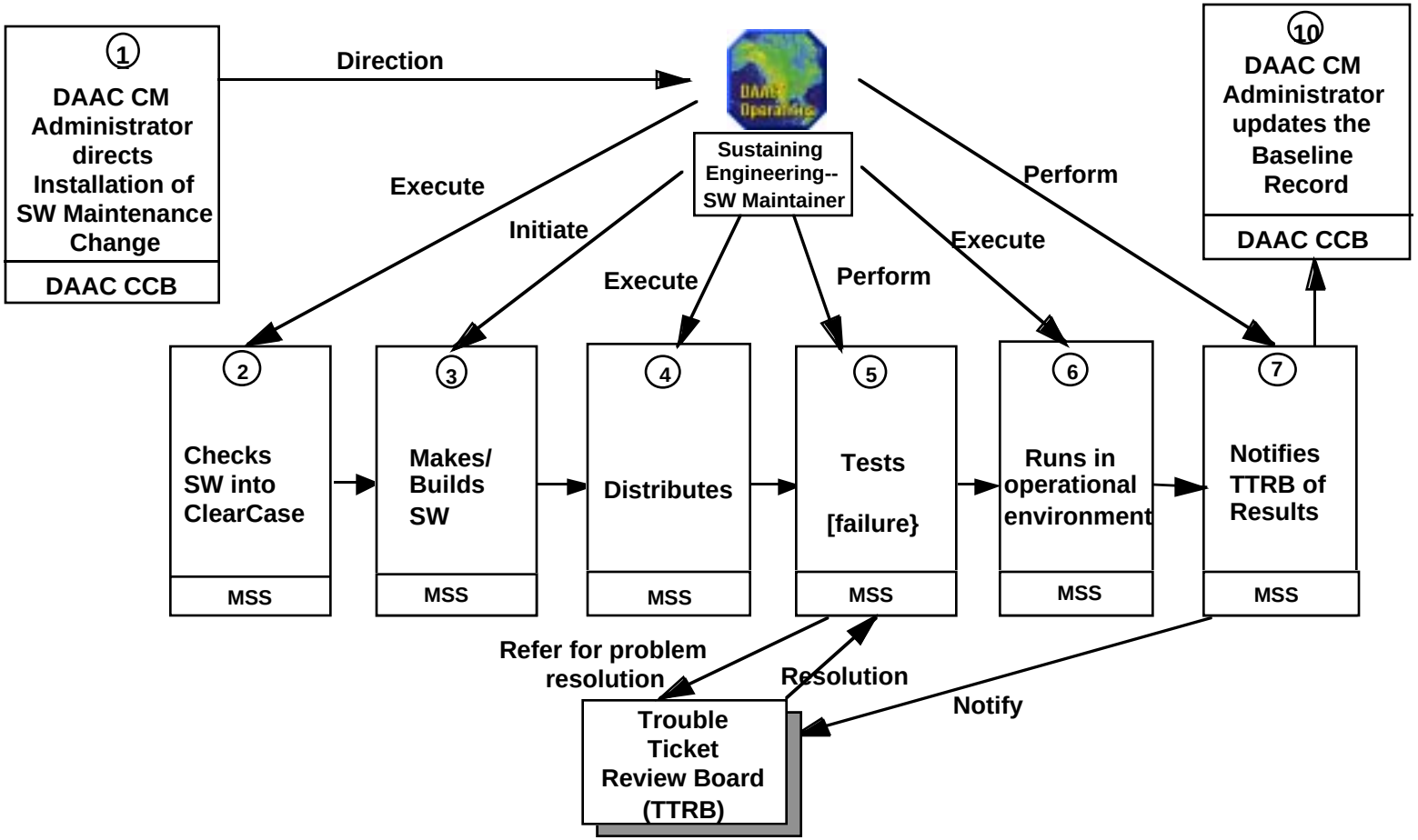


Figure 9.7.4-2. SW Installation Functional Flow

9.7.4.2 Operator Roles

SEO CM Administrator--Ensures that changes to the hardware, software, and procedures are properly documented and coordinated. Maintains control of all configured hardware and software.

SMC CM Administrator--Provides ECS system-wide configuration management and exercise control and/or monitoring over the configurations.

DAAC CM Administrator--Ensures that changes to the hardware, software, and procedures are properly documented and coordinated. Maintains control of all configured hardware and software. Assists in the development and administration of the library with respect to configuration management procedures.

DAAC Sustaining Engineering--SW Maintainer--Produce, deliver, and document the corrections, modifications, and enhancements made to ECS software (including COTS), and/or adapt or incorporate COTS software for ECS use.

9.7.4.3 Detailed Procedures

The following figures are a three part Point of View chart that steps through all the procedure and showing how all relevant roles interact.

Software Transfer & Install
Points of View I (Transfer)

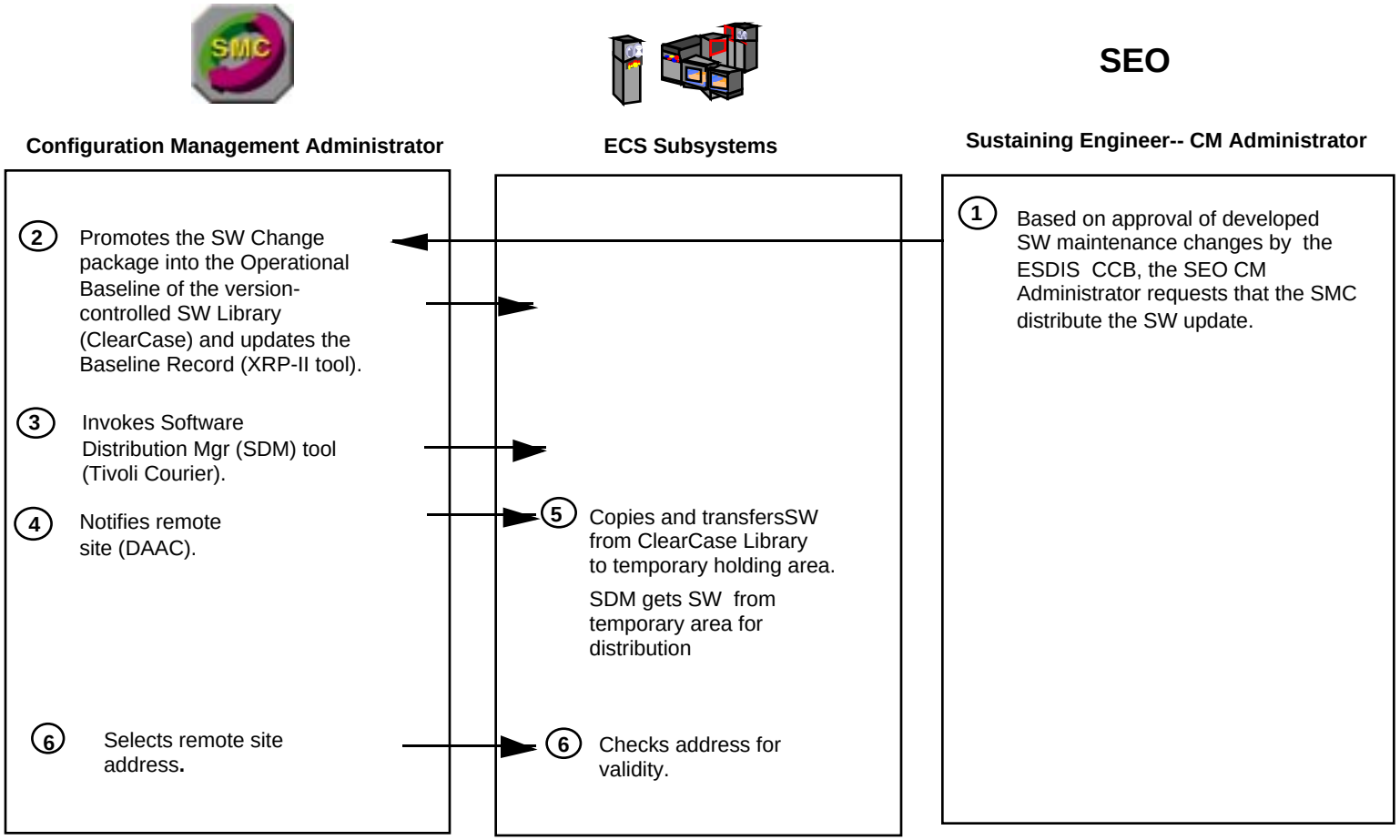


Figure 9.7.4-3. Detailed Points of View

Software Transfer & Install
Points of View II (Transfer)

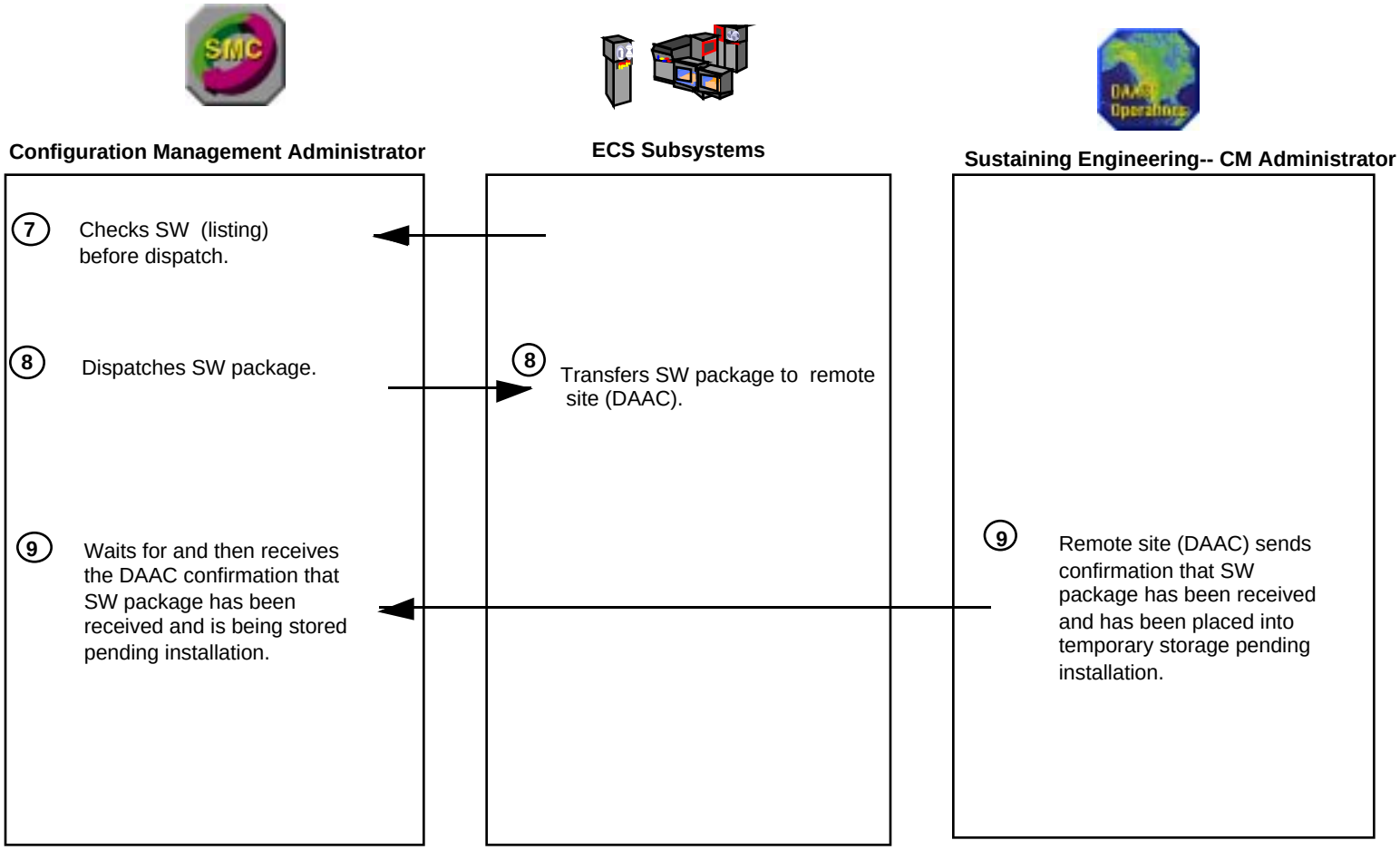


Figure 9.7.4-4. Detailed Points of View

Software Transfer & Install
Points of View III (Install)

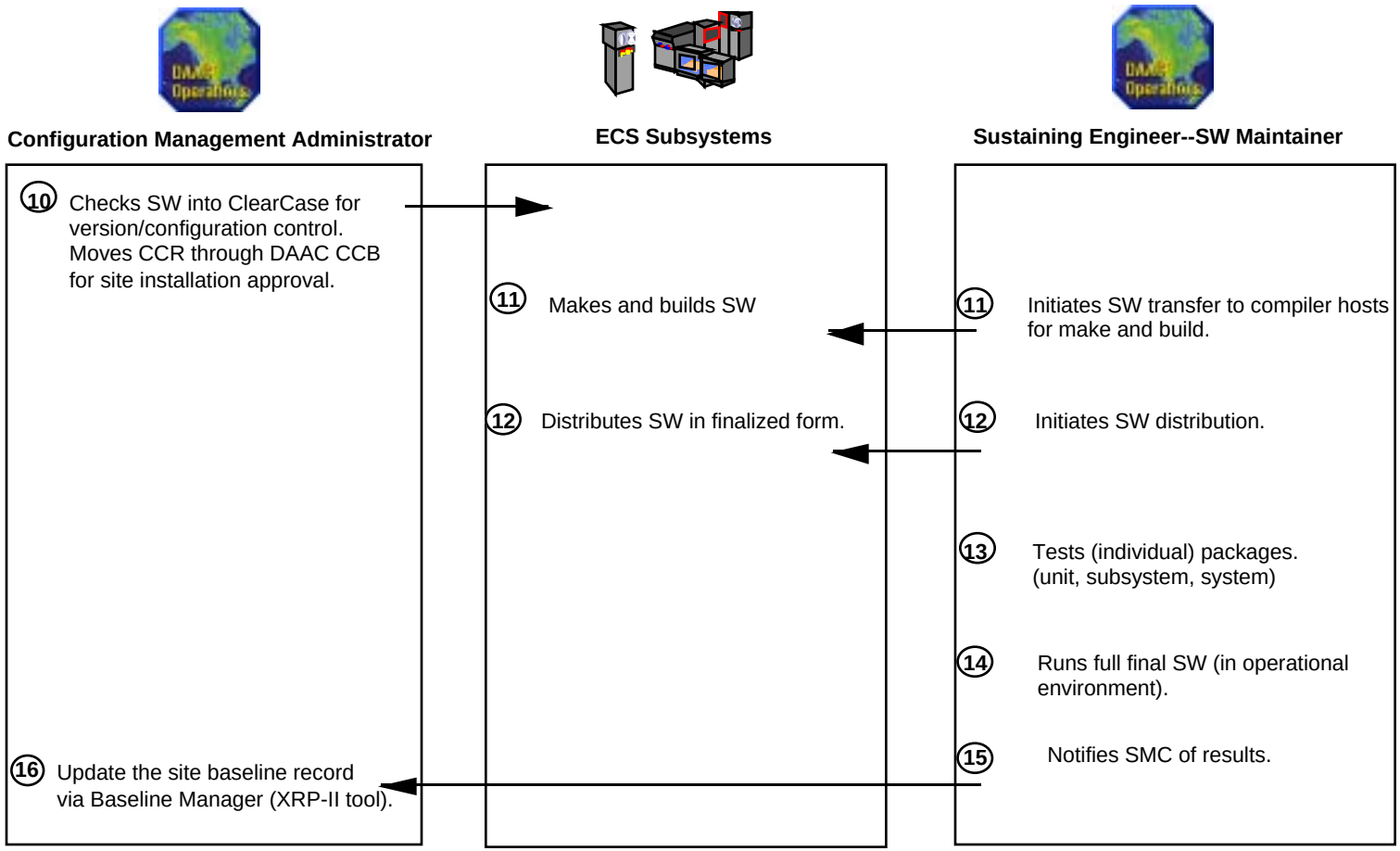


Figure 9.7.4-5. Detailed Points of View

9.7.4.4 Data Activity

The following four tables, 9.7.4-1 through 9.7.4-4, describe data activities associated with the SW transfer and installation workflows of Figures 9.7.4-1 and 9.7.4-2A for SMC Configuration Management Administrator's data maintenance activities and 9.7.4 B&C for site-level (DAAC) configuration management data maintenance by the SEO Maintence Engineer and CM Administrator.

Table 9.7.4-1. Data Activity for Workflow at the SMC

SMC Configuration Distrib tion Manager Management Administrator Role	Data Element	Operator Interactions (Edit, Input, Display)
1. Invoke Software	Package ID Package Name SW Upgrade Name Version Description File Structure Type	Input/Edit Input/Edit Input/Edit Edit Edit Edit Input/Edit
2. Select remote site address (DAAC).	Destination	Input/Edit
3. Check, dispatch S/W package.	Destination	Edit

Table 9.7.4-2A. Data Activity for Workflow at the DAAC

DAAC CM Administrator	Data Element (Object Attribute)	Operator Interactions (Edit, Input, Display)
1. Receives message of SW ready for site installation	E-Mail (send/receive)	Display
2. Processes CCR for DAAC Installation of Software Upgrade using developer's install script stored in ClearCase.	DDTS: DAAC CCR # SW Package ID Package Name SW Upgrade Name Version Description File Structure Type Installation Schedule	Input/Edit

Table 9.7.4-2B. Data Activity for Workflow at the DAAC

DAAC Sustaining Engineer SW Maintainer	Data Element (Object Attribute)	Operator Interactions (Edit, Input, Display)
3. Installs SW Upgrade using the developer's installation script stored in ClearCase	Package ID Package Name SW Upgrade Name Version Description File Structure Type	Input/Edit
4. Verifies that all of the paths and directory structures have been created and are correct.	ClearCase: Description File Structure Type	Input/Edit
5. Runs all diagnostic tests to verify that the new upgrade is operating as expected.	ClearCase: Description File Structure Type	Input/Edit
6. Informs the Resource Manager that the upgrade is completed.	E-Mail (send/receive)	Input/Edit

Table 9.7.4-2C. Data Activity for Workflow at the DAAC

DAAC CM Administrator	Data Element (Object Attribute)	Operator Interactions (Edit, Input, Display)
7. Update the Baseline Record	XRP-II: DAAC CCR # SW Package ID Package Name SW Upgrade Name Version File Structure Type Installation Date	Input/Edit

9.8 Change Request Manager

The commercial off-the-shelf (COTS) product, Distributed Defect Tracking System (DDTS), serves as the Change Request Manager (CRM). DDTS provides the functionality necessary to compose, submit, report, and track status of changes proposed for ECS resources. It provides the M&O staff at the sites and the SMC the capability to register Configuration Change Requests (CCRs). DDTS prompts for relevant information, assigns an identifier, and mails notification of the newly submitted requests to pre-designated personnel. As the CCRs advance through approval and implementation processes, DDTS maintains status, disposition, resolution, and closure information as entered by the M&O staff. It sends notification to pre-designated personnel when the status of the CCR record changes and makes data available for viewing by authorized staff members.

The Activity Checklist table that follows provides an overview of Change Request Manager capabilities. Although Column one (**Order**) shows the order in which tasks might be accomplished, please note that they are independent of each other and can be performed in any order, as necessary. Column two (**Role**) lists the site/organization Configuration Management Administrator (CMA) responsible for performing the task. Column three (**Task**) provides a brief explanation of the task. Column four (**Section)** provides the Procedure (P) section number or Instruction (I) section number where details for performing the task can be found.

Change Request Manager - Activity List

Order	Role	Task	When and Why to Use	Section
1	CMA	View CCR	Operator uses this function whenever he/she wants to quickly view the contents of CCRs in the Index.	(P) 9.8.3
2	CMA	Submit CCR	Operator uses this function whenever there is a new CCR to be entered into the DDTS database.	(P) 9.8.4
3	CMA	Change Status of CCR	Operator uses this function whenever the activities of a particular state have been completed and its time to move to the next state.	(P) 9.8.5
4	CMA	Modify CCR	Operator uses this function to change previously entered data and/or to enter data into fields previously left blank.	(P) 9.8.6
5	CMA	Print CCR	Operator uses this function to obtain a hard or soft copy of a CCR or all of the CCRs in the CCR index.	(P) 9.8.7
6	CMA	Generate Reports	Preformatted reports will be generated for each CCB.	(I) 9.8.9

NOTE: All changes made to the CCR record are monitored by the system and logged in the History enclosure. To view this log, click on the History icon on the DDTS main screen (PureDDTS).

9.8.1 The Configuration Change Request (CCR)

The CCR form has been developed as a medium for processing CCRs throughout the ECS Maintenance and Operations environment for changes processed by the ESDIS CCB and its ECS site-level chartered CCBs at the SMC, EOC, and DAACs (GSFC, LaRC, ASF, EDC, JPL, NSIDC, and ORNL). The information included on the CCR is described below. Copies of the CCR, CCR Impact Analysis Form, and CCR Impact Summary form are provided in Section 9.2. Each CCB will have unique CCR identification sequence numbers. Each CCB can forward CCRs and reports from the Change Request Manager to SMC, where SEO processes system-level CCRs for ESDIS CCB. The ESDIS CM Plan will determine the charter of the respective CCBs and thus the scope of CCR issues to be addressed by the site CCBs.

Section 9.8.4 defines the procedures to enter data from the hardcopy CCR into the Change Request Manager.

Many of the numbered items on the form correspond to the data entry required for CCRs maintained in the Change Request Manager. [Where the hardcopy CCR information is entered in the Change Request Manager tool is defined by referencing appropriate tables found in the subsections that follow.] The Configuration Management Administrator oversees maintenance of the CCR records in the Change Request Manager for his or her respective CCB.

1. **Configuration Control Board (CCB) —** The designated CCB is checked-off for changes processed by the ESDIS CCB and its ECS site-level chartered CCBs at the SMC, DAACs (GSFC, LaRC, ASF, EDC, JPL, NSIDC, and ORNL), and EOC. [This information is not entered into the Change Request Manager.]

2. **CCR Number —** The unique serialized CCR number is applied at each site. This number is system-generated. [Submit record field; see Table 9.8.4-1.]

3. **Submitted Date —** The date that the CCR was prepared is documented. [Submit record field; see Table 9.8.4-1.]

4. **Revision —** The current revision is designated for tracking changed versions of the original CCR. The Revision number is assigned by the Configuration Management Administrator or by the originator of the CCR with approval of the Configuration Management Administrator. [Submit record field; see Table 9.8.4-1.]

5. **Priority —** The priority level of the CCR is assigned by the CCB. Emergency CCRs may have already been implemented on a temporary basis by the Trouble Ticket Review Board (TTRB) with concurrence from the CCB Chair who later receives the CCR to document/implement the permanent change. Urgent items will be reviewed by the next CCB meeting. Routine items will be reviewed as soon as the schedule permits. [Submit record field; see Table 9.8.4-1.]

6. **Change Class —** Change Classes are either I or II. Class I will be handled by ESDIS-only because of cost, schedule, and/or mission impacts that may require requirements changes. Class II items do not affect mission requirements but may have cost and/or schedule implications which affect maintenance, operations, procedures, documentation, site-tailored items, COTS implementation, site installations of core system changes, science SW changes, etc. [Submit record field; see Table 9.8.4-1.]

7. **Status —** Status of the CCR is updated in the Change Request Manager until closed by the CCB. Note that the hard copy form will not be updated but will be kept in the master suspense file of the CM Administrator until closed-out with a stamp (Item 15, below) and appropriate signatures (see Items 16 and 17, below). [Submit record field; see Table 9.8.4-1.]

   Eleven valid status indicators (states) are listed below. The corresponding Change Request Manager State Code (upper-case single character) is provided in parentheses after the status descriptive name. This code is stored in the change request record. The Change Request Manager uses this value to search and extract the descriptive name to display in reports. The descriptive names, for example, Assign-Eval, are as they appear in the

Change Request Manager for selection during input. Some query and report codes use the descriptive name rather than the single letter code to facilitate querying. Following the definition is the status as it appears on the CRR. These status indicators appear in the Change_State Menu, described in Section 9.8.5, below.

Submitted (S) — not used on CCR hardcopy but is system-generated when CCR is input to the Change Request Manager prior to a CCB decision. (Submitted)

New (N) — the initial state for all newly entered change requests. (New)

Assign-Eval (A) — state entered when the change request is being assigned to an engineer for evaluation/analysis. (Assigned-Eval)

Assign-Implement (O) — state entered when the change request is being assigned to an engineer for development. (Assigned-Implement)

Implement (R) — state entered when the proposed change has been developed. (Implemented)

Assign-Verify (T) — state entered when the developed change is being assigned to an engineer for verification testing. (Assigned-Verify)

Verify (V) — state entered when a developed change has been tested and verified that it functions properly. (Verified)

Close (C) — state entered when all activity specified in the change request has been completed or that the approval authority has decided to close it prior to completion of all activity. Referred to as Close-out stamp. (Closed)

Duplicate (D) — state entered when a change request is determined to be a duplicate of an existing change request. Duplicate change request identifies change request being duplicated. (Duplicate)

Defer (P) — state entered when activity on a change proposal has to be postponed. (Deferred)

Forward (F) — state entered when a change request needs to be forwarded to another DDTS-defined project. In DDTS terminology, a project is a grouping of change requests. For example, a change request from a site project can be forwarded to an ECS project. (Forwarded)

8. **CCR Title —** The CCR title is supplied by the originator. [Submit record field; see Table 9.8.4-1.]

9. **Originator —** The originator name, organization, e-mail address, and phone number is given. [Submit record fields; e-mail address not included; see Table 9.8.4-1.]

10. **Approval —** The CCR is approved by the designated management authority which is assigned by the CCB. This sponsorship requirement acts as a primary filter to eliminate from consideration those CCRs that cannot be implemented or which have no ECS site management support. [Assign-Implement field; see Table 9.8.5-2.]

11. **Reason for Change —** The reason for the change is narrated on the form and/or the designated attachment. [This information may be included in the Proposed Change Enclosure; see Section 9.8.4-1, Step 2.]

12. **Description of Change —** The proposed implementation of the change is narrated along with any known impacts, resources, and expenses to be incurred. [This information may be included in the Impact Summary Enclosure; see Section 9.8.5.1, Step 2.]

13. **Impact Analysis —** Impact analysis is documented in the CCR Impact Analysis form. The impact analysis is collected by the CCB Chair appointed Evaluation Engineer in coordination with the CM Administrator who maintains the CM records and assembles the review package for the CCB. The Evaluation Engineer documents the list of Impact Evaluators and derives and/ or verifies cost, technical, and schedule impact of the proposed change based on all inputs received. The results of the coordinated CCR Impact Analysis inputs are presented in the CCR Impact Summary form as part of the CCR review package. [This information may be included in the Impact Summary Enclosure; see Section 9.8.5.1, Step 2.]

14. **Comments —** Comments are added to the CCR to summarize sites and/or organizations affected by the CCR. Additional comments may address proposed CCB dispositions and recommendations to be indicated by resolutions in Item 15, below. [This information may be included in the Resolution Enclosure; see Section 9.8.5.2, Step 2.]

15. **Board Action —** CCB actions and follow-up actions that will be facilitated and tracked by the CM Administrator are indicated. [Assign-Implement field; see Table 9.8.5-2. Also may be applicable to Resolution Enclosure; see Section 9.8.5.2, Step 2.] Possible CCB dispositions are given as approved, withdrawn, disapproved, and deferred (pending follow-up activities by the indicated schedule date). Further actions are indicated as:

    Engineering Change Proposal (ECP) — changed scope of contract requirements.

    Waiver — declaration that certain contract requirements no longer apply.

    Deviation — change of contract terms or substitution of terms or deliverable requirements.

    Technical Direction — order by Contracting Officer's Technical Representative (COTR) to perform certain tasks within the scope of the contract.

    Contract Modification — changes to the terms of a contract.

    Document Change Notice (DCN) — notification of changes to published documents.

    Others — Engineering Change Notice, Change Order, Escalation to higher CCB authority, etc.

16. **CCB Approval —** CCB approval signature authority by CCB Chair or designate. [Assign-Implement field; see Table 9.8.5-2.]

17. **CCR Implemented —** This signature and close-out stamp (Item 7, above) are executed by the CM Administrator witnessing the completion of the CCR implementation process, which is tracked in the Change Request Manager automated tool and updated in Baseline

Manager (XRP-II) for affected version control status changes. [Assign-Implement field; see Table 9.8.5-2.]

Sections 9.8.3 through 9.8.7 define procedures to process a CCR using the Distributed Defect Tracking System (DDTS) application software database tool that implements the Change Request Manager. The procedures, though step-by-step, are not detailed for the novice user. Please refer to the PureDDTS User's Manual whenever further explanation may be required. Relevant sections of the Manual are identified where applicable.

9.8.2 Accessing Change Request Manager

Depending on your site configuration, access to the Change Request Manager, DDTS, will be by clicking an icon from your desktop, or by typing the following at the command line prompt:

➢ **xddts**

The PureDDTS screen is the main screen. It consists of three major areas:

- the CCR Index Display which shows a listing of CCRs;
- the CCR Record page, which displays some of the content of the highlighted CCR in the Index; and
- the Enclosure Display, which shows the initial set of enclosure icons available for CCR update.

From this screen, you initiate all DDTS functions: View CCR, Submit CCR, Change state of CCR, Modify CCR, Print CCR. Reference Chapter 3 of the PureDDTS User's Manual for information concerning the menus and buttons on the DDTS Main Screen.

9.8.3 View a CCR

Entering DDTS brings you to the main screen (PureDDTS). To view any CCR listed in the CCR Index, simply highlight the desired CCR. The CCR is accessed.

9.8.4 Submit a CCR

Clicking the Submit button on the main screen (PureDDTS) will bring up the "Submit A New Change Request" screen. This screen enables the operator to select a class of projects (the Change Request Class is the default class) and a specific project (group of CCRs within the selected class) to which he/she wants to add a CCR. Reference Chapter 2 of the PureDDTS User's Manual for a detailed explanation of the terms, class and project.

As stated above, the selection for class of projects is defaulted for CCR processing, so you won't need to change it:

Submit to which class of projects: Change_Request

To select Project name, either type in your selection or type a question mark as shown:

Project name: ?

A drop-down menu will appear from which to make your selection. (The Configuration Management Administrator can add a project to the list. See the PureDDTS Administrator Manual.)

Click on **Help** to get an explanation for each of the fields shown, how to move within a screen, and how to terminate the submit process.

Once the Configuration Management Administrator enters the desired class of projects and project name, the CCR page displays the CCR record form. This form enables the operator to enter detailed information concerning the proposed change request. Descriptions of the Submit Record fields are listed in Table 9.8.4-1. Table 9.8.4-2 presents the steps performed using DDTS to submit CCRs. It summarizes the procedures as a quick reference. If you are already familiar with the procedures, you may refer to this table. If you are new to the system or have not performed this task recently, please refer to the more detailed steps provided in this section.

Table 9.8.4-1. Submit Record Field Descriptions (1 of 2)

Field Name	Data Type	Size	Entry	Description
CCR Number	character	10	System generated	a unique identifier for this resource change request
Submitted	date	6	System generated	the date this proposed change was first registered
Revision	character	2	Optional	the current revision/amendment to the proposed change
Priority	character	9	Required	the urgency with which a proposed change is needed. Answer must be one of the following: routine, urgent, emergency. The default is routine
Change Class	character	2	Required	the classification that distinguishes change requests according to management level needed for approval. Answer must be I or II. The default is II.
Status	character	17	System generated	the stage this proposed change has reached in its life cycle
Title	character	72	Required	the nomenclature used to identify the proposed change

Table 9.8.4-1. Submit Record Field Descriptions (2 of 2)

Field Name	Data Type	Size	Entry	Description
Originator Name	character	25	Required	name of the person who is the author of the proposed change
Organization	character	30	Required	name of the originator's organization
Phone Number	character	13	Required	phone number where originator can be reached
Organization Evaluation Engineer	character	25	Required	name of the person who initially determines whether or not the proposal has merit and should be entered into the DDTS database
CM Admin. Name	character	8	System generated	name of the individual who registered this proposed change/enters the proposed change into the DDTS database. Note, DDTS uses User's Login ID
Organization	character	5	Required	name of the CM administrator's organization. Answer must be one of the following: ASF, EDC, EOC, GSFC, JPL, LaRC, NSIDC, ORNL, SMC
Phone Number	character	13	Optional	phone number of the CM Administrator

1. Access DDTS by clicking on the DDTS icon from your desktop, or by typing the following command on the command line: **xddts**
2. Enter data in the Submit Record fields.
3. Display the Proposed Change Enclosure Screen by traversing all of the CCR record fields or by clicking on the Proposed Change icon. This enclosure is used to hold additional information about a proposed change. It enables the operator to enter a free text description of the perceived need or problem and a proposed solution. For more information on the enclosure screen see Chapter 3 (Enclosures Section) of the PureDDTS User's Manual.
4. Click the File menu on the enclosure screen and select its "save as" option to save the contents of the enclosure. You will be brought back to the main screen (PureDDTS).
5. When the main screen display reappears, click the "Commit" button to store the CCR record into the DDTS database.

Table 9.8.4-2. Submit a CCR - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	PureDDTS Main Screen. Click on Submit button.	Initiate the CCR record submission process.
2	Enter data in fields and Proposed Change enclosure.	Enter data in Record screen.
3	Click on Commit button.	Store record into DDTS database.

9.8.5 Change State of CCR

The first status (state) assigned to a CCR after it is committed to the DDTS database is "New." Refer to the upper left corner of the center section of screen for the current status of the CCR. When it is time to move the CCR to its next life cycle state, the Change_State Menu on the main screen is used. See Number 7, Status, in Section 9.8.1, above. Table 9.8.5 presents the steps performed using DDTS to change the state of a CCR. It summarizes the procedures as a quick reference. If you are already familiar with the procedures, you may refer to this table. If you are new to the system or have not performed this task recently, please refer to the more detailed steps provided below and to Sections 9.8.5.1 through 9.8.5.5.

1. Access DDTS by clicking on the DDTS icon from your desktop, or by typing the following command on the command line: **xddts**

2. Click on the Change_State Menu to access the available state options available for the CCR record.

3. Select the next state to be assigned. After the next state is selected, the associated data fields (if any) for this new state are accessed.

4. Enter data into the associated data fields for the state, as indicated on the screen. These vary according to state being entered. For descriptions of these states, see Sections 9.8.5.1 through 9.8.5.5: Assign-Eval, Assign-Implement, Assign-Verify, Verify, Close. For the states Duplicate, Defer, and Forward, refer to the DDTS User's Manual (also Section 9.8.1, above).

Table 9.8.5. Change State of a CCR - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	PureDDTS Main Screen. Click on Change_State menu.	Select State.
2	Enter data in fields	Enter data in applicable record screen.
3	Click on applicable Enclosure icon.	Enter data.
4	Click on Commit button.	Store record into DDTS database.

9.8.5.1 Assign-Eval State

The Assign-Eval state indicates that the change request is being assigned to an engineer for evaluation/analysis. Table 9.8.5-1 lists the fields for which values are entered.

Table 9.8.5-1. Assign-Eval Field Descriptions (1 of 2)

Field Name	Data Type	Size	Entry	Description
Evaluation Engineer	character	8	Required	name of the responsible engineer designated to analyze the proposed system change. Use Login user name of the engineer
Organization	character	5	Required	name of the evaluation engineer's organization. Answer must be one of the following: SEO, GSFC, LaRC, ASF, EDC, JPL, NSIDC, ORNL, SMC, EOS
Eval. Engr. Email Address	character	25	Optional	electronic mail address of the evaluation engineer
Impact Evaluators (evaluators 1 -12)	character	5	Optional	collection of names of organizations designated to assess the impact of this proposed change. Answer (s) must be from the following: SEO, ESDIS, GSFC, LaRC, ASF, EDC, JPL, NSIDC, ORNL, SMC, EOC, EDF
Sites Affected (sites 1-9)	character	5	Optional	the collection of names of ECS sites affected by this proposed changes. Answer (s) must be from the following: SMC, GSFC, LaRC, ASF, EDC, JPL, NSIDC, ORNL, EOC

Table 9.8.5-1. Assign-Eval Field Descriptions (2 of 2)

Field Name	Data Type	Size	Entry	Description
Related CCR#	character	10	Optional	the number of another CCR that is related to/associated with this CCR
CI Affected	character	15	Optional	the identifier of the principal configuration item affected by this proposed system change
Docs. Affected	character	56	Optional	the documents identifiers of the system documents affected by the proposed system change
Release Affected	character	10	Optional	the ECS release in which the proposed change is targeted for implementation
Baselines Affected	character	56	Optional	the identifiers of system baselines affected by the proposed change

1. Enter data in the Assign-Eval fields.
2. Display the Impact Summary Enclosure Screen by traversing all of the Assign-Eval record fields or clicking on the Summary Enclosure icon. This enclosure is used to hold free text information concerning the impact of the proposed change based on inputs received from the evaluators.
3. Click the File menu on the enclosure screen and select its “save as” option to save the contents of the enclosure. You will be brought back to the main screen (PureDDTS).
4. When the main screen display reappears, click the “Commit” button to store the next state data into the DDTS database.
5. The selected state, “Assigned-Eval,” is now shown as the current state (Status) of the CCR record.

9.8.5.2 Assign-Implement State

The Assign-Implement state indicates when the change request is being assigned to an engineer for development. Table 9.8.5-2 lists the fields for which values are entered.

Table 9.8.5-2. Assign-Implement Field Descriptions

Field Name	Data Type	Size	Entry	Description
Disposition	character	14	Required	the final decision made by a designated approval official concerning this proposed change. Answer must be one of the following: Approved, Approved_w/cmt, Disapproved, Withdrawn, Deferred
CCB Approval Official	character	25	Required	name of the individual whose decision is reflected in the proposed change's disposition
CCB Approval Date	date	6	Required	the date the final decision was made concerning this proposed change. Required format is yymmdd
CCB Org.	character	5	Required	the name of the organization whose configuration control board have authority to approve the change request. Answer must be one of the following: ESDIS, SMC, GSFC, LaRC, ASF, EDC, JPL, NSIDC, ORNL
Implementation Organization	character	5	Required	name of the organization assigned to implement this proposed change. Answer must be one of the following: SEO, GSFC, LaRC, ASF, EDC, JPL, NSIDC, ORNL
Implement. Engineer	character	8	Required	name of the responsible engineer designated to implement the proposed system change. Use Login user name of the engineer
E-mail Address	character	20	Optional	electronic mail address of the implementing engineer
Start Date-	date	6	Required	date implementation activity is to begin. Required format is yymmdd
Estimated Time to Complete	character	20	Optional	estimated time it will take to develop and unit test proposed change in days or months
Completion-Date	date	6	Optional	the date that the proposed change was completed. Required format is yymmdd
Effective Date-	date	6	Optional	the date that the proposed change is go into operation. Required format is yymmdd

1. Enter data in the Assign-Implement fields.
2. Display the Resolution Enclosure Screen by traversing all of the Assign-Implement record fields or by clicking on the Resolution Enclosure icon. This enclosure is used to hold free text description of the solution for the proposed change request.
3. Click the File menu on the enclosure screen and select its "save as" option to save the contents of the enclosure. You will be brought back to the main screen (PureDDTS).

4. When the main screen display reappears, click the “Commit” button to store the Assign-Implement state’s data into the DDTS database.
5. The Implement state (state entered when the proposed change has been developed) is the state that follows Assign-Implement on the Change_State Menu. No data fields are associated with the Implement state. When Implement is selected, the status is simply changed to Implemented.

9.8.5.3 Assign-Verify State

The Assign-Verify state indicates that the developed change is being assigned to an engineer for verification testing. Data fields appear under the heading, “TESTING INFORMATION” on the Record screen. Table 9.8.5-3 identifies these fields.

Table 9.8.5-3. Assign-Verify Field Descriptions

Field Name	Data Type	Size	Entry	Description
Test. Engr. Name	Character	25	Required	Name of the engineer designated to test the system change
Test Org.	character	5	Required	name of the test engineer's organization. Answer must be one of the following: ASF, EDC, EOC, GSFC, JPL, LaRC, NSIDC, ORNL, SMC
Est. Testing Completion Date	date	6	Optional	the date that the tester expects to have completed his testing activity. Required format is yymmdd

1. Enter data in the Assign-Verify fields.
2. After you have completed your input, the system may automatically bring you back to the main screen (Pure DDTS) or click on **Prev**.
3. When the main screen display reappears, click the “Commit” button to store the Assign-Verify state’s data into the DDTS database.

9.8.5.4 Verify State

The Verify state indicates that a developed change has been tested and verified that it functions properly. The data fields appear under the heading, “VERIFICATION INFORMATION” on the Record screen. Table 9.8.5-4 identifies these fields.

Table 9.8.5-4. Verify State Field Descriptions

Field Name	Data Type	Size	Entry	Description
Test Status	character	1	Required	this is an indicator as to whether or not the item (s) being tested has passed the test. Answer must be Passed or Failed
Enclosure Added	character	1	Required	this is an indicator as to whether or not an enclosure has been to further describe the testing activity. Answer must be Yes or No

1. Enter data in the Verify fields.
2. After you have completed your input, the system may automatically bring you back to the main screen (Pure DDTS) or click on **Prev**.
3. When the main screen display reappears, click the "Commit" button to store the Verify state's data into the DDTS database.

9.8.5.5 Close State

The Close state indicates that all activity specified in the change request has been completed or that the approval authority has decided to close it prior to completion of all activity. Data fields appear under the heading, " CLOSING INFORMATION" on the Record screen. Table 9.8.5-5 identifies these fields.

Table 9.8.5-5. Close State Field Descriptions

Field Name	Data Type	Size	Entry	Description
Closed By	character	25	Required	name of the individual that is closing the CCR
Closing Date	date	6	Required	date that the CCR is closed. Required format is yymmdd
Closer's Organization-	character	5	Required	name of the closing official's organization. Answer must be one of the following: ASF, EDC, EOC, GSFC, JPL, LaRC, NSIDC, ORNL, SMC

1. Enter data in the Close fields.
2. After you have completed your input, the system may automatically bring you back to the main screen (Pure DDTS) or click on **Prev**.
3. When the main screen display reappears, click the "Commit" button to store the Close state's data into the DDTS database.

9.8.6 Modify CCR

There will be times when the operator needs to change the information that was entered previously into the database or to enter information into fields that were not completed initially. The Modify Menu enables modification of database data. Table 9.8.6 presents the steps performed using DDTS to modify a CCR. It summarizes the procedures as a quick reference. If you are already familiar with the procedures, you may refer to this table. If you are new to the system or have not performed this task recently, please refer to the more detailed steps provided below:

1. Access DDTS by clicking on the DDTS icon from your desktop, or by typing the following command on the command line: **xddts**
2. From the Main Menu (PureDDTS) select the CCR you want to modify.
3. Click the "Modify" menu on the main screen Record to bring up the modify options.
4. Select the "Modify Record" option to change existing information and/or to enter information into fields left blank previously.
5. The cursor appears at the first field that may be modified. The modify record mode enables the operator to go through all of the fields that are associated with the current status of the CCR and make changes where appropriate.
6. Once the changes have been made, return to the main screen (Pure DDTS) and click the Commit" button on the main screen to add the changes to the database.
7. Reference Chapter 3, "Modify Menu and the Enclosure Sections," of the PureDDTS User's Manual for additional information as necessary.

Table 9.8.6. Modify a CCR - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	From Record screen, click on Modify menu.	Select option.
2	Enter data in fields	Modify record and/or enclosure.
3	Click on Commit button.	Store record into DDTS database.

9.8.7 Print CCR

The DDTS Print option allows the operator to display a CCR or several CCRs in the CCR index in a selected format on his or her monitor; print a CCR or several CCRs in the CCR index to a printer; or print a CCR or several CCRs in the CCR index to a designated file. Table 9.8.7 presents the steps performed using DDTS to print CCRs. It summarizes the procedures as a quick reference. If you are already familiar with the procedures, you may refer to this table. If you are new to the system or have not performed this task recently, please refer to the more detailed steps provided below:

1. Access DDTS by clicking on the DDTS icon from your desktop, or by typing the following command on the command line: **xddts**
2. From the main screen (Pure DDTS), select CCR or CCRs to be printed.
3. Click on the Options Menu then select **Print**, or click on the "Print" button, to bring up the Printing Options Screen. This screen provides the operator the capability to print a highlighted CCR or all of the CCRs in the index on the main screen in a full page, index, one-line, or three-line format.
4. Select the desired option under "Where to Print" and provide appropriate information for printer and file options. Refer to Chapter 3, "Setting PureDDTS Options," of the PureDDTS User's Manual for additional details about the Printing Options screen.

Table 9.8.7. Print a CCR - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	PureDDTS Main Screen. Click on Print button.	Select options.
2	From Format to Print screen, click on Print button..	Store record into DDTS database.

9.8.8 Required Operating Environment

9.8.8.1 Interfaces and Data Types

The DDTS application will interface only with the ClearCase application at each of the sites. This DDTS/ClearCase interface is facilitated by a ClearCase/DDTS Integration COTS package. DDTS has no interfaces external to the ECS.

9.8.8.2 Databases

The PureDDTS database is a proprietary database that supports the SQL 89 standard. Reference Appendixes F and G of the PureDDTS Administrator's Manual for details about the PureDDTS database.

9.8.8.3 Database Schema

Reference Appendix F of the PureDDTS Administrator's Manual for a description of DDTS' database schema.

9.8.8.4 Database Parameters

Reference Appendixes F and G of the PureDDTS Administrator's Manual for DDTS' database parameters.

9.8.8.5 Command Line Interface

DDTS interfaces with ClearCase only. This interface is facilitated through a ClearCase/DDTS Integration COTS package. No custom command line interface has been developed. This interface supports a verification query by ClearCase for authorization by maintenance engineering staff to change configuration controlled files based on specific CCR authorization.

9.8.8.6 Event and Error Messages

Standard DDTS event and error messages are used. There are no messages unique to the ECS implementation. A list of the PureDDTS event and error messages is not provided in the PureDDTS User's and Administrator's Manuals. However, messages provided during execution of DDTS are self explanatory.

9.8.9 Reports

Standard DDTS reports are to be used. Reference Chapter 3 of the DDTS User's Manual (Setting PureDDTS Options) for information concerning the printing of a CCR report and a description of the available report formats.

Table 9.8.9. Reports

Report Type	Report Description	When and Why Used
Not Applicable		

9.8.9.1 Sample Reports

9.8.9.2 Sample Report (Full Page Format)

Below is a sample CCR report resulting from the use of the DDTS Printing Option (full page format).

```
                    ECS_CHNG_REQ
                                        Page 1/3
CCR Number: MSSdd00630  Submitted   : 960529   Revision:
Priority  : routine     Change Class: II
Status    : Closed      Enclosures  : 3
Title:
 Revise Data Input Screen (Example Only)

CCR ORIGINATOR INFORMATION
  Originator Name: Frank Pace
  Organization   : LaRC
  Phone Number   : (999)234-1289
  Organization Evaluation Engineer: J. Bellamy

CONFIGURATION MANAGEMENT ADMINISTRATOR
  CM Admin. Name: bfloyd
  Organization  : LaRC
  Phone Number  : (999)234-1830
                    ECS_CHNG_REQ
                                        Page 2/3
CCR Number: MSSdd00630

ANALYSIS INFORMATION
  Evaluation Engineer : bfloyd
        Organization : LaRC
       Email Address: bfloyd@larc.com

  Impact Evaluators:
        1. GSFC    2. LaRC   3. EDF    4.        5.        6.
        7.         8.        9.        10.       11.       12.

  Sites Affected:
        1. GSFC    2. LaRC    3. SMC    4.        5.       6.
        7.         8.         9.

  Related CCR# :
  CI Affected  : Planning CSCI
  Documents Affected:
  Release Affected  : Release X
  Baselines Affected:
                    ECS_CHNG_REQ
                                        Page 3/3
CCR Number: MSSdd00630

DISPOSITION: Approved              TESTING INFORMATION:
  CCB Approval Official: John Wana     Engr. Name: Joe Tester
  Date: 960607                       Organization: LaRC
  CCB Organization: ESDIS            Est. Testing Completion
```

Date: 960614

IMPLEMENTATION VERIFICATION INFORMATION:
Organization: SEO Test Status (Pass/Fail): P
Engineer: bfloyd Enclosure Added (Y/N): N
E-mail: efinch@eos.com
Start Date: 960610
Est. Time to Complete: 2 days
Completion Date: 960612
Effective Date: 960710

CLOSING INFORMATION:
Closed by: Authur Closer Date: 960618 Org.: SMC

********** Proposed Change **********
Need or Problem: Describe the need or problem.

The need is -----

Proposed Solution: Describe the proposed solution.

Suggest that the following capability be changed as follows:
- capability changes

********** Impact Summary **********
Summarize the impact statements received from the organizations requested to provide impacts.

Summary of impacts received from GSFC and EDF is ---------

Resources Summarized: [description of resources]

Technical Summary:

ROM Summary (BOE, Cost & Schedule):

Recommendation: [Insert Recommendation]
********** Resolution **********
Describe how the request will be resolved/completed.

This request will be resolved as follows:
- Capability x will be modified to ----.

****

The Resolution Enclosure is the end of the full page format report.

9.8.9.3 Sample Report (Three Line Format)

Below is a sample CCR report resulting from the use of the DDTS Printing Option, Three Line Format. [Note: this document's margins forced the data to a fourth line.]

Submitted 960529, CCR# MSSdd00630, Originator Frank Pace
Title Revise Data Input Screen (Example Only)
Priority routine, Class II, CCB Org. ESDIS, Disp. Approved, Status Closed.

Submitted 960521, CCR# MSSdd00617, Originator Joseph Winkler
Title Add GUI to X11 Program (Example Only)
Priority routine, Class II, CCB Org. LaRC, Disp. Approved, Status Implemented.

9.8.9.4 Sample Report (Index Format)

Below is a sample CCR report resulting from the use of the DDTS Printing Option (Index format). Fields displayed are CCR Identifier, Title, Change Class, Priority, and Status.

MSSdd00630 Revise Data Input Screen(Example Only) II routine C
MSSdd00617 Add GUI to X11 Program (Example Only) II routine R

9.8.9.5 Sample Report (One Line Format)

Below is a sample CCR report resulting from the use of the DDTS Printing Option (One Line format). The operator selects the fields desired for the one line format. In this case, the Identifier, CCR Originator, Originator Organization, Implementing Organization, and Status fields were selected and their data values are displayed.

MSSdd00630 Frank Pace LaRC SEO C
MSSdd00617 Joseph Winkler GSFC LaRC R

9.8.9.6 Report Customization

Refer to Chapter 8 of the PureDDTS Administrator's Manual for explanations of how to customize DDTS reports. Chapter 8 explains how to customize canned reports and how to create and add new reports.

9.9 Use of the Baseline Manager

9.9.1 Overview

Baseline Manager (BM) is used to record and report the as-built operational baseline of ECS products. It contains the configuration record for each baselined product. It identifies products by CI name, description, location, model/version, and component configured articles. It provides traceability to previous configurations.

XRP-II is a commercially available manufacturing management application specially configured to serve as the ECS Baseline Manager. It helps the M&O staff at the DAACs, EOC, and SMC maintain records that describe what comprises baselined operational system configurations. These records identify baselined versions of hardware and software items as well as their assembly

structures and interdependencies. XRP-II keeps chronological histories of baseline changes and traceability of items to predecessor versions and system releases.

Baseline Manager does this primarily by maintaining a catalog of version-controlled items along with data about how they interrelate. The most significant relationship is product structure, which is a parent-component pairing of two control items. Using XRP-II's engineering change notices and effectivity data (active and inactive dates), product structure changes can be reported, checked, and approved before they go into effect, and tracked by control item as well as related Configuration Change Request and/or trouble ticket.

The Baseline Manager data schema is shown in Figure 9.9-1. This is a system-level view consistent with the records kept at the System Management Center spanning all ECS sites in one database. Control items may be mapped via development baselines as shown on the right-hand-side (RHS) of the chart known as the *CIL/CAL View* which is derived from the functional mapping of the Configuration Items List (CIL) and the Configured Articles List (CAL), respectively. Control items may also be mapped via operational baselines (ECS, $Site_i$, M&O, & Release) as shown on the left-hand-side (LHS) of the chart known as the *Logical Operations View of Baselines* which derives from the on-site implementation of ECS. Both views are traceable to the CAL which constitutes the inventory of ECS product delivered end-items. The RHS provides a hierarchical, functional trace of the CAL to configuration items, subsystems, and common items. The LHS provides logical abstractions of the operational implementation view of the CAL items. The RHS is relatively static since it derives from the ECS design while the LHS is more fluid because it is subject to change with operational modifications due to on-site implementation.

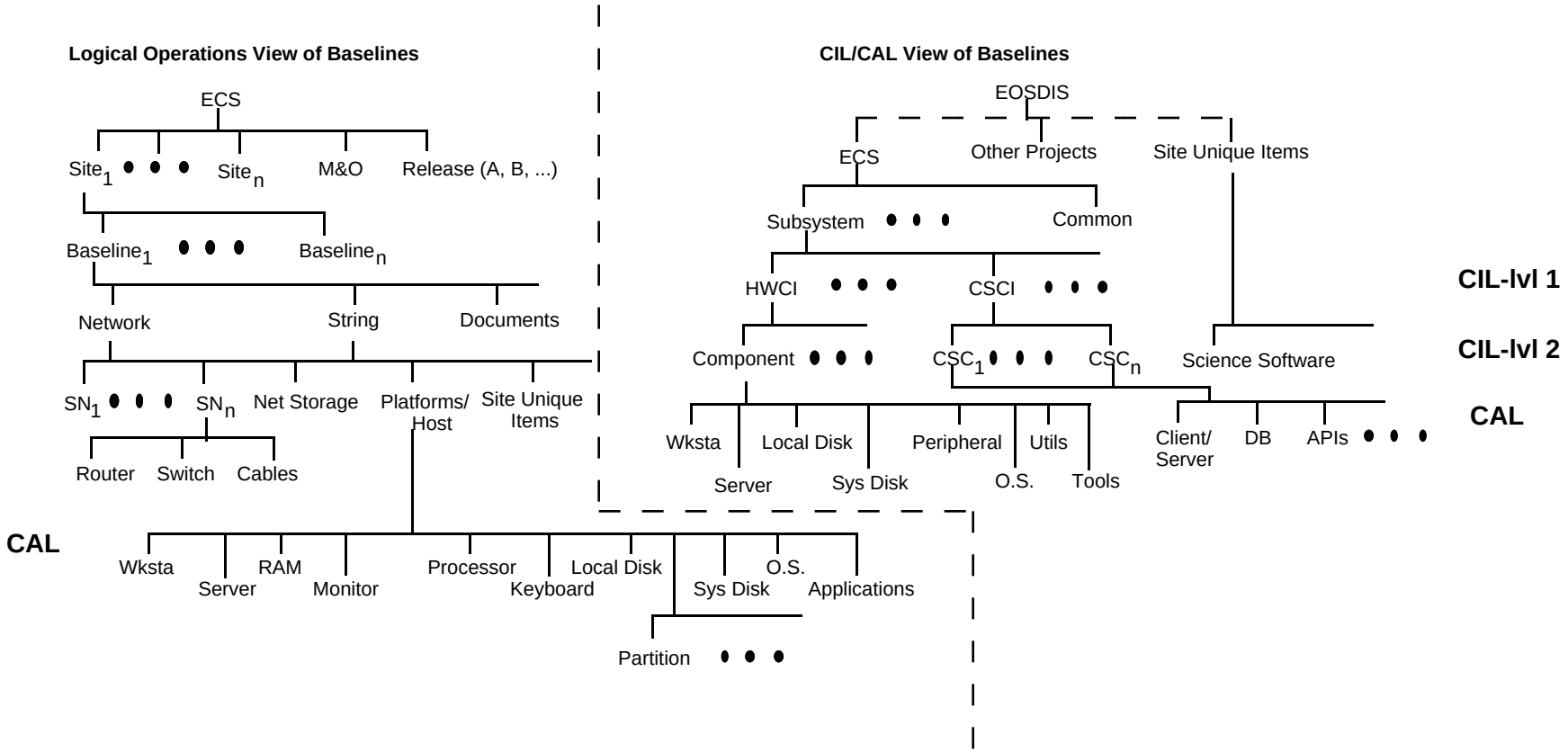
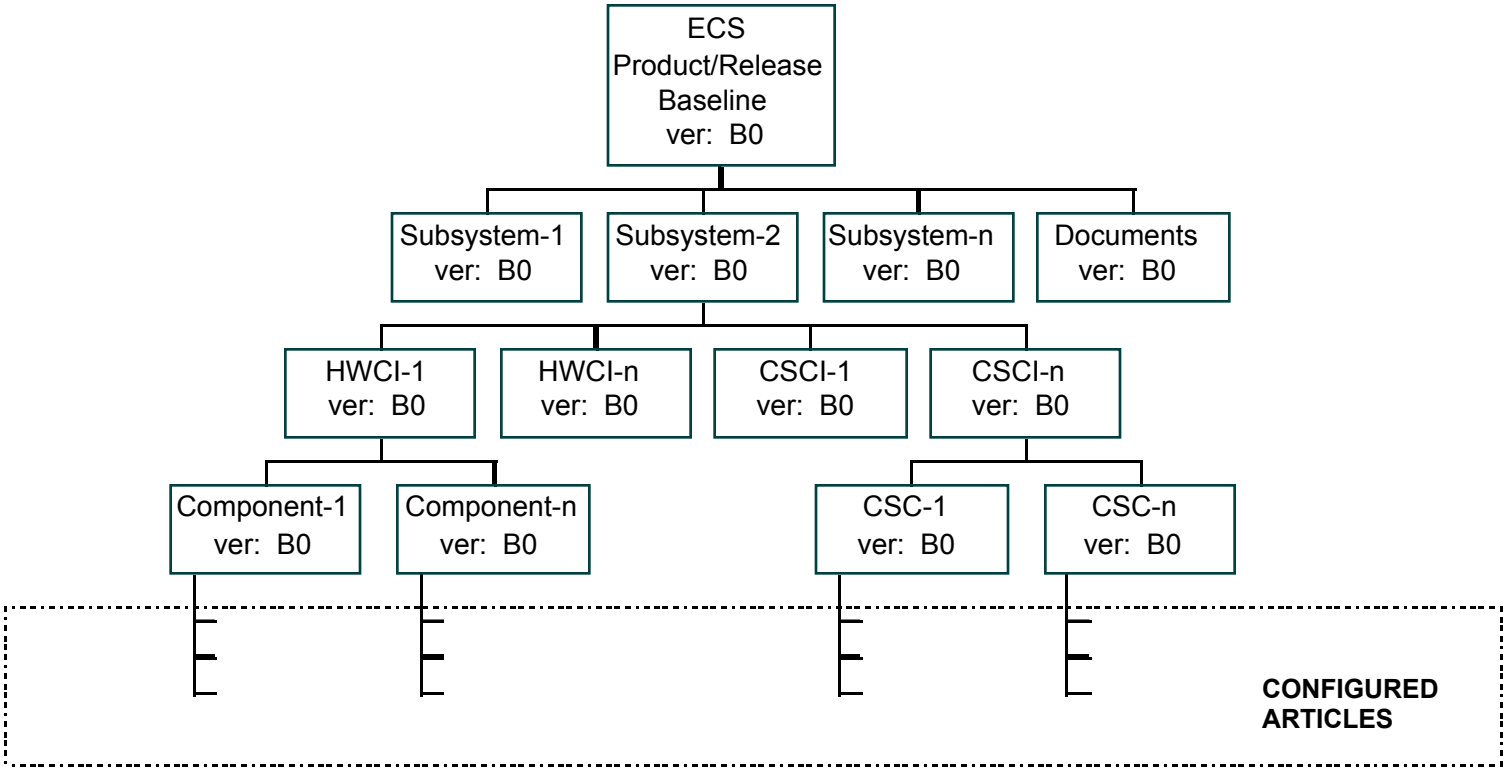


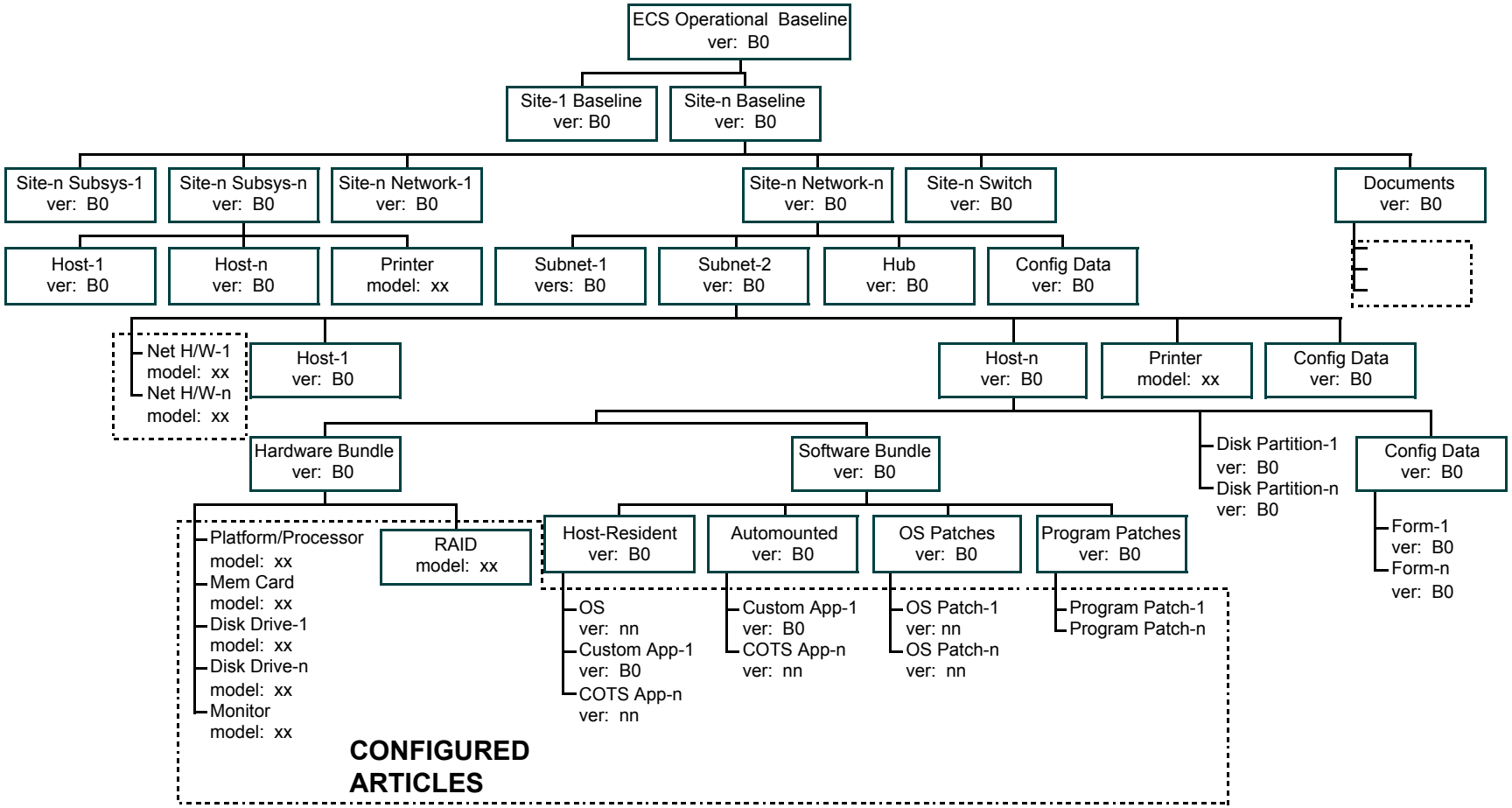
Figure 9.9-1. Hierarchy of Control Items for Baseline Management

A further detailed instantiation of the Baseline Manager (BLM) data schema is shown in figures 9.9-2 and 9.9-3 which show the Design (CIL/CAL) View and the Operational (Logical) View, respectively, for Release B0. These drawings represent the delivered EOSDIS Core System design and on-site build-out, respectively. This is a composite view of the release data delivered to each ECS site. Site-unique items, site created baselines, science software, maintenance changes, and baseline add-ons will be added to produce a site-unique BLM database.



Design (CIL/CAL) View

Figure 9.9-2. Design (CIL/ CAL) View



Operational (Logical) View

Figure 9.9-3. Operational (Logical) View

A series of Figures 9.9-4 (part 1 & 2) for the design view, Figures 9.9-5 (parts 1 thru 4) for the operational view, and Figures 9.9-6 (part 1 &2) for details of the operational view introduce the nomenclature for BLM records. This series of figures depicts the major types of records consistent with the database schema. The records are distinguished as a set of record classes with a common set of parameters for the design view and operational view. Further details elaborate necessary parameters of operational records in Figures 9.9-6. Subsection 9.9.2.2 will supply detailed procedures for the use of of BLM records, screens, and fields.

Parameter	**EOSDIS**	**ECS**	**Site Unique Items**	**Science SW Bundle**	**Common "reserved"**
Control Item ID	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)	<*site prefix*>b<*seq #*> (e.g., GSFb00000001)	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)
Name/ Doc. No. (24 chars max)	EOS Data and Info Sys	EOSDIS Core System	Site Uniques Collection	Science Software Bundle	Common Subsystem
Mnemonic	EOSDIS	ECS	n/a		Common
Description/Title	EOSDIS for AM Launch Ready Release	EOSDIS Core System for AM Launch Ready Release	Collection of Site Unique Control Items for Release B0	System view of science software used for production	Multiple subsystem-level bundle
Item Class	other	design	other	other	design
Item Subclass	program	project	uniques bundle	science bundle	subsystem
Model/ Version	(as appropriate)	B0	(as appropriate)	(as appropriate)	B0
Predecessor/ID	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)
Configured Article	N (no)	N (no)	N (no)	N (no)	N (no)
Resource Planning	N (no)	N (no)	N (no)	N (no)	N (no)
Scope	C (core)	C (core)	U (site-unique)	U (site-unique)	C (core)
Mfr/Dev	n/a	ECS	GSFC	*DAAC mnemonic*	ECS
Commodity Code	n/a	ECS	n/a	other	custom
Detail Items					
Notes **1. control item ID**	**Core**: b*xxxxxxxx* 8 char. with “b” (BLM baseline) or “i” (ILM) suffix	**Site specific**: GSFb*xxxxxxxx* 3 char. prefix for site			

Figure 9.9-4. Common Nomenclature for Baseline Manager Design View Items (1 of 2)

Parameter	Subsystem	HWCI	CSCI	HW Component	SW Component (CSC)
Control Item ID	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)
Name/ Doc. No. (24 chars max)	*subsystem name* (e.g., System Mgmt Subsystem)	*HWCI name* (e.g., Mgmt Hardware CI)	*CSCI name* (e.g., Management Software CI)	*hw component name* (e.g., Ent Mon Svr Comp)	*sw component name* (e.g., Mode Mgmt Service CSC)
Mnemonic	*mnemonic* (e.g., MSS)	*mnemonic* (e.g., MHCI)	*mnemonic* (e.g., MCI)	*mnemonic* (e.g., EMS)	*mnemonic* (e.g., MMS)
Description/Title	*description* (e.g., System Management Subsystem (AM-1))	*description* (e.g., Management Hardware config item (AM-1))	*description* (e.g., Management Software config item (AM-1))	*description* (e.g., Enterprise Monitoring Server hw component (AM-1))	*description* (e.g., Mode Management Service sw component (AM-1))
Item Class	design	design	design	design	design
Item Subclass	subsystem	HWCI	CSCI	component	CSC
Model/ Version	B0	B0	B0	B0	B0
Predecessor/ID	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)
Configured Article	N (no)	N (no)	N (no)	N (no)	N (no)
Planning Resource	N (no)	N (no)	N (no)	N (no)	N (no)
Scope	C (core)	C (core)	C (core)	C (core)	C (core)
Mfr/Dev	ECS	ECS	ECS	ECS	ECS
Commodity Code	custom	other	other	other	other
Detail Items					
Notes **1. control item ID**		**Core**: bxxxxxxxx 8 char. with "b" (BLM baseline) or "i" (ILM) suffix	**Site specific**: GSFbxxxxxxxx 3 char. prefix for site.		

Figure 9.9-4. Common Nomenclature for Baseline Manager Design View Items (2 of 2)

Parameter	Release	M&O	Site	Baseline	Network
Control Item ID	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)	<*site prefix*>b<*seq #*> (e.g., GSFb00000001)	<*site prefix*>b<*seq #*> (e.g., GSFb00000001)	<*site prefix*>b<*seq #*> (e.g., GSFb00000001)
Name/ Doc. No. (24 chars max)	Release <*id*> (e.g., Release B0)	Maintenance & Operations Release	*site name* GSFC DAAC	*baseline name* GDAAC Baseline	*network name* GSFC Local Area Network
Mnemonic			(e.g., GSFC, LaRC)		(e.g., GDAAC LAN)
Description/Title	*description* (e.g., EOS AM-1 Launch Ready ECS Release)	*description* (e.g., EOS AM-1 Launch Ready ECS Maint Release B0.1)	*description* (e.g., GDAAC site baseline)	*description* (e.g., GDAAC Site Operational Baseline for AM-1)	*description* (e.g., GDAAC Local Area Network for AM-1)
Item Class	baseline	baseline	baseline	baseline	network
Item Subclass	release	maint release	site	site	FDDI <u>or</u> Ethernet <u>or</u> user subnet <u>or</u> segment
Model/ Version	*release version* (e.g., B0)	*baseline version* (e.g., B0.x)	*site version* (e.g., GSFC.B0.x)	*baseline version* (e.g., GSF.B0.x.n)	*network version* (*e.g.*, GSFC.B0.x)
Predecessor/ID	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)	<*site prefix*>b<*seq #*> (e.g., GSFb00000001)	<*site prefix*>b<*seq #*> (e.g., GSFb00000001)	<*site prefix*>b<*seq #*> (e.g., GSFb00000001)
Configured Article	N (no)	N (no)	N (no)	N (no)	N (no)
Planning Resource	N (no)	N (no)	N (no)	N (no)	N (no)
Scope	C (core)	C (core) <u>or</u> S (site-specific)	S (site-specific)	U (site-unique)	S (site-specific)
Mfr/Dev	ECS	ECS	ECS	ECS	ECS
Commodity Code	other	other	other	other	other
Detail Items					
Notes **1. control item ID**	**Core**: bxxxxxxxx 8 char. with "b" (BLM baseline) or "i" (ILM) suffix	**Site specific**: GSFbxxxxxxxx 3 char. prefix for site,			

Figure 9.9-5. Common Nomenclature for Baseline Manager Operational View (1 of 4)

Parameter	**Site Subsystem Collection**	**Host**	**Documents**	**Disk Partitions**	**Configuration Data**
Control Item ID	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)
Name/ Doc. No. (24 chars max)	*collection name* (e.g., GSFC Collection of MSS Resources)	*host name* (e.g., g0mss02)	*document number* (e.g., 102-CD-002-001)	*partition name* (e.g., MSS Data)	*Form or Doc #*
Mnemonic	(e.g., GSFC MSS)				
Description/Title	(e.g., Release B0 MSS resources at GSFC)	(e.g., Release B0 MSS CM Server)	(e.g., M&O CM Plan)	*partition desc*	*document title*
Item Class	bundle	host	document	partition	document
Item Subclass	subsystem	*host subclass* (e.g., server, xterm, workstation)	*document type* (e.g., plan)	*partition type* (e.g., system, user)	*config data type* (e.g., assembly drawing)
Model/ Version	B0	B0	n/a	B0	B0
Predecessor/ID	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)
Configured Article	N (no)	Y (yes)	Y (yes)	N(o)	N (no)
Planning Resource	N (no)	Y (yes)	N (no)	Y (yes)	N (no)
Scope	S (site-specific)	S (site-specific)	C (core) <u>or</u> S (site-specific) <u>or</u> U (site-unique)	C (core) <u>or</u> S (site-specific) <u>or</u> U (site-unique)	C (core) <u>or</u> S (site-specific) <u>or</u> U (site-unique)
Mfr/Dev	ECS	ECS	ECS	ECS	ECS
Commodity Code		config	COTS <u>or</u> Custom	Custom	custom
Detail Items					
Notes **1. control item ID**	**Core**: bxxxxxxxx 8 char. with "b" (BLM baseline) or "i" (ILM) suffix	**Site specific**: GSFbxxxxxxxx 3 char. prefix for site,			

Figure 9.9-5. Common Nomenclature for Baseline Manager Operational View (2 of 4)

Parameter	**HW Bundle**	**SW Bundle**	**Host Resident Bundle**	**Automount Bundle**	**Patch Bundle**
Control Item ID	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)
Name/ Doc. No. (24 chars max)	*bundle name* (e.g., Ingest Svr swbndl)	*bundle name* (e.g., Ingest Svr hwbndl	*bundle name* (e.g., Ingest Svr resbndl)	*bundle name* (e.g., Ingest Svr autombndl)	*bundle name* (e.g., SGI ptchbndl for Indigo)
Mnemonic					
Description/Title	(e.g., Core hardware bundle for Ingest Server)	(e.g., Core software bundle for Ingest Server)	(e.g., Core host resident sw bundle for Ingest Server)	(e.g., Core automounted sw bundle for Ingest Server)	(e.g., OS patch bundle for SGI Indigos)
Item Class	hardware	software	software	software	software
Item Subclass	bundle	bundle	resident bundle	automount bundle	OS patch bundle
Model/ Version	B0	B0	B0	B0	B0
Predecessor/ID	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)
Configured Article	N (no)	N (no)	N (no)	N (no)	N (no)
Planning Resource	N (no)	N (no)	N (no)	N (no)	N (no)
Scope	C (core) <u>or</u> S (site-specific) <u>or</u> U (site-unique)	C (core) <u>or</u> S (site-specific) <u>or</u> U (site-unique)	C (core) <u>or</u> S (site-specific) <u>or</u> U (site-unique)	C (core) <u>or</u> S (site-specific) <u>or</u> U (site-unique)	C (core) <u>or</u> S (site-specific) <u>or</u> U (site-unique)
Mfr/Dev	ECS	ECS	ECS	ECS	ECS
Commodity Code					
Detail Items					
Notes **1. control item ID**	**Core**: bxxxxxxxx 8 char. with "b" (BLM baseline) or "i" (ILM) suffix	**Site specific**: GSFbxxxxxxxx 3 char. prefix for site			

Figure 9.9-5. Common Nomenclature for Baseline Manager Operational View (3 of 4)

Parameter	Hardware	Software	Patches		
Control Item ID	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)		
Name/ Doc. No. (24 chars max)	*Vendor Product Name* (e.g., SPARCstorage Array)	*Vendor Product Name* (e.g., Clearcase)	*Vendor Patch ID* (e.g., PHSS_9824)		
Mnemonic		CC			
Description/Title	*Vendor Prod Desc* (e.g., 6.3 GB SPARCstorage Array	*Vendor Prod Desc* (e.g., SW Library, CM Tool)	*Vendor description of Patch*		
Item Class	hardware	software	software		
Item Subclass	*device type* (e.g., ext. hard disk)	*software type* (e.g., application)	*patch type* (e.g., OS patch)		
Model/ Version	*Mfr. Model No.* 101	*product version* (e.g., 2.1 or B0)	*product version* (e.g., 10.01)		
Predecessor/ID	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)	b<*seq #*> (e.g., b00000001)		
Configured Article	Y (yes)	Y (yes)	Y (yes)		
Planning Resource	Y (yes)	Y (yes) (OS only)	N (no)		
Scope	C(ore) or S (site-specific) or U (site-unique)	C(ore) or S (site-specific) or U (site-unique)	C(ore) or S (site-specific) or U (site-unique)		
Mfr/Dev	*Mfr Code* (e.g., SUN)	*Developer Code* (e.g., ATR)	*vendor code* or ECS		
Commodity Code	COTS	COTS	COTS or Custom		
Notes **1. control item ID**	**Core**: bxxxxxxxx 8 char. with "b" (BLM baseline) or "i" (ILM) suffix	**Site specific**: GSFbxxxxxxxx 3 char. prefix for site,			

Figure 9.9-5. Common Nomenclature for Baseline Manager Operational View (4 of 4)

Parameter	Hardware	Software	Host	Document	Disk Partitions
Capacity/Size	*Cap/Size Description*				
Format	*Format Classification*				
Mx Exp Slots	*Total Expansion Slots*				
Total Ports	*Number of serial ports or network connections*				
IO Protocols	*List of Protocols*				
Variant		*Platform Type* (e.g., Sun, HP, SGI, etc.)			
TCP/UDP Port		*Port ID* (e.g., 1744)			
Principal Dir		*Directory Path* (e.g., /usr/ecs/OPS/COTS/atria)			
License Type		(e.g., floating, nodelocked, etc.)			
Total Licenses		*License Qty* (e.g., 150)			
# CPUs			*Total CPUs* *<nnn>*		
Processor ID			*Mfr's Processor ID* (e.g., SuperSPARC+)		
Total RAM			*Total RAM (MB)* *<nnn>*		
# Logical Disks			*# of Logical Partitions* *<nnn>*		
Swap Cache			*Cache Size (in blocks)* *<nnn>*		
Disk Cache			*Cache Size (in blocks)* *<nnn>*		
MAC Address			*<address string>*		
IP Address			*<address string>*		
IO Ports			*Desc or List of IO Port*		

Figure 9.9-6. Common Nomenclature for Baseline Manager Operational View - DETAILS (1 of 2)

Parameter	Hardware	Software	Host	Document	Disk Partitions
Publisher				*Publisher's Code* (e.g., SUN)	
Publ Date				*mm/dd/yy*	
Latest Chg Notice				*Change Notice ID*	
Issue				*Document Edition* (e.g., draft, final, etc.)	
Directory Name					*Logical Name for Device* (e.g., /dev/dsk1/…)
Device Name					*Phys. Name for Device* (e.g., /)
Logical Allocation					(e.g., system, user, etc.)
Partition Size					*Partition Size (in blocks)*
Block Size					*Block Size (in bytes)*

Figure 9.9-6. Common Nomenclature for Baseline Manager Operational View - DETAILS (2 of 2)

At each site, XRP-II manages data about baselined resources deployed to and established at that site. At the SMC, it also offers a consolidated view of control item baseline data system-wide. A variety of reports can be viewed, printed, or saved in a file for posting to an ECS bulletin board for global viewing, and the DAACs, EOC, and SMC can exchange baseline records and reports electronically via formatted files. Additionally, a special file identifying the resources of an operational, production baseline can be produced for the Planning Subsystem and its operators doing resource planning.

Table 9.9-1 summarizes the operator functions that XRP-II supports.

Table 9.9-1. Common ECS Operator Functions Performed with XRP-II (1 of 2)

Operating Function	CHUI	Description	When and Why to Use
Catalog control items (Section 9.9.2.2)	Control Item Master menu; Deployment Maintenance screen	Operators select and use a data entry screen to update records identifying individual control items and to identify the sites at which each is (to be) deployed	Used whenever a new control item is to be added or deleted, or when the characterization of an existing one needs changing
Define/Update what comprises baselines and other control item assemblies (Section 9.9.2.3)	Bill Of Material menu	Operators select and use data entry screens to record engineering change notices and maintain product structure records for control.	Used whenever the component structure of a control item has to be defined or changed
Distribute baseline change records for a release (Section 9.9.2.4)	Export Release Records screen	Operators select one or more control items and initiate creation of formatted files which they can ftp to one or more sites. The files contain all appropriate, related records.	Used whenever a baseline change is to be released from one site for distribution to others.
Incorporate release records at a site (Section 9.9.2.5)	Import Data screen	Operators at the sites import data from tar files ftp'ed from the SMC (or other site)	Used any time an XRP-II formatted file containing database updates is available at a site
Provide site baseline change records to the SMC (Section 9.9.2.6)	Export Site-Unique Changes screen	Operators at the sites create files containing site-unique records changed since the last data export, and can ftp the files to the SMC (and other sites if desired.)	Used as required so the SMC can maintain a current, consolidated database of what comprises operational site baselines system-wide.
Incorporate site baseline change records at the SMC (Section 9.9.2.5)	Import Data screen	Operators at the sites import data from tar files ftp'ed from a site	Used any time an XRP-II formatted file containing database updates is available at the SMC

Table 9.9-1. Common ECS Operator Functions Performed with XRP-II (2 of 2)

Operating Function	CHUI	Description	When and Why to Use
Maintain control item deployment data (Section 9.9.2.7)	Deployment Maintenance screen	Operators maintain records describing the implementation status and installation dates of control items at a site	Used whenever the installation date or implementation status of a control item is established or changed. (Note: Records in the bill of materials for site "production" baselines are used by Resource Planning to maintain its resource inventory.)
Update dependencies among control items (Section 9.9.2.8)	Control Item Interdependency Maint screen	Operators maintain records that define operator-specifiable relationships between any two control items	Used primarily to identify resources that are version-dependent and to correlate documents with the resources they describe
Query Control Item Records (Section 9.9.2.9)	Query menu	Provides access to data retrieval screens	Used to view information about control items, product structures, and change histories, primarily by operators not authorized to change it
Generate Pre-defined Reports (Section 9.9.2.10)	Report menu	Provides access to a variety of pre-defined reports that can be generated according to operator-specified record selection criteria	Used whenever a hard or soft copy of a report is desired
Perform baseline management master files maintenance (Section 9.9.2.11)	Utilities menu	Provides access to a variety of screens for accomplishing tasks that sustain local baseline management operations	Used whenever changes are needed to reference files for baseline management and to force recalculation of certain underlying codes and dates.
Perform XRP-II master files maintenance (Section 9.9.2.12)	System Utilities menu	Provides access to a variety of screens for accomplishing tasks that sustain XRP-II operations system-wide	Used whenever changes are needed to the system's tunable parameters and reference files used throughout the XRP-II system.

The spreadsheet in Table 9.9-2 shows the myriad uses of Baseline Manager by a broad spectrum of ECS Operations personnel. The number sequences in the table have a connotation of access to a defined sequence of BM screens provided via the XRP-II application program.

Table 9.9-2. Screen Allocations to Tasks Groups (1 of 9)

Table 9.9-2: Screen Allocations (2)

Tasks Groups	Tasks	Main	Baseline Management	1. Control Item Master	Data Screens	2. Bill of Material	Data Screens
					1.1 All Control Items		**2.1 Engineering Change Entry**
					1.2 Hardware Items Only		**2.2 Engineering Change Approval**
	N.B.				**1.3 Software Items Only**		**2.3 Product Structure Maint**
	1. Access to the XRP-II screens is controlled via Groups as indicated below:				**1.4 Document Items Only**		**2.4 Replace a Component in All Bills**
	a. mrp = programmer's access				**1.5 Host Items Only**		
	b. xrpadmin = database administrator				**1.6 Partition Items Only**		
	c. xrpupdate = allows changes to records						
	d. xrpquery = allows queries and reports						
	2. Access to individual screens is controlled by set-up as						
	I = Inquire (query & report), A= Add Record, M= Modify Record & D= Delete Record						
Data Base Administrator at Site and SMC							
	Establish access permissions for XRP-II users and groups	1					
	Define screen group names	1					
	Register authorized XRP-II users	1					
	Define printer programs	1					
	Edit screen/ report attributes	1					
	Backup data base	1					
	Restore data base	1					
	Install/ verify XRP-II SW updates	Command Line Procedure					
Site-level CM Admin (DAAC, EOC, & SMC) = xrpadmin							
	Site XRP DBA	All screens plus tools (other applications)					
	I,A,M,D item catalog records	1	2	3	4 (1.1--1.6)		
	I,A,M,D engineering change records	1	2	3			4 (2.1, 2.2)
	I,A,M,D product structure records	1	2	3			4 (2.1, 2.2)
	I,A,M,D implementation status records	1	2	3	4 (1.1--1.6)		
	I,A,M,D interdependency records	1	2	3			
	Generate site-level reports	1	2	3		3	
	Export site-owned record changes to SMC	1					
	Establish/change release/ maintenance baselines	1	2	3	4 (1.1--1.6)	3	4 (2.1, 2.2)
	Define user/group access to XRP	1					
	Synchronize XRP w/other CM tools and sites (###)						
	###DDTS (CCRs)	1	2				
	###ClearCase (SW Versions)						
	###Remedy (Trouble Tickets)						
	###System Updates						
	Import records from SMC						
	Maintain Resp Eng list associated with control items	1	2	3	4		
	Record temp or emergency control item changes	1	2			3	4 (2.1)
	Read all XRP records (incl historical records)	1	2	3			

Table 9.9-2. Screen Allocations to Tasks Groups (2 of 9)

Table 9.9-2: Screen Allocations (2)

Tasks Groups	Tasks	Main	Baseline Management	1. Control Item Master	Data Screens	2. Bill of Material	Data Screens
					1.1 All Control Items		**2.1 Engineering Change Entry**
					1.2 Hardware Items Only		**2.2 Engineering Change Approval**
	N.B.				**1.3 Software Items Only**		**2.3 Product Structure Maint**
	1. Access to the XRP-II screens is controlled via Groups as indicated below:				**1.4 Document Items Only**		**2.4 Replace a Component in All Bills**
	a. mrp = programmer's access				**1.5 Host Items Only**		
	b. xrpadmin = database administrator				**1.6 Partition Items Only**		
	c. xrpupdate = allows changes to records						
	d. xrpquery = allows queries and reports						
	2. Access to individual screens is controlled by set-up as						
	I = Inquire (query & report), A= Add Record, M= Modify Record & D= Delete Record						
	Modify logical operations views of baselines (baseline product structure data)	1	2			3	4
	Generate Resource Planning BOM; forward updated report to RP W/S	**Command Line Only**					
	XRP Backups	"ACCELL" tool					
ECS-level CM Admin = xrpadmin (@ SMC) and xrpquery (@ DAAC & EOC)							
	Import XRP records from site/EDF						
	Generate system-level pre-defined reports	1	2	3		3	
	Generate system-level ad hoc reports	1	2	3		3	
	Export (deploy) release records						
	Establish/change ECS-level baselines	1	2			3	4
Maint Eng/ Sys Admin/ SEO = xrpquery (screens: impstat (M,I) and pdmaint]							
	Read all XRP records (incl historical records)	1	2				
	Modify implementation status for ECN	1	2			3	4
	Maintain Resource Dependency Table						
Test Engineer (on-site or ECS-level) = xrpquery (screens: impstat (M,I) and pdmaint]							
	Query test string records	1	2			3	4
	Modify implementation status for ECN						
Resource Planner = xrpquery							
	Generate data output files report	Command Line Only					
	Review processing string contol items	1	2				
ILS (inventory/ audit capability) = xrpupdate; [screen: pibomsa (I)]							
	Read all records (incl historical records)	1	2				
	Generate reports	1	2				
	Modify implementation status	1	2				
	Add records (COTS)	1	2	3	4 (1.1--1.3)		

Table 9.9-2. Screen Allocations to Tasks Groups (3 of 9)

Table 9.9-2: Screen Allocations (2)

Tasks Groups	Tasks	Main	Baseline Management	1. Control Item Master	Data Screens	2. Bill of Material	Data Screens
					1.1 All Control Items		**2.1 Engineering Change Entry**
					1.2 Hardware Items Only		**2.2 Engineering Change Approval**
	N.B.				**1.3 Software Items Only**		**2.3 Product Structure Maint**
	1. Access to the XRP-II screens is controlled via Groups as indicated below:				**1.4 Document Items Only**		**2.4 Replace a Component in All Bills**
	a. mrp = programmer's access				**1.5 Host Items Only**		
	b. xrpadmin = database administrator				**1.6 Partition Items Only**		
	c. xrpupdate = allows changes to records						
	d. xrpquery = allows queries and reports						
	2. Access to individual screens is controlled by set-up as						
	I = Inquire (query & report), A= Add Record, M= Modify Record & D= Delete Record						
	Audit property against CIL/ CAL	1	2				
Trainer = xrpquery							
	Generate reports for OJT/formal training	1	2				
Librarian = xrpupdate; [screen: pibomsa (I)]							
	Update document records	1	2	3	4 (1.4)		
	Maintain document/control item trace records	1	2				

Table 9.9-2. Screen Allocations to Tasks Groups (4 of 9)

Table 9.9-2: Screen Allocations (2)

Tasks Groups	Tasks	3. Query	Data Screens	4. Report	Data Screens	5. Utilities	Data Screens
			3.1 Control Items Query		**4.1 Multilevel Bill Reports**		**5.1 Unit Of Measure Manager**
			3.2 Hardware Items Query		**4.2 Summarized Bill Report**		**5.2 Unit Of Measure Conversion**
	N.B.		**3.3 Software Items Query**		**4.3 Multilevel Where-Used Reports**		**5.3 Vendor Master Manager**
	1. Access to the XRP-II screens is controlled via Groups as indicated below:		**3.4 Host Items Query**		**4.4 Config Items List - One**		**5.4 Control Item Interdependency Mgr**
	a. mrp = programmer's access		**3.5 Document Items Query**		**4.5 Config Items List - Two**		**5.5 Implementation Status**
	b. xrpadmin = database administrator		**3.6 Partition Items Query**		**4.6 Configured Articles List**		**5.6 Control Item Date Manager**
	c. xrpupdate = allows changes to records		**3.7 Product Structure Query**		**4.7 Version Description Reports**		**5.7 Low Level Code Manager**
	d. xrpquery = allows queries and reports		**3.8 Engineering Change Query**		**4.8 Site Baseline Report**		**5.8 Responsible Engineer Maintenance**
					4.9 Change History		**5.9 Item Class Manager**
	2. Access to individual screens is controlled by set-up as				**4.10 BOM Comparison Report**		
	I = Inquire (query & report), A= Add Record, M= Modify Record & D= Delete Record						
Data Base Administrator at Site and SMC							
	Establish access permissions for XRP-II users and groups						
	Define screen group names						
	Register authorized XRP-II users						
	Define printer programs						
	Edit screen/ report attributes						
	Backup data base						
	Restore data base						
	Install/ verify XRP-II SW updates						
Site-level CM Admin (DAAC, EOC, & SMC) = xrpadmin							
	Site XRP DBA						
	I,A,M,D item catalog records	3	4 (3.1--3.5)				
	I,A,M,D engineering change records	3	4 (3.8)				
	I,A,M,D product structure records	3	4 (3.7)			3	4
	I,A,M,D implementation status records					3	4 (5.5)
	I,A,M,D interdependency records					3	4 (5.4)
	Generate site-level reports			3	4 (4.1--4.10)		
	Export site-owned record changes to SMC						
	Establish/change release/ maintenance baselines						
	Define user/group access to XRP						
	Synchronize XRP w/other CM tools and sites (###)						
	###DDTS (CCRs)			3	4		
	###ClearCase (SW Versions)						
	###Remedy (Trouble Tickets)						
	###System Updates						
	Import records from SMC						
	Maintain Resp Eng list associated with control items	3	4				
	Record temp or emergency control item changes						
	Read all XRP records (incl historical records)	3	4 (3.1--3.8)			3	4 (5.1--5.9)

Table 9.9-2. Screen Allocations to Tasks Groups (5 of 9)

Table 9.9-2: Screen Allocations (2)

Tasks Groups	Tasks	3. Query	Data Screens	4. Report	Data Screens	5. Utilities	Data Screens
			3.1 Control Items Query		**4.1 Multilevel Bill Reports**		**5.1 Unit Of Measure Manager**
			3.2 Hardware Items Query		**4.2 Summarized Bill Report**		**5.2 Unit Of Measure Conversion**
	N.B.		**3.3 Software Items Query**		**4.3 Multilevel Where-Used Reports**		**5.3 Vendor Master Manager**
	1. Access to the XRP-II screens is controlled via Groups as indicated below:		**3.4 Host Items Query**		**4.4 Config Items List - One**		**5.4 Control Item Interdependency Mgr**
	a. mrp = programmer's access		**3.5 Document Items Query**		**4.5 Config Items List - Two**		**5.5 Implementation Status**
	b. xrpadmin = database administrator		**3.6 Partition Items Query**		**4.6 Configured Articles List**		**5.6 Control Item Date Manager**
	c. xrpupdate = allows changes to records		**3.7 Product Structure Query**		**4.7 Version Description Reports**		**5.7 Low Level Code Manager**
	d. xrpquery = allows queries and reports		**3.8 Engineering Change Query**		**4.8 Site Baseline Report**		**5.8 Responsible Engineer Maintenance**
					4.9 Change History		**5.9 Item Class Manager**
	2. Access to individual screens is controlled by set-up as				**4.10 BOM Comparison Report**		
	I = Inquire (query & report), A= Add Record, M= Modify Record & D= Delete Record						
	Modify logical operations views of baselines (baseline product structure data)						
	Generate Resource Planning BOM; forward updated report to RP W/S						
	XRP Backups						
ECS-level CM Admin = xrpadmin (@ SMC) and xrpquery (@ DA							
	Import XRP records from site/EDF						
	Generate system-level pre-defined reports			3	4 (4.1--4.10)		
	Generate system-level ad hoc reports	3	4 (3.1--3.8)	3	4	3	4 (5.3--5.5, 5.8)
	Export (deploy) release records						
	Establish/change ECS-level baselines						
Maint Eng/ Sys Admin/ SEO = xrpquery (screens: impstat (M,I)							
	Read all XRP records (incl historical records)	3	4 (3.1--3.8)			3	4 (5.1--5.7. 5.8)
	Modify implementation status for ECN						
	Maintain Resource Dependency Table					3	4 (5.4)
Test Engineer (on-site or ECS-level) = xrpquery (screens: imps							
	Query test string records	3	4 (3.1--3.6)				
	Modify implementation status for ECN						
Resource Planner = xrpquery							
	Generate data output files report						
	Review processing string contol items						
ILS (inventory/ audit capability) = xrpupdate; [screen: pibomsa							
	Read all records (incl historical records)	3	4 (3.1--3.7)				
	Generate reports			3	4 (4.1--4.10)		
	Modify implementation status					3	4 (5.5)
	Add records (COTS)						

Table 9.9-2. Screen Allocations to Tasks Groups (6 of 9)

Table 9.9-2: Screen Allocations (2)

Tasks Groups	Tasks	3. Query	Data Screens	4. Report	Data Screens	5. Utilities	Data Screens
			3.1 Control Items Query		**4.1 Multilevel Bill Reports**		**5.1 Unit Of Measure Manager**
			3.2 Hardware Items Query		**4.2 Summarized Bill Report**		**5.2 Unit Of Measure Conversion**
	N.B.		**3.3 Software Items Query**		**4.3 Multilevel Where-Used Reports**		**5.3 Vendor Master Manager**
	1. Access to the XRP-II screens is controlled via Groups as indicated below:		**3.4 Host Items Query**		**4.4 Config Items List - One**		**5.4 Control Item Interdependency Mgr**
	a. mrp = programmer's access		**3.5 Document Items Query**		**4.5 Config Items List - Two**		**5.5 Implementation Status**
	b. xrpadmin = database administrator		**3.6 Partition Items Query**		**4.6 Configured Articles List**		**5.6 Control Item Date Manager**
	c. xrpupdate = allows changes to records		**3.7 Product Structure Query**		**4.7 Version Description Reports**		**5.7 Low Level Code Manager**
	d. xrpquery = allows queries and reports		**3.8 Engineering Change Query**		**4.8 Site Baseline Report**		**5.8 Responsible Engineer Maintenance**
					4.9 Change History		**5.9 Item Class Manager**
	2. Access to individual screens is controlled by set-up as				**4.10 BOM Comparison Report**		
	I = Inquire (query & report), A= Add Record, M= Modify Record & D= Delete Record						
	Audit property against CIL/ CAL			3	4 (pick your reports)		
Trainer = xrpquery							
	Generate reports for OJT/formal training	3	4 (3.1--3.8)	3	4 (4.1--4.10)		
Librarian = xrpupdate; [screen: pibomsa (I)]							
	Update document records	3	4 (3.4)				
	Maintain document/control item trace records					3	4 (5.4)

Table 9.9-2. Screen Allocations to Tasks Groups (7 of 9)

Table 9.9-2: Screen Allocations (2)

Tasks Groups	Tasks	6. System Utilities	Data Screens	Data Screens	7. System Tools	Data Screens
			6.1 System Parameters Manager	**6.9 Site Master Manager**		**7.1 Screen Manager**
			6.2 System Defaults Manager	**6.10 Machine Network Maintenance**		**7.2 User Manager**
	N.B.		**6.3 Note Code Maintenance**	6.11 Commodity Code Maintenance		**7.3 Groups Manager**
	1. Access to the XRP-II screens is controlled via Groups as indicated below:		**6.4 Data Delete Utility**	**6.12 Import BLM Records**		**7.4 Screen Permission Control**
	a. mrp = programmer's access		**6.5 Calendar Utility**	**6.13 Export Release Records**		**7.6 Printer Management**
	b. xrpadmin = database administrator		6.6 Calendar Report	**6.14 Export Site-Unique Change Records**		**7.7 Data Dump Utility**
	c. xrpupdate = allows changes to records		**6.7 Transaction Log**	**6.15 Export SMC Change Records**		**7.8 Data Load Utility**
	d. xrpquery = allows queries and reports		**6.8 Transaction Archive**			
	2. Access to individual screens is controlled by set-up as					
	I = Inquire (query & report), A= Add Record, M= Modify Record & D= Delete Record					
Data Base Administrator at Site and SMC						
	Establish access permissions for XRP-II users and groups				2	3 (7.4)
	Define screen group names				2	3 (7.3)
	Register authorized XRP-II users				2	3 (7.2)
	Define printer programs				2	3 (7.6)
	Edit screen/ report attributes				2	3 (7.1)
	Backup data base				2	3 (7.7)
	Restore data base				2	3 (7.8)
	Install/ verify XRP-II SW updates					
Site-level CM Admin (DAAC, EOC, & SMC) = xrpadmin						
	Site XRP DBA					
	I,A,M,D item catalog records					
	I,A,M,D engineering change records					
	I,A,M,D product structure records	2	3	3		
	I,A,M,D implementation status records	2	3	3		
	I,A,M,D interdependency records	2	3	3		
	Generate site-level reports					
	Export site-owned record changes to SMC	2		3 (6.14)		
	Establish/change release/ maintenance baselines					
	Define user/group access to XRP				2	3 (7.2--7.4)
	Synchronize XRP w/other CM tools and sites (###)					
	###DDTS (CCRs)					
	###ClearCase (SW Versions)					
	###Remedy (Trouble Tickets)					
	###System Updates					
	Import records from SMC	2		3		
	Maintain Resp Eng list associated with control items					
	Record temp or emergency control item changes					
	Read all XRP records (incl historical records)	2	3 (6.1--6.3, 6.5, 6.7, 6.9--6.11)		2	3 (7.1--7.6)

Table 9.9-2. Screen Allocations to Tasks Groups (8 of 9)

Table 9.9-2: Screen Allocations (2)

Tasks Groups	Tasks	6. System Utilities	Data Screens	Data Screens	7. System Tools	Data Screens
			6.1 System Parameters Manager	**6.9 Site Master Manager**		**7.1 Screen Manager**
			6.2 System Defaults Manager	**6.10 Machine Network Maintenance**		**7.2 User Manager**
	N.B.		**6.3 Note Code Maintenance**	6.11 Commodity Code Maintenance		**7.3 Groups Manager**
	1. Access to the XRP-II screens is controlled via Groups as indicated below:		**6.4 Data Delete Utility**	**6.12 Import BLM Records**		**7.4 Screen Permission Control**
	a. mrp = programmer's access		**6.5 Calendar Utility**	**6.13 Export Release Records**		**7.6 Printer Management**
	b. xrpadmin = database administrator		6.6 Calendar Report	**6.14 Export Site-Unique Change Records**		**7.7 Data Dump Utility**
	c. xrpupdate = allows changes to records		**6.7 Transaction Log**	**6.15 Export SMC Change Records**		**7.8 Data Load Utility**
	d. xrpquery = allows queries and reports		**6.8 Transaction Archive**			
	2. Access to individual screens is controlled by set-up as					
	I = Inquire (query & report), A= Add Record, M= Modify Record & D= Delete Record					
	Modify logical operations views of baselines (baseline product structure data)					
	Generate Resource Planning BOM; forward updated report to RP W/S					
	XRP Backups					
ECS-level CM Admin = xrpadmin (@ SMC) and xrpquery (@ DA						
	Import XRP records from site/EDF	2		3		
	Generate system-level pre-defined reports					
	Generate system-level ad hoc reports					
	Export (deploy) release records	2		3		
	Establish/change ECS-level baselines					
Maint Eng/ Sys Admin/ SEO = xrpquery (screens: impstat (M,I)						
	Read all XRP records (incl historical records)					
	Modify implementation status for ECN					
	Maintain Resource Dependency Table					
Test Engineer (on-site or ECS-level) = xrpquery (screens: imps						
	Query test string records					
	Modify implementation status for ECN					
Resource Planner = xrpquery						
	Generate data output files report					
	Review processing string contol items					
ILS (inventory/ audit capability) = xrpupdate; [screen: pibomsa						
	Read all records (incl historical records)					
	Generate reports					
	Modify implementation status					
	Add records (COTS)					

Table 9.9-2. Screen Allocations to Tasks Groups (9 of 9)

Table 9.9-2: Screen Allocations (2)

Tasks Groups	Tasks	6. System Utilities	Data Screens	Data Screens	7. System Tools	Data Screens
			6.1 System Parameters Manager	**6.9 Site Master Manager**		**7.1 Screen Manager**
			6.2 System Defaults Manager	**6.10 Machine Network Maintenance**		**7.2 User Manager**
	N.B.		**6.3 Note Code Maintenance**	6.11 Commodity Code Maintenance		**7.3 Groups Manager**
	1. Access to the XRP-II screens is controlled via Groups as indicated below:		**6.4 Data Delete Utility**	**6.12 Import BLM Records**		**7.4 Screen Permission Control**
	a. mrp = programmer's access		**6.5 Calendar Utility**	**6.13 Export Release Records**		**7.6 Printer Management**
	b. xrpadmin = database administrator		6.6 Calendar Report	**6.14 Export Site-Unique Change Records**		**7.7 Data Dump Utility**
	c. xrpupdate = allows changes to records		**6.7 Transaction Log**	**6.15 Export SMC Change Records**		**7.8 Data Load Utility**
	d. xrpquery = allows queries and reports		**6.8 Transaction Archive**			
	2. Access to individual screens is controlled by set-up as					
	I = Inquire (query & report), A= Add Record, M= Modify Record & D= Delete Record					
	Audit property against CIL/ CAL					
Trainer = xrpquery						
	Generate reports for OJT/formal training					
Librarian = xrpupdate; [screen: pibomsa (I)]						
	Update document records					
	Maintain document/control item trace records					

9.9.2 Procedures

9.9.2.1 Getting Started

XRP-II has a character-based user interface which employs screens for data entry and report generation, and menus for navigating to the screens. Data is entered via the keyboard in fields that are traversed from left to right row by row. On data entry screens, labels for fields whose values can be modified are displayed in upper case; those that can not are in initial caps. The database is updated every time a field's value changes, and a record of that change is written to a transaction log. The System Reference Manual describes how to use XRP-II's menus and screens.

Most data entry screens have a form and a table view. Form views offer full screen layouts of a data record's fields, whereas table views offer rows of records in a window that is panned to see columns of fields. System limitations, however, preclude all Baseline Manager form view fields from being displayed in table view.

Numerous functions can be performed at data entry screens. Commands available to an operator are screen-dependent and are listed near the bottom of each screen (hence their name: bottom-line commands). The More command helps the operator cycle through them. Most bottom-line functions are described in the System Reference Manual and Product Information Manual. Any that were added for Baseline Manager are described in the sections below along with the screens to which they pertain.

It is important to note that the UNIFY database management system that XRP-II uses does not support rules requiring entries in specific fields. Baseline Manager attempts some enforcement via the data entry screens, either by establishing default values where feasible when new records are created, or by blocking an operator from cursoring past a null field when in Add, Insert, or Modify modes. However, database updates can occur in ways that bypass these mechanisms. Operators must take care to enter all data needed for generating reports and for producing configuration profiles for resource planners.

To invoke XRP-II, double-click on its icon on the ECS Desktop. This executes a script named "pcs" which determines the operator's terminal type from the environment, prompts for a terminal id, and establishes a Baseline Manger operating environment. The script then starts XRP-II, passing it the operator's userid which it obtains from the system.

The first screen[1] an operator sees upon logging onto[2] XRP-II varies depending on the entry menu and screen group assigned to the operator. These assignments are made in the "users" configuration file described in section D.5 of the System Reference Manual. The assignments are made by groups of allocated tasks and functions that are controlled by the CM Administrator as

---

[1] By convention, XRP-II uses the term "screen" when referring to a window containing a Datalook data entry form or table. Other windows, like the one described here, are called "menus", navigation paths to screens.

[2] An ECS customization bypasses XRP-II's standard login screen, normally the first screen an XRP-II system user would see.

explained in Table 9.9-2, Section 9.9.2.12 *Edit Groups File*, and Section 9.9.3 *Tunable Parameters.*

All XRP-II menus are similar in appearance and function the same way. Only the titles and selections vary. See the System Reference Manual for details on using the menus and screens.

9.9.2.2 Cataloging the Control Items

Data about one or more control items can be added, modified, or deleted by selecting an appropriate data entry screen from the Control Item Master menu. These screens modify the master file (or catalog) that describes control items individually. Each screen accesses a particular set of records and contains a unique set of fields corresponding to a control item's class. However, all screens function the same and all but one have identical bottom-line commands.

Control item master screens are Baseline Manager equivalents of the Part Master Maintenance screen described in section 3.4 of the Product Information Manual. Their fields are tailored for baseline management and the following bottom-line commands are added:

- Bom - navigates the operator to a Multilevel Bill of Materials Query screen for the current item. There, an operator can examine the historical record of first-level components for the item, then exit back to the All Control Items screen.

- Where - navigates the operator to a Multilevel Where-Used Query screen for the current control item. There, an operator can determine where the item (or any other item) is used in any bill of material, then exit back to the All Control Items screen.

- Ec - navigates the operator to an Engineering Change Query screen for the current item, where an operator can view each historical engineering change recorded for the item.

The screens neither set item effectivity dates nor link items to baseline records. Therefore, operators can add or modify control item data anytime a new item is established or whenever details about a previously recorded control item change, without affecting any bills of materials. Operators cannot use these screens to delete an item if the item is a component in another's bill. In this case, the Data Delete system utility must be used instead (see Section 9.9.2.12).

Most fields in a control item's master record are for information only. However, the following play a role in XRP-II processing.

- Control item identifier - Every item in the catalog has its own record and must have a unique identifier. By convention, core and site-specific control items are assigned numerical designators; site-unique control items (those established by a site) are given a 3-character site prefix. XRP-II relies on this differentiation when synchronizing site baseline changes with the database at the SMC (see Section 4.3.3.4 *Special Constraints*).

- Item class - as previously mentioned, the class to which a control item is assigned determines the set of characteristics or details that can be recorded about the item and the screens to be used to update and view the data.

- Planning resource - items marked as planning resources will be reported when Baseline Manager receives requests from resource planners for resource configuration data.

- Config Article - items marked as configuration articles form the basis for several custom ECS reports, including the Configured Articles List and the Version Description Report.
- Deployment - deployment data recorded for a control item influences:
  a) the contents of several ECS custom reports which use deployment site data for selecting and ordering data;
  b) resource configuration data generated for resource planners, which is generated for operational baselines whose implementation status is "production"

The subsections below describes the screens and corresponding fields.

9.9.2.2.1 All Control Items Screen

The All Control Items screen adds, modifies, and deletes general identification information about individual control items. It accesses every control item record, but does not contain fields describing an item's detailed characteristics. Details are handled by class-specific data entry screens described in Sections 9.9.2.2.2 thru 9.9.2.2.6.

Fields on this screen serve as a template for all control item master screens, and are described in Table 9.9-3. The following bottom-line command, however, is unique to this screen:

- **.Details** - navigates the operator to a control item update screen where details can be recorded about the current control item record.

Use bottom-line commands such as Select, Find, Next, and Prior to locate a record, and use Add, Insert, Modify, and Delete to enter update mode. Then edit the catalog via the keyboard, or use the Zoom command (on fields where it is active) to pick values from a list. Use Copypart to create a new item with data values from an existing one. When finished, exit database update mode. Then either proceed to other records or exit the screen.

Table 9.9-3. All Control Items Field Description (1 of 3)

Field Name	Data Type	Size	Entry	Description
control item id	string	20	required	Unique code for a version- or configuration-controlled item
name/ doc #	string	24	optional	Name by which a specific item, engineer, or vendor is known
description/title	string	54	optional	Textual characterization of an entity
mnemonic	string	8	optional	Code (or short name) by which item is referenced (e.g., MLCI)

Table 9.9-3. All Control Items Field Description (2 of 3)

item class	string	16		Group name for control items having common attribute types (software, software, design, document, host, network, site, baseline, partition, project, ...)
item subclass	string	16	optional	Group name that distinguishes among types of control items within a class
model/version	string	24	optional	Textual identifier for a level of functional capability for a control item
pred item	string	20	optional	Identifier of the previous version of a control item
current revision	string	3	(system supplied)	Identifier for the currently-active revision level of the item's product structure
highest revision	string	3	(system supplied)	Identifier for a control item's latest product structure revision
config article	string	1	Y, N	Code for distinguishing between control items that are configured articles and those that are not
planning resource	string	1	Y, N	Code for distinguishing between control items that are reportable for PDPS resource planning and those that are not
scope	string	1	optional ; c, s, u	Code indicating whether an item is core, site-specific or site-unique
deployment	N/A		“optional; zoom to list of deployment records	Link to the deployment table containing the list of sites to which a control item is deployed together with the installation date and implementation status of the control item at each site
mfr/dev	string	3		Coded name of the company/organization that produced a hardware control item

Table 9.9-3. All Control Items Field Description (3 of 3)

resp engineer	string	6	optional; zoom to list of engineers	Link to the name, address, and phone number of an item's responsible engineer or organization
commodity code	string	8	optional	Classification for how the item was produced or obtained (COTS, custom, mod-COTS, GFE, shareware, freeware, other)
site	string	6	optional	Code for the site where the control item record was created
project	string	10	optional; defaults to "ECS"	Name of the principal project under which the item was procured or developed
comment	string	60	optional	Miscellaneous information specific to the control item
code	string	2	optional	Identifier for a type or category of message that can be associated with a control item
note	string	30	optional	A message, used in conjunction with a code, that can be associated with a control item

9.9.2.2.2 Hardware Items Only Screen

The Hardware Items Only screen adds, modifies, and deletes information that identifies and characterizes hardware control items. These would typically be processor units, tape drives, disk systems, CD-ROM drives, and other similar system and network equipment. Only those catalog records whose Item Class field contains the value "hardware" are accessed.

Fields that identify a control item are the same as discussed in Section 9.9.2.2.1. Fields that store the detailed characteristics applicable only to hardware items are described in Table 9.9-4.

Use this screen the same way as described earlier for the All Control Items screen.

Table 9.9-4. Hardware Items Only Field Description

Field Name	Data Type	Size	Entry	Description
mfr	string	6	• optional • zoom to list of manufacturers and developers	Coded name of the company/ organization that produced a hardware control item
capacity/size	string	10	optional	Total capacity (e.g. storage) for a control item
format	string	10	optional	Classification that distinguishes hardware control items according to some technical specification
max exp slots	numeric	4	optional	Number of expansion slots an item contains
total ports	numeric	8	optional	Number of ports (e.g., serial ports, network connections) a control item has
io protocols	string	60	optional	List of protocols a hardware control item supports
install inst	text	N/A	optional	Text containing/ specifying where to find instructions for installing the control item

9.9.2.2.3 Software Items Only Screen

The Software Items Only screen adds, modifies, and deletes information that identifies and characterizes software control items. These would typically be application clients, application servers, databases, and patches whether custom or COTS. It accesses only those catalog records whose Item Class field contains the value “software”.

Fields that identify a control item are the same as discussed in Section 9.9.2.2.1. Fields that store the detailed characteristics applicable only to software items are described in Table 9.9-5.

Use this screen the same way as described earlier for the All Control Items screen.

Note: The installation instructions field is a text field. Text fields can only be displayed in a text box window activated via the Zoom bottom-line command. A “T” at the field indicates data exists in the text box.

Table 9.9-5. Items Only Field Description

Field Name	Data Type	Size	Entry	Description
variant	string	4	optional	Name of the type of computer on which a software control item has been built to run
tcp/udp port	numeric	8	optional	Numeric identifier of port used for communicating with control item
principal dir	string	50	optional	Pathname(s) at which the item is stored
license type	string	10	optional	Method by which software licenses are assigned (floating, user, machine, site…)
total licenses	numeric	4	optional	Number of seats, computers, or persons authorized to use a COTS application concurrently
install inst	text	N/A	optional	Text containing, or specifying where to find, instructions for installing the control item

9.9.2.2.4 Document Items Only Screen

The Document Items Only screen adds, modifies, and deletes information that identifies and characterizes baselined documents, such as specifications, manuals, and version-controlled forms. The screen accesses only catalog records whose Item Class field contains the value “document”.

Fields that identify a control item are the same as discussed in Section 9.9.2.2.1. Fields that store the detailed characteristics applicable only to document items are described in Table 9.9-6.

Use this screen the same way as described earlier for the All Control Items screen.

Table 9.9-6. Document Items Only Field Description

Field Name	Data Type	Size	Entry	Description
title	string	54	optional	Nomenclature used to identify document volumes
document number	string	24	optional	Code by which a document is known
publisher	string	3	optional	Coded name of the company/organization that authored a document control item
publ date	date	N/A	optional	Date associated with a document
latest chg notice	string	6	optional	The most recent change notice or the list of change notices issued for a document
issue	string	8	optional	Nomenclature used to distinguish among versions of a single edition of a document (e.g., draft, final, …)
refs	N/A		• optional • zoom to list of related control items	List of other, associated control items

9.9.2.2.5 Host Items Only Screen

The Host Items Only screen adds, modifies, and deletes information that identifies and characterizes control items that are system hosts. These would be assemblies of processors, software and peripherals such as fully configured servers and workstations, but may include other network-addressable resources for which the data fields on this screen are suited, including routers and switches. The screen accesses only those catalog records whose Item Class field contains the value “host”.

Fields that identify a control item are the same as discussed in Section 9.9.2.2.1. Fields that store the detailed characteristics applicable only to host items are described in Table 9.9-7.

Use this screen the same way as described earlier for the All Control Items screen.

Table 9.9-7. Host Items Only Field Description

Field Name	Data Type	Size	Entry	Description
DNS name/host id	string	24	optional	Local name by which the host is accessed; its Domain Name System name
# cpu's	numeric	4	optional	Number of CPUs in a control item
processor id	string	20	optional	Vendor-provided name for a family of processors
total ram	numeric	4	optional	Amount of main memory (in MB) the item contains
# logical disks	numeric	8	optional	Number of logical disk partitions in the bill of materials for a host
swap cache	numeric	8	optional	Total swap space (in blocks) allocated on a host
disk cache	numeric	8	optional	Total disk cache (in blocks) for a host
MAC address	string	17	optional	Machine's Media Access Control number
IP address	string	15	optional	Network address for a host
IO ports	string	60	optional	List of ports and associated services on a host
install inst	text	N/A	optional	Text containing, or specifying where to find, instructions for installing the control item

9.9.2.2.6 Partition Items Only Screen

The Partition Items Only screen adds, modifies, and deletes information that identifies and characterizes boot and data disk partitions associated with a host computer. It accesses only those catalog records whose Item Class field contains the value "partition".

Fields that identify a control item are the same as discussed in Section 9.9.2.2.1. Fields that store the detailed characteristics applicable only to disk partition items are described in Table 9.9-8.

Use this screen the same way as described earlier for the All Control Items screen.

Table 9.9-8. Partition Items Only Field Description

Field Name	Data Type	Size	Entry	Description
directory name	string	50	optional	Logical name used to access a device
device name	string	32	optional	Physical name used by a machine to access the device
logical allocation	string	32	optional; usually either "system" or "user"	Classification that distinguishes between disk partitions used by the system and those available for use by applications
block size	numeric	4	optional	Size of a block (in bytes) on a device
install inst	text	N/A	optional	Text containing, or specifying where to find, instructions for installing the control item

9.9.2.3 Define/Update What Comprises Baselines and Other Control Item Assemblies

Baselines are control items containing various components, each of which is itself another control item. Accordingly, baselines are sets of control items and can be thought of as assemblies.

XRP-II uses product structure records to describe assemblies. Each such record defines a single parent-component control item pair and contains data pertinent to the pairing, such as its effective dates and the quantity of the component needed to form the parent.

Product structure records are the basis for XRP-II's bill of material processing. They are used to generate most ECS reports, to extract data for resource planning, and to select control item records to distribute when a baseline change is released. Once the component structure of a baseline (or other configuration assembly) has been recorded, operators can generate multilevel bill of material reports to determine what comprised the baseline on any given date.

Control items and their product structure records are associated with engineering change notices (ECNs) in order to implement engineering change control. An engineering change notice is an XRP-II construct that defines for an item the addition of a new first-level component or a change in component quantity, unit of measure, or effective dates. Control items can be affected by more than one engineering change, and an engineering change can encompass more than one assembly. Operators can assign a revision code to a control item that undergoes a product structure change which doesn't affect its form, fit, or function. (This is not the same as an item's version code, which operators assign when an item's form, fit, or function does change.) Operators also assign a configuration change request and/or trouble ticket number for the ECN in order to document the

basis for implementing the change. Engineering changes can be recorded in Baseline Manager at any time, but they must be marked "approved" before any effective dates can be recorded.

The four data entry screens operators use to maintain product structure data are each accessed from the Bill of Material menu which lists

- Engineering Change Entry,
- Engineering Change Approval,
- Product Structure Maintenance, and
- Replace a Component in all Bills.

The first two screens are discussed in the subsections below. The last two are fully described in section 4.4 of the Product Information Manual.

9.9.2.3.1 Engineering Change Entry Screen

Operators use the Engineering Change Entry screen whenever an engineering change for a control item is to be defined or modified. The screen is similar to the Engineering Change Maintenance screen described in the Product Information Manual. Differences are:

- Fields Project, CCR #, TT, Baseline/Release and Approval Date appear in lieu of certain other fields not pertaining to baseline management. Each is an information only field. These fields are described in Table 9.9-9.
- Neither the approval code nor active and inactive dates can be updated from this screen; an Engineering Change Approval screen is used instead (see Section 9.9.2.3.2). This allows operators to record new product structure data without setting an effective date or causing premature adjustments to existing bills of material. Both active and inactive dates are automatically set to the latest system date: Jan 5, 2072.
- The Items command can be used irrespective of the approval code's value.

Enter information about the change using Table 9.9-9 as a guide, then use the Items command to modify the list of components in the parent control item's product structure. The Items page is the same as that described in Section 4.3 of the Product Information Manual, but without the "offset" field which is not used in Baseline Manager. Be sure to enter an appropriate quantity for each component. Components having zero quantity will not appear in their parent's bill of materials.

Table 9.9-9. Engineering Change Entry Field Description

Field Name	Data Type	Size	Entry	Description
parent part	string	20	required; zoom to a list of control items	Identifier of the control item associated with the engineering change
engineering change	string	8	required	Number that uniquely identifies every product structure change action
description	string	54	system supplied	Textual characterization of an entity
project	string	10	optional; defaults to “ECS”	Name of the principal project under which the item was procured or developed
date entered	date		system supplied	Date that the ECN was created
operator id	string		system supplied	Identifier of the operator who created the ECN
approval date	date	N/A	system supplied	Date an engineering change is formally sanctioned by an approval authority
approval code	string	1	system supplied; defaults to “E” for entered	Code that distinguishes among lifecycle stages for engineering changes; must be "A" (approved) for active date to be set
ccr #	string	30	optional	Reference to the CCR authorizing the configuration change
tt	string	15	optional	Reference to the trouble ticket authorizing the configuration change
sort order	string	1	optional ; P, R, or S	Code that specifies how component parts are to be sorted on an engineering change screen's items page
drawing	string	20	optional	Control item identifier of a drawing associated with a control item and engineering change
revision	string	3	optional	Change level for a control item that has undergone a product structure change
implementation code	string	2	optional	Code that distinguishes between permanent and temporary changes
baseline/ release	string	3	optional	Name, version, description, or identifier of a baseline with which an engineering change is associated
active date	date		system supplied (Jan 5, 2072)	Date on which a product structure relationship between two control items is effective
inactive date	date		system supplied (Jan 5, 2072)	Date on which the subject product structure is to be obsoleted or superceded

9.9.2.3.2 Engineering Change Approval Screen

Operators use the Engineering Change Approval screen whenever the approval status of an engineering change is to be modified. This screen is the same as the Engineering Change Entry screen, except that it can modify the approval code and date. Follow the field defintions in Table 9.9-10 for data entries. Changing the approval code to "A" for approved enables active and inactive date fields, allowing operators to set effectivity dates for new product structures.

Enter the new approval code and optional approval date, then add an active and inactive date. These active and inactive entries are used as defaults by the Items page when adding records to the parent's component list. The Copy_dates bottom-line function can be used to propagate active and inactive dates to component items already in the list.

Table 9.9-10. Engineering Change Approval Field Description (1 of 2)

Field Name	Data Type	Size	Entry	Description
parent part	string	20	required	Identifier of the control item associated with the engineering change
engineering change	string	8	required	Number that uniquely identifies every product structure change action
description	string	54	system supplied	Textual characterization of an entity
project	string	10	optional	Name of the principal project under which the item was procured or developed
date entered	date	N/A	system supplied	Date that the ECN was created
operator id	string	8	system supplied	Identifier of the operator who created the ECN
approval date	date		optional	Date an engineering change is formally sanctioned by an approval authority
approval code	string	1	optional; default is "E" (entered)	Code that distinguishes among lifecycle stages for engineering changes; must be "A" (approved) for active date to be set
ccr #	string	30	optional	Reference to the CCR authorizing the configuration change
tt	string	15	optional	Reference to the trouble ticket authorizing the configuration change

Table 9.9-10. Engineering Change Approval Field Description (2 of 2)

Field Name	Data Type	Size	Entry	Description
sort order	string	1	P, R, or S	Code that specifies how component parts are to be sorted on an engineering change screen's items page
drawing	string	20	optional	Control item identifier of a drawing associated with a control item and engineering change
revision	string	3	optional	Change level for a control item that has undergone a product structure change
implementation code	string	2	optional	Code that distinguishes between permanent and temporary changes
baseline/release	string	3	optional	Name, version, description, or identifier of a baseline with which an engineering change is associated
active date	date	N/A	optional; default is latest system date (Jan 5, 2072)	Date on which a product structure relationship between two control items is effective
inactive date	date	N/A	optional; default is latest system date (Jan 5, 2072)	Date on which the subject product structure is to be obsoleted or superceded

9.9.2.4 Distribute Baseline Change Records for a Release

Appropriate baseline management data can be distributed to affected sites whenever a new baseline, configuration item, or other control item is released. This data includes control item catalog, product structures, engineering changes, and deployment records pertaining to the control item being released and all the items in its bill of materials.

Operators use the Export Release Records utility to generate and tar formatted files that can be shipped to any ECS site for loading into XRP-II there. The program is accessed via the System Utilities menu and is described in Section 9.9.2.12.

9.9.2.5 Incorporate Release Records at a Site/ SMC

Occasionally, bulk Baseline Manager data must be loaded into XRP-II. The data, stored as a tar file, is usually received in support of a new release being distributed or a consolidation of site baseline changes at the SMC.

Operators load this data using XRP-II's Import Data utility. This program is accessed via the System Utilities menu and is described in Section 9.9.2.12.

9.9.2.6 Provide Site Baseline Change Records to the SMC

The Baseline Manager at the SMC can maintain consolidated records about operational baselines system-wide. Records created at local sites are shipped to the SMC where they are added to records centrally created at the SMC. To distinguish among them, centrally-created records begin with a number, while records originating at a site have a 3-character site prefix.

Baseline Manager supports this cross-site database synchronization with a transaction log and a program accessed via the System Utilities menu. XRP-II automatically logs a transaction whenever the contents of a Baseline Manager field changes. Operators at the sites periodically use the Export Site-Unique Changes utility to generate a formatted data file based on transactions logged but not previously reported. Operators can have XRP-II ftp the file to the SMC where SMC operators can load the data into the Baseline Manager there. Details about the transaction log and the Export Site-Unique Changes utility can be found in the discussions about system utilities within Section 9.9.2.12.

9.9.2.7 Maintain Control Item Deployment Data

Baseline Manager can maintain status information about the implementation of control items. For each item, an operator can create a record for a site and specify its implementation status and installation date. Deployment records about enterprise control items would normally be maintained at the SMC and distributed whenever a set of release records is shipped to the sites. The implementation status of these items is maintained at the sites as are like records about site-unique control items (those establish the sites). Status updates for the site-unique control items are shipped to the SMC whenever site-unique changes are reported (see Sections 9.9.2.6 and 9.9.2.12 (13) Export Site-Unique Changes Screen). Site updates to enterprise control item records remain at the sites so that SMC deployment data is preserved.

Operators must maintain deployment records because several Baseline Manager functions depend on them. For example, any report which lists control items according to site is generated using the deployment records to determine the site(s) under which the item should appear. This includes the Configuration Item List reports, the Configured Article List report, and the Version Description report. Also, the resource configuration file created for resource planners is built using these records to determine which of the site's operational baseline records has "production" status.

Operators use the Implementation Status Maintenance data entry screen to create and update records. This screen can be accessed via both the Utilities and System Utilities menus. The data entry screen is described in Section 9.9.2.11 (3) *Deployment Maintenance.*

9.9.2.8 Update Dependencies Among Control Items

Individual control items can have relationships -- or interdependencies -- among each other that need to be tracked. For example, application software needs certain operating system versions in order to function correctly, specification documents are associated with specific subsystems and configuration items, certain manuals describe particular configured articles, and each operational baseline derives from a specific release baseline.

XRP-II correlates interdependent pairs of control items via the Control Item Interdependency Maintenance screen. The screen is accessible from the Utilities menu and is described along with other baseline management utilities within Section 9.9.2.11.

9.9.2.9 Query Control Item Records

Operators granted read only permissions to Baseline Manager can retrieve, sort, filter and generate ad hoc reports of XRP-II database records using eight query screens all control items, hardware items, software items, document items, product structure, host items, partition items, and engineering change which are accessed from the Query menu. The six "items" screens closely mirror data base update screens described in Section 9.9.2.2, with an added field to display an item's local implementation status. The menu also has two screens for querying product structure and engineering change records, respectively.

Use the Control Items Query screen to browse the entire control item catalog minus its configuration details. To view full configuration details, use the individual query screens designed specially for hardware, software, document, host, and disk partition items.

Use the Product Structure Query screen to browse historical product structure records and display information about any parent-component control item pair. This is helpful, for example, when trying to determine why a control item is not appearing in a bill of materials as expected. Descriptions of this screen's fields can be found in section 4 of the Product Information Manual.

Use the Engineering Change Query screen to browse the chronology of changes to a particular control item and examine which control items were affected by each engineering change. Refer to Section 4 of the Product Information Manual and Section 9.9.2.3.1 above for descriptions of the screen's fields.

9.9.2.10 Generate Pre-defined Reports

XRP-II produces several reports specifically tailored to support ECS configuration management activities. All contain information derived from records stored only in the XRP-II database on the host where the report is requested. Reports are written to the default printer in effect via the operator's environment settings when XRP-II is started, unless XRP-II has been configured to recognize specific print device destinations (see Section 9.9.2.12 (7) Edit Printers File Screen).

Some reports accept a range of control items on which to report. Ranges are specified the same way as when using the Select and Find bottom-line commands on data entry screens. Section 3.4.5 in the System Reference Manual explains how to enter range selection specifications.

All pre-defined reports available to the operator are accessed via XRP-II's Report menu of screens which includes selections for

(1) **Multi-Level Bill Reports -** defined in Section 6.5 of the XRP-II Product Information Manual.

(2) **Summarized Bill Reports-** defined in Section 6.6 of the XRP-II Product Information Manual.

(3) **Multi-Level Where Used Reports-** defined in Section 6.7 of the XRP-II Product Information Manual.

(4) **Configuration Items List - One:** Table 9.2-11 describes the screen's fields.

Specify a site or range of sites, a date for the configuration, and the number of copies of the report wanted. Then enter "E" for execute. XRP-II will report all ECS configuration items active and deployed at the specified sites as of the specified date.

Table 9.9-11. Configuration Items List - One Menu Screen Field Description

Field Name	Data Type	Size	Entry	Description
site or range	string	16	required; zoom to select from a list of sites	Short name for an ECS site
date of configuration	date	N/A	required	"As of " date used in selecting records from the configuration history of the site(s)
note	string	40	optional	Textual information to be added to the header of the report
configuration items list-1	numeric	2	required	Number of copies wanted

(5) **Configuration Items List - Two:** Operators use this report to produce a list of ECS configuration item components - both hardware and software - grouped by subsystem and configuration item. Table 9.9-11 above describes the screen's fields.

Specify a site or range of sites, a date for the configuration, and the number of copies of the report wanted. Then enter "E" for execute. XRP-II will report all ECS design components active and deployed at the specified sites as of the specified date.

(6) **Configured Articles List:** This report generates a list of the approved set of ECS configured articles in effect on a specified date at a specified site or sites, grouping the articles by parent configuration item. Table 9.9-11 above describes the menu screen's fields.

Specify a site or range of sites, a date for the configuration, and the number of copies of the report wanted. Then enter "E" for execute. XRP-II will report all active ECS configured articles for the specified sites as of the specified date.

(7) **Version Description Reports:** The Version Description Reports are used to generate a summarized bill of the approved set of ECS version controlled configured articles for a specified configuration item, subsystem, or release. Table 9.9-11 above describes the screen's fields.

Specify a site or range of sites, a date for the configuration, and the number of copies of the report wanted. Then enter "E" for execute. XRP-II will report all active ECS configured articles for the specified sites as of the specified date.

(8) Site Baseline Report: Operators use the Site Baseline Report to produce an indented bill of materials that lists what comprises one or more sites' operational baselines down to the configured article level. The screen works the same as the multilevel bill screen described in section 6.5 of the Product Information Manual, except it rejects identifiers for items that are not members of class "baseline".

Specify the identifier of the baseline control item and date of bill, then set the remaining parameters for the report. Enter "E" for execute. XRP-II will report all ECS configured articles active and deployed as part of the specified baseline.

(9) Change History: The Change History report generates a list containing the revision history of an ECS control item. Table 9.9-12 describes the screen's fields.

Specify the control item's identifier, then enter "E" for execute. XRP-II will report all versions and product structure revisions for the specified item together with details associated with the item change.

Table 9.9-12. Change History Field Description

Field Name	Data Type	Size	Entry	Description
control item id	string	20	required; zoom to select from a list of control items	Unique code for a version- or configuration-controlled item
note	string	40	optional	Textual information to be added to the header of the report
change history report	numeric	2	required	Number of copies wanted

(10) BOM Comparison Reports: The BOM Comparison Reports menu screen generates a list of the differences in the bills of material for any two control items. Operators use it, for example, to determine how approved operational baselines at two sites differ. Table 9.9-13 describes the screen's fields.

Specify identifiers for the two control items whose bills are to be compared. Next, indicate a bill of materials date for the comparison and the number of report copies wanted. Then enter "E" for execute, and XRP-II will produce a four-part report. For each of the two control items, XRP-II will first list all the control items that are in its bill and in the other's, then all the control items in its bill that are <u>not</u> in the other's.

Table 9.9-13. BOM Comparison Reports Field Description

Field Name	Data Type	Size	Entry	Description
control item id	string	20	required; zoom to select from a list of control items	Unique code for a version- or configuration-controlled item
date of bill	date		required	"As of " date used in selecting records from the configuration history of the control item
note	string	40	optional	Textual information to be added to the header of the report
bom comparison reports	numeric	2	required	Number of copies wanted

Screens for generating Multilevel Bill, Summarized Bill, and Multilevel Where Used Reports are fully described in Sections 6.5, 6.7, and 6.9 of the Product Information Manual. Section 9.9.8 contains sample reports.

9.9.2.11 Perform Baseline Management Master Files Maintenance

XRP-II groups together several programs that help standardize values for certain baseline management data, support product structure administration, and manage types of data for which a separate menu is not warranted. The screens supporting these programs are accessed via the Utilities menu.

The Unit of Measure Maintenance screen maintains codes and descriptions for Baseline Manager's units of measure. These codes are used on bill of material data entry screens, which can access them via "zoom" command to facilitate data entry and promote data consistency. The screen is described in section 3.2 of the Product Information Manual.

The Unit of Measure Conversion screen maintains factors used for converting between units of measure. This standard XRP-II capability, not used by Baseline Manager, is retained for use with inventory, logistics, and maintenance management functions scheduled for delivery in Release B. It is described in section 3.3 of the Product Information Manual.

The Control Item Date Maintenance screen supports a utility that is functionally equivalent to the Part Master Date Maintenance utility described in section 4 of the Product Information Manual. It cycles through product structure and product history files, determining the earliest and latest dates on which each control item is employed as a parent or component, and updates the active and inactive dates in the control item records accordingly. This utility should be run periodically (perhaps monthly), but need only be run if any product structures have been (or may have been) deleted, or if the bill of material screen "Replace a Component in All Bills" has been used. This utility is described in section 7.2 of the Product Information Manual.

The items that follow provide details for the Baseline Manager Utility menu's screens not discussed above

(1) **Vendor Master Maintenance Screen:** This screen maintains a list of companies in order to facilitate entering manufacturer and developer codes for control items and promote data consistency across control item records. An operator updating the control item catalog can select manufacturer/developer codes from this list via the "zoom" command on control item master screens.

Use this screen to update the list of manufacturers or developers. Table 9.9-14 describes its data entry fields.

Table 9.9-14. Vendor Master Maintenance Field Description

Field Name	Data Type	Size	Entry	Description
vendor id	string	6	required	Coded name of the company/organization that produced a control item
name	string	30	optional	Name by which a specific manufacturer or developer is known
qualification code	string	1	optional	Code that distinguishes vendors according to status as a source for a control item

(2) **Control Item Interdependency Maintenance Screen:** This screen maintains information about dependencies between any two control items. Any control item pair can have multiple dependency relationships; however, each of the control items must exist in the control item catalog. Dependency types are operator-specified and non-constrained. Baseline Manager does <u>not</u> maintain an on-line chronological history of dependencies or dependency changes for a control item, but records can be both downloaded and printed so they can be saved for historical purposes.

Use this screen to add, delete, modify, or browse dependency records. The table view is particularly well-suited for displaying lists of all dependencies associated with a given control item. Placing the cursor on either of the control items activates the bottom-line commands "Where" and "Bom". Use the "Where" command to view product structure records in which the control item is a component in some other item's bill. Use the "Bom" command to view product structure records in which the control item is a parent with a bill containing other control items.

This screen's fields are described in Table 9.9-15.

Table 9.9-15. Control Item Interdependency Maintenance Field Description

Field Name	Data Type	Size	Entry	Description
item one	string	20	required; zoom to select from control items list	Unique code for a version- or configuration-controlled item (normally, a control item id) that is the subject of the dependency relationship.
name	string	24	system supplied	Name by which a specific item, engineer, or vendor is known
description	string	54	system supplied	Textual characterization of an entity
version	string	24	system supplied	Textual identifier for a level of functional capability for a control item
revision	string	3	system supplied	Identifier for the currently-active revision level of the item's product structure
dependency type	string	20	required	Code or nomenclature describing the relationship between two control items
item two	string	20	required; zoom to select from control items list	Unique code for a version- or configuration-controlled item (normally, a control item id) that is the object of the dependency relationship

(3) Control Item Deployment Maintenance Screen: This screen maintains information about the deployment and implementation of control items system-wide. One record can be created for each control item for each site. The control item must exist in the control item catalog, and the site must exist in the site list (see Section 9.9.12 (8) *Site Master Maintenance Screen*). Baseline Manager does <u>not</u> maintain a chronological history of deployment data or deployment data changes for a control item or a site.

Use this screen to maintain and browse all control item deployment records. Use the table view to display comprehensive lists of the sites to which a control item is (being) deployed and of the control items (being) deployed to a site.

Note: The field “Implementation Status” has special significance in extracting resource profiles for resource planners. Control records are selected only if the items they describe are in the bill of materials of site operational baselines whose implementation status is “production”.

Table 9.9-16. Control Item Deployment Maintenance Field Description

Field Name	Data Type	Size	Entry	Description
control item id	string	20	required; zoom to select from control items list	Unique code for a version- or configuration-controlled item
site	string	6	required; zoom to select from a list of sites	Mnemonic or short name for an ECS site
installation date	date	N/A	optional	Designated date a control item is to be or was installed at a site
implementation status	string	20	optional;	Classification of a control item according to operational life cycle state (e.g., projected, installed, production, inactive, maintenance, and test

(4) **Low Level Code Maintenance Screen:** This utility is equivalent to the Low Level Code Maintenance utility described in section 4 of the Product Information Manual, except options are available for specifying a subset of records to process. It regenerates low level codes maintained in the control item file by the system logic for its own use (see section 4.2 of the Product Information Manual). These codes affect the efficiency of certain XRP-II processing. They also appear in the headers of bill of material and where used reports to identify the deepest level at which a subject control item is found in any recorded product structure tree.

This utility should be run periodically (perhaps monthly), but need only be run if any product structure records have been (or may have been) deleted. The screen explains options provided to operators.

(5) **Responsible Engineer Maintenance Screen:** This screen maintains data about individuals in order to facilitate identifying points of contact for individual control items. An operator updating the control item catalog can select a code for an engineer from this list via the "zoom" command on control item master screens.

Use this screen to update the list of responsible engineers for control items. Table 9.9-17 describes its data entry fields.

Table 9.9-17. Responsible Engineer Field Description

Field Name	Data Type	Size	Entry	Description
engineer	string	6	required	Organization or task code assigned to an engineer assigned responsibility for a control item
name	string	30	optional	Name of the responsible engineer
address	string	30	optional	Street address where the responsible engineer is located
city	string	20	optional	Name of the city in which the responsible engineer is located
state	string	2	optional	Name of the state in which the responsible engineer is located
zip	string	10	optional	Postal code where the responsible engineer is located
phone	string	8	optional	Phone number for the responsible engineer

(6) Serial Numbers Maintenance Screen: The Serial Numbers Maintenance screen maintains a list of ECS-issued serial numbers or other inventory-related identifier numbers indexed to control items, providing a rudimentary inventory tracking mechanism. The same serial/inventory number can be used in multiple records, one for each control item associated with it.. Similarly, a control item identifier can be used in multiple records, one for each serial number associated with it. Multiple records for a serial- or inventory-control item pair can also be maintained so item movements can be tracked by date. The fields for this screen are described in Table 9.9-18.

Table 9.9-18. Serial Numbers Maintenance Field Description

Field Name	Data Type	Size	Entry	Description
serial number	string	12	required	Identifier for a control item unit produced or deployed
control item id	string	20	optional; zoom to select from control item list	Unique code for a version- or configuration-controlled item
date	date	N/A	optional	Date of transaction
site	string	6	optional	Mnemonic or short name for an ECS site

9.9.2.12 Perform XRP-II Master Files Maintenance

XRP-II groups together several programs that help standardize values for all of XRP-II and support system administration functions such as user access control and inter-site data exchange.

The screens supporting these programs are accessed via the System Utilities menu. The Note Code Maintenance, Data Delete Utility, and Shop Calendar Utility are fully described in the System Reference Manual. The remaining utilities are discussed in the items that follow.

(1) **System Parameter Maintenance Screen:** This screen maintains system-wide, XRP-II parameters and is principally used when first installing the system. Since Baseline Manager uses only a subset of the full XRP-II capabilities, this is a scaled down version of the screen described in section 6 of the System Reference Manual. It contains only the fields needed to tailor the system to the site at which it operates as summarized in Table 9.9-19.

The system parameter key is the key field of the system parameter file. The value "A" designates the active record, which is set when the database is created and must always be present in order for XRP-II to function.

In the site identifier field, enter the code for the ECS site where the copy of XRP-II that the operator is using is installed. This data is used primarily in report headers and file names that XRP-II creates.

In the last control item identifier field, enter a value XRP-II is to use in determining the next available identifier when an operator requests a default for a new item being added or inserted in the control item catalog. The value must end in a numeral, which XRP-II automatically increments. This identifier should always have the site's designated prefix: the first three characters of the site identifier.

Table 9.9-19. System Parameters Maintenance Field Description

Field Name	Data Type	Size	Entry	Description
system parameter key	string	1	required	Code that designates the active record in XRP-II's system parameter table
site	string	6	optional; zoom to select from a list of sites	Mnemonic or short name for an ECS site
last control item id	string	20	optional	Unique code for a version- or configuration-controlled item

(2) **Transaction Log Screen:** This screen allows operators to browse, and maintain if necessary, the database transaction log file. All fields in this file as described in Table 9.9-20, are system supplied. When a transaction is processed, the system provides the next available transaction number and records sufficient information to determine when any field was modified and by whom.

The log is an essential element in coordinating database changes among sites. The Export DAAC-Unique Changes utility analyzes the transactions to determine the database records that have been modified but not yet exported for transfer. It also sets the Transferred flag so the transaction can be bypassed next time the utility runs.

Table 9.9-20. Transaction Log Field Description

Field Name	Data Type	Size	Entry	Description
transaction key	numeric	5	system supplied	Number that uniquely identifies each update transaction
field number	numeric	8	system supplied	Identifier for the XRP-II field affected by the transaction
table name	string	10	system supplied	Name of the XRP-II table affected by the update transaction
operator id	string	8	system supplied	Userid of the operator making the update transaction
date	date	N/A	system supplied	Date of the update transaction
time	time	N/A	system supplied	Time of the update transaction
transaction type	string	1	system supplied	Code for the type of transaction: A (add), M (modify), or D (delete)
transferred	string	1	system supplied	Code that indicates that the transaction has been used to export and transfer database records that have changed.
record location	numeric	8	system supplied	Identifier for the transaction record's location within XRP-II

(3) **Transaction Archive Screen:** Control item data update transactions should periodically be deleted from the database after changed records have been exported. This makes room to log new transactions.

This screen copies to a named file the records of transactions that occurred on or prior to a specified cutoff date. It then deletes the records from the database.

Specify the date of the last transaction to archive and the name of a file in which to store the data as described in Table 9.9-21.

Table 9.9-21. XRP II GUI/CHUI Field Description

Field Name	Data Type	Size	Entry	Description
file name	string	8	required	Name of the file in which to store transaction records being archived
cutoff date	date	N/A	required	Date of the most recent transaction to be archived

(4) **Serial Numbers Maintenance Screen:** This screen maintains a list of ECS-issued serial numbers indexed to control items, providing a rudimentary inventory tracking mechanism. The screen can be accessed via the System Utilities menu for the convenience of the XRP-II administrator. It also appears on the Utilities menu and is described in detail in item (6) above.

(5) **Edit Groups File Screen:** The Edit Groups File screen provides access to a means for re-configuring Baseline Manager to recognize new groups of XRP-II menus and screens.

Groups enable system administrators to grant access to menus and screens collectively rather than individually. Groups are identified and defined in a "groups" configuration file. The file specifies what menus and screens an operator is allowed to access. It does not define the menu from which a submenu or data entry screen can be accessed.

The Edit Groups File screen displays a descriptive message and prompts for confirmation to proceed before executing the "vi" program against the "groups" configuration file. Edit the file, then exit "vi". The screen automatically recompiles group configuration binaries, making the new group definitions available to XRP-II's menu handler program. Note: This utility should not be run when other operators are logged on the system.

Section D.5 of the System Reference Manual describes the "groups" file in detail.

(6) Edit Users File Screen: The Edit Users File screen provides access to XRP-II's list authorized Baseline Manager operators, the "users" file. The list contains userids and specifies which group of menus and screens an operator can use. It can also specify an entry menu for each user as well as any deviations for the user from default permissions associated with each data entry screen.

The screen displays a descriptive message and prompts for confirmation to proceed before executing the "vi" program against the file. Edit the file, then exit "vi". XRP-II automatically recompiles user configuration binaries, making the new user definitions available to its menu handler program. Note: This utility should not be run when other operators are logged on the system. After it runs, you will receive a warning message then get logged out of XRP-II at your next keystroke. Simply log back in.

Section D.5 of the System Reference Manual describes the "users" file in detail.

(7) Edit Printers File Screen: The Edit Printer File screen provides access to XRP-II's list that defines printer selections and options available for generating both custom and ad hoc reports. The screen displays a descriptive message and prompts for confirmation to proceed before executing the "vi" program against the "msprinters" configuration file. After editing the file, operators exit "vi". The screen automatically recompiles printer configuration binaries, making the new printer definitions available to all operators.

Section 8.1.4 of the System Reference Manual describes the printers file's format and use.

(8) Site Master Maintenance Screen: The Site Master Maintenance screen lets operators maintain an index of ECS-related sites. This data is used solely to generate a site pick list to facilitate entering control item deployment data and report generation parameters on other screens. This screen is identical to the Branch Master Maintenance screen described in the section 6.8 of the System Reference Manual, except that two "site" labels are used in lieu of "branch" labels and the tax code field has been deleted. Baseline Manager does not implement the organizational branch processing described in the System Reference Manual, so site entries are not required for any XRP-II processing. Table 9.9-22 describes the "site" fields that were tailored for ECS.

Table 9.9-22. Site MasterMaintenance Field Description

Field Name	Data Type	Size	Entry	Description
site	string	6	required	Mnemonic or short name for an ECS site.
site name	string	46	optional	Full name of an ECS site.
phone number	string	8	optional	Phone number of a point of contact at the site
street address	string	30	optional	Address for the site
city	string	20	optional	Name of the city in which the site is located
state	string	2	optional	Abbreviation for the state in which the site is located
zip	string		optional	Postal code for the site
country	string		optional	Code for the country in which the site is located

(9) **Machine Network Maintenance Screen:** Operators use the Machine Network Maintenance screen to keep a list of MSS CM Server hosts to which Baseline Manager records may be shipped. Entries are accessible to the two System Utilities screens used when exporting site-unique records for the SMC and when exporting release records from the SMC to the sites, respectively. The list is not required; it exists solely as an aid for correctly specifying the target host for an ftp session.

Table 9.9-23. Machine Network Maintenance Field Description

Field Name	Data Type	Size	Entry	Description
machine name	string	32	required; must be	Full, network-addressable name of a host
site	string	6	optional; zoom to select from a list of sites	Mnemonic or short name for an ECS site
description	string	30	optional	Textual characterization of a host machine

(10) **Commodity Code Maintenance:** The Commodity Code Maintenance screen maintains standard codes and names that are used to classify a control item according to how it was produced or obtained for the project. This data, used as a pick list by control item update screens, promotes data standardization and facilitates recording control item data.

Use this screen to update the list of commodity codes to use for control items. Table 9.9-24 describes its data entry fields.

Table 9.9-24. Commodity Code Maintenance Field Description

Field Name	Data Type	Size	Entry	Description
commodity code	string	8	required	Classification for how a control item was produced or obtained (e.g., COTS, heritage, GFE, custom, mod-COTS, shareware, freeware, etc.)
description	string	20	optional	Full name for the commodity type

(11) Import Data Screen: The screen initiates loading of XRP-II data from tar files created by Baseline Manager's data export utilities. Selecting "Y" at the prompt causes XRP-II to process all files in the directory named in the IMPORTPATH environment variable. Tar files -- whose names identify the destination host, originator's site, and date and time created -- are processed in alphabetical order.. Upon completion, the original files are moved to an archive directory named in the IMPORTARC environment variable.

(12) Export Release Records Screen: Operators use this screen to prepare for distribution those records associated with the release of a specified baseline, configuration item, or other control item. They can choose to have XRP-II ftp the files to up to six remote hosts via a SEND NOW feature.

The screen with fields as defines in Table 9.9-25 extracts data about a specified item and all other items in its bill of materials as of a given date. XRP-II copies appropriate records from control item master, product structure, engineering change, interdependency, and deployment tables and stores them in the directory named in the EXPORTPATH environment variable in a format compatible with the Import Data utility. One tar file is created for each destination and given a name that identifies the destination machine name and site, the control item's identifier, and the effective date used for the bill. If the SEND NOW feature is used, XRP-II ftp's the files then moves them to the archive directory named in the EXPORTARC environment variable. If not, the files remain in the export directory to be transferred manually or until a subsequent export operation employs the SEND NOW feature.

Enter the identifier of one or more control items, then specify one or more hosts to receive the data. (Include domain names, or use IP addresses. Machine names can be selected from a managed list by using the "zoom" command.) Next, enter an effective date to use for the item's bill, and choose whether or not to ftp the files automatically. Begin data extraction by using the "execute" command. XRP-II will prompt for a login account and a password if it's been instructed to perform the ftp.

Table 9.9-25. Export Release Record Field Description

Field Name	Data Type	Size	Entry	Description
control item id	string	20	required; zoom to select from a control item list	Unique code for a version- or configuration-controlled item
machine	string	40	required; zoom to select from a list of machines	Full, network-addressable name of a host
date of release	date	N/A	required	"As of" date used in selecting records from the configuration history of a control item
send now	string	1	Y or N	Code that indicates whether or not to ship files immediately

(13) Export Site-Unique Changes Screen: This screen is used to extract copies of site-unique records that are to be transferred to another site such as the SMC. Extracted data is stored in a collection of files compatible with XRP-II's Import Data utility and converted to tar format. The tar file is then ftp'ed to specified hosts if the operator so chooses.

XRP-II analyzes the transaction log to determine which records have changed since the last time the utility was run and exports those associated with control items having a designated site prefix. (XRP-II expects that the site prefix to be the first three characters of the code for the site specified in the system parameters table (see item (1) above). Tar files are stored in the directory named in the EXPORTPATH environment variable. Once they have been ftp'ed, the files are moved to an archive directory named in the EXPORTARC environment variable.

Enter the name of one or more machines in the format described in Table 9.9-26 to receive the data (using their domain names or IP addresses), and choose whether or not to ftp the data files immediately after they are created. Names can be selected from a managed list by using XRP-II's "zoom" command. Use the "execute" command to begin data extraction. If the file is to be transferred too, supply the name of a login account and a password for the ftp when XRP-II prompts.

Table 9.9-26. Export Site-Unique Changes Field Description

Field Name	Data Type	Size	Entry	Description
name	string	40	required; zoom to select from a list of machines	Full domain name or network address of the host to receive the exported data file(s)

(14) Export SMC Data Screen: This screen extracts, and distributes to remote sites, copies of records about centrally-managed control items that have changed since the last time this function was used. (XRP-II expects centrally-managed control items to have identifiers that begin with a numeral.) XRP-II can ftp the files to up to ten remote hosts specified by the operator.

XRP-II analyzes the transaction log to determine what data has changed and which control items were affected. Records that reference qualifying control items are then copied from control item master, product structure, and engineering change tables into files compatible with XRP-II's Import Data utility (see section xxx). XRP-II uses ftp to transfer the files that are stored in a local directory (named in the EXPORTPATH environment variable) to a directory (named in the IMPORTPATH environment variable) on the remote hosts. Once they have been transferred, the files are moved from the export directory to an archive directory named in the EXPORTARC environment variable.

Enter the name of one or more hosts in the format described in Table 9.9-27 to receive the data (using either domain names or IP addresses), and choose whether or not to ftp the data files immediately after they are created. Names can be selected from a list of servers (see section xxx) by using the "zoom" command. Use the "execute" command to begin data extraction. If the file is to be transferred too, supply the name of a login account and a password for the ftp when XRP-II prompts.

Table 9.9-27. Export SMC Data Field Description

Field Name	Data Type	Size	Entry	Description
name	string	40	required; zoom to select from a list of machines	Full domain name or network address of the host to receive the exported data file(s)

9.9.3 Setting Tunable Parameters

XRP-II can be configured to limit access to authorized operators and regulate the operations each can perform. A system administrator at each site maintains group lists of related screens that operators can be assigned to access for performing work tasks. They also maintain a list of operators authorized to use XRP-II, together with their group assignments and any special access privileges.

XRP-II can also be configured to define system printers. Again, a system administrator at each site maintains a list of approved printer destinations for XRP-II reports. While specific printers can be named, the list often just offers operators choices of orientation and pitch for output that will be printed to their default printers.

Special programs, accessible via the System Utilities menu, are available to help maintain the files that contain these lists. They require that UNIX vi be used to edit the files. Section 5 of the

System Reference Manual describes formats and use of the groups and users files; section 8.1.4 describes the printer file.

Table 9.9-28. Tunable Parameters

Operator Task	Affected Data/ Parameter	Range of Values	Source	Tool/ Change Control	Description
Modify "groups" list	menu and data entry screen sets	alpha-numeric strings	File $MSPATH/xrp/mms/menudata/groups	XRP-II and UNIX vi	defines named groups of menus/screens to which operators can be given access; can also specify inquire, add, modify, and delete privileges by screen in each group
Modify "users" list	operator authorizations	userids; group names; operator permissions (I, A, M, D)	File $MSPATH/xrp/mms/menudata/users	XRP-II and UNIX vi	Identifies users authorized to run XRP-II, assigning to each a group of menus/ screens that can be accessed; can also specify for each user an entry menu and changes to the list of screens and associated privileges the group has
Modify "printers" list	valid report destinations	printer names (may have up to 12 alpha-numeric chars); printer programs (limited to 20 characters)	File $MSPATH/xrp/lib/msprinters	XRP-II and UNIX vi	Identifies names by which system printers are known to XRP-II, and specifies printer program arguments for each

9.9.4 Special Constraints

The ECS environment imposes the following constraints on how XRP-II is used :

- ↑ Control item identifiers - XRP-II uses centralized database technology and is separately installed at each ECS site. This necessitates a special scheme for assigning identifiers to control items so that sites may safely exchange database records. For example, the SMC must be able to distribute centrally-maintained release records to multiple sites without interfering with records the sites locally maintain there. Similarly, the SMC must be able to absorb copies of site-maintained records forwarded by the sites to form the consolidated picture of system-wide baselines without contaminating centrally-maintained data. To distinguish between centrally-maintained and site-maintained records, Baseline Manager expects that identifiers of site-maintained control items have a 3-character prefix that matches the first three characters of the site's code (see Section 9.9.2.12, item (1)). To preclude confusion, identifiers of centrally-maintained control items should begin with a numeral.

- ↑ Database schemas - The XRP-II database schema must be identical at all ECS sites so that database records can be uniformly exchanged among them.

- ↑ Data entry screens - Table view driver programs cannot handle the number and size of fields used in the form view of many Baseline Manager data entry screens. Where limitations exist, fields that appear in table view were chosen either because they help identify control items or because they are likely to be used in multi-record operations.

9.9.5 Command Line Interface

Operators run XRP-II primarily via its menus and data entry screens. However, several standard utility programs can be run from the command line. These are documented in numerous places in the XRP-II System Reference and Product Information manuals.

An additional interface, configured specially for ECS, generates resource configuration files needed by resource planners using the Planning Subsystem. Usually employed on demand via Tivoli, operators can invoke it via the command line and view output in file "/usr/tmp/resplan" on the XRP-II host.

Table 9.9.29. Command Line Interfaces

Command Line Interface	Description and Format	When and Why Used
resplan	resplan mm/dd/yy (rolename), where: • mm/dd/yy is a valid date • rolename is a code for signalling Tivoli which sentry-log and administrator is to receive status messages resulting from the operation.	Used to generate a resource configuration file, primarily for resource planners. This operation is normally invoked by Tivoli

9.9.6 Outputs

XRP-II is a database application whose principal outputs as Baseline Manager are formatted data files. These files are used to exchange records among XRP-II systems, to describe production baseline configurations for resource planners, and to store copies of reports. Other, ancillary files are occasionally produced in the course of processing, but they are for XRP-II's internal use and generally of no interest to operators. Table 9.9-30 lists and describes each of the above types of outputs.

Table 9.9-30. Outputs

Output	Description and Format
export/import tar files	Produced by XRP-II's "export" utilities, these are archives containing one or more .dasc files. (See sections 4.3.3.1.5 and 4.3.3.1.7 above.)
.dasc files	Transient, ASCII files produced by XRP-II's "export" utilities and used by its import utility. Each file contains a header record followed by one or more detail records from a particular XRP-II database table. Each detail record contains values for a single database record, separated by pipe symbols. The header record contains the specification of the dump, identifying the database table and the names of the fields that correspond to values in the detail records. (See section 1.6 of the Database Tools and Techniques Manual.)
.dspc files	Transient, ASCII files produced by XRP-II's import and export utilities. Each file identifies the name of a database file (table) and the names of fields in that file in the order required to load data in the database. Data is separated by pipe symbols. (See section 1.6 of the Database Tools and Techniques Manual.)
resplan data file	An ASCII file consisting of one header record followed by one or more detail records. Each detail record identifies a control item that is marked as a planning resource and is part of a specified production baseline; it also identifies specific characteristics about the item. The header record contains a message that identifies the production baseline that was specified. This file is stored in directory /usr/tmp on the MSS CM Server, and a copy of it is transmitted, via Tivoli, to the requestor. (See section xxx above.)
print files	ASCII text files that contain reports requested via XRP-II data entry and report generation screens. These files are stored in the operator's home directory.
temporary files	Various working files XRP-II creates for its own internal use. These are stored in directory /usr/tmp on the MSS CM Server but not deleted by XRP-II.

9.9.7 Event and Error Messages

XRP-II issues both status and error messages to those using it. Unify manuals discuss common messages an operator may encounter, but no listing of standard XRP-II messages is provided in the COTS documentation. Messages are generally self-explanatory; however, some refer operators to log files which, in most cases, are intended for XRP-II programmers and require special training to interpret.

In addition, XRP-II records every database update event in a Transaction Log. This log, stored in the database, can be browsed via the Transaction Log screen (see Section 9.9.2.12, item (2)) to determine the date, time, operator, and type of changes made to XRP-II database tables. The transaction log does not contain messages, per se, and is not monitored or used by ECS' system management applications.

Many errors XRP-II reports result from an error returned by the Unify RDBMS. Details about fatal errors are written to the following error log files located within the XRP-II principal directory:

- /bin/errlog
- /def/errlog
- /dicty/errlog
- /bin/dbrbld.err
- /bin/uniload.err
- /dicty/uniload.err

These files generally do not contain the actual messages displayed to the operator, and they are meaningful mainly to the system administrator or XRP-II programmer. Appendix E of the Unify Direct HLI Programmer's reference manual describes some of the common messages written to the logs.

9.9.8 Reports

XRP-II can generate the pre-defined reports listed in Table 9.9-31. Each can be routed to the operator's display, a named file, or a printer.

XRP-II may make several printers available for a particular report. These printers often represent formatting choices rather than specific devices. Reports directed to a printer that does not correspond to a specific device will be printed on the operator's default printer device.

All of the pre-defined reports are generated according to specifications that are "compiled" using XRP-II's Datarite report writer. Authorized operators can develop custom specifications. See Section 8 of the Datalook/Datarite Reference Manual for details.

Table 9.9-31. Reports

Report Type	Report Description	When and Why Used
Bill of Materials	A list that identifies and describes an assembly's constituent control items	Ad Hoc for Resource Mgt, Maintenance & Admin
Indented Bill of Materials	A list that identifies and depicts the full assembly structure of a control item	Monthly/Ad Hoc for Resource Mgt, Maintenance & Admin
Summarized Bill Report	A list that identifies the control items in an assembly along with the quantity of each	Ad Hoc for Resource Mgt, Maintenance & Admin
Multilevel Where Used Display	A list that identifies and describes the assemblies in which a specified control item is used	Ad Hoc for Maintenance and Admin
Multilevel Where Used Report	A list that identifies and describes the assemblies in which a specified control item is used together with the effectivity dates for each	Monthly/Ad Hoc for Resource Mgt, Maintenance & Admin
Configuration Items List - One	A list of all ECS configuration items active and deployed at the specified sites as of a specified date	Ad Hoc for verification of deployed capabilities
Configuration Items List - Two	A list of all ECS design components active and deployed at the specified sites as of a specified date	Monthly/Ad Hoc for Resource Mgt, Maintenance & Admin
Configured Articles List	A list of all active ECS configured articles for specified sites as of a specified date	Monthly for inventory, audit, and accounting
Version Description Reports	A list of all active versions of ECS configured articles and associated versions of CIs for specified sites as of a specified date	Monthly/Ad Hoc for Release Mgt (VDD), Resource Mgt, Maintenance & Admin
Site Baseline	A list of all ECS configured articles active and deployed as part of a specified baseline	Monthly/Ad Hoc for Release Mgt (VDD), Resource Mgt, Maintenance & Admin
Change History	A list of all versions and product structure revisions for the specified item together with details associated with the item change	As required to support CCB activities
BOM Comparison Reports	A list that identifies the control items that two bills have in common as well as the control items in one that are not in the other.	As required to support CCB activities

9.9.8.1 Sample Reports

The figures that follow contain samples of Baseline Manager's pre-defined reports. One sample is provided for each report listed in Table 9.9-31.

```
(pibomr0)                                                          DATE:  08/19/96   TIME:  20:35
ECS Development Fac                                                                     PAGE:  1
                                          BILL OF MATERIALS
Control Item IDs: b0101400                                                  Number of levels: 99
Explosion quantity: 99                                                     Date of bill: 08/19/96

-------------------------------------------------------------------------------------------------
Control Item ID:  b0101400    Project: ECS
          TRMM Release of MLCI              uom: EA   Engineer: 34A678
Current Revision: 0  Versjon:  B0

                                                                                   QUANTITY
LVL   CONTROL ITEM ID       DESCRIPTION                                                 PER   CUOM
===   ====================  ====================================================  ============   ====
  1   b0101410              TRMM Release of SCM                                       1.0000   EA
  2   b0101412              Clearcase Client for Sun Solaris 2.4                      1.0000   EA
  2   b0101414              ClearCase Server for Sun Solari 2.4                       1.0000   EA
  2   b0101416              ClearCase Scripts for TRMM                                1.0000   EA
  1   b0101420              TRMM Release of CRM                                       1.0000   EA
  1   b0101430              TRMM Release of BM                                        1.0000   EA
There are 6 components in this bill.
-----------------------------------------------------------------------------------------------------
```

Figure 9.9-7. Bill of Materials Report

```
(pibomr1)                                                                                                                    DATE:  08/19/96   TIME:  20:35
ECS Development Fac                                                INDENTED BILL OF MATERIALS                                                          PAGE:  1
Control Item IDs: b0101400                                                                                                                          Number of
levels: 99
Explosion quantity: 99                                                                                                                   Date of bill: 08/19/96

-----------------------------------------------------------------------------------------------------------------------------------------------------------------
Assembly: b0101400              Project: ECS   uom: EA
          TRMM Release of MLCI                                   Low level code:  2   Responsible Engineer: 34A678
Active date: 05/20/96  Inactive date: **/**/**

                                                                 IMPLEMENTATION        ITEM CLASS /                                      QUANTITY   ACTIVE
INACTIVE
LEVEL          CONTROL ITEM ID       NAME                        STATUS                SUBCLASS          MODEL/VERSION                        PER   DATE
DATE
============   ====================  ========================    ====================  ================  ========================  ============   ========
========
1              b0101410              Software Change Mgr         production            design            A                                1.0000   05/20/96
**/**/**
                TRMM Release of SCM
.2             b0101412              ClearCase Client            production            software          2.1                              1.0000   05/23/96
**/**/**
                Clearcase Client for Sun Solaris 2.4
.2             b0101414              ClearCase Server            production            software          2.0.2                            1.0000   05/23/96
**/**/**
                ClearCase Server for Sun Solaris 2.4
.2             b0101416              ClearCase Scripts           production            software          A                                1.0000   05/23/96
**/**/**
                ClearCase Scripts for TRMM
1              b0101420              Change Request Mgr          production            design            A                                1.0000   05/20/96
**/**/**
                TRMM Release of CRM
1              b0101430              Baseline Manager            production            design            A                                1.0000   05/20/96
**/**/**
                TRMM Release of BM
There are 6 components in this bill.
```

Figure 9.9-8. Indented Bill of Materials

```
(pisumr)                                                                                    DATE:  10/04/96   TIME:  09:35
ECS Development Fac                                                                                              PAGE:  1
                                                SUMMARIZED BILL REPORT

ASSEMBLY              DESCRIPTION                                          NAME                      CLASS
====================  ====================================================  ========================  ================
b0101400              TRMM Release of MLCI                                 Management Logistics      design

COMPONENT             DESCRIPTION                                          NAME                      CLASS
====================  ====================================================  ========================  ================
b0101410              TRMM Release of SCM                                  Software Change Mgr       design
b0101420              TRMM Release of CRM                                  Change Request Mgr        design
b0101430              TRMM Release of BM                                   Baseline Manager          design
b0101412              Clearcase Client for Sun Solaris 2.4                 ClearCase Client          software
b0101414              ClearCase Server for Sun Solaris 2.4                 ClearCase Server          software
b0101416              ClearCase Scripts for TRMM                           ClearCase Scripts         software
There are 6 parts in this summarized bill.
-------------------------------------------------------------------------------------------------------------------------------
```

Figure 9.9-9. Summarized Bill Report

```
(piwur1)                                          DATE:  10/04/96   TIME:  09:43
ECS Development Fac                                                     PAGE:  1
Components:  b0101412                                     Number of levels: 99
                        MULTI-LEVEL WHERE-USED DISPLAY
-------------------------------------------------------------------------------
Component: b0101412        Clearcase Client for Sun Solaris 2.5.1
   Project:  ECS

                                    CMDTY         QUANTITY
LEVEL         CONTROL ITEM ID       CODE               PER   CUOM
===========   ====================  ========  ============   ====
1             b0101410              Other           1.0000   EA
              TRMM Release of SCM
.2            b0101400              Other           1.0000   EA
              TRMM Release of MLCI
..3           b0100001              Other           1.0000   EA
              TRMM Release of MSS
...4          b0000001                              1.0000   EA
              EOSDIS Core System for TRMM
1             b01200                Other           1.0000   EA
              SPRE4SUN Software
.2            b01000                Other           1.0000   EA
              QUEUING Workstation
..3           b010                  Other           1.0000   EA
              TRMM with Extension for ...
..3           b011                  Other           1.0000   EA
              TRMM with Extension for ...
..3           b0100                 Other           1.0000   EA
              Science Processing String
...4          b010                  Other           1.0000   EA
              TRMM with Extension for ...
There are 16 assemblies using this component.
-------------------------------------------------------------------------------
```

Figure 9.9-10. Multilevel Where-Used Display

```
(piwur)                                                                                    DATE:  10/04/96   TIME:  09:43
ECS Development Fac                        MULTI-LEVEL WHERE-USED REPORTS                                      PAGE:  1
Components: b0101412                                                                          Number of levels: 99

-----------------------------------------------------------------------------------------------------------------------
Component:  b0101412          Project:     uom:  EA
            Clearcase Client for Sun Solaris 2.4                     Low level code:   4    Planner:  34A678
Active date:  05/23/96   Inactive date:  **/**/**

                                                                                   EXTENDED                   INACTIVE
LEVEL         CONTROL ITEM ID                                CMDTY CODE  CUOM   QUANTITY      ACTIVE DATE  DATE
===========   ===================                            ========    =====  ===========   ===========  =========
1             b0101410                                       Other       EA             1.0   05/23/96     **/**/**
              TRMM Release of SCM
.2            b0101400                                       Other       EA             1.0   05/20/96     **/**/**
              TRMM Release of MLCI
..3           b0100001                                       Other       EA             1.0   05/20/96     **/**/**
              TRMM Release of MSS
...4          b0000001                                                   EA             1.0   05/05/94     **/**/**
              EOSDIS Core System for TRMM
1             b01200                                         Other       EA             1.0   05/23/96     **/**/**
              SPRE4SUN Software
.2            b01000                                         Other       EA             1.0   05/23/96     **/**/**
              QUEUING Workstation
..3           b010                                           Other       EA             1.0   05/01/96     **/**/**
              TRMM with Extension for ...
..3           b011                                           Other       EA             1.0   05/23/96     **/**/**
              TRMM with Extension for ...
..3           b0100                                          Other       EA             1.0   05/25/96     **/**/**
              Science Processing String
...4          b010                                           Other       EA             1.0   05/01/96     **/**/**
              TRMM with Extension for ...
There are 16 assemblies affected by this component.
-----------------------------------------------------------------------------------------------------------------------
```

Figure 9.9-11. Multilevel Where-Used Report

```
(cil1)                                                                                                  DATE:  08/19/96   TIME:  19:26
ECS Development Fac                      ECS CONFIGURATION ITEMS LIST - LEVEL ONE                                          PAGE:  1
SITE:  GSFC                                                                                                Date of bill: 05/30/96

-----------------------------------------------------------------------------------------------------------------------------------
Subsystem:   MSS             Management Subsystem

CI MNEMONIC CI NAME                    CI CONTROL ITEM ID    CI DESCRIPTION               CI MODEL/VERSION        CI SCOPE
===========  =======================  ====================  ==========================   =====================  ========
MACI         Management Agent          b0101100              Management Agent CSCI        A.IT.01                 core
MCI          Management Software       b0101200              Management Software CSCI     A.IT.01                 core
MHCI         Management Hardware       b0101300              Management Hardware CI       A.IT.01                 core
MLCI         Management Logistics      b0101400              Management Logistics CSCI    A.IT.01                 core
-----------------------------------------------------------------------------------------------------------------------------------
Subsystem:  CSS              Communication Subsystem

CI MNEMONIC CI NAME                    CI CONTROL ITEM ID    CI DESCRIPTION               CI MODEL/VERSION        CI SCOPE
===========  =======================  ====================  ==========================   =====================  ========
DCCI         Distr Computing Software  b0102100               Distributed Computing Sof   A.IT.01                 core
DCHCI        Distr Computing Hardware  b0102200               Distributed Computing Har   A.IT.01                 core
-----------------------------------------------------------------------------------------------------------------------------------
```

Figure 9.9-12. Configuration Items List - One

```
(cil2)                                                                    DATE:  08/19/96   TIME:  19:48
ECS Development Facility            ECS CONFIGURATION ITEMS LIST - LEVEL TWO                          PAGE:  1
Site or range:  EDF                                                                 Date of bill: 05/30/96


Subsystem:  MSS     :  Management Subsystem      Version:  A.IT.01

  CI Item:  MHCI    :  Management Hardware       Version:  A.IT.01

COMPONENT                     COMPONENT
MNEMONIC  COMPONENT NAME      CONTROL ITEM ID       COMONENT DESCRIPTION             COMPONENT MODEL/VERSION  COMP SCOPE
========= ================== ===================== ================================ ======================== ===========
          Enterprise Mgmt Svr b0101310              TRMM Release of EMC Server       A.IT.01                  core
          Local Sys Mgmt Svr  b0101320              TRMM Release of LSM Server       A.IT.01                  core
          Enterprise Mgnt Wks b0101330              TRMM Release of EMC Wkstation    A.IT.01                  core
          Local Sys Mgmt Wks  b0101340              TRMM Release of LSM Wkstation    A.IT.01                  core
          Management Printer  b0101350              TRMM Release of MSS Printer      A.IT.01                  core


  CI Item:  MLCI        Management Logistics CSCI     Version:  A.IT.01
```

Figure 9.9-13. Configuration Items List - Two

```
(cal7)                                                                                                  DATE:  08/19/96   TIME:  20:12
ECS Development Fac                                ECS CONFIGURED ARTICLES LIST                                                PAGE:  1
Site(s): EDF                                                                                        Date of Configuration: 05/30/96

-------------------------------------------------------------------------------------------------------------------------------------

  CONFIGURATION ITEM:  Management Logistics CSCI      CONTROL ITEM ID:  b0101400

CONFIGURED ARTICLE NAME   MOD/VER                   CONTROL ITEM ID       IMPL STATUS           ITEM SUBCLASS     CMDTY CD  SCOPE
========================  ========================  ====================  ====================  ================  ========  ======
ClearCase Client          2.1                       b0101412              production            appl-client       COTS      core
ClearCase Server          2.0.2                     b0101414              production            appl-server       COTS      core
ClearCase Scripts         A.IT.01                   b0101416              production            appl-scripts      custom    core


-------------------------------------------------------------------------------------------------------------------------------------
  CONFIGURATION ITEM:  Management Sftware CSCI   CONTROL ITEM ID:  b010120
```

Figure 9.9-14. Configured Articles List

```
(vdd)                                                                                                                    DATE:  09/05/96   TIME:  08:47
ECS Development Fac                                          VERSION DESCRIPTION DOCUMENT                                                          PAGE:  1
Control Item ID:  b0000001                                                                                          Date of Configuration:  06/03/96
Site:  EDF

SITE:  EDF     :  ECS Development Fac


         Subsystem:   MSS       :  Management Subsystem  Control Item ID:  b0100001

Configuration Item:   MLCI      :  Management Logistics  Control Item ID:  b0101400

         Component:   SCM       :  Software Change Mgr   Control Item ID:  b0101410
                                                                                                                  ACTIVE    INACTIVE  CMDTY
CONTROL ITEM NAME         MODEL/VERSION             CONTROL ITEM ID       MFR/DEV  CLASS             SUB CLASS            NUM LIC  DATE      DATE      CODE      SCOPE
========================  ========================  ===============       =======  ================  =================    =======  ========  ========  ========  ======
ClearCase Scripts         A.IT.01                   b0101616              ECS      software          appl-scripts           99999  05/23/96  **/**/**  custom    Core
ClearCase Server          2.0.2                     b0101414              ATR      software          appl-server               75  05/23/96  **/**/**  COTS      Core
ClearCase Client          2.1                       b0101412              ATR      software          appl-client               75  05/23/96  **/**/**  COTS      Core
ClearCase DDTS Integrati  2                         b0101418              ATR      software          appl-scripts              75  05/23/96  **/**/**  COTS      Core

         Component:   CRM       :  Change Request Mgr    Control Item ID:  0101420
                                                                                                                  ACTIVE    INACTIVE  CMDTY
CONTROL ITEM NAME         MODEL/VERSION             CONTROL ITEM ID       MFR/DEV  CLASS             SUB CLASS            NUM LIC  DATE      DATE      CODE      SCOPE
========================  ========================  ===============       =======  ================  =================    =======  ========  ========  ========  ======
Dist Defect Track'g Sys   A.IT.01                   b0101422              PUR      software          application             250  05/23/96  **/**/**  COTS      Core

         Component:   BLM        :  Basseline Manager     Control Item ID:  0101430
                                                                                                                  ACTIVE    INACTIVE  CMDTY
CONTROL ITEM NAME         MODEL/VERSION             CONTROL ITEM ID       MFR/DEV  CLASS             SUB CLASS            NUM LIC  DATE      DATE      CODE      SCOPE
========================  ========================  ===============       =======  ================  =================    =======  ========  ========  ========  ======
XRP-II                    3.0                       b0101432              HTG      software          application              50  05/23/96  **/**/**  COTS      Core
UNIFY DBMS                5.0.7.2.0                 b0101434              UNI      software          application              50  05/23/96  **/**/**  COTS      Core
```

Figure 9.9-15. Version Description Report

```
(sitebase)                                                                                    DATE:  08/19/96   TIME:  20:22
ECS Development Fac                                SITE BASELINE REPORT                                                PAGE:  1
All Control Items                                                                                   Number of levels: 99
                                                                                                 Date of configuration: 05/30/96

-----------------------------------------------------------------------------------------------------------------------------
Baseline: bLAR010
          TRMM Baseline with LaRC extenstions for …

                                                                                                          ACTIVE   INACTIVE
LEVEL         CONTROL ITEM ID      DESCRIPTION                      MFR/DEV MODEL/VER                      DATE     DATE
===========  ==================== ==============================  ======= ======================== ======== ========
1             bLAR01000            Queuing Server                           A…                            05/01/96 **/**/**
.2            bLAR73005            Queuing Server boot partition   ECS      A…                            05/23/96 **/**/**
.2            bLAR73006            Queuing Server user partition   ECS      A…                            05/23/96 **/**/**
.2            bLAR1100             sprl7sun Hardware                        A…                            05/23/96 **/**/**
..3           b01101               SUN SPARCSTATION SX 20/71       SUN      SPARCSTATION SX 20/71         05/23/96 **/**/**
..3           b01102               20 INCH COLOR MONITOR           SUN                                    05/23/96 **/**/**
..3           b01103               EXPANDED 101 KEYBOARD           SUN                                    05/23/96 **/**/**
..3           b01104               3 BUTTON TRACKBALL MOUSE        SUN                                    05/23/96 **/**/**
..3           b01105               64 MB EXP MEM MODULE            SUN                                    05/23/96 **/**/**
..3           b01107               IBM 1.05GB INTERNAL DISK DRIVE  IBM                                    05/23/96 **/**/**
..3           b01109               SBUS FAST SCSI-2 ETHERNET CARD  SUN                                    05/23/96 **/**/**
..3           b01112               INTERNAL CD ROM                 SUN      Ultra 2                       05/23/96 **/**/**
.2            b01200               sprl7sun Software               ECS      A…                            05/23/96 **/**/**
..3           b0101414             ClearCase Server for Sun Solar  ATR      2.0.2                         05/23/96 **/**/**
..3           b0101416             ClearCase Scripts for TRMM      ATR      A                             05/23/96 **/**/**
..3           b01201               SOLARIS OS                      SUN      Solaris 2.4                   05/23/96 **/**/**
```

Figure 9.9-16. Site Baseline Report

```
(chghist)                                                    DATE:  10/28/96   TIME:  15:04
ECS Development Fac                  CHANGE HISTORY REPORT                            PAGE:  1


Control Item ID:  b0101401              :  Management Logistics      Project:  ECS
Management Logistics CSCI
Mfr/Dev:  ECS     Model/Version:  A.IT.01                          Latest Inactive Date:  **/**/**
Current Revision:    1                                     Earliest Active Date:  04/17/96

-----------------------------------------------------------------------------------------------
                                                  TROUBLE         ACTIVE   INACTIVE APPROVAL
REVISION ENG CHANGE CCR #                         TICKET          DATE     DATE     DATE
======== ========== ============================ =============== ======== ======== ========
0        96001      ESDIS001                                     12/13/96 04/22/97 10/23/96
1        GSF96001   GSFed02826                                    04/23/97 **/**/** 04/07/97
-----------------------------------------------------------------------------------------------
```

Figure 9.9-17. Change History Report

```
(bomcmp)                                                
```

9.9.8.2 Report Customization

All of the pre-defined reports discussed in Section 9.9.8.1 are generated according to specifications that are "compiled" using XRP-II's Datarite report writer. Authorized operators can modify these specifications or develop their own. See Section 8 of the Datalook/Datarite Reference Manual for details.

This page intentionally left blank.

10. Metadata Administration

Every science data product generated and archived by the ECS must be described to the system by metadata which are put into an inventory and then used to retrieve and distribute the data to users of the system. The ECS Earth Science Data Model, described in documents 420-TP-015-002 and 420-TP-016-002, organizes the metadata into groups of related attributes and services to be performed on the data products. These "core" attributes are necessary to identify, interpret and perform services on granules and collections. The Data Model also provides for "product-specific" attributes, i.e. attributes which are unique to a specific data product.

The smallest aggregation of data that is independently described and inventoried in ECS is referred to as a data granule. Granules are organized into logical groupings called collections in which the granule metadata varies principally by time or location, called single-type collections. In Release 2 only single-type collections are supported (more complex organizations will be supported in later ECS releases).

Every collection is described by an Earth Science Data Type (ESDT) and is made known to the system by an ESDT descriptor file and associated software code which is built into the Data Server's dynamic link library (DLL) to perform the services.

Metadata administration includes creating and updating ESDTs within ECS. Collections may be modified and updated over time. In addition, quality assurance will be performed after ESDTs have been installed and granules have been generated and stored in the archives. Collection-level metadata can only be updated by installing a revised ESDT. In Release 2, the only granule-level metadata which can be updated manually (i.e. not as a result of an operation such as Subsetting which modifies the science data content of a granule) are the Quality Assurance flags and explanations. Procedures for updating these flags are provided in Section 15.

10.1 ESDT Descriptor Files

The primary task in establishing a collection is providing the core and product-specific metadata attribute values. This is done by creating an Earth Science Data Type (ESDT) descriptor file. The descriptor file is also used to specify the data services that are available for granules that belong to the collection. These services are implemented as methods of a Dynamic Link Library (DLL) containing C++ code to accomplish each service. The descriptor file and the DLL are the means by which a collection is made known to the Science Data Server (SDSRV).

The ESDT descriptor is composed of sections, all in ODL, containing the following information:

- Collection level metadata attributes with values contained in the descriptor.
- Granule level metadata attributes whose values are supplied primarily by the Product Generation Executives (PGEs) during runtime.
- Valid values and permitted ranges for all product-specific attributes.

- List of services for all the granules in the collection and events which trigger responses throughout the system.

The ESDT descriptor file is created by Metadata Works (MDWorks in Figure 10.2-1), discussed in Section 10.3.3.

The services which apply to a collection are specified in the ESDT descriptor file. Metadata Works automatically inserts standard ECS-supplied services such as insert and search into the descriptor file. Product-specific services, such as Subsetting or a product-specific acquire, require executable code to enact those services. This code is contained in the DLL. The DLL is written and tested by either the ECS developers or sustaining engineering personnel at the DAAC.

After the ESDT (both descriptor file and DLL) has been generated it must be installed on the Science Data Server before the first data granule can be inserted. During this installation process, information from the ESDT Descriptor File is propagated to the Data Management and Interoperability subsystems, and to the Subscription Server, which must all be operating during the ESDT installation process. The detailed procedures for ESDT installation into ECS are described in Section 11.3.

10.1.1 Steps in Generating a Descriptor File

ESDTs for Distributable Product

These are the typical steps used in generating a descriptor file:

.Identify desired collection-level metadata attributes

- For permanent and interim files use only the minimum attributes.
- For distributable products identify all applicable attributes. This will involve reading appropriate documentation and interacting with the data provider.

.Identify granule-level attributes

- If a sample metadata configuration file is available from the data provider, use this.

.Check valids for core attributes (write CCR if new valids are required).

.Check PSAs (register PSAs if new).

.Gather metadata into a spreadsheet, or use Metadata Works to enter metadata directly.

.Use custom built scripts to generate the descriptor file from the spreadsheet, if used.

.Verify the descriptor file as outlined below.

.Check descriptor files into ClearCase.

.Notify the DLL Team Lead of the newly prepared descriptor files.

10.1.2 Verifying Descriptor Files

- Run the PERL script "update.pl", following the instuctions in the script prologue. (This script makes sure that the inventory metadata attributes are all listed as event qualifiers in the EVENT group.)
- Run the PERL script "esdtQC.pl", following the instructions in the script prologue. Make any necessary corrections in response to errors issued, and rerun. Repeat until there are no errors. (This script checks for more than 30 common descriptor file errors.)
- Run the PERL script "required.pl", following the instructions in the script prologue. Add any missing attributes as indicated.
- Run the "testodl.csh" utility to ensure that there are no errors in the ODL structure for the descriptor file. Make any necessary corrections in response to errors issued, and rerun. Repeat until there are no errors.

10.2 Preparation of Earth Science Data Types

An ESDT goes through pre-operational life cycle steps starting with an analysis of the collection's need and continuing through development and operational installation. This process involves actions by the Data Provider or User in addition to ECS. These procedures are detailed in Project Instruction SO-1-002, "Earth Science Data Type Generation Procedures". The overall workflow is shown in Figure 10.2-1.

DEFINITIONS

Archive - A File Type indicating granules will be inserted to Data Server for long term storage and acquisition for distribution.

Full - A level of metadata coverage intended for data products which are produced within ECS.

Collection - A related group of data granules.

Granule - The smallest data element which is identified in the inventory tables.

Interim - A File Type indicating granules are temporarily stored in support of product generation.

Intermediate - A level of metadata coverage intended for contemporaneous data products which are not produced within ECS.

Limited - A level of metadata coverage intended for heritage data products brought into ECS for distribution

Minimal -A level of metadata coverage sufficient to uniquely identify a collection or granule.

Permanent - A File Type indicating static or semi-static granules which are used only as inputs in product generation.

Product - Attributes that are defined by the data provider in support of searching for specific granules

Valid - An allowable metadata value.

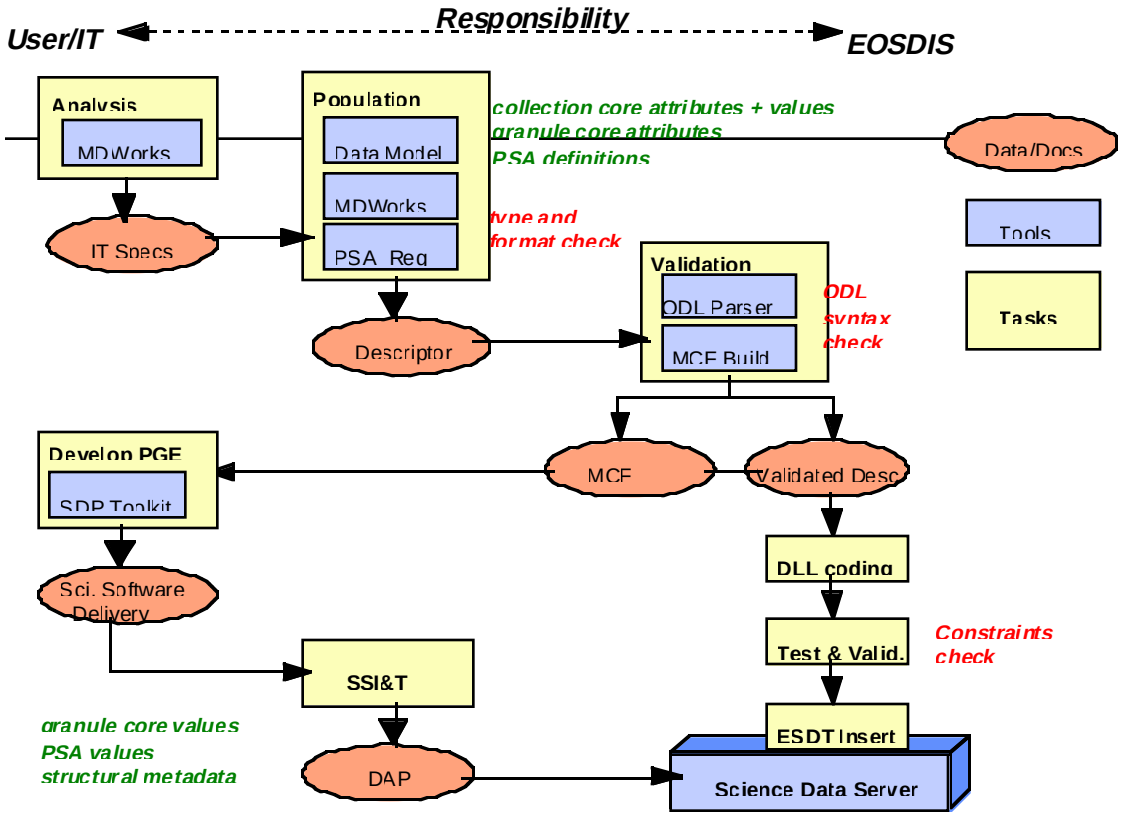


Figure 10.2-1. The ESDT Generation Process

PROCESS:

1. Need Analysis

- The baseline list of science ESDTs and their services is controlled by the ECS CCB. This baseline was established through an analysis of the ECS Functional and Performance Requirements Specification, the Technical Baseline established from inputs from the Ad Hoc Working Group on Production, and meetings with the individual data providers to define the basic requirements of each ESDT.

  These basic requirements are:

  - Data Provider File Designation,
  - File Type (Permanent, Interim, Archive)
  - Level of Metadata Coverage (Minimal, Limited, Intermediate, Full)

- For new ESDTs not currently in the development baseline, the result of the Need Analysis forms the basis for approving the inclusion of the ESDT into ECS. This is accomplished through the CCR process, governed by PIs "Configuration Change Request Preparation" [CM-1-003] and "Change Control Board Process" [CM-1-004].

2. ESDT Specification

- This step results in a set of specifications extending the results of the needed analysis and providing the information needed to implement an ESDT. This step is executed only if the ESDT has been included in the baseline. The roles and responsibilities for developing the specification are as above.

  The specifications must include:

  - ShortName and VersionID of the ESDT

  - A list of the metadata attributes needed, valids, and any constraints on attributes. This list is to be drawn from the B.x Descriptor File Template. This template, under CCB control, is based on the attributes defined in "Release-B SDPS Database Design and Database Schema Specifications" [311-CD-008-001] (i.e., DID 311), as modified for B.0 (for example) by "B.0 Earth Science Data Model " [420-TP-015-002].

  - A list and specification of the services needed (e.g., specification of the INSERT, SEARCH, ACQUIRE and SUBSCRIPTION semantics).

3. ESDT Generation

- Once the ESDT Specification has been developed and the applicable attributes identified, the necessary metadata has to be gathered, the metadata values checked against the valid values and the product-specific attributes (PSA) need to be checked against the list of PSAs that are already defined (see Fig. 10.2-2).

- Once the collection-level metadata and granule-level attributes have been checked, then the descriptor file is generated, the Dynamic Link Library produced, and testing and validation of the ESDT performed. This process is further elaborated in the sections below.

- For a one-of-a-kind, distributable product with Full metadata coverage, this process can take up to six weeks to accomplish. For a related group of products with identical services, much of the Descriptor File and DLL of the first ESDT can be reused, and the cycle time for preparing subsequent ESDTs in the related group is much less.

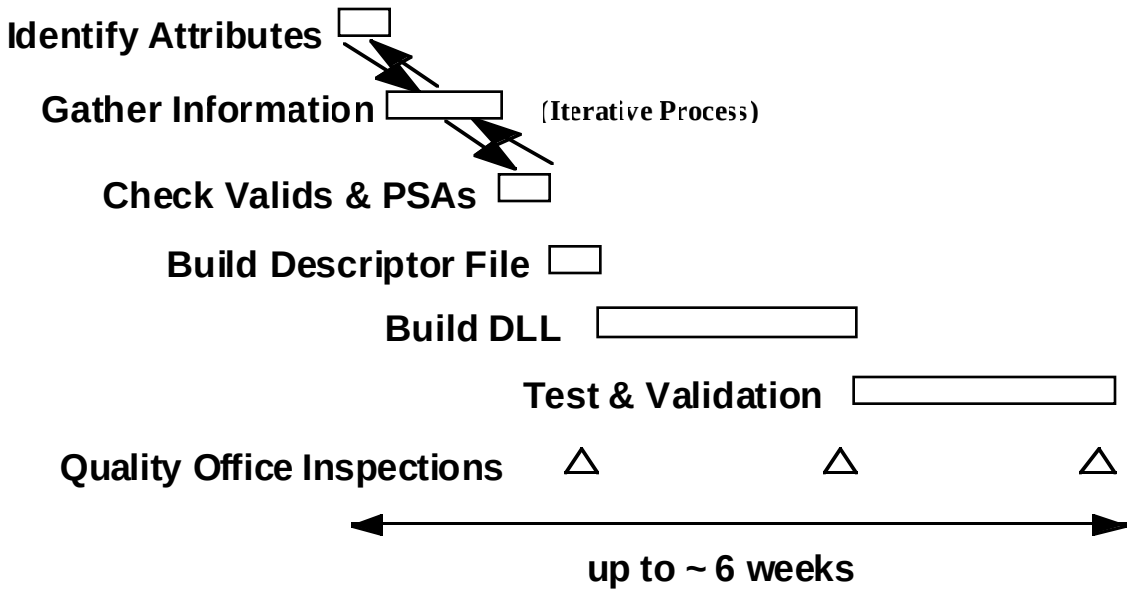


Figure 10.1-2. Steps in ESDT Development

10.3 Tools used in Generating a Descriptor File

Figure 10.3-1 shows the some of the tools that have been developed, and indicates supporting information flows between these tools. Brief descriptions of these tools are provided below.

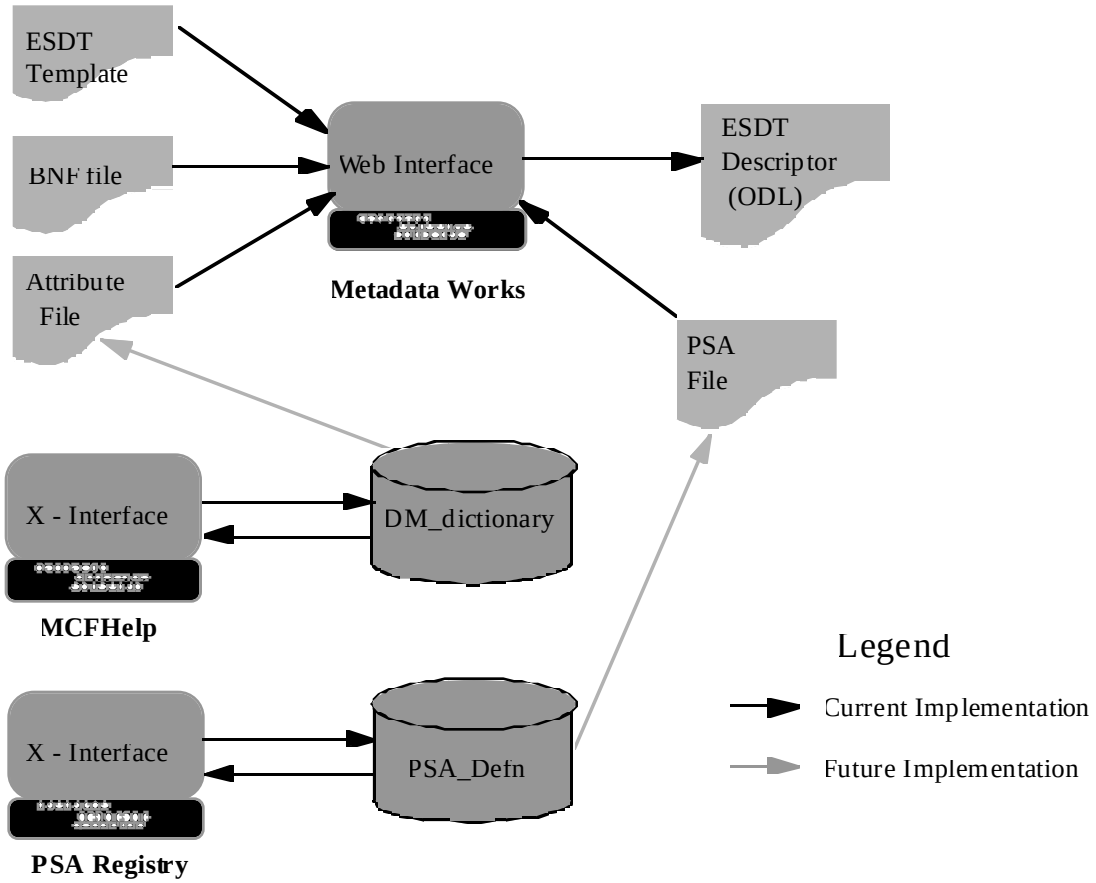


Figure 10.3-1. Tools and Supporting Information Flows

10.3.1 Data Dictionary and Valids Checking

Each proposed new valid must be reviewed by the Data Modeling group in order to validate that they fit within the model. When approved, within a couple days at most, the new valid will be available for use.

10.3.2 PSA Registry

Each Product Specific Attribute must be registered to ensure that the name is uniquely defined across all products and that no two names are applied to the same definition (aliases may be applied at the user's Client and in Data Management, but should not be applied within the inventory). An X-Windows based GUI has been developed to assist in the registration process.

10.3.3. Metadata Works

To support the entry of metadata for ESDTs with Full or Intermediate coverage, an HTML-based GUI has been developed. Metadata Works allows the person entering the metadata to start with a completely empty form, or to define a new descriptor file employing defaults of already defined groups or entire previously defined descriptor files. When the metadata entry is complete for a descriptor file, Metadata Works is then used to generate a complete Descriptor file employing the proper ODL syntax.

10.4 Metadata Population

10.4.1 Collection-Level Metadata

A majority of the attributes in the ECS Data Model apply to all the granules in the collection. These are known as collection-level attributes. There can be both core and product-specific collection-level attributes, defined once prior to establishing the collection.

Collection-level metadata is input via an HTML-forms GUI tool called Metadata Works, described further in Section 10.3.3. Based on the ECS Data Model it is designed for use by the data provider; e.g., an instrument team scientist or other person having extensive knowledge of the data. It can import as defaults the attribute values from a collection that has already been populated. A sequence of screens is presented to the user enabling specification of all required and optional attributes, with a list of permitted values presented where appropriate. Help screens give attribute definitions, data types and other relevant information to assist in the specification.

10.4.2 Granule-Level Metadata

The attributes in the ECS Data Model which can vary on a granule-by-granule basis are known as granule-level attributes. There can be both core and product-specific granule-level attributes.

Granule-level metadata are specified and populated using the Metadata Configuration File (MCF). The MCF is derived from information contained in the ESDT descriptor file and delivered by the Science Data Server for use by the Science Data Processing or Ingest Subsystems. The MCF specifies how the searchable metadata attributes will be populated in the SDSRV database. For data products generated by ECS, the science software or Product Generation Executive (PGE) interacts with the MCF using metadata tools contained in the Science Data Processing Toolkit. Through this process values are set for metadata attributes specified in the "source" MCF, such as the temporal or spatial coverage of each granule. These values are then inserted into a "target" MCF at PGE run time. The MCF is used in a similar manner for data entering ECS through the Ingest subsystem.

Procedures for entering data into ECS through the Ingest subsystem are described in Chapter 16. "Ingest". Procedures for running a PGE are described in Chapters 11.10 "Running a PGE in a Simulated SCF Environment at the DAAC" and 11.13 "PGE Planning Processing and Product Retrieval".

The Inventory Metadata section of Metadata Works is used to capture granule-level metadata specifications. An unofficial MCF may be generated as output from Metadata Works, for testing

purposes. Final testing should always be done with the MCF provided by ECS which is guaranteed to be identical to the one delivered by SDSRV at run time.

The actual population of the granule-level attribute values into ECS inventory data bases takes place during the insert of a data granule into the SDSRV. Each data granule consists of one or more physical files. Accompanying each granule is a metadata record; i.e., an ASCII file containing the granule level attributes and their values in ODL. Only one metadata record is allowed per granule, i.e. no. sub-granule records are allowed, and no metadata records are shared between granules.

Procedures used to initiate the running of PGEs are described in Chapter 11.13 "PGE Planning and Processing".

10.4.3 Product-Specific Metadata

Product-specific metadata can be at both the granule-level and the collection-level. Product-specific metadata may (at the data providers election) be contained in the inventory tables in the database, in which case it will be searchable by ECS. There is also a provision to store product-specific metadata within granules that is available only when the granule has been ordered and delivered. This is termed archive metadata and is specified in a separate ODL group in the MCF.

In the granule metadata, the core attribute that is available to store product-specific metadata is called ParameterValue. The metadata describing this attribute is specified by the data provider through the AdditionalAttributes class at the collection-level. The units of measure, range, accuracy and resolution for this is specified in the PhysicalParameterDetails class, also at the collection-level.

Product-specific metadata at the collection level is specified with Metadata Works at the time the other collection level metadata attributes values are defined. At the granule-level, product-specific metadata is defined in the MCF. In both cases, a list of valid values and permitted ranges are specified in the ESDT data dictionary.

10.5 Testing and Validation

Testing and validation involves installation of the ESDT on a Data Server, and subsequent tests of the data services for the ESDT. These tests include insertion of actual or simulated data, search acquire, and other services that may apply and be available and supported under the extant version of ECS. (Section 11.3)

After testing, the ESDT Descriptor File and DLL are promoted to the development CM environment if pre-ECS release, or to the operational environment if after the ECS release is made operational.

This page intentionally left blank.

11. SSIT Operational Procedures

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

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

A Note About the Order of Procedures

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

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

Assumptions

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

Conventions

The following conventions are followed for explaining procedures:

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

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

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

11.1.1 Key Operator Roles

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

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

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

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

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

MODIS Science Data Processing Software Version 2.0 System Description Manual

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

11.1.2 COTS Software Tools

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

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

11.1.3 General Process

The SSI&T process consist of two activities:

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

Key Terms:

- **Product Generation Executives (PGEs)** - The smallest scheduled unit of science software.

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

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

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

11.1.3.1 General

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

- SSI&T activities can be broadly separated into two categories: pre-SSI&T and formal SSI&T. Pre-SSI&T activities are those which do not involve the ECS Planning and Data Processing (PDPS) or the Science Data Server (SDSRV), but the formal SSI&T activities do involve the full ECS including the PDPS and the SDSRV.

- Most steps in the SSI&T process are inter-related and some steps may assume that another step has been completed. The ordering of the steps is very important but it cannot, however be interpreted as a detailed, step-by-step guide to SSI&T activities.

- Science Software Integration and Test consists of the following activities most of which are fully detailed in Science Software Integration & Test Operational Procedures for the ECS Project (162-TD-001).
- The activities described in the following are also depicted in a (SSI&T Process Flow Diagrams 1 and 2) see **Figure** 11.1.3.1-1 and **Figure** 11.1.3.1-2.

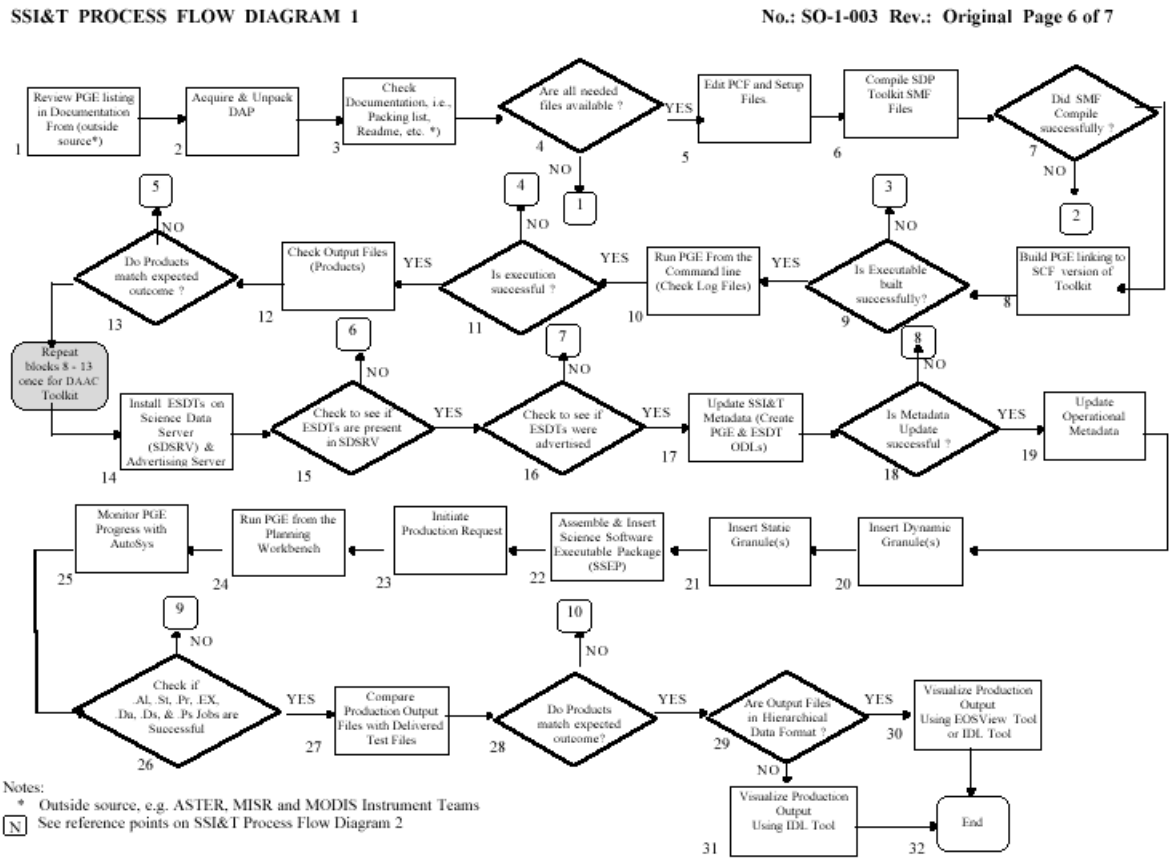


Figure 11.1.3.1-1. SSI&T Process Flow Diagrams 1

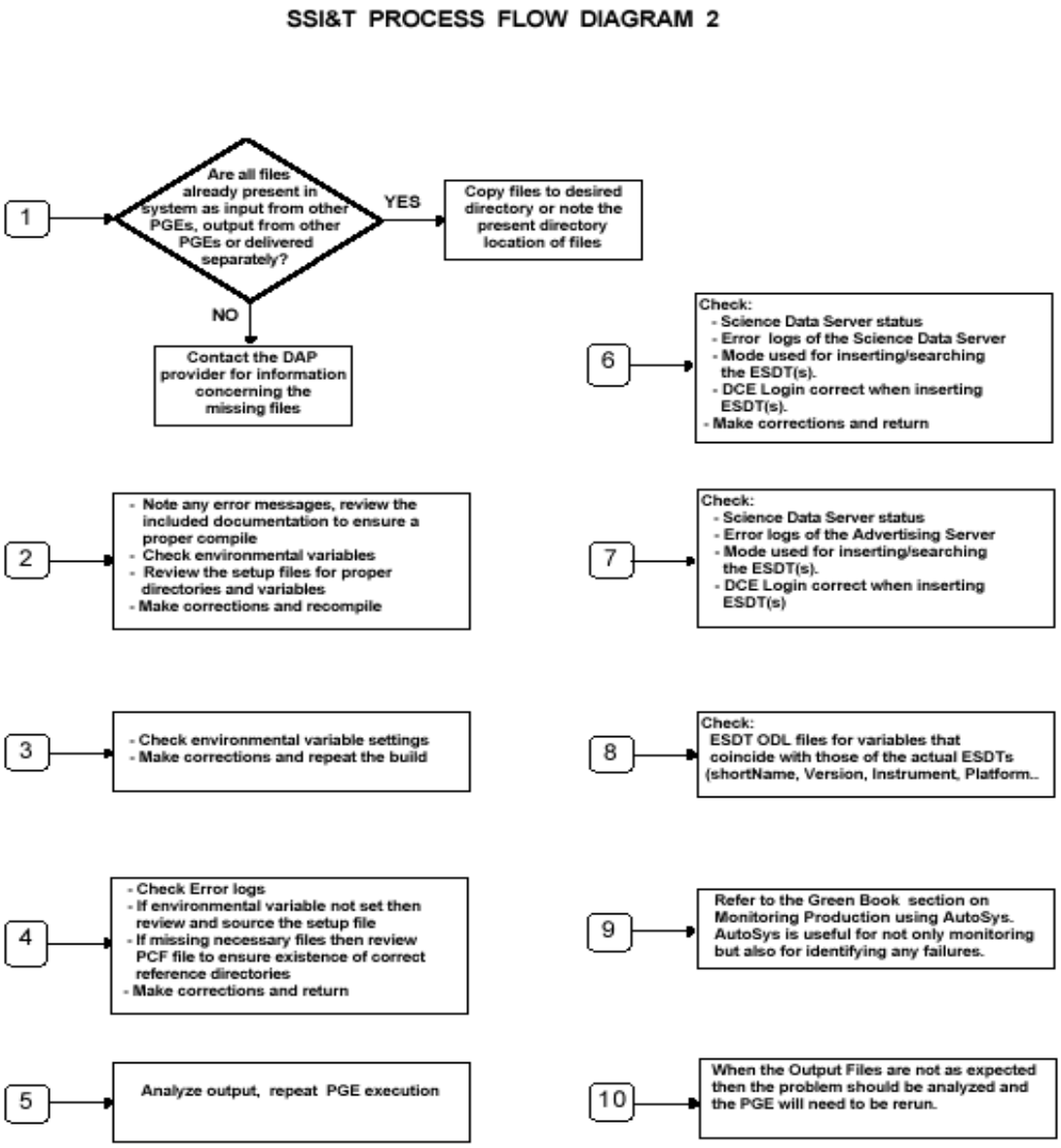


Figure 11.1.3.1-2. SSI&T Process Flow Diagrams 2

11.1.3.2 Pre-SSI&T Activities

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

2 The DAP is acquired and unpack and the documentation (i.e., packing list, readme, etc.) checked. The DAP contents are further checked by the Science Data Specialist to verify that the contents match the packing list, agreed-upon directory structures are employed, location of files are correct, and all intended files and directories are present.

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

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

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

- Note any error messages and review the included documentation to ensure a proper compile;
- Check environmental variables;
- Review the setup files for proper directories and variables;
- Make corrections and recompile.
- If the executable builds successfully, proceed to Step 12. If the build fails, it may necessary to do one or more of the following before proceeding:
- Check environmental variables;
- Make corrections and repeat the build.

6 Run the PGE from the Command Line.

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

- Check error logs;
- Check environmental variables;
- Review and source the setup files;
- If necessary files are missing, then review the PCF file to ensure the existence of correct reference directories.

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

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

9 The SSI&T team examines the output log files from the PGE runs for any anomalous message. The SSI&T team compares the output product data with the delivered test data using the file comparison tools. If the products do not match the delivered test outputs (expected outcome), the outputs should be analyzed and the PGE must be re-run. If the products match the delivered test outputs then

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

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

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

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

11.3.3.3 Formal SSI&T Activities

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

RECORDS

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

PERFORMANCE MEASUREMENTS

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

11.1.4 Preparation and Setup

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

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

If you access the workstation through a remote login (rlogin), you must enter **xhost <remote_workstation_name>** then press the **Enter** key.

Once you have entered xhost <remote_workstation_name>, but prior to the remote login, enter **setenv DISPLAY** <local workstation IP address>**:0.0** where the local workstation IP address represents the IP address you where you are located**.**

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

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

The **Enter Password** prompt is displayed.

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

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

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

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

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

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

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

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

- **ECS Assistant, ADSRV/DM/IOS, incagold,**
- **ECS Assistant, SDSRV/DSS, texas**
- **ECS Assistant, DPS, taltos**
- **ECS Assistant, SBSRV/CSS/IOS, sahara**
- **SSIT Manager tools, AITTL/DPS calahans**
- **Production Request, PLS, odyssey**
- **Planning Workbench, PLS, odyssey**
  Note: NETSCAPE should be closed to allow for a full screen GUI to be activated.

- **Monitor PGE, odyssey**

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

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

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

11 The above servers have unique hosts assigned. Each host needs to be logged into the **generic login: cmts1, pw: ecsu$er, dce_login awhitele awhitele** before activating ECS Assistant to carryout the downing and bringing up of servers assigned to their respective hosts.

11.1.5 SSIT Software Operating Instructions:

Starting the SSIT Manager GUI:

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

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

If you access the workstation through a remote login (rlogin), you must enter **xhost <remote_workstation_name>** prior to the rlogin, and enter **setenv DISPLAY <local_workstation IP address>:0.0** after the rlogin, before entering the command. The **<ipaddress>** is the ip address of **x0ais##,** and **xterm** is required when entering this command on a Sun terminal.

---

1 **Example:** Log into an Algorithm and Test Tools (AITTL) environment using using a machine so configured. At the mini-daac this machine is **calahans.** A special host has been established using the id: **cmts1** and password: **ecsu$er**. Type: **setenv DISPLAY ….:0.0**

2 *Login to DCE* **(dce_login <awhitele awhitele**>*)*

What the user must do before trying SSIT functionality:

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

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

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

What must be done via SSIT tools:

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

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

The checklist (first window on the GUI) allows you to check off various activities by double clicking on them. You may enter a commentary on the activity in the second window when checking off a particular item. The file checklist.sample in the data/DPS directory can be edited to change the items in the checklist or its' location.

11.1.6 Updating the Leap Seconds and the Earth Motions files

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

- **update_leapsec.sh**

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

- **update_utcpole.sh**

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

- **The Leap Seconds file:**

leapsec - file ID: **$PGSHOME/database/common/TD/leapsec.dat**

(Atomic time from International Earth Rotation Service)

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

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

- **The Earth Motion file:**

utcpole – file ID: **$PGSHOME/database/common/CSC/utcpole.dat**

(Record of the Earth's variable of slowing rotation with respect to UTC Time.)
Interval of update recommended: Twice weekly except Sundays 17:45 hours

to 17:55 Eastern US time. Recommended scripts be run in the afternoon or evening each Tuesday and Thursday.

11.1.7 Script Name: update_leapsec.sh

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

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

- **Function to be applied: PGS_TD_NewLeap,** excerpts and reformats the new information and appends new data and date to **leapsec.dat** file. A remark that the file was either "Checked" (no new Leap second) or "Updated" (new leap second). 11.2.2 Script Name: update_utcpole.sh

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

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

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

Guidelines:

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

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

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

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

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

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

Hardware Needed and Setup Procedures

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

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

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

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

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

NOTE: The **x** in the workstation name will be a letter designating your site:

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

If you access the workstation through a remote login (rlogin), you must enter **xhost <remote_workstation_name>** prior to the rlogin, and enter **setenv DISPLAY <local_workstation IP address>:0.0** after the rlogin, before entering the command. The **<ipaddress>** is the ip address of **x0spg##,** and **xterm** is required when entering this command on a Sun terminal.

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

1 telnet to a machine that supports the Toolkit. (**telnet lasher**)
2 login: **cmops**, Password: **opsu$er**
3 **setenv DISPLAY ….:0.0**
4 **setenv PGSHOME /usr/ecs/OPS/CUSTOM/TOOLKIT/toolkit**
5 **cd /usr/ecs/OPS/CUSTOM/TOOLKIT/toolkit/bin/sgi_daac_f77** then
6 **source pgs_dev-env.csh**
7 For **leapsec: cd /usr/ecs/OPS/CUSTOM/TOOLKIT/toolkit/database/common/TD**
8 **cp leapsec.dat leapsec.dat_old**
9 Know thread for Leap Second run:
10 **cd /usr/ecs/OPS/CUSTOM/TOOLKIT/toolkit/src/TD** then do **ls** – select:
update_leapsec.sh or run script for Leap Second type in: **update_leapsec.sh**

A successful update will look like the following

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

12 For **utcpole:**
13 **cd /usr/ecs/OPS/CUSTOM/TOOLKIT/toolkit/database/common/CSC**
14 **utcpole.dat utcpole.dat_old**
15 Know thread for utcpole run:
16 **cd /usr/ecs/OPS/CUSTOM/TOOLKIT/toolkit/src/CSC** then do **ls** – select:
update_utcpole.sh or run script for utcpole type in: **update_utcpole.sh**
17 A successful update will look like the following:
lasher{cmops}[294]->update_utcpole.sh
Status of PGS_CSC_UT1_update call was (0)
Status of MOVE command was (0)

11.2 Science Software Integration and Test (SSIT) Manager

11.2.1 SSIT Manager Overview

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

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

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

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

- **Code Analysis** contains
    - **SPARCwork** - A COTS package provided by Sun that allows for various coding activities including memory checking and debugging.
- **Office Automation** contains
    - **MSWindows** - a Microsoft Windows emulator with MS Office (Word, Excel, PowerPoint) installed.
    - **Ghostview** - for viewing PostScript formatted documents.
    - **Netscape** - WWW browser and useful for viewing HTML formatted documents.
    - **Acrobat** - for viewing PDF formatted documented.
    - **DDTS** - for entering and tracking science software problems.
- **Standards Checkers** contains
    - **FORCHECK** - for standards checking for FORTRAN 77 and Fortran 90 science software source code.
    - **Prohibited Function Checker** - for checking science software source code for prohibited functions.
    - **Process Control File Checker** - for checking Process Control Files (PCFs) delivered with science software.
    - **Prolog Extractor** - for extracting prologs from science software source code.
- **Product Examination** contains
    - **IDL** - Interactive Data Language tool supported by Sun. SSIT puts the user in the

IDL environment.

    - **EOSView** - for viewing HDF and HDF-EOS files.
    - **File Comparison** contains
        - **ASCII** - for comparing two output products that are in ASCII format.
        - **Binary** - for comparing two output products that are in binary format.
        - **HDF (GUI)** - for comparing two output products that are in HDF or HDF-EOS format, GUI version.
        - **HDF (hdiff)** - for comparing two output products that are in HDF or HDF-EOS format, command line tool.

- **Text Editors** contains
  - **Emacs**
  - **Xedit**
  - **vi**
- **PDPS Database** contains
  - **PCF ODL Template** - for converting delivered PCFs into ODL during PGE registration.
  - **Check ODL** - for verifying ODL syntax of ODL files.
  - **SSIT Science Metadata Update** - for updating the PDPS database with PGE information during PGE registration.
  - **SSIT Opnl Metadata Update** - GUI for updating the PDPS database with PGE information during PGE registration.
  - **Copy SSIT -> Production** - for copying PGE registration database information from SSI&T mode to Production mode.
- **Data Server** contains
  - **Acquire DAP** - for acquiring a Delivered Algorithm Package (DAP).
  - **Get MCF** - Source MCF is a Metadata Configuration File used to create a Target MCF (.met) for a Dynamic/Static Granule
  - **Insert Static** - for inserting a static data file to the Data Server.
  - **Insert Test Dynamic** - for inserting a dynamic test data file to the Data Server.
  - **Insert EXE TAR** - for inserting a Science Software Executable Package (SSEP) to the Data Server.
  - **SSAP Editor** - for editing and creating a Science Software Archive Package (SSAP) and inserting it to the Data Server.

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

11.2.2 SSIT Manager GUI

This GUI (Figure 11.2-1) is the starting point for SSI&T activities. It provides access to a collection of tools that will be useful for this purpose.

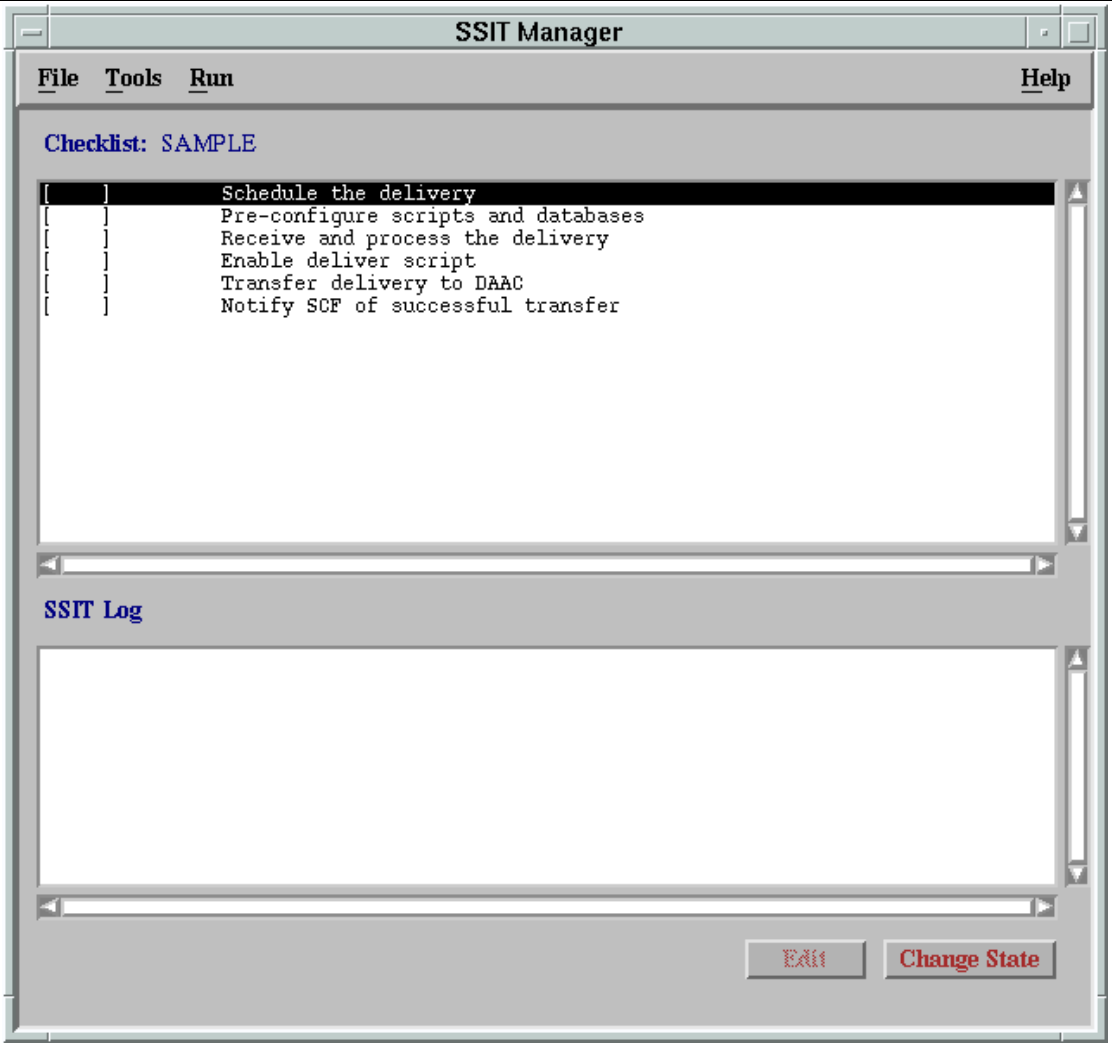


Figure 11.2-1. SSIT Manager Window

11.2.2.1 General Set Up of the SSIT Manager

The SSIT Manager requires a configured environment within which to run; it runs only on the AIT Suns. The set up steps described in this section need only be done the first time a SSI&T operator uses the SSIT Manager. At a UNIX prompt on an AIT Sun, (type in: **xhost <remote_workstation_name>**).

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

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

1 login to: dce_login awhitele awhitele

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

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

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

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

- The ***mySSITpcf*** is the full path name to the private copy of the PCF to be used with the SSIT Manager when you run it (from step 1).
- It may be useful to add this line to your .cshrc (or other start up script) so that it is set every time you login.

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

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

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

- This invokes the **SSIT Manager GUI** which should be displayed.
- The checklist displayed within the GUI will be the default.
- This sets environment variables and other settings needed for running the SSIT Manager.

11.2.2.2 Set Up of a Checklist for the SSIT Manager

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

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

Creating a User-Defined Checklist for the SSIT Manager:

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

- The EOSView GUI will be displayed.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

11.2.3 SSIT Manager Tools

There are several tools that are accessible through the SSIT Manager GUI. After selecting the TOOLS menu option of the menu bar, a set of options is available. See Figure 11.2.3-1, which indicates the use of the Tool menu item.

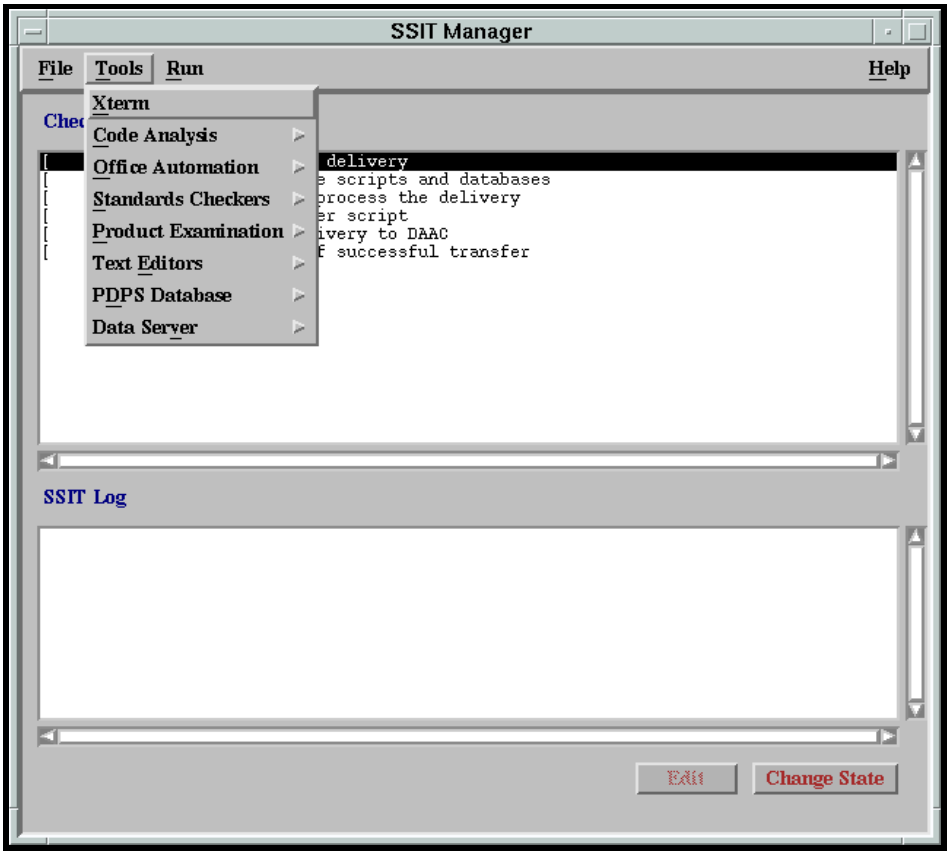


Figure 11.2.3-1. SSIT Manager Window - Tools Menu

11.2.4 Using the SSIT Manager:

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

1. The SSIT Manager is running.
2. The source file(s) are available, accessible, and have read permissions.
3. The below listed formatted text (ASCII) files containing the list of prohibited functions exist in the directory stored in the environment variable DPATMGR_DAT:
4. prohibitedFunctionsAda.txt
5. prohibitedFunctions.C++.txt
6. prohibitedFunctions.C.txt
7. prohibitedFunctions.F77.txt

8. prohibitedFunctions.F90.txt
9. If the source code files to be checked are in a VOB in ClearCase, a view has been set before the SSIT Manager was started.

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

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

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

11.3.1 Acquiring the Delivered Algorithm Package (DAP)

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

11.3.1.1 Acquiring the DAP via FTP

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

Acquiring the DAP via FTP

---

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

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

If you access the workstation through a remote login (rlogin), you must enter **xhost <remote_workstation_name>** then press the **Enter** key.

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

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

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

The ***machineIPaddress*** is the IP address or fully qualified domain name of the remote SCF machine. For example, **ftp 192.116.53.2**, then press the **Enter** key.

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

An ftp session is established.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

11.3.1.2 Acquiring the DAP from the Archive after Ingest

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

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

When a DAP is inserted into the Data Server by Ingest, an email is sent to all users who subscribe to that event (Section 11.3.2).

11.3.1.3 Performing a DAP Acquire Using SSIT Manager

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

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

Assumptions:

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

DAP Acquire Procedures:

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

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

3 After a short while, the SSIT Manager GUI will appear. From the SSIT Manager top menu bar, select **Tools -> Data Server -> Acquire DAP**
See figure 11.2.3-1. If the SSIT Manager GUI is used to initiate the DAP processing, Step 4 can be skipped.

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

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

5 The user will be prompted with:

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

To respond, type **<RETURN>**

6 The user will be prompted with:

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

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

7 The user will be prompted with:

Name of email message file (including path)?

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

The user will be prompted with:

Directory to receive staged file?

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

11.3.2 Unpacking a DAP

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

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

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

Unpacking a DAP

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

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

If you access the workstation through a remote login (rlogin), you must enter **xhost <remote_workstation_name>** then press the **Enter** key.

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

The ***UnpackPathname*** is the path name of the directory that has been set aside for unpacking of DAPs.

This directory now contains the DAP tar file. For example, **cd /home/*user***, where ***user*** is the user's login directory, then press the **Enter** key.

4 If the tar file is compressed, at a UNIX prompt, type **uncompress *PackedDAP.Z***, then press the **Enter** key.

The ***PackedDAP.Z*** is the file name of the compressed DAP file.

The file name extension of **.Z** is a convention indicating UNIX compressed files. The **uncompress** utility expects this file name extension by default. A resulting error may indicate that the DAP file was not compressed or that another compression utility was used. If the file name extension was **.Z**, the uncompressed version will have the same file name but without the **.Z**, for example ***PackedDAP***.

The tar file for the SSI&T Training will not be compressed.

5 At the UNIX prompt, type **tar xvf *PackedDAP***, then press the **Enter** key.

The ***PackedDAP*** is the file name of the uncompressed DAP file.

The tar archive will be unpacked in the current directory. If the archive contained directories and subdirectories, these will be created by the tar utility and populated by the files that belong.

11.4 Science Software Configuration Management

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

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

11.4.1 ClearCase Overview

All data managed under ClearCase are stored in Versioned Object Bases (VOBs), which are the "public" storage areas and Views, which are the "private storage areas. VOBs are data structures that can only be created by the CM administrator using the mkvob ("make vob") command. A VOBs is mounted as a file system and when viewed through a view, it appears as a standard UNIX directory tree structure. This file system, accessed through its mount point, has a version-control dimension which contains file elements and versions of file elements. Once reviewed, the System Administrator will place these files under configuration control. In order to accomplish this task, a view must be created in ClearCase. A view is necessary in order to make visible and accessible files and directories that have been checked in to a VOB.

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

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

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

A Versioned Object Base is defined by the following characteristics:

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

A VOB is comprised of:

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

11.4.2 Creating a View in ClearCase

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

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

A view is defined by the following characteristics:

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

A view is comprised of:

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

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

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

Assumptions:

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

11.4.2.1 Creating a View in ClearCase Using Command Lines

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

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

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

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

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

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

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

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

11.4.2.2 Creating a View in ClearCase using the File Browser Screen

Selecting a view listed in the View Tag Browser screen brings up the File Browser, or main screen, shown in Figure 11.4.2-1.

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

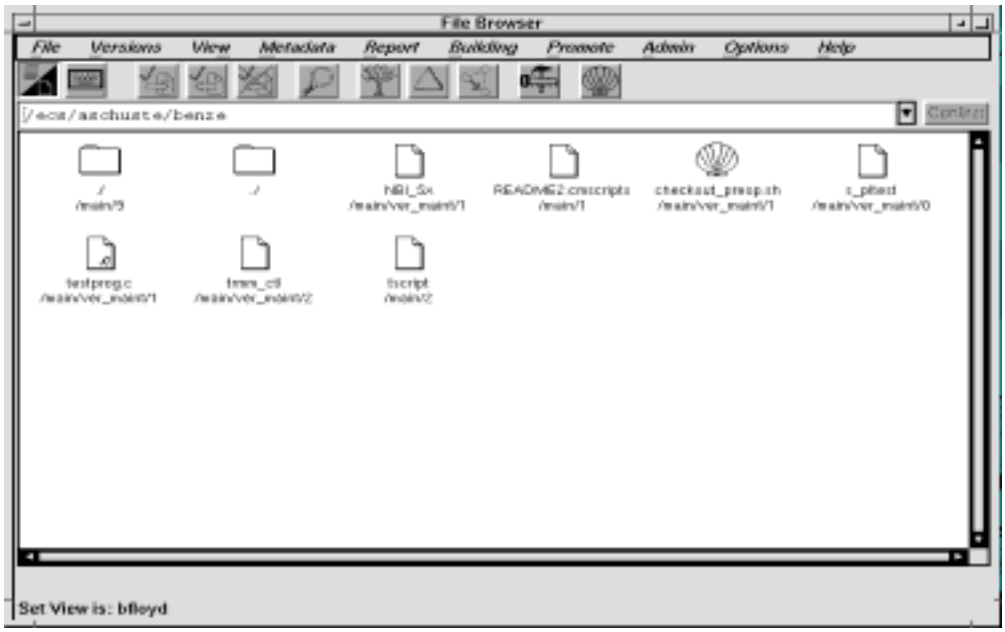


Figure 11.4.2-1. ClearCase File Browser Screen (Main Screen)

Procedures

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

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

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

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

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

6 Invoke the make view command by typing **mkview [filename]** on the UNIX command line and press the **Enter** key.
The **tempdisp** window appears.
The **View [filename] Created Successfully** and the **Cache Updated for View [filename]** prompts are displayed.

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

8 Select **View →List** from the menu.
- The **View Tag Browser** is displayed.

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

11.4.3 Setting a View in ClearCase

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

11.4.3.1 Setting a View in ClearCase Using Command Lines

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

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

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

11.4.3.2 Setting a View Using the File Browser Screen in ClearCase

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

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

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

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

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

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

11.4.4 Creating a New Directory

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

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

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

11.4.4.1 Creating a New Directory in ClearCase Using Command Lines

---

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

This command checks in the current directory.

11.4.4.2 Entering a New Directory Using the File Screen Browser into ClearCase

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

Cursor moves to the **Password** field.

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

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

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

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

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

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

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

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

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

9 On the **File Browser** screen, follow the menu path **File→Exit**.
The ClearCase Graphical User Interface session is closed.

11.4.5 Importing files into ClearCase

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

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

1. A VOB and subdirectory are created to hold these files.
2. No object files or executables exist in the source code directory.
3. The PGE was received with a directory structure that contains various types of files.
4. These files will be entered into ClearCase and will maintain the same directory structure as the delivery structure.

11.4.5.1 Importing a Single File into ClearCase

Procedure:

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

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

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

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

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

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

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

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

9 At the UNIX prompt, type **cleartool checkin -nc .**, press **Enter** (note the space and then "dot" at the end of the command).

This command checks in the current directory (represented by the "dot") into ClearCase.

The adding of an element (here, a file) represents a modification to the directory and hence, the new version of the directory must be checked back in.

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

11.4.5.2 Importing Multiple Files into ClearCase

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

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

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

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

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

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

Importing Multiple Files Into ClearCase

---

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

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

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

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

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

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

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

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

The remaining steps are accomplished by the VOB Administrator.

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

Cursor moves to the **Password** field.

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

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

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

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

The File Browser window is displayed.

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

The String Browser window is displayed.

The prompt Enter shell command to run is displayed.

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

The **tempdisp** window appears.

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

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

The **tempdisp** window closes.

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

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

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

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

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

A separate shell window is displayed.

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

A separate shell window is displayed.

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

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

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

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

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

The ClearCase Graphical User Interface session is closed.

11.4.6 Checking Out a File From ClearCase

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

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

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

Checking Out an Element/File from the Command Line

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

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

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

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

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

Checking Out an Element/File from the File Screen Browser

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

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

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

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

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

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

- **User Group Others**
- **r=read, w=write, x=execute**

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

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

11.4.7 Checking a Modified Element into ClearCase

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

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

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

11.4.7.1 Checking a Modified Element/File into ClearCase

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

Cursor moves to the **Password** field.

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

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

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

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

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

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

This command checks in the current directory.

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

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

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

11.5 Standards Checking of Science Software

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

11.5.1 Checking FORTRAN 77 ESDIS Standards Compliance

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

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

Assumptions:

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

FORCHECK is available only on the AIT Suns.

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

---

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

- If necessary, before logging onto AIT Sun, use the xhost command to allow X Window reception.
- Once logged onto proper Sun, remember to set the DISPLAY environmental variable to point to your X Window screen.

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

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

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

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

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

- Type **do_buildrc**
- Type **do_ssit_man**

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

- From the top menu bar, select **Tools.**
- From the Tools menu, select **Standards Checkers.**
- From the Standards Checkers menu, select **FORCHECK.**
- See Figure 11.8.5-2. for a screen snapshot of this step.

6 A separate FORCHECK window will now open.

- The user will be prompted for input. The first prompt will be *global option(s) and list file?*
- The second prompt will be *local option(s) and file(s)?*
- The second prompt will be repeated until there is a blank line and carriage return.

- In order to understand what the proper responses should be, the user is encouraged to find hardcopy documentation for FORCHECK or to use the UNIX man facility and type *man forchk* .

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

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

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

- The ***ListFile*** is the file name for the list file specified at the FORCHECK prompt.
- The ***ListFile*** file will contain FORCHECK messages similar to the ***FORCHECKoutput*** file embedded in the source code listing.

Any text editor may be used for this procedure step.

11.5.2 Checking for ESDIS Standards Compliance in Fortran 90

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

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

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

Assumptions:

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

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

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

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

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

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

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

- The ***ViewName*** is the name of a view allowing the Fortran 90 source files to be accessible.
- This step is only necessary if any of the Fortran 90 source files are in ClearCase (in the VOB under configuration management).

4 At the UNIX prompt on the SPR SGI, type **cd *SrcPathname***, press **Return**.

- The ***SrcPathname*** is the full path name to the location of the Fortran 90 source files to be checked.
- The ***SrcPathname*** will be in the ClearCase VOB is the Fortran 90 source files are checked into ClearCase.

5 At the UNIX prompt on the SPR SGI, type **f90 -c -ansi *[*-I$PGSINC*] [*-I$HDFINC*] [[*-IOtherIncFiles*…] SourceFiles* >& *ReportFile***, press **Return**.

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

- For example, type **f90 -c -I$PGSINC -I$HDFINC -I/ecs/modis/pge5/include/ *.f90 >& pge10.report**, press **Return**.

6 At the UNIX prompt on the SPR SGI, type **vi *ReportFile***, press **Return**.

- The ***ReportFile*** is the file name for the compilation results as produced in step 5.
- Any text editor may be used for this procedure step.

11.5.3 Checking for ESDIS Standards Compliance in C

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

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

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

Assumptions:

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

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

---

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

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

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

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

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

- The ***ViewName*** is the name of a view allowing the C source files to be accessible.

- This step is only necessary if any of the C source files are in ClearCase (in the VOB under configuration management).

4 At the UNIX prompt on the SPR SGI, type **cd *SrcPathname***, press **Return**.

- The ***SrcPathname*** is the full path name to the location of the C source files to be checked.
- The ***SrcPathname*** will be in the ClearCase VOB is the C source files are checked into ClearCase.

5 At the UNIX prompt on the SPR SGI, type **cc -c -ansi *[*-I$PGSINC*] [*-I$HDFINC*] [[*-IOtherIncFiles*…] SourceFiles* >& *ReportFile***, press **Return**.

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

6 At the UNIX prompt on the SPR SGI, type **vi *ReportFile***, press **Return**.

- The ***ReportFile*** is the file name for the compilation results as produced in step 5.
- Any text editor may be used for this procedure step.

11.5.4 Checking for ESDIS Standards Compliance in Ada

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

Unlike with FORTRAN 77, Fortran 90, or C, Ada compilers are subjected to a validation process by the DoD Ada Committee. Thus, any code that compiles successfully by a validated compiler is, by definition, fully ANSI compliant. Since the Ada compiler is used, the checking for

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

11.5.4.1 Checking for ESDIS Standards Compliance in Ada: Verdix COTS

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

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

Assumptions:

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

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

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

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

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

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

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

6 At the UNIX prompt on the SPR SGI, type **a.make -v -f *SourceFiles* >& *ReportFile***, press **Return**.

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

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

- The ***ReportFile*** is the file name for the compilation results as produced in step 6.
- Any text editor may be used for this procedure step.

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

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

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

Assumptions:

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

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

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

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

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

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

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

3 At the UNIX prompt on the SPR SGI, type **setenv SGI_ABI -32**, press **Return**.

- This command sets the environment variable **SGI_ABI** for 32-bit mode compilation.

4 At the UNIX prompt on the SPR SGI, type **cd *SrcPathname***, press **Return**.

- The ***SrcPathname*** is the full path name to the location of the Ada source files to be checked.
- The ***SrcPathname*** will be in the ClearCase VOB is the Ada source files are checked into ClearCase.

5 At the UNIX prompt on the SPR SGI, type **gcc -c -gnat83 *SourceFiles* >& *ReportFile***, press **Return**.

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

6 At the UNIX prompt on the SPR SGI, type **vi *ReportFile***, press **Return**.

- The ***ReportFile*** is the file name for the compilation results as produced in step 5.
- Any text editor may be used for this procedure step

11.5.5 Prohibited Function Checker

The use of certain functions in the PGE is prohibited. The Prohibited Function Checker (Figure 11.5.4-1) is used to check C, FORTRAN 77, FORTRAN 90, and Ada language source files for the occurrence of functions that are prohibited in the ECS DAAC production environment.

11.5.5.1 Checking for Prohibited Functions: Command-Line Version

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

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

Assumptions:

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

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

---

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

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

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

- The ***SrcPathname*** is the full path name to the location of the source files to be checked.
- The ***SrcPathname*** will be in the ClearCase VOB if the source files are checked into ClearCase.
- The ***SrcPathname*** can contain other directories that contain source files and/or more directories. The Prohibited Function Checker will search out all source files in subdirectories recursively.

3 At the UNIX prompt on the AIT Sun, type **/data3/ecs/TS1/CUSTOM/bin/DPS/EcDpAtMgrBadFunc ConfigFile /data3/ecs/TS1/CUSTOM/cfg/EcDpAtBA.CFG *FilesOrDirectories* > *ResultsFile***, press **Return**.

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

4 At the UNIX prompt on the AIT Sun, type **vi *ResultsFile***, press **Return**.

- The ***ResultsFile*** is the file name for the output results as produced in step 3.
- Any text editor may be used for this procedure step.

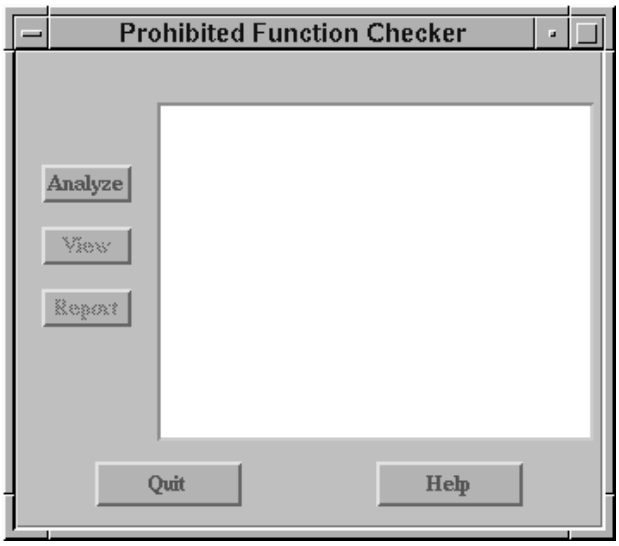


Figure 11.5.5-1. Prohibited Function Checker

Prohibited Function Checker GUI

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

The Prohibited Function Checker GUI will be displayed.

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

The File Selector GUI will be displayed.

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

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

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

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

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

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

The File Selector GUI will disappear.

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

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

The **Report** GUI will be displayed.

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

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

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

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

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

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

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

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

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

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

The Prohibited Function Checker GUI will disappear.

This ends the session.

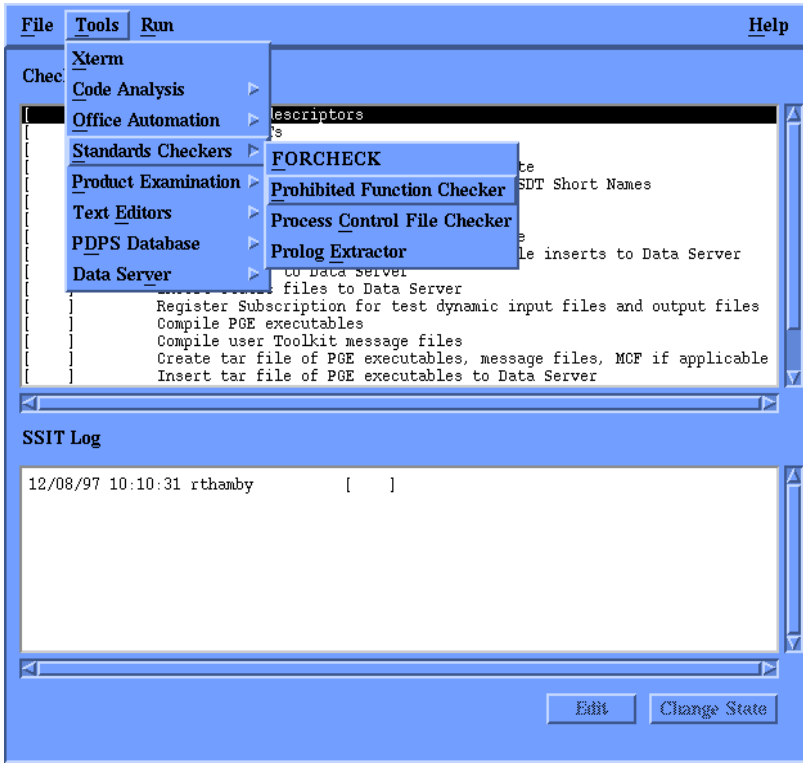


Figure 11.5.5-2. Invoking the Prohibited Function Checker•

11.5.6 Checking Process Control Files

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

11.5.6.1 Checking Process Control Files GUI

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

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

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

Checking Process Control Files GUI

1 From the SSIT Manager, select **Tools → Standards Checkers → Process Control File Checker** from the menu, see figure 11.5.6-1.

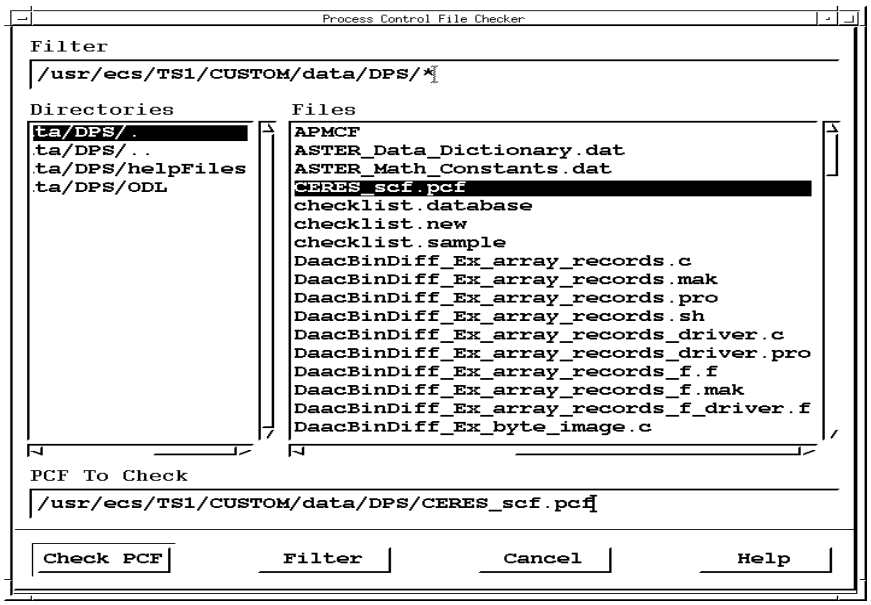


Figure 11.5.6-1. Process Control File Checker GUI

The Process Control File Checker GUI will be displayed.

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

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

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

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

The file clicked on will be highlighted.

Only one PCF can be checked at a time.

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

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

Results will be displayed in this window.

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

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

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

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

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

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

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

11.5.6.2 Checking Process Control Files: Command-Line Version

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

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

Assumptions:

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

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

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

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

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

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

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

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

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

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

11.5.7 Extracting Prologs

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

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

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

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

Prolog Delimiters•

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

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

File Name Extensions

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

11.5.7.1 Extracting Prologs

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

or

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

An xterm will be displayed on the AIT Sun.

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

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

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

Separate items with spaces.

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

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

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

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

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

- The xterm will disappear.

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

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

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

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

```
SOURCE CODE PROLOG EXTRACTOR

Configuration filename? (enter for default: ../../cfg/EcDpAtPrologs.CFG)

ECS mode? (enter for default: OPS)
TS1
File(s)?  (enter -h for help)
/home/dps/ssit/*.c
Warning: Could not open message catalog "oodce.cat"
[Warning:
Invalid Resource Catalog directory path or no catalog installed
Applications can run with or without Resource Catalog
FYI : Values of ECS_HOME env variable and RC Directory path:/usr/ecs/ecsmode/CU
STOM/data/DPS/ResourceCatalogs
]

EcDpAtPrologs: Process Framework: ConfigFile ../../cfg/EcDpAtPrologs.CFG  ecs_m
ode ecsmode

Output written to file: /usr/ecs//TS1/CUSTOM/logs/prologs.txt

Hit return for another, 'q <return>' to quit:
```

Figure 11.5.7-1. Prolog Extractor Sample Run.

11.6 Compiling and Linking Science Software

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

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

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

11.6.1 Updating the Process Control File

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

Assumptions:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

- Each of the above section heading lines will then be followed (not necessarily immediately; there may be comment lines) by a line that begins with a **!** in the first column. These lines specify the default path names for each section.
  - If the line reads:
    `! ~/runtime`

    leave it unchanged. The tilde (~) is a symbol that represents $PGSHOME.
  - If another path name is listed instead, it will probably need to be changed to a path name that exists at the DAAC on the SGI. When specifying a path name, use an absolute path name, not a relative path name.

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

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

  change /MODIS/run/input to the appropriate path name in the DAAC where the file A.granule is stored.
- When specifying a path name, use an absolute path name, not a relative path name.
- Do not include the file name with the path name. The file name belongs in field 2 by itself.

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

- Look for the entry:
  `10111|ShmMem|~/runtime||||1`

  The third field may be blank; this will work too.
- If this entry is not within this section, add it.

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

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

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

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

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

11.6.2 Setting up a SDP Toolkit Environment

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

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

The SDP Toolkit is divided into two groups of tools:

11.6.2.1 Mandatory Tools

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

11.6.2.2 Optional Tools

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

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

TOOL TITLE

NAME:	Procedure or routine name
SYNOPSIS:	C: C language call
FORTRAN:	FORTRAN77 or Fortran90 language call
DESCRIPTION:	Cursory description of routine usage
INPUTS:	List and description of data files and parameters input to the routine
OUTPUTS:	List and description of data files and parameters output from the routine
ENTERS:	List of returned parameters indicating success, failure, etc.

EXAMPLES: Example usage of routine
NOTES: Detailed information about usage and assumptions
REQUIREMENTS: Requirements from PGS Toolkit Specification, Oct. 93 which the routine satisfies

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

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

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

New 64-bit: cc flag = -64
New 32-bit: cc flag = -n32
Old 32-bit: cc flag = -32

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

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

Table 11.6.2-1. SDP Toolkits used by the SGI science processors.

SDP Version	Language Type	Library Object Type	$PGSHOME	$PGSBIN
SCF	FORTRAN 77 or C	Old 32-bit mode	$CUSTOM_HOME/bin/ scf_toolkit_f77/TOOLKIT/	$CUSTOM_HOME/bin/ scf_toolkit_f77/TOOLKIT/ bin/sgi/
SCF	Fortran 90 or C	Old 32-bit mode	$CUSTOM_HOME/bin/ scf_toolkit_f90/TOOLKIT/	$CUSTOM_HOME/bin/ scf_toolkit_f90/TOOLKIT/ bin/sgi/
SCF	FORTRAN 77 or C	New 32-bit mode	$CUSTOM_HOME/bin/ scf_toolkit_f77/TOOLKIT/	$CUSTOM_HOME/bin/ scf_toolkit_f77/TOOLKIT/ bin/sgi32/
SCF	Fortran 90 or C	New 32-bit mode	$CUSTOM_HOME/bin/ scf_toolkit_f90/TOOLKIT/	$CUSTOM_HOME/bin/ scf_toolkit_f90/TOOLKIT/ bin/sgi32/
SCF	FORTRAN 77 or C	64-bit mode	$CUSTOM_HOME/bin/ scf_toolkit_f77/TOOLKIT/	$CUSTOM_HOME/bin/ scf_toolkit_f77/TOOLKIT/ bin/sgi64/
SCF	Fortran 90 or C	64-bit mode	$CUSTOM_HOME/bin/ scf_toolkit_f90/TOOLKIT/	$CUSTOM_HOME/bin/ scf_toolkit_f90/TOOLKIT/ bin/sgi64/
DAAC	FORTRAN 77 or C	Old 32-bit mode	$CUSTOM_HOME/bin/ daac_toolkit_f77/TOOLKIT/	$CUSTOM_HOME/bin/ daac_toolkit_f77/TOOLKIT/ bin/sgi/
DAAC	Fortran 90 or C	Old 32-bit mode	$CUSTOM_HOME/bin/ daac_toolkit_f90/TOOLKIT/	$CUSTOM_HOME/bin/ daac_toolkit_f90/TOOLKIT/ bin/sgi/
DAAC	FORTRAN 77 or C	New 32-bit mode	$CUSTOM_HOME/bin/ daac_toolkit_f77/TOOLKIT/	$CUSTOM_HOME/bin/ daac_toolkit_f77/TOOLKIT/ bin/sgi32/
DAAC	Fortran 90 or C	New 32-bit mode	$CUSTOM_HOME/bin/ daac_toolkit_f90/TOOLKIT/	$CUSTOM_HOME/bin/ daac_toolkit_f90/TOOLKIT/ bin/sgi32/
DAAC	FORTRAN 77 or C	64-bit mode	$CUSTOM_HOME/bin/ daac_toolkit_f77/TOOLKIT/	$CUSTOM_HOME/bin/ daac_toolkit_f77/TOOLKIT/ bin/sgi64/
DAAC	Fortran 90 or C	64-bit mode	$CUSTOM_HOME/bin/ daac_toolkit_f90/TOOLKIT/	$CUSTOM_HOME/bin/ daac_toolkit_f90/TOOLKIT/ bin/sgi64/

11.6.2.3 Setting Up the SDP Toolkit Environment

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

Assumptions:

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

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

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

- The ***ToolkitPathname*** is the home directory of the particular SDP Toolkit version being used. Refer to Table 11.6.2-1. Note that the setting of PGSHOME shown in this table may differ in your local DAAC.
- Korn shell users, type **PGSHOME=*ToolkitPathname*; export PGSHOME**, press **Return**.

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

- The ***sgiX*** is one of: **sgi** for 32-bit version of the Toolkit or **sgi64** for 64-bit version of the Toolkit. Refer to the last column of Table 11.6.2-1. for path names to the file to source.
- Korn shell users, type **. $PGSHOME/bin/*sgiX*/pgs-dev-env.ksh**, press **Return** (note the "dot" and then space at the beginning of this command).

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

- The ***aliasname*** is the name of the alias. For example, to set up an environment for the DAAC version of the Toolkit for FORTRAN 77 (or C), you might use **DAACf77** as an ***aliasname***.
- The ***ToolkitPathname*** is the home directory of the particular SDP Toolkit version being used. Refer to Table 11.6.2-1. Note that the setting of PGSHOME shown in this table may differ in your local DAAC.
- The ***sgiX*** is one of: **sgi** for 32-bit version of the Toolkit or **sgi64** for 64-bit version of the Toolkit.
- The ***textmessage*** is a message that will be echoed to the screen signifying that a new Toolkit environment has been set up. It must be enclosed within double quotes ("). An example may be, **"DAAC F77 Toolkit environment is now set."**
- A complete example (it should be all on one line in the .cshrc file):

  alias DAACf77 'setenv PGSHOME /$CUSTOM_HOME/bin/daac_toolkit_f77/TOOLKIT; source $PGSHOME/bin/sgi/pgs-dev-env.csh; echo "DAAC F77 Toolkit environment is now set" '
- Other aliases for other versions of the Toolkit can be set up similarly.

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

1 Setup for PGE07:

```
/home/emcleod/MODIS/STORE/PGE07/MOD_PR10/source
rm MOD_PR10.exe
rm *.o
setenv PGSHOME /usr/ecs/OPS/CUSTOM/daac_toolkit_f77/TOOLKIT
source $PGSHOME/bin/sgi32/pgs-dev-env.csh
source
/home/emcleod/MODIS/STORE/PGE07/MOD_PR10/source/MODIS_setup.csh.pge07
alias
n32_f77
env
make -f MOD_PR10.mk &
ls -l *exe
setenv PGS_PC_INFO_FILE
/home/emcleod/MODIS/STORE/PGE07/MOD_PR10/source/MOD_PR10.pcf
ls
MOD_PR10.exe &
confirm execution when done by looking at   file :  vi
MOD_PR10_ClopyL1BmetaToSnow.c
see if job is running:  ps -u emcleod  "time updating for MOD_PR10"
lasher{emcleod}88: ps -u emcleod
   PID TTY     TIME CMD
   267 ?       3:13 biod
 25825 pts/11  0:01 csh
 21994 pts/10  0:01 csh
 23215 pts/16  0:00 csh
 26242 pts/10  0:07 MOD_PR10.
 26089 pts/11  0:01 xedit
 26318 pts/10  0:00 ps
lasher{emcleod}105: pwd
/tmp_mnt/home/emcleod/MODIS/STORE/PGE07/MOD_PR10/source
lasher{emcleod}106: ls
MODIS_setup.csh.pge07        MOD_PR10_CopyL1BmetaToSnow.c
MODIS_setup_OPS              MOD_PR10_CopyL1BmetaToSnow.o
MOD_PR10.exe                 MOD_PR10_MakeMeta.c
MOD_PR10.h                   MOD_PR10_MakeMeta.o
MOD_PR10.mcf                 MOD_PR10_Process_Cloud.c
MOD_PR10.mk                  MOD_PR10_Process_Cloud.o
MOD_PR10.pcf                 MOD_PR10_Process_GEO.c
MOD_PR10_AAmain.c            MOD_PR10_Process_GEO.o
MOD_PR10_AAmain.o            MOD_PR10_Process_L1B.c
MOD_PR10_Compute_Snow.c      MOD_PR10_Process_L1B.o
MOD_PR10_Compute_Snow.o      MOD_PR10_Process_SnowFile.c
```

```
MOD_PR10_CopyGEOmetaToSnow.c  MOD_PR10_Process_SnowFile.o
MOD_PR10_CopyGEOmetaToSnow.o  compile_smf.csh
lasher{emcleod}107:
```

11.6.2.5 Example of a PGE Executables Tar File Insertion Script

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

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

11.6.3 Compiling Status Message Facility (SMF) Files

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

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

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

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

Assumptions:

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

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

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

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

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

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

5 At the UNIX prompt on the SGI, type **smfcompile -f *textfile*.t -lang** , press **Return**.
- The ***-lang*** is a flag that indicates for what language to compile. This flag can be one of **-c** to produce C header files, **-f77** to produce FORTRAN 77 include files,

and **-ada** to produce Ada include files. The default is for C include files. For example, type **smfcompile -f77 PGS_MODIS_39123.t**, press **Return**.

- The ***textfile*** is the file name of the SMF text file delivered with the science software.
- The SMF text files will typically have .t file name extensions.
- File names for SMF text files usually have the "seed" value used by the file as part of its file name (*e.g.* PGS_MODIS_39123.t where 39123 is the seed number).
- Only one such SMF text file can be compiled at a time; wildcards cannot be used.
- The SMF compiler may be run with the additional flags **-r** and **-i** as in, **smfcompile -f *textfile*.t -r -i**. The **-r** automatically places the runtime message file in the directory given by the environment variable PGSMSG. The **-i** automatically places the include file in the directory given by the environment variable PGSINC. For example, type **smfcompile -ada -r -i -f PGS_MODIS_39123.t**, press **Return**. Note that the **-f** flag must always be immediately followed by the name of the text file.

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

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

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

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

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

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

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

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

Assumptions:

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

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

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

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

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

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

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

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

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

- The ***ViewName*** is the name of a view allowing the make files to be accessible.

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

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

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

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

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

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

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

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

9 Build the software in accordance with instructions delivered.

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

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

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

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

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

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

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

Assumptions:

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

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

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

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

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

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

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

- The ***ToolkitPathname*** is the home directory of the desired SDP Toolkit version, in this case, a DAAC version.

- The ***sgiX*** refers to the appropriate processor. For example, type **source $PGSHOME/bin/sgi/pgs-dev-env.csh**, press **Return**.

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

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

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

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

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

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

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

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

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

9 Build the software in accordance with instructions delivered.

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

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

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

11.7 Running a PGE in a Simulated SCF Environment

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

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

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

11.7.1 Setting Up the Environment for Running the PGE

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

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

Assumptions:

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

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

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

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

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

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

- The ***PCFpathname*** is the full path name to the location of the Process Control File (PCF) to be associated with this PGE.
- The ***PCFfilename*** is the file name of the PCF.
- For example, **setenv PGS_PC_INFO_FILE /disk2/PGE32/PCF/PGE32.pcf**, press **Return**.

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

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

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

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

11.7.2 Running and Profiling the PGE

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

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

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

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

Assumptions:

1. The PGE has been built successfully with the SCF version of the SDP Toolkit .
2. The required SMF runtime message files have been produced and placed in the correct locations.
3. The Process Control File (PCF) exists and has been tailored for the DAAC environment.
4. The required environment for running the PGE has been set up.
5. The required input files are available and accessible.
6. The C shell or a derivative (*e.g.* T shell) is the current user shell.

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

1 At the UNIX prompt on the SPR SGI in the window containing the set up environment, type **cd *PGEbinPathname***, press **Return**.

- The ***PGEbinPathname*** is the full path name of the directory containing the built PGE binary executable. For example, **cd /disk3/PGE32/bin/**, press **Return**.

2• At the UNIX prompt on the SPR SGI, type **/usr/ecs/<mode>/CUSTOM/bin/DPS/E*cDpPrR*usage *PGE* >& *ResultsOut***, press **Return**.

- The ***PGE*** is the name given to the PGE binary executable.
- The ***ResultsOut*** is the file name in which to capture the profiling results as well as any messages from standard output (stdout) and standard error (stderr) that may be produced by the running PGE. Note that PGEs should *not* write to stdout or stderr.
- The **EcDpPrRusage** is the profiling program that outputs information about the runtime behavior of the PGE.
- Depending upon the PGE, it may take some time before the UNIX prompt returns.

3 At the UNIX prompt on the SPR SGI, type **echo $status**, press **Return**.

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

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

- The ***ResultsOut*** is the file name under which the profiling output was saved. Other output of the PGE may also be in this file.

- The **EcDpPrRusage** results may then be recorded and used when the PGE is registered in the PDPS.
- Any text editor/viewer may be used.

Sample of an Rusage File produced:

```
lasher{emcleod}6: more Profile.out
        #     source .cshrc
        #     cd TEST/MOD*
        #     ls
        #     /usr/ecs/OPS/CUSTOM/bin/DPS/EcDpPrRusage MOD_PR10.exe > Profile.out
        lasher{emcleod}9: more profile.out
        # Resource Usage Information
        COMMAND=MOD_PR10.exe
        EXIT_STATUS=0
        ELAPSED_TIME=233.583145
        USER_TIME=10.046158
        SYSTEM_TIME=7.555547
        MAXIMUM_RESIDENT_SET_SIZE=4080
        AVERAGE_SHARED_TEXT_SIZE=0
        AVERAGE_UNSHARED_DATA_SIZE=0
        AVERAGE_UNSHARED_STACK_SIZE=0
        PAGE_RECLAIMS=151
        PAGE_FAULTS=0
        SWAPS=0
        BLOCK_INPUT_OPERATIONS=2
        BLOCK_OUTPUT_OPERATIONS=2710
        MESSAGES_SENT=0
        MESSAGES_RECEIVED=0
        SIGNALS_RECEIVED=0
        VOLUNTARY_CONTEXT_SWITCHES=1095
        INVOLUNTARY_CONTEXT_SWITCHES=2
        lasher{emcleod}10:
```

11.8 File Comparison and Data Visualization

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

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

11.8.1 Using the GUI HDF File Comparison GUI

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

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

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

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

The HDF File Comparison GUI window will be displayed.

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

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

11.8.2 Using the hdiff HDF File Comparison Tool

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

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

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

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

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

The HDF File Comparison Tool window will be displayed.

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

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

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

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

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

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

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

11.8.3 Using the ASCII File Comparison Tool

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

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

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

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

Comparing Two ASCII Files

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

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

3• In xterm window at the prompt **2nd file to compare?**, type ***filename2***, then press the **Enter** key.
The ***filename2*** is the file name of the second of two ASCII files to be compared. Select another studentÕs corresponding file.
If ***filename2*** is not in the current directory (the directory from which the SSIT Manager was run), include the full path name with the file name.
A window labeled **xdiff** will be displayed.

4 In the window labeled **xdiff**, view the differences between the two files displayed.
File ***filename1*** will be displayed on the left side of the window. File ***filename2*** will be displayed on the right.
Only sections of file in which there are differences will be displayed. A "bang" character (!) at the beginning of a line indicates that a difference was found.
For further help on *xdiff*, type **man xdiff**, in an xterm window then press the **Enter** key.
Close the display window by using the pull down menu from the X window in the upper left corner.

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

11.8.4 Using the Binary File Difference Assistant

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

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

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

1. The SSIT Manager is running.

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

Comparing two Binary Files

1• From the SSIT Manager, select **Tools→ Product Examination → File Comparison → Binary** from the menu.
The Binary File Difference Assistant tool GUI will be displayed.

2• In the Binary File Difference Assistant tool GUI, click on one of the languages listed under the **Select Language** label. The choices are C, FORTRAN, or IDL.
The choice of language depends largely on preference. It does not necessarily have to be the language that was used to create the files being compared.

3• Optionally, click on either the **Image** button or the **Structure** button located under the label **Compare Function**.
Clicking on the **Image** button will display a code example for comparing binary files containing images.
Clicking on the **Structure** button will display a code example for comparing binary files containing structures or records.
The displayed listing well documented and should be read.
The language of the code will depend on the language selection made in step 2.

4• Optionally, click on either the **Image** button or the **Structure** button located under the label **Driver**.
Clicking on the **Image** button will display a code example for a driver invoking the compare function for binary files containing images.
Clicking on the **Structure** button will display a code example for a driver invoking the compare function for binary files containing structures or records.
The displayed listing well documented and should be read.
The language of the code will depend on the language selection made in step 2.

5 Optionally, click on either the **Help** button.
A Help window will be displayed.
To end help, click on the **Dismiss** button.
The Help window may remain displayed while using the Binary File Difference Assistant.

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

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

  C:

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

FORTRAN:

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

IDL:

DaacBinDiff_***fileID***.pro	Compare function
DaacBinDiff_***fileID***_driver.pro	Driver
DaacBinDiff_***fileID***.sh	Shell script with here document

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

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

11.8.5 Data Visualization

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

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

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

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

11.8.5.1 Data Visualization with EOSView

11.8.5.1.1 Viewing Product Metadata with the EOSView Tool

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

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

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

1 Telnet into **calahans.**

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

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

4 **Select EOSView, The EOSView GUI will be displayed.**

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

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

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

Viewing Product Metadata

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

- The EOSView GUI will be displayed.

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

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

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

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

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

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

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

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

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

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

11.8.5.1.2 Viewing HDF Image Objects

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

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

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

Viewing HDF Image Objects

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

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

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

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

5• Optional panning while zooming. In the GUI labeled **EOSView - Image Display Window - *MyImageObject***, click on the **Options** menu, then select **Pan Window**, a thumbnail representation of the entire Image will be displayed in the subwindow labeled

Pan Window. The portion of the zoomed Image shown in the main window will be the portion indicated by the hollow rectangle on the thumbnail image. Use the mouse left button to click and drag the rectangle to a new location on the thumbnail image.
The panning option may be repeated as desired.

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

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

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

11.8.5.1.3 Viewing HDF-EOS Grid Objects

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

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

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

Viewing HDF-EOS Grid Objects

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

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

3 In the GUI labeled **EOSView - Grid Select**, click on the **Data Fields** checkbox and then click on the **OK** button. Then double click on one of the data fields listed.

- A GUI labeled **EOSView - Grid - *GridObjectName* - Start/Stride/Edge** will be displayed where *GridObjectName* will be replaced by the name of the Grid object selected in step 1.

4 To display the data in the form of a table of values, in the GUI labeled **EOSView - Grid - *GridObjectName* - Start/Stride/Edge**, click on the checkboxes for both **YDim** and **XDim** and then click on the **OK** button.

- A GUI labeled ***MyDataField*** will be displayed where *MyDataField* will be replaced by the name of the data field selected in step 2.

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

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

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

11.8.5.1.4 Viewing HDF-EOS Swath Objects

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

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

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

Viewing HDF-EOS Swath Objects

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

2• In the GUI labeled **EOSView - *MyOutputFile.hdf***, double click on a Swath object listed for which data is to be inspected.
A GUI labeled **EOSView - Swath Select** will be displayed.
Information on **Dimensions, Geolocation Mappings, Indexed Mappings, Geolocation Fields**, **Attributes** for the selected Swath Object can be displayed by clicking on the corresponding checkboxes.

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

4 To display the data in the form of a table of values, in the GUI labeled **EOSView - Swath - *SwathObjectName* - Start/Stride/Edge**, click on the checkboxes for both **ScanLineTra** and **PixelsXtrac** and then click on the **OK** button.

5• To display the data field in image form, in the GUI labeled ***MyDataField***, click on the **File** menu and then select **Make Image**.

A GUI labeled **EOSView - Swath/Grid Image** will appear.

6• Optional colorization, zooming, panning while zooming features can be used in the GUI labeled **EOSView - Swath/Grid Image** to obtain your desired image.

7• To end the session with displaying Swath object, in the GUI labeled **EOSView - Swath/Grid**, click on the **File → Close**.

11.8.5.1.5 Viewing HDF SDS Objects

This procedure describes how to use the EOSView tool to view science data in the HDF-EOS output file that are in HDF SDS (standard HDF science data set) format. To view an HDF SDS object with the EOSView tool, execute the procedure steps that follow:

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

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

Viewing HDF SDS Objects

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

The EOSView GUI will be displayed.

2• In the GUI labeled **EOSView - *MyOutputFile.hdf***, double click on a SDS object listed for which data is to be inspected.

A GUI labeled **EOSView - Multi-Dimension SDS** will be displayed.

A number of checkboxes will be displayed, one for each of the dimensions in the selected SDS (there will be at least two, an X and a Y).

3• In the GUI labeled **EOSView - Multi-Dimension SDS**, click on two of the dimension checkboxes and then click on the **Table** button. Then double click on one of the data fields listed.

A GUI labeled ***MySDS*** will be displayed where *MySDS* will be replaced by the name of the SDS object selected in step 1.

4• To display the data field in image form, in the GUI labeled ***MySDS***, click on the **File** menu and then select **Make Image**.

A GUI labeled EOSView - Image Display Window - *MySDS* will appear,

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

6• To end the session with displaying Swath object, in the GUI labeled **EOSView - Image Display Window - *MySDS***, select **File → Close** from the menu.

- The **EOSView - Image Display Window - *MySDS*** GUI will disappear.

11.8.5.2 Data Visualization with the IDL Tool

11.8.5.2.1 Viewing Product Data with the IDL Tool

The following procedures describe how to use the IDL (Interactive Data Language) COTS tool to inspect the data in the output file from a PGE. These procedures are geared toward binary and ASCII formats, but can be extended to other formats supported by IDL including HDF, NetCDF, and PGE. Consult the IDL references for details on these other formats.

The major activities addresses here include creating an image display, saving an image display, creating a plot display, and saving a plot display.

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

1. The SSIT Manager is running.
2. The output file is binary, ASCII, or one of the other IDL supported data formats.
3. IDL has been properly installed and is accessible to the user.

Viewing Product Data with the IDL Tool

1• From the SSIT Manager, select **Tools→ Product Examination → IDL** from the menu.
An xterm (on the AIT Sun) will be displayed within which the IDL command interpreter will be run.

2 Select the procedure depending upon the activity to perform.

3 To end the IDL session, close any display windows remaining, then at the IDL prompt type **quit**, then press the **Enter** key.
The IDL session will be closed.

11.8.5.2.2 Creating an Image Display Using IDL

The following procedure describes how to use the IDL Tool to create an image display.

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

1. The SSIT Manager is running.
2. The PGE output file to be examined is of an IDL-supported type/format (if in doubt, consult the IDL Reference Guide

3. IDL has been properly installed and is accessible to the user.
4. For binary files, data is assumed to be 8-bit characters

Creating an Image Display Using the IDL Tool - Binary Data

1 From the SSIT Manager, select **Tools→ Product Examination → IDL** from the menu.
An xterm (on the AIT Sun) will be displayed within which the IDL command interpreter will be run.

2 At the IDL prompt, type ***OPENR,1,'MyBinaryFilename'***, press the **Enter** key.
The ***MyBinaryFilename*** is the full path name and file name of the binary data file of known dimensions to read in.
The single quotes (') must be included around the path/file name.
The **1** is the logical unit number.

3• At the IDL prompt, type ***MyImage=BYTARR(dim1, dim2)***, press the **Enter** key.
The ***MyImage*** is the name to be given to the image once created.
The ***dim1*** and ***dim2*** are the dimensions of the input data.

4 At the IDL prompt, type ***READU,1,MyImage***, press the **Enter** key.

5 At the IDL prompt, type ***TV,MyImage***, press the **Enter** key.
The image, ***MyImage***, should then be displayed.

6 At the IDL prompt, type ***LOADCT,3***, press the **Enter** key.
This command loads color table number 3. Other color tables are available

7• At the IDL prompt, type ***CLOSE,1***, press the **Enter** key.
This closes logical unit 1.
Always close logical units or an error will result the next time an access is attempted.

Creating an Image Display Using the IDL Tool - ASCII Data

1 From the SSIT Manager, select **Tools→ Product Examination → IDL** from the menu.
An xterm (on the AIT Sun) will be displayed within which the IDL command interpreter will be run.

2 At the IDL prompt, type ***OPENR,1,('MyASCIIfilename')***, then press the **Enter** key.

The **MyASCIIfilename** is the full path name and file name of the ASCII data file of known dimensions to read in.
The single quotes (') must be included around the path/file name.
The **1** is the logical unit number.

3 At the IDL prompt, type ***MyImage=BYTARR(dim1,dim2)***, then press the **Enter** key.

The ***MyImage*** is the name to be given to the image once created.
The ***dim1*** and ***dim2*** are the dimensions of the input data.

4 At the IDL prompt, type ***READF,1,MyImage***, then press the **Enter** key.

5• At the IDL prompt, type ***TV,MyImage***, then press the **Enter** key.

The image, **MyImage**, is displayed.

6 At the IDL prompt, type ***LOADCT,3***, then press the **Enter** key.

This command loads color table number 3. Other color tables are available; refer to the IDL Reference Guide for more details.

7 At the IDL prompt, type ***CLOSE,1***, then press the **Enter** key.

This closes logical unit 1.
Always close logical units or an error will result the next time an access is attempted.

Creating an Image Display Using the IDL Tool - PGM Data:

---

1 From the SSIT Manager, select **Tools→ Product Examination → IDL** from the menu.
An xterm (on the AIT Sun) will be displayed within which the IDL command interpreter will be run.

2• At the IDL prompt, type ***READ_PPM,"MyPGMfilename",MyImage,r,g,b***, then press the **Enter** key.
The MyPGMfilename is the full path name and file name of the PGM formatted data file.
The double quotes (") must be included around the path/file name.
The MyImage is the name to be given to the image created.

3 At the IDL prompt, type ***TVLCT,r,g,b***, then press the **Enter** key.
Note that r,g,b color table syntax is used for most formatted file types in IDL.

4 At the IDL prompt, type ***TV,MyImage***, then press the **Enter** key.
The image, **MyImage**, should then be displayed.

11.8.5.2.3 Saving an Image Display Using IDL

The next procedure describes how to save an image display (once created) to either a data file or a graphic file.

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

1. The SSIT Manager is running, IDL is running
2. The PGE output file to be examined is of an IDL-supported type/format (if in doubt, consult the IDL Reference Guide
3. For binary files, data is assumed to be 8-bit characters
4. The image display is to be saved in a binary (8-bit) or ASCII (comma-delimited characters) format.

Save an image display using IDL - Binary Data

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

An xterm (on the AIT Sun) will be displayed within which the IDL command interpreter will be run.

2• At the IDL prompt, type ***OPENW,1,('MyBinaryFilename.bin'***), then press the **Enter** key.

The ***MyBinaryFilename.bin*** is the full path name and file name of the binary data file to write out.

The single quotes (') must be included around the path/file name.

The **1** is the logical unit number.

3 At the IDL prompt, type ***WRITEU,1,MyImage***, then press the **Enter** key.

The ***MyImage*** is the name of the image to save.

4 At the IDL prompt, type ***CLOSE,1***, then press the **Enter** key.

This closes logical unit 1.

Always close logical units or an error will result the next time an access is attempted.

Save an image display using IDL - ASCII Data

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

An xterm (on the AIT Sun) will be displayed within which the IDL command interpreter will be run.

2 At the IDL prompt, type ***OPENW,1,('MyASCIIfilename.asc')***, then press the **Enter** key.

The ***MyASCIIfilename.asc*** is the full path name and file name of the binary data file to write out.

The single quotes (') must be included around the path/file name.

The **1** is the logical unit number.

3 At the IDL prompt, type ***PRINTF,1,MyImage***, then press the **Enter** key.

The **MyImage** is the name of the image to save.

4 At the IDL prompt, type **CLOSE,1**, then press the **Enter** key.

This closes logical unit 1.

Always close logical units or an error will result the next time an access is attempted.

Save an image display using JPEG Data

1• At the IDL prompt, type **WRITE_JPEG,*"MyJPEGfilename.jpg",MyImage***, press **Return**.

- The ***MyJPEGfilename.jpg*** is the full path name and file name of the JPEG data file to write out.

Creating a Plot Display Using IDL

The procedures for creating a plot display are clearly described in the IDL manuals; some exceptions are clarified below.

Setting axis limits for a plot:

1 At the IDL prompt, type **SURFACE,MyPlot,AX=70,AZ=70,xrange=[0,20],yrange=[0,20]zrange=[0,30]**, and press **Return**.

- The ***MyPlot*** is the IDL session variable (to which you have assigned some math function, program output, image, etc.).
- ***AX*** sets the displayed rotation about the X axis.
- ***AZ*** sets the displayed rotation about the Z axis.
- The values of ***xrange*** set the displayed portion of the X axis.
- The values of ***yrange*** set the displayed portion of the Y axis.
- The values of ***zrange*** set the displayed portion of the Z axis.
- The plot will then be displayed to the screen.

Setting axis titles for a plot:

1 At the IDL prompt, type **SURFACE,MyPlot,AX=70,AZ=70,xtitle='this is X', ytitle='this is Y',ztitle='this is Z'**, and press **Return**.

- The ***MyPlot*** is the IDL session variable (to which you have assigned some math function, program output, image, etc.).
- The value of ***xtitle*** sets the displayed title of the X axis.
- The value of ***ytitle*** sets the displayed title of the Y axis.
- The value of ***ztitle*** sets the displayed title of the Z axis.
- The plot will then be displayed to the screen.

Saving a Plot Display Using IDL

Saving a displayed plot to a permanent file:

1• At the IDL prompt, type **MyPlotDisplay=SURFACE,MyPlot,AX=80,AZ=20**, and press **Return**.

- The ***MyPlotDisplay*** is session name for the displayed plot of ***MyPlot***.
- The ***MyPlot*** is the IDL session variable (to which you have assigned some math function, program output, image, etc.).

2 At the IDL prompt, type **SAVE,MyPlotDisplay,4,'MyPlotOutput.ps'**, press **Return**.

- The ***MyPlotDisplay*** is the session name of the plot display .
- The ***MyPlotOutput.ps*** is the desired name for the saved file.
- The SAVE option number 4 sets the output file type to PostScript (ps). There are other options, of course (consult the IDL manuals).

11.8.5.2.4 Raster Mapping Fundamentals

This procedure describes how to use the IDL Tool to perform basic raster mapping functions. These are spatial functions involving map projections, but do not include surface modeling (also called “2.5D”) or two-dimensional spectral functions.

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

1. The SSIT Manager is running.
2. IDL is running

Raster Mapping - Global Data Set Image

1 From the SSIT Manager, select **Tools→ Product Examination → IDL** from the menu.
An xterm (on the AIT Sun) will be displayed within which the IDL command interpreter will be run.

2• At the IDL prompt, type ***TV,MyImage***, then press the **Enter** key.
The ***MyImage*** is the image name of the global image data set.
The image, ***MyImage***, should then be displayed.

3• At the IDL prompt, type ***MAP_SET,/ORTHOGRAPHIC***, then press the **Enter** key.
IDL also supports other map projections. Refer to IDL Reference Guide.

4• At the IDL prompt, type ***MyNewImage=MAP_IMAGE(MyImage,startx,starty,/BILIN)***, then press the **Enter** key.
The ***MyNewImage*** is the name to assign to the resulting image.
The ***MyImage*** is the name of the original global image data set.

5• At the IDL prompt, type ***TV,MyNewImage,startx,starty***, then press the **Enter** key.
The image ***MyNewImage*** should then be displayed.

6• Optional overlay Lat/Long. At the IDL prompt, type ***MAP_GRID***, then press the **Enter** key.
This overlays Lat/Long graticule onto ***MyNewImage***.

7• Optional overlay world coastlines. At the IDL prompt, type ***MAP_CONTINENTS***, press the **Enter** key.
This overlays world coastlines onto **MyNewImage**.

For a sub-global data set image, one having geocentric-LLR coordinates defined for subintervals of longitude and latitude (e.g. from -88 to -77 degrees East Longitude and 23 to 32 degrees North Latitude.

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

1. The SSIT Manager is running.
2. IDL is running

Raster Mapping - Sub-Global Data Set Image

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

An xterm (on the AIT Sun) will be displayed within which the IDL command interpreter will be run.

2 At the IDL prompt, type ***TV,MyImage***, then press the **Enter** key.

The ***MyImage*** is the image name of the sub-global image data set.

The image, ***MyImage***, should then be displayed.

3• At the IDL prompt, type ***MAP_SET,/MERCATOR,LIMIT=[lat1,lon1,lat2,lon2]***, then press the **Enter** key.

The ***lat1***, ***lon1***, ***lat2***, and ***lon2*** specify the latitude and longitude intervals of the sub-global image data set.

4• At the IDL prompt, type ***MyNewImage=MAP_IMAGE(MyImage,startx,starty,/BILIN.LATMIN=lat1,LATMAX =lat2,LONMIN=lon1,LONMAX=lon2)***, then press the **Enter** key.

The ***MyNewImage*** is the name to assign to the resulting image.

The ***MyImage*** is the name of the original global image data set.

The ***lat1***, ***lon1***, ***lat2***, and ***lon2*** specify the latitude and longitude intervals of the sub-global image data set.

5 At the IDL prompt, type ***TV,MyNewImage,startx,starty***, then press the **Enter** key.

The image ***MyNewImage*** should then be displayed.

6• Optional overlay Lat/Long. At the IDL prompt, type ***MAP_GRID***, then press the **Enter** key.

This overlays Lat/Long graticule onto ***MyNewImage***.

7• Optional overlay world coastlines. At the IDL prompt, type ***MAP_CONTINENTS***, then press the **Enter** key.

This overlays world coastlines onto ***MyNewImage***.

11.9 Science Software Integration and Test (SSI&T) Release RELEASE 4.

The process of SSI&T or integration of EOS Instrument Science Software into the ECS has been developed and refined over three iterations of ECS. IR1, the Pre-Release B Testbed and Release RELEASE 4. The latter will be the at-launch system supporting EOS-AM1 instruments. Although every attempt has been made to keep integration procedures for science software consistent through succeeding releases, basis architectural differences have led to significant variances. This section describes the architecture of the last iterations of ECS.

11.9.1 RELEASE 4 Architecture: Overview

The Release RELEASE 4 architecture can be grouped into the following four categories:

- Data storage and management is provided by the Data Server Subsystem (DSS), with the functions needed to archive science data, search for and retrieve archived data, manage the archives, and stage data resources needed as input to science software or resulting as output from their execution. The Data Server Subsystem provides access to earth science data in an integrated fashion through an Application Programming Interface that is common to all layers.
- Information search and data retrieval is provided by the science user interface functions in the Client Subsystem (CLS), by information search support functions in the Data Management Subsystem (DMS), and by capabilities in the Interoperability Subsystem (IOS) which assist users in locating services and data.
- Data processing is provided by the Data Processing Subsystem (DPS) for the science software; and by capabilities for long and short term planning of science data processing, as well as by management of the production environment provided by the Planning Subsystem (PLS). Routine data processing and re-processing will occur in accordance with the established production plans. In addition ECS will provide “on-demand processing”, where higher level products are produced only when there is explicit demand for their creation.
- Data ingest is provided by the Ingest Subsystem (INS), which interfaces with external applications and provides data staging capabilities and storage for an approximately 1-year buffer of Level 0 data (so that reprocessing can be serviced from local storage).The number of external interfaces which ECS will have is potentially very large, and the interfaces can serve very diverse functions, such as high-volume ingest of level 0 data and low-volume ingest of data from field campaigns.

Table 11.9.3-1 provides procedural differences: Testbed to RELEASE 4.

11.9.1.1 ECS Subsystems

The following sub-sections provide brief overviews for each of these subsystems. More detailed discussions of their design breakdown can be found in 305-CD-020-002.

11.9.1.2 Client Subsystem (CLS)

The Client provides users with an interface through which they can access ECS services and data. It also gives science software access to the ECS services, as well as direct access to ECS data. Access is provided through graphic user interface (GUI) application tools for displaying the various kinds of ECS data (e.g., images, documents, tables), and libraries representing the client APIs to ECS services. The client subsystem follows an object oriented design. The design is built around a core set of ‘root’ objects from which all other software will inherit its behavior.

11.9.1.3 Interoperability Subsystem (IOS)

The Interoperability subsystem provides an advertising service. It maintains a database of information about the services and data offered by ECS, and provides interfaces for searching this database and for browsing through related information items. For example, ESDTs are made visible through the advertising service. The Client Subsystem provides the user interface which enables access to the IOS.

11.9.1.4 Data Management Subsystem (DMS)

The Data Management subsystem provides three main functions:

- Provide end-users with a consolidated logical view of a distributed set of data repositories.
- Allow end-users to obtain descriptions for the data offered by these repositories. This also includes descriptions of attributes about the data and the valid values for those attributes.
- Provide data search and access gateways between ECS and external information systems.

11.9.1.5 Data Server Subsystem (DSS)

The Data Server subsystem provides the management, cataloging, access, physical storage, distribution functions for the ECS earth science data repositories, consisting of science data and their documentation. The Data Server provides interfaces for other ECS subsystems which require access to data server services. The Data Server Subsystem consists of the following principal design components:

- Database Management System - The Data Server subsystem will use database technology to manage its catalog of earth science data, and for the persistence of its system administrative and operational data.
- Document Management System - Web server and database technology are used to implement a document management system to provide storage and information retrieval for guide documents, science software documentation, and ECS earth science related documents.
- Data Type Libraries - The Data Server will use custom dynamic linked libraries (DLLs) to provide an extensible means of implementing the variety of ECS earth science data types and services, and will provide a consistent interface for use by other ECS subsystems requiring access to those services and data.
- File Storage Management System - This component provides archival and staging storage for data.
- Distribution System - The Data Server provides the capabilities needed to distribute bulk data via electronic file transfer or physical media.

11.9.1.6 Ingest Subsystem (INS)

This subsystem deals with the initial reception of all data received at an EOSDIS facility and triggers subsequent archiving and processing of the data. The ingest subsystem is organized into a collection of software components (e.g., ingest management software, translation tools, media handling software) from which those required in a specific situation can be readily configured. The resultant configuration is called an ingest client. Ingest clients can operate on a continuous basis to serve a routine external interface; or they may exist only for the duration of a specific ad-hoc ingest task. The ingest subsystem also standardizes on a number of possible application protocols for negotiating an ingest operation, either in response to an external notification, or by polling known data locations for requests and data.

11.9.1.7 Data Processing Subsystem (DPS)

The main components of the data processing subsystem - the science algorithms or Product Generation Executives (PGEs) - will be provided by the science teams. The data processing subsystem provides the necessary hardware resources, as well as a software environment for queuing, dispatching and managing the execution of these algorithms. The processing environment will be highly distributed and will consist of heterogeneous computing platforms. The AutoSys COTS tool is used as the scheduling engine. The tool is designed to manage production in a distributed UNIX environment. The DPS also interacts with the DSS to cause the staging and de-staging of data resources in synchronization with processing requirements.

11.9.1.8 Planning Subsystem (PLS)

The Planning Subsystem provides the functions needed to plan routine data processing, schedule on-demand processing, and dispatch and manage processing requests. The subsystem provides access to the data production schedules at each site, and provides management functions for handling deviations from the schedule to operations and science users. The Planning subsystem provides several functions to account for:

- a processing environment which eventually will be highly distributed and consist of heterogeneous computing platforms
- existence of inter-site and external data dependencies
- dynamic nature of the data and processing requirements of science algorithms
- need for high availability
- providing a resource scheduling function which can accommodate hardware technology upgrades
- support for on-demand processing (as an alternative to predominantly routine processing)
- ability to provide longer-term (e.g., monthly) processing predictions as well as short term (e.g., daily) planning and scheduling

11.9.1.9 Communications Subsystem (CSS)

The CSS helps manage the operation of distributed objects in ECS, by providing a communications environment. The environment allows software objects to communicate with each other reliably, synchronously as well as asynchronously, via interfaces that make the location of a software object and the specifics of the communications mechanisms transparent to the application.

In addition, CSS provides the infra-structural services for the distributed object environment. They are based on the Distributed Computing Environment (DCE) from the Open Software Foundation (OSF). DCE includes a number of basic services needed to develop distributed applications, such as remote procedure calls (rpc), distributed file services (DFS), directory and naming services, security services, and time services.

Finally, CSS provides a set of common facilities, which include legacy communications services required within the ECS infrastructure and at the external interfaces for file transfer, electronic mail, bulletin board and remote terminal support. The Object Services support all ECS applications with inter-process communication and specialized infra- structural services such as security, directory, time, asynchronous message passing, event logging, lifecycle service, transaction processing and World Wide Web (WWW) service.

11.9.1.10 Management Subsystem (MSS)

The Management Subsystem (MSS) provides enterprise management (network and system management) for all ECS resources: commercial hardware (including computers, peripherals, and network routing devices), commercial software, and custom applications. With few exceptions, the management services will be fully decentralized, such that no single point of failure exists.

MSS provides two levels of an ECS management view: the local (site/DAAC specific) view, provided by Local System Management (LSM), and the enterprise view, provided by the Enterprise Monitoring and Coordination (EMC) at the SMC. Enterprise management relies on the collection of information about the managed resources, and the ability to send notifications to those resources. For network devices, computing platforms, and some commercial of the shelf software, MSS relies on software called “agents” which are usually located on the same device/platform and interact with the device’s or platform’s control and application software, or the commercial software product. However, a large portion of the ECS applications software is custom developed, and some of this software - the science software - is externally supplied. For these components, MSS provides a set of interfaces via which these components can provide information to MSS (e.g., about events which are of interest to system management such as the receipt of a user request or the detection of a software failure). These interfaces also allow applications to accept commands from MSS, provided to MSS from M&O consoles (e.g., an instruction to shut down a particular component). Applications which do not interact with MSS directly will be monitored by software which acts as their “proxies”. For example, the Data Processing Subsystem (DPS) acts as the proxy for the science software it executes. DPS notifies MSS of events such as the dispatching or completion of a PGE, or its abnormal termination.

MSS uses HP OpenView as the centerpiece of its system management solution. The information collected via the MSS interfaces from the various ECS resources is consolidated into an event history database, some on a near real-time basis, some on a regular polling basis (every 15-to 30 minutes) as well as on demand, when necessitated by an operator inquiry. The database is managed by Sybase, and Sybase query and report writing capabilities will be used to extract regular and ad-hoc reports from it. Extracts and summaries of this information will be further consolidated on a system wide basis by forwarding it to the SMC (also on a regular basis).

MSS provides fault and performance management and other general system management functions such as security management (providing administration of identifications, passwords, and profiles); configuration management for ECS software, hardware, and documents; Billing and Accounting; report generation; trending; request tracking; and mode management (operational, test, simulation, etc.).

11.9.1.11 Internetworking Subsystem (ISS)

The ISS provides local area networking (LAN) services at ECS installations to interconnect and transport data among ECS resources. The ISS includes all components associated with LAN services including routing, switching, and cabling as well as network interface units and communications protocols within ECS resources.

The ISS also provides access services to link the ECS LAN services to Government-furnished wide-area networks (WANs), point-to-point links and institutional network services. Examples include the NASA Science Internet (NSI), Program Support Communications Network (PSCN), and various campus networks “adjoining” ECS installations.

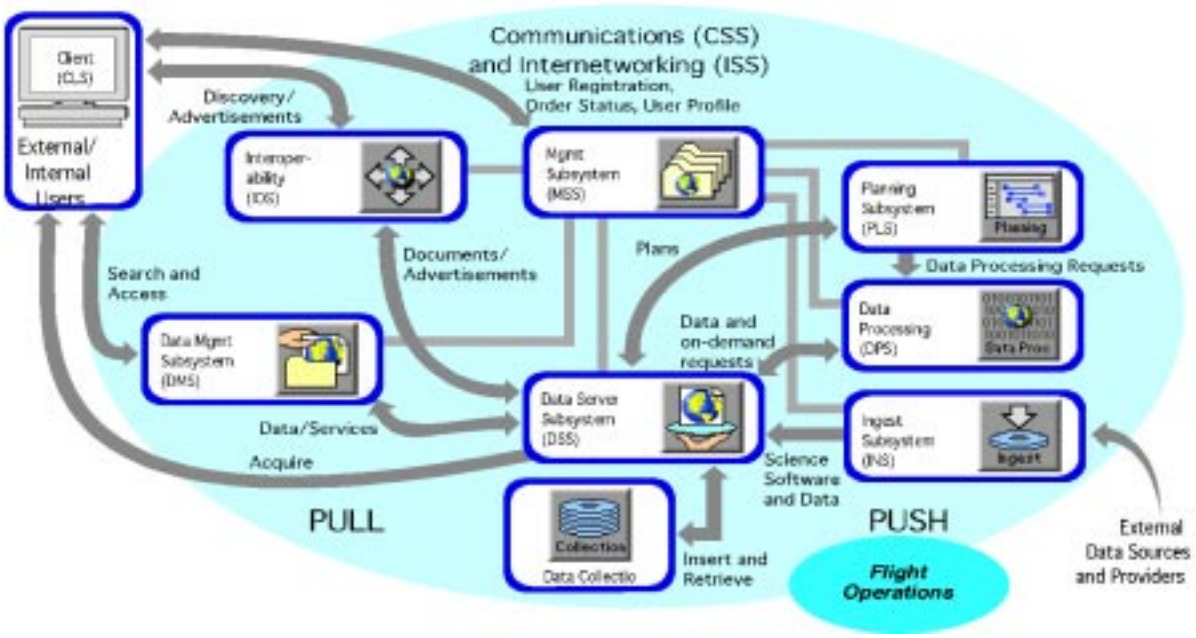


Figure 11.9.1-1. ECS Communications /Interetworking Subsystems

11.9.2 Implications for SSI&T Functions.

Table 11.9.3-1 . List the major Functions that will be encountered by SSI&T staff. These architectural functions characterize the RELEASE 4 systems. The major functions are due to the presence of Ingest and Data Server Subsystems in RELEASE 4.

Table 11.9.3.1. Major SSI&T Functions within VERSION 2.0

Function		**Release RELEASE 4**
System Operation		All servers must run and communicate with each other; bring up manually, or use ECS Assist tool.
Ingest Ancillary Data Granules		Ingest GUI, ESDTs must be visible to ADV server.
ESDT Insert		Use Ingest
ESDT Verification		verify through ADV
DAP, SSAP Insert		Use Ingest
PDPS Database Population		More attributes, production rules
PGE Operation		When all data is available; DPR activated. No automatic reprocessing Complex chaining through production rules.
File Access		verify presence through ADV; ftp from SDSRV; access to multiple sites
Multi-file Granule Support		Files inserted together, accessed as a single granule.
Subscription Management		Subscription Manager

11.10 Using ECS Assistant to View ECS Science Data Server Database

ESDTs and their granules stored in the archive are managed using an ECS Science Data server database. ECS Assistant provides an easy way to review the records stored in this database by using the ECS Assistant DB Viewer. There are two main windows in the DB Viewer. The first is called Collections and is used to display ESDT information included in the Collection database table. Information listed in this table includes ESDT short names, times last updated, types, etc. If an ESDT is added to the Science Data Server, its record will be shown in this window. The other window is called Granules and is used to display information included in the Granule database table. If a granule is inserted for an ESDT, the granule information will be listed in this window if its ESDT is highlighted in the Collection window. In addition to these two main windows, this DB Viewer GUI can also show ESDT database validation rules, Product Specific Attributes, (PSA) information, and summary information about the database reviewed.

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

Assumptions:

1. The ECS Assistant has been properly installed.
2. The ECS Subsystem Manager is running.
3. The environment variables for using the database have been set correctly.

11.10.1 Starting ECS monitor GUI:

1 Invoke the ECS Assistant GUI.

- The ECS Assistant GUI will be launched.

2 At the ECS Assistant GUI, select ESDT Manager GUI by clicking the ESDT Manager.

- The ESDT manager GUI will appear.

3 At the ECS ESDT Manager GUI, select the DB Viewer by clicking the **DB Viewer** button.

- The Database Login GUI will appear as shown in Figure 11.10-1.
- Fill in the fields to point to the specific database for the mode used.
- Click Login to open the DB Viewer.
- The DB Viewer GUI will appear as shown in Figure 11.-10-2
- ESDTs are listed in the Collections window.

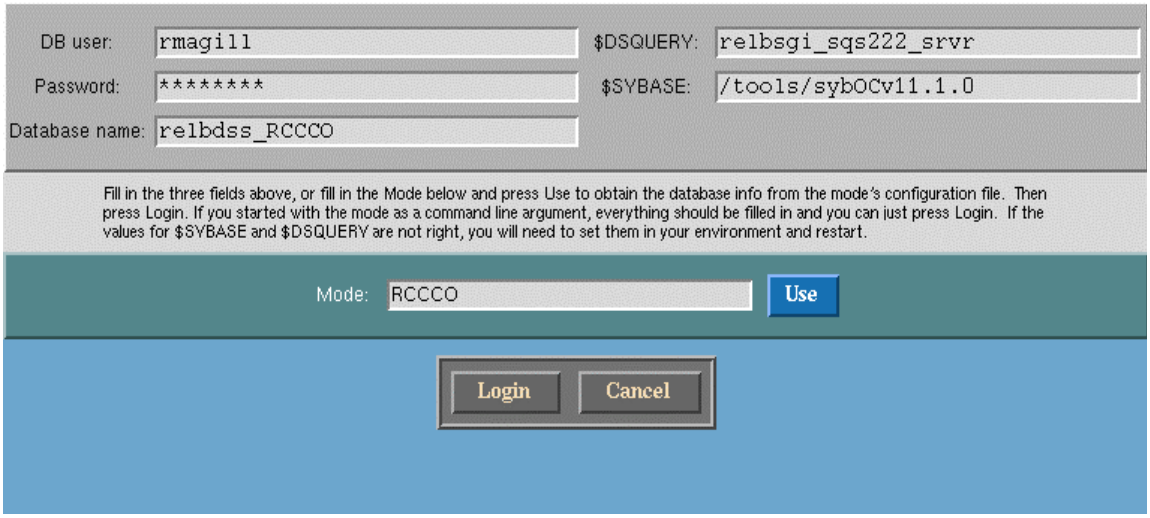
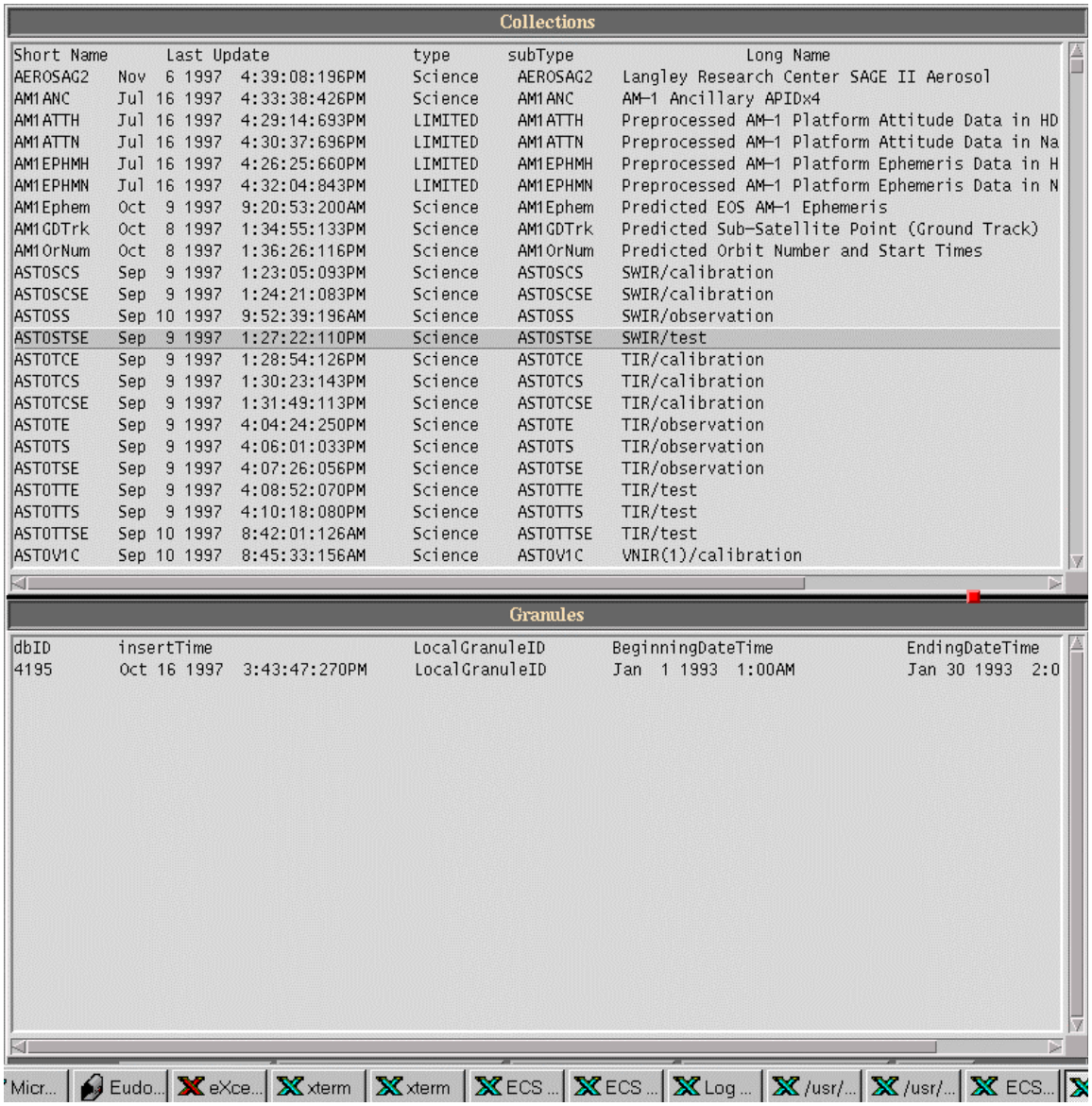


Figure 11.10-1. Database Login GUI



Collections

Short Name	Last Update	type	subType	Long Name
AEROSAG2	Nov 6 1997 4:39:08:196PM	Science	AEROSAG2	Langley Research Center SAGE II Aerosol
AM1ANC	Jul 16 1997 4:33:38:426PM	Science	AM1ANC	AM-1 Ancillary APIDx4
AM1ATTH	Jul 16 1997 4:29:14:693PM	LIMITED	AM1ATTH	Preprocessed AM-1 Platform Attitude Data in HD
AM1ATTN	Jul 16 1997 4:30:37:696PM	LIMITED	AM1ATTN	Preprocessed AM-1 Platform Attitude Data in Na
AM1EPHMH	Jul 16 1997 4:26:25:660PM	LIMITED	AM1EPHMH	Preprocessed AM-1 Platform Ephemeris Data in H
AM1EPHMN	Jul 16 1997 4:32:04:843PM	LIMITED	AM1EPHMN	Preprocessed AM-1 Platform Ephemeris Data in N
AM1Ephem	Oct 9 1997 9:20:53:200AM	Science	AM1Ephem	Predicted EOS AM-1 Ephemeris
AM1GDTrk	Oct 8 1997 1:34:55:133PM	Science	AM1GDTrk	Predicted Sub-Satellite Point (Ground Track)
AM1OrNum	Oct 8 1997 1:36:26:116PM	Science	AM1OrNum	Predicted Orbit Number and Start Times
ASTOSCS	Sep 9 1997 1:23:05:093PM	Science	ASTOSCS	SWIR/calibration
ASTOSCSE	Sep 9 1997 1:24:21:083PM	Science	ASTOSCSE	SWIR/calibration
ASTOSS	Sep 10 1997 9:52:39:196AM	Science	ASTOSS	SWIR/observation
ASTOSTSE	Sep 9 1997 1:27:22:110PM	Science	ASTOSTSE	SWIR/test
ASTOTCE	Sep 9 1997 1:28:54:126PM	Science	ASTOTCE	TIR/calibration
ASTOTCS	Sep 9 1997 1:30:23:143PM	Science	ASTOTCS	TIR/calibration
ASTOTCSE	Sep 9 1997 1:31:49:113PM	Science	ASTOTCSE	TIR/calibration
ASTOTE	Sep 9 1997 4:04:24:250PM	Science	ASTOTE	TIR/observation
ASTOTS	Sep 9 1997 4:06:01:033PM	Science	ASTOTS	TIR/observation
ASTOTSE	Sep 9 1997 4:07:26:056PM	Science	ASTOTSE	TIR/observation
ASTOTTE	Sep 9 1997 4:08:52:070PM	Science	ASTOTTE	TIR/test
ASTOTTS	Sep 9 1997 4:10:18:080PM	Science	ASTOTTS	TIR/test
ASTOTTSE	Sep 10 1997 8:42:01:126AM	Science	ASTOTTSE	TIR/test
ASTOV1C	Sep 10 1997 8:45:33:156AM	Science	ASTOV1C	VNIR(1)/calibration

Granules

dbID	insertTime	LocalGranuleID	BeginningDateTime	EndingDateTime
4195	Oct 16 1997 3:43:47:270PM	LocalGranuleID	Jan 1 1993 1:00AM	Jan 30 1993 2:0

Figure 11.10-2. DB Viewer GUI

4 To view the inserted granules for a selected ESDT, first select an ESDT by clicking its short name in the Collections window.

- The selected ESDT is highlighted.
- Granule information for that ESDT, if there is any, will be listed in the Granules window.

5 To exit, click the **EXIT** button. This will end the DB Viewer GUI.

11.10.2 Using ECS Assistant to View Database

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

Assumptions:

1. The ECS Assistant has been properly installed.
2. The ECS Subsystem Manager is running.

To routinely start up the ECS monitor GUI, execute the procedure steps that follow:

1 At the ECS **Subsystem Manager** GUI, select a mode by **clicking** a mode in the mode list.
- The mode should be the one to be used for SSI&T.
- Once the mode is selected, the color of the subsystem name list is changed.

2 Select a subsystem by **clicking** the radio button next to the subsystem name under the subsystem component window.
- The selected subsystem radio button will be highlighted.
- The components corresponding to that the subsystem will be displayed in the component window.

3 Select a component by **clicking** a component name under the component window.
- All the servers for that selected component will be displayed in the server window.

4 **Click** the **monitor** button from the common tasks.
- This will invoke the Server Monitor GUI window as shown in Figure 11.10-3.
- The status "UP/DOWN" indicates whether the server is running.

5 To see which host each server running on, click the **cdsping all servers…** button.
- This will invoke the **cdsping GUI** as indicated in Figure 11.10.4.
- The host name for each running server is listed

6 Both **Server monitor GUI** and **cdsping GUI** can be updated by clicking the **update** button in the GUI.

This will cause the list to update to the current status.

7 To monitor other servers, repeat steps 1-6.

8 To exit, click the EXIT button. This will end the monitor GUI.

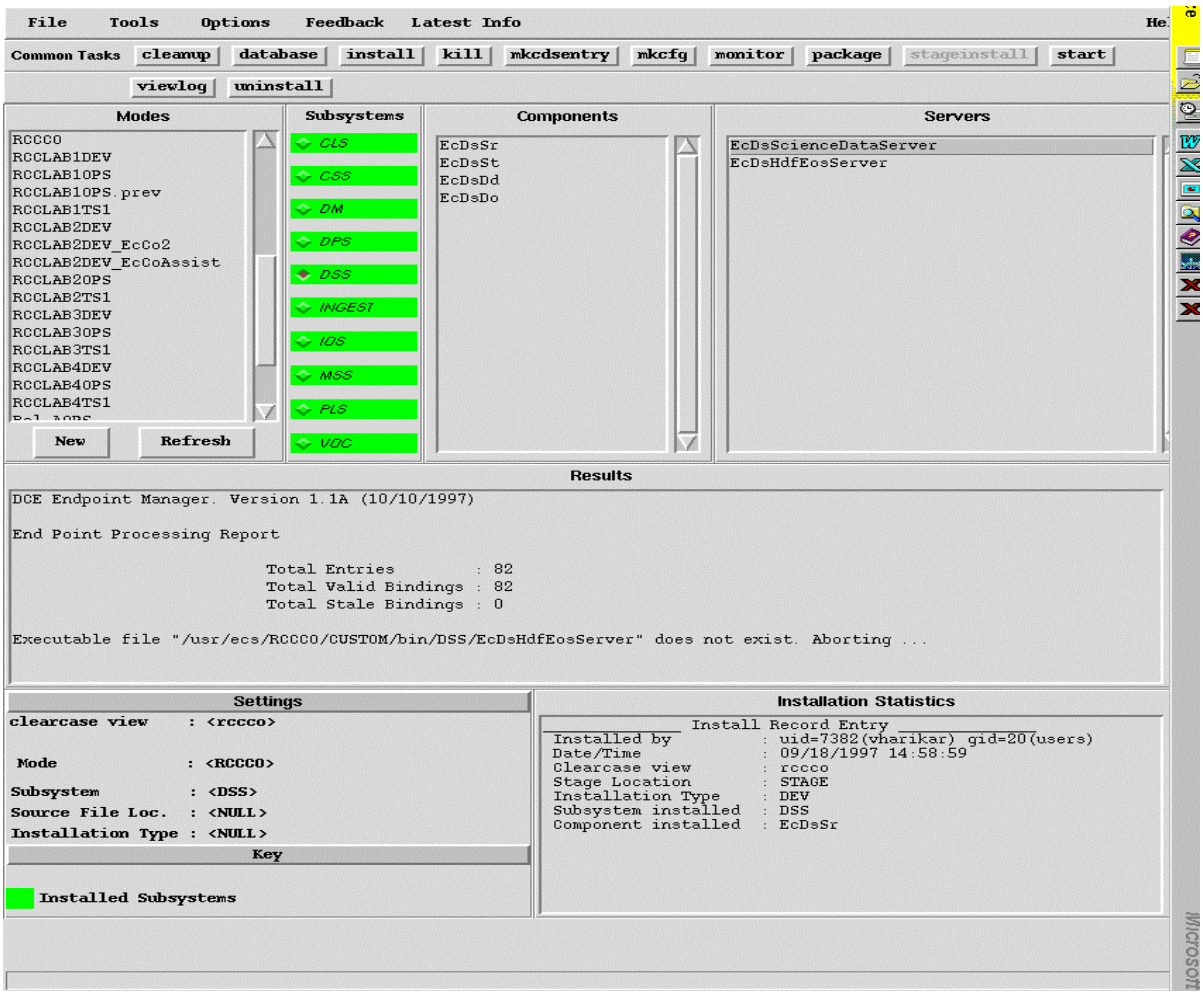


Figure 11.10-3. Server Monitor GUI

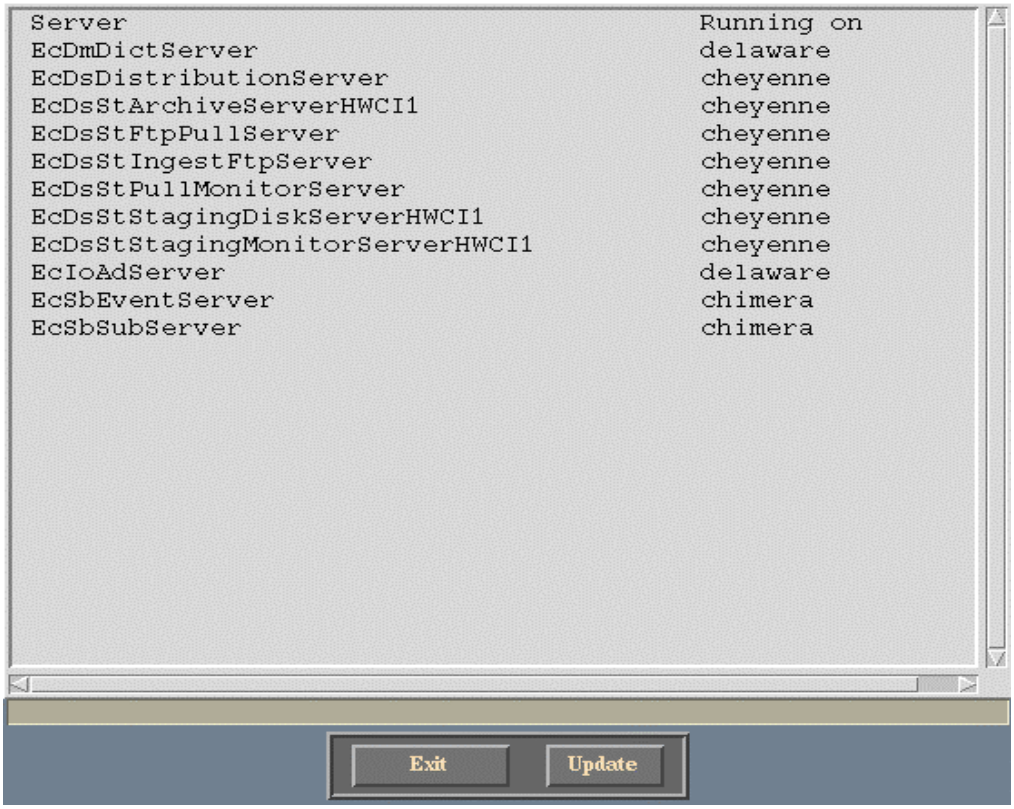


Figure 11.10-4. cdsping GUI

11.10.3 Using Browser to View ECS Science PDPS/IOS Database

Connect to the **SDSRV** database with login information as follows:

Server Name: **journey_srvr**

User Name: **sdsrvApp**

Password: **welcome**

The Browser lets you view all the tables in the **SDSRV** database with the mode you have selected.

The following tables are useful to track down the problems in insert ***.met**:

1 DsDeDictionaryAttribute

2 DsMdAddtionalAttributes

3 DsMdCollections

11.11 Installing ESDTs and Inserting Granules on the Science Data Server

11.11.1 Required Servers for Installing ESDT's

The following servers need to be started and running before installing ESDTs (with GDAAC machine names as examples):

- **Science Data Server (g0acs03)**
- **Storage Management Servers (g0icg01, g0drg01, g0dps02)**
- **Data Distribution Servers (g0dps02)**
- **Subscription Server (g0ins01)**
- **Advertising Server (g0ins02)**
- **Data Dictionary Server (g0ins02)**

11.11.2. Installing/Removing (ESDT/DLL) using the Science Data Server Operator GUI

Before the ECS can process data, an Earth Science Data Type must be installed into the system via the Science Data Server (SDSRV). The ESDT allows the system to recognize a particular data type and also provides services for accessing the data in the form of a Dynamic Link Library (DLL). The following procedures give step-by-step instructions on configuring the ESDT and installing the ESDT using the Science Data Server GUI., see Figure 11.11.1. Science Dataserver Operator GUI.

Installing a single Earth Science Data Type (ESDT) or Dynamic Link Library (DLL)

1 Copy the ESDT descriptor file and ESDT/DLL file from the source directory to the directory under the current mode of operations, The ESDT descriptor files are installed in the specified mode.

DLL's located : /usr/ecs/<mode>/CUSTOM/lib/ESS

ESDT Descriptors Located: /usr/ecs/<mode>/CUSTOM/data/ESS

2 Ensure that the following servers are currently executing: **Advertising Service** on the appropriate ADSHW HWCI server machine, **Data Dictionary Service** on the appropriate DMGHW HWCI server machine, **Science Data Server** on the appropriate ACMHW HWCI server machine and the **Subscription Service** that operates on the appropriate CSS server machine.

3 Start the **SDSRV GUI** by entering the following at the UNIX prompt on the SDSRV GUI workstation:

a) telnet to (SDSRV) **texas** [e.g.]

b) login: **cmops**
password: **opsu$er**

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

d) **cd /usr/ecs/<mode>/CUSTOM/utilities/EcDsSdSrvGuiStart <mode>**

4 On the main screen select the **Data Types** tab. A list of the ESDTs that have already been installed on the SDSRV will be displayed.

5 Click the **Add** button below to bring up the **Add Data Type** window.

6 **Descriptor Filename:** enter path to where the ESDT/DLL is located, including the full ESDT descriptor. **Archive ID:** field. **Note:** The descriptor filename and DLL Filename will require the complete directory path name as part of the file name which is the same directory as was specified in step 1 above. **(isolate the particular Data Type from the larger List, by using a unique sequence of letters or numbers at the end of the full path to better identify the Data Types ie;/*___*).** To specify specific directories, the **File..** button to the right of the Descriptor Filename and DLL Filename data entry fields will bring up a standard file selection GUI for this purpose. Also note that the **Archive ID** field will be constructed using the DSS Storage Management Staging Server UR that is found in the Science Data Server configuration file. The Science Data Server Configuration file is located in:
/usr/ecs/<mode>/CUSTOM/cfg/EcDsScienceDataServer.CFG.

Example: If the **DSSSTMGSTAGEINGSERVERUR** field was set to **DRP1_OPS** then the **Archive ID** fields would be set to **DRP1_OPS.**

7 Click the **Ok** button, this will cause the **Add Data Type** window to initiate installation of the ESDT/DLL into the Science Data Server.

8 The Science Data Server GUI will respond in a short time with a window stating that **MM/DD/YY HH/MM Finished adding ESDT's**.Also, the ESDT will appear alphabetically on the **Science Data Server - Data Types** list under the **Data Types** tab.

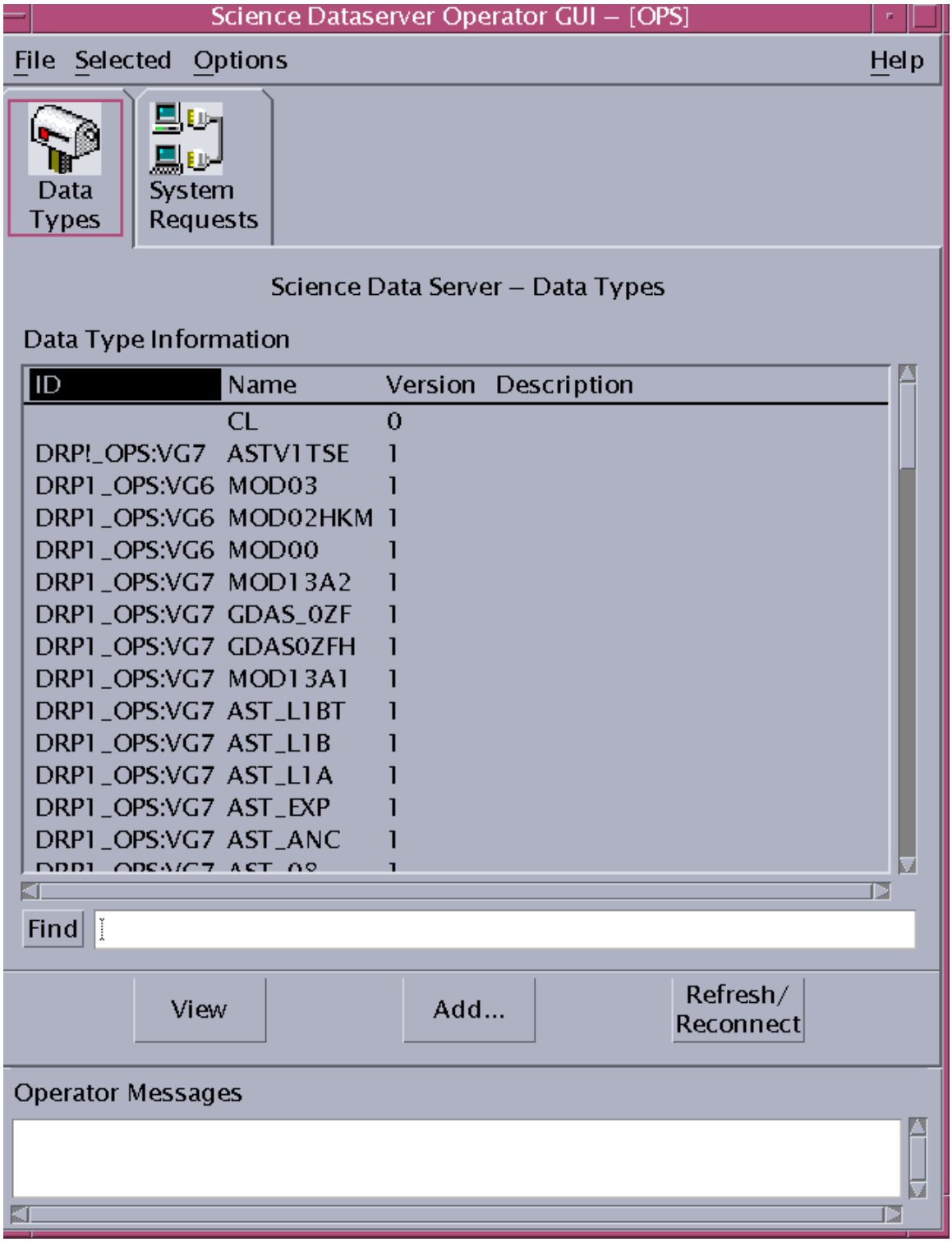


Figure 11.11.1. Science Dataserver Operator GUI.

11.11.3 Granule Insert into Science Data Server

The Science Data Server subsystem includes a utility that will allow users to manually insert a data granule into the ECS. The tool will prompt the user for key inputs for The Science Data Server subsystem includes a utility that will allow users to manually insert a data granule into the ECS. The tool will prompt the user for key inputs for inserting the granule. The following procedures describe this process.

1 Ensure that the **Science Data Server** subsystem is currently executing on the appropriate ACMHW HWCI server machine. Tested on **texas.**

2 Start the Science Data Server test utility by entering the following at the UNIX prompt on the SDSRV workstation:

setenv MODE TS1•
cd /usr/ecs/<mode>/CUSTOM/bin/DSS/•
source ../ ../utilities/EcCoEnvCsh•
/usr/ecs/<mode>/CUSTOM/bin/DSS/dttest6 ConfigFile•
/usr/ecs/<mode>/CUSTOM/cfg/EcDsScienceDataServerClient.CFG•
ecs_mode <mode>•

3 The following selection menu is displayed:

1. **INSERT granule**
2. **INSERT granule with browse file**
3. **ACQUIRE granule with date**
4. **ACQUIRE granule with a UR**
5. **DELETE granule**
6. **Exit•**

Please make selection=>•

Choose option **1** to perform the search and acquire.

4• The program indicates that the search process will take place first by displaying the message "**Executing insert .....** ". The user is prompted to enter the datatype of the data that will be inserted. For example:

Enter data type=> **AST_L1BT**

5 The program will then prompt the user for the full path name of the data file. For example:

Enter datafile name (full path)=>**/tmp/:SC:AST_L1BT:1391:1.EOSHDF**

6 The program will then prompt the user for the full path name of the metadata file. For example: Enter data metafile name(full path)=>**/tmp/AST_L1BT.MCF**

7 The program will then give status on the success of the insert. The following messages should appear on the successful insert:

Insert science data only…
Trying to make a request to {:DSSDSRV} Success.

8 The user should hit return at this prompt and the program will redisplay the first menu that was given in step 3. The user can then choose option 6 to exit.

11.11.4 Acquire a Granule from Science Data Server

The Science Data Server subsystem includes a utility that will allow users to manually search and retrieve (acquire) a data granule from the ECS. The tool will prompt the user for key inputs for acquiring the granule. The following procedures describe this process.

Acquiring a Granule from the Science Data Server

1 Ensure that the **Science Data Server** subsystem is currently executing on the appropriate ACMHW HWCI server machine. Tested on **texas.**

2 Start the Science Data Server test utility by entering the following at the UNIX prompt on the SDSRV workstation:

setenv MODE TS1•
cd /usr/ecs/<mode>/CUSTOM/bin/DSS/•
source ../ ../utilities/EcCoEnvCsh•
/usr/ecs/<mode>/CUSTOM/bin/DSS/dttest6 ConfigFile•
/usr/ecs/<mode>/CUSTOM/cfg/EcDsScienceDataServerClient.CFG•
ecs_mode <mode>•

3 The following selection menu is displayed:

1. **INSERT granule**
2. **INSERT granule with browse file**
3. **ACQUIRE granule with date**
4. **ACQUIRE granule with a UR**
5. **DELETE granule**
6. **Exit•**

Please make selection=>•

Choose option **3** to perform the search and acquire.

4• The program indicates that the search process will take place first by displaying the message "**Executing search .....** ". The user is prompted to enter a hostname. Enter the hostname of the machine on which the Science Data Server process is executing:

Enter hostname=> **dss2**

5 The user is prompted to enter a data type. Enter the ESDT short name of the type of data that is to be acquired. An example:

Enter data type=> **AST_L1BT**

6• The user is prompted to enter a start and end date. These dates indicate a range over which the user would like to search the database for data of the given type. The start and end dates will narrow the search to those data granules that were collected within that time range. Times are given in the format **mm/dd/yy**. For example:

Enter starting date(mm/dd/yy)=> **07/04/97**
Enter end date(mm/dd/yy)=> **07/05/97**

7 After the start and end dates are entered the utility will make a request of the Science Data Server and the following message will be displayed:

Trying to make a request to [:DSSDSRV]

A table of data granules will be displayed to the user if any were found within the given time range. For example, if the user had requested to display all of the data granules for the ESDT **AST_L1BT** within a certain time range, the following table would be displayed:

UR Type Create Date Size
-- ------- ----------- ----
AST_L1BT SC:AST_L1BT:1390
AST_L1BT SC:AST_L1BT:1391
AST_L1BT SC:AST_L1BT:1322
AST_L1BT SC:AST_L1BT:1289
AST_L1BT SC:AST_L1BT:1299

The table displays the data type (AST_L1BT) and UR (i.e., SC:AST_L1BT:1390) for each granule found.

In addition to the table, a list of the granules with a corresponding numerical index will be displayed with a prompt to the user to enter the index of the granule that they wish to acquire. An example of the indexed list follows:

NOW ENTERING ACQUIRE•
There is(are)5 in the collection•
Index = 0 AST_L1BT SC:AST_L1BT:1390 Index = 1 AST_L1BT•
SC:AST_L1BT:1391 Index = 2 AST_L1BT SC:AST_L1BT:1322 Index = 3•
AST_L1BT SC:AST_L1BT:1289 Index = 4 AST_L1BT SC:AST_L1BT:1299•
Please enter the index of the associated UR•

8 At this time the user should enter the numerical index of the granule that they wish to acquire and hit enter.

9 The SDSRV utility will prompt the user for a media type, the type of media on which the data will be retrieved:

Valid Media Types:
FtpPull
FtpPush
8MM
4MM
CDROM
9TRK
Enter media type (case sensitive)=>

The user should enter one of the values of the valid media types. If the user entered FtpPush, the user will expect the data to be ftp'd to a given directory.

10 After the user has entered the Media type, the utility will prompt the user for the media format. The media format should be entered as "FILEFORMAT"

Enter mediaformat=> **FILEFORMAT**

11 After the user has entered the Media format, the utility will prompt the user for the user profile id. This entry can be any alphanumeric character.

Enter userProfID => **a**

12 After the user has entered the user profile id, the utility will prompt the user for a user id and associated password. The user id/password will be used for authorization to perform

the ftp of the data file from the archive area to the user-specified directory. The password field will not be echoed to the screen. The following is only an example. The user should use a valid user id/password within the current environment.

Enter username=> **sdsrv**
Enter password=>

13 The next entry that the user must enter is the host id of the machine to which they want the data ftp'd. Enter a valid host name as follows:

Enter host=> **dss2**

14 After the host id, the user must enter the fully qualified destination directory to which the file will be ftp'd:

Enter destination=> **/tmp**

15 After the destination directory has been entered, the utility will give the following message:

Trying to make a request to [:DSSDSRV]
If the acquire operation is successful, the utility will give the following message:
Acquire successful.
Please <CR> to continue.

16 The user should hit return at this prompt and the program will redisplay the first menu that was given in step 3. The user can then choose option 5 to exit.

11.11.5 Add ESDTs through the ECS Assistant GUI

Execute the procedure steps that follow:

1 Launch the ECS Assistant GUI.

- The GUI will display three selection buttons, one of which is **ESDT Manager**.

2 Click the **ESDT Manager** button to open the ESDT Manager GUI.

The ESDT Manager GUI will be invoked.

3 Select a mode by clicking the drop down list in the **Mode** window.

- All the modes are listed in the **Mode** window.
- Select a mode by clicking its name in the mode list.

4• Input a DLL path where the shared object files are located by typing a full path name in the **DLL Path** window. **(i.e.; "/usr/ecs/OPS/CUSTOM/lib/ESS"),**

ESDT Descriptor Path: (i.e.; "/usr/ecs/OPS/CUSTOM/data/ESS").

5 Select a subdirectory for the instrument team by clicking the corresponding abbreviation.

- The descriptor files installed for that instrument will be listed.

6• Select descriptor files to be added to the archive by clicking the descriptor names in the list.

- The selected descriptor files are highlighted.

7 Move the selected files to the **Selected Files** window by clicking the "==>" button.

- The short names for the selected descriptor files are listed in the Selected Files window.

8 To add the selected files into the archive, click the **Add** button.

- The ESDTs are added to the archive.

9• To exit, select **Exit** in the **File** pull-down menu.
This will terminate the ESDT Manager GUI.

11.11.6 Using ECS Assistant to Manage ESDTs

With the ECS Assistant Tool, ESDTs can be added to and removed from the Science Data Server database by means of a GUI. These operations will be described in the following sections.

11.11.6.1 Installing ESDTs/DLL's to the Storage Area

In the ECS system, ESDTs are used to define data granules. Prior to executing a PGE, the relevant ESDT descriptor files and their DLL codes must be installed into the system through the science data server. ECS Assistant provides an easy-to-use GUI to perform these activities. If the installation is successful, the science data server, advertising server, subscription server, Management subsystem (MSS) and the new ESDT implementation library (DLLs) will all be updated.

ESDT's, (both components; descriptor and corresponding DLL files) to be installed must exist and have been verified for syntax, valids, and other metadata attributes correctness.

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

Assumptions:

1. The ECS Assistant is up with the necessary servers listening.
2. The ECS Assistant GUI is running with **ESDT Manager** selected.
3. Proper ClearCase view has been set.
4. The ESDT descriptor files are installed in the specified mode.

   DLL's located : /usr/ecs/<mode>/CUSTOM/lib/ESS

   ESDT Descriptors Located: /usr/ecs/<mode>/CUSTOM/data/ESS

11.11.6.2 Adding ESDTs through the ECS Assistant GUI:

1 Launch the ECS Assistant GUI.

- The GUI will display three selection buttons, one of which is **ESDT Manager**.

2 Click the **ESDT Manager** button to open the ESDT Manager GUI.

3 Select a mode by clicking the drop down list in the **Mode** window.

- All the modes are listed in the **Mode** window.
- Select a mode by clicking its name in the mode list.

4• Input a DLL path where the shared object files are located by typing a full path name in the **DLL Path** window. **(i.e.; "/usr/ecs/OPS/CUSTOM/lib/ESS")**

5 Select a subdirectory for the instrument team by clicking the corresponding abbreviation.

- The descriptor files installed for that instrument will be listed.

6• Select descriptor files to be added to the archive by clicking the descriptor names in the list.

The selected descriptor files are highlighted.

7 Move the selected files to the **Selected Files** window by clicking the "==>" button.

- The short names for the selected descriptor files are listed in the Selected Files window.

8 To add the selected files into the archive, click the **Add** button.

- The ESDTs are added to the archive.

9• To exit, select **Exit** in the **File** pull-down menu.
This will terminate the ESDT Manager GUI.

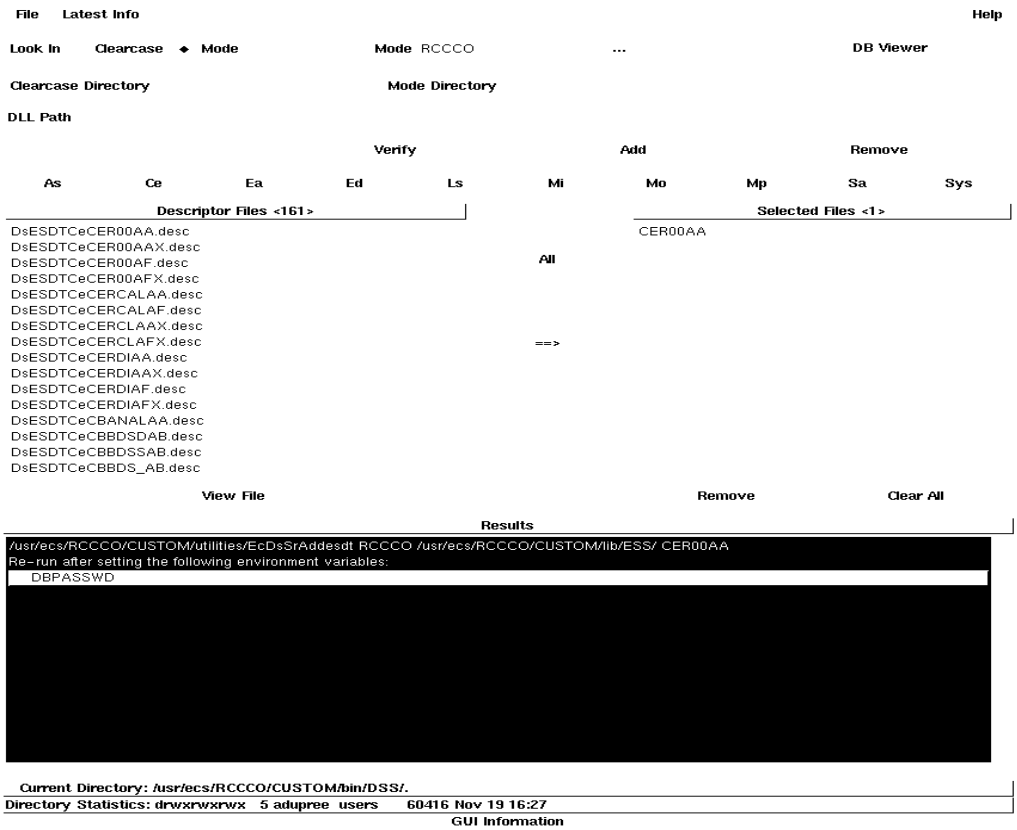


Figure 11.11-6. ESDT Manager GUI.

11.11.6.3 Validating Successful ESDT Installation

Criteria for success:

The **SDSRV** will display an Event ID to the fact that a new ESDT has been installed successfully.

The following servers will also need to have acknowledged a successful ESDT Event ID before additional work can be done: **ADSRV, DDICT, & SBSRV**

11.11.6.4 Removing ESDTs with ECS Assistant

If an ESDT is installed incorrectly or a new version of it becomes available, the installed ESDT will need to be removed from the archive, so that the new ESDT can be added. As with addition of ESDTs, their removal can also be accomplished with the ECS Assistant. Doing so performs the following actions:

- The selected ESDT is removed from DsSrConfiguration.acfg file
- The selected ESDT is removed from the database table
- The event file for the selected ESDT is removed
- The destination descriptor and DLL files for the selected ESDT are removed.

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

Assumptions:

1. The ECS Assistant has been properly installed.
2. The ECS Assistant GUI is running.
3. ESDTs are not stored under ClearCase.
4. The ESDT descriptor files are installed in the specified mode.

11.11.6.5 Removing ESDTs from Archive Area using ECS Assistant GUI:

1 Open the ECS Assistant GUI **-** The GUI will display three selection buttons, one of which is **ESDT Manager**.

2 At the ECS Assistant GUI, click the **ESDT Manager** button to open the ESDT Manager GUI.
- The ESDT Manager GUI will be invoked as shown in Figure 11.11-1

3 Select a mode by clicking the drop down list in the Mode window.
- All the modes are listed in the Mode window.
- Select a mode by clicking its name in the mode list.

4 Input a DLL path where the shared object files are located by typing a full path name in the DLL Path window.

5 Select a subdirectory where the ESDTs are staged by clicking a subdirectory name (as indicated by the instrument team abbreviation).

- The descriptor files stored in that directory are listed.

6 Select descriptor files to be removed from the archive by clicking their names in the list. The selected descriptor files are highlighted.

7 Move the selected files to the Selected Files window by clicking the "==>" button.

- The short name for the selected descriptor files are listed in the Selected Files window.

8 To remove the selected ESDTs from the archive, click the **Remove** button.

- The ESDTs are removed from the archive.
- The necessary cleanups are performed.

9 To exit, click the **EXIT** in the **File** pull down menu.

- This will end the ESDT Manager GUI.

11.11.6.6 Removing ESDTs using the Command Line:

Procedures:

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

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

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

4 **cd dbr**

5 **source dx.csh**

%dbr

First delete ESDT"s from the Advertisement Subsystem:

****

6 **rlogin incagold -l cmts1**

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

8 **rlogin incagold -l ios**

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

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

11 **setenv MODE OPS**

12 **source EcCoEnvCsh**

13 **cd /usr/ecs/OPS/CUSTOM/bin/IOS**
ContributionDriver OPS
awhitele
awhitele
3
2

14 your_short_name_here

- y

Success is when the "<" prompt returns

To make sure the advertisements are deleted from the database

15 incagold% isql -Uios_role -Pwelcome -Sincagold_srvr

- [If not OPS mode]
- 1> use IoAdAdvService_MODE
  [where MODE is your mode, e.g. TS1]
- [if OPS mode}
  1> use IoAdAdvService
  2> go
  1> select * from IoAdAdvMaster where title like "%your_short_name_here%"
  2> go

Result should be no rows returned.
If you do get rows returned, the delete from advertisement did not work.
****
Then delete ESDT
****

16 **rlogin** [**texas**] -l **cmts1**, pw: **ecsu$er**

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

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

19 **EcDsSrRmesdt OPS** your_short_name_here

Success is no error msgs
****
Kill servers -- AFTER WARNING EVERYONE WORKING IN YOUR MODE!
****

20 Using ECS Assistant : # kill Sdsrv & HdfEosSrv & AdSrvr
****
#This will clean up DCE ‘s. Using ECS Assistant : # Restart serversl Sdsrv & HdfEosSrv & AdSrvr
****

- # start Sdsrv & HdfEosSrv on [texas]
- # start SubSrvr on [incagold]

(cleanup done automatically)

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

****
Now reinstall the ESDT on SDSRV

11.12 Production Rules

11.12.1 Purpose and Scope

This section describes the production rules supported by the Release B CDR design. It is intended to explain the implementation of these production rules in ECS and to document the information required for each rule. This section does not cover the exact design and

implementation of these production rules in the PDPS system. That information may be found in the PDPS design documents, primarily in the Planning Subsystem document 305-CD-026-002, Release B SDPS Planning Subsystem Design Specification for the ECS Project.

While this section does address the syntax and operational procedures for specifying production rules, it does provide detailed information about what data is required for each rule and how that data will be used

11.12.2 Overview of Production Rules

11.12.2.1 Data Processing in ECS

Before discussing production rules, it is important to have a basic understanding of how the Earth Observing System Data and Information System (EOSDIS) Core System (ECS) conducts its data processing. ECS provides facilities for running product generation executives (PGEs) to produce science data products. The support for this function primarily resides in three subsystems: the Data Processing Subsystem (DPS) which actually executes the PGEs, the Planning Subsystem (PLS) which plans the execution of PGEs to efficiently utilize the local computing resources and the Data Server Subsystem (DSS) which provides PGEs with input data and archives output data. The design and implementation of the DPS and the PLS are carried out by the ECS organization known as the Planning and Data Processing System (PDPS).

Data Processing Subsystem (DPS) - The DPS provides the environment under which PGEs are run. It works with the SDP toolkit to interface with the science software. The DPS coordinates the acquisition of input data (staging) and the archiving of output data (de-staging) with the DSS.

The DPS also ensures that there are adequate resources (disk space, RAM, etc.) available before a PGE begins execution. The DPS uses the AutoSys COTS product to implement the PLS production plan.

Planning Subsystem (PLS) - The PLS generally manages the ECS data processing activities and provides the main interface between the DPS and the rest of ECS. The PLS plans the execution of PGEs to provide efficient use of computing resources. The PLS receives. Production Requests (PRs) either from a software tool (Production Request Editor) or, for on-demand production requests (OPRs), from the DSS or DPS (see Error Handling). The PLS breaks up PRs into data processing requests (DPRs), which are individual runs of a PGE. It maintains information about input data for specific DPRs and passes that information to the DPS. It queries the DSS for data which meets required criteria and provides the universal references (URs) of that data to the DPS for acquisition.

Data Server Subsystem (DSS) - The DSS provides input data to the DPS and archives output data from the DPS. It responds to queries from the PLS and provides the PLS notification of the arrival of input data for all subscriptions the PLS has placed with it. The DSS also submits On-Demand Production Requests (OPRs) to the PLS.

These three subsystems form the core of the ECS data processing services. Since it has responsibility for managing these activities, most of the information about production rules will be maintained by the PLS.

11.12.2.2 General Definition of Production Rules

Simply stated, production rules are the instructions about how a particular PGE is to be run.These instructions specify a wide range of information such as the input and output data types, the frequency of execution, activation conditions and error handling instructions. This section focuses on the other types of production rules such as alternate inputs to be used or conditional PGE activations. These and other production rules are defined in the following sections.

Production rules in the ECS system will be tied to PGEs. This means that each PGE will use one or more production rules. The production rules will be initially entered by the Instrument Team with further modifications if necessary when a PGE undergoes Science Software Integration and Test (SSI&T) at the DAAC. Where applicable, default parameter values will be entered at that time. Many of these defaults can be overridden in the production environment when a Production Request is entered.

11.12.3 Production Rule Definitions

Introduction

This section presents a basic definition and examples of each of the production rules currently supported by the Release B CDR design. The production rules are divided into three categories:

Input Data Specification
Conditional Activation
Error Handling

Input Data Specification rules specify how the inputs for a particular run of a PGE are identified. Conditional Activation rules stipulate the conditions of when a particular PGE is to be run. Error Handling rules specify automatic actions to be taken when a PGE fails. Each sub-section is titled with the name of the production rule being discussed.

It is hoped that providing a common nomenclature will facilitate any future discussions between ITs and ECS about production rules. After each rule there is a list of the information required to implement that rule in the production environment (i.e. post-launch, at the DAACs). This information will initially be provided at SSI&T time. In the operational environment many of the values can be set when a Production Request is entered. Most production rule sections also include a simple diagram to help illustrate the concept being presented.

11.12.3.1 Input Data Specification

These production rules generally stipulate how inputs for a particular run of a PGE (a DPR) are identified/selected/created. Most PGEs have at least one input and one output. The most basic information about these inputs and outputs is the Earth Science Data Type (ESDT). It is assumed

in the examples in this chapter that the ESDT is specified for all data sets which are used or produced by a PGE.

11.12.3.2 Basic Temporal

The most basic type of production rule specifies, for each input ESDT, the temporal range of that ESDT required by the PGE.

For example, the run of a PGE which produces a particular 2.5 minute MODIS Level 1B data granule requires as input the specific 2.5 minute Level 1A granule which is ontemporaneous with the Level 1B granule to be produced.

Another example would be the production of a CERES Level 1A granule from the Level 0 data. CERES L1A granules cover 24 hours and require the PLS to identify and the DPS to stage all of the Level 0 data files that cover a particular 24 hour period. This sort of simple temporal specification is basic to the system and will be used by every PGE.

Inputs required to implement the Basic Temporal rule

For each PGE which uses this rule, the following data must be provided for each ESDT (input oroutput): Time period - Length of time of ESDT (e.g. 12 hours, 1 week or one year).

Boundary - Date/time to start counting periods

Both of the above parameters are specified at SSI&T time.

11.12.3.2.1 Example use of the Basic Temporal Rule

Suppose there is a PGE which creates a daily (24 hour) data set and those data sets should correspond with calendar days. In that case Time Period = 24 hours and Boundary = 1/1/98 00:00 (this could be any date). When a Production Request is entered, the Planning Subsystem will scan backwards from the start time of the PR until it finds the start of a period. It then produces DPRs for each period of time covered by the PR. Figure 3-1 illustrates how this would work. Given the period and boundary defined above, two PRs are entered, the first for the time between 6/3/99 12:00 - 6/7/99 6:00 while the second specifies 6/7/99 18:00 - 6/16/99 12:00. When the first PR (#1) is entered, the Planning Subsystem determines that the start time of the PR is not the start time of a period. It then scans backwards, finding the start time of the period to be 6/3/99 00:00 and creates a DPR (1.1) to process that period. It then continues producing DPRs (1.2, 1.3, 1.4 and 1.5) until it reaches the end of the period which contains the end time of the PR. When the second PR (#2) is entered, the same process takes place. Note that since the end time of PR #1 and the start time of PR #2 fall in the same period, duplicate DPRs (1.5 and 2.1) are created. This sort of duplication is checked for by the Planning Subsystem and the second DPR is not run.

Table 11.12.3.2-1. Basic Temporal

Boundary (1/1/98 00:00)
= Period (24 Hrs)
PR #1 PR #2
DPR 1.1
DPR 1.2
DPR 1.3
DPR 1.4
DPR 1.5 = DPR 2.1
DPR 2.2
DPR 2.3
DPR 2.10
DPR 2.9
6/1/99 6/23/99

11.12.3.3 Alternate Inputs

This is a fairly simple rule, although there are several options which give it great flexibility and power. There are cases where, for one of its input data sets, a PGE can use any one of several inputs. For example, a particular PGE might require model wind data as an input and would be capable of accepting wind data from a DAO model, an NCEP model or, as a last resort, could use climatology. Variants of this rule allow the alternate input to be optional or allow alternates to be grouped so that more than one of the alternates may be needed (i.e. use any M of N ESDTs in the group).

11.12.3.4 Inputs required to implement the Alternate Inputs rule

ESDT of Primary Alternate - This is the data set ID for the most desirable alternative.

Timer for Primary Alternate - Amount of time to wait before attempting to use second alternate. Number Needed - Number of the alternate data sets (from this group of alternatives) required by the PGE. Usually 1, but can be more. If set to 0, this set of alternates is optional (i.e. the PGE can run without any of the alternates) ESDT of Second Alternate -

Data set ID for Second Alternate

Timer for second Alternate - Amount of time to wait before attempting to use third alternate.. (include any number of alternates, no practical limit).

ESDT of Last Alternate - Data set ID for Last Alternate.

Timer for Last Alternate - Note that this timer is used only if Number Needed = 0, which indicates that this set of alternates is optional. In that case, if this timer expires, the DPR will be executed even if none of the alternates is available. If however, Number Needed > 0, then this timer is not used; the DPR will be held indefinitely, waiting for one of the

alternates to become available.

11.12.3.4.1 Example use of the Alternate Inputs Rule:

Let's assume that a PGE has two required input data sets and a third data set which can be any one of three alternates. Of the three alternates, the first two choices are almost equivalent, while the last choice is less desirable. In this case we designate the Primary Alternate (first choice) and set a timer for 1 hour. The second choice is designated with a timer set for 4 hours. This situation is shown in Table 11.12.3.4-1. What will happen is that, after the two required inputs are available, an attempt is made to acquire the first choice. If that choice is unavailable, the Planning Subsystem holds the DPR and starts the first timer (1 hour). If the first choice becomes available in that hour, the DPR is started immediately. If the first timer expires and the first alternate is still unavailable, an attempt is made to use the second choice. If the second choice is unavailable, the second timer (4 hours) is started. If the first or second alternates become available at any time during that 4 hours, the DPR is started immediately. However, if the second timer expires (a total of 5 hours after the two required inputs are available), an attempt is made to use the third choice. Unless this set of alternates is optional, most PGEs are expected to have a last alternate which is always available (e.g. climatology). If not, the DPR is held indefinitely waiting for one of the alternates to become available.

Table 11.12.3.4-1. Alternate Inputs

Required Dataset 1
Primary Alternate
Second Alternate
DPR
< or > First Choice –Timer set to 1 hour.
Wait 1 hour after required data sets are availabe before trying to use second alternate.
Second Choice –Timer set to 4 hours.
Wait 4 hours after this choice was first tried before attempting to use third alternate.
Output Dataset
Third Alternate < or > Third (in this case, last) choice. Attempt to use if primary and secondary alternates are unavailable 5 hours after required datasets are available.
Required Dataset 2

11.12.3.5 Tiling

In this case a PGE is set up to run for a series of pre-defined, rectangular areas (tiles). The tile definitions need to be entered into the PDPS database so that the proper input data granules can be identified. The implementation of this rule calls for the PGE developers to create tile definition files which would have descriptions of each tile. The Planning subsystem would then use those definitions to query the Data Server for input data granules for each tile. Tiles are grouped together into clusters which are likely to share common input granules. This is done for efficiency of operations.

11.12.3.5.1 Inputs required to implement the Tiling rule:

The primary input for the tiling rule is a tile definition file. This file will contain one entry for every tile which will include: Tile ID - Unique identifier used to refer to the tile Lat./Lon. of tile corners -

Four coupled pairs of coordinates which bound the tile. Cluster ID - ID of cluster in which tile falls.

Table 11.12.3.6-1. Tiling

Tile 1 Input Data
Tile 2 Input Data
Tile 3 Input Data
Tiles 4-6
Tiles 7 and 8
DPR-1
DPR-2
DPR-3
Output granule for tile 1
Output granule for tile 2
Output granulefor tile 3
Tiles 4-8 are processed in the same manner as tiles 1-3.
8 Tiles clustered together since they are likely to have overlapping inputs
Original
Swath
Data

11.12.3.5.2 Example use of the Tiling Rule

A tile definition file is part of the package delivered with the PGE for SSI&T. In the production environment, a PR is entered to run the PGE on all the tiles. A DPR is generated for each tile. These DPRs are grouped by Cluster ID so that, since they will share some of their inputs, these DPRs will be run on the same machine which will minimze the the staging and copying of those inputs amongst the various production machines. The PLS will use the tile definitions to query the DSS to idenify all input data granules covering a tile. It is likely that many of these input granules will have data outside as well as inside each tile. The URs for these input granules and the specific tile definition are then passed to the DPS as part of each DPR. Each DPR will then run, the DPS will aquire the inputs from the DSS and the PGE will run and, using only the data in the tile, create an output data granule for its tile.

11.12.3.6 Data Server Proxy (Subsetting/Subsampling)

There are cases where a PGE would like to use Data Server services on its input data sets prior to running the PGE. In these cases the Planning and Data Processing subsystems simply act as proxies for the PGE by calling those Data Server services. The services provided here are totally

dependent what services the DSS offers for each specific product. Instrument teams wishing to use specific services on specific products should coordinate with the Data Server Subsystem to ensure those services will be available.

11.12.3.6.1 Inputs required to implement the Data Server Proxy rule

Each PGE which uses this rule will be need to identify:

Input ESDT - ID of data set upon which Data Server services will be performed.

Service Requested - Data server service ID Interface Parameters - Parameters specific to the service being requested. For example, a spatial subset request would specify the geographic extent while a temporal subset request would specify the time period requested.

Table 11.12.3.7-1. Subsetting & Subsampling

Subset of Original Dataset
DPR Output Data Granule
DPR Output Data Granule Subsample of Original Dataset
Spatial Subsetting Subsampling

11.12.3.6.2 Example use of the Data Server Proxy Rule

Assume a PGE with an input data set which covers the entire globe and is gridded at a .1 o by .1 o resolution. During the post-launch, PGE shakedown phase, the science software developer might want to generate a 1 o by 1 o resolution product for a quick look at the output data. Rather than have the PGE do its own averaging/subsampling, Data Server subsampling services could be invoked so that the PGE only has to run on 1/100th the amount of the original data. In this case a second PGE profile would be created at SSI&T which would include the service request and track the different runtime characteristics (e.g. CPU time, disk storage for input/outputs, RAM usage, etc.)

11.12.3.7 Level 0 to Level 1A

This is a special case of the simple temporal range rule and is of interest only to certain Level 1A PGE developers. The issue is what rules PDPS uses to determine what Level 0 data files are required to produce a particular Level 1A granule. In many cases (such as CERES), this is handled by the Basic Temporal rule rather than the method described here. This rule is intended to handle orbit-based Level 1A granules. Level 0 'granules' received by EDOS will be not have any discernible relationship with the spacecraft orbit. The Level 0 'granules' have not even been fully defined for some instruments flying on other platforms. In these cases, the Planning Subsystem will access a crude orbit model (really just a lookup table) to determine the start/stop times of a given orbit. It will then use those start/stop times (after adding a small cushion to ensure complete coverage of an orbit) to query the data server for the Level 0 data covering the orbit. The response to that query is given to the Data Processing Subsystem which acquires the data. The PGE then uses SDP Toolkit calls to determine the exact extent of the orbit. A similar

process will be used for instruments having orbit-based Level 1A granules but which will be flying on platforms other than AM-1. In the case of SeaWinds, flying on the ADEOS II platform, a temporal offset may have to be used if the orbit model is based on equator crossing rather than on pole crossing as the SeaWinds want for their Level 1A granules.

Time

Table 11.12.3.7-1. Level 0 to Level 1A

Time
Level 0 Data
Orbit-based
Level 1A Granules
Using temporal definitions of orbits, the PLS identifies and the DPS stages the appropriate Level 0 Data for each orbit-based Level 1A granules

11.12.3.8 Conditional Activation:

Most PGEs have well defined times and conditions when they are to be executed. The most common activation condition is the availability of all input data sets. Similarly, the frequency of execution is usually well defined (e.g. run once for every granule or run monthly averages once a month). However, some PGEs might have additional/different constraints on when they get run. This section addresses those cases.

11.12.3.9 Intermittent Execution

A PGE can be set up to run on every Nth instance of input data. For example, a QA PGE that is run on a daily product may only need to be run every fifth day to provide a spot check. Note that this does not refer to the common case of only running a weekly averaging PGEs once each week, which would be handled by the Basic Temporal rule and the time ranges specified for the input and output ESDTs. Rather this is a special case where a PGE can be run every day (or hour, week, etc.), but, for whatever reason, it is only desired to be run every Nth day.

11.12.3.9.1 Inputs required to implement the Intermittent Execution rule

This rule is implemented by using two parameters:

Number to Skip - Number of DPRs to be skipped (not executed)

Number to Keep - After skipping the specified number of DPRs, how many are to be kept. This number will usually be one, but could be any number. The use of these parameters will allow a pattern of execution to be established. This pattern is not maintained between PRs.

Table 11.12.3.9-1. Intermittent Execution

PR #2
PR #1
DPR
1.1
DPR
1.2
DPR
1.3
DPR
1.4
DPR
2.1
DPR
2.2
DPR
2.3
DPR
2.4
DPR
2.5
DPR
1.2
DPR
1.3
DPR
2.3
DPR
2.2
DPR
2.5
PRs #1 and #2 are entered at different times.
PR #1generates 4 DPRs and PR #2 Generates 5.
Only 5 of the 9 are ‘kept’and actually run.
Number Skipped = 1
Number Kept =2
Created Run
Seperate PRs,
pattern is restarted for PR#2

11.12.3.9.2 Example use of the Intermittent Execution Rule

Using the above example of a QA PGE to be run every fifth day, let's assume a Production Request is entered which covers the period from 6/1 - 6/30. As part of the PR values are entered for number skipped (4) and number kept (1). The PR would be expanded into 30 daily DPRs, of which, four out of every five would be discarded, leaving one DPR every fifth day.

11.12.3.10 Metadata-based Conditional Activation

It is possible to determine if a given DPR should be run, based on the metadata of one or more of its input data sets. For example, a PGE could be set up so that a QA flag must be set to an acceptable level/state within the metadata of an input data set or the PGE should not be run. This production rule will work for both core and product-specific metadata. Note that this is different than data-based conditional activation, which will not be supported in Release B. If that sort of conditional activation is desired, the IT will need to define a product-specific metadata field which will be filled by the PGE producing that data.

11.12.3.10.1 Inputs required to implement the Metadata-based Conditional Activation rule

This rule is implemented by using a series of statements for each ESDT to be checked. These statements take the basic form of:

Metadata field Operand Value. These statements would be 'AND'ed together so that if any of the checks fail, the DPR will not be run. These statements would be entered in at SSI&T time, however, the values could be changed when a Production Request is entered.

Table 11.12.3.10-1. Metadata-based Conditional Activation

Input 1
QAFlag=7
DayNight=‘DAY’
Clouds=20%
Input 2
QAFlag=4
Input 1
QAFlag=9
DayNight=‘DAY’
Clouds=70%
Input 2
QAFlag=8
Input 1
QAFlag=7
DayNight=‘DAY’
Clouds=30%
Input 2
QAFlag=7
DPR 1
DPR 2
DPR 3
Metadata Checks:
Input 1:
QAFlag > 5
DayNight = ‘Day’
Cloud < 40%
Input 2:
QAFlag > 6
DPR 1 is NOT run since Input 2 failed the QAFlag check.
DPR 2 is NOT run since Input 1failed the cloud cover check.
DPR 3 IS run since Inputs 1 and 2 met all metadata conditions

11.12.3.10.2 Example use of the Metadata-based Conditional Activation Rule

Assume there is a PGE which uses multiple ESDTs as inputs. Two of the inputs (Input 1 and Input 2) need to have their metadata checked prior to the PGE being executed. Input 1 needs to have a certain quality, less than a certain percentage of clouds and be daytime data while Input 2 just needs to be of a certain quality. Table 11.12.3.4-1, illustrates this situation and shows three DPRs created for different instances of Inputs 1 and 2 and the disposition of those DPRs based on the metadata of the inputs. As stated earlier, if it was decided that the cloud cover rule was too strict, the value used for comparison could be changed when Production Requests are entered. If, however, a new check were needed for some other metadata field, this change would have to be done through SSI&T.

11.12.3.10.3 Mode-based Conditional Activation

In this case, the mode a given instrument is in will determine which PGE is run. For example an instrument might go into a calibration mode which requires that a special calibration PGE is run. Actually, this is just a specialized case of the metadata-based conditional activation. The added functionality here is that at SSI&T time PGEs can be grouped into PGE collections. In such cases, the instrument mode would determine which PGE in the collection is run. This mechanism is used primarily to improve the accuracy of plans generated by the PLS .

11.12.3.11 Error Handling

Error handling is a little different from the other two categories of production rules. This is because it is mostly an operational procedure which occurs in response to an (hopefully) anomalous event. The key mechanism for implementing error handling will be PGE exit codes. Release A PDPS is currently working to define PGE exit codes and determine how best to associate actions with exit codes. (Establishing Science Software Exit Conditions for the Production Environment, 420-WP-006-001) is in preparation. The key enhancement to error handling by Release B is the reuse of the on-demand processing mechanism to automate the response to a specific error code. In this case, when a PGE undergoes SSI&T, On-Demand Production Requests (OPRs) are then associated with various PGE exit codes.

11.12.3.11.1 Inputs required to implement the Automated Error Handling

At SSI&T time, for every exit condition which will need another PGE to be run, the following information will need to be entered:

Exit Code - PGE exit code which triggers action Message - Message to be displayed to operation console OPR - On-Demand Production Request to be executed. The OPR will have the same timeframe as the original DPR so if the DPR were run on data from 6/20/99 12:00-13:00, then the OPR would be given the same timeframe before being passed to PLS.

11.12.3.11.2 Example use of Automated Error Handling

It is difficult for a short example to capture the full error handling options of the ECS production system. The following example simply gives a flavor for what is possible with regard to automatically submitting OPRs based on a PGEs exit code. While a certain PGE (PGE1) is undergoing SSI&T it is set up so that if PGE1 has an exit code of 6, it should be re-run in debug mode. If it has an exit code of 12 (can only happen from debug mode), then a second PGE (PGE2) should be run. At a later date, in the operational environment, a DPR for PGE1 is running and terminates with exit code 6. An OPR is generated for PGE1 which includes the runtime flag used to send it into debug mode. A message is displayed on the operator's console informing them of this event. The OPR is run and it finishes with an exit code of 12. Now another OPR is created, this one for PGE2. A different message is displayed on the operator's console.

11.12.4 Combinations of Production Rules

11.12.4.1 Introduction

One of the most powerful features of production rules is the ability to combine multiple rules. This gives the science software developers greater flexibility and control over the production of their data. This section is intended to illustrate how some of the production rules can be used together. It is not intended to be exhaustive, but rather to give a sampling of the more common combinations. Most combinations are quite easy to understand and the implications of these combinations are quite clear. However, while theoretically any and all of the production rules can be combined, some combinations will make little sense. For instance, combining tiling with intermittent execution will only produce DPRs for some tiles and it will not be easy for the science software developer to determine those tiles in advance. Conversely, intermittent execution works quite well with the basic temporal rule and behaves in an easily predictable manner. While the software can handle complex combinations, in practice these combinations might have results which are not especially intuitive. For example, while combining alternate inputs with metadata based activation, tiling and intermittent execution would seem quite reasonable to the Planning Subsystem software, it would be difficult for a human to determine the results in advance. However, if intermittent execution is removed from the above combination, the remaining combination might be a perfectly valid production recipe. The following sections provide a few examples of how production rules may be combined.

11.12.4.2 Basic Temporal

The Basic Temporal rule is fundamental to the production system. All Production Requests will have a temporal component. Consequently, all of the other production rules are being combined with the Basic Temporal rule.

11.12.4.3 Alternate Inputs and Metadata-Based Conditional Activation

It is possible to combine these two rules so that the metadata of the inputs determines which alternate is used. For example, suppose there is a PGE which, along with its required inputs, can use one of two alternates, but the primary alternate must have a certain level of QA set in its metadata. If the primary data set becomes available, its metadata is checked and, if it fails the check, an attempt is made to use the second alternate.

11.12.4.4 Alternate Inputs and Data Server Proxy

This combination if quite straightforward. After the input data sets are determined via the Alternate Inputs rule, the Data Server Proxy service is invoked.

11.12.4.5 Alternate Inputs and Tiling

In this combination determining which data granules fall within a tile is a completely separate activity from determining which alternate is most should be used. In this case the tile definition could be used to acquire the data sets which had be chosen as part of the alternate input process.

11.12.4.6 Intermittent Execution

While it was mentioned above that all PGEs use the Basic Temporal rule, it is worth emphasizing that point in the case of Intermittent Execution. This rule is somewhat unique since it doesn't create DPRs, it removes them. By definition this rule needs to be used in combination with another rule.

11.12.4.7 Changes to rules

The following changes have occurred in the listing and organization of the rules documented by this section:

· The Release A Basic Temporal rule has been added.

· The current Alternate Inputs rule is the consolidation of Optional Inputs, Alternate Inputs - Hierarchical Preference and Alternate Inputs - No Preference. Subsetting and Subsampling have been combined and renamed Data Server Proxy· Alternate Inputs - Temporal/Spatial Tradeoff has been left out. This is because the PDPS team is still determining the best manner to implement this rule. There are several reasons for this including the lack of an IT provided algorithm, the nature of which could impact the design and implementation. Depending on what variables the algorithm uses, this rule might actually be a variant of the Data Server Proxy rule, or it might require the algorithm(s) to be integrated into PLS code.

11.12.5 Production Rules Technical Information Sources

1 ECS Baseline Information System :

2 PDPS Home Page: http://dmserver.gsfc.nasa.gov/ecsdev/relb/pdps/index.html

http://pete.hitc.com/baseline/ -choose drop4, you will find latest

patch documentation and much more.

2 EOS Instrument Team Science Software (PGE's):

http://ecsinfo.hitc.com/iteams/Science/science.html. Production Rules White Paper and much more.

3 MODIS - Science Data Processing software Release 4 System Description, SDST-104 dated August 25, 1998.

4 Test Scenarios for selected Production Rules can be viewed by accessing the SCF at: **</home/dheroux/DPS/TESTBED/MISR/SSIT/V2/ODL/Scenarios**>

11.12.6 Production_Rules_Syntax

The Production Rules Syntax are presented as part of the Powerpoint Slides that accompany this document.

Production Rules identified thus far are listed as follows:

Basic Temporal, Advanced Temporal, Period Specification, Alternate Inputs, Optional Inputs, Metadata-based Activation, Metadata-Based Query -Static, Metadata-based Query - Dynamic, Orbit-Based Activation, Orbit Path, Runtime Parameter, Multi-File Granules, Multi-Granule ESDT's, Spatial Query, Minimum Number of Granules, Land Tiling, Tiling with Metadata-based Query, Optional DPRs, Ocean Data Day, Most Recent Granule, Alternates based on Minimum number of Granules, Zonal Tiling, Tile Clustering and Smart_Start_of_Year.

11.12.7 Production Requests

11.12.7.1 Science Software and Production Requests

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

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

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

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

The Planning Subsystem performs the following functions:

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

- Determines what data will be included in the DPR output so the system can make predictions concerning the availability of data for subsequent PGEs.

Figure 11.12.7.1-1 shows the relationships among the PGEs, PRs, and DPRs as they are accessed through the Production Request Editor GUI.

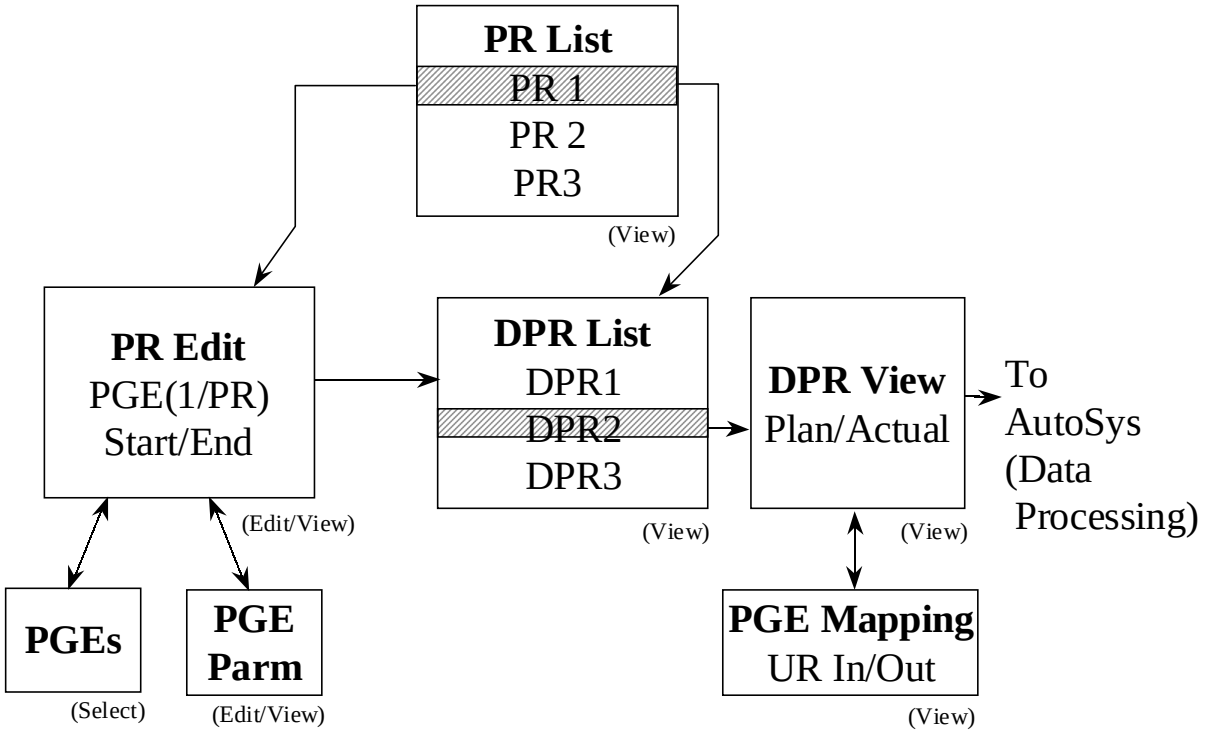


Figure 11.12.7.1-1. Production Request Editor Flow

11.12.7.2 Types of Processing

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

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

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

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

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

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

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

Another future feature is automated cross-DAAC planning. It is a process that will be undertaken when products produced at one DAAC require inputs being produced at another DAAC. The predicted production time of remote input products will be used in creating local production plans. The primary mechanism for cross-DAAC planning will be the use of Predicted Data Availability Schedules (PDAS), created when a plan is created. A DAAC's PDAS will be made available to remote DAACs via the Data Server.

Production Rules used in Planning PGE Processing

11.12.8.1 Production Rules Defining the PGE

Production rules are the instructions about how a particular PGE is to be run. The instructions specify a wide range of information such as the input and output data types, the frequency of execution, activation conditions and error handling instructions.

A single PGE may use one or more sets of production rules, known as PGE profiles, since it may be desirable to run the same PGE with different input data sets, or activation conditions. The production rules are entered when a PGE undergoes Science Software Integration and Test (SSI&T) at the DAAC. Where applicable, default parameter values are entered at that time. The

initially selected runtime parameters, metadata check parameters, tile IDs, and many of the default parameters can be overridden in the production environment when a Production Request is entered.

Production rules define the PGE to the Planning and Data Processing Subsystems (PDPS). The following types of conditions can be specified for each PGE:

- The time period for which the PGE will run.
  - A PGE can run every hour, every day, or for every orbit of a satellite. The frequency of how often a PGE runs must be defined to PDPS so that it knows when to plan and execute the PGE. A definition of a satellite's orbit could be included if the PGE were to be executed for some number of satellite passes.
- PGE Inputs.
  - A PGE can have any number of inputs. The types of the inputs and how frequently they are available helps determine on what basis the PGE is scheduled.
  - Most inputs to a PGE are retrieved based on time; the specified inputs are retrieved from the Data Server Subsystem for the time which the PGE is defined to execute. Production Rules allow other conditions to be added to the mix, such as checks or queries against the metadata of the input granules, or the lists of inputs as alternates (for when a primary input is not available) or optionals (for inputs without which the PGE can still run successfully). If inputs are defined as alternate or optional, the number of inputs staged for the execution of the PGE may vary from one run to the next.
- PGE Outputs.
  - A PGE can have any number of outputs. The characteristics of the outputs can have effects on any downstream PGEs that use them as inputs. For example, it is possible for an output to be defined as optional, in which case it may or may not even be produced. (When an output is not generated, it cannot be used as input for a downstream PGE.)
- Runtime Parameter Values.
  - A PGE can have any number of runtime parameters, which are values that are placed in the process control file (PCF) under specified logical IDs before the PGE executes. The PGE treats them as constants and normally they are set either during SSI&T or when the Production Request is entered.
  - For some production rules (such as Orbital Processing) there is specific information that can be placed in a runtime parameter if so desired by the PGE.
- Geographic Tiles.
  - A PGE can define a geographic location for which it will process data. Tiles are defined through production rules, and change the staging of inputs from time-based, to a combination of time- and geographically-based. Data are retrieved

based on their location on the Earth with respect to the tile that it is currently being processed.

Some (but not all) production rules can work with other production rules.

11.12.8.2 Production rules are often used for the selection of dynamic inputs.

- **Dynamic inputs** can be either internal or external.
  - **Dynamic internal** inputs are produced by other PGEs (they are called dynamic internal inputs because they are produced at an ECS DAAC).
  - **Dynamic external** inputs are periodically ingested and stored in the Data Server Subsystem (they are termed dynamic external inputs because they are produced outside of the DAAC).
- **Static inputs** are granules that are inserted during the SSI&T process and are retrieved not on the basis of time but by Earth Science Data Type (ESDT) and science group.
  - The Metadata Query Production Rule is the only production rule that works for choosing static inputs.

PGE profiles allow a PGE to be defined to PDPS multiple times, each with a different set of inputs, outputs, or even scheduling information. Each PGE's definition is made up of its name, its version and its profile number. Different PGE name/version pairs define different PGEs to PDPS. The addition of the profile allows for multiple definitions of a PGE name/version pair. There can be up to 99 profiles for each PGE.

11.12.8.3 Syntax of Production Rules

Production rules are defined in the following two ways:

- Through science metadata that is entered in various types of files during the SSI&T process.
- By entering parameter values when a Production Request is created to schedule the PGE.

During SSI&T, production rules are defined in files written in Object Description Language (ODL) in a parameter equals value format. There are three general categories of ODL files:

- PGE Science Metadata ODL Files.
- ESDT Science Metadata ODL Files.
- Production Rule-Specific Science Metadata ODL Files
  - Orbit Definition ODL Files.
  - Path Map Definition ODL Files.
  - Tile Definition ODL Files.

When a Production Request is created to schedule a PGE, it is necessary to enter certain information that is essential to implementing the production rules that affect the particular PGE. The information may concern the date and time-range

11.12.8.4 PGE Science Metadata ODL Files

The PGE science metadata ODL file defines a PGE (or at least the current plan for its operation) to PDPS. It specifies everything from the PGE name and version, to the period for the PGE (how often it runs), all inputs and outputs, any runtime parameters and any exit messages or dependencies. A template version of the PGE science metadata ODL file is created by the **SSIT Create ODL Template** program from a PCF from the PGE.

11.12.8.5 ESDT Science Metadata ODL Files

The ESDT science metadata ODL file defines a PGE input or output to PDPS. Each input and output of a PGE must have a corresponding ESDT science metadata ODL file defined. It describes everything that PDPS needs to know about the subject input or output file, from its name and version, to its period (how often data is collected), to where it is used and archived. Note that many PGEs can use the same input or output type, and thus can share the same ESDT science metadata ODL file.

Unlike the PGE science metadata ODL file, there is no tool for automatically generating a template ESDT science metadata ODL file. A template version exists under the data directory called ESDT_ODL.template. The template must be copied to a file that follows the naming convention *ESDTShortName#ESDTVersionID*.odl.

11.12.8.6 Production Rule-Specific Science Metadata ODL Files

The production rule-specific science metadata ODL files provide specific information to PDPS about production rules used by a PGE. They are needed only when the PGE is subject to one of the following conditions:

- Is executed on the basis of a satellite orbit.
- Needs to know the orbital path of a satellite.
- Requires data based on geographic tiling of the Earth.

Since not every PGE is based on orbits or tiles, not all PGEs require these files. The comments in the PGE_ODL.template describe when setting a specific parameter means that a production rule-specific science metadata ODL file needs to be created.

The production rule-specific science metadata ODL files are broken into three types, which are defined as follows:

- Orbit ODL File.
  - Defines the orbital period of the satellite from which the PGE's input data is created.
  - Defines when a given orbit starts, how long it lasts, and the number of the orbit.
  - PDPS uses the information in the orbit ODL file to extrapolate future orbits and is able to plan PGEs that are required to run every so many orbits of the satellite.
- Pathmap ODL File.

  - Defines the mapping between the cyclic 0-233 orbits that the satellite makes with the actual path number that the PGE requires.
  - PDPS computes the path number from the orbit number (specified in the orbit ODL file) by incrementing it until it reaches the 233 maximum, then resetting it to zero.
  - Many instruments expect the path number to be a fixed swath on the Earth, so it is not just incremented for each satellite pass.
  - The pathmap ODL file creates a mapping from the sequential path numbers to the path numbers expected by the PGEs.
- Tile ODL File.
  - Defines the coordinates of the tiles used by some instruments to specify geographic locations on the Earth.
  - The tile definitions are used by PDPS to schedule the PGE (one execution per tile) and to acquire the necessary data (using the geographic coordinates to acquire data for the tile being processed only).

Unlike the PGE science metadata ODL file, there is no tool to automatically generate a template production rule-specific science metadata ODL file. Because the files themselves tend to be small, this is not usually a problem. A template version of each kind of production rule-specific science metadata ODL file (e.g., ORBIT_ODL.template, TILE_ODL.template) exists in the /usr/ecs/<MODE>/CUSTOM/data directory on the AIT Workstation. The templates must be copied, named properly, and edited in order to create the appropriate production rule-specific science metadata ODL file.

11.12.8.7 Release 5 Production Rules

The following statements provide some simplified descriptions of production rules that are scheduled to be made available in Release 5:

- **Basic Temporal** - Temporal (time) range of inputs matches the temporal range of outputs.
- **Advanced Temporal** - Temporal range of inputs is offset from the expected temporal range of inputs and outputs.
- **Alternate Input** - PGE is run with different inputs based on the availability of various alternate input data sets.
- **Optional Input** - PGE is run with specified optional inputs if available; otherwise, PGE is run without them.
- **Minimum/Maximum Number of Granules** - Minimum number of input granules needed for full data coverage and maximum number of input granules to search for may be specified. Minimum and maximum number of outputs expected from the PGE may be specified.

- **Optional DPRs** – The only DPRs executed are those for which the non-routine key input data actually become available (i.e., are either produced in data processing or can be acquired from the archive).
- **Intermittent Activation** - Every n^{th} DPR is activated; all other DPRs are skipped.
- **Metadata Checks** - DPR is run only if input data's metadata value(s) meet(s) certain criteria.
- **Metadata Query** - Input granule selection is based on metadata value.
- **Data Day** - Input data selection is based on Data Day.
- **Spatial Query** - Input granule selection is based on the spatial coverage of another input (i.e., the key input).
- **Tiling** - Input data is chosen on the basis of Instrument Team-defined tiles (geographic areas).
- **Closest Granule** – DPR is generated if a required input granule within a particular time range (rather than an exact time) is available; otherwise, no DPR is generated. (Supersedes the Most Recent Granule Production Rule)
- **Orbital Processing** - Selection of input times is based on orbit information.

11.12.8.8 Basic Temporal Production Rule

The Basic Temporal Production Rule defines the timeframe for the PGE along with its input and output data. PGEs subject to the Basic Temporal Production Rule generally have the following characteristics in common:

- Typically scheduled to run using input data that become available periodically (every hour, every day, etc.).
- Use input data for a particular period of time.
- Produce output for a specified length of time.

The data the PGE takes in (its input) and the data it produces (its output) have the same period (or some subset of the same period) as the PGE.

- Example One:
  - A MODIS PGE processes data for five-minute intervals, producing Level 1B granules.
  - The PGE requires as input the specific five-minute Level 1A granule that is contemporaneous with (covers the same five-minute time period as) the Level 1B granule to be produced.
  - Using the Basic Temporal Production Rule, a five-minute Level 1A granule is staged as input to the PGE and a five-minute Level 1B granule is expected as output, both matching the timeframe for which the PGE is run.
- Example Two:
  - A CERES PGE processes data for 24-hour intervals, producing 24-hour Level 1A granules as output.
  - As input the PGE takes Level 0 data that is ingested every two hours.

- Using the Basic Temporal Production Rule, twelve two-hour Level 0 granules are staged as input to the PGE and a 24-hour Level 1A granule is expected as output, matching the timeframe for which the PGE is run.

The fundamental elements used to define the Basic Temporal Production Rule are “period: and “boundary.”

- **Period** is the length of time for which a PGE processes data or the length of time for which input and output data is collected.
  - A PGE that is subject to the Basic Temporal Production Rule only and that processes data in two-hour blocks, takes in data that relates to a particular two-hour interval and produces output data for that same two-hour period.
  - Data that has a period of 15 minutes was collected or produced for a 15-minute time period.
- **Boundary** is the starting point for the data or PGE. Depending on the characteristics of the data or PGE, the boundary may be the start of a minute or hour or day or week (etc.).
  - If a PGE's boundary is the start of the hour, it processes data that starts every hour and runs on data for the length of its period.
  - If data comes in every day, PDPS predicts that the data is going to be available at the start of the day and allows scheduling of PGEs that use the data as input accordingly.

Both the PGE itself and the input data have a boundary and period associated with them. That is how PDPS determines the frequency of processing for a Basic Temporal PGE and the time period for its inputs and outputs.

PDPS uses **period** and **boundary** in combination to plan the processing of each PGE, including determining its input requirements and anticipated output (which may be input to other PGEs). If a PGE has a period of one hour and a boundary of “start of day,” it is scheduled every hour, beginning at midnight. If an input has a period of 15 minutes and boundary of “start of hour,” PDPS predicts it every 15 minutes beginning on the hour.

Boundary offset is an addition to the Basic Temporal Production Rule that allows a PGE or data to start on an offset from a given boundary. For example, if a PGE would normally run every day but not start until two or three hours into the day (e.g., beginning at 3:00 a.m. instead of midnight), a boundary offset can be used to add three hours to the “start of day” boundary. This would mean the PGE would run on data that occurred three hours after the boundary.

Data with offset times refers to data where the start time is a few minutes off of the start time that PDPS expects. For example: if data is defined to PDPS as follows:

```
BOUNDARY = "START_OF_HOUR"
PERIOD   = "HOURS=1"
```

but the data actually starts at 1:05 and ends at 2:05, the data is said to have **offset times**. There is a flag in the production rule syntax that tells PDPS to shift granule time specifications to match the granules in the archive.

The **end-of-month anomaly** is an addition to the Basic Temporal Production Rule that allows a PGE or data to cover a specific number of days within a month. The month is broken into thirds. The first third is composed of the first 10 days of the month. The second third consists of days 11 through 20. And the last third varies in length depending on the total number of days in the month (i.e., for November it would have 10 days; for December it would have 11 days). A specific **boundary** and **period** allow a PGE or its data to be scheduled into thirds of a month.

Figure 5 provides an illustration of the Basic Temporal Production Rule. The PGE has a boundary of "start of day" and a period of one hour, so it is scheduled for every hour through the day. If a Production Request were entered for two full days of processing, a DPR would be created for the PGE to run every hour; i.e., 48 DPRs total. If a Production Request were created for a four-hour period in the middle of a single day (for example, from 12:00 noon to 4:00 p.m.), then four DPRs would be created, one for 12:00-1:00, one for 1:00-2:00, one for 2:00-3:00, and one for 3:00-4:00.

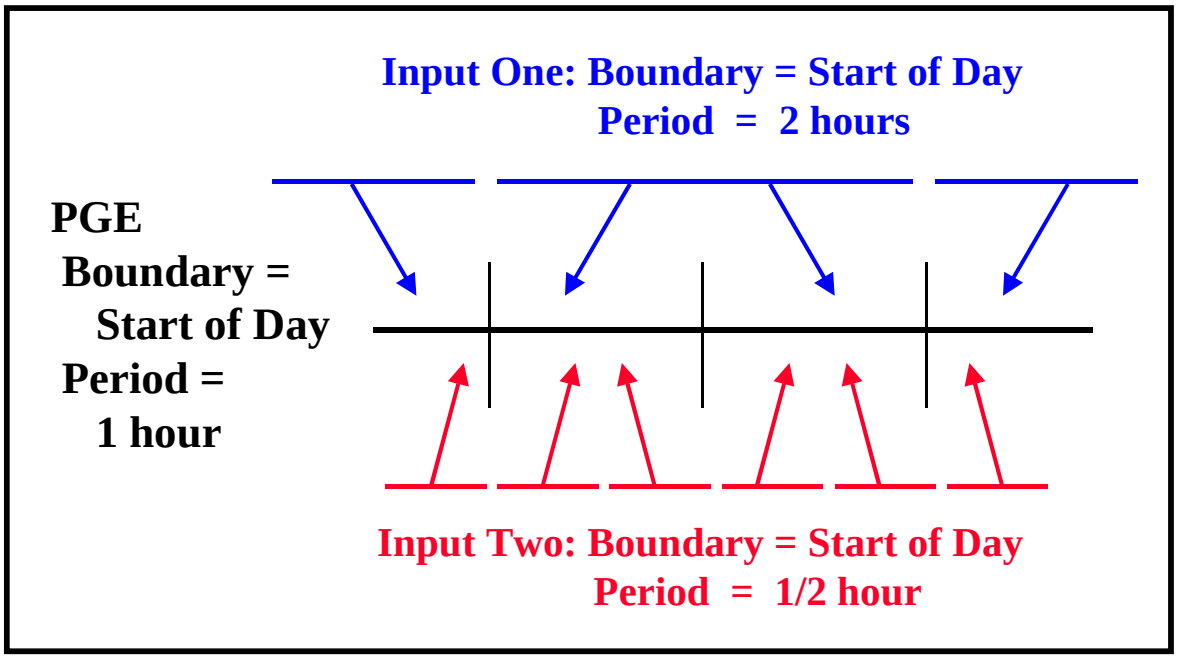


Figure 11.12.8.8-1. Example of the Basic Temporal Production Rule

In the example (Figure 11.12.8.8-1), Input One has a boundary of "start of day" and a period of two hours, so when PDPS plans for its availability, it expects a granule every two hours

beginning at midnight. Consequently, each granule of Input One is associated with two DPRs for the PGE, because the PGE encompasses only one hour of the two-hour granule's period.

Input Two has a boundary of “start of day” and a period of ½ hour, so when PDPS plans for its availability, it expects a granule every ½ hour beginning at midnight. As a result two granules of Input Two are associated with each DPR for the PGE, because the PGE encompasses an hour of the ½-hour granule's Period. Thus, every DPR of the PGE will wait for two granules of Input Two to arrive before it can be processed.

11.12.8.9 PGE Science Metadata ODL File Parameters

The following parameters must be set properly in the applicable PGE science metadata ODL file in order to implement the Basic Temporal Production Rule:

- SCHEDULE_TYPE.
- PROCESSING_PERIOD.
- PROCESSING_BOUNDARY.

The SCHEDULE_TYPE parameter specifies the type of scheduling that will be done for the PGE. The following values are applicable to the Basic Temporal Production Rule:

- "Time"
  - The PGE is scheduled on the basis of the specified boundary/period and the availability of data for that boundary/period.
- "Snapshot"
  - The PGE is scheduled for a single date/time.
  - Note that PROCESSING_PERIOD and PROCESSING_BOUNDARY are **not** needed when "Snapshot" is specified.

Other values for SCHEDULE_TYPE apply to other production rules, such as the following values:

- “Data"
  - The PGE is scheduled on the basis of the availability of data produced by other PGEs.
- "Tile"
  - The PGE is scheduled based on the definition of geographic tiles.
- "Orbit"
  - PGE scheduling is based on the orbit of the spacecraft.

The PROCESSING_PERIOD parameter describes the length of time for which the PGE executes. Data will be acquired (barring any combination of Production Rules) for the specified period and output data will be planned for the given period. It is of the format "<Period Type>=<Length of Period>". Note that “length of period” can be specified as a positive integer only. The following values are acceptable “period type” entries for the Basic Temporal Production Rule:

- "YEARS"
  - PGE processes data applicable to a given year or years.
  - "YEARS" might be specified for a PGE that computes a yearly average.
  - For example, PROCESSING_PERIOD = "YEARS=1" relates to a PGE that processes one year's worth of data.
- "MONTHS"
  - PGE processes data applicable to a particular month or several months.
  - "MONTHS" is most likely to be used for some kind of averaging PGE.
  - For example, PROCESSING_PERIOD = "MONTHS=2" relates to a PGE that processes two months' worth of data at a time.
- "THIRDS"
  - PGE processes data applicable to some number of thirds of the month.
  - For example, PROCESSING_PERIOD = "THIRDS=1" relates to a PGE that processes data applicable to 1/3 of the month.
- "WEEKS"
  - PGE processes data applicable to some number of weeks.
  - For example, PROCESSING_PERIOD = "WEEKS=2" relates to a PGE that processes two weeks' worth of data every time it runs.
- "DAYS"
  - PGE processes data applicable to some number of days.
  - For example, PROCESSING_PERIOD = "DAYS=5" relates to a PGE that processes five days' worth of data.
- "HOURS"
  - PGE processes data applicable to some number of hours.
  - For example, PROCESSING_PERIOD = "HOURS=4" relates to a PGE that processes four hours' worth of data when it is executed.
- "MINS"
  - PGE processes data applicable to some number of minutes.
  - For example, PROCESSING_PERIOD = "MINS=5" relates to a PGE that processes five minutes' worth of data.
- "SECS"
  - PGE processes data applicable to some number of seconds.
  - For example, PROCESSING_PERIOD = "SECS=2" relates to a PGE that runs on two seconds' worth of data.

There are other types of values for PROCESSING_PERIOD but they apply to other production rules (as described in the applicable sections of the lesson).

The PROCESSING_BOUNDARY parameter specifies the boundary (starting point in time) of the PGE. It tells when each instance of the PGE should start. Note that the PROCESSING_BOUNDARY and PROCESSING_PERIOD are used in conjunction to schedule the PGE.

The following PROCESSING_BOUNDARY values are used for implementing the Basic Temporal Production Rule:

- "START_OF_HOUR" – PGE processes data for each hourly interval.
- "START_OF_6HOUR" - PGE processes data for every 6-hour interval.
- "START_OF_DAY" - PGE processes data for every daily interval.
- "START_OF_WEEK" - PGE processes data for every weekly interval.
- "START_OF_ONE_THIRD_MONTH" - PGE processes data for every 1/3 of a month.
- "START_OF_MONTH" - PGE processes data for every monthly interval.
- "START_OF_YEAR" - PGE processes data for every yearly interval.
- "START_DATE=*DD/MM/YYYY*" - PGE processes data for the specified date only.

There are other values for PROCESSING_BOUNDARY that apply to other production rules (as described in the applicable sections of the lesson).

11.12.8.10 Handling Data with Offset Times

When the ALIGN_DPR_TIME_WITH_INPUT_TIME flag is set to "Y" (i.e., ALIGN_DPR_TIME_WITH_INPUT_TIME = “Y”) PDPS shifts the expected times for input data to the actual times found in the archive. If the flag is NOT set, data with offset times can cause problems when generating Production Requests.

11.12.8.11 ESDT Science Metadata ODL File Parameters

The following parameters must be set properly in the applicable ESDT science metadata ODL file in order to implement the Basic Temporal Production Rule:

- DYNAMIC_FLAG.
- PERIOD.
- BOUNDARY.

The DYNAMIC_FLAG describes the type of data that is defined in the ESDT science metadata ODL file. It specifies to PDPS what kind of data the PGE requires as input or produces as output. It can have any of the following four possible values, all of which are valid for Basic Temporal data:

- "S"
  - Static Data.
  - Data do not change at regular intervals.

  - The same granule can be used as input for many runs of the PGE.
  - Calibration files are a good example of static data.
- "I"
  - Dynamic Internal.
  - Data are produced by a PGE running at the local DAAC.
  - All output products are either “dynamic internal” or “interim” kinds of data.
- "E"
  - Dynamic External.
  - Data are produced by an external source (not a PGE running at the local DAAC).
  - EDOS data is a primary example.
  - Dynamic external can be set for PGE inputs only.
- "T"
  - Interim/Intermediate.
  - Data are stored only temporarily by the Data Server Subsystem.

The PERIOD parameter specifies the length of time covered by the data. Data are expected to be either ingested or produced for the length of the PROCESSING_PERIOD described in PGE science metadata ODL files. However, the PERIOD of the data does **not** have to match the PROCESSING_PERIOD defined for the PGE. PDPS plans for data where the ESDT period is less or more than the processing period of the PGE that uses it. For example, if the PGE PROCESSING_PERIOD = "HOURS=1" and the input data PERIOD = "MINS=5", then PDPS plans to acquire twelve granules of the input data to cover the PROCESSING_PERIOD.

The following “period type” values are used for implementing the Basic Temporal Production Rule:

- "YEARS"
  - Data span a year or years.
  - “YEARS” might be selected for a yearly average output product.
  - For example, PERIOD = "YEARS=1" specifies data that cover a period of a year.
- "MONTHS"
  - Data span a month or several months.
  - “MONTHS” is most likely used for some kind of averaging output product.
  - For example, PERIOD = "MONTHS=2" specifies data that cover a period of two months.
- "THIRDS"

  - Data span some number of thirds of a month.
  - For example, PERIOD = "THIRDS=1" specifies data that cover a period of 1/3 month.
- "WEEKS"
  - Data span some number of weeks.
  - For example, PERIOD = "WEEKS=2" specifies data that cover a period of two weeks.
- "DAYS"
  - Data span some number of days.
  - For example, PERIOD = "DAYS=5" specifies data that cover a period of five days.
- "HOURS"
  - Data span some number of hours.
  - For example, PERIOD = "HOURS=4" specifies data that cover a period of four hours.
- "MINS"
  - Data span some number of minutes.
  - For example, PERIOD = "MINS=5" specifies data that cover a period of five minutes.
- "SECS"
  - Data span some number of seconds.
  - For example, PERIOD = "SECS=2" specifies data that cover a period of two seconds.
- "ORBITS"
  - Data span some number of orbits of the spacecraft.
  - For example, PERIOD = "ORBITS=1" specifies data that cover one orbit.
  - A PGE can be time-scheduled (using the Basic Temporal Production Rule) but use orbit-based data.

The BOUNDARY parameter is the starting point in time of the data granule. It tells when each data granule should start. Note that the BOUNDARY and PERIOD are used in conjunction to determine the starting and ending time for the granules.

The following values for BOUNDARY apply to the Basic Temporal Production Rule:

- "START_OF_HOUR"
  - Data granules start every hour.
- "START_OF_6HOUR"

  - Data granules start every six hours.
- "START_OF_DAY"
  - Data granules start every day.
- "START_OF_WEEK"
  - Data granules start every week.
- "START_OF_ONE_THIRD_MONTH"
  - Data granules start every 1/3 of a month.
- "START_OF_MONTH"
  - Data granules start every month.
- "START_OF_YEAR"
  - Data granules start every year.
- "START_OF_ORBIT"
  - Data granules start every orbit.

11.12.8.12 Advanced Temporal Production Rule

The Advanced Temporal Production Rule allows for input data to be acquired for a time period other than that of the PGE or its planned inputs/outputs. It provides an offset mechanism, specifying on an input basis that the data required for processing is some number of seconds earlier or later than the planned time period for the PGE.

- Example One:
  - A PGE requires data from its previous execution for interpolation purposes (e.g., one of its inputs is the output of the very same PGE the last time that it ran).
  - If the PGE processes data for each one-hour interval (producing an hourly product), the Advanced Temporal Production Rule is specified with an offset of minus 3600 seconds (one hour) for the input of the ESDT produced by previous runs.
- Example Two:
  - A PGE takes as input two-hour Level 0 data to produce an L1A product.
  - Because the edges of the Level 0 data can be difficult to process without preceding and succeeding data, the PGE requires three Level 0 granules, one from the time period before it runs, one for the time period it is currently processing and one for the next time period.
  - The PGE is defined as having three inputs, the first with and Advanced Temporal offset of minus 7200 seconds (two hours), the second with no Advanced Temporal offset and the third with an Advanced Temporal offset of plus 7200 seconds (two hours).

The Advanced Temporal Production Rule uses the times specified in the Basic Temporal Production Rule as a reference point for specifying offset(s) to request data from a “period” and/or “boundary” different from that of the DPR or its input. The offsets are specified as either negative or positive numbers to indicate whether the time period of the input data is before or after that of the DPR (a particular run of a PGE).

- **Begin Period Offset** is an amount of time (in seconds) that is specified with respect to the DPR start time. A negative beginning offset requests data that was collected before the DPR start time. A positive beginning offset requests data with a collection time after the start time of the DPR.
- **End Period Offset** is an amount of time (in seconds) that is specified with respect to the DPR end time. A negative ending offset requests data that ended collection before the DPR end time was reached. A positive ending offset requests data that ended collection after the end time of the DPR boundaries.

Note that the beginning and ending offsets are not absolute cut-offs for data. Overlapping granules (granules that start or end outside of the offsets) will be staged as inputs to the DPR.

Figure 11.12.8.12-1 provides an illustration of the Advanced Temporal Production Rule. The PGE shown in the example processes data for every one-hour interval. However, Input One comes in at two-hour intervals and Input Two is produced every 1/2 hour.

Both the Begin Period Offset and End Period Offset for Input One are -7200 seconds (minus two hours). Consequently, every DPR will stage the "previous" Input One. This could be used to get the "previous" or "next" granule of an input.

The Begin Period Offset for Input Two is zero, meaning that it will match the Start Time of the DPR. The End Period Offset is +1800 seconds (plus 1/2 hour). Therefore, all Input Two granules will be staged that fall within the time period of the DPR plus 1/2 hour. The effect is to acquire all Input Two granules within the time period of the DPR, plus the one from the next 1/2-hour time period, for a total of three granules. The additional granule acquired by means of the End Period Offset might be used for interpolation purposes at the end point.

The same types of parameter settings that apply to the Basic Temporal Production Rule apply to the Advanced Temporal Production Rule. In addition, there are some parameters in the PGE science metadata ODL file that apply specifically to the Advanced Temporal Production Rule. However, the values applicable to the Basic Temporal Production Rule must be set before the Advanced Temporal Production Rule syntax is added.

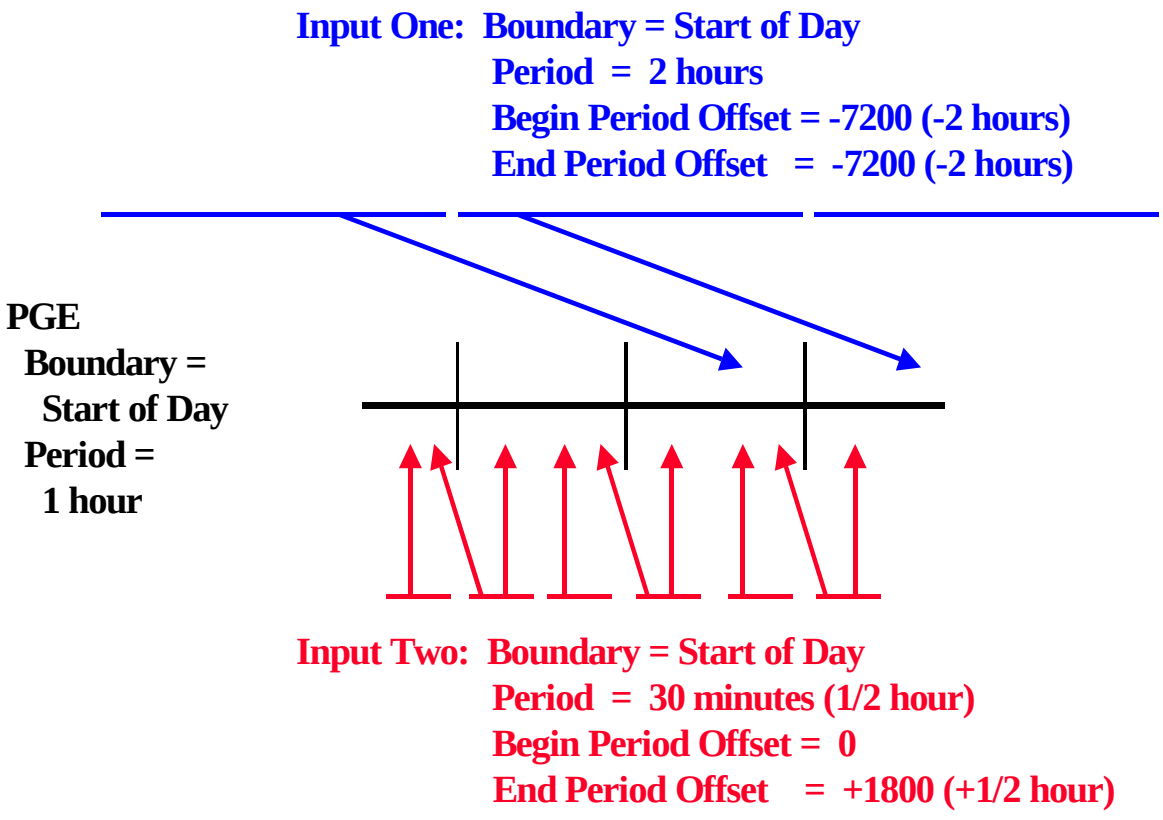


Figure 11.12.8.12-1. Example of the Advanced Temporal Production Rule

11.12.8.13 PGE Science Metadata ODL File Parameters

During the SSI&T process the PGE science metadata ODL file is generated from the PCF delivered with the science algorithm. A PCF_ENTRY object is generated for each file entry in the PCF. In order to implement the Advanced Temporal Production Rule the PCF_ENTRY object for each type of input file to which the rule applies uses the following syntax:

```
OBJECT = PCF_ENTRY
   .
   .
   .
   BEGIN_PERIOD_OFFSET =
   END_PERIOD_OFFSET =
   .
   .
   .
```

```
END_OBJECT = PCF_ENTRY
```

Accordingly, the following parameters must be set properly in order to implement the Advanced Temporal Production Rule:

- BEGIN_PERIOD_OFFSET.
- END_PERIOD_OFFSET.

BEGIN_PERIOD_OFFSET is the offset added to or subtracted from the Data Start Time of the DPR. The value assigned to BEGIN_PERIOD_OFFSET can be either a positive or negative value, specified in seconds. If the value is positive, it is added to the Data Collection Start Time (looking for the input forward in time). If the value is negative, it is subtracted from the Data Collection Start Time (looking backward in time). For example, BEGIN_PERIOD_OFFSET = -3600 requests data that was collected one hour (3600 seconds) before the DPR start time.

END_PERIOD_OFFSET is the offset added to or subtracted from the Data Collection End Time of the DPR. The value assigned to END_PERIOD_OFFSET can be either a positive or negative value, specified in seconds. If the value is positive, it is added to the Data Collection End Time (looking for the input forward in time). If the value is negative, it is subtracted from the Data Collection End Time (looking backward in time). For example, END_PERIOD_OFFSET = +2700 requests data that was collected 45 minutes (2700 seconds) after the DPR end time.

The BEGIN_PERIOD_OFFSET and END_PERIOD_OFFSET parameters can be specified for any input PCF_ENTRY in the PGE science metadata ODL file. If not specified, the parameters are set to zero (0) and the Advanced Temporal Production Rule does not apply to the PGE.

11.12.8.14 Alternate Input and Optional Input Production Rules

The Alternate Input and Optional Input Production Rules are very similar and use much the same processing in PDPS. Both rules allow a PGE to select various inputs based on timers and priority lists. The major difference is that Alternate Inputs requires that one of alternates on the list be used, whereas Optional Inputs allows successful execution of the PGE if no optional input on the list is available.

The Alternate Input Production Rule allows for a PGE to evaluate a list of inputs in priority order and be scheduled and executed with the best priority input that could be found. In essence, a PGE using Alternate Inputs is saying "I would like to run with Input A, but if it's not available, I am willing to run with Input B." A timer can be used to specify how long to wait for a given alternate choice before proceeding with a choice of lesser priority. The PGE is not executed until one of the alternate choices has been found.

- Example:
  - A PGE requires model wind data as an input but is capable of accepting wind data from a Data Assimilation Office (DAO) model, a National Centers for Environmental Prediction (NCEP) model, or (as a last resort) climatology.
  - The PGE would use the Alternate Input Production Rule to list each input in priority order, giving a timer value for how long to wait before trying the next input.

- If the DAO data are most desirable, DAO would be listed as first choice or "primary" data.
- NCEP would be the second choice.
- Climatology would be the last choice.
- If a timer value is specified for DAO data, the PGE will wait for that timer to expire before running with either NCEP data or climatology.
- If a timer had been placed on the NCEP input, the PGE would wait before running with the climatology data.

The Optional Input Production Rule allows for a PGE to list inputs that are desired but not required for it to execute. The inputs are ranked as previously stated and timers are set to wait before choosing a lower-priority type of input. However, if none of the inputs on the list becomes available, the PGE starts because the alternatives are classified as "optional." In essence the PGE is saying "I would like to run with Input A, but if its not available, I can run (and produce reasonable output) without it."

- Example:
  - It would be preferable to run a particular MODIS PGE with the output of a MISR PGE as input.
  - However, the MISR output may not be produced every day.
  - So the MODIS PGE lists the MISR input as optional with a two-hour timer.
  - On those occasions when no MISR output is produced, the MODIS PGE waits for two hours and then is executed without the MISR input.

Figure 11.12.8.14-1 provides an illustration of the Alternate Input Production Rule. The PGE in the illustration has two inputs that are “required” so they must available for the PGE to be run. It also has one input that is “alternate.” The alternate input can be one of three choices, the first choice is the **primary**, then there are second and third choices.

After the pair of required inputs has become available, the alternate inputs are evaluated as follows:

- If the primary alternate is available, it is used as input and the PGE is scheduled for execution.
- There is a one-hour timer on the primary alternate. If the primary alternate is unavailable, the PGE waits until the primary alternate becomes available or the one-hour timer expires, whichever occurs first.
- If the second alternate is available after the timer for the primary alternate has expired, the second alternate is used as input and the PGE is scheduled for execution.

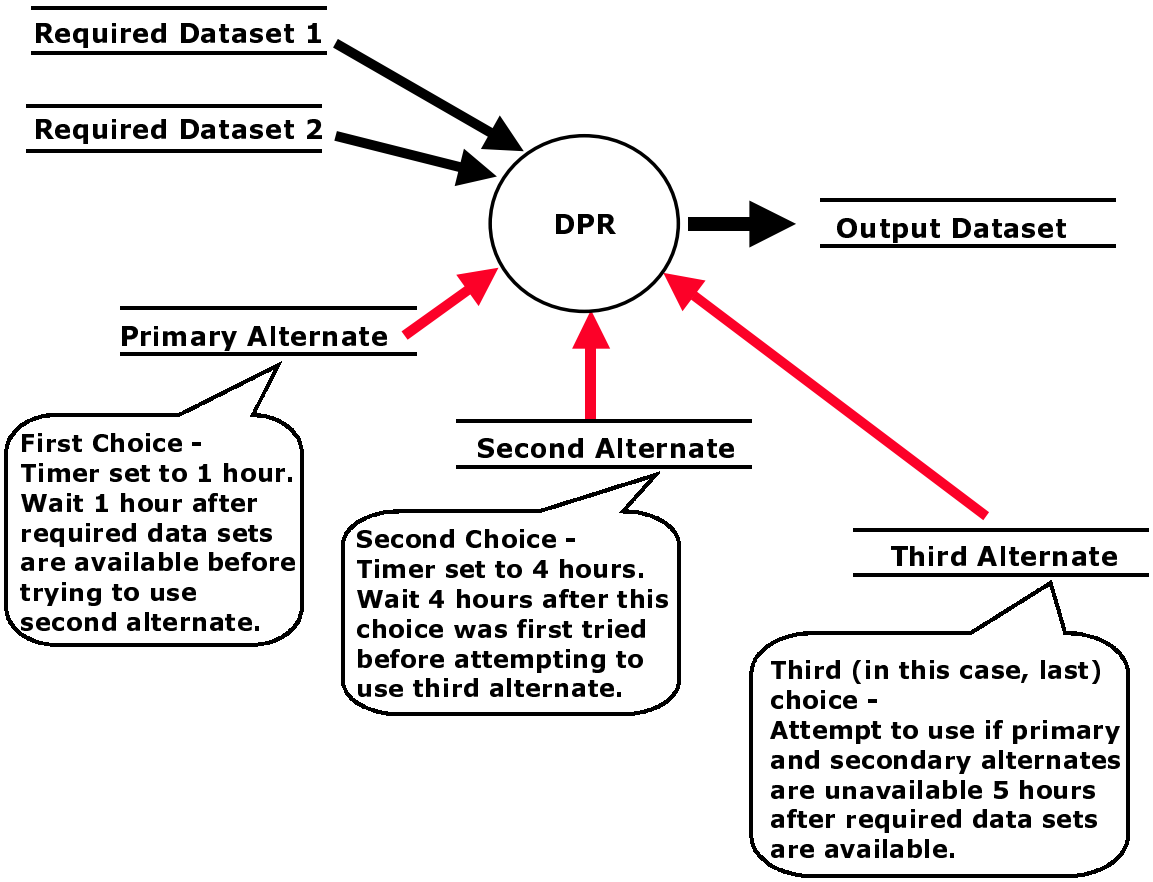


Figure 11.12.8.14-1. Example of the Alternate Input Production Rule

- There is a four-hour timer on the second alternate. If the second alternate is unavailable, the PGE waits until either the primary alternate or the secondary alternate becomes available or the four-hour timer expires, whichever occurs first.
- If the third alternate is available after the timer for the second alternate has expired, the third alternate is used and the PGE is scheduled for execution.
- There is no timer on the third alternate. If the third alternate is not available, the PGE waits until either the primary alternate, the secondary alternate, or the third alternate becomes available, whichever occurs first.
- The PGE will not start processing until one of the alternates becomes available.

If instead of an alternate the third input for the PGE had been defined as an optional input, the preceding scenario would have been the same, except that if neither the primary alternate, the second alternate nor the third option was available after the timers had expired, the PGE would not wait; it would be scheduled for execution without the third input. It would run with the two required inputs only.

The Alternate Input and Optional Input Production Rules are additions to settings/syntax put into the ODL files for other production rules. Inputs deemed “optional” or “alternate” can be searched for and acquired by other production rules (e.g., Basic Temporal or Metadata Checks/Query). The syntax for the rules used to search for the inputs have to be filled out in addition to the syntax required to make the input an alternate or optional input.

11.12.8.15 PGE Science Metadata ODL File Parameters

The following parameter must be set properly in the applicable PGE science metadata ODL file in order to implement the Alternate Input or Optional Input Production Rule:

- INPUT_TYPE.

In addition, one of the following two ODL objects is used within a PCF_ENTRY to define either the Alternate Input Production Rule or the Optional Input Production Rule:

- ALTERNATE_INPUT object.
- OPTIONAL_INPUT object.

INPUT_TYPE is a type of data defined by a PCF_ENTRY object (i.e., between OBJECT = PCF_ENTRY and END_OBJECT = PCF_ENTRY). It can have one of four possible values, only three of which are used to define an alternate or optional inputs:

- "Required"
  - A required input.
  - The data must be available or the PGE does not execute.
  - It is the "normal" value for the parameter (i.e., INPUT_TYPE = “Required”); consequently, the input is neither an alternate input nor an optional input.
- "Primary"
  - The primary alternate input.
  - The data is the first choice in a list of alternates.
- "Alternate"
  - An alternate input (except the primary alternate) in a list of alternates.
  - The data is not the first choice in a list of alternates; it is a subsequent choice if the primary (or a higher-priority alternate) is not available.
- "Optional"
  - An optional input.
  - Availability of the data will be checked and if a timer has been specified, execution of the PGE will wait.
  - The PGE can be executed without the data if it is not available.

Although the Alternate Input and Optional Input Production Rules are similar, there are two different ODL objects used to define them within a PCF_ENTRY; i.e., the ALTERNATE_INPUT object and the OPTIONAL_INPUT object.

The ALTERNATE_INPUT object has the following syntax:

```
OBJECT = PCF_ENTRY
   .
   .
   .
   .
   .
   OBJECT = ALTERNATE_INPUT
      .
      .
      .
   END_OBJECT = ALTERNATE_INPUT
END_OBJECT = PCF_ENTRY
```

The ALTERNATE_INPUT ODL object surrounds an Alternate Input definition. An OBJECT/END_OBJECT pair separates the parameters defining the Alternate Input from the rest of the parameters defining the PCF_ENTRY. The following parameters define an ALTERNATE_INPUT object:

- CLASS.
- CATEGORY.
- ORDER.
- RUNTIME_PARM_ID.
- TIMER.
- WAITFOR.
- TEMPORAL [not implemented].

CLASS is a simple counter used to differentiate the different ALTERNATE_INPUT objects within the file. Since each ALTERNATE_INPUT object resides within a different PCF_ENTRY object, the CLASS for an ALTERNATE_INPUT object can always be 1.

CATEGORY is the name of the list of alternates to which the ALTERNATE_INPUT belongs. The PDPS uses CATEGORY to associate different alternates within a list. CATEGORY can be set to any string value of 20 characters or less (e.g., CATEGORY = "Snow Ice"). Alternates that are part of the same list should have matching CATEGORY values.

ORDER is the numerical place that the particular alternate holds in the list of alternates. The first choice or Primary Alternate (with the INPUT_TYPE = "Primary") should have ORDER = 1.

RUNTIME_PARM_ID specifies the Logical ID (in the PCF) for which the PGE will find the Logical ID of the alternate chosen. Since all alternates must be contained within different PCF_ENTRY objects, they all must have different Logical IDs (but all alternates within the same CATEGORY should have the same value of RUNTIME_PARM_ID). The RUNTIME_PARM_ID parameter specifies the Logical ID of a runtime parameter that the PGE may read to find out which alternate was chosen for the particular execution of the PGE.

The TIMER parameter specifies how long to wait for the particular alternate before checking for the next alternate in the list. The parameter value is expressed in the format "<Period Type>=<Length of Period>". Note that “Length of Period” can be specified as a positive integer only. The Alternate Input Production Rule accepts the following “Period Type” values:

- "WEEKS"
  - PDPS should wait for some number of weeks before searching for the next alternate in the list.
  - For example, TIMER = "WEEKS=2" would make PDPS wait two weeks before checking for the next alternate input.
- "DAYS"
  - PDPS should wait for some number of days before searching for the next alternate in the list.
  - For example, TIMER = "DAYS=5" would make PDPS wait five days before checking for the next alternate input.
- "HOURS"
  - PDPS should wait for some number of hours before searching for the next alternate in the list.
  - For example, TIMER ="HOURS=4" would make PDPS wait four hours before checking for the next alternate input.
- "MINS"
  - PDPS should wait for some number of minutes before searching for the next alternate in the list.
  - For example, TIMER = "MINS=5" would make PDPS wait five minutes before checking for the next alternate input.
- "SECS"
  - PDPS should wait for some number of seconds before searching for the next alternate in the list.
  - For example, TIMER = "SECS=2" would make PDPS wait two seconds before checking for the next alternate input.

The WAITFOR parameter specifies whether or not the PGE can be run without the alternate input. Setting WAITFOR = "N" means that the PGE can run without the input if it cannot be found. In a list of alternate inputs, this would have meaning for the last choice only. If WAITFOR = "Y", the PGE is not executed (even after the last alternate timer expires) until one of the alternates in the list can be found.

The TEMPORAL parameter is an unimplemented feature that would allow for searching for alternates from the same time period but a different date. It is currently stored in the PDPS database but is not used.

The OPTIONAL_INPUT object has the following syntax:

```
OBJECT = PCF_ENTRY
   .
   .
   .
   .
   .
   OBJECT = OPTIONAL_INPUT
      .
      .
      .
   END_OBJECT = OPTIONAL_INPUT
END_OBJECT = PCF_ENTRY
```

The OPTIONAL_INPUT ODL object surrounds an Optional Input definition. An OBJECT/END_OBJECT pair separates the parameters defining the Optional Input from the rest of the parameters defining the PCF_ENTRY. The following parameters define an OPTIONAL_INPUT object:

- CLASS.
- CATEGORY.
- ORDER.
- RUNTIME_PARM_ID.
- TIMER.
- TEMPORAL [not implemented].

The parameters that apply to the Optional Input Production Rule are defined in the same way that the corresponding parameters are defined for the Alternate Input Production Rule. However, note that the Optional Input Production Rule has no WAITFOR parameter. It is irrelevant; in fact, the very essence of the Optional Input Production Rule depends on not “waiting for” the last option but going ahead with the execution of the PGE without the unavailable optional input(s).

Table 11.12.8.5-1. Extract of PGE Metadata ODL File Template Showing Alternate Inputs (1 of 2)

```
>OBJECT = PCF_ENTRY
>   CLASS = 16
>   LOGICAL_ID = 1500
>   PCF_FILE_TYPE = 1
>   DATA_TYPE_NAME = "MOD10L2G"      [MODIS Level 2G Snow Cover]
>   DATA_TYPE_VERSION = "1"          [ESDT versioning in release B.0]
>   DATA_TYPE_REQUIREMENT = 1
>   SCIENCE_GROUP = ""
>   OBJECT = FILETYPE
>     CLASS = 1
>     FILETYPE_NAME = "Single File Granule"
>   END_OBJECT = FILETYPE
>   INPUT_TYPE = "Primary"
>   NUMBER_NEEDED = 1
>   OBJECT = ALTERNATE_INPUT
>     CLASS = 1
>     CATEGORY = "Snow Ice"          [User defined]
>     ORDER = 1                      [This data type is sought first]
>     RUNTIME_PARM_ID = 1509         [Run-time parameter holds LID of alternate]
>     TIMER = "HOURS=6"
>     WAITFOR = "N"                  [Force time-out on wait]
>     TEMPORAL = "N"                 [Use most currently produced]
>   END_OBJECT = ALTERNATE_INPUT
>END_OBJECT = PCF_ENTRY
>
>OBJECT = PCF_ENTRY
>   CLASS = 17
>   LOGICAL_ID = 1501
>   PCF_FILE_TYPE = 1
>   DATA_TYPE_NAME = "MOD10A1"       [MODIS Level 3 Daily Gridded Snow Cover data set]
>   DATA_TYPE_VERSION = "1"
>   DATA_TYPE_REQUIREMENT = 1
>   SCIENCE_GROUP = ""
>   OBJECT = FILETYPE
>     CLASS = 2
>     FILETYPE_NAME = "Single File Granule"
>   END_OBJECT = FILETYPE
>   INPUT_TYPE = "Alternate"
>   OBJECT = ALTERNATE_INPUT
>     CLASS = 2
>     CATEGORY = "Snow Ice"          [User defined]
>     ORDER = 2                      [This data type is sought last]
```

Table 11.12.8.5-1. Extract of PGE Metadata ODL File Template Showing Alternate Inputs (2 of 2)

```
>     RUNTIME_PARM_ID = 1509            [Run-time parameter holds LID of alternate]
>     TIMER = "HOURS=6"                 [Wait 6 additional hours]
>     WAITFOR = "N"
>     TEMPORAL = "N"
>   END_OBJECT = ALTERNATE_INPUT
>END_OBJECT = PCF_ENTRY
>
>OBJECT = PCF_ENTRY
>   CLASS = 18
>   LOGICAL_ID = 1502
>   PCF_FILE_TYPE = 1
>   DATA_TYPE_NAME = "MIANTASC"       [MISR Terrestrial Atmosphere and Surface
Climatology]
>   DATA_TYPE_VERSION = "1"
>   DATA_TYPE_REQUIREMENT = 1
>   SCIENCE_GROUP = ""
>   OBJECT = FILETYPE
>     CLASS = 3
>     FILETYPE_NAME = "Single File Granule"
>   END_OBJECT = FILETYPE
>   INPUT_TYPE = "Alternate"
>   OBJECT = ALTERNATE_INPUT
>     CLASS = 3
>     CATEGORY = "Snow Ice"             [User defined]
>     ORDER = 3                         [This data type is sought last]
>     RUNTIME_PARM_ID = 1509            [Run-time parameter holds LID of alternate]
>     TIMER = ""                        [Don't wait for this one]
>     WAITFOR = "Y"                     [Start anyway]
>     TEMPORAL = "N"
>   END_OBJECT = ALTERNATE_INPUT
>END_OBJECT = PCF_ENTRY
```

Minimum/Maximum Number of Granules Production Rule

The Minimum/Maximum Number of Granules Production Rule makes it possible to specify a range of possible granules for a given input or output for a PGE.

- Inputs.
  - Minimum number of granules the PGE needs for full data coverage.
  - Maximum number of granules for the time period.
- Outputs.

- Minimum number of outputs that the PGE is expected to produce.
- Maximum number of outputs that the PGE is expected to produce.

For example, a PGE processes data for every 90-minute interval, has a period of 90 minutes, and takes as input a granule with a period of two hours.

- In many instances one granule of the input will satisfy the PGE.
- In other instances, because of the way the two-hour and 90-minute periods overlap, the PGE needs two input granules to cover the time period.
- Therefore,…
  - Minimum Number of Granules = 1.
  - Maximum Number of Granules = 2.

The Minimum/Maximum Number of Granules Production Rule is different from most production rules because it works for both input and output granules. It allows the PGE to request of a range of inputs (i.e., 1-10 granules), so that it runs with as few as one granule but with as many as ten granules. If a PGE needs at least three granules of a particular input, the minimum number of granules is defined as three and the PGE is not executed until at least three granules are available.

Optional outputs are defined when the Minimum Number of Granules is set to zero. In such cases the PGE can produce none of the particular type of output and still be considered to have executed successfully. If a PGE has a non-zero value for a Minimum Number of Granules associated with an output, and fails to produce any granules of that output type, it is marked as failed.

Figure 8 provides an illustration of the Minimum/Maximum Number of Granules Production Rule. In the example the PGE processes data related to a one-hour period and takes in both Input 1 and Input 2. Since Input 1 has a PERIOD of 1/2 hour, every PGE run requires two Input 1 granules. Input 2 has a PERIOD of 15 minutes, so there are four Input 2 granules for every PGE run.

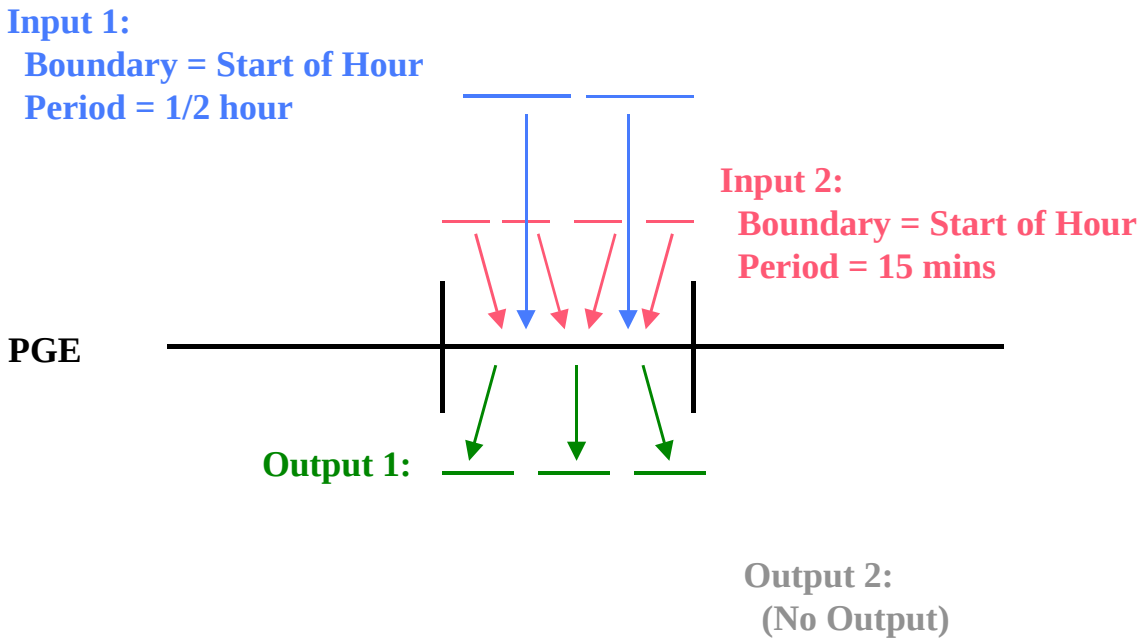


Figure 11.12.8.15-1. Example of the Minimum/Maximum Number of Granules Production Rule

The PGE produces three Output 1 granules for each run. In this case it does not produce any Output 2 granules.

Minimum and maximum values can affect each input and output as follows:

- Input 1:
  - If Minimum Granules is set to anything equal to or less than two for Input 1, the PGE is scheduled and executed.
  - If Minimum Granules is set to three, the PGE is not scheduled because there are not enough Input 1 granules to make the minimum.
  - If Maximum Granules is set to anything equal to or greater than two for Input 1, the PGE is scheduled and executed.
  - If Maximum Granules is set to one, the PGE is not scheduled because there are too many Input 1 granules (the number exceeds the maximum that the PGE can process).

- Input 2:
  - If the Minimum Granules is set to anything equal to or less than four for Input 2, the PGE is scheduled and executed.
  - If Minimum Granules is set to five, the PGE is not scheduled because there are not enough Input 2 granules to make the minimum.
  - If Maximum Granules is set to anything equal to or greater than four for Input 2, the PGE is scheduled and executed.
  - If Maximum Granules is set to three, the PGE is not scheduled because there are too many Input 2 granules (the number exceeds the maximum that the PGE can process).
- Output 1:
  - If Minimum Granules is set to anything equal to or less than three for Output 1, the PGE is scheduled and executes successfully.
  - If Minimum Granules is set to four, the PGE is marked as failed because it did not produce the expected number of output granules.
  - If Maximum Granules is set to anything equal to or greater than three for Output 1, the PGE is scheduled and executes successfully.
  - If Maximum Granules is set to two, the PGE is marked as failed because it produced too many output granules.
- Output 2:
  - If Minimum Granules is set to anything other than zero, the PGE is marked as failed because it did not produce the expected number of output granules.
  - If Maximum Granules is set to anything equal to or greater than zero for Output 2, the PGE is scheduled and executes successfully.

The Minimum/Maximum Granules Production Rules are additions to settings/syntax put into the ODL files for other production rules. All Production Rules have a Minimum and Maximum Granule setting for both inputs and outputs, even though both values may be set to one (1).

11.12.8.16 PGE Science Metadata ODL File Parameters

The PGE science metadata ODL file syntax for implementing the Minimum/Maximum Production Rule for **input** data includes the following types of entries:

```
OBJECT = PCF_ENTRY
   .
   PCF_FILE_TYPE =
   .
   .
   MIN_GRANULES_REQUIRED =
```

```
        MAX_GRANULES_REQUIRED =
        .
        .
        .
    END_OBJECT = PCF_ENTRY
```

Accordingly, the following parameters must be set properly in order to implement the Minimum/Maximum Production Rule:

- PCF_FILE_TYPE.
- MIN_GRANULES_REQUIRED.
- MAX_GRANULES_REQUIRED.

The PCF_FILE_TYPE parameter is defined by integers in the range of 1 to 8 (inclusive). The integers are codes for the following types of files:

- 1 - product input files.
- 2 - product output files.
- 3 - support input files.
- 4 - support output files.
- 5 - user defined runtime parameters.
- 6 - interim/intermediate input files.
- 7 - interim/intermediate output files.
- 8 - temporary input/output.

For inputs (any PCF_ENTRY with a PCF_FILE_TYPE equal to 1, 3 or 6) the following pair of values must be set for each PCF_ENTRY:

- MIN_GRANULES_REQUIRED
  - Minimum number of granules required for the input.
  - A value of zero (MIN_GRANULES_REQUIRED = 0) would mean that the PGE could execute if no granules for that particular input could be found (in effect, the input is an **optional input**).
  - A value of three (for example) would mean that the PGE must have at least three granules of the input before the PGE can be executed.
- MAX_GRANULES_REQUIRED
  - Maximum number of granules for the input that the PGE is able to successfully process.
  - A value of four (for example) would mean that the PGE would process at most four granules for the input.
  - If MAX_GRANULES_REQUIRED = 4 and more than four granules are found for the given input, the PGE is not executed.

The PGE science metadata ODL file syntax for implementing the Minimum/Maximum Production Rule for **output** data includes the following types of entries:

```
OBJECT = PCF_ENTRY
   .
   PCF_FILE_TYPE =
   .
   .
   MIN_GRANULE_YIELD =
   MAX_GRANULE_YIELD =
   .
   .
   .
END_OBJECT = PCF_ENTRY
```

For outputs (any PCF_ENTRY with a PCF_FILE_TYPE equal to 2, 4 or 7) the following pair of values must be set for each PCF_ENTRY.

- MIN_GRANULE_YIELD
  - Minimum number of granules that the PGE produces for the output.
  - A value of zero (MIN_GRANULE_YIELD = 0) means that the PGE produces no granules for the output (the output is an **optional output**).
  - A value of three (for example) means that the PGE produces at least three granules of the output during a successful execution.
- MAX_GRANULE_YIELD
  - Maximum number of granules that the PGE produces for this output.
  - A value of four (for example) means that at most the PGE produces four granules for the output.
  - Note that sizing of disk space is based on this number, so making it too small could cause problems on the science processor disks.

11.12.8.17 Optional DPRs Production Rule

The Optional DPRs Production Rule (also called the Data-Scheduled Production Rule) makes the execution of a PGE subject to the availability of a **key input**. The system generates DPRs for every possible instance of the key input data but executes only the DPRs for which data are either produced in data processing or can be acquired from the archive.

The Optional DPRs Production Rule applies to PGEs that process certain kinds of **non-routine data**.

- **Routine Data**
  - Data that can be predicted, that come in at specific intervals and are always of a specified length.
  - Routine data makes it possible for the Basic Temporal Production Rule to schedule PGEs based on their input data.
- **Non-Routine Data**

- Data that cannot be predicted because they come in at random periods and/or their length is variable.
- Examples include an "optional" output of an upstream PGE, or data that are archived at random periods (e.g., some forms of ASTER data).

An Optional DPR has as its **key input** a non-routine data type. There are two sets of circumstances that lead to the scheduling of Optional DPRs:

- Every possible time that the input is produced in data processing (i.e., the key input is produced as an "optional" output by an upstream PGE).
- Whenever a new granule (of a particular data type) can be acquired from the archive (e.g., archived data that were inserted at unpredictable times).

An example of the first condition starts with a MODIS PGE that produces a certain product only when the input data were collected during the satellite's "Day" mode. A second MODIS PGE is scheduled to use the optional ("Day"-mode) product from the first MODIS PGE as its key input. The second MODIS PGE is scheduled to run after every instance of the first MODIS PGE; however, only the DPRs that can use the optional products resulting from runs of the first MODIS PGE are executed. The remaining DPRs cannot be executed because there is no input data for them.

The second condition is illustrated by ASTER routine processing, which makes use of the Optional DPRs Production Rule to schedule and execute ASTER PGEs for new data that have been archived. (Note that the DAAC ingests and archives ASTER production data from tapes supplied by the ASTER Ground Data System on a frequent but not entirely predictable basis.) When the Production Planner creates a Production Request for an ASTER PGE, it is necessary to specify the **insertion time** range (i.e., the time period when the desired data were archived) as opposed to the **collection time** (when the satellite instrument gathered the data). DPRs specifying the ASTER PGE are scheduled and executed for the data granules that were actually inserted in the archive during the time period specified in the Production Request.

An illustration of the Optional DPRs production rule is presented in Figure 11.12.8.17-1. In the figure there are two DPRs (i.e., DPR-1 and DPR-2) for the upstream PGE and two DPRs (i.e., OPT-1 and OPT-2) for the PGE subject to the Optional DPRs Production Rule. The "Optional DPRs" PGE takes as input the optional output of the upstream PGE. When it is executed, DPR-1 produces the optional output, so the dependent DPR (OPT-1) is executed. However, OPT-2 is not executed because DPR-2 (on which OPT-2 depends) does not produce the optional output.

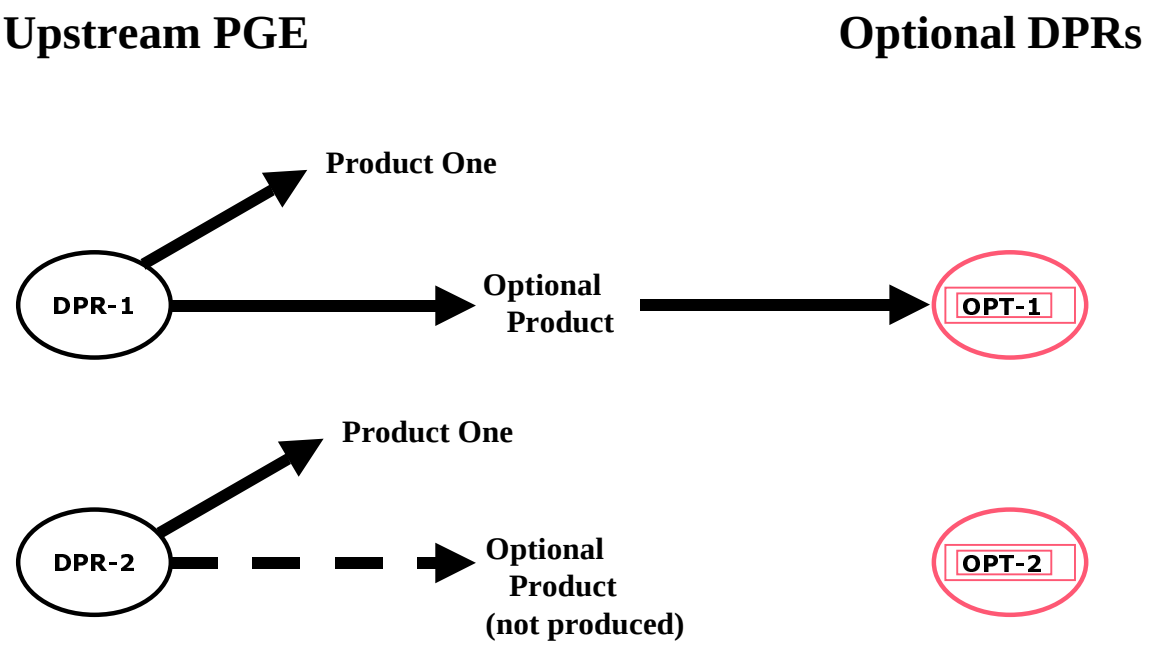


Figure 11.12.8.17-1 . Example of the Optional DPRs Production Rule

The Optional DPRs Production Rule is set up during the SSI&T process. It uses many of the same parameter settings as the Basic Temporal Production Rule so the values specified in the Basic Temporal Production Rule (or other production rules) are set first, then the Optional DPRs Production Rule syntax is added.

11.12.8.18 PGE Science Metadata ODL File Parameters

The following two types of PGE science metadata ODL file entries must be made in order to set up the Optional DPRs Production Rule:

- SCHEDULE_TYPE.
- KEY_INPUT.

The SCHEDULE_TYPE parameter is set as follows:

- SCHEDULE_TYPE = “Data”
  - This demonstrates the appropriateness of the term “Data-Scheduled Production Rule.”
  - Other schedule types include Time, Tile, Orbit, and Snapshot.

The key input is designated by including the following parameter in the PCF_ENTRY for whichever input is to be the key input:

- KEY_INPUT = “Y”
  - Assigning a value of “Y” to the KEY_INPUT parameter identifies the data as a key input and it is subsequently treated as such.
  - Either assigning a value of “N” to the KEY_INPUT parameter or leaving out the parameter entirely identifies the non-key input data.
  - Only one key input is allowed per PGE profile.

The Production Planner’s role in the implementation of the Optional DPRs Production Rule was described in the MODIS and ASTER examples previously described and varies with the kind of key input:

- Optional output of an upstream PGE (MODIS example).
  - Production Planner creates Production Requests for the PGE subject to the Optional DPRs Production Rule and specifies the same date/time range as for the upstream PGE.
  - Some of the DPRs generated as a result of the Production Request will never run due to lack of input data.
- Ingested on an irregular time schedule (ASTER example).
  - Production Planner specifies the data **insertion time** range when creating Production Requests.
  - All DPRs generated as a result of the Production Requests should be capable of running.

11.12.8.19 Intermittent Activation Production Rule

The conditions for executing most PGEs are well defined. The most common activation condition is the availability of all input data sets. Similarly, the frequency of execution is usually well defined (e.g., run once for every granule or run monthly averages once a month). However, some PGEs have additional or different constraints on when they are run.

A PGE can be set up to run on every n^{th} instance of input data. For example, a QA PGE that is run on a daily product may need to be run only every fifth day to provide a spot check. Note that this does **not** refer to the common case of running a weekly averaging PGE only once each week, which would be handled by the Basic Temporal Production Rule and the time ranges specified for the input and output ESDTs. Rather, this is a special case where a PGE **can** be run every day (or hour, week, etc.), but for some reason (such as a QA check) it is desired to run the PGE only every n^{th} day.

To implement the Intermittent Activation Production Rule the Production Planner supplies the following information (via the Production Request Editor) when creating a production request:

- **Number to Skip**
  - Number of DPRs to be skipped (not executed).
  - Entered in the **Skip** field on the Production Request Editor.

- **Number to Keep**
  - After skipping the specified number of DPRs, how many are to be kept?
  - Entered in the **Keep** field on the Production Request Editor.
  - The number to keep is usually one but could be any number.
- **Skip First**
  - Button on the Production Request Editor.
  - Selected to skip the first DPR.
  - Not selected if the first DPR is to be run.

The Planning Subsystem uses the preceding information to establish a pattern of execution. The pattern is effective for the single PR in which the "number to skip" and the "number to keep" are specified; it is not maintained between PRs.

The following example of the Intermittent Activation Production Rule is shown in Figure 10:

- The Production Planner prepares a production request for a 14-day period, generating 14 DPRs.
- The Production Planner made the following selections on the Production Request Editor:
  - Entered "4" in the **Number to Skip** field.
  - Entered "1" in the **Number to Keep** field.
  - Did **not** select the **Skip First** button.
- Consequently, the following results are obtained:
  - First DPR runs.
  - Four DPRs (second through fifth) are skipped.

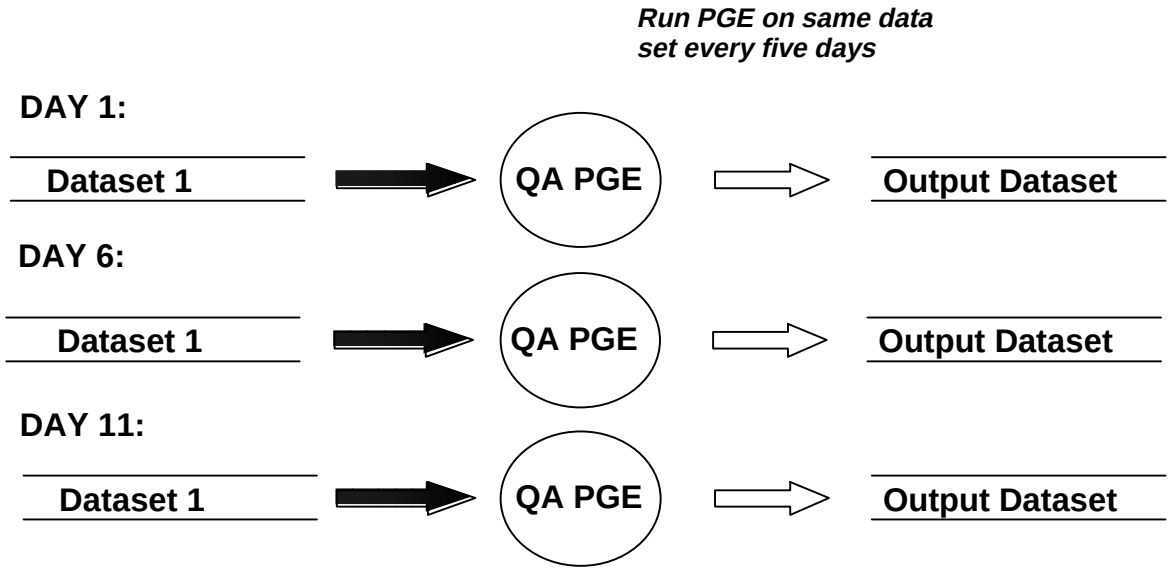


Figure 11.12.8.19-1. Example of the Intermittent Activation Production Rule

- Sixth DPR runs.
- Four DPRs (seventh through tenth) are skipped.
- Eleventh DPR runs.
- Remaining three DPRs (twelfth through fourteenth) are skipped.

11.12.8.20 Metadata Checks and Metadata Query Production Rules

The Metadata Checks and Metadata Query Production Rules are similar in definition and use. Both production rules allow the PGE to specify granule-level metadata values that define whether the PGE can accept one (or more) of its inputs. The rules differ only in the results of metadata search performed.

- Metadata Checks Production Rule.
  - When PLS requests the Science Data Server to search for the input(s), the Science Data Server "checks" the metadata of all granules that match the time frame with respect to the value(s) allowed by the PGE.
  - If any granule fails to match the specified value(s), the PGE is not executed.
- Metadata Query Production Rule.
  - When PLS requests the Science Data Server to search for the input(s), the Science Data Server adds to the query the metadata value(s) desired by the PGE.

  - Only the granules that match the time frame of the PGE plus the granule-level metadata value(s) specified by the PGE are staged for the PGE to use as input.
  - If no granules are found matching the conditions and the input is not optional, the PGE is not executed.
- Example of Metadata Checks:
  - A MODIS PGE is run when the Percent Cloud Cover of its inputs is greater than 25 percent.
  - The Metadata Checks Production Rule is used to specify the granule-level metadata value of greater than 25.
  - When the PGE is scheduled and is ready to start, two granules match the timeframe of the Production Request for the input with the Metadata Check.
  - If both granules have a Percent Cloud Cover greater than 25 percent, execution of the PGE starts and both granules are staged.
  - If one of the granules has a Percent Cloud Cover of 15 percent, the PGE is not executed.
- Example of Metadata Query:
  - A MODIS PGE is run when as many granules as possible of one of its inputs have a QA Value = "Good".
  - The Metadata Query Production Rule is used to specify the granule-level metadata value = "Good".
  - When the PGE is scheduled and is ready to start, two granules match the time frame of the production request for the input with the Metadata Query.
  - If both granules have a QA Value = "Good", execution of the PGE starts and both granules are staged.
  - If one of the granules has a QA Value = "Bad", the PGE executes but with only one granule (the one with QA Value = "Good").

The Metadata Checks and Metadata Query Production Rules are used in conjunction with the times specified in the Basic Temporal Production Rule or other production rules. The Metadata Check or Query is added information that further refines what granules are sought by the PGE.

Multi-Granule ESDTs are a special case of the Metadata Query Production Rule. Multi-Granule ESDTs are used for PGE inputs or outputs when more than one granule of the same ESDT exists for the same temporal range (time period). The Multi-Granule ESDT mechanism employs a metadata parameter to differentiate between the "equal in time" granules. A metadata parameter is selected that is unique across granules for the same time period and that is used by PDPS to keep track of which granule is which when the granules are produced. Later, if only one of a pair of granules for a particular time period is needed as input to the PGE, the Metadata Query is used to ensure that PDPS schedules the correct granule as input.

The **Data Day Production Rule** is actually an addition to the Metadata Query Production Rule involving runtime parameter values. There is a pair of settings (Start Data Day and End Data Day) that allow a PGE to perform a Metadata Query for the start of the Data Day and the end of the Data Day. A separate section of this lesson is devoted to the Data Day Production Rule.

Using runtime parameter values is a capability of the Metadata Query and Metadata Checks Production Rules. Rather than use a hard-coded value for the check or query, a value computed from one of the other production rules can be used.

Figure 11 illustrates the Metadata Checks and Metadata Query Production Rules. If no Metadata Check or Query were applicable, the PGE shown in the figure would use three granules of input (i.e., Granules A through C). However, let us assume that the metadata value to be checked/queried is %CloudCover. Each granule has a different value for %CloudCover.

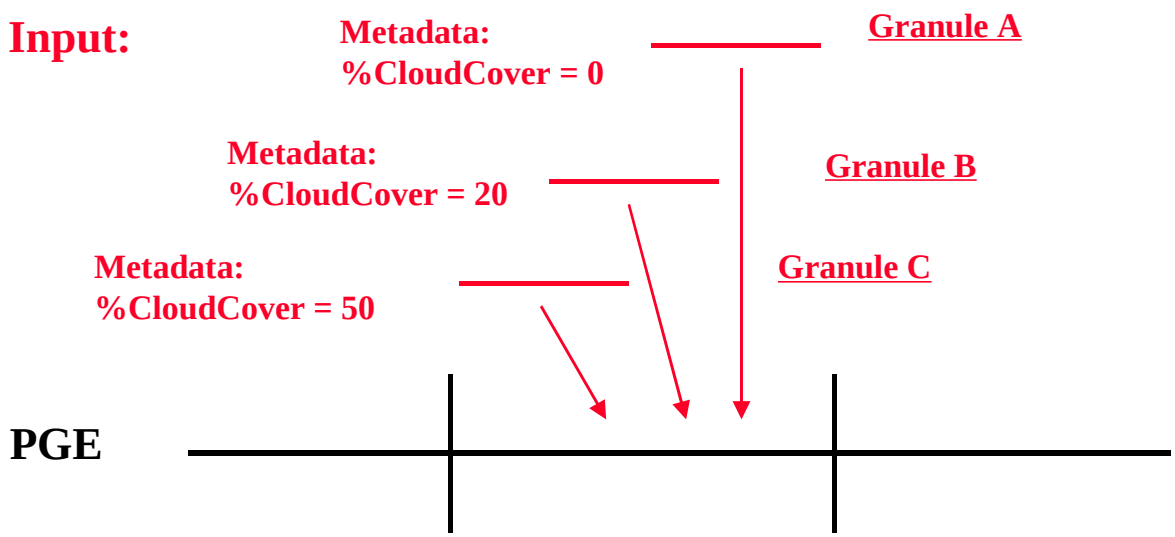


Figure 11.12.8.20-1. Example of the Metadata Checks and Query Production Rules

The following results demonstrate the differences between the Metadata Checks and Metadata Query Production Rules, especially with respect to the number of inputs that the PGE receives when different values are specified:

- Metadata Check of %CloudCover < 80:
  - In this case all three granules are acquired and the PGE is scheduled and executed.
- Metadata Query of %CloudCover < 80:
  - All three granules are acquired and the PGE is scheduled and executed.
- Metadata Check of %CloudCover = 50:

  - The PGE is not scheduled because only one of the three granules (Granule C) meets the criterion.
- Metadata Query of %CloudCover = 50:
  - Granule C is found and if the PGE's Min/Max Granules parameters are set to allow one granule, that one granule is acquired and the PGE is scheduled and executed.
- Metadata Check of %CloudCover < 50:
  - The PGE is not scheduled because only two of the three granules (Granule A and B) meet the criterion.
- Metadata Query of %CloudCover < 50:
  - Granules A and B are found and if the PGE's Min/Max Granules parameters are set to allow two granules, the granules are acquired and the PGE is scheduled and executed.
- Metadata Check of %CloudCover <= 50:
  - The PGE is scheduled and executed because all three granules meet the criterion.
- Metadata Query of %CloudCover <= 50:
  - All three granules are found and acquired and the PGE is scheduled and executed.
- Metadata Check of %CloudCover = 20:
  - The PGE is not scheduled because only one of the three granules (Granule B) meets the criterion.
- Metadata Query of %CloudCover = 20:
  - Granule B is found and if the PGE's Min/Max Granules parameters are set to allow one granule, the granule is acquired and the PGE is scheduled and executed.
- Metadata Check of %CloudCover < 20:
  - The PGE is not scheduled because only one of the three granules (Granule A) meets the criterion.
- Metadata Query of %CloudCover < 20:
  - Granule C is found and if the PGE's Min/Max Granules parameters are set to allow one granule, the granule is acquired and the PGE is scheduled and executed.
- Metadata Check of %CloudCover = 10:
  - The PGE is not scheduled because none of the three granules meets the criterion.
- Metadata Query of %CloudCover = 10:
  - The PGE is not scheduled because no granules are returned from the query (unless Minimum Granules is set to 0).

Note that there can be more than one Metadata Check or Metadata Query on a given input. In the preceding example, a Metadata Check on %CloudCover can be combined with a Metadata Query on another parameter to further limit the input.

The Metadata Checks and Metadata Query Production Rules are additions to settings/syntax put into the ODL files for other production rules. The addition of a Metadata Check or a Metadata Query to an input means that other production rules used to evaluate that input will be applied in combination with the Metadata Check or Metadata Query.

11.12.8.21 PGE Science Metadata ODL File Parameters

Although the Metadata Checks and Metadata Query Production Rules are similar, there are two different ODL objects used to define them within a PCF_ENTRY in the PGE science metadata ODL file; i.e., the METADATA_CHECKS object and the METADATA_QUERY object.

The METADATA_CHECKS object has the following syntax:

```
OBJECT = PCF_ENTRY
  .
  .
  .
  .
  .
  OBJECT = METADATA_CHECKS
    .
    .
    .
  END_OBJECT = METADATA_CHECKS
END_OBJECT = PCF_ENTRY
```

The METADATA_QUERY object has the same syntax except “METADATA_QUERY” replaces “METADATA_CHECKS” in every instance.

Most of the following parameters must be set in the PGE science metadata ODL file within the METADATA_CHECKS or METADATA_QUERY ODL object (as applicable) in order to implement either the Metadata Checks or Metadata Query Production Rule:

- CLASS.
- PARM_NAME.
- OPERATOR.
- VALUE.
- DATABASE_QUERY.
- KEY_PARAMETER_NAME (optional).
- KEY_PARAMETER_VALUE (optional).

CLASS is a simple counter used to differentiate the different Metadata Checks or Metadata Query objects within the file. Since each Metadata Checks or Metadata Query object resides

within a different PCF_ENTRY object, the CLASS for an METADATA_CHECKS or METADATA_QUERY object can always be 1 (e.g., CLASS = 1).

PARM_NAME is the name of the metadata parameter on which the check or query is to be performed. The value specified for PARM_NAME (e.g., PARM_NAME = “%CloudCover”) must be part of the granule-level metadata of the ESDT. In addition, it must match the parameter name specified in the ESDT science metadata ODL file.

OPERATOR is the operator (e.g., OPERATOR = “==”) on which the check/query is to be performed. The following values are valid for OPERATOR:

- ">"
  - Value in metadata must be greater than.
- "<"
  - Value in metadata must be less than.
- ">="
  - Value in metadata must be greater than or equal to.
- "<="
  - Value in metadata must be less than or equal to.
- "=="
  - Value in metadata must be equal to.
- "!="
  - Value in metadata must be **not** equal to.

VALUE is the value (e.g., VALUE = 50) against which the metadata parameter (defined by PARM_NAME) is compared (using the operator specified by the OPERATOR parameter). The value for the VALUE parameter should be the type of data (e.g., integer, string) as defined in the ESDT ODL metadata for the parameter.

DATABASE_QUERY indicates whether the value for the Metadata Check or Query should be retrieved from the PDPS database rather than through the use of the VALUE parameter. Specifying DATABASE_QUERY permits **runtime parameter values** to be used for Metadata Query or Metadata Checks. The following values are valid for the DATABASE_QUERY parameter:

- "NONE"
  - Use the value in the VALUE parameter; no value from the PDPS database is used.
- "PATH NUMBER"
  - Use the Path Number (0-233) of the orbit for which the PGE is scheduled.
- "ORBIT NUMBER"
  - Use the Orbit Number of the orbit for which the PGE is scheduled.
- "TILE ID"

  - Use the Tile ID of the current Data Processing Request.
- "START DATA DAY"
  - Use the Start Data Day for the current Data Processing Request.
- "END DATA DAY"
  - Use the End Data Day for the current Data Processing Request.

KEY_PARAMETER_NAME is an optional parameter that is used to specify the container within a multi-container metadata group (i.e., the MeasuredParameters metadata group in most ESDTs). The KEY_PARAMETER_NAME (e.g., KEY_PARAMETER_NAME = "ParameterName" for metadata checks or qeuries within the MeasuredParameters group) in conjunction with the KEY_PARAMETER_VALUE allows PDPS to determine which container within the multi-container group is to be the object of the check or query. KEY_PARAMETER_NAME is **not** used for product-specific attributes.

KEY_PARAMETER_VALUE is an optional parameter that is used to specify the **value** (e.g., KEY_PARAMETER_VALUE = "LandCoverage") for the container within a multi-container metadata group (i.e. the MeasuredParameters metadata group in most ESDTs). The KEY_PARAMETER_VALUE in both the PGE science metadata ODL file and ESDT science metadata ODL file must match.

Multi-Granule ESDTs are created by adding the following parameter to the PCF_ENTRY in the PGE science metadata ODL file:

- DISTINCT_VALUE.

The DISTINCT_VALUE must be set to the value of the metadata parameter that is used to differentiate granules within the Multi-Granule ESDT. In addtion, the input or output defined by the PCF entry must have a corresponding DISTINCT_PARAMETER entry in the ESDT science metadata ODL file.

11.12.8.22 ESDT Science Metadata ODL File Parameters

The METADATA_DEFINITION ODL object surrounds the definition for Metadata Checks or Metadata Query information within the ESDT science metadata ODL file. An OBJECT/END_OBJECT pair is needed to separate the parameters defining the Metadata Definition from the rest of the parameters defining the ESDT with the following syntax:

```
OBJECT = METADATA_DEFINITION
  .
  .
  .
END_OBJECT = METADATA_DEFINITION
```

A METADATA_DEFINITION object can match multiple Metadata Checks or Metadata Query objects in various PGE science metadata ODL files. There is no difference between the two production rules with respect to the parameters that need to be set in the ESDT science metadata ODL file. Most of the following parameters must be set:

- CLASS.
- PARM_NAME.
- CONTAINER_NAME.
- TYPE.
- KEY_PARAMETER_NAME (optional).
- KEY_PARAMETER_VALUE (optional).

CLASS is a simple counter used to differentiate the different Metadata Definition objects within the file. Each Metadata Definition object within the file must have a **different** CLASS value.

PARM_NAME is the name of the Metadata parameter on which the check or query will be performed. The value specified for PARM_NAME must be part of the granule-level metadata of the ESDT. It must also match the parameter name specified in the PGE science metadata ODL file(s).

CONTAINER_NAME is the name of the Metadata Group within which the metadata parameter defined by PARM_NAME is contained. For product-specific attributes CONTAINER_NAME is set to the string "AdditionalAttibutes" (i.e., CONTAINER_NAME = "AdditionalAttibutes").

TYPE indicates the type of data within the metadata parameter. The following values are valid for TYPE:

- "INT"
  - Integer data.
- "FLOAT"
  - Floating point data.
- "STR"
  - String or character data.
  - Note that dates and times are considered string data.

KEY_PARAMETER_NAME is an optional parameter that is used to specify the container within a multi-container metadata group (i.e., the MeasuredParameters metadata group in most ESDTs). The KEY_PARAMETER_NAME allows PDPS to determine which container within the multi-container group is to be the object of the check or query.

KEY_PARAMETER_VALUE is an optional parameter that is used to specify the value for the container within a multi-container metadata group (i.e., the MeasuredParameters metadata group in most ESDTs). The KEY_PARAMETER_VALUE in both the ESDT science metadata ODL file and PGE science metadata ODL file must match.

The ESDT science metadata ODL file for an input specifying Multi-Granule ESDTs needs to have the following parameter added:

- DISTINCT_PARAMETER.

The DISTINCT_PARAMETER must be set to the name of the metadata parameter that is used to differentiate granules within the Multi-Granule ESDT. A corresponding

METADATA_DEFINITION must be created to help PDPS find the specified metadata parameter when querying the Science Data Server.

11.12.8.23 Data Day Production Rule

The Data Day Production Rule is an addition to the Metadata Query Production Rule involving runtime parameter values. The Data Day Production Rule uses a query to the PDPS database for the time period for the DPR and a Metadata Query for data matching the Data Day. The Data Day is defined as a day within twelve hours of the current day. There is a pair of settings (Start Data Day and End Data Day) that provide parameters for the Metadata Query.

The Start Data Day and End Data Day values are calculated by subtracting twelve hours from the starting day for which the PGE is executing and adding twelve hours onto the ending day for which the PGE is running.

- Data Day for a PGE is running on data from 07/04 00:00:00 to 07/05 00:00:00 is defined as follows:
  - START_DATA_DAY = 07/03 12:00:00
  - END_DATA_DAY = 07/06 12:00:00.

11.12.8.24 Spatial Query Production Rule

The Spatial Query Production Rule allows a PGE to select input(s) based on the spatial coverage of another input (called the **key input**). The PDPS queries the Science Data Server for the spatial coverage of the key input, then uses it in acquiring any subsequent inputs that the PGE has requested that have the same spatial coverage.

- Example:
  - Level 0 input data for an ASTER DPR covers a small section of the Earth.
  - The PGE requires ancillary data that covers the same area to complete its processing.
  - The PGE uses the Spatial Query Production Rule to mark the geographic input as its key input.
  - The PGE specifies that the ancillary input is to be retrieved for the same spatial coverage as that of the key input.
  - When PDPS finds an input granule for the PGE, it performs a Spatial Query to acquire the ancillary input with the same spatial coverage as that of the key input.

Without specifying coordinates, PDPS can match inputs against the spatial constraint of the key input, and give to a PGE only those granules which overlap in area.

For Release 5B Spatial Pad will be added to the Spatial Query Production Rule. Spatial Pad is a means of padding the spatial constraints of the key input. The specified pad is added to all sides of the key input's spatial shape. All granules that intersect the expanded area are retrieved.

Figure 11.12.8.24-1 is an illustration of the Spatial Query Production Rule. The figure shows a PGE that has two input types, one of which is the key input. The other type of input has granules labeled with the names of various colors. One granule (i.e., "green") of the key input is found. The spatial coordinates of the granule are retrieved and all inputs of the second ESDT are checked for overlap with the key input's coordinates.

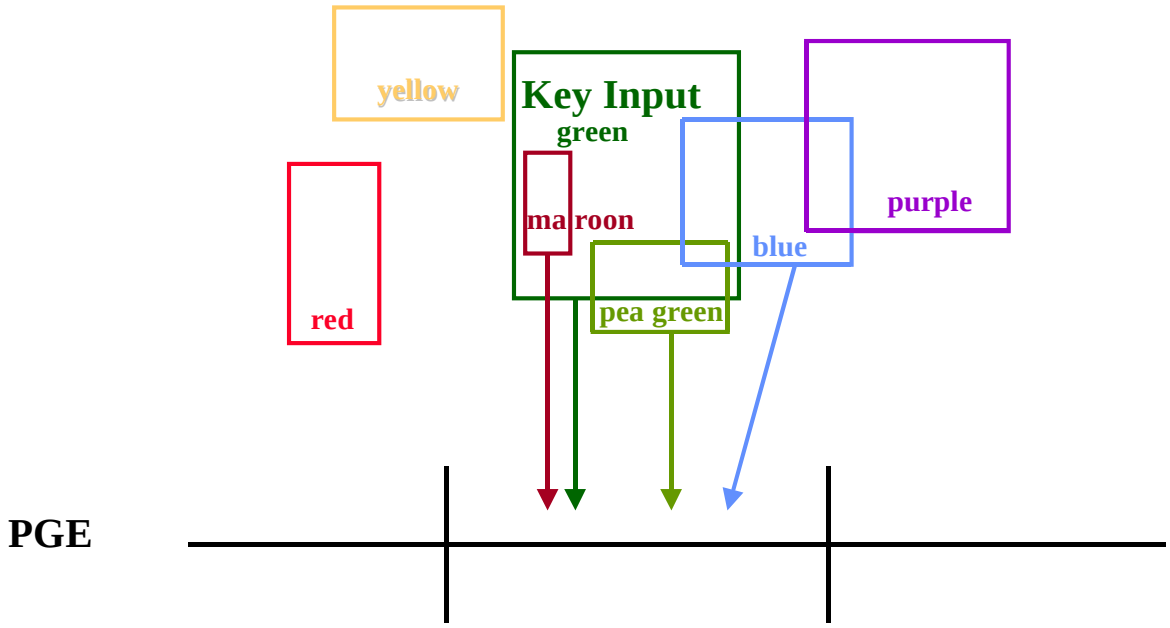


Figure 11.12.8.24-1. Example of the Spatial Query Production Rule

Assuming that all granules relate to the same time period, the granules are evaluated as follows:

- The "yellow" granule is not retrieved as an input because its spatial coordinates do not overlap with those of the key input.
- The "red" granule is not retrieved as an input because its spatial coordinates do not overlap with those of the key input.
- The "blue" granule is retrieved as an input because its spatial coordinates overlap with those of the key input. Part of its spatial constraint is within the constraint of the key input.
- The "maroon" granule is retrieved as an input because its spatial coordinates overlap with those of the key input. The spatial constraint of this granule is completely within the constraint of the key input.
- The "pea green" granule is retrieved as an input because its spatial coordinates overlap with those of the key input. Part of its spatial constraint overlaps with that of the key input.

- The “purple” granule is not retrieved as an input because its spatial coordinates do not overlap with those of the key input. It does not matter that it overlaps with another input that is accepted (i.e., the “blue” granule).

The Spatial Query Production rule is somewhat of an addition to other production rules. As such, it needs the same parameter settings as the Basic Temporal Production Rule. The values specified in the Basic Temporal Production Rule (or other production rules) are set first, then the Spatial Query Production Rule syntax is added.

11.12.8.25 PGE Science Metadata ODL File Parameters

In order to implement the Spatial Query Production Rule the following two parameters must be defined in the applicable PCF_ENTRY (each input is defined by a separate PCF_ENTRY in the PGE science metadata ODL file):

- KEY _INPUT.
- QUERY_TYPE.

The entries are made in the following format:

```
OBJECT = PCF_ENTRY
  .
  .
  .
  QUERY_TYPE = “Spatial”
  KEY_INPUT = “Y”
  .
  .
  .
END_OBJECT = PCF_ENTRY
```

QUERY_TYPE indicates what type of query is to be done to acquire the input defined by the PCF_ENTRY object. Valid values are as follows:

- "Temporal" - Input is acquired based on time.
  - The Basic Temporal and/or the Advanced Temporal Production Rules is/are used to get the input.
  - “Temporal” is the value that is assumed if the parameter is left out of the PCF_ENTRY object.
- "Spatial" - Input is acquired based on spatial coordinates (as well as time).
  - An input must be designated the key input to be used in determining the spatial constraints of the search.
  - “Spatial” is the value specified for each input that uses the Spatial Query Production Rule.
- "Tile" - Input is acquired by the spatial definition of a tile.
  - Refer to the Tiling Production Rule for additional information.

- "Already Created Tile" - Input is acquired based on the tile ID of an already created tile.
  - Refer to the Tiling Production Rule for additional information.

The KEY_INPUT is the input on which the spatial queries for other inputs will be based. When a KEY_INPUT parameter is assigned a value of “Y” the corresponding input is designated a key input and is treated as such. A value of “N” or leaving out the parameter entirely specifies a non-key input. Only one (1) key input is allowed per PGE Profile.

11.12.8.26 Tiling Production Rule

The Tiling Production Rule allows a PGE to run over a series of specific geographic locations called "tiles". The tiles are defined before the PGE is scheduled, specifying the longitude and latitude of four points that outline each tile. When the PGE is scheduled, it is scheduled for an entire day, and data is queried based on both a timeframe and the geographic location specified. Each run of the PGE for that day is for a specific tile, and only data that overlap or fit within the geographical coordinates of the tile are staged for the PGE.

- Example:
  - A MODIS PGE is designed to run on data for a specific geographic location every day.
  - The location is expressed as a polygon defined by latitude and longitude coordinates.
  - The MODIS PGE is scheduled every day, and data are retrieved that match the time period (the day for which the PGE is being executed) and some part of it falls within the geographic constraints of the tile.
  - The PGE runs and produces data that define information about the particular tile.

Period and **boundary** are used to specify the timing of input data and provide indications of how often the PGE should be executed. But at least some of the input data are retrieved on the basis of the coordinates defined for the tile on which the PGE is executing. In fact there are really two kinds of tiling:

- The PGE takes in data based on geographic shapes (tiles) and produces an output or outputs for the specified geographical coverage.
- The PGE takes in an already tiled product as input.
  - This form of tiling is more like a Metadata Query using a runtime parameter value to acquire the correct tiled data.

There are some possible future enhancements to the Tiling Production Rule but they have not been scheduled yet:

- **Zonal Tiling** supports tiles that cover a band around the Earth between two given latitudes.

- **Tile Clustering** involves grouping tiles that cover nearby geographic locations together so that data that span the tiles may be staged only once.
  - Intended to improve the performance of Tiling.
  - Also provides for the ability to prioritize one group of tiles over others (so specific geographic outputs are produced before other geographic outputs).

Runtime parameters can be set to the ID of the tile being processed. Since PDPS schedules a Tiling PGE to run once per tile, it can pass the identifier of the tile to the PGE. The identifier can be placed under a specified runtime parameter in the PCF, or it can be used in a Metadata Query for a PGE that would use already tiled data as input.

Figure 13 provides an example of the Tiling Production Rule. The PGE runs once per defined tile. So for every tile in the Tile Scheme a Data Processing Request is created to run using data that match the geographic extent of the tile. The PDPS sends the coordinates of the tiles (e.g., Tiles 1 through 3 in Figure 11.12.8.27-1) to the Science Data Server when requesting data and acquires only the granules that fall fully or partially within the defined tile.

The PGE itself must be set up to handle the fact that the entire area of the tile may not be covered by available data. In addition, because PDPS does not keep track of tiles once they have been produced, the PGE must set the metadata of the output products so a downstream Tiling PGE can acquire the correct granules for a given tile. The PDPS matches up the granules needed for a downstream PGE via a query to the Data Server Subsystem.

11.12.8.27 Tiling Based on Already Tiled Data

As previously stated, the second form of Tiling concerns PGEs based on tiles that have already been created by other PGEs. Tiling based on already tiled data is really a combination of the Metadata Query Production Rule and the Tiling Production Rule. The latter is used in running the PGE(s) once per tile, just like any other Tiling PGE. The Metadata Query Production Rule is used in acquiring the previously tiled data by querying the Science Data Server for metadata that match the tile ID that is currently being executed. The query depends on the “runtime parameters” function of Tiling to provide the tile ID relevant to the PGE that is currently being executed.

The Tiling Production Rule is based (at least for the PGE science metadata ODL file) on the same fields used for the Basic Temporal Production Rule. A PGE that performs Tiling still needs a **boundary** and **period** and other such parameters. The difference is that values specified for some of the fields provide Tiling information. Furthermore, Tiling requires that a tile scheme be identified in the PGE science metadata ODL file. The tile scheme is defined in a tile science metadata ODL file.

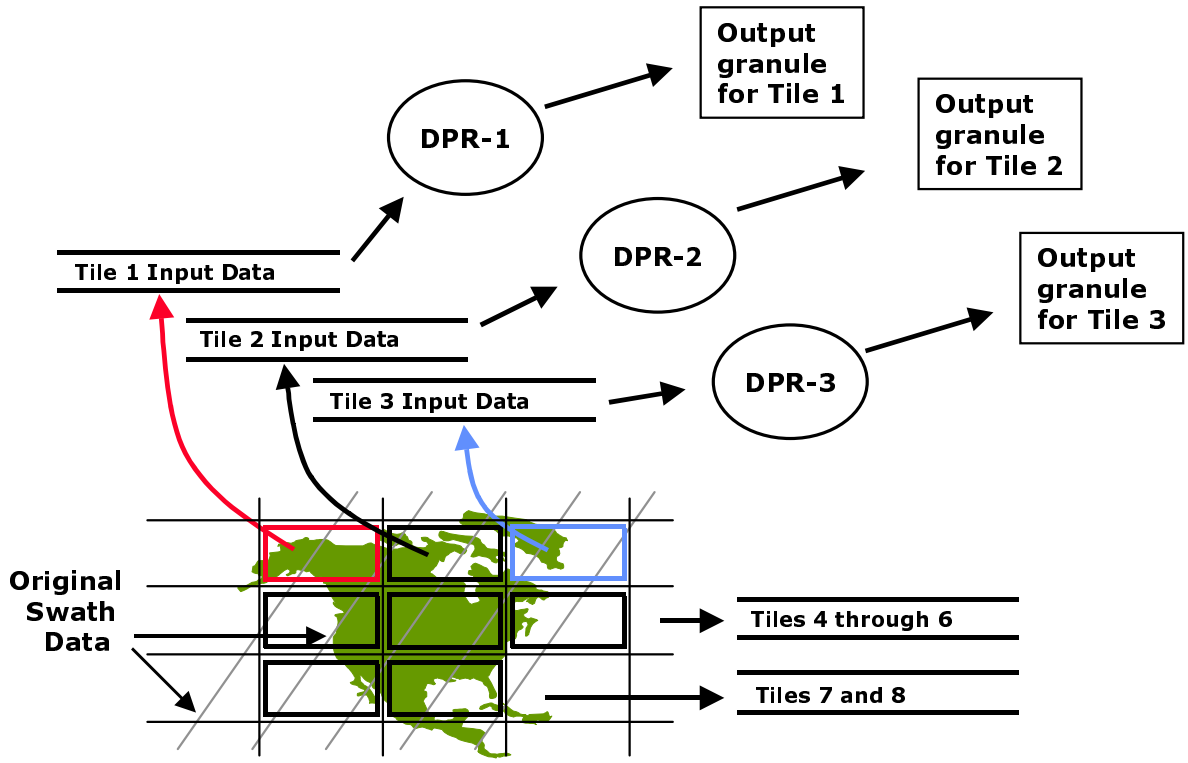


Figure 11.12.8.27-1. Example of the Tiling Production Rule

PGE Science Metadata ODL File Parameters

The following parameters must be set in the PGE science metadata ODL file in order to implement the Tiling Production Rule:

- SCHEDULE_TYPE.
- TILE_SCHEME_NAME.

In addition, the following parameter is used within a PCF_ENTRY when defining the Tiling Production Rule:

- QUERY_TYPE.

The SCHEDULE_TYPE parameter defines the type of scheduling that will be done for the PGE. Values for the Tiling Production Rule are:

- "Tiling"
  - Tile-Scheduled.
  - The PGE is scheduled based on the specified PROCESSING_PERIOD and PROCESSING_BOUNDARY, but a DPR is created for each defined tile.

The TILE_SCHEME_NAME parameter is the name of the Tile Scheme to be used by PDPS when scheduling and executing PGEs for each defined tile. There must be a tile ODL file that matches the specified scheme name.

The QUERY_TYPE parameter specifies the type of query to be performed on the input defined by the PCF_ENTRY Object. It uses the following syntax:

```
OBJECT = PCF_ENTRY
  .
  .
  .
  QUERY_TYPE =
  .
END_OBJECT = PCF_ENTRY
```

For Tiling PGEs there are two possible values for QUERY_TYPE:

- "Tile"
  - The data for the input are acquired on the basis of the spatial constraints of the current tile.
  - Used for a PGE that takes in raw data and produces one or more tiles of data.
- "Already Created Tile"
  - The input is a tiled output of another Tiling PGE.
  - Used for a PGE that takes input from one or more other Tiling PGEs.
  - A Metadata Query must be added to this PCF_ENTRY in order for the correct tiled input to be acquired.

11.12.8.28 Tile Science Metadata ODL File Parameters

The following parameter must be set in the Tile science metadata ODL file in order to implement the Tiling Production Rule:

- TILE_SCHEME_NAME.

In addition, the following ODL objects are used within a PCF_ENTRY to define the Tiling Production Rule:

- TILE object.
- TILE_COORDINATE object.

The TILE_SCHEME_NAME parameter identifies the tile scheme for which the tile information is being specified. Values are limited by the following constraints:

- The string specified can be no more than 20 characters.
- The string specified should match the string specified for TILE_SCHEME in the PGE science metadata ODL file.

The TILE object is an ODL object that surrounds each tile definition. An OBJECT/END_OBJECT pair (as shown in the example that follows) is needed for each tile that is going to be expressly defined:

```
OBJECT = TILE
  .
  .
  .
END_OBJECT = TILE
```

The following parameters are set in the TILE object in order to implement the Tiling Production Rule:

- CLASS.
- TILE_ID.
- TILE_DESCRIPTION.

CLASS is a simple counter used to differentiate the different TILE objects within the file. Each TILE object needs to have a different CLASS value.

TILE_ID is the tile identifier for the tile being defined. The TILE_ID must be an integer (e.g., TILE_ID = 12) and must be greater than zero but less than the maximum integer. If a Tile ID is defined in other tile schemes, it must have the same coordinates and description.

TILE_DESCRIPTION is a string of characters (255 characters maximum) that describes what the tile is for, such as its geographic location or area that it covers (e.g., TILE_DESCRIPTION = "Upper North America").

The TILE_COORDINATE object is an ODL object that defines a coordinate (latitude and longitude) for a tile. An OBJECT/END_OBJECT pair is needed for each coordinate that is defined. Each tile must have four TILE_COORDINATE objects defined. (Currently only four-sized polygons are allowed; however, a possible future enhancement would provide for polygons with more than four points.) Coordinate objects must follow a clockwise sequence so that if lines were drawn between the points in the order they are given the desired shape would be drawn.

Coordinate objects conform to the following format:

```
OBJECT = TILE
  .
  .
  .
  OBJECT = TILE_COORDINATE
    CLASS =
    LATITUDE =
    LONGITUDE =
  END_OBJECT = TILE_COORDINATE
  .
```

```
.
.
END_OBJECT = TILE
```

The following parameters are set in the TILE_COORDINATE object in order to implement the Tiling Production Rule:

- CLASS.
- LATITUDE.
- LONGITUDE.

The CLASS parameter (e.g., CLASS = 1) is an object counter that is used only to distinguish objects. The value assigned to CLASS must be an integer greater than zero and must be unique in the file for the particular type of object.

The LATITUDE parameter (e.g., LATITUDE = 12.15) describes the latitude component of the tile coordinate. There is one LATITUDE entry per TILE_COORDINATE object.

The LONGITUDE parameter (e.g., LONGITUDE = -43.22) describes the longitude component of the tile coordinate. There is one LONGITUDE entry per TILE_COORDINATE object.

11.12.8.29 Closest Granule Production Rule

The Closest Granule Production Rule allows a PGE to request the nearest input granule from the Data Processing Request time. The PDPS requests a search forward or backward for a specified period of time until it finds a granule that matches the request. However, there is a limit to the number of queries that are performed. The number of queries and the period length of the query are specified during SSI&T.

- Example:
  - A PGE processes data at daily intervals and could use a particular type of calibration granule that would allow it to determine the nearest parameters of the instrument.
  - Although most calibration coefficients are defined as static granules, in this case there is a dynamic granule that is received about once a month.
  - The closest such granule would be optimum, so the PGE uses the Closest Granule Production Rule to search forward or backward from the time of the DPR to find the nearest calibration granule.

The Closest Granule Production Rule supersedes the Most Recent Granule Production Rule. The latter allowed the search for inputs to go backward in time from the start of the DPR. The Closest Granule Production Rule allows the search for input granules to go either backward or forward in time, increasing the flexibility of the rule. The Closest Granule Production Rule has all of the ability of Most Recent Granule, plus the ability to search forward in time for input data.

The Closest Granule Production Rule uses two values to determine the period of the query. The two values are concerned with the direction of the query and the number of queries allowed.

- Offset.

  - Tells the PDPS software the query duration.
  - The sign (+ or -) indicates whether the query goes forward (positive) or backward (negative) in time.
- Retries.
  - Tells the PDPS software how many time periods (as defined by the offset) to search (either forward or backward) in time for matching granule.

The PDPS does a Basic Temporal query before using Closest Granule to find the input. If the desired input is not found within the time period of the DPR, PDPS performs a query against the Science Data Server for the period defined by the offset. Again, if no matching granule is found, PDPS repeats the query, going backward or forward in time by the value specified in the offset. If no acceptable granule has been found before the maximum number of queries is reached, PDPS fails to generate the DPR due to insufficient input data.

Figure 11.12.8.29-1 illustrates the Closest Granule Production Rule. In the example, the PGE has a boundary of "start of day" and a period of one hour, so it is scheduled to run for one hour's worth of input data. The input has a period of one hour, and can come in at any hour of the day. Consequently, the PGE requests one granule of input.

The PGE has defined the Closest Granule Production rule with a –6-hour period of the query, meaning that it queries back in time in six-hour intervals. The number of retries is two. The PDPS performs a query for the input based on the time period of the DPR. Not finding any matching data, it uses the Closest Granule information to query for a six-hour period beginning six hours before the start time of the DPR. Again nothing is found, so a second Closest Granule query is performed, this one six hours before the last Closest Granule query. The second query results in the discovery of two matching granules. The PDPS selects the granule that is later in time and schedules the PGE to use it as input.

If the Closest Granule Production Rule were used in conjunction with the Minimum/Maximum Number of Granules Production Rule, it might be possible for both granules to be selected in the previously described Closest Granule query. If the example included setting the Maximum Number of Granules to two, both granules would be selected as input to the PGE.

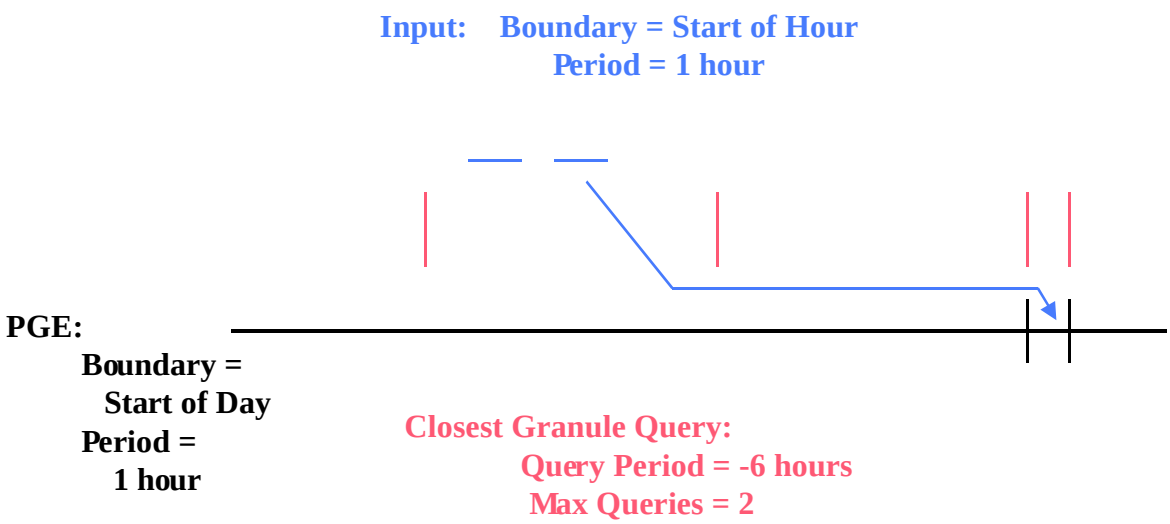


Figure 11.12.8.29-1. Example of Closest Granule Production Rule

The Closest Granule Production Rule needs the same parameter settings as the Basic Temporal Production Rule. The values needed for the Basic Temporal Production Rule must be set before the Closest Granule Production Rule syntax is added.

11.12.8.30 PGE Science Metadata ODL File Parameters

In addition to the parameter settings for the Basic Temporal Production Rule, the following parameters must be set within the appropriate PCF_ENTRY in the PGE science metadata ODL file in order to implement the Closest Granule Production Rule:

- CLOSEST_QUERY_OFFSET.
- CLOSEST_QUERY_RETRIES.

CLOSEST_QUERY_OFFSET is the offset added to or subtracted from the Data Start Time of the DPR and uses as the query for the requested input data type. The specified value has the format "<Period Type>)=<Length of Period>" (e.g., CLOSEST_QUERY_OFFSET = “HOURS=6”). The following “Period Type” values are used in implementing the Closest Granule rule:

- "WEEKS"
  - Offset is some number of weeks.
  - For example, "WEEKS=2" would be a 14-day offset.
- "DAYS"
  - Offset is some number of days.
  - For example, "DAYS=5" would be a 120-hour offset.

- "HOURS"
  - Offset is some number of hours.
  - For example, "HOURS=4" would be a 240-minute offset.
- "MINS"
  - Offset is some number of minutes.
  - For example, "MINS=5" would be a 300-second offset.
- "SECS"
  - Offset is some number of seconds.

CLOSEST_QUERY_RETRIES is the maximum number of Closest Granule queries before the DPR fails due to insufficient input data. The specified value is an integer value that is limited only by the maximum size of an integer on the executing hardware.

Note that the longer the offset value or the greater the number of retries, the more time that each query requires due to search time at the Science Data Server and processing time of any granules returned. The combination of a large offset with a large number of retries, can (if no data granules are found) consume a lot of time while failing to generate a DPR.

11.12.8.31 Orbital Processing Production Rule

The Orbital Processing Production Rule is similar to the Basic Temporal Production Rule in that both define the time period for the inputs and outputs of the PGE. The difference is that the Orbital Processing Production Rule uses the orbit of the spacecraft to determine that time period. A PGE that processes data related to every orbit of a satellite uses data related to a time period that is computed from the orbit of that satellite.

- Example:
  - A PGE processes Level 0 data related to each orbit of the AM-1 satellite.
  - The AM-1 satellite has an orbital period of 98 minutes so the PGE is scheduled to process data for each 98-minute interval.
  - Since Level 0 data are received every two hours, the data staged for the PGE include every Level 0 granule that falls within the 98 minute PGE interval.
  - Only one granule of Level 0 data is relevant to some 98-minute orbits.
  - Two granules of Level 0 data are relevant to other 98-minute orbits.

The Orbital Processing Production Rule uses the “period” and “boundary” concept just like the Basic Temporal Production Rule. The difference is that for Orbital Processing, the orbit of the spacecraft is taken into account when a PGE or its data are marked as **orbit scheduled**.

When responding to a Production Request for orbit-scheduled processing, PDPS determines the orbit of the satellite via information provided during the SSI&T process. The information (stored in the PDPS database) gives the start time and length of a particular orbit or set of orbits. PDPS extrapolates (or interpolates in the case of an orbit between two orbital periods stored in the

database) the start and end times of the PGE that is specified in the Production Request. Data are sought on the basis of the derived start and stop times and the appropriate data granule(s) is/are staged before the PGE is executed.

Orbital path is the path of the satellite over the Earth. It is a number from 0-233 that indicates the region of the Earth covered by a particular orbit. Note that because of the implementation of Orbital Path, there needs to be a mapping between the orbital path calculated by PDPS and the orbital path number expected by the PGEs.

Runtime parameters can be set to values associated with Orbital Processing. The following list of orbital parameters can be placed under runtime parameters:

- Orbit Number.
  - The number of the orbit (starting from zero) and continually increasing.
- Orbital Path Number.
  - The number of the path that maps to the orbit number.
  - The orbital path number is the 0-233 orbital path traversed by the satellite.
- Orbit Number within the Day.
  - The number of the orbit within the given day.
  - It includes any orbit that starts within the given day.
  - This is a **Release 5B** capability.
- Granule Number within the Orbit.
  - The number of the granule within a given orbit.
  - It includes any granule that starts within the given orbit.
  - This is a **Release 5B** capability.

Figure 11.12.8.31-1 provides an illustration of the Orbital Processing Production Rule. The PGE in the diagram takes a two-hour input, but is scheduled based on the orbit time and period of the satellite. PDPS uses the data collected at SSI&T to predict the time of the orbit and performs the query to the Science Data Server for the input based on that extrapolated or interpolated orbital time. Granules of input data are allocated to DPRs based on their ability to cover the time period relevant to the DPR.

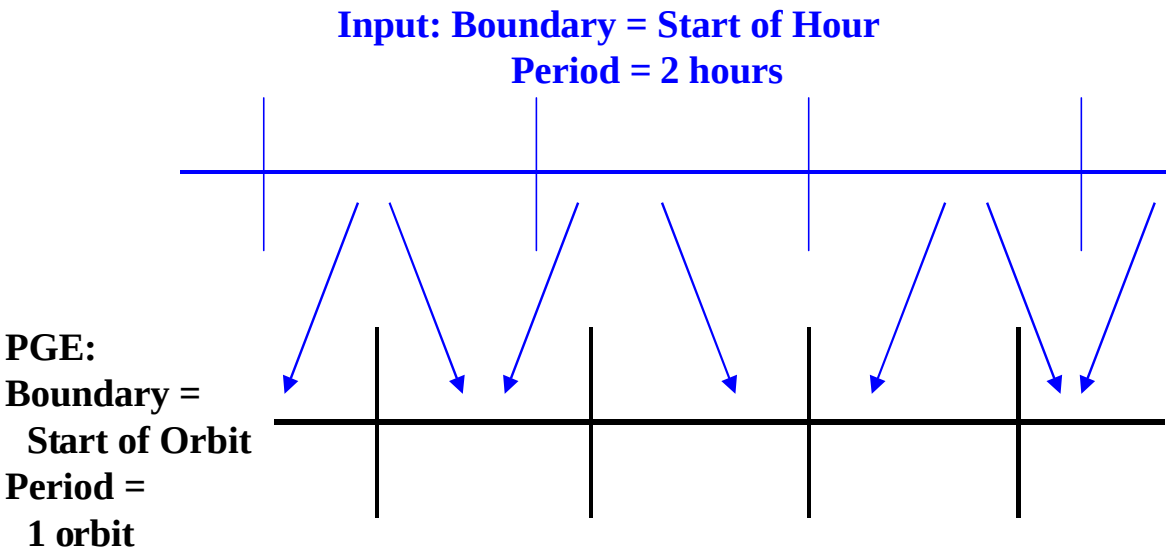


Figure 11.12.8.31-1. Example of the Orbital Processing Production Rule

In the example shown in Figure 11.12.8.31-1 the length of an orbit is less than the period of the two-hour input, so sometimes a single granule may cover the input time range of a PGE execution and at other times two granules are required. The production rule would work equally well if the data were of a shorter period (e.g., 1/2 hour) than the orbit of a satellite (e.g., 90 minutes). In such a case three granules would be staged for every execution of the PGE.

The Orbital Processing Production Rule is based (at least for the PGE science metadata ODL file) on the same fields used for the Basic Temporal Production Rule. However, the values specified for the parameters provide orbit information rather than time-period information.

11.12.8.32 PGE Science Metadata ODL File Parameters

The following parameters must be set in the PGE science metadata ODL file in order to implement the Orbital Processing Production Rule:

- PLATFORM.
- SCHEDULE_TYPE.
- PROCESSING_PERIOD.
- PROCESSING_BOUNDARY.

The PLATFORM parameter is the name of the platform (satellite) for which the PGE is processing data. Information concerning the orbits of a satellite are stored in the PDPS database. Values that can be assigned to the parameter are subject to the following constraints:

- The string specified can have no more than 25 characters.

- The string specified should match the string specified in the orbit science metadata ODL file. If no matching file is found, an error is reported during SSI&T.

The SCHEDULE_TYPE parameter describes the type of scheduling that is required for the PGE. "Orbit" is the value used for Orbital Processing. As a result, the PGE is scheduled based on the start time and period of the satellite's orbit. Note that PROCESSING_PERIOD and PROCESSING_BOUNDARY must be set correspondingly.

The PROCESSING_PERIOD is the time interval for the data that the PGE processes. Assuming no combination of production rules that would affect the period, data are acquired for the specified PROCESSING_PERIOD and output data are planned for the given period. The value assigned to PROCESSING_PERIOD is of the format "<Period Type>=<Length of Period>". The “Period Type” applicable to the Orbital Processing Production Rule is "ORBITS". For example, "ORBITS=1" would be applied to a PGE that processes data related to one orbit’s worth of data.

The PROCESSING_BOUNDARY is the boundary (starting point in time) of the PGE. It specifies when each instance of the PGE should start. Note that the PROCESSING_BOUNDARY and PROCESSING_PERIOD are used in conjunction when scheduling the PGE. Consequently, "START_OF_ORBIT" is the acceptable PROCESSING_BOUNDARY for the Orbital Processing Production Rule. It indicates that the PGE processes data related to each satellite orbit. It must be used in conjunction with a PROCESSING_PERIOD that specifies a “Period Type” of “ORBITS”.

11.12.8.33 Orbit Science Metadata ODL File Parameters

The following parameter must be set in the orbit science metadata ODL file in order to implement the Orbital Processing Production Rule:

- PLATFORM.

In addition, the following ODL object is used in defining orbits for the Orbital Processing Production Rule:

- ORBIT_MODEL object.

The value assigned to the PLATFORM parameter in the orbit science metadata ODL file must be exactly the same as that specified for the same parameter in the PGE science metadata ODL file.

The ORBIT_MODEL object is an ODL object that surrounds each orbit definition. An OBJECT/END_OBJECT pair (as shown in the example that follows) is needed for each orbit that is to be expressly defined: PDPS extrapolates or interpolates orbits that are not specifically defined within the file.

```
OBJECT = ORBIT_MODEL
 CLASS = 1
 ORBIT_NUMBER = 1000
 ORBIT_PATH_NUMBER = 68
 ORBIT_PERIOD = "MINS=98"
```

```
    ORBIT_ START = “09/21/1999 14:50:00”
  END_OBJECT = ORBIT_MODEL
```

The following parameters are set in the ORBIT_MODEL object in order to implement the Orbital Processing Production Rule:

- CLASS.
- ORBIT_NUMBER.
- ORBIT_PATH_NUMBER.
- ORBIT_PERIOD.
- ORBIT_START.

CLASS is a simple counter used to differentiate the different ORBIT_MODEL objects within the file. Each ORBIT_MODEL object needs to have a different CLASS value.

ORBIT_NUMBER is simply the number of the orbit being specified. Each orbit of the satellite has a sequential number associated with it. This is the integer value of the orbit number for the orbit being defined in the ORBIT_MODEL object.

ORBIT_PATH_NUMBER is value of the path for the specified orbit. The orbital path is a number from 0-233 that repeats every 16 days. This is the integer value of the orbital path number for the orbit being defined in the ORBIT_MODEL object.

ORBIT_PERIOD is the length of time it takes for the satellite to complete one orbit. The value assigned to ORBIT_PERIOD has the format "<Period Type>=<Length of Period>" (e.g., “MINS=98”). Note that the “Length of Period” is specified as a positive integer only.

Period Type values for the orbit model science metadata OFL file are:

- "WEEKS"
  - Orbit spans some number of weeks.
  - For example, "WEEKS=2" would be an orbit that takes two weeks to complete.
- "DAYS"
  - Orbit spans some number of days.
  - For example, "DAYS=5" would be an orbit that takes five days to complete.
- "HOURS"
  - Orbit spans some number of hours.
  - For example, "HOURS=4" would be an orbit that takes four hours to complete.
- "MINS"
  - Orbit spans some number of minutes.
  - For example, "MINS=85" would be an orbit that takes eighty-five minutes to complete.
- "SECS"
  - Orbit spans some number of seconds.

- For example, "SECS=7200" would be an orbit that takes 7200 seconds (two hours) to complete.

ORBIT_START is the start date and time for the orbit defined by the particular ORBIT_MODEL object. Its format is either "MMM DD YYYY HH:MM:SS" or “MM/DD/YYYY HH:MM:SS”.

11.12.8.34 Production Planning Considerations

During normal operations it is expected that the Production Planner will not have to add PRs to the PDPS database very frequently. The frequency of this activity is, to some extent, determined by the SCF responsible for the science software.

- The PR is a template request to generate a particular data product and results in a production run of the associated SCF-provided PGE.
  - PR specifies a range (temporal, orbit, or tile) over which the data products are to be produced or the PGEs are to be scheduled.
    - PR might request that the data product be produced for only a single day’s data.
    - PR might request that data products be produced for every opportunity of input data for several months, resulting in several hundred jobs being planned and run as the input data become available.
  - Early in a mission the SCF may prefer to request processing for a short time period only (e.g., a week or less).
    - At that time the SCF is gaining an understanding of the on-orbit behavior of the instrument, the resulting data, and the interaction of the science processing software with real data.
    - SCF reviews the quality of the products and notifies the Production Planner of the need for any changes to the PR (e.g., discontinue the PR, change time ranges, or modify input parameters).
  - When the SCF has developed a good understanding of the instrument’s behavior, the team may be comfortable requesting processing for months at a time.
  - DAAC operations may have operational reasons for wanting to issue processing requests for a more limited time period.
- The Production Planner has to balance the various considerations when determining whether or not to create or update a PR.

Planning decisions are made on the basis of locally defined planning strategies for supporting the SCFs’ data processing needs. The production planning tools are intended to be flexible enough in their design to support the particular planning and scheduling cycles of the operations organization at each DAAC.

Before planning production the Production Planner must coordinate with the Resource Planner to resolve all resource allocation issues. The Resource Planner notifies the Production Planner of the resources available for use in processing. Furthermore, the Production Planner may well have direct access to the Resource Plan.

The Production Planner prepares monthly and weekly production plans. In addition, the Production Planner develops a daily production schedule from the most current weekly plan. However, the first step in the planning process is creating production requests using the Production Request Editor.

11.12.8.35 DPREP Considerations

DPREP (data preprocessing) is a set of three PGEs that are supplied by ECS, unlike most PGEs, which are provided by the Science Computing Facilities that ECS supports. DPREP consists of the following three PGEs:

- EcDpPrAm1EdosEphAttDPREP_PGE (Step 1).
- EcDpPrAm1FddAttitudeDPREP_PGE (Step 2).
- EcDpPrAm1FddEphemerisDPREP_PGE (Step 3).

The PGEs run separately and in a particular sequence.

Two files describe the PGEs and how to run them:

- "DPREP_README"
- "HowtoRunDPREP"

The files are installed on the science processor hosts (e.g., e0spg01, g0spg01, l0spg01, n0spg03) in the /usr/ecs/*MODE*/CUSTOM/data/DPS directory.

The DPREP PGEs process Level Zero (L0) Terra (AM-1) spacecraft data (e.g., ESDT AM1ANC) provided by EDOS. The output files/granules of the DPREP PGEs are subsequently used in the processing of data from various instruments on the satellite. They provide the following types of ancillary (non-science) data:

- Ephemeris
  - Spacecraft location: ephemeris (or orbit) data include: latitude, longitude, and height.
- Attitude
  - Orientation of the satellite, including yaw, pitch, and roll angles; and angular rates about three axes.

There are two profiles for DPREP PGEs:

- Profile 1 runs routinely at the DAACs using previous DPREP output in addition to new Terra ancillary (e.g., AM1ANC) data.
- Profile 2 (the boot-up procedure) takes in the Terra ancillary data only and is run under two sets of conditions:
  - First run of DPREP (because there is no previous output) to initialize DPREP processing.

– Following any long period of time during which EDOS L0 ancillary data are unavailable. (Short gaps in the ephemeris data are filled by EcDpPrAm1EdosEphemerisRepair, one of the executables in the EcDpPrAm1EdosEphAttDPREP_PGE.)

In order to run Profile 2 successfully following a long period of data unavailability, DPREP must be told where to resume orbit counting. The initial orbit number in the Step 1 process control file (PCF), must be set to the orbit number corresponding to the timestamp at which data availability resumes.

Until an automated process can be implemented, whenever there is a telemetry drop-out, a member of the DAAC science support team takes the following actions:

- Calls the Flight Operations Team (FOT).
- Asks for the on-line engineer.
- Requests the orbit number that coincides with the start time of the first L0 ancillary data set that follows the data drop-out.
- Sets the orbit number in the Step 1 PCF.

Then Profile 2 can be run successfully. Afterward, routine operations can be resumed using Profile 1 PGEs.

11.13 PGE Registration and Test Data Preparation

The integration of science software with ECS requires that information about the Product Generation Executives (PGEs) be made known to the PDPS in its database. In addition, the PGEs themselves and the test files that they use (both input and output) need to be placed on the Data Server. These steps must be accomplished before the science software can be run and tested within the ECS.

The following procedures describe how to register a new PGE with ECS. This involves updating the PDPS database with information needed to plan, schedule, and run the PGE. The first step in the PGE registration process is to determine which ESDTs are needed for the PGE. You must Verify that an ESDT metadata ODL file exists for each ESDT or generate an ODL file. The next step in the process is to create a PGE metadata ODL file using the delivered PCF. Finally, additional operational information (resource requirements and runtime statistics) must be input into the PDPS database. This is the last step in the PGE registration process. The order in which these procedures are done is important and should be done as indicated. Please reference Appendix C. for Examples of PGE and ESDT ODL Files for Each Instrument Team**.**

11.13.1 PGE ODL Preparation

. **This section describes how to prepare PGE ODL files. It is assumed that the SSIT Manager is running** .

1 From the SSIT Manager, click on the **Tools** menu, then choose **PDPS Database** and then **PCF ODL Template**.

- An xterm with title “SSIT: Science Metadata ODL Template Creation” will be displayed.

2 At the program prompt **Configuration Filename (enter for default: ../../cfg/EcDpAtCreatODLTemplate.CFG)?**

- Press **Enter** for the default configuration file

3 At the program prompt **ECS mode of operations?,** type ***mode***, press **Return** or just press **Return** if the default shown is correct.

- The ***mode*** refers to the database used and will typically be **OPS** or **TS1**.

4 At the program prompt **Process Control file name (PCF to generate template from)?**, type ***PCFpathname***/***PCFfilename***, press **Return**.

- The ***PCFpathname*** is the full path name to the location of the PCF. If not specified, the directory from which the SSIT Manager was run will be assumed.
- The ***PCFfilename*** is the file name of the PCF.

5 At the program prompt **PGE name (max 10 characters)?**, type ***PGEname***, press **Return**.

- The ***PGEname*** is the name of the PGE that will be registered.

6 At the program prompt **PGE version (max 5 characters)?**, type ***PGEversion***, press **Return** or just press **Return** if the default shown is correct.

- The ***PGEversion*** is the version of the PGE that will be registered.

7 At the prompt **PGE Profile ID (0 for Null, max 99)?,** type 1 or any valid profile ID.

- After a brief time, the message “Successfully created ODL template file” should be displayed if the task was successful.
- The program will output a file with the filename **PGE_*PGEname#PGEversion#ProfileID*.tpl**.
- For example, if the PGE name was **PGE35**, and the version and profile ID were both **1** this output file will be named **PGE_PGE35#001#01.tpl**.

8 At the program prompt **Hit return to run again, ‘q <return>’ to quit:**, press **Return** to repeat process with another PCF or type **q** and press **Return** to quit.

- The xterm will disappear.

9 At a UNIX prompt on an AIT Sun, type **cd *SSITrunPathname***, press **Return.**

- The ***SSITrunPathname*** is the full path to the directory from which the SSIT Manager was run, for example /usr/ecs/TS1/CUSTOM/bin/DPS. This will be the directory where the file **PGE_*PGEname#PGEversion#ProfileID*.tpl** will reside.

10 At a UNIX prompt on the AIT Sun, type **cp PGE_*PGEname#PGEversion#ProfileID*.tpl PGE_*PGEname#PGEversion#ProfileID*.odl**, press **Return.**

- The **PGE_*PGEname#PGEversion#ProfileID*.tpl** is the file name of the ODL template file created in step 7.
- The **PGE_*PGEname#PGEversion#ProfileID*.odl** is the file name of a copy which can be safely edited. This file name convention must be used.

11 At a UNIX prompt on the AIT Sun, type
mv PGE_*PGEname#PGEversion#ProfileID*.odl /usr/ecs/<mode>/CUSTOM/data/DPS/ODL

- This will place the ODL file in the directory where the executable that populates the PDPS database will read from. **PGE_*PGEname#PGEversion#ProfileID*.odl** is the file name of the copy created in step 10.

12 At a UNIX prompt on the AIT Sun, change the directory to the one in step above and type **vi PGE_*PGEname#PGEversion#ProfileID*.odl**, press **Return.**

- The **PGE_*PGEname#PGEversion#ProfileID*.odl** is the file name of the copy created in step 10.
- Any text editor may be used such as *emacs*. For ex**ample, emacs PGE_PGE35#001#01.odl, press Return.**

13 In the file, add required metadata to the ODL template.

- For an explanation of what metadata is required, see file /usr/ecs/<mode>/CUSTOM/data/DPS/PGE_ODL.template.
- Note that the ShortNames typed into this file must each have a corresponding PDPS ESDT metadata ODL file (sec. 11.13.2).
- All objects corresponding to output ESDTs will automatically have the SCIENCE_GROUP and YIELD set during the generation of PGE ODL.
- All objects corresponding to output ESDTs will have an attribute "ASSOCIATED_MCF_ID. Place here the Logical Unit Number (LUN) listed in the PCF for the associated MCF listing.
- All objects corresponding to static input ESDTs must have the SCIENCE_GROUP set. Objects corresponding to *dynamic* input ESDTs should NOT have the SCIENCE_GROUP set.
- See Appendix E for an example of PCF and corresponding PGE ODL files.

14 Save the changes made to the ODL template file and exit the editor.

- The specifics depend upon which editor is being used. If using *vi*, the command sequence to enter is **:wq**, press **Return**.
- For other editors, refer to that editor's documentation.
- If you make a mistake entering any values, press **Return** here; your previous enteries are restored as defaults and you won't have to retype them.
- A comment should be received: "**Update of PDPS/SSIT database with PDPS SCIENCE METADATA SUCCESSFUL"**

11.13.2 ESDT ODL Preparation

Assumption:

The PGE ODL file has been created and edited for the required PGE.

Follow the steps below to prepare ESDT ODL files for each ESDT required by the PGE.

---

1 Determine ShortName for required ESDTs corresponding to a Logical Unit Number (LUN) in the PGE ODL file.

2 At a UNIX prompt on an AIT Sun, type **ls /usr/ecs/<mode>/CUSTOM/data/DPS/ODL/ESDT_*ShortName#Version*.odl**, press **Return**.

- The **ESDT_*ShortName#Version*.odl** is the file name of the ESDT ODL file you are looking for where ***ShortName*** is the ESDT's ShortName and ***Version*** is the ESDT version. If a file for the desired ESDT is listed, then it has already been prepared and this procedure can be exited now.
- For example, if the desired ESDT has the ShortName MOD03 and version 001, type **ls /usr/ecs/TS1/S/CUSTOM/data/DPS/ODL/ESDT_MOD03#001.odl,** press **Return**.
- If the desired file is *not* listed, continue on to step 3.

3 At a UNIX prompt on the AIT Sun, type **cd *WorkingPathname***, press **Return.**

- The ***WorkingPathname*** is the full path name to a working directory for which the user has write permissions.
- For example, **cd /home/jdoe/working/**, press **Return**.

4 At a UNIX prompt on the AIT Sun, type **cp /usr/ecs/<mode>/CUSTOM/data/DPS/ESDT_ODL.template ESDT_*ShortName#Version*.odl**, press **Return**.

- For <mode> enter the mode you are working in, for example **OPS** or **TS1**.
- The **ESDT_*ShortName#Version*.odl** is the file name of the ESDT ODL file to be created.
- This command copies a template ESDT ODL file to the ESDT ODL file to be created. The template is well commented.
- For example, type **cp /usr/ecs/<mode>/CUSTOM/data/DPS/ESDT_ODL.template ESDT_MOD03#001.odl,** press **Return**.
- The **ESDT_*ShortName#Version*.odl** file naming convention *must* be observed.

5 At a UNIX prompt on the AIT Sun, type **vi ESDT_*ShortName#Version*.odl**, press **Return**.

- The **ESDT_*ShortName#Version*.odl** represents the file name of the ESDT ODL template file created in step 4.
- Any text editor may be used such as *emacs*. For example, **emacs ESDT_MOD03#001.odl**, press **Return**.

6 In the file, add required metadata to the ODL template.

- Use the internal documentation contained in the ODL file (from the original template) to aid in populating with metadata.
- Note that the ShortName specified within the file must match the ShortName of the file name itself.
- In addition, the ShortNames used in the PDPS PGE metadata ODL file must match the ShortNames in these files.

7 Save the changes made to the ESDT metadata ODL file and exit the editor.

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

8 Next type **mv ESDT_*ShortName#Version.odl* /usr/ecs/<mode>/CUSTOM/data/DPS/ODL.**

- This will place the just created ESDT ODL file in the directory where PDPS will read it from.

9 Repeat steps 1 through 8 for each ESDT required by a particular PGE. When all ESDT metadata ODL files have been completed, continue on to next section.

11.13.3 Update PDPS/SSIT Database with PGE Science Metadata

In order to update the PDPS Database with PGE metadata, the ESDT metadata ODL files must first be prepared for each ESDT required by the PGE. This section describes how to perform the next step, running the SSIT Science Update program.

Assumptions:

1. The SSIT Manager is running.
2. The directory used for containing the PDPS PGE metadata ODL files. Nominally, this is **/usr/ecs/<mode>/CUSTOM/data/DPS/ODL.**

Updating the PDPS Database with PGE Metadata

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

The directory used for containing the PDPS ESDT metadata ODL files can be accessed by the following commands:

---

1 telnet to (AITTL/DPS) **calahans** or a machine that matches the SSIT Manager host.

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

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

4 The directory used for containing the PDPS ESDT metadata ODL files. is **/usr/ecs/<mode>/CUSTOM/data/DPS/ODL**

5 From the SSIT Manager, click on the **Tools** menu, then choose **PDPS Database** and then **SSIT Science Metadata Update**.

- An xterm with title "SSIT: Science Metadata Database Update" will be displayed.

6 At the program prompt **Configuration Filename (enter for default: ../../cfg/EcDpAtRegisterPGE.CFG)?**

- Press **Return**.

7 At the program prompt **ECS mode of operation?**, type ***mode***, press **Return** or just press **Return** if the default shown is correct.

- The ***mode*** refers to the database used and will typically be **OPS** or **TS1.**

8 At the program prompt **PGE name (max 10 characters)?**, type ***PGEname***, press **Return**.

- The ***PGEname*** is the name of the PGE that will be registered. This name must match the PGE name specified.

9 At the program prompt **PGE version (max 5 characters1)?**, type ***PGEversion***, press **Return** or just press **Return** if the default shown is correct.

- The ***PGEversion*** is the version of the PGE that will be registered. This version must match the PGE version specified.

10 At the program prompt **PGE Profile ID (0-99, 0 means null)?** Type in a valid profile ID and press **Return,** or if already listed just press **Return.**

- The PDPS database will then be updated with the information contained in the file **PGE_*PGEname#PGEversion#ProfileID*.odl**

11 At the program prompt **Hit return to run again, q <return> to quit:**, press **Return** to update the PDPS database with another PGE ODL metadata file or type **q** and press **Return** to quit.

- If you make a mistake entering any values, press **Return** here; your previous entries are restored as defaults and you won't have to retype them.
- NOTE: If you make mistakes while editing the PGE and ESDT ODL files, you can run the ODL checker (Tools → PDPS Database → Check ODL) via the SSIT manager to locate any errors.

  ODL files must have been created to define the PGE to PDPS. Examples of the ODL files are under the data directory: PGE_ODL.template, ESDT_ODL.template, ORBIT_ODL.template, TILE_ODL.template and PATHMAP_ODL. A tool can be run to generate a template ODL file for the PGE from the SSIT Manager via Tools->PDPS Database->PCF Odl Template script. This then has to be populated with all information that can not be garnered from the PCF. The CheckOdl tool from the SSIT Manager via Tools->PDPS Database->Check ODL can be used to flag any errors in ODL before trying to put it in the database.

Sample of ESDT.odl files being established in ECS

```
home/emcleod/MODIS/STORE/PGE07
calahans{emcleod}10: ls
ESDT_MD10L2#001.odl     MOD_PR10          pge_cfg
ESDT_MD35L2#001.odl     MOD_PR10.mk       scf_cfg
ESDT_MOD02H#001.odl     PGE07.mk          script
ESDT_MOD03#001.odl      doc
calahans{emcleod}11: cp ESDT_MD10L2#001.odl ESDT_MD35L2#001.odl
ESDT_MOD02H#001.odl ESDT_MOD03#001.odl /usr/ecs/OPS/CUSTOM/data/DPS/ODL/
```

AlternativeTool for SSIT Metadata Update:

Source the buildrc file for the mode in which you are working (*source .buildrc)*.

/usr/ecs/<MODE>/CUSTOM/utilities, Note that this only has to be done once per login. Then **(*cd* /usr/ecs/<MODE>/CUSTOM/bin/DPS)**

(The tool can also be executed by being in the **/usr/ecs/<MODE>/CUSTOM/bin/DPS** and executing **EcDpAtDefinePGE..)**

Shell script prompts user for information.

1 Enter in the location of the configuration file **(../../cfg/EcDpAtRegisterPGE.CFG).**
2 Filename Enter the MODE of operation (<MODE>).
3 Enter name of PGE (it must match what is in the PGE ODL file).
4 Enter the version of the PGE (it must match what is in the PGE ODL file).
5 Enter the Profile ID (it must match what is in the PGE ODL file). Note that the ODL file for the PGE must have the of: PGE_<PGE NAME>#<PGE VERSION>#<PROFILE ID>.

- Each ODL file is displayed as it is processed. A good status message should be displayed as a result. Information about the PGE (inputs and outputs, Production Rules, etc) should be entered in the Database.

11.13.3.1 Examples of PGE and ESDT ODL Files for Each Instrument Team

This section is taken from the latest **Green Book 162-TD-001-005** and are listed in Appendix C. Depicted are examples of ODL files in SSI&T activities. Then, examples of specific ODL files are listed by instrument (ASTER, MISR or MODIS).

Template ODL Files

There are three Template ODL files listed therein. The specific or tailored ODL files listed were derived from these templates by appropriate editing and filling-in of values. The three ODL Template files listed reside, on the AIT Sun host, at **/usr/ecs/<mode>/CUSTOM/data/DPS** . They are

PGE_ODL.template
ESDT_ODL.template
ORBIT_ODL.template

Example of a successful PDPS Science Metadata Update:

PDPS/SSIT SCIENCE Metadata Database Update **
Configuration filename? (enter for default: ../../cfg/EcDpAtRegisterPGE.CFG)
ECS Mode of operations? (enter for default: OPS)
OPS
PGE name (max 10 characters)?
PGE07
PGE version (max 5 characters)?
001
PGE Profile ID (0-99, 0 means null)? (enter for default: 1)
1
Warning: Could not open message catalog "oodce.cat"
EcDpAtRegisterPGE: Process Framework: ConfigFile
../../cfg/EcDpAtRegisterPGE.CFG ecs_mode OPS
' PGE profile id = '1' ...
Do you wish to overwrite the previous PGE PGE07((y)es or (n)o):
y
FILES PROCESSED
: PGE SCIENCE ODL file =/usr/ecs//OPS/CUSTOM/data/DPS/ODL/PGE_PGE07#0#001.odl

```
  ESDT SCIENCE ODL file = /usr/ecs//OPS/CUSTOM/data/DPS/ODL/ESDT_MOD02H#001.odl
  ESDT SCIENCE ODL file = /usr/ecs//OPS/CUSTOM/data/DPS/ODL/ESDT_MD35L2#001.odl
  ESDT SCIENCE ODL file = /usr/ecs//OPS/CUSTOM/data/DPS/ODL/ESDT_MOD03#001.odl
  ESDT SCIENCE ODL file = /usr/ecs//OPS/CUSTOM/data/DPS/ODL/ESDT_MD10L2#001.odl
********** Update of PDPS/SSIT database with  PDPS SCIENCE metadata SUCCESSFUL **********
Hit return to run again, 'q <return>' to quit:
```

11.13.4 Operational Metadata

The SSIT version of the PDPS database is initialized and updated with SSIT Operational Metadata so that the Planning and Processing Subsystem can schedule and run PGEs. Here, PDPS Operational Metadata refers to PGE information which is supplied to the DAAC/SSIT Operator and may change frequently.

The operator enters this data directly into the SSIT Operational Metadata Update GUI. The program then writes the data directly to the SSIT version of the PDPS database.

Before running the SSIT Operational Metadata Update from the SSIT Manager, you must first update the PDPS with SSIT Science Metadata. In addition, to get initial PGE Performance data which will be entered into the GUI, you need to run the profiling utility, EcDpPrRusage on the PGE or have the information on profiling provided. See section 11.13.2.

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

Assumptions:

1. The required UNIX environment variables have been set.
2. The Science metadata has been updated to the PDPS database for this PGE.

To update the SSIT version of the PDPS database with operational metadata, execute the steps that follow:

1 From the SSIT Manager, click on the **Tools** menu, then choose **PDPS Database** and then **SSIT Opnl Metadata Update**.

- The PDPS/SSIT Database Update GUI will be displayed.

2 Click on the radio button labeled **NEW PGE** in the lower left quadrant.

- The PGE that you are working on should appear in the subwindow labeled **PGE Names** along with its version number in the subwindow labeled **PGE Versions.**

3 In the subwindow labeled **PGE Names**, click on a PGE name. Then in the subwindow labeled **PGE Versions**, click on the PGE version for that PGE. Then click on the button labeled **EDIT**.

- The PGE name and version will be highlighted when you click on them.
- The page tabs **PROFILE**, **RUNTIME**, and **ESDT** will change from gray (indicating disabled) to black (indicating enabled).
- To see the contents of PGE Metadata, click on the button labeled **DISPLAY** and then click on the button labeled **DONE.**

- If the PGE name and/or version does not appear in the lists, it means that updating of PDPS database with PGE metadata was not successful.

4 Click on the **PROFILE** page tab.

- The Profile page will be displayed.

5 In the fields under the label **Performance Statistics**, enter the information specified.

- In the field labeled **Wall clock time**, enter the amount of wall clock time it takes for one execution of the PGE, in seconds. The tab **PROFILE** will change from black (indicating enabled) to red (indicating database needs to be updated by APPLY button).
- In the field labeled **CPU time (user)**, enter the so-called *user* time of the PGE, in seconds. This value should come from profiling the PGE .
- In the field labeled **Max memory used**, enter the maximum amount of memory used by the PGE, in megabytes (MB). This value should come from profiling the PGE .
- In the fields labeled **Block input ops** and **Block output ops**, enter the integer number of block inputs and block outputs, respectively. These values should come from profiling the PGE .
- In the field labeled **Swaps**, enter the integer number of page swaps from the PGE. This value should come from profiling the PGE .
- In the field labeled **Page faults**, enter the integer number of page faults from the PGE. This value should come from profiling the PGE .

6 In the fields under the label **Resource Requirements**, enter the information specified.

- In the field labeled **DISK SPACE used for PGE run**, enter the maximum amount of disk used by the PGE during execution, in megabytes (MB). Space should be allowed for the executable(s), input files, output files, ancillary files, static files, MCFs, and the PCF. (This number should also be in the PGE metadata ODL file; yes, there is duplication here.)
- Click on the radio button labeled **Proc. String** (if not already clicked on).
- A list of processing strings should appear in the scrollable window to the left of the two radio buttons **Proc. String** and **Computer Name**. Nominally, only one item should be listed and should be highlighted.
- In the field labeled **Number of CPUs**, the number 1 should appear.

7 Once the fields on the **PROFILE** page have been completed, click on the **APPLY** button.

- This will update the PDPS database with the information just entered. The tab **PROFILE** will change from red (indicating database needs to be updated) to black (indicating enabled).
- An information box will be displayed; click on **Ok**.
- To start over, click on the **RESET** button. This will clear all fields.

8 Click on the **File** menu and select **Exit**.

- This will end the session with PDPS/SSIT Database Update and the GUI will disappear.

11.13.5 SSIT Operational Metadata Update GUI

The SSIT version of the PDPS database is initialized and updated with SSIT Operational Metadata so that the Planning and Processing Subsystem can schedule and run PGEs. Here, PDPS Operational Metadata refers to PGE information which is supplied to the DAAC/SSIT Operator and may change frequently.

The operator enters this data directly into the SSIT Operational Metadata Update GUI (Figure 11.13.5-1). The program then writes the data directly to the SSIT version of the PDPS database. The SSIT Operational Metadata Update GUI is used to view or update the following operational parameters for a particular PGE:

- Performance parameters for the PGEs.
- Resource parameters for the PGEs.
- PGE user-defined static parameter.
- View the PGE science metadata file.

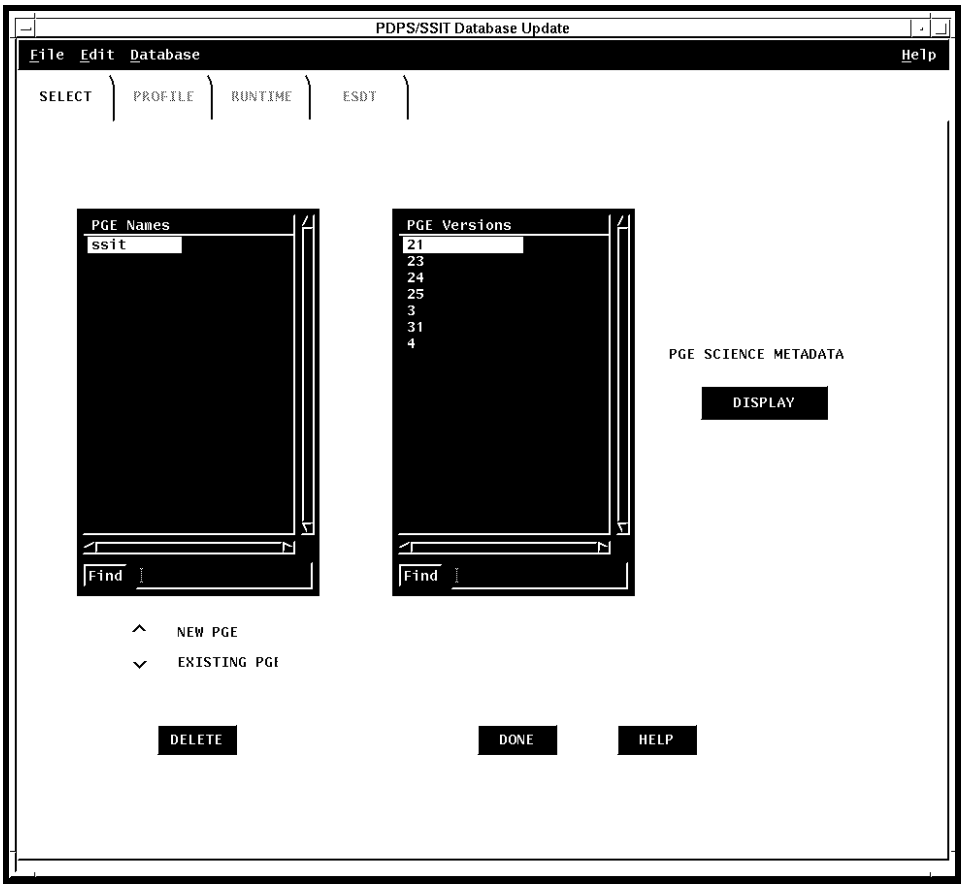


Figure 11.13.5-1. SSIT Database Operational Metadata Update GUI – SELECT view

11.13.6 Test Data Preparation and Insertion of Data Granules

This section describes how to prepare test data for use by registered PGEs. When PGEs are first delivered to the DAAC and registered within the PDPS, they will typically be run in isolation. That is, they will be run without any PGE dependencies. For this testing to be possible, test input data granules required by the PGE need to be pre-Inserted to the Data Server.

Data granules can be *dynamic* or *static*. Dynamic data granules are those whose temporal locality differs for each instance of the granule. Examples of dynamic granules are Level 0, Level 1, and Level 2 data sets. Static data granules are those whose temporal locality is static over long periods of time. Examples of static granules are calibration files which may only change with a new version of a PGE. For any granule to be Inserted to the Data Server, a Target MCF is needed (also known as an ASCII metadata ODL file or a **.**met file).

In the actual production environment, a Target MCF is produced by the PGE during execution. Thus, the data granule can be Inserted. In isolation testing of a PGE, however, the inputs needed by it will not have been Inserted by a previous PGE in the chain. This Insertion must be done manually. The next two sections describes how to use the Source MCF for a dynamic data granule to create a Target MCF. and then describes how to do the Insert. In this way, a dynamic data granule can be Inserted to the Data Server as if a PGE had produced it.

11.13.6.1 Generating a Metadata Configuration File (Source MCF)

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

Assumptions:

1. The SSIT Manager is running.
2. ESDT's are installed onto the **Science Data Server.**

To Generate the Metadata Configuration File (Source MCF) for the input and output ESDT's, execute the steps that follow.

1 From the SSIT Manager, click on the **Tools** menu, then choose **Data Server** and then **Get MCF.**

- An xterm in which EcDpAtGetMCFis running will be displayed as SSIT: Acquire MCF..
- Alternatively, the same tool can be invoked by typing at a UNIX prompt on an AIT Sun **EcDpAtGetMCF.sh**, press **Return**.

2 At the program prompt **Configuration Filename (default *defaultConfigFile*)?**

- Type in **../../cfg/** ***defaultConfigFile*** and press **Return**.
- The ***defaultConfigFile*** will be replaced by the full path name and file name of the default configuration file. The file name will be **EcDpAtGetMCF.CFG** where *daac* will be replaced by one of {GSFC, EDC, LARC, NSIDC}.

3 At the program prompt **ECS mode of operation (enter for default: def*aultMode*)?**, type ***mode***, press **Return** or just press **Return** if the default shown is correct.

- The ***mode*** refers to the database used and will typically be **TS1**.

4 At the program prompt **ESDT Short Name?**, type ***ESDT ShortName***, press **Return**.

- The ***ESDTShortName*** is the name of the ESDT that the EcDpAtGetMCF tool will use to generate the MCF.

5 At the program prompt **ESDT Version?**, type ***ESDTversion***, press **Return** or just press **Return** if the default shown is correct.

- The ***ESDTversion*** is the version of the ESDT.

6 At the program prompt **Directory to receive MCF (must be full path)?**, type ***MCFpathname***, press **Return**.

- The ***MCFpathname*** is the full path name to the location where the source MCF will be placed. For example, /home/jdoe/ssit.

7 To the final prompt **Hit return to run again, 'q <return> to quit:**, press **Return** to generate another Source MCF or type **q** and press **Return** to quit.

- If you make a mistake entering any values, press **Return** here; your previous entries are restored as defaults and you won't have to retype them.

Example of a successful installation of a Source MCF:

Configuration filename? (enter for default: ../../cfg/EcDpAtGetMCF.CFG)
ECS Mode of operations?
OPS
ESDT Short Name?
MOD03EM
ESDT Version?
0
Directory to receive MCF? (must be full path)
/home/emcleod/MCF/
Warning: Could not open message catalog "oodce.cat"
EcDpAtGetMCF: Process Framework: ConfigFile ../../cfg/EcDpAtGetMCF.CFG
ecs_mode
OPS
incomplete group entries in the configfile,using default G1
Request for MCF successful for:
ESDT name = 'MOD03EM'
ESDT version = '0'
directory = '/home/emcleod/MCF/'
Hit return to run again, 'q <return>' to quit:

11.13.7 Creating a Target MCF (.met) for a Dynamic/Static Granule

A Target MCF file for a corresponding data granule can be created based on the information provided in the Source MCF file and the involved science software package (PGE).

In standalone or isolation testing of a PGE, the inputs needed by it will not have been Inserted by a previous PGE in the chain. This Insertion must be done manually. A Target MCF file for a

corresponding data granule is required to run a standalone PGE. This way a dynamic data granule can be Inserted to the Science Data Server as if a PGE had produced it.

Creating a Metadata ODL File for a Static Granule

1 At the UNIX prompt on the AIT Sun, type cd WorkingPathname, then press the Enter key. Example: **cd /usr/ecs/{MODE}/CUSTOM/data/DPS/ODL/**

The ***WorkingPathname*** *is* the full path name of the working directory containing the template metadata ODL file.

2 At the UNIX prompt on the AIT Sun, type **cp *StaticODLmet.tpl filename*.met**, then press the **Enter** key.

The ***StaticODLmet.tpl*** is the file name of the template Target MCF.

The ***filename*.met** is the file name of the Target MCF for this static file. The file name extension must be .met.

This command will copy the template Target MCF to ***filename*.met**. For example, type **cp StaticODLmet.tpl CER11T.mcf.met**, then press the **Enter** key.

3 At a UNIX prompt on the AIT Sun, type **vi *filename*.met**, then press the **Enter** key.

This command invokes the *vi* editor and reads in the Target MCF created above.

4 Edit the Target MCF with the specific information for the static data granule to be Inserted. The following guidelines should be followed when editing on the template MCF:

The value for the ShortName object should be filled out with proper instrument name.
The value for the Version ID object should be filled out with the proper version number.
In the **INFORMATIONCONTENTCONTAINER** object enter the following:

- The value for the **PARAMETERNAME** object of the class "1" should be filled out with the name of static data file.
- The value for the **PARAMETERVALUE** object of the class "2" should be filled out based on the following guideline:
  - If the data granule is a coefficient file, a "C" followed by a numerical number n (n=1,2,….) will be used. Here n stands for the number of the coefficient file.
  - If the data granule is a MCF file, a "M" followed by a numerical number n (n=1,2,….) will be used. Here n stands for the number of the MCF file.

5 Save the changes made to the Target MCF (***filename*.met**) and exit the editor.

The specifics depend upon which editor is being used. If using *vi*, the command sequence to enter is **:wq**, then press the **Enter** key.

11.13.8 Inserting Static Data Granules into the Data Server

Inserting a Static Data File:

The following Servers/Services must be up and operational:

Science Data Server, Storage Management.

The following must have occurred between those Servers/Services:

The ESDT of the static file must have been installed at the Data Server.

What the user must do before trying SSIT functionality:

Create a metadata file for the static file to insert. To do this, an MCF (See "Getting an MCF in this section") must be gotten from the Data Server for the ESDT of the file to insert. Mandatory fields are filled into the MCF, creating a metadata file.

If the tool is NOT run from the SSIT Manager then go to the executables directory **(cd /usr/ecs/<MODE>/CUSTOM/utilities)**

Source the buildrc file for the mode in which you are working (source .buildrc). Note that this only has to be done once per login.

If the tool is NOT run from the SSIT Manager then go to the executables directory (**cd /usr/ecs/<MODE>/CUSTOM/bin/DPS)**

From the **SSIT Manager** choose **Tools** menu and then **Data Server** submenu. Choose **Insert Static File.**

The tool can also be executed by being in the **/usr/ecs/<MODE>/CUSTOM/bin/DPS and executing EcDpAtInsertStatic**

Shell script prompts user for information.

1 Enter in the location of the DpAtInsertStaticFile configuration file **(../../cfg/EcDpAtInsertStaticFile.CFG).**

2 Enter the MODE of operation (<*MODE*>). At the program prompt **mode (default ops)?**, or press **Enter** to take default.

3 Enter the short name of the ESDT (for the static file). This value is in the pdps database under the PlDataTypeMaster table and must be in the PGE ODL file.

At the program prompt **ESDT name?** type ***ESDTShortName***, then press the **Enter** key. For example type: **MOD02LUT.**

4 Enter the version of the ESDT for the static file. This value is also in the pdps database under the PlDataTypeMaster table and must be in the PGE ODL file.

At the program prompt **PGE version (default 1)?**, type ***PGEVersion***, then press the **Enter** key.

The ***PGEVersion*** must match exactly the PGE version entered into the PDPS for this PGE.

5 Enter the science group for this static (this will be from the ODL created during Populating the PGE information in the Database).

At the program prompt **Science group for Static file(one of{C,L,D,O} followed by a 3 digit number)?**, type ***ScienceGroupID***, then press the **Enter** key.

The ***ScienceGroupID*** is an identifier used to define the file type as a coefficient file, a lookup table file, or a MCF. It distinguishes static granules of different types which

share the same ESDT. For instance, for a coefficient file, use **C*n***, where number *n* could be 0, 1, 2…; this number *n* needs to be matched with the number *n* in the PGE_PGEName#Version.odl file. For an MCF. For example, type **C001**, press **Return.**

- The Science Group ID must match what was edited into the PGE metadata ODL file for that PCF entry.

6 At the program prompt **Is there more than one data file for this Static (Y = Yes, N = No)? (enter for default: N).** If there is only one data file, press **Return** and go to next step. If there are more than one data files, type **Y,** press **Return** and go to step 10.

7 At the program prompt **Single Static Filename to Insert (including FULL path)?**, type ***pathname/GranuleFileName***, press **Return**

- The ***pathname/GranuleFileName*** is the full path name and file name of the static data granule to be Inserted. For example, type **/home/MODIS/PGE10/MOD_PR28/coeff/emissivity.dat**, press **Return**.

8 At the program prompt **Associated ASCII Metadata Filename to Insert (including FULL path).** Type ***pathname/GranuleFileName.met*)?**, press **Return**.

- The ***pathname/GranuleFileName.met*** is the full path name and file name of the .met file for the associated static data granule to be Inserted. For example, type **/home/MODIS/PGE10/MOD_PR28/MOD28LUT.met** press **Return**.

9 At the program prompt **Directory where all data files and .met file exist (FULL path)?** Type ***pathname*** press **Return**.

- where ***pathname*** is the full path of the directory where all data files and .met file exist.
- Note for a multifile granule, the data files and .met file should be placed in the same working directory.

10 At the program prompt **Name of MFG file (enter to end list)?** Type in the ***GranuleFileName,*** one at a time and press **Return.** To end the list press **Return.**

- where ***GranuleFileName*** is the names of the multifile granules.

11 At the program prompt **Associated ASCII Metadata Filename to Insert?** Type ***GranuleFileName.met***, press **Return.**

- where ***GranuleFileName.met*** is the name of one .met file that is used with all data granules in the even of a multifile granule.
- The dynamic data granule will be Inserted to the Data Server. For reference, the Data Server Universal Reference (UR) will be printed on the screen.

12 At the program prompt **Hit return to run again, 'q <return>' to quit:** type **q** and press **Return** to quit or just press **Return** to insert additional dynamic granules.

- If continuing, repeat steps 2 through 9.

11.13.9 Inserting Dynamic Data Granules to the Science Data Server

In order for dynamic data files to be used both during the SSI&T and in production, this file must exist in the Data Server and be accessible by the local machine. A program called the Insert Test Dynamic File can be used for Inserting a dynamic data granule into the Data Server.

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

Assumptions:

1. The ESDT's have been installed on the Data Server.
2. The Target MCF for this data granule has been created for the Insert.

To Insert a dynamic granule to the Data Server, execute the following steps:

1 From the SSIT Manager, click on the **Tools** menu, then choose **Data Server** and then **Insert Test Dynamic**.

- An xterm with title "SSIT: PGE Test Dynamic Input File Insertion" will be displayed.

2 At the program prompt **Configuration filename? (enter for default: ../..cfg/EcDpAtInsertTestFile.CFG)**, press **Return**.

3 At the program prompt **ECS Mode of operations?**

- Type in the **<mode>** you are working in. For example, **TS1** or **OPS.** Press **Return.**

4 At the program prompt **ESDT short name for the file(s) to insert?** type ***ESDTShortName***, press **Return**

- The ***ESDTShortName*** is the ShortName of the ESDT descriptor file corresponding to this granule to be Inserted. For example, type **MOD021KM** press **Return**.

5 At the program prompt **ESDT Version for the file(s) to insert?** Type in the ESDT version and press **Return.**

6 At the program prompt **Is there more than one data file to this Dynamic Granule (Y = Yes, N = No)? (enter for default: N)?** If there are no multifiles for this ESDT, press **Return** and go to step 7. If there are more than one file for this granule go to step 9.

7 At the program prompt **Single Filename to Insert? (including FULL path)** type ***pathname/GranuleFilename***, press

- The ***pathname/GranuleFileName*** is the full path name and file name of the data granule to be Inserted. For example, type **/home/MODIS/PGE10/MOD021KM.A1996217.0014.002.hdf**, press **Return**.

8 At the program prompt **Associated ASCII Metadata Filename to Insert (including FULL path)** , Type ***pathname/GranuleFileName.met*** and press **Return**.

- ***pathname*** is full name of the path and ***GranuleFileName.met*** is the name of the associated .met file. For example, **/home/MODIS/PGE10/MOD021KM.met**
- The dynamic data granule will be Inserted to the Data Server. For reference, the Data Server Universal Reference (UR) will be printed on the screen.

9 At the program prompt **Directory where all data files and .met file exist (FULL path)?** Type ***pathname*** press **Return**.

where ***pathname*** is the full path of the directory where all data files and .met file exist.
Note for a multifile granule, the data files and .met file should be placed in the same working directory.

10 At the program prompt **Name of MFG file (enter to end list)?** Type in the ***GranuleFileName,*** one at a time and press **Return.** To end the list press **Return.**

where ***GranuleFileName*** is the names of the multifile granules.

11 At the program prompt **Associated ASCII Metadata Filename to Insert?** Type ***GranuleFileName.met***, press **Return.**

where ***GranuleFileName.met*** is the name of one .met file that is used with all data granules in the even of a multifile granule.

- The dynamic data granule will be Inserted to the Data Server. For reference, the Data Server Universal Reference (UR) will be printed on the screen.

12 At the program prompt **Hit return to run again, 'q <return>' to quit:** type **q** and press **Return** to quit or just press **Return** to insert additional dynamic granules.

- If continuing, repeat steps 2 through 8.

Example of a successful insertion of a Dynamic Input Data Granule into the Data Servers:

PGE Test Dynamic Input File Insertion **
Configuration filename? (enter for default:
../../cfg/EcDpAtInsertTestFile.CFG)
ECS Mode of operations? (enter for default: OPS)
OPS
ESDT name
MOD02H
ESDT Version (enter for default: 1)
0
Staged Filename to Insert? (including FULL path)
/home/emcleod/MCF/MOD02HKM.A1997217.1730.002.hdf
Associated ASCII Metadata Filename to Insert? (including FULL path)
/home/emcleod/MCF/MOD02H.met
Warning: Could not open message catalog "oodce.cat"
EcDpAtInsertTestFile: Process Framework: ConfigFile
../../cfg/EcDpAtInsertTestFile.CFG ecs_mode OPS
incomplete group entries in the configfile,using default G1
Trying to make a request to [MDC:DSSDSRV]
incomplete group entries in the configfile, using default
Trying to make a request to [MDC:DSSDSRV]
incomplete group entries in the configfile, using default
Insert to Data Server successful:
ESDT Version = '0'

```
  staged file = '/home/emcleod/MCF/MOD02HKM.A1997217.1730.002.hdf'
  metadata file = '/home/emcleod/MCF/MOD02H.met'
Inserted at UR:
'UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[MDC:DSSDSRV]:16:SC:M
OD02H:1757'
Hit return to run again, 'q <return>' to quit:
```

11.13.10 Science Server Archive Package (SSAP)

The SSAP is used to provide a record of the science software, documentation, and other related files stored at the DAAC. The SSIT SSAP GUI provides a method for grouping required data about a PGE.

The SSAP is not to be confused with the Delivered Algorithm Package (DAP) received from the SCF. Much of what is in the DAP will make it into the SSAP. The key difference is that SSAP data is prepared after initial testing of the science software and will include data that reflects site integration as well as fixes required for performance at the DAAC.

The SSAP is made up of 2 different data types. The first data type is the Algorithm Package which contains metadata (name of the PGE, name of the instrument, date accepted, etc...) about the SSAP. The second data type is the source code, documentation, and test data which will be stored as a SSAP, with its own metadata in addition to the files. SSAP components such a source code will be tared to retain the directory structure.

The executables and static files are stored separately from the SSAP and will have their own data types (ESDTs).

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

1. The SSIT Manager is running.
2. The PGE has been successfully built with the SCF and DAAC version of the Toolkit.

11.13.10.1 Creating an SSAP

The following Servers/Services must be up and operational:

Science Data Server, Storage Management.

The following must have occurred between those Servers/Services:

NONE.

What the user must do before trying SSIT functionality:

1 From the SSIT Manager choose **Tools** menu and then **Data Server** submenu. Choose **SSAP Editor.**
The GUI starts. Note that it will first query Data Server for a list of SSAPs that have previously been created. This list will appear in the window at the center (if any SSAPs

already exist). Current SSAP field will be blank, and only Refresh and Create buttons will be active. All three tabs (Main, Files, and Metadata) will be active.
The SSAP GUI will be displayed.

2 Click on the **Create** to create a new SSAP.

- The **Create SSAP** window appears. If no OK button is visible, resize the window such that the OK button is visible.

3 Enter the name of the SSAP in the first field . Enter SSAP version in the second field. Note that version has a limit of 20 characters.

4 Click OK and the window disappears
On the main GUI, the SSAP created (what was entered in the step above) will appear. Current SSAP is now set to that value. All buttons are now active.

5 To set up the SSAP components, click on the **File List** tab.

- The File List Tab displays files in the local directory to the left and files in the selected SSAP component to the right. On the bottom left is a directory listing and a method to move through the directory tree on the local machine. Delete and Reset buttons—both active – are to the right.

6 To select a file in the left column, click on the **File Type** button, highlight a file (or files) and click on the **Add** arrow button to add the files..
The files selected to be added will be displayed in the right column.
T o change directories (and thus add files from other directories to the SSAP component), click on the listing in the window on the bottom left of the GUI. The ".." is to go up one directory level. A single click with move to the directory chosen and change the display to show the directories under the new current directory. Note that the list of files in the upper left window changes to show the files within the current directory.

7 To add metadata for the new SSAP, select the **Metadata** tab.
The Metadata window will be displayed.
The Metadata Tab displays the metadata for the new SSAP. Only the Name and Version will be filled in automatically. The rest of the fields will have default information. While the SSAP can be submitted with the default information, it is wise to fill in valid values. To change a value:

8 To change the default information, click on the **Edit Assoc Collections** button.
The **Edit Associated Collections** window displays a list of associated collections and fields for the entry of new ShortNames and Versions.

9 Enter a ShortName, and version (of the ESDT that has been installed in the Data Server) - must be eight or fewer characters. Note that the Data Server will verify if the Shortname exists.

10 Enter the version (of the installed ESDT).
Then select the **OK** button. Select **Done** to close the window.

11 To save the updated metadata, click **Save** on the **Metadata** tab**.**

12 To get back to the **Main** tab**,** select the **Main** tab button.

13 To submit the new SSAP to the Data Server, select the **Submit** button.

When the SSAP has been submitted, the **SSAP Successfully inserted to the Data Server** prompt will appear.

11.13.10.2 Updating an SSAP

The following Servers/Services must be up and operational:

Data Server, Storage Management.

The following must have occurred between those Servers/Services:

An SSAP must have already been inserted to the Data Server.

What the user must do before trying SSIT functionality:

The SSAP Editor has been used to insert an SSAP to the Data Server.

What must be done via SSIT tools:

If SSAP Editor is not running, use the directions from the first 2 paragraphs of (Creating an SSAP) to bring up the SSAP GUI. Note that the added SSAP should appear in the window of the Main tab.

If the SSAP Editor is already running, the added SSAP should appear in the window of the Main tab.

---

1 Click on added SSAP in the main display.

2 Click on the Metadata tab to update the SSAP.
The Metadata Tab displays the metadata for the SSAP. All fields will be set to the values entered when the SSAP was created, and the Algorithm Name field will be grayed out (because it may not be updated). If you want to create a new SSAP from the an existing one, go back to the Main tab and hit the Create With button.

3 Click on the Algorithm Version field (currently called Algorithm Description) and enter a new version (different from what is in the field when the tab is clicked).

4 Update any other fields that you wish to change. You can even add a new Associated Collection by clicking on the Assoc Collection button and following the steps described in Creating an SSAP.

5 Before you leave the Metadata tab, click Save to save the updated metadata.

6 Click on the File List tab to set up new SSAP components.
The File List Tab displays files in the local directory to the left and files in the selected SSAP component to the right. On the bottom left is a directory listing and a method to move through the directory tree on the local machine. Delete and Reset buttons—both active –are to the right.

7 Click on the File Type button to select the additional SSAP component to manipulate.
Choose one of the menu items.
Select a file (or files) from the left window to add to the component.

Click the Add Arrow button to add the files. They will appear in the right window because they are now part of that SSAP Component.
Click Main to get back to the Main tab.
On the Main tab:
Click Submit to send the new SSAP to Data Server. When finished, a message should pop up that says "SSAP Successfully inserted to the Data Server".

11.13.11 PGE Checkout

The following Servers/Services must be up and operational:

NONE.

The following must have occurred between those Servers/Services:

NONE.

What the user must do before trying SSIT functionality:

In normal SSIT (at the DAACs) the DAP would be untared, and the source code recompiled and tested.

What must be done via SSIT tools:

Since SSIT is just a calibration of various tools, there is no specific order for which they must be run. All tools can be started from the SSIT Manager and can be executed on their own.

For SparcWorks (for code analysis and debugging), choose Tools menu and then Code Analysis submenu. See SparcWorks manuals for SparcWorks operation.

For various office tools, choose Tools menu and the Office Automation submenu. Choose from MS Windows (a simulator to allow the user to run Windows programs), Ghostview (a viewer), Netscape (for web access), Acrobat (for document viewing), and DDTS (for problem reporting).

For Standards checkers, choose Tools and then the Standards Checkers submenu.

FORCHECK is a COTS Fortran language checking program.

The Prohibited Function Checker will examine source code for functions that are not permitted On Prohibited Function Gui, choose Analyze to select files to examine. Hit the Ok button once selections are made and a message at the top of the Gui will indicate if prohibited functions have been found. If prohibited functions HAVE been found, use the View button to view the source code with the prohibited call. The Help button gives further information on how to work the Gui.

The Process Control File Checker examines selected PCFs and highlights any errors. The Process Control File Gui allows the user to work through the directory structure on the local machine and select PCFs to be checked. Click the Check PCF button to check a selected PCF. Again, the Help button provides more information.

11.13.12 Placing the Science Software Executable (SSEP) on the Data Server

In order to be able to run a PGE within the ECS system, the EXE TAR file has to be inserted to the Science Data Server. This tar file consists of all files needed to run a PGE, except for input data files. This includes the executables, any scripts, and the SDP Toolkit message files.

Assembling a Science Software Executable Package (SSEP)

This section describes how to assemble a Science Software executables Package (SSEP) and create a corresponding Target MCF. A SSEP is a UNIX tar file which contains PGE executables and SDP Toolkit message files.

In order to Insert a PGEEXE tar file into the Science Data Server, a corresponding Target MCF (.met) must be generated before insertion. Such an ASCII metadata ODL file can be obtained by editing an existing template ODL file with the information of the specific PGE. The following procedures describe how to assemble a PGEEXE tar file and create an ASCII metadata ODL file.

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

Assumptions:

1. PGE executables and message files required by this PGE are available to make a SSEP.

To create an SSEP, execute the steps that follow:

1 At the UNIX prompt on an AIT Sun, type **mkdir *SSEPpathname***, press **Return**.

- The ***SSEPpathname*** is the full path name of a *new* directory which will contain all the files to be placed into the SSEP as well as the SSEP itself.
- It is recommended that ***SSEPpathame*** be named with a convention that indicates the PGE for which a SSEP will be created. For example, type **mkdir PGE35.ssep**, press **Return**.

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

- The ***SSEPpathname*** is the directory name of the new directory created in step 1.

3 At the UNIX prompt on the AIT Sun, type **cp *pathname/file1 pathname/file2 … pathname/filen* .**, press **Return** (note the "dot" and then space at the end of the command).

- The ***pathname/file1, pathname/file2,…pathname/filen*** represents a list of path names and file names (delimited by spaces) to copy into the current directory, ***SSEPpathame*** (the "dot" represents the current directory and must be last in the command).
- For example, type **cp /data/MODIS/pge/PGE35.exe /data/MODIS/mcf/mod35.mcf /data/MODIS/MOD_13453 .**, press **Return** (note the space and then "dot" at the end of the command).
- The files copied into this directory should be the PGE executable, any shell scripts or other executables that are part of the PGE and SDP Toolkit message files.
- Files can be individually copied into the ***SSEPpathame*** directory. For example, type **cp /data/MODIS/pge/PGE35.exe .**, press **Return** (note the space and then

"dot" at the end of the command). Repeat for each file needed in the SSEP for this PGE.

4 At the UNIX prompt on the AIT Sun, type **tar cvf *SSEPfilename*.tar ***, press **Return**.

- The ***SSEPfilename*.tar** is the file name for the SSEP tar file. The file name extension .tar is recommended but not required.
- The asterisk (*****) is a file name wildcard that represents all files in the current directory. This will place all files in the SSEP tar file.
- Once created, the contents of the SSEP tar file can be viewed by typing **tar tvf *SSEPfilename*.tar**, press **Return**.
- Do not apply compression (*e.g.* UNIX compress or gzip) to the tar file.

5 At the UNIX prompt on the AIT Sun, type **cp *filename.met.tpl filename.met*,** press **Return**.

- The ***filename.met.tpl*** is the file name of the template Target MCF for this SSEP. If a template is not available, see Appendix D or use one used for another SSEP.
- The ***filename.met*** is the file name of the Target MCF to be tailored for this SSEP.

6 At the UNIX prompt on the AIT Sun, type **vi *filename.met***, press **Return**.

- The ***filename.met*** is the Target MCF for this SSEP.
- This command invokes the *vi* editor. Edit the ***filename*.met** with the specific information for the SSEP to be inserted.
- The following guidelines should be followed when editing on the Target MCF (***filename*.met**):
  - The value for the VERSIONID object should be filled out with the proper PGE version. For example: "1" .
  - In the INFORMATIONCONTENTCONTAINER object,
    - The value for the PARAMETERNAME object of the class "1" should be filled out with the PGE name. For example: "BTS".
    - The value for the PARAMETERNAME object of the class "2" should be filled out with the PGE Science Software Version. For example: "1".
    - The value for the PARAMETERNAME object of the class "3" should be filled out with the Platform Name. For example: "IRIX".
    - The value for the PARAMETERNAME object of the class "4" should be filled out with the Platform Version. For example: "6.2".
    - The value for the PARAMETERNAME object of the class "5" should be filled out with the date to perform the Insertion. For example: "970319".
    - The value for the PARAMETERNAME object of the class "6" should be filled out with the time to perform the Insertion. For example: "14:45:00".

7 Save the changes made to the SSEP's Target MCF (***filename*.met**) and exit the editor.

- The specifics depend upon which editor is being used. If using *vi*, the command sequence to enter is **:wq**, press **Return**.

For other editors, refer to that editor's documentation

1 At the UNIX prompt on the AIT Sun, type **mkdir *SSEPpathname*** then press the **Enter** key. For example, type **mkdir MOD35.ssep, press Enter.**

The ***SSEPpathname*** is the full path name of a *new* directory which will contain all the files to be placed into the SSEP as well as the SSEP itself.

2 At the UNIX prompt on the AIT Sun, type **cd *SSEPpathname***, then press the **Enter** key.

The ***SSEPpathname*** is the directory name of the new directory created in step 1.

3 At the UNIX prompt on the AIT Sun, type **cp *pathname/file1 pathname/file2 … pathname/filen* .**, then press **Enter** (note the space then the "dot" at the end of the command).

The ***pathname*** is the location of the files. The ***file1, file2,…filen*** represents a list of file names (delimited by spaces) to copy into the current directory, ***SSEPpathame*** (the "dot" represents the current directory and must be last in the command). For example, type **cd /data/MODIS/pge/MOD35.pge /data/MODIS/mcf/MOD35.mcf /data/MODIS/MOD_13453 ., press Enter.** (note the space then the "dot" at the end of the command).

For the synthetic PGE, only the executable needs to be copied.

4 At the UNIX prompt on the AIT Sun, type **tar cvf *SSEPfilename*.tar ***, then press the **Enter** key.

The ***SSEPfilename*.tar** is the file name for the SSEP tar file.

The file name extension .tar is recommended but not required.

The asterisk (*****) is a file name wildcard that represents all files in the current directory which will place all files in the SSEP tar file.

Once created, the contents of the SSEP tar file can be viewed by typing **tar tvf *SSEPfilename*.tar**, then press the **Enter** key.

5 At the UNIX prompt on the AIT Sun, type **cp *filename.met.tpl filename.met,*** then press the **Enter** key.

The ***filename.met.tpl*** is the file name of the Target MCF for this SSEP.

For the synthetic PGE, the **met** file has already been renamed and modified for use by the student when the file was unpacked.

6 At the UNIX prompt on the AIT Sun, type **vi *filename.met***, then press the **Enter** key.

The ***filename.met.tpl*** is the Target MCF for this SSEP.

7 Edit the ***filename*.met** with the specific information for the SSEP to be inserted.

The value for the **VERSIONID** object should be filled out with the proper PGE version.

In the **INFORMATIONCONTENTCONTAINER** object enter the following:

The value for the **PARAMETERNAME** object of the **class "1"** should be filled out with the PGE name. The synthetic PGE should be "userid".

The value for the **PARAMETERNAME** object of the **class "2"** should be filled out with the PGE Science Software Version.

The value for the **PARAMETERNAME** object of the **class "3"** should be filled out with the Platform Name.

The value for the **PARAMETERNAME** object of the **class "4"** should be filled out with the Platform Version.

The value for the **PARAMETERNAME** object of the **class "5"** should be filled out with the date to perform the Insertion.

The value for the **PARAMETERNAME** object of the **class "6"** should be filled out with the time to perform the Insertion.

8 Save the changes made to the SSEP's Target MCF (***filename*.met**) and exit the editor.

The specifics depend upon which editor is being used. If using *vi*, the command sequence to enter is **:wq**, then press the **Enter** key.

11.13.12.2 Inserting a Science Software Executable Package onto the Data Server

. Science software, like any other data that are managed in the ECS, must be placed on the Science Data Server. A program called the Insert EXE TAR Tool can be used for Inserting a Science Software Executable Package into the Data Server.

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

Assumptions:

1. The ESDT called PGEEXE has been installed on the Science Data Server.
2. A Target MCF (.met) for this PGEEXE tar file has been created for the Insert.
3. The PGEEXE tar file has been created .\
4. The following Servers/Services must be up an operational:

   Science Data Server, Storage Management.

To Insert the SSEP to the Science Data Server, execute the steps that follow:

---

1 From the SSIT Manager, click on the **Tools** menu, then choose **Data Server** and then **Insert EXE TAR**.

- An xterm with title "SSIT: PGE Executable Tar File Insertion" will be displayed.

2 At the program prompt **Configuration filename? (enter for default: ../../EcDpAtInsertExeTarFile.CFG)**, press **Return**.

3 At the program prompt **ECS mode of operations?**, Type <***mode***> press **Return.**

- <***mode***> can either be **OPS** or **TS1**.

4 At the program prompt **Name of PGE?**, type ***PGEName***, press **Return**.

- The ***PGEName*** is the name of the PGE for which this static granule is being Inserted. For example, type **PGE01**, press **Return**.

- The ***PGEName*** must match exactly the PGE name entered into the PDPS for this PGE.

5 At the program prompt **Science software version of PGE?**, type ***SSWversion***, press **Return**.

- The ***SSWversion*** is the version of the science software which is being Inserted in this SSEP. Press **Return** to accept the default or enter in a version and press **Return**.

6 At the program prompt **Staged filename to insert (including Full path)?**, type ***pathname/SSEPFileName***, press **Return**

- The ***pathname/SSEPFileName*** is the full path name and file name of the SSEP tar file to be Inserted. For example, type **/data/MOD35/ssep/PGE35_1.tar**, press **Return**.
- The SSEP tar file must not be compressed (*e.g.* with UNIX compress or gzip).

7 At the program prompt **Associated ASCII metadata filename to insert (including Full Path)? *pathname/SSEPFileName.met*)?**, press **Return**.

- The default is the file name of the granule to insert with the .met file name extension. If the default is not correct, then the file name of this file must be entered.

8 At the program prompt **Top level shell filename within tar file?**, type ***ExecFileName***, press **Return**.

- The ***ExecFileName*** is the file name of the top level executable or script within the SSEP tar file. It should be the same as was entered into the PDPS/SSIT Database Update GUI.
- The SSEP will be Inserted to the Science Data Server.

9 At the program prompt **Hit return to run again, 'q <return>' to quit:** type **q** and press **Return** to quit or just press **Return** to insert additional dynamic granules.

- If continuing, repeat steps 3 through 8.

Example of a successful insertion of a SSEP EXE TAR:

PGE Executable Tar File Insertion Script

Configuration filename? (enter for default:../../cfg/EcDpAtInsertExeTarFile.CFG)

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

OPS

Name of PGE? (enter for default: PGE07)

PGE07

Science software version of PGE? (enter for default: 0)

0

Staged filename to insert (including FULL path)? (enter for default:

/home/emcleod/SSEP/MODPGE07.tar)

Associated ASCII metadata filename to insert (including FULL path)? (enter for

default: /**home/emcleod/SSEP/MOD_PR10.tar.met**)

Top level shell filename within tar file? (enter for default:

/home/emcleod/SSEP/MOD_PR10.exe)

MOD_PR10.exe (note: this entry is done a second time) Note: If you get **core dump**, execute using "dbx command: type in: **dbx filename .exe**. This will help isolate error message that caused core dump.

Warning: Could not open message catalog "oodce.cat"

EcDpAtInsertExeTarFile: Process Framework: ConfigFile

../../cfg/EcDpAtInsertExeTarFile.CFG ecs_mode OPS

Performing INSERT.....

incomplete group entries in the configfile,using default G1

Trying to make a request to [MDC:DSSDSRV]

incomplete group entries in the configfile, using default

Trying to make a request to [MDC:DSSDSRV]

incomplete group entries in the configfile, using default

Insert to Data Server and PDPS database update successful for:

PGE name = 'PGE07'

Ssw version = '0'

ESDT = 'PGEEXE'

ESDT Version = '0'

staged file = '/home/emcleod/SSEP/MODPGE07.tar'

metadata file = '/home/emcleod/SSEP/MOD_PR10.tar.met'

Top level shell name = 'MOD_PR10.exe'

Inserted at UR:

'UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[MDC:DSSDSRV]:14:LM:PGEEXE:1787'

Hit return to run again, 'q <return>' to quit:

11.14 PGE Planning Processing and Product Retrieval

11.14.1 Using the Production Request Editor

When standalone tests (Run from the command line) have completed successfully and information about the PGE has been entered into the PDPS Database (through PGE registration), the PGE is ready to be run through the automated ECS PDPS environment.

To process Science data, a Production Request (PR) must be submitted to the ECS system. The Production Request Editor GUI accomplishes this function. Only one PR may be submitted at a time. A single PR is exploded by the PDPS into one or more jobs called Data Processing Requests (DPRs). The number of DPRs that are created for a single PR is determined by the number needed to cover the requested time interval, orbital extent and tile schema. Some PRs may only require one DPR.

11.14.2 Invoking the Production Request

Currently, the Production Request Editor is invoked from a command line script. In the future, this will be done by clicking on the icon for the PR Editor on the ECS Desktop. Once the Production Request Editor is invoked, it brings up a screen with five tabs at the top for selection (as shown in Figure 11.14.2-1). The first tab is labeled "Planning". Selection of this tab displays a list of four capabilities available for the PR Editor by selecting the other tabs at the top of the primary GUI screen: PR Edit, PR List, DPR View, and DPR List.

********* Please be advised: Only one user per mode can be executing a DPR at a time. If more than one occurs the system will cancel the remaining DPR's. Problem discovered when chained PGE's failed to kickoff. *********

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

Assumptions:

1. The PGE has been registered in the PDPS Database.
2. The PGE has been successfully compiled and linked with the DAAC version of the SDP Toolkit.
3. The required servers are up and running.

To invoke the Production Request Editor GUI, execute the procedure steps that follow:

1 In any currently available xterm window, spawn a new session: type **xterm &**, press **Return**. Then telnet to the PLS host.

- It is recommended that this procedure begin within a new command shell on a PLS Host.
- Set the DISPLAY environment variable. At the UNIX prompt on the PLS host (e.g. **odyssey**), type **setenv DISPLAY *terminal_id***, press **Return**.

2 Set the DCE environment variable. At the UNIX prompt on the PLS host (e.g. odyssey), type **dce_login *DCE_user_name DCE_password***, press **Return**.

3 Set the UNIX environment variable. At the UNIX prompt on the PLS host, change to the directory where the scripts are located (e.g. **cd** /usr/ecs/TS1/CUSTOM/utilities), then

- Type **setenv MODE *mode*** (e.g. TS1).
- Type **source *environment_setup_file*** (e.g. EcCoEnvCsh for C shell users).

4 At the UNIX prompt on the PLS host (e.g. odyssey), under the directory where the scripts are located (e.g. **cd** /usr/ecs/TS1/CUSTOM/utilities), type **EcPlPRE_IFStart** ***mode application_id*** **&**, then press **Return**.

- The ***mode*** is the operations mode (e.g. TS1).
- The ***application_id*** is a numerical number (e.g. 1).
- For example, type **EcPlPRE_IFStart** ***TS1 1*** **&**, press **Return**.
- Various messages from the Production Request Editor may appear in this window as it is running. For this reason, avoid using this window for other tasks until the Production Request Editor has terminated.

5 In the Production Request Editor, click on one of the tabs PR Edit, PR List, DPR View, or DPR List corresponding to desired task.

- To define a new Production Request or edit a Production Request, click on PR Edit. Proceed to Section Defining a New Production Request.
- To review or list a Production Request, click on PR List.
- To view or inspect a Data Processing Request, click on View.
- To review or inspect a Data Processing Request, click on List.

6 When tasks are completed in the Production Request Editor GUI, click on the **File** menu, then choose **Exit**.

- The Production Request Editor will disappear.
- Refer to Section (Troubleshooting and General Investigation) if fail to bring up the Production Request Editor GUI.

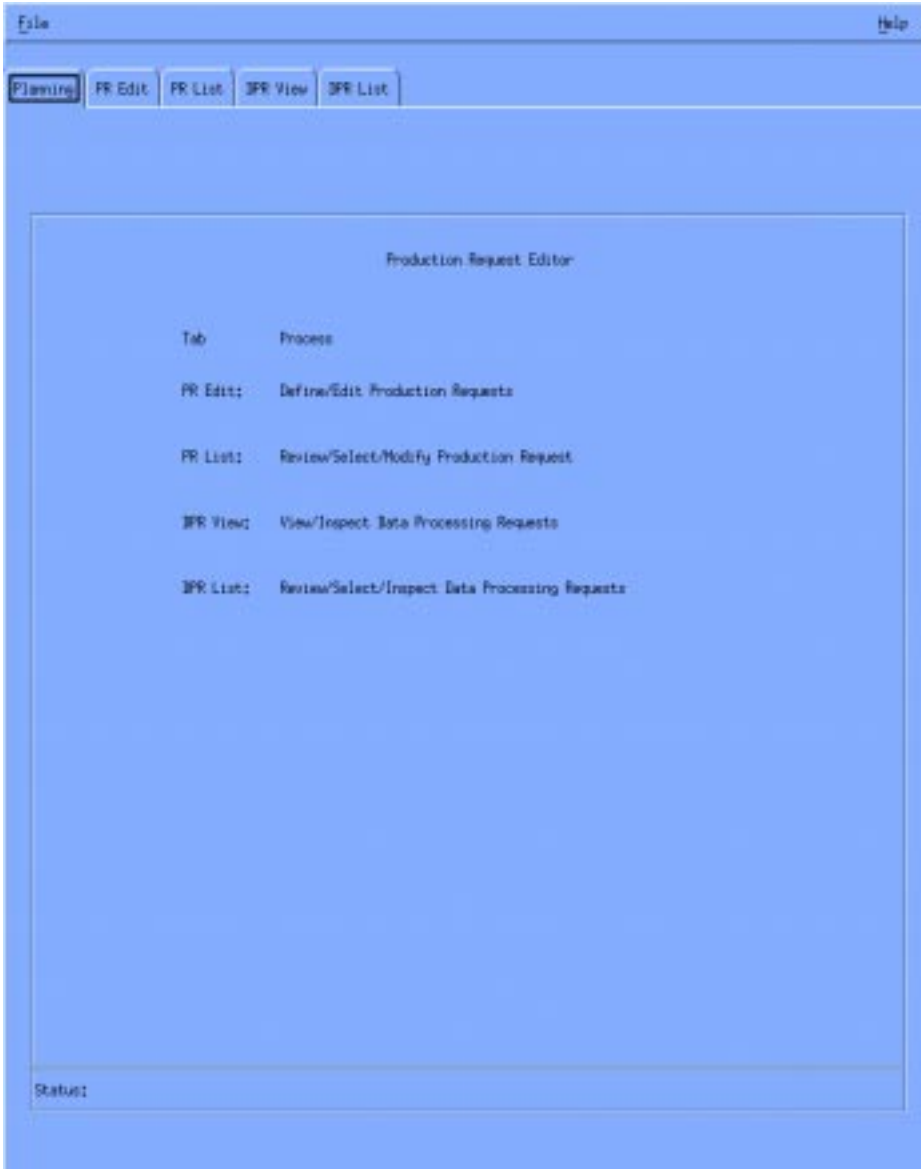


Figure 11.14.3-1. Production Request Editor Introductory GUI

11.14.3 Defining a New Production Request

A Production Request (PR) is a request for data production of granules between a start date/time and an end date/time. A PR will explode into one or more Data Processing Requests (DPR) depending upon the time interval involved. Each DPR corresponds to the execution of a PGE. Therefore, a PR results in the execution of a PGE one or more times. Only one PGE is involved in a single Production Request.

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

Assumptions:

1. The user has selected the **PR Edit** tab from the Production Request Editor .
2. The PGE involved in the Production Request has been registered in the PDPS database.

On workstation **x0pls##**, at the UNIX prompt in a terminal window, type as in step 1 below your user id and password.

NOTE: The **x** in the workstation name will be a letter designating your site:
g = GSFC, **m** = SMC, **l** = LaRC, **e** = EDC, **n** = NSIDC, **o** = ORNL, **a** = ASF, **j** = JPL; the ## will be an identifying two-digit number (e.g.,**g0pls02** indicates a Planning Subsystem (PLS) workstation at GSFC).

If you access the workstation through a remote login (rlogin), you must enter **xhost <remote_workstation_name>** prior to the rlogin, and enter **setenv DISPLAY <local_workstation IP address>:0.0** after the rlogin, before entering the command. The **<ipaddress>** is the ip address of **x0pls##,** and **xterm** is required when entering this command on a Sun terminal.

To define a new Production Request, execute the procedure steps that follow:

1 From the Production Request Editor GUI, click on the **PR Edit** tab.

- The PR Edit page will be displayed as shown in **Figure.** 11.14.3-2.

2 In the field labeled **PR Name:**, enter **New** or verify that **New** is already entered as the default.

3 The PGE for the Production Request must be selected from a list. To do this, click on the **PGE…** button.

- A GUI labeled **PGE Selection** will be displayed within which registered PGEs will be listed. The appropriate PGE can then be selected by clicking on it and then on the **OK** button.
- The selected PGE will then be used to populate **Satellite Name**, **Instrument Name**, **PGE Name**, and **PGE Version** fields of the main GUI.

4 In the field labeled **Priority:**, enter ***priority***.

- The ***priority*** is the priority to be assigned to this Production Request in the range 0 through 99 with 0 being the highest priority and 99 the lowest. For example, enter **40**.

5 In the Production Request Editor GUI, a **Duration** option is selected automatically based on the PGE registered into the PDPS. Two options are provided:

- **UTC Time** for time range.
- **Orbit** for orbit number range.

6 In the Production Request Editor GUI, enter ***StartDate*** and ***StartTime*** in fields labeled **Begin**, respectively.

- The ***StartDate*** and ***Starttime*** are the start date and time of the Production Request and should be entered in the mm/dd/yy and hh:mm:ss formats.

7 In the case of UTC Time duration, enter ***EndDate*** and ***EndTime*** in fields labeled **End**, respectively.

- The ***Enddate*** and ***EndTime*** are the end date and time of the Production Request and should be entered in the mm/dd/yy and hh:mm:ss formats.

8 In the case of Orbit duration, enter ***StartOrbit*** and ***EndOrbit*** in fields labeled **From** and **To**, respectively.

- The ***StartOrbit*** and ***EndOrbit*** are the orbit range of the Production Request.

9 Optionally, enter ***Comment*** in field labeled **Comment:**.

- This comment will be displayed whenever this Production Request is brought up and viewed.

10 When Production Request is complete, click on **File** menu and select **Save As…**.

- A GUI labeled **File Selection** will be displayed.
- In the field labeled **Selection**, enter a user-defined name to be assigned to the Production Request. Then click on the **OK** button. A message box will be displayed stating "Production Request Explosion into DPRs ok, *n* DPRs Generated", where *n* will be the number of DPRs (e.g. a 2-hr PR time will generate 24 DPRs for the 5-min processing period). Click on the **Ok** button. A second message box stating "Write to Database of Production Request ok"; again click **Ok**.

Note that you will not be allowed to enter a PR name that already exists. PR names that already exist will be displayed in the main window. The Production Request will then be saved under the name specified.

- Refer to Section (Troubleshooting and General Investigation) if fail to generate the DPRs.

11 When tasks are completed with the Production Request Editor GUI, click on the **File** menu, then choose **Exit**.

- The Production Request Editor GUI will disappear.

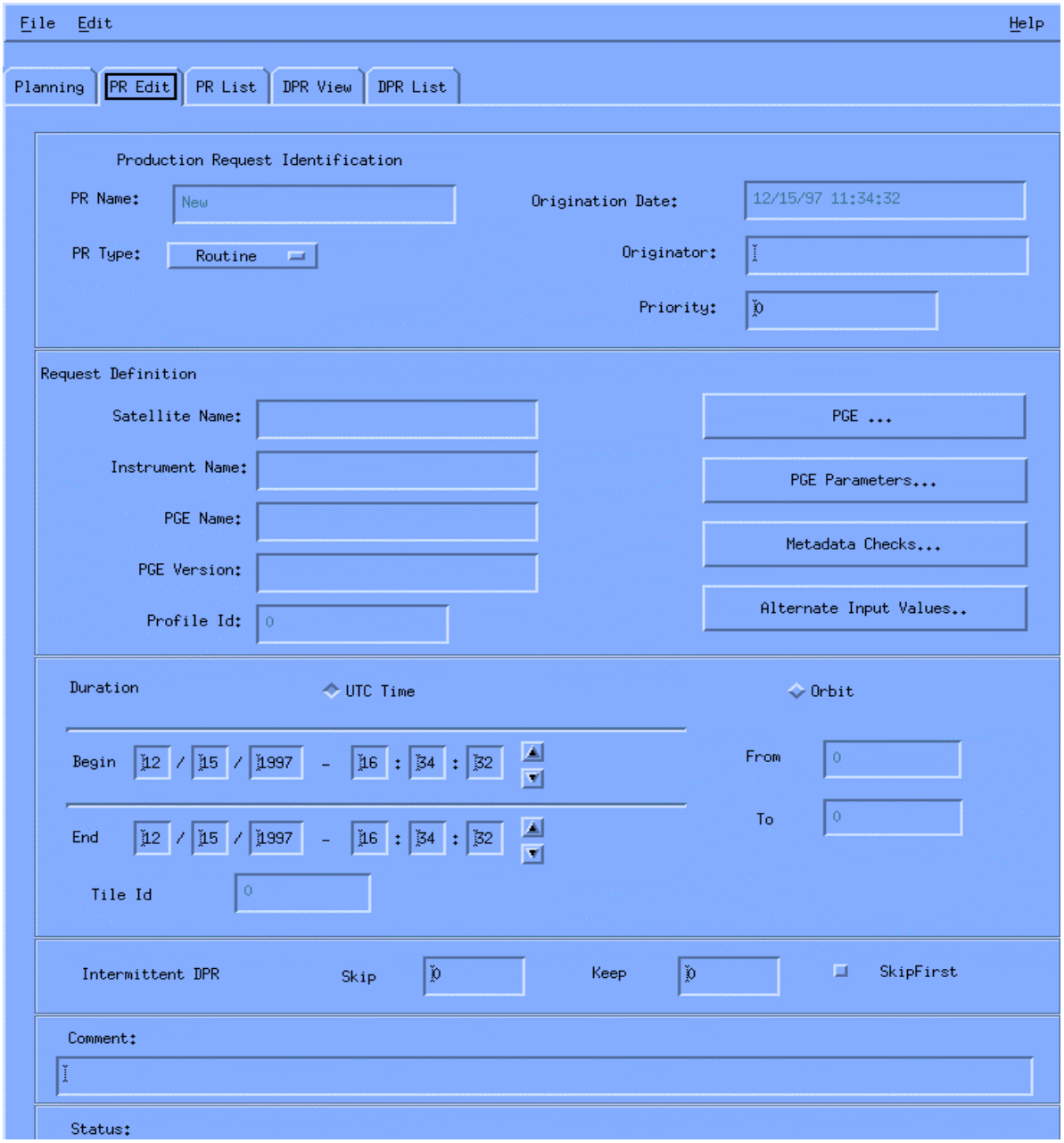


Figure 11.14.3-2. Production Request Editor GUI(Planning)

11.14.3 Processing

Once a candidate plan has been activated, each of the DPRs will result in subscriptions to the Data Server for the data needed. A request will go to the Data Server asking for notification when the required input data arrives.

Planning knows what data to request from the Data Server because the PDPS database stores this information as determined by the ESDT for each PGE. When the Data Server receives new data, it routinely checks to see if there are any outstanding subscriptions. If there are subscriptions, Planning will be notified. Once the input data required by a DPR becomes available, the DPR can be queued for processing.

Staging - The Data Processing Subsystem requests that the required input data, PGE (binary executables and shell scripts) and SDP Toolkit files be placed on a disk set aside for processing.

Process Control File (PCF) - establishes a linkage between logical Ids that the science software uses and the physical files that exist on the staging disk.

After the PGE has completed, the DPS will deallocate resources.

A Production History file will be created and will contain information concerning the conditions that the data products were generated by the PGE.

11.14.3.1 Viewing Production Requests

A Production Request (PR) is a request for data production of granules between a start date/time and an end date/time. A PR will explode into one or more Data Processing Requests (DPR) depending upon the time interval involved. Each DPR corresponds to the execution of a PGE. Therefore, a PR results in the execution of a PGE one or more times. Only one PGE is involved in a single Production Request.

This procedure describes how to view PRs that have already been defined . It assumes that the **PR List** tab has been selected from the Production Request Editor.

The information listed for each PR is:

- PR Name - The name assigned to the Production Request when it was defined.
- PGE ID - The name of the PGE involved in the PR.
- Priority - The priority (0 - 99) of the PR assigned when it was defined.
- Start - The start date and time of the PR.
- End - The end date and time of the PR.
- Comment - Any comment that was entered when the PR was defined.

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

Assumptions:

1. The user has selected the **PR List** tab from the Production Request Editor.

1 From the Production Request Editor GUI, click on the **PR List** tab.

- The PR List page will be displayed as shown in **Figure.** 11.14.3-3.

2 View the listed PRs. Optionally, find a PR by entering a search string in the field next to the **Find** button and then clicking on the **Find** button.

3 To modify a PR listed, click on the PR in the list and from the **File** menu select **Save As….**

- ↑ In the **File Selection** GUI, replace the current PR name shown in the **Selection** field with a new PR name. Then click on the **OK** button.
- When modifying an existing PR, it must be saved under a new PR name.
- ↑ Next, click on the **PR Edit** tab. The PR Edit page will be displayed with fields populated from the existing PR name, but having the new PR name chosen above.
- See Section on using the PR Edit page and saving any changes made

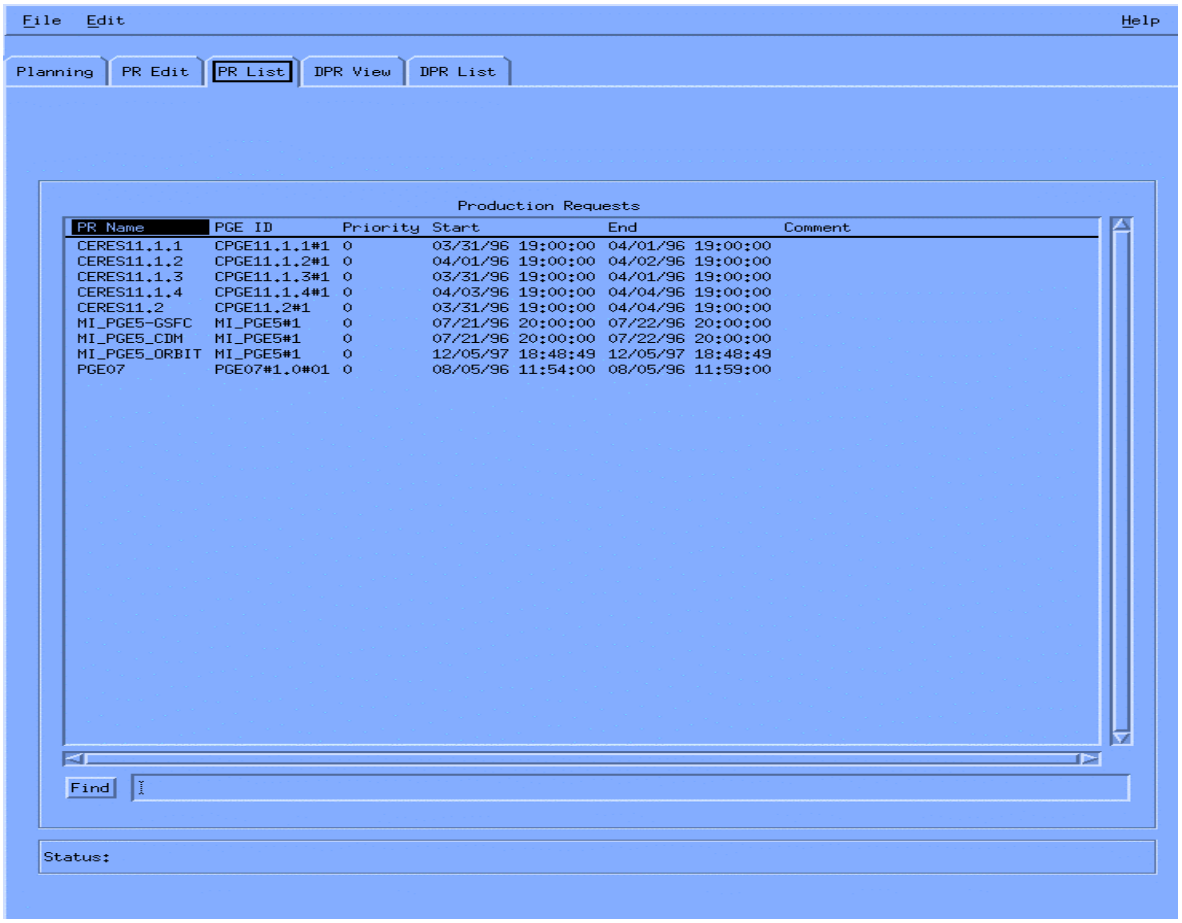


Figure 11.14.3-3. PR List GUI.

11.14.3.2 Viewing Data Processing Requests

Clicking on the DPR View GUI tab displays a list of all DPRs for all PRs entered into the system.

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

1 From the PR Editor GUI, click on **DPR View** tab. The following information will be displayed:

- Data Processing Request Identification.
- PGE ID and its parameters.
- Request Data and Status.

11.14.3.3 Listing Data Processing Requests

Selection of one PR on the PR List by highlighting it and then clicking on the DPR List tab, brings up a detailed display of all DPRs associated with the selected PR. These may be examined in order to develop production plans and schedule jobs.

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

1 From the PR Editor GUI, click on **DPR List** tab**.** The following information is displayed:

- DPR Id.
- PGE Id.
- PR Name.
- Data Start Time.
- Data Stop Time.

11.14.4 Using the Production Planning Workbench

The Production Planner uses the Production Planning Workbench to create new production plans and display a planning timeline.

11.14.4.1 Using the Planning Workbench to Run a PGE

Once a PGE has been fully registered, its test data files have been inserted to the Science Data Server, and a single Data Processing Request (DPR) has been generated, the Planning Workbench can be used to plan for one execution run of a single PGE.

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

Assumptions:

1. The required UNIX environment variables have been set properly.
2. The required servers of the ECS System are up and running.

3. A DPR has been generated successfully.

1 **Telnet to (PDPS) odyssey** or from the **SSIT Manager**, click on the **Tools** menu, then choose **Xterm**. Then telnet to a PLN Host.

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

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

3 *Login to DCE* **(dce_login awhitele awhitele***), **setenv DISPLAY ….:0.0***

- Change the working directory to the location where the script for starting the Autosys is stored. (e.g**. cd /usr/ecs/<mode>/CUSTOM/utilities).**
- Type**: source EcCoEnvCsh**

4 At the UNIX prompt on an PLN Host, Type **EcPlAllStart <mode> <application_id>**, press **Return**.

- The ***mode*** is one of modes used in the ECS system. e.g. TS1.
- The ***application_id*** is a numerical number. e.g. 1.
- For example: **EcPlAllStart TS1 1, Return.**
- A Planning Workbench GUI will be appeared as shown in **Figure 11.14.4-1.**

5 In the Planning Workbench GUI, go to the subwindow labeled **Unscheduled** and click on a Production Request name.

- The Production Request name is the name under which the PR was saved.
- The PR name entry will be highlighted.

6 In the Planning Workbench GUI, click on the button next to the label **Schedule** (the button has an inverted triangle on it).

- The PR highlighted in step 3 will appear in the subwindow labeled **Scheduled**.

7 In the Planning Workbench GUI, click on the **Activate** button

- A small GUI labeled **Plan Activation** will be displayed.

8 In the Plan Activation GUI, set the time in the time field forward to allow ample time for the PGE to run. Then click on the **Ok** button.

- ↑ All that is necessary is for there to be sufficient time for the PGE run. There is no penalty for allowing *too* much time.
- ↑ The Production Request thus planned will be submitted to processing and its progress can be monitored with AutoSys.

10 When tasks are completed with the Planning Workbench GUI, click on the **File** menu, then choose **Exit**.

- The Planning Workbench GUI will disappear.

Important Note **:** The creation of Production Requests and Plans requires close coordination among all who are using the same mode in SSIT. Otherwise, the submittal of a Plan may prevent a waiting DPR from starting. In particular, when submitting a new Plan in a mode being shared with others, one should:

1. Ask everyone else if they have a DPR awaiting data as part of a chain.
2. Check the PDPS database table PlDataProcessingRequest for any DPRs that are in state CQ_HOLD and include these in the new Production Request and Plan.
3. Also, include any DPRs (except those marked SUCCESS) upon which the given DPR depends in the new Production Request and Plan.

11.14.4.2 Creating and Activating a Production Plan

The Production Planner creates a plan for production data processing at the DAAC by selecting specific PRs whose DPRs are to be run. The planning tool provides a forecast of the start and completion times of the jobs based upon historical experience in running these PGEs. Through the planning tool, when the generated plan is "activated," the information included in the plan is transferred to the Data Processing subsystem and loaded into the Platinum AutoSys tool where production processing is managed.

The Production Planner creates the plan by selecting PRs from two lists of PRs, i.e., the list of available "Unscheduled" PRs and the list of "Scheduled" PRs. Using arrow buttons, the Production Planner moves the PRs between lists until the "Scheduled" list contains the desired set of PRs that define the new plan. Only one user can use the **Planning Work Bench** at a time. It is recommended for SSI&T that only one person do the planning for the group.

Before creating a new production plan the Production Planner must have available the following information:

Name of the plan.
Comments (if any).
PRs to be included in the new production plan.

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

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

If you access the workstation through a remote login (**rlogin**), you must enter **xhost <remote_workstation_name>** then press the **Enter** key.

Once you have entered xhost <remote_workstation_name>, but prior to the remote login, enter **setenv DISPLAY** <local workstation IP address>**:0.0** where the local workstation IP address represents the IP address you where you are located**.**

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

3 At a UNIX prompt type cd to the directory where the scripts are located. (e.g.

/usr/ecs/TS1/CUSTOM/utilities).

4 At a UNIX prompt type **setenv DISPLAY *hostname*:0.0**

5 At a UNIX prompt on the PLN host (e.g. odyssey), type **EcPlPRE_IFStart mode 3** a **Planning Workbench** GUI is displayed (Figure 11.14.4-1)

Figure 11.14.4-1. Planning Workbench GUI

Data concerning the currently active production plan are displayed.

If you want to "kill" (deactivate) the currently active production plan without activating a replacement, click on the **Kill** button.

Whenever you activate a plan (by clicking on the **Activate** button), you automatically "kill" the currently active plan.

6 Select **File → New** from the pull-down menu.
The "New" window appears.

7 Type a name for the new plan, then press the **Tab** key on the keyboard.

The **Planning Workbench** GUI is displayed.
The **Plan Name** is displayed.
The **Status** displayed is **Candidate**.

8 Type the desired date (in ***MM/DD/YY*** format), then press the **Tab** key on the keyboard to advance to the next field.

9 Type the desired time (in ***hh:mm*** format), then press the **Tab** key on the keyboard.
The **Rollover Time** is displayed.

10 Type any relevant comments (up to 255 characters) in the **Comments** field.

11 Move PRs between the **Unscheduled** and **Scheduled** lists as necessary by selecting (highlighting) the PR to be moved by clicking on the PR in the list from which it is to be moved then clicking on the up or down arrow button (as applicable) to move the PR to the other list.
Highlighted PR disappears from one list and appears on the other.

The unscheduled and scheduled PR lists are scrollable.

12 When the **Scheduled** list accurately reflects the PRs to be scheduled in the production plan, select **File → Save** (or **File → Save As**) from the pull-down menu to save the new production plan.

The new production plan is saved.

13 If the new plan is to be activated immediately, click on the **Activate** button to activate the new plan.

The currently active plan is killed (deactivated) and the new plan is activated.

- The **Production Planning Timeline** GUI is displayed.

14 If the new production plan is to be used as a baseline plan, click on the **Baseline** button.

The "New" window appears.

The plan is recorded as well as the time of baselining so that it can be used in comparing future processing results with planned objectives.

15 If the production plan being displayed is active and should be deactivated, click on the **Kill** button.

The "New" window appears.

The plan is deactivated without activating another plan.

The progress of one or more PGEs running within the PDPS may be monitored. The COTS tool used for this purpose is AutoSys® by Atria Software. Each Data Processing Request results in seven AutoSys jobs that are boxed together. An AutoSys job name follows the template:

PGEname#Suffix

where *PGEname* is replaced by the name of the PGE, and *Suffix* is a character indicating the job phase of the DPR

For example, for a scheduled PGE named MOPITT4, the AutoSys jobs making up that DPR would be:

MOPITT4#A
MOPITT4#S
MOPITT4#P
MOPITT4#E
MOPITT4#p
MOPITT4#I
MOPITT4#D

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

Assumptions:

1. The required servers of the ECS System are up and running.
2. A DPR has been scheduled successfully.

To monitor production, execute the procedure steps that follow:

1 In any currently available xterm window, spawn a new session: type **xterm &**, press **Return**. Then telnet to the DPS host.

- It is recommended that this procedure begin within a new command shell on a DPS Host.
- Set the DISPLAY environment variable. At the UNIX prompt on the DPS host (e.g. illiad), type **setenv DISPLAY *terminal_id***, press **Return**.

2 Set the DCE environment variable. At the UNIX prompt on the DPS host (e.g. illiad), type **dce_login *DCE_user_name DCE_password***, press **Return**.

3 Set the UNIX environment variable. At the UNIX prompt on the DPS host, change to the directory where the scripts are located (e.g. **cd /usr/ecs/TS1/CUSTOM/utilities**), then

- Type **setenv MODE *mode*** (e.g. TS1).
- Type **source *environment_setup_file*** (e.g., **EcCoEnvCsh** for C shell users).

4 At the UNIX prompt on the DPS Host, under the directory where the scripts are located (e.g. **cd /usr/ecs/TS1/CUSTOM/utilities**), type **EcDpPrAutosysStart *mode application_id* &**, then press **Return**.

- The ***mode*** is one of modes used in the ECS system (e.g., TS1).
- The ***application_id*** is a numerical number (e.g., 1).
- For example: **EcDpPrAutosysStart TS1 1, Return.**
- A GUI labeled **AutoSys** will be displayed.
- This GUI will contain eight buttons for invoking various tools available under AutoSys.

5 In the AutoSys GUI, click on the **Ops Console** button.

- A GUI labeled **AutoSys Job Activity Console** GUI will be displayed.
- The main subwindow of this GUI will contain a dynamically updated list of AutoSys jobs (seven jobs make up one DPR) currently scheduled.
- To disable dynamic updating of the main subwindow (which may be distracting), click on the **View** menu, then choose **Select Jobs**. A GUI labeled **Job Selection** will be displayed. Under the label **Select by Name**, click on the square labeled **All jobs**, then click on the **OK** button.
- DPRs will be listed in the column labeled **Job Name** and their statuses (e.g. SUCCESS) will be listed in the column labeled **Status**.
- To view job status information for a particular DPR, click on a DPR. Below the main subwindow, available job status information will be displayed.
- To view the existing event report, under the label **Reports**, click on the middle diamond labeled **Event**. The current event status for the selected DPR will be displayed in the subwindow labeled **Event Report**. Alternatively, the summary report for the selected DPR will be displayed by clicking on the middle diamond labeled **Summary**.
- To view the job definition, under the label **Show**, click on the **Job Definition** tab. A GUI labeled **Job Definition** will be displayed. The selected DPR is shown in **Job Name**. To kill the DPR, click on the **Delete** tab. ***(Warning: Be very careful to avoid deleting wrong DPR!)***
- Exit the **AutoSys Job Activity Console** by clicking on the **Exit** button.

6 In the AutoSys GUI, click on the **JobScape** button.

- A GUI labeled **JobScape** will be displayed.

- The main subwindow of this GUI will contain a dynamically updated list of AutoSys jobs (seven jobs make up one DPR) currently scheduled. The colors indicate the job statuses for DPRs and their jobs at present.
- There are eleven job statuses: **ACTIVATED, STARTING, RUNNING, SUCCESS, FAILURE, TERMINATED, RESTART, QUE_WANT, ON_ICE, OFF_HOLD,** and **INACTIVE**. The color chart is on the left of GUI.
- To view job status information for a particular DPR (or job) in detail, click on a DPR (or job), then click on the **Job Console** button. A GUI labeled **Job Console** will be displayed. The information shown here is similar to that shown in **Ops Console** as mentioned above.
- To change the status of a job for a particular DPR, click on a job under that DPR, then press the right button of the mouse. Choose one of the functions in the lower part of the menu. For example, to hold a job, choose **On Hold**, then click the **Yes** button.
- ***Suggestion:*** For a scheduled DPR, hold all jobs for that DPR except the first one (Allocation). After the first job runs successfully, release the next job (Staging) by choosing **Off Hold**. Repeat until all jobs are done. Therefore, if a job fails, it can be fixed and rerun without interfering with the others.
- To exit the **JobScape**, click on the **File** menu, then choose **Exit**.

7 In the AutoSys GUI, click on the **TimeScape** button.

- A GUI labeled **TimeScape** GUI will be displayed.
- The main subwindow labeled **Job Name** of this GUI will contain a dynamically updated list of AutoSys jobs (seven per DPR) currently scheduled.
- To disable dynamic updating of AutoSys jobs (which may be distracting), click on the **Freeze Frame** button.
- The color of each job indicates its status according to the legend on the left side of the GUI.
- The time line is shown on the right side of the GUI with time marked at the top. A red vertical line (dashed) indicates the current time.
- Exit the **TimeScape** GUI by clicking on the **File** menu and selecting **Exit**.

8 To quit AutoSys, in the AutoSys GUI, click on the **Exit** button.

- The AutoSys GUI will disappear.
- Refer to Section (Troubleshooting and General Investigation) if any job fails on AutoSys.

On workstation **x0pls##**, at the UNIX prompt in a terminal window, enter steps as in step 1 below.

NOTE: The **x** in the workstation name will be a letter designating your site: **g** = GSFC, **m** = SMC, **l** = LaRC, **e** = EDC, **n** = NSIDC, **o** = ORNL, **a** = ASF, **j** = JPL; the ## will be an identifying two-digit number (e.g.,**g0pls01** indicates a Planning and Data Processing subsystem(PLS) workstation at GSFC).

If you access the workstation through a remote login (rlogin), you must enter **xhost <remote_workstation_name>** prior to the rlogin, and enter **setenv DISPLAY**

<local_workstation IP address>:0.0 after the rlogin, before entering the command. The **<ipaddress>** is the ip address of **x0pls##,** and **xterm** is required when entering this command on a Sun terminal.

11.14.5 Monitoring Production in PDPS Subsystem

Example of monitoring production usisng above procedures:

1 **telnet to (PDPS) taltos** or from the SSIT Manager, click on the **Tools** menu, then choose **Xterm**. Then telnet to a PLN Host.

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

3 *Login to DCE* **(dce_login awhitele awhitele***), **setenv DISPLAY ….:0.0***

- Change the working directory to the location where the script for starting the Autosys is stored. (e.g**. cd /usr/ecs/<mode>/CUSTOM/utilities).**
- **setenv MODE <MODE>**
- **source EcCoEnvCsh**

4 At the UNIX prompt on the PLN Host, type: **EcDpPrAutosysStart <MODE> <Autosys Instance>, example: TS1 1, &**, press **Return**.

- A GUI labeled **AutoSys** will be displayed.
- This GUI will contain eight buttons for invoking various tools available under AutoSys.

5 In the AutoSys GUI, click on the **Ops Console** button.

- The AutoSys GUI will be displayed.

6 Select the desired display and view the contents.

- Selections include:
  - Autosys Job Activity Ops Console
  - HostScape
  - TimeScape
  - JobScape
- The main subwindow of this GUI will contain a dynamically updated list of AutoSys jobs (seven jobs make up one DPR) currently scheduled.
- To disable dynamic updating of the main subwindow (which may be distracting), click on **Freeze Frame** in the small subwindow labeled **Show**.
- DPRs will be listed in the column labeled **Job Name** and their statuses (SUCCESS, FAILURE, TERMINATED) will be listed in the column labeled **Status**.
- To view job status information for a particular DPR, click on a DPR. Below the main subwindow, available job status information will be displayed.
- To view the existing event report, under the label **Reports**, click on the middle diamond labeled **Event**. The current event status for the selected DPR will be displayed in the subwindow labeled **Event Report**.
- Exit the **AutoSys Job Activity Console** by clicking on the **Exit** button.

Suggestion: For a scheduled DPR, hold all jobs for that DPR except the first one (Allocation). After the first job runs successfully, release the next job (Staging) by choosing Off Hold. Repeat until all jobs are done. Therefore, if a job fails, it can be fixed and rerun without interfering with the others.

This is especially important for the Postprocessing job. If the PGE fails and the Postprocessing job is not **On Hold,** the DPR will have to be deleted and a new one created. This happens because PDPS will have deleted all references to the output granules in the PDPS database

7 In the AutoSys GUI, click on the **TimeScape** button.

- A GUI labeled **TimeScape** GUI will be displayed.
- The main subwindow labeled **Job Name** of this GUI will contain a dynamically updated list of AutoSys jobs (seven per DPR) currently scheduled.
- To disable dynamic updating of AutoSys jobs (which may be distracting), click on the **Freeze Frame** button.
- The color of each job indicates its status according to the legend on the left side of the GUI.
- The time line is shown on the right side of the GUI with time marked at the top. A red vertical line (dashed) indicates the current time.
- Exit the **TimeScape** GUI by clicking on the **File** menu and selecting **Exit**.

8 To quit AutoSys, in the AutoSys GUI, click on the **Exit** button.

- The AutoSys GUI will disappear.

9 Use the "**tail -f em.log**" to create a permanent record of a log file if debugging is necessary. Note: the xx.log file has to be created first: "**vi em.log**"

10 To look at the Data Base type: **setenv MODE <MODE>, cd utilities**

source EcCoEnvCsh, cd dbr, dbrowser-syb <MODE> 2 &

11 Scrips to restart the servers and subsystems are located in: **/home/cmts1/restart/drop4/**

11.14.6 Using the Q/A Monitor

The Q/A Monitor allows the output products produced during a PGE run to be accessed and examined. Input test data granules and Production History files can be retrieved in the same manner. The Q/A Monitor retrieves output products based on the collection name (*i.e.* the ESDT) and time of Insertion to the Science Data Server.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow. The Q/A Monitor is discussed in more detail in Section 15.

Assumptions:

1. The required UNIX environment variables have been set properly.
2. The desired output products have been successfully Inserted to the Science Data Server.

To use the Q/A Monitor, execute the procedure steps that follow:

1 **telnet to (PDPS) taltos** or from the SSIT Manager, click on the **Tools** menu, then choose **Xterm**. Then telnet to a PLN Host.

- Alternatively, in any currently available xterm window, spawn a new session: type **xterm &**, press **Return**. Then telnet to the PLN Host.
- Set the DISPLAY environment variable: **setenv DISPLAY terminal_id, press Return.**
- Type **setenv <mode> (**e.g..,**TS1).**
- Type **source** environment_setup_file **(**e.g**., EcCoEnvCsh** for C shell users**).**

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

3 *Login to DCE* **(dce_login awhitele awhitele***), **setenv DISPLAY ….:0.0***

- Change the working directory to the location where the script for starting the Autosys is stored. (e.g**. cd /usr/ecs/<mode>/CUSTOM/utilities).**

4 At the UNIX prompt on the PLN Host, type: **EcDpPrQaMonitorGUIStart <MODE> <Q/A Monitor Instance>, example: TS1 1, &**, press **Return**.

- The ***mode*** is the operations mode.
- The Q/A Monitor GUI will be displayed.
- Various messages from the Q/A Monitor will appear in this window as it is running.

5 In the Q/A Monitor subwindow labeled **Data Types,** select an ESDT from the list presented and click on it.

- Use the scroll bars if necessary to locate desired ESDT.
- Optionally, use the **Find** field and button to locate an ESDT.

6 In the Q/A Monitor, under the label **Data Granule Insert Date (mm/dd/yy)**, set the date range within which the search for granules of the ESDT selected will be conducted.

- The date range can be made arbitrarily large to select all granules of a particular collection (ESDT).
- The dates refer to date of granule Insert to the Science Data Server.

7 In the Q/A Monitor, click on the **Query** button.

- The results of the query will be displayed in the bottom window, labeled **Data Granules**.
- All granules having the ESDT selected in step 3 and having Insert times within the date range specified in step 4 will be listed in this window.

8 In the Q/A Monitor subwindow labeled **Data Granules**, click on one of the data granules listed to be examined.

- The data granule selected will be highlighted.

9 To retrieve the data granule's Production History file, click on the **Retrieve Prod History** button.

- The Production History (PH) tar file corresponding to the selected data granule will be retrieved from the Science Data Server and placed on the local machine (a PLN Host) in the directory /var/tmp.
- The PH can then be moved or copied manually from the /var/tmp directory to a user working directory for examination.
- Only the PH file is retrieved with the **Retrieve Prod History** button.
- If only the PH is desired, exit this procedure. To retrieve the data granule itself, continue on to step 8.

10 To retrieve the data granule, click on the **Retrieve Data Granule** button and note its file name (listed in the entry for the granule; you may have to scroll over to the right to see it).

- The data granule selected will be retrieved from the Science Data Server and placed on the local machine (a PLS Host) in the directory /var/tmp.
- A granule of any format (binary, ASCII, HDF, HDF-EOS) may be retrieved in this manner. Only HDF and HDF-EOS granules, however, may be further visualized using EOSView as described in the next steps.

11 To examine the data granule, click on the **Visualize data** tab.

- The **Visualize data** page will be displayed.

---

12 In the main subwindow on the **Visualize data** page, locate the file name of the granule retrieved in step 8 and click on it.

- The item selected will be highlighted and will appear in the **Selection** subwindow below.

13 Click on the **Visualize** button.

- This action will invoke **EOSView** with the granule selected.
- The granule must be HDF or HDF-EOS format.

14 When tasks are completed with the Q/A Monitor GUI, click on the **File** menu, then choose **Exit**.

11.15 Postprocessing and General Investigation

An important part of SSI&T is verifying that the output files produced at the DAAC are identical (within particular tolerances) to the test output files delivered with the DAPs. A successful comparison is a strong indication that the porting of the science software from the development facility at the SCF to the operational facility at the DAAC has not introduced any errors.

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

It is assumed that the Instrument Team has delivered test output files (produced at their SCF) with which to perform the comparison.

11.15.1 Examining PGE Log Files

Three log files are produced by PGEs during runtime: the Status log, User Log, and the Report log. These log files are written by the SDP Toolkit and by the science software using the Toolkit's Status Message Facility (SMF). The location of these log files is specified in the Process Control File (PCF). When the PGE is built and run with the SCF version of the SDP Toolkit, the location and file names of the log files can be set as desired. When the PGE is built with the DAAC version of the SDP Toolkit and run within the PDPS, the location and file names of the log files is set by the system in the instantiated PCF.

The Status log file captures all error and status information. The User log file captures a subset of messages which are more informational. The Report log file captures arbitrary message strings sent by the PGE.

The section aforementioned describes how to examine log files produced by PGEs that have been built with the SCF version of the SDP Toolkit and run from the command line.

The section aforementioned describes how to examine log files (within the Production History) produced by PGEs that have been built with the DAAC version of the SDP Toolkit and run within the PDPS.

11.15.1.1 Log Files From PGEs Run Outside of the PDPS

When the PGE is run outside of the PDPS, the PCF specifies the location and file names of the log files produced. This procedure describes how to locate that information from the PCF and use it to examine the log files.

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

Assumptions:

1. The PGE has been successfully built with the SCF version of the SDP Toolkit.
2. The PGE's PCF has been updated properly for the DAAC environment .

1 At the UNIX prompt on an AIT Sun or on the SPR SGI, type **cd *PCFpathname***, press **Return**.

- The ***PCFpathname*** is the full path name to the location of the PCF used by the PGE for which log files are to be examined.

2 At the UNIX prompt on the AIT Sun or on the SPR SGI, type **vi *PCFfilename***, press **Return**.

- The ***PCFfilename*** is the file name of the PCF used by the PGE for which log files are to be examined.
- This brings up the file named ***PCFfilename*** in the *vi* editor.
- Any text editor may be used such as *emacs*. For example, **emacs MOD35.pcf**, press **Return**.

3 In the editor, search for logical IDs (beginning in the first column) **10100**, **10101**, and **10102**. These are the PCF entries for the LogStatus, LogReport, and LogUser respectively. For each, note the file names in field 2 and the path names in field 3. Then quit the editor.

- If field 3 is blank, then the location is given by the default location specified in a line above the entries beginning with the "!" character.

4 At the UNIX prompt on the SPR SGI, type **vi *StatusLogPathname/filename***, press **Return**.

- The ***StatusLogPathname/filename*** is the full path name and file name of the Status log file noted in step 3 associated with PCF logical ID 10100. When finished, quit the editor.
- Note any error or warning messages in file.

- Any text editor may be used such as *emacs*. For example, **emacs /PGE/MOD35/LogStatus**, press **Return**.

5 At the UNIX prompt on the SPR SGI, type **vi *UserLogPathname/filename***, press **Return**.

- The ***UserLogPathname/filename*** is the full path name and file name of the Status log file noted in step 3 associated with PCF logical ID 10101. When finished, quit the editor.
- Note any error or warning messages in file.
- Any text editor may be used such as *emacs*. For example, **emacs /PGE/MOD35/LogUser**, press **Return**.

6 At the UNIX prompt on the SPR SGI, type **vi *ReportLogPathname/filename***, press **Return**.

- The ***ReportLogPathname/filename*** is the full path name and file name of the Status log file noted in step 3 associated with PCF logical ID 10102. When finished, quit the editor.
- Note any anomalous messages in file.
- Any text editor may be used such as *emacs*. For example, **/PGE/MOD35/LogReport**, press **Return**.

11.15.1.2 Production History Log Files From PGEs Run Within the PDPS

The Production History (PH) is created during PGE execution within the PDPS and then Inserted into the Data Server upon PGE completion. The PH is a UNIX tar file that includes the PGE log files.

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

Assumptions:

1. The PDPS archive configuration area is properly set up.
2. The environment variable DataServer contains the full path name to the archive. This is typically /imf/archive/ or /imf_data/archive/ and varies at each DAAC.

1 At the UNIX prompt on the SPR SGI, type **cd $DataServer/PH**, press **Return**.

- The **$DataServer** is an environment variable containing the full path name of the Data Server archive and **PH** is a subdirectory under **$DataServer** containing the Production History tar files.
- For example, type **cd $DataServer/PH**, press **Return**.

2 At the UNIX prompt on the AIT Sun or on the SPR SGI, type **ls -al**, press **Return**.

- A list of the current contents will be displayed. These will be Production History tar files.
- The file names of PH files are named *PGEname#versionMMDDYYhhmm*_runtime.tar_*UR* where *PGEname* is replaced by the PGE name, *version* is replaced by the PGE version, *MMDDYY* is replaced by the Insert date in the month-day-year format, *hhmm* is replaced by the Insert time in the hours-minutes format, and *UR* is replaced by the Universal Reference.

For example, the PH file name for Release 4.1 of a SAGE III PGE named sage_1t Inserted on December 14, 1999 at 2:00pm could be: sage_1t#2.11214991400_runtime.tar_YAAa005Li_19991214135815.

- Look for the PH of interest.

3 At a UNIX prompt on the AIT Sun or on the SPR SGI, type **cp *PHtarFilename WorkingPathname***, press **Return**.

- The ***PHtarFilename*** is the file name of the Production History tar file.
- The ***WorkingPathname*** is the full path name to some working directory in which the Production History tar file is to be placed and examined.

4 At the UNIX prompt on the AIT Sun or on the SPR SGI, type **cd *WorkingPathname***, press **Return**.

- The ***WorkingPathname*** is the full path name to the working directory specified in step 3.

5 At the UNIX prompt on the AIT Sun or on the SPR SGI, type **tar xvf *PHtarFilename***, press **Return**.

- The ***PHtarFilename*** is the file name of the Production History tar file in the working directory.
- This command will untar the Production History tar file, extracting its component files into the current directory.

6 At the UNIX prompt on the AIT Sun or on the SPR SGI, type **vi *StatusLogFilename***, press **Return**.

- The ***StatusLogFilename*** is the file name of the Status log file within the PH. When finished, quit the editor.
- Note any error or warning messages in file.
- Any text editor may be used such as *emacs*. For example, **emacs LogStatus**, press **Return**.

7 At the UNIX prompt on the AIT Sun or on the SPR SGI, type **vi *UserLogFilename***, press **Return**.

- The ***UserLogFilename*** is the file name of the User log file within the PH. When finished, quit the editor.
- Note any error or warning messages in file.
- Any text editor may be used such as *emacs*. For example, **emacs LogUser**, press **Return**.

8 At the UNIX prompt on the AIT Sun or on the SPR SGI, type **vi *ReportLogFilename***, press **Return**.

- The ***ReportLogFilename*** is the file name of the Report log file within the PH. When finished, quit the editor.
- Note any error or warning messages in file.
- Any text editor may be used such as *emacs*. For example, **emacs LogReport**, press **Return**.

11.15.1.3 History Log Files From Failed PGEs Run Within the PDPS

The History Log(HL) is created during PGE execution within the PDPS and then Inserted into the Data Server upon failure of the PGE. The HL is a UNIX file that includes the PGE log files, the PCF and PH.

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

Assumptions:

1. The PDPS archive configuration area is properly set up.
2. The environment variable DataServer contains the full path name to the archive.

1 At the UNIX prompt on the SPR SGI, type **cd $DataServer/FAILPGE**, press **Return**.

- The **$DataServer** is an environment variable containing the full path name of the IMF Data Server archive and **FAILPGE** is a subdirectory under **$DataServer** containing the History Log files.
- For example, type **cd $DataServer/FAILPGE**, press **Return**.

2 At the UNIX prompt on the SPR SGI, type **ls -al**, press **Return**.

- A list of the current contents will be displayed. These will be History Log files.
- The file names of HL files are named *PGEname#versionMMDDYYhhmm*_runtime.tar_*UR* where *PGEname* is replaced by the PGE name, *version* is replaced by the PGE version, *MMDDYY* is replaced by the Insert date in the month-day-year format, *hhmm* is replaced by the Insert time in the hours-minutes format, and *UR* is replaced by the Universal Reference. For example, the HL file name for Release 4.1 of a SAGE III PGE named sage_1t Inserted on December 14, 1999 at 2:00pm could be: sage_1t#2.11214991400_runtime.tar_YAAa005Li_19991214135815.
- Look for the HL of interest.

3 At a UNIX prompt on the SPR SGI, type **cp *HLFilename* *WorkingPathname***, press **Return**.

- The ***HLFilename*** is the file name of the History Log file.
- The ***WorkingPathname*** is the full path name to some working directory in which the History Log file is to be placed and examined.

4 At the UNIX prompt on the SPR SGI, type **cd *WorkingPathname***, press **Return**.

- The ***WorkingPathname*** is the full path name to the working directory specified in step 3.

5 At the UNIX prompt on the SPR SGI, type **vi *HLFilename***, press **Return**.

- The ***HLFilename*** is the file name of the History Log. When finished, quit the editor.
- Note any error or warning messages in file.
- Any text editor may be used such as *emacs*. For example, **emacs *HLFilename***, press **Return**.

11.15.2 The Production History

The Production History (PH) is a UNIX tar file that is produced and archived for every run of a PGE in the PDPS. Each PH can be uniquely retrieved from the Data Server.

PH files are located in the $DataServer/PH directory and have the following naming convention:

*PGEname#versionMMDDYYhhmm*_runtime.tar_*UR*

where *PGEname* is replaced by the PGE name, *version* is replaced by the PGE version, *MMDDYY* is replaced by the Insert date in the month-day-year format, *hhmm* is replaced by the Insert time in the hours-minutes format, and *UR* is replaced by the Universal Reference.

For example, a PH file name for Release 4.1 of a SAGE III PGE named sage_1t Inserted on December 14, 1999 at 2:00pm would be:

sage_1t#2.11214991400_runtime.tar_YAAa005Li_19991214135815.

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

Assumptions:

1. The PDPS archive configuration area is properly set up.
2. The environment variable DataServer contains the full path name to the archive.

1 At the UNIX prompt on an AIT Sun or on the SPR SGI, type **cd $DataServer/PH**, press **Return**.

- The **$DataServer** is an environment variable containing the full path name of the IMF Data Server archive and **PH** is a subdirectory under **$DataServer** containing the Production History tar files.
- For example, type **cd $DataServer/PH**, press **Return**.

2 At the UNIX prompt on the AIT Sun or on the SPR SGI, type **ls -al**, press **Return**.

- A list of the current contents will be displayed. These will be Production History tar files.
- The file names of PH files are named *PGEname#versionMMDDYYhhmm*_runtime.tar_*UR* where *PGEname* is replaced by the PGE name, *version* is replaced by the PGE version, *MMDDYY* is replaced by the Insert date in the month-day-year format, *hhmm* is replaced by the Insert time in the hours-minutes format, and *UR* is replaced by the Universal Reference. For example, the PH file name for Release 4.1 of a SAGE III PGE named sage_1t Inserted on December 14, 1999 at 2:00pm could be: sage_1t#2.11214991400_runtime.tar_YAAa005Li_19991214135815.
- Look for the PH of interest.

3 At a UNIX prompt on the AIT Sun or on the SPR SGI, type **cp *PHtarFilename WorkingPathname***, press **Return**.

- The ***PHtarFilename*** is the file name of the Production History tar file.
- The ***WorkingPathname*** is the full path name to some working directory in which the Production History tar file is to be placed and examined.

4 At the UNIX prompt on the AIT Sun or on the SPR SGI, type **cd *WorkingPathname***, press **Return**.

- The ***WorkingPathname*** is the full path name to the working directory specified in step 3.

5 At the UNIX prompt on the AIT Sun or on the SPR SGI, type **tar xvf *PHtarFilename***, press **Return**.

- The ***PHtarFilename*** is the file name of the Production History tar file in the working directory.
- This command will untar the Production History tar file, extracting its component files into the current directory.

6 At the UNIX prompt on the AIT Sun or on the SPR SGI, type **vi *PHcomponentFile***, press **Return**.

- The ***PHcomponentFile*** is the file name of one of the PH component files. The component files are:
  - *PGEname#versionMMDDYYhhmm*.Log - Contains the DPR ID, the actual command used to run the PGE, resource usage information, the PGE exit status, and files used by the PGE.
  - *PGEname#versionMMDDYYhhmm*.Pcf - The actual instantiated PCF used when running the PGE.
  - *PGEname#versionMMDDYYhhmm*.ProdLog - Contains the DPR ID, the PGE ID, and resource usage information (same as in the .Log file).
  - *PGEname#versionMMDDYYhhmm*.Profile - Contains the environment variables defined during the execution of the PGE including the contents of the PATH environment variable.
  - *PGEname#versionMMDDYYhhmm*.TkReport - The Report log file, same as is produced when run outside of the PDPS.
  - *PGEname#versionMMDDYYhhmm*.TkStatus - The Status log file, same as is produced when run outside of the PDPS.
  - *PGEname#versionMMDDYYhhmm*.TkUser - The User log file, same as is produced when run outside of the PDPS.
  - *ESDTmmddyyHHMM*.met - All target MCFs for all Inserts on behalf of the PGE. The *ESDT* is the ESDT ShortName into which the file was Inserted, *mmddyy* is the month, day, and year of the Insert and *HHMM* is the time of the Insert.

11.16 Examining PDPS-Related Scripts and Message Files

This section describes how users may access files, in addition to the PGE-produced log files, which are created during the execution of a DPR job and which may hold information useful in tracing processing problems.

Some of these files are written by default to directory paths that can only be accessed on either the SGI processor machine or one of the Sun workstations. More detailed descriptions of these files and the conditions under which they are generated will be supplied in future Green Book versions.

11.16.1 Examining AutoSys JIL Scripts

JILxxxxxxxxx is the Job Information Language (JIL) script that defines the DPR job to **AutoSys** and which must be submitted to the **AutoSys** Database before a DPR job can be run. The name of the file created is system-generated and begins with the characters 'JIL' followed by nine characters (e.g. JILAAAa0066c).

Sample file content:

```
insert_job: 5251_823122483_1
job_type: command
command:  /usr/ecs/{mode}/CUSTOM/data/bin/sgi/EcDpAtExecutionMain
5251_823122483_1
machine: spr1sgigsfc
std_out_file: /home/cboettch/mockpge_msfc/out/dpat_std.out
std_err_file: /home/cboettch/mockpge_msfc/out/dpat_std.err
profile: /usr/ecs/<MODE>/CUSTOM/data/bin/sgi/EcDpAtRunProfile.sh
```

To examine JILxxxxxxxx scripts on the AIT Sun, execute the procedure steps that follow:

1 At the UNIX prompt on an AIT Sun, type **cd *JILscriptPathname***, press **Return**.

- The ***JILscriptPathname*** is the full path name to the location of the JILxxxxxxxx scripts to be examined.

2 At the UNIX prompt on the AIT Sun, type **vi *JILscriptFilename***, press **Return**.

- The ***JILscriptFilename*** is the file name of the JILxxxxxxxx script to be examined.
- This brings up the file named ***JILscriptFilename*** in the *vi* editor.
- Any text editor may be used such as *emacs*. For example, **emacs *JILscriptFilename***, press **Return**.

11.16.2 Examining Application Log Files (ALOG)

Most of the custom code used during SSI&T routinely produce log files. For example, the SSIT Manager produces a log file named **EcDpAtMgr.log** and the tool used to Insert SSEPs to the Data Server (EcDpAtInsertExeTarFile.sh) produces a log file named **EcDpAtInsertExeTarFile.log.** These files are placed in the directory in which the tool was executed. If the **SSIT Manager** is run from the user's home directory, then the log files for each of the associated tools will be found in the user's home directory. Log files are produced at the first invocation of the tools, even if no messages are written to them. During subsequent use of the tools, the associated log files will be appended.

Log files are generally named according to the convention:

ApplicationName.log

where *ApplicationName* is replaced with the name of the tool's executable binary. For tools that are shell scripts (*e.g.* .sh files), the shell name is left out of the log file name. For example, the tool EcDpAtInsertStaticFile.sh produces a log file named **EcDpAtInsertStaticFile.log** and not EcDpAtInsertStaticFile.sh.log.

Where an **SSIT Manager** application has been run using login **cmts1**, pw: **ecsu$er** with **dce_login**,the log files will be found using path: **/usr/ecs/{MODE}/CUSTOM/logs/.**

DCE failures have been encountered when installing **ESDT's , MCF's** and **.met files**. The term bounce the servers has been widely used in conjunction with the effort to re-install or delete files. **Bounce** means to **shut down a server** and then **bring them back up** to rid the servers of unwanted or old bindings. The nature of what needs to be done is outlined as follows:

1 Install or Delete **ESDT's** - the **SDSRV** and **ADSRV** need to be bounced after installation or removal of **ESDT**'s to allow for a refresh of the **DCE** cell management.

2 For **PGE…..odl, MCF's** and **.met files,** bouncing the servers **SDSRV** and A**DSRV** need to done after installation and reinstallation.

3 This can be done by loging into **ECS Assistant** for each server. The login should be with generic ID: **cmts1** and PW: **ecsu$er,** and **dce_login awhitele awhitele**.

11.17 DPREP

11.17.1 Introduction

This section contains information to run DPREP.

- DPREP is made up of three PGE's each run separately. The PGE's are titled Step1 DPREP, Step 2 DPREPand Step 3 DPREP.
- The input files come from INGEST. These files are depicted in two of the three step DPREP process in **Figure 11.17.1-1.** DPREP Step 3 will be available with Drop 5A.
- The output files generated from each of the DPREP PGE's contains Ancillary Attitude, and Ephemeris data that becomes new inputs to Instrument PGE's. These Instrument PGE's will then process its satellite data with similar time span files created by DPREP.
- The DPREP registration process for each of the three PGE's creates in the Science Data Server Archive a subscription for each of the DPREP PGE's. PGE execution then takes place in the PDPS.
- The SSI&T effort for DPREP PGE's is similar in effort to what would be required to register any other PGE.

11.17.2 SSI&T Activity for DPREP

The Level Zero datasets are received in 2 hour chunks. The file processes in Figure 11.17.1-1 depicts the minimal time span allowable for a DPREP run. In a normal operation of DPREP, a twenty four hour time span would be prepared for. This would require additional 2 hour chunks and thus additional files of data would need to be registered. Before the registration process can take place a number of files will have to be updated to process a block of data for a particular time period. Therefore, DPREP input files will have to be identified and various templates for the SSI&T process will require annotation.

The sections that follow have been highlighted with notations as to what SSI&T process applies in the preparation of each template and the function required to register each section. With a particular function identified, other portions of this manual can be referred to for more detailed precedures to be used to carry out the full SSI&T process.

Whenever new input files are introduced or updated executables are re-introduced it is wise to set up the PGE to run from the Command Line. This will determine if what has been introduced will run error free. After a successful Command Line run it is advisable then to complete the SSI&T effort to run from the PDPS. Command Line Runs include the use the PCF to run from in the Science Data Server. PDPS runs include ESDT's and ODL files to generate internal PCF's.

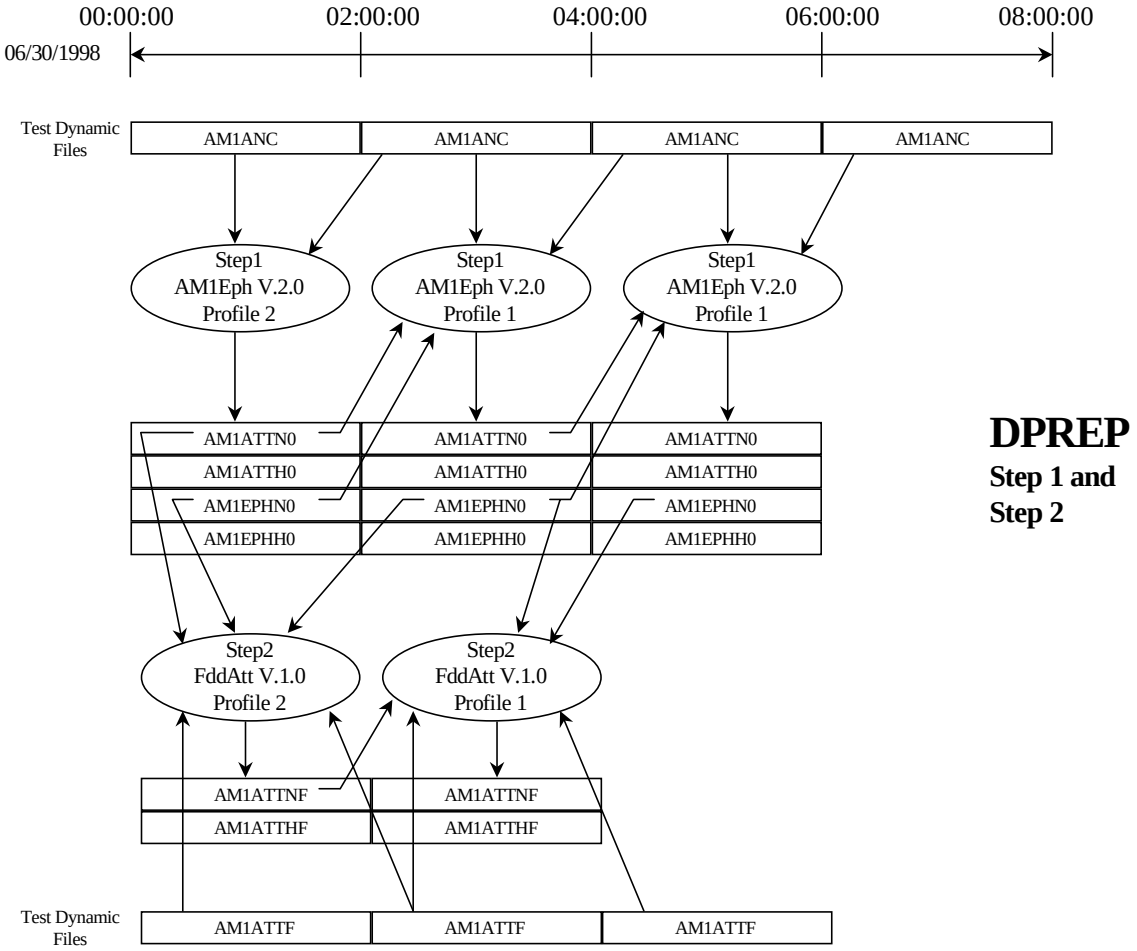


Figure 11.17.1-1. DPREP File Processes

11.17.3 DPREP Processes and Procedures

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

DPREP README and HowToRunDPREP files located at
:/usr/ecs/OPS/CUSTOM/data/DPS/

DPREP binary located: **:/usr/ecs/OPS/CUSTOM/bin/DPS/**

The following files are taken from an SSI&T SGI machine using the /data/DPS/ thread for referencing DPREP files in this section.

:

/data3/ecs/OPS/CUSTOM/data/DPS

lasher{emcleod}21: ls -lrt

total 13568

```
-rwxr-xr-x    1 cmops    cmops     449920 Jul 16  1998
P0420004AAAAAAAAAAAAAA96182015958001
-rwxr-xr-x    1 cmops    cmops        376 Jul 16  1998
P0420004AAAAAAAAAAAAAA96182015958000
-rwxr-xr-x    1 cmops    cmops      93408 Jul 16  1998
EOS_AM1_Ephemeris_FDF.eph
-rwxr-xr-x    1 cmops    cmops     451104 Jul 16  1998
EOS_AM1_Ephemeris_EDOS.eph
-rwxr-xr-x    1 cmops    cmops     449920 Jul 16  1998
P0420004AAAAAAAAAAAAAA96182035958001
-rwxr-xr-x    1 cmops    cmops        376 Jul 16  1998
P0420004AAAAAAAAAAAAAA96182035958000
-rwxr-xr-x    1 cmops    cmops      93152 Jul 16  1998 EOSAM1_1998-01-13.eph
-rwxr-xr-x    1 cmops    cmops      93184 Jul 16  1998 EOSAM1_1998-01-14.eph
-rwxr-xr-x    1 cmops    cmops      92672 Jul 16  1998 EOSAM1_1998-01-13.att
-rwxr-xr-x    1 cmops    cmops      29430 Jul 16  1998 DpPrPreProcess.pcf
-rwxr-xr-x    1 cmops    cmops      15058 Jul 16  1998 DpPrEphReplacer.pcf
-rwxr-xr-x    1 cmops    cmops      13840 Feb  8 15:21 PGS_101
-rwxr-xr-x    1 cmops    cmops       6207 Feb  8 15:21 HowToRunDPREP
-rwxr-xr-x    1 cmops    cmops       2834 Feb  8 15:21 HowToCreateDprepTarFile
-rwxr-xr-x    1 cmops    cmops        384 Feb  8 15:21 EDOS_LEVEL_ZERO_00.curr
-rwxr-xr-x    1 cmops    cmops     449344 Feb  8 15:21 EDOS_LEVEL_ZERO_01.curr
-rwxr-xr-x    1 cmops    cmops        384 Feb  8 15:21 EDOS_LEVEL_ZERO_00.next
-rwxr-xr-x    1 cmops    cmops     449984 Feb  8 15:21
EDOS_LEVEL_ZERO_01.next_2
-rwxr-xr-x    1 cmops    cmops     449984 Feb  8 15:21
EDOS_LEVEL_ZERO_01.next_1
-rwxr-xr-x    1 cmops    cmops     449280 Feb  8 15:21 EDOS_LEVEL_ZERO_01.next
-rwxr-xr-x    1 cmops    cmops        384 Feb  8 15:21
EDOS_LEVEL_ZERO_00.next_2
```

```
-rwxr-xr-x    1 cmops    cmops       384 Feb  8 15:21
EDOS_LEVEL_ZERO_00.next_1
-rwxr-xr-x    1 cmops    cmops       384 Feb  8 15:21
EDOS_LEVEL_ZERO_00.next.step3
-rwxr-xr-x    1 cmops    cmops    449984 Feb  8 15:21
EDOS_LEVEL_ZERO_01.next.step3
-rwxr-xr-x    1 cmops    cmops    225280 Feb  8 15:21
AM1_DEFINITIVE_ATT.fdd.next_1
-rwxr-xr-x    1 cmops    cmops    225280 Feb  8 15:21
AM1_DEFINITIVE_ATT.fdd.next
-rwxr-xr-x    1 cmops    cmops    225280 Feb  8 15:21
AM1_DEFINITIVE_ATT.fdd.curr
-rwxr-xr-x    1 cmops    cmops      5909 Feb  8 15:21 FDDAttitude.tar.met
-rwxr-xr-x    1 cmops    cmops     14000 Feb  8 15:21 AM1_REPAIR_EPH.fdd
-rwxr-xr-x    1 cmops    cmops      1681 Feb  8 15:21
AM1_DEFINITIVE_ATT.fdd.next_1.met
-rwxr-xr-x    1 cmops    cmops      1681 Feb  8 15:21
AM1_DEFINITIVE_ATT.fdd.next.met
-rwxr-xr-x    1 cmops    cmops      1681 Feb  8 15:21
AM1_DEFINITIVE_ATT.fdd.curr.met
-rwxr-xr-x    1 cmops    cmops      1654 Feb  8 15:21 AM1ANC.next_2.met
-rwxr-xr-x    1 cmops    cmops      1654 Feb  8 15:21 AM1ANC.next_1.met
-rwxr-xr-x    1 cmops    cmops      1654 Feb  8 15:21 AM1ANC.next.met
-rwxr-xr-x    1 cmops    cmops      1654 Feb  8 15:21 AM1ANC.curr.met
-rwxr-xr-x    1 cmops    cmops     26819 Feb  8 15:21
EcDpPrAm1FddEphemerisDPREP_PGEDPREP.Step3.PCF.curr
-rwxr-xr-x    1 cmops    cmops     28627 Feb  8 15:21
EcDpPrAm1FddAttitudeDPREP_PGE.Step2.PCF.next
-rwxr-xr-x    1 cmops    cmops     29026 Feb  8 15:21
EcDpPrAm1FddAttitudeDPREP_PGE.Step2.PCF.curr
-rwxr-xr-x    1 cmops    cmops     30520 Feb  8 15:21
EcDpPrAm1EdosAncillary.Steps1ab.PCF.next_1
-rwxr-xr-x    1 cmops    cmops     30553 Feb  8 15:21
EcDpPrAm1EdosAncillary.Steps1ab.PCF.next
-rwxr-xr-x    1 cmops    cmops     30485 Feb  8 15:21
EcDpPrAm1EdosAncillary.Steps1ab.PCF.curr
-rwxr-xr-x    1 cmops    cmops      5909 Feb  8 15:21
AM1_Ancillary_DPREP.tar.met
-rwxr-xr-x    1 cmops    cmops     26819 Feb  8 15:21
EcDpPrAm1FddEphemerisDPREP_PGEDPREP.Step3.PCF.next
-rwxr-xr-x    1 cmops    cmops      5816 Feb  8 15:21 DPREP_README
```

11.17.3.1 DPREP consists of three PGE's each run separately.

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

EcDpPrAm1EdosAncillary
EcDpPrAm1EdosEphemerisRepair
EcDpPrAm1ToolkitHdf

2 The second step is **EcDpPrAm1FddAttitudeDPREP_PGE.**

3 The third step is **EcDpPrAm1FddEphemerisDPREP_PGE**.

STEP ONE

EcDpPrAm1EdosAncillary reads in AM-1 L0 (EDOS) Ancillary Dataset (logical id 1000, ESDT AM1ANC). It also reads another set of AM-1 L0 (EDOS) Ancillary Dataset (logical id 1010, ESDT AM1ANC). The second set of L0 data is required to insure that incomplete orbits in the first data set get complete orbit metadata records. The only data that will be extracted from the second data set is the descending node time and longitude.

EcDpPrAm1EdosAncillary also reads in ephemeris and attitude data (toolkit native format) under logical ids 1500 (ESDT AM1EPHN0) and 1502 (ESDT AM1ATTN0). These would be the last ephemeris/attitude data sets generated from a previous run of this PGE.

EcDpPrAm1EdosEphemerisRepair

If EcDpPrAm1EdosAncillary signals a short gap was detected then EcDpPrAm1EdosEphemerisRepair reads the scratchfile created by EcDpPrAm1EdosAncillary and performs the gap fill and writes a gap filled native format ephemeris file. If no short gap is signaled then the scratch file is simply renamed to the native format ephemeris file.

EcDpPrAm1ToolkitHdf

This process takes the native format ephemeris file and produces a corresponding HDF file and a metadata file.

This PGE produces reformatted attitude and ephemeris data sets. The attitude data sets are produced using logical ids 1001 (ESDT AM1ATTN0) and 1100 (ESDT AM1ATTH0). The ephemeris data sets are produced using logical ids 1151 (ESDT AM1EPHN0) and 1152 (ESDT AM1EPHH0). The corresponding MCFs are accessed using logical ids 1400 (ESDT AM1EPHH0), 1401 (ESDT AM1ATTH0), 1402 (ESDT AM1EPHN0) and 1403 (ESDT AM1ATTN0).

The PGE produces an ASCII report file under logical id 2000

EcDpPrAm1EdosAncillary.Steps1ab.PCF.curr is a sample pcf used by this PGE. This pcf has directory locations and file names which may not be valid in all environments. Be careful before directly using this pcf. A sample set of input files for this PGE are

1000 EDOS_LEVEL_ZERO_00.curr
1000 EDOS_LEVEL_ZERO_01.curr
1010 EDOS_LEVEL_ZERO_00.next
1010 EDOS_LEVEL_ZERO_01.next

and a copy of each of these files is provided. The files in this group do contain small data gaps which will cause EcDpPrAm1EdosEphemerisRepair to be run.

The remaining pcfs for Step 1 DPREP:

EcDpPrAm1EdosAncillary.Steps1ab.PCF.next

EcDpPrAm1EdosAncillary.Steps1ab.PCF.next_1
will move through the sequence of the Level Zero files provided.
example; EDOS_LEVEL_ZERO_00.next is treated as current and
EDOS_LEVEL_ZERO_00.next_1 is treated as next etc...These pcfs can be utilized at the testers discretion and are not necessary for the running of steps 2 and 3. The input files indicated in EcDpPrAm1EdosAncillary.Steps1ab.PCF.next_1 contain no gaps so testers shouldn't be concerned that EcDpPrAm1EdosEphemerisRepair is not invoked when using this pcf.
Note:
The Level Zero datasets are received in 2 hour chunks but processing can't be performed on the most recently available 2 hour chunk. Step 1 processing needs to look forward in the time stream in order to complete orbit metadata processing.

STEP TWO

EcDpPrAm1FddAttitudeDPREP_PGE reads in the current FDD Attitude Dataset under logical ID 1000 and the next FDD Attitude Dataset under logical ID 1010. It also reads in the attitude data set it produced with it's last run under logical ID 1502. The output of this process is a native format attitude file (logical ID 1001) and an HDF format attitude file (logical ID 1100). A .met file is also produced for each.
EcDpPrAm1FddAttitudeDPREP_PGE.Step2.PCF.curr is a sample pcf used by this PGE. This pcf has directory locations and file names which may not be valid in all environments. Be careful before directly using this pcf. A sample set of input files for this PGE are
1000 AM1_DEFINITIVE_ATT.fdd.curr
1010 AM1_DEFINITIVE_ATT.fdd.next
(IMPORTANT: These files contain incorrect data and were delivered only as a placeholder. EcDpPrAm1FddAttitudeDPREP_PGE will not run with these files as input. When valid FDD Definitive Attitude files become available they should be moved to these filenames and EcDpPrAm1FddAttitudeDPREP_PGE should run successfully.)
and a copy of each of these files is provided.

Note: Step 2 processing must follow 2 hours behind step 1 processing.

STEP THREE

If step1 finds too many missing data points in the ephemeris data it signals that a definitve ephemeris file is needed from FDD which EcDpPrReplaceEphemeris will use to replace the ephemeris dataset that was generated from the Level Zero data.
EcDpPrReplaceEphemeris reads in the definitive Ephemeris dataset received from FDD (logical ID 1000) and the EOS_AM1 Ephemeris data (logical id 1001) in toolkit native format.
It produces Replacement ephemeris datasets (logical id 1101, ESDT AM1EPHH0) and (logical id 1100, ESDT AM1EPHN0). The corresponding MCFs are accessed using logical ids 1400 (ESDT AM1EPHH0) & 1402 (ESDT AM1EPHN0)
EcDpPrAm1FddEphemerisDPREP_PGE.Step3.PCF.curr is a sample pcf used by this PGE. This pcf has directory locations and file names which may not be valid in all environments. Be careful before directly using this pcf. A sample FDD input file for this PGE is
1000 AM1_REPAIR_EPH.fdd
and a copy of this file is provided.
Toolkit Initialization Settings

SGI Application Binary Interface New 32
Disk Space Used for PGE Run 50.000 MB
Shared Memory ON
Use Test Files if SM Fails ON
Use Log Files ON
Continue if Logging Fails OFF

11.17.4 HowToRunDPREP

DPREP is made up of 3 PGEs that are named:
Step1 DPREP, Step2 DPREP, Step3 DPREP.
Step1 DPREP is what used to be called **Am1Ancillary DPREP**.
Step2 DPREP is the **FDD Attitude DPREP**.
Step3 DPREP is **a combination of FDD Ephemeris and the replacer DPREPs**.
Note: Each PGE will use both the **Science Metadata update tool** and the **Operational Metadata Update tool** to incorporate the specific parameters listed in each of the direction sections below.

11.17.4.1 Step1 DPREP directions

There are 2 profiles for DPREP. Profile 1 takes in previous DPREP output and is expected to be run most of the time at the DAACs. Profile 2 takes in only the AM1 Ancillary data and will be run only for the first run of DPREP (because there is no previous output). Both profiles should be registered and executed.
PGENAME : AM1Eph
PGEVERSION : 2.0
PROFILE : 1
TopLevelShellName : EcDpPrAm1EdosEphAttDPREP_PGE
PGENAME : AM1Eph
PGEVERSION : 2.0
PROFILE : 2
TopLevelShellName : EcDpPrAm1EdosEphAttDPREP_PGE (same executable, only insert it once)
PGE Tar file location :
$ECS_HOME/<MODE>/CUSTOM/data/DPS/AM1_Ancillary_DPREP.tar
PGE Met file location :
$ECS_HOME/<MODE>/CUSTOM/data/DPS/AM1_Ancillary_DPREP.tar.met
Science Software Release 4
The PGE tar file contains the executables: EcDpPrAm1EdosEphAttDPREP_PGE, EcDpPrAm1EdosAncillary, EcDpPrAm1EdosEphemerisRepair, EcDpPrAm1ToolkitHdf.and the toolkit message file PGS_101
This PGE has no static data. This means Insert Static from SSIT can be skipped for this PGE.

However, there are 4 dynamic granules (2 files each) that this PGE needs as input. These 4 granules can be inserted from SSIT **Insert Test Dynamic Tool**

Granule 1:
ESDT Short Name : AM1ANC
Version : 001
Multi File granule : Y
Directory : $ECS_HOME/<MODE>/CUSTOM/data/DPS
File Names : EDOS_LEVEL_ZERO_00.curr, EDOS_LEVEL_ZERO_01.curr
Met file : AM1ANC.curr.met

Granule 2:
ESDT Short Name : AM1ANC
Version : 001
Multi File granule : Y
Directory : $ECS_HOME/<MODE>/CUSTOM/data/DPS
File Names : EDOS_LEVEL_ZERO_00.next, EDOS_LEVEL_ZERO_01.next
Met file : AM1ANC.next.met

Granule 3:
ESDT Short Name : AM1ANC
Version : 001
Multi File granule : Y
Directory : $ECS_HOME/<MODE>/CUSTOM/data/DPS
File Names : EDOS_LEVEL_ZERO_00.next_1, EDOS_LEVEL_ZERO_01.next_1
Met file : AM1ANC.next_1.met

Granule 4:
ESDT Short Name : AM1ANC
Version : 001
Multi File granule : Y
Directory : $ECS_HOME/<MODE>/CUSTOM/data/DPS
File Names : EDOS_LEVEL_ZERO_00.next_2, EDOS_LEVEL_ZERO_01.next_2
Met file : AM1ANC.next_2.met
3 PRs need to be created, two for Profile 1 and one for Profile 2. Create the PR for Profile 2 first (it runs on earlier data) and then create the PR for Profile 1.

The time frame for Production request editor for Profile 2 is 1998 June 30 00:00:00 to 1998 June 30 02:00:00

The time frame for Production request editor for Profile 1 is 1998 June 30 02:00:00 to 1998 June 30 04:00:00

The time frame for Production request editor for the second Profile 1 is 1998 June 30 04:00:00 to 1998 June 30 06:00:00

11.17.4.2 Step2 DPREP directions:

****
Step 2 DPREP can run ONLY after all three PRs for Step 1 DPREP have completed

****
There are 2 profiles for Step 2 DPREP. Profile 1 takes in previous Step 2 DPREP output and is expected to be run most of the time at the DAACs. Profile 2 takes in only the Fdd Att data and will be run only for the first run of DPREP (because there is no previous output). Both profiles should be registered and executed.

PGENAME : FddAtt
PGEVERSION : 1.0
PROFILE : 1
TopLevelShellName : EcDpPrAm1FddAttitudeDPREP_PGE

PGENAME : FddAtt
PGEVERSION : 1.0
PROFILE : 2
TopLevelShellName : EcDpPrAm1FddAttitudeDPREP_PGE (same executable, only insert it once)

PGE Tar file location : $ECS_HOME/<MODE>/CUSTOM/data/DPS/FDDAttitude.tar
PGE Met file location : $ECS_HOME/<MODE>/CUSTOM/data/DPS/FDDAttitude.tar.met
Science Software Version: 1.0

The PGE tar file contains the executable EcDpPrAm1FddAttitudeDPREP_PGE and the toolkit message file PGS_101
This PGE has no static data. This means Insert Static from SSIT can be skipped for this PGE. However, there are 3 dynamic granules that this PGE needs as input. These 3 granules can be inserted from SSIT insert test dynamic tool

Granule 1:
ESDT Short Name : AM1ATTF
Version : 001
Multi File granule : N
File Name :
$ECS_HOME/<MODE>/CUSTOM/data/DPS/AM1_DEFINITIVE_ATT.fdd.curr
Met file :
$ECS_HOME/<MODE>/CUSTOM/data/DPS/AM1_DEFINITIVE_ATT.fdd.curr.met

Granule 2:
ESDT Short Name : AM1ATTF
Version : 001
Multi File granule : N
File Name :
$ECS_HOME/<MODE>/CUSTOM/data/DPS/AM1_DEFINITIVE_ATT.fdd.next
Met file :
$ECS_HOME/<MODE>/CUSTOM/data/DPS/AM1_DEFINITIVE_ATT.fdd.next.met

Granule 3:
ESDT Short Name : AM1ATTF
Version : 001
Multi File granule : N
File Name :
$ECS_HOME/<MODE>/CUSTOM/data/DPS/AM1_DEFINITIVE_ATT.fdd.next_1
Met file :
$ECS_HOME/<MODE>/CUSTOM/data/DPS/AM1_DEFINITIVE_ATT.fdd.next_1.met

2 PRs need to be created, one for Profile 1 and one for Profile 2. Create the PR for Profile 2 first (it runs on earlier data) and then create the PR for Profile 1.

The time frame for Production request editor for Profile 2 is 1998 June 30 02:00:00 to 1998 June 30 04:00:00

The time frame for Production request editor for Profile 1 is 1998 June 30 04:00:00 to 1998 June 30 06:00:00

Note: The Data files and tar files are delivered only to the Science Data Server and cannot be directly accessed by SSIT. In order to insert these files they must be moved to the SSIT server manually.

11.17.4.3 DPREP Step1 profile1 PCF

PRODUCT INPUT FILES
##

(**Initial Construction Record**:)

1000|P0420004AAAAAAAAAAAAAA98300080000000.PDS|/net/raven/dss_amass/testdata/instrument_data/AM1/EDOS_HK_ANC/EDOSL0_ANC_PDS_0002||<A) insert UR here>||2

(**Current 2 hour Data File**)

1000|P0420004AAAAAAAAAAAAAA98300080000001.PDS|/net/raven/dss_amass/testdata/instrument_data/AM1/EDOS_HK_ANC/EDOSL0_ANC_PDS_0002||<B) insert UR here>||1
:

(**Future 2 hour Construction Record**)

1010|P0420004AAAAAAAAAAAAAA98300100000000.PDS|/net/raven/dss_amass/testdata/instrument_data/AM1/EDOS_HK_ANC/EDOSL0_ANC_PDS_0002||<C)
insert UR here>||2

(**Future 2 hour Data File**)

1010|P0420004AAAAAAAAAAAAAA98300100000001.PDS|/net/raven/dss_amass/testdata/instrument_data/AM1/EDOS_HK_ANC/EDOSL0_ANC_PDS_0002||<D)
insert UR here>||1

```
# The last ephemeris/attitude data sets generated.      (Found in PGE template)•
# ----------------------------------------------------------------------
# Ephemeris (Toolkit native format)              (Previous Time Range)•
1500|EOS_AM1_Ephemeris.21206.step1b.eph|/home/cmts1/DPREP/demooutput||<E) insert UR
here>||1
# Attitude (Toolkit native format)
1502|EOS_AM1_Attitude.21206.step1.att|/home/cmts1/DPREP/demooutput||<F) insert UR
here>||1
:
?   PRODUCT OUTPUT FILES
##########################################################################
                                        (Current Time Range)
# ----------------------------------------------------------------------
# Toolkit attitude datasets (output of EcDpPrAm1EdosAncillary).
# ----------------------------------------------------------------------
1001|EOS_AM1_Attitude.21208.step1.att|/home/cmts1/DPREP/demooutput||||1
1100|EOS_AM1_Attitude.21208.step1.hdf|/home/cmts1/DPREP/demooutput||||1
# ----------------------------------------------------------------------
# Toolkit ephemeris datasets (output of EcDpPrAm1EdosAncillary).
# ----------------------------------------------------------------------
1102|EOS_AM1_Ephemeris.21208.hdf|/home/cmts1/DPREP/demooutput||||1
# ----------------------------------------------------------------------
# Gap-filled HDF ephemeris dataset (output of EcDpPrAm1EphemerisGapFill).
# ----------------------------------------------------------------------
1151|EOS_AM1_Ephemeris.21208.step1b.eph|/home/cmts1/DPREP/demooutput||||1
# ----------------------------------------------------------------------
# Gap-filled Toolkit ephemeris dataset (output of
# EcDpPrAm1FormatNativeEphemeris).
# ----------------------------------------------------------------------
1152|EOS_AM1_Ephemeris.21208.step1b.hdf|/home/cmts1/DPREP/demooutput||||1
:
SUPPORT OUTPUT FILES
##########################################################################
10100|LogStatus|/home/cmts1/DPREP/logs||||1
10101|LogReport|/home/cmts1/DPREP/logs||||1
10102|LogUser|/home/cmts1/DPREP/logs||||1
10103|TmpStatus|||||1
10104|TmpReport|||||1
10105|TmpUser|||||1
10110|MailFile|||||1
:
?   USER DEFINED RUNTIME PARAMETERS
##########################################################################
999|No Previous Data Set; 1=true, 0=false.|0          (Profile 1 set to 0, Profile 2 set to 1)•
```

```
# ----------------------------------------------------------------------
# Toolkit version for which this PCF was intended.
# DO NOT REMOVE THIS VERSION ENTRY!
# ----------------------------------------------------------------------
10220|Toolkit version string|DAAC B.0 TK5.2.4     (Caution:  May need to change to higher•
level Toolkit)•
END
```

11.17.4.4 DPREP Step1 profile2 PCF (Profile 2 is start up PCF)

```
PRODUCT INPUT FILES
##########################################################################
1000|P0420004AAAAAAAAAAAAAA98300060000000.PDS|/net/raven/dss_amass/testdata/ins
trument_data/AM1/EDOS_HK_ANC/EDOSL0_ANC_PDS_0002||<A) insert UR here>||2
1000|P0420004AAAAAAAAAAAAAA98300060000001.PDS|/net/raven/dss_amass/testdata/ins
trument_data/AM1/EDOS_HK_ANC/EDOSL0_ANC_PDS_0002||<B) insert UR here>||1
1010|P0420004AAAAAAAAAAAAAA98300080000000.PDS|/net/raven/dss_amass/testdata/ins
trument_data/AM1/EDOS_HK_ANC/EDOSL0_ANC_PDS_0002||<C) insert UR here>||2
1010|P0420004AAAAAAAAAAAAAA98300080000001.PDS|/net/raven/dss_amass/testdata/ins
trument_data/AM1/EDOS_HK_ANC/EDOSL0_ANC_PDS_0002||<D) insert UR here>||1
# ----------------------------------------------------------------------
# The last ephemeris/attitude data sets generated.
# ----------------------------------------------------------------------
# Ephemeris (Toolkit native format)
#1500|EOS_AM1_Ephemeris.prev.eph|.||<E) insert UR here>||1
# Attitude (Toolkit native format)
#1502|EOS_AM1_Attitude.21206.step1.att|/home/cmts1/DPREP/output||<F) insert UR here>||1
:
?   PRODUCT OUTPUT FILES
##########################################################################
# ----------------------------------------------------------------------
# Toolkit attitude datasets (output of EcDpPrAm1EdosAncillary).
# ----------------------------------------------------------------------
1001|EOS_AM1_Attitude.21206.step1.att|/home/cmts1/DPREP/demooutput||||1
1100|EOS_AM1_Attitude.21206.step1.hdf|/home/cmts1/DPREP/demooutput||||1
# ----------------------------------------------------------------------
# Toolkit ephemeris datasets (output of EcDpPrAm1EdosAncillary).
# ----------------------------------------------------------------------
1102|EOS_AM1_Ephemeris.21206.hdf|/home/cmts1/DPREP/demooutput||||1
# ----------------------------------------------------------------------
# Gap-filled HDF ephemeris dataset (output of EcDpPrAm1EphemerisGapFill).
# ----------------------------------------------------------------------
1151|EOS_AM1_Ephemeris.21206.step1b.eph|/home/cmts1/DPREP/demooutput||||1
# ----------------------------------------------------------------------
```

```
# Gap-filled Toolkit ephemeris dataset (output of EcDpPrAm1FormatNativeEphemeris).
# ----------------------------------------------------------------------
1152|EOS_AM1_Ephemeris.21206.step1b.hdf|/home/cmts1/DPREP/demooutput||||1
?   SUPPORT OUTPUT FILES
##########################################################################
10100|LogStatus|/home/cmts1/DPREP/logs||||1
10101|LogReport|/home/cmts1/DPREP/logs||||1
10102|LogUser|/home/cmts1/DPREP/logs||||1
10103|TmpStatus|||||1
10104|TmpReport|||||1
10105|TmpUser|||||1
10110|MailFile|||||1
:
?   USER DEFINED RUNTIME PARAMETERS
##########################################################################
999|No Previous Data Set; 1=true, 0=false.|1
# ----------------------------------------------------------------------
# Toolkit version for which this PCF was intended.
# DO NOT REMOVE THIS VERSION ENTRY!
# ----------------------------------------------------------------------
10220|Toolkit version string|DAAC B.0 TK5.2.4
?   END
```

11.17.4.5 DPREP Step2 profile1 PCF

```
PRODUCT INPUT FILES
##########################################################################
1000|AM1_DEFATT_080000_212_1997_01.FDD|/home/cmts1/DPREP/am1defatt||<A) insert
UR here>||1
:
1010|AM1_DEFATT_100000_212_1997_01.FDD|/home/cmts1/DPREP/am1defatt||<B) insert
UR here>||1
# ----------------------------------------------------------------------
# The last ephemeris/attitude data sets generated.
# ----------------------------------------------------------------------
# Attitude (Toolkit native format)
1502|EOS_AM1_Attitude.21206.step2.att|/home/cmts1/DPREP/demooutput||<C) insert UR
here>||1
:
#1400|AM1EPHMH.mcf|.||||1
1401|AM1ATTHF.MCF|/home/cmts1/DPREP/mcf||||1
#1402|AM1EPHMN.mcf|.||||1
1403|AM1ATTNF.MCF|/home/cmts1/DPREP/mcf||||1
10250|MCF|||||1
10252|GetAttr.temp|||||1
10254|MCFWrite.temp|||||1
:
10501|EOS_AM1_Ephemeris.21208.step1b.eph|/home/cmts1/DPREP/demooutput||||2
10501|EOS_AM1_Ephemeris.21210.step1b.eph|/home/cmts1/DPREP/demooutput||||1
10502|INSERT_ATTITUDE_FILES_HERE|||||1
:
?   PRODUCT OUTPUT FILES
##########################################################################
:1001|EOS_AM1_Attitude.21208.step2.att|/home/cmts1/DPREP/demooutput||||1
1100|EOS_AM1_Attitude.21208.step2.hdf|/home/cmts1/DPREP/demooutput||||1
:
?   USER DEFINED RUNTIME PARAMETERS
##########################################################################
:
999|First Mission Data Set; 1=true, 0=false.|0
:10220|Toolkit version string|DAAC B.0 TK5.2.4
:END
```

11.17.4.6 DREP Step2 profile2 PCF

```
PRODUCT INPUT FILES
##########################################################################
1000|AM1_DEFATT_060000_212_1997_01.FDD|/home/cmts1/DPREP/am1defatt||<A) insert
UR here>||1
:
1010|AM1_DEFATT_080000_212_1997_01.FDD|/home/cmts1/DPREP/am1defatt||<B) insert
UR here>||1
# ----------------------------------------------------------------------
# The last ephemeris/attitude data sets generated.
# ----------------------------------------------------------------------
# Attitude (Toolkit native format)
1502|EOS_AM1_Attitude.21206.step1.att|/home/cmts1/DPREP/demooutput||<C) insert UR
here>||1
:
1400|AM1EPHMH.mcf|.||||1
1401|AM1ATTHF.MCF|/home/cmts1/DPREP/mcf||||1
1402|AM1EPHMN.mcf|.||||1
1403|AM1ATTNF.MCF|/home/cmts1/DPREP/mcf||||1
10250|MCF|||||1
10252|GetAttr.temp|||||1
10254|MCFWrite.temp|||||1
:
10501|EOS_AM1_Ephemeris.21206.step1b.eph|/home/cmts1/DPREP/demooutput||||2
10501|EOS_AM1_Ephemeris.21208.step1b.eph|/home/cmts1/DPREP/demooutput||||1
10502|INSERT_ATTITUDE_FILES_HERE|||||1
:
?   PRODUCT OUTPUT FILES
##########################################################################
1001|EOS_AM1_Attitude.21206.step2.att|/home/cmts1/DPREP/demooutput||||1
1100|EOS_AM1_Attitude.21206.step2.hdf|/home/cmts1/DPREP/demooutput||||1
:
?   USER DEFINED RUNTIME PARAMETERS
##########################################################################
:999|First Mission Data Set; 1=true, 0=false.|1
:
10220|Toolkit version string|DAAC B.0 TK5.2.4
:
?   END
```

11.17.4.7 Setups for DPREP

set path = (/usr/ecs/TS1/CUSTOM/TOOLKIT/toolkit/bin/sgi_daac_f90 $path)
setenv PGS_PC_INFO_FILE /home/cmts1/DPREP/pcf/Step1DP21206.PCF
setenv PGSMSG /home/cmts1/DPREP/msg
setenv PGSHOME /usr/ecs/TS1/CUSTOM/toolkit
setenv PGS_PC_INFO_FILE /home/cmts1/DPREP/pcf/Step1DP21208.PCF
setenv PGSMSG /home/cmts1/DPREP/msg
setenv PGSHOME /usr/ecs/TS1/CUSTOM/toolkit
setenv PGS_PC_INFO_FILE /home/cmts1/DPREP/pcf/Step1DP21210.PCF
setenv PGSMSG /home/cmts1/DPREP/msg
setenv PGSHOME /usr/ecs/TS1/CUSTOM/toolkit
setenv PGS_PC_INFO_FILE /home/cmts1/DPREP/pcf/Step2DP21206.PCF
setenv PGSMSG /home/cmts1/DPREP/msg
setenv PGSHOME /usr/ecs/TS1/CUSTOM/toolkit
setenv PGS_PC_INFO_FILE /home/cmts1/DPREP/pcf/Step2DP21208.PCF
* setenv PGSMSG /home/cmts1/DPREP/msg
* setenv PGSHOME /usr/ecs/TS1/CUSTOM/toolkit
Note: The **setenv** parameters are used for the **Command Line PGE** test. Only the **(*)** are used for **PDPS PGE** runs.

11.18 Learn more about SSI & T

11.18.1 References:

http://m0mss01.ecs.nasa.gov/smc/dc_master.html

` URL for ECS Project Training Material Volume 16: SSI&T December 1997:

MISR Science Data Processing Software Test Plan, Volume 2, Detailed Procedures and Facilities Version 2.0, Part 1 (PGE 1) June 1998.

http://dmserver.gsfc.nasa.gov/relbit/relbit.htm

SV DOC

REPOSITORY

Home

Drop Build Plans

Acceptance Test Plan

System Verifcation Test Plan

Access Database

Release B Testdata

Site Install and Checkout Test

End To End Test Procedures
Goddard DAAC M&O Status
VATC Status Page

ECS TEST
PAGES

Advertising Service (VATC)
User Registration Tool
(VATC)
V0 Web Client (VATC TS2)
V0 Web Client (DAAC
MODE TS2)
V0 Web Client (DAAC
MODE TS1)
V0 Web Client (DAAC
MODE OPS)

ECS TOOLS

EP7
RTM
CCR
EDHS
DDTS
Network Status Page
ECS Newsroom Server
Configuration Management
Release B Integration
ECS Telephone & Email Dir

ISO 9001

Business Manual Tab

Job Description Tab

Organization Charts Tab

Process Directives Tab

Training Tab

Miscellaneous Tab

Frequently Asked questions

EDHS

Raytheon Company

ESDT Basics **- http://dmserver.gsfc.nasa.gov/esdt/EsdtSection1/index.html**

GDAAC Directory for SSI&T:http://gsfcsrvr8.gsfcmo.ecs.nasa.gov/SSIT/

The ESDT Process (updated for drop 4) by Karl W. Cox, 22 December 1997

DCE Cell Manager Common User Tasks provided by IDG, February 6, 1998

- Tools and Techniques for Diagnosing Potential DCE Problems

MODIS - Science Data Processing Software Release 4 System Description

SDST-104, May 19, 1998

ECSINFO: **http://ecsinfo.hitc.com/iteams/Science/science.html**

PDPS howto are located on the EDF machines at: **/home/PDPS/docs/**

PDPS Web Page: **http//dmserver.gsfc,nasa.gov/ecsdev/relb/pdps/index.html**

For Troubleshooting or use the following EDF machines:

PDPS Troubleshooting Techniques are located on the EDF machines at **: /home/PDPS/troubleshooting/**

DPREP README files located at **:/usr/ecs/TS1/CUSTOM/data/DPS/**

DPREP binary located: **:/usr/ecs/TS1/CUSTOM/bin/DPS/**

11.18.2 Server Node Names Convention:

The naming convention is as follows:

Machine names are defined to be equivalent to the network hostname of the

machine. Network hostnames are limited to eight characters. On ECS we are now formatting these hostnames as svcimnni

Wheres : Site

g – GSFC
e – EDC
l – LaRC
n – NSIDC
a – ASF
j – JPL
m -- Mini-DAAC
t – VATC

v : Version

0 -- Release 4 At-Launch COTS design
1 -- Stood for B.1 COTS design; OBE, but still used in VATC
s -- Used for special SSI&T machines set up at GSFC and EDC

ci : Hardware Configuration Item

sp -- Science Processing (SPRHW)

ai -- Algorithm Integration and Test (AITHW)

aq -- Algorithm Quality Assurance (AQAHW)

pl -- Planning (PLNHW)

ms -- Management Subsystem (MSSHW)

cs -- Communications Subsystem (CSSHW)

in -- Interface (INTHW)

dm -- Data Management (DMGHW)

dr -- Data Repository (DRPHW)

ac -- Access Control Management (ACMHW)

ic -- Ingest Client (ICLHW)

wk -- Working Storage (WKSHW)

di -- Distribution (DIPHW)

as -- ASTER (ASTHW) [Occurs only at EDC]

te -- Test Equipment

m : Manufacturer

s -- Sun

g -- SGI

h -- HP

x -- X Terminal

p -- PC

nn : One-up number (01, 02, et cetera -- should be unique for the CI)

i : Interface type

<null> -- Production network

u -- User network

h -- HiPPI

Note that the machine name leaves off the last letter (the interface); hence, we generally refer to machines as "g0spg01", vice "g0spg01h". A machine may have multiple interfaces -- production, user, and HiPPI. So a single machine may show up in network documentation multiple times (g0spg01, g0spg01h, g0spg01u).

11.18.3 A Handy Alias file to use while conducting SSI&T:

lasher{emcleod}51: **alias**
+ pushd
- popd
More more !* |grep -v "Msg: Caught dce error: No more bindings (dce / rpc)" |grep -v "MsgLink :0 meaningfulname :EcAgManager::Recovery" |grep -v "MsgLink :0 meaningfulname :DsShSRequestRealSetStateSettingState" |grep -v "Command 1/1 execution complete"
cdstagebin cd /ecs/formal/STAGE/DSS/bin/sun5.5
dbg debugger -bg NavajoWhite -fn 12x24 !* &
dbs /home/jzhuang/bin/dbrowser-syb &
disp setenv DISPLAY !*
mgr DpAtMgr ConfigFile /usr/ecs/TS1/CUSTOM/cfg/DpAtMG.CFG ecs_mode TS1&
ops cd /usr/ecs/OPS/CUSTOM
ts1 cd /usr/ecs/TS1/CUSTOM
xslq_autosys isql -Uautosys -Pautosys -Staltos_srvr
xsql_css isql -Ucss_role -Pwelcome -Ssahara_srvr
xsql_dss isql -UsdsrvApp -Pwelcome -Sjourney_sqs222_srvr
xsql_ios isql -Uios_role -Pwelcome -Sincagold_srvr
xsql_pdps isql -UpdpsUsers -Pwelcome -Sodyssey_srvr

 # alias for browser
Note: On mini daac, dbbrowser has to originate from workstation ODYSSEY to execute alias db_pdps to reach PDPS DB on ILLIAD.

alias db_dss '/home/opscm/dbr/dbrowser-syb -UsdsrvApp -Pwelcome -Sjourney_sqs222_srvr &'
alias db_ios '/home/opscm/dbr/dbrowser-syb -Uios_role -Pwelcome -Sincagold_srvr &'
alias db_css '/home/opscm/dbr/dbrowser-syb -Ucss_role -Pwelcome -Ssahara_srvr &'
alias db_autosys '/home/opscm/dbr/dbrowser-syb -Uautosys -Pautosys -Staltos_srvr &'
alias db_pdps '/home/opscm/dbr/dbrowser-syb -UpdpsUsers -Pwelcome -Silliad_srvr &'
alias db_ing '/home/opscm/dbr/dbrowser-syb -UEcInPolling -P3nWK0fG1 -Shuckfinn_srvr &'
alias db_stmgt '/home/opscm/dbr/dbrowser-syb -UEcDsStFtpDisServer -PS71Oq4y3 -Shuckfinn_srvr &'

ls -laF look at root and .cshrc, .alias

11.18.4 HOWTO_SSIT HELPFUL NOTES

Xterm format to bring up xterm windows for servers all at once

This list of xterm identifiers should be assigned a file name and placed into your home directory. Then invoke the file name when you want to create the entire list of xterms.

Example of filename: - xterm_pls

```
xterm -sb -sl 10000 -fg green -bg black -name "Resource Editor" &
xterm -sb -sl 10000 -fg green -bg black -name "Resource Model" &
xterm -sb -sl 10000 -fg green -bg black -name "Production Request Editor" &
xterm -sb -sl 10000 -fg green -bg black -name "Planning Workbench" &
xterm -sb -sl 10000 -fg green -bg black -name "Database" &
xterm -sb -sl 10000 -fg green -bg black -name "Planning Timeline" &
xterm -sb -sl 10000 -fg green -bg black -name "Logs" &
xterm -sb -sl 10000 -fg green -bg black -name "ECS Assist" &
xterm -sb -sl 10000 -fg green -bg black -name "g0sps06" &
xterm -sb -sl 10000 -fg green -bg black -name "g0ais01" &
xterm -sb -sl 10000 -fg green -bg black -name "g0spg01" &
xterm -sb -sl 10000 -fg green -bg black -name "g0drg01" &
xterm -sb -sl 10000 -fg green -bg black -name "g0pls02" &
# setenv ECS_HOME /usr/ecs
# setenv MODE TS2
```

11.18.4.1 howto_start_and_kill_servers_drop4

```
**********************************************************************
Start servers
**********************************************************************
% rlogin <any_machine> -l cmops
# To start ALL servers:
/home/cmops/restart/Drop4/Ec.mgr start all TS1
# Substitute subsystem name for "all" to do for a single subsystem
```

```
# Valid susbsystem names
# EcDm
# EcDp
# EcDsDist
# EcDsSr
# EcDsSt
# EcDsStIn
# EcIn
# EcIo
# EcMsAc
# EcPl
# EcSb
**********************************************************************
Kill servers -- AFTER WARNING EVERYONE WORKING IN YOUR MODE!
**********************************************************************
% rlogin <any_machine> -l cmops
# To kill ALL servers:
/home/cmops/restart/Drop4/Ec.mgr stop all TS1
# Substitute subsystem name for "all" to do for a single subsystem
EcDm
EcDp
EcDsDist
EcDsSr
EcDsSt
EcDsStIn
EcIn
EcIo
EcMsAc
EcPl
EcSb
```

11.18.4.2 howto_install_esdt_drop4pL

```
INSTALLING ESDTs:
mini-daac% rlogin texas -l cmops
# passwd opsu$er
##########################################################################
# Make sure IOS and SDSRV are up
##########################################################################
# Use either ECS Assist on texas *and* incagold,
# Alternatively, use ps:
texas:> ps -ef | grep EcDsScienceDataServer | grep ConfigFile
   sdsrv 24410     1  0 10:17:57 pts/3    0:33
/usr/ecs//OPS/CUSTOM/bin/DSS/EcDsScienceDataServer ConfigFile /usr/ecs//OPS/CUS
incagold:> ps -ef | grep  EcIoAdServer
```

```
    ios  5864     1  0 10:17:07 pts/0    0:28
/usr/ecs//OPS/CUSTOM/bin/IOS/EcIoAdServer ConfigFile /usr/ecs//OPS/CUSTOM/cfg/E
# If either is not present, that server is down.
# You must arrange to have it up before continuing the install
##############################################################################
# If you have ever installed this ESDT before,
# you must first delete its advertisement
##############################################################################
See howto_remove_esdt
##############################################################################
# Install ESDT on texas
##############################################################################
setenv MODE OPS
source /usr/ecs/${MODE}/CUSTOM/utilities/EcCoEnvCsh
# First make sure correct descriptor is in the right directory
texas:> cd /usr/ecs/${MODE}/CUSTOM/data/ESS
texas:> ls -l *<shortname>*
# Substitute your ShortName for <shortname>
# You should see 1 file ending in .desc
# Run GUI to add ESDT
cd /usr/ecs/${MODE}/CUSTOM/utilities
EcDsSdSrvGuiStart $MODE

Click Add, then type in:
Name: <shortname>
Version: 1
Click first File.., then edit Filter so it gets your ESDTs:
(for example, if your ESDTs all contain the string "CER")
/usr/ecs/OPS/CUSTOM/data/ESS/*CER*
Select your ESDT from the list, click OK
Click second File.., then edit Filter so it gets your ESDTs:
(for example, if your ESDTs all contain the string "CER")
/usr/ecs/OPS/CUSTOM/lib/ESS/*CER*
Select your ESDT from the list, click OK
Type in:
Archive ID: DRP1_OPS
Click OK
ESDT begins to install
If you get a "Failure to Add Datatype" message,
go to SDSRV logs and examine them
cd /usr/ecs/OPS/CUSTOM/logs
vi EcDsScienceDataServer.ALOG
vi EcDsScienceDataServerDebug.log
# Go to end of log file
```

```
:$
# Now look for your ShortName, then read error messages if any
# Make sure you are looking at recent error messages by comparing
# to the current time
##########################################################################
# Check that ESDT is in SDSRV database
##########################################################################
# Invoke ISQL to look at database
texas% isql -UsdsrvApp -Pwelcome -Sjourney_sqs222_srvr
# Get correct Dsrv database
1> use relbdss
2> go
# See your ShortName
1> select ShortName from DsMdCollections where ShortName="<shortname>"
# Substitute your ShortName for <shortname>
2> go
# Your ShortName should appear on the screen
##########################################################################
# Check that ESDT was advertised
##########################################################################
# Open URL http://incagold:4900 with netscape
# Click "Search All"
# Enter ShortName in "Search for" box
# Click "Service entry" box
# Click "Search" button at bottom of page
# You should see a list of items, all for your ShortName
# Make sure Insert Service is one of them, so you can do Inserts
# Alternatively, run a script to see all the services
mini-daac% rlogin incagold -l ios
# passwd Mak!thpn
cd /home/ios/scripts
listESDT.csh <shortname>
# passwd welcome
# You should see a bunch of lines with your ShortName in them
```

11.18.4.3 howto_do bulk install of ESDT's

Below is a script you can modify to install several ESDTs at once.
I tested up installation of up to 8 ESDTs at once in 4PX/TS1, and it worked.
(Haven't tried to Insert any granules yet though ;-)
Success for each ESDT install occurs when you see the message
/usr/ecs/TS1/CUSTOM/data/ESS/DsESDTMiMISL0CF.001.desc added successfully.
scroll by on your screen.
Note that you will also see a message

Details:
-UnnamedPL[]
/usr/ecs/TS1/CUSTOM/utilities/EcDsSrAddesdt[144]: 11668 Bus Error(coredump)
-- just ignore this ;-)
Suggest you start a "script" session to save all the screen output before you
start, so you'll know which ESDTs successfully installed.

```
#!/bin/sh -f
#
# Install MPGE1 ESDTs not yet installed for MPGE7
#
# Usage:
#     install_mpge1_esdts.sh <mode>
#
# To use with other ESDTs:
#    Change PFX to your instrument prefix, e.g., "Mo" for MODIS
#    Change the ESDT list "for ESDT in" to your own list

if [ ! "$1" ] ; then
   echo Usage: $0 \<mode\>
   exit
fi
m=$1
if [ "$m" != "OPS" -a "$m" != "TS1" -a "$m" != "TS2" -a "$m" != "TS3" ]; then
   echo Mode must be one of {OPS,TS1,TS2,TS3}
   exit
fi
EXE=/usr/ecs/${m}/CUSTOM/utilities/EcDsSrAddesdt
VOLUME_GROUP=VG5

PFX=Mi
V=001

# for ESDT in MIRCCT MIANCSSC MIANPP MIRFOI MIANCARP MISL0AA MISL0AF
MISL0AN
MISL0BA MISL0BF MISL0CA MISL0CF MISL0DA MISL0DF
for ESDT in MIANRCCH MIL1A MI1B2T MI1B2E MIRCCM MISQA MI1B1
do
   $EXE $m $VOLUME_GROUP NULL NULL DsESDT${PFX}${ESDT}.${V}.desc
done
```

11.18.4.4 howto_remove_esdt_drop4pX

First delete advertisement
rlogin

```
incagold -l opscm
rlogin incagold -l cmts1
setenv MODE TS1
cd /usr/ecs/${MODE}/CUSTOM/utilities
source EcCoEnvCsh
cd /usr/ecs/${MODE}/CUSTOM/bin/IOS
ContributionDriver $MODE
awhitele
awhitele
3
2
your_short_name_here
y
# Success is when the "<" prompt returns
# To make sure the advertisements are deleted from the database
incagold% isql -Uios_role -Pwelcome -Sincagold_srvr
[If not OPS mode]
1> use IoAdAdvService_MODE
[where MODE is your mode, e.g. OPS]
[if OPS mode}
1> use IoAdAdvService
2> go
1> select title from IoAdAdvMaster where title like "%your_short_name_here%"
2> go

Result should be no rows returned.
If you do get rows returned, the delete of the advertisement
did not work.
*****************************************************************************
Clean up DDICT
*****************************************************************************
rlogin incagold -l cmts1
setenv MODE TS1
cd /usr/ecs/${MODE}/CUSTOM/dbms/DMS
DmDbCleanCollection.csh <short_name> 1
****IMPORTANT******
CHANGE Database Name FOR THE MODE IN QUESTION
****IMPORTANT******
Enter Server Name (DSQUERY): incagold_srvr
Enter Database Name: EcDmDictService_TS1
Enter User Name: dms_role
Enter password: welcome
incagold_srvr
EcDmDictService_TS1
```

dms_role
welcome
EcCoEnvCsh appears to have wrong value of SYBASE
setenv SYBASE /tools/sybOCv11.1.0
****
Then delete ESDT from Data Server
****
rlogin texas -l cmts1
setenv MODE TS1
cd /usr/ecs/${MODE}/CUSTOM/utilities
EcDsSrRmesdt $MODE descriptor_FILENAME_here
e.g. descriptor_FILENAME_here is
DsESDTMiMISL0AA.001.desc
(NOTE: Do not put the directory in front of the filename in Drop 4PL)
Success is no error msgs
****
Kill servers -- AFTER WARNING EVERYONE WORKING IN YOUR MODE!
****
kill Sdsrv & HdfEosSrv & AdSrvr
% rlogin <any_machine> -l opscm
cd /home/jzhuang/test/ShutDown
sdsrv.shutdown OPS
ios-dm-mss.shutdown OPS
****
Clean up DCE & Restart servers
****
(Same login as last step)
start Sdsrv & HdfEosSrv on texas
start SubSrvr on incagold
(cleanup done automatically)
****
Now reinstall the ESDT
****
texas
See howto_install_esdt.t

11.18.4.5 howto_delete_granules_frm_sdsrv_drop4pX

Delete granules from Sdsrv
rlogin texas -l cmops
To delete all granules for a given ESDT from the Data Server database:
MAKE SURE YOU USE THE RIGHT MODE!
else people may get mad at you
setenv DSQUERY journey_sqs222_srvr
setenv DBNAME relbdss

setenv DBUSERNAME sdsrv_role
setenv DBPASSWD popcorn11
setenv MODE TS1
/usr/ecs/${MODE}/CUSTOM/dbms/DSS/DsDbCleanGranules your_short_name_here 1
****
Delete granules from Stmgt
****
Once you have done this, there is no way for the system
to ever retrieve these granules from Stmgt again.
Therefore you can delete them from Stmgt:
rlogin raven -l cmops
cd /dss_amass/OPS
WARNING: BE VERY CAREFUL USING THIS COMMAND
If you type it incorrectly, you could delete *everyone's* granules!
rm *your_short_name_here*

****

Delete granules from Pdps
****
See howto_delete_single_file_granule_in_pdps

11.18.4.6 howto_ingest_4px

4PX/OPS
On raven,
tail -f EcDsStArchiveServerDebug.log
On huckfinn,
tail -f EcInPolling.EDOSDebug.log
tail -f EcInGranDebug.log
Login to huckfinn as cmops.
Polling directory is $POLL = /usr/ecs/OPS/CUSTOM/icl/huckfinn/data/pollEDOS
Prepare data to Ingest in a directory, e.g.,
$ING = /SSIT/MISR/PGE1/2.01/L0
If necessary, determine correct filename for PDS from
Eric Chomko, & rename your L0 files, and create
the Ingest input files *.PDR and *.PDR.XFR .
For MISR AA camera these files are
huckfinn:/SSIT/MISR/PGE1/2.01/L0>lt *0420330AAAAAAAAAAAAAA960071629590*
-rw-rw-rw- 1 cmts1 cmts1 258278400 Oct 7 16:50
P0420330AAAAAAAAAAAAAA96007162959001.PDS
-rw-rw-rw- 1 cmts1 cmts1 384 Oct 7 17:51
P0420330AAAAAAAAAAAAAA96007162959000.PDS
-rwxrwxr-x 1 fhuang users 975 Oct 7 18:06
X0420330AAAAAAAAAAAAAA960071629590.PDR*

```
-rwxrwxr-x    1 fhuang   users        39 Oct  7 18:06
X0420330AAAAAAAAAAAAAAA960071629590.PDR.XFR
```
For MISR DF camera these files are
calahans:/SSIT/MISR/PGE1/2.01/L0>lt *0420320*
```
-rw-rw-rw-   1 cmts1    cmts1    299059200 Oct  7 16:35
P0420320AAAAAAAAAAAAAAA96007162959001.PDS
-rwxrwxrw-   1 cmts1    cmts1         39 Oct  7 16:58
X0420320AAAAAAAAAAAAAAA960071629590.PDR.XFR*
-rwxrwxrw-   1 cmts1    cmts1        989 Oct  7 17:41
X0420320AAAAAAAAAAAAAAA960071629590.PDR*
-rw-rw-rw-   1 cmts1    cmts1        384 Oct 13 09:33
P0420320AAAAAAAAAAAAAAA96007162959000.PDS
```
Make sure the .PDR file contains the path pointing
to the L0 files *.PDS .
When ready to Ingest, copy the *.PDR and *.PDR.XFR files
to the $POLL directory.
Ingest automatically picks up the files every 120 sec.
If need to Start servers:
```
cd /usr/ecs/OPS/CUSTOM/utilities
EcInStart OPS cold
grep "End of Start" $LOGS/EcInGranDebug.log
```
Also check ECS Monitor in EcsAssist: make sure EcInPolling.EDOS is UP.

11.18.4.7 howto_register_misr_pges

Copy over ODL
 (MISR PGEs 7,1,8c)
```
setenv OTS1 /usr/ecs/TS1/CUSTOM/data/DPS/ODL
setenv OOPS /usr/ecs/OPS/CUSTOM/data/DPS/ODL
\cp -p $OTS1/ESDT_MI*.odl $OOPS
\cp -p $OTS1/ESDT_AM1*.odl $OOPS
\cp -p $OTS1/PGE_MPGE* $OOPS
\cp -p $OTS1/ORBIT_MISR.odl $OOPS
\cp -p $OTS1/PATHMAP_MISR.odl $OOPS
```
****
MPGE7
****
Install ESDTs
++++++++++++
MISANCGM
MICNFG
MIANCAGP
AM1EPHN0
AM1ATTN0

MIB2GEOP

Scripts are in
/usr/ecs/TS1/CUSTOM/ssit/lasher/data/MISR/PGE7.FullSwath/pdps

Examples are for OPS mode
++++++++++++
Register PGE
++++++++++++
run_sci.sh OPS
EcDpAtOpDbGui ConfigFile /usr/ecs/OPS/CUSTOM/cfg/EcDpAtOpDbGui.CFG ecs_mode
OPS &
++++++++++++++++++
Insert static input
++++++++++++++++++
insert_stat_mpge7.sh OPS
++++++++++++++++++
Insert dynamic input
++++++++++++++++++
insert_stat_mpge7.sh OPS
++++++++++++++++++
Insert tar file
++++++++++++++++++

****
MPGE1
****
Install ESDTs
+++++++++++++
MIRCCT
MIANCSSC
MIANPP
MIRFOI
MIANCARP
MISL0AA
MISL0AF
MISL0AN
MISL0BA
MISL0BF
MISL0CA
MISL0CF
MISL0DA
MISL0DF
MIANRCCH

MIL1A
MI1B2T
MI1B2E
MIRCCM
MISQA
MI1B1

Archive ID:
DRP1_OPS:VG4
Scripts are in
/usr/ecs/TS1/CUSTOM/ssit/lasher/data/MISR/PGE1/2.01/pdps
++++++++++++
Register PGE
++++++++++++
DefinePGE.sh OPS
EcDpAtOpDbGui ConfigFile /usr/ecs/OPS/CUSTOM/cfg/EcDpAtOpDbGui.CFG ecs_mode
OPS &
For copying over the OpDbGui stuff for all 9 PGEs:
update PlPgePerformance set
cpuTime=7561,pgeElapsedTime=3844,dprElapsedTime=3844,maxMemory=2905.5,runCpuTim
e=7561,runMaxMemory=2905.5,runPgeElapsed=3844,runDprElapsed=3844
where pgeId like "MPGE1%:"
update PlPgeMaster set pgeTestOperational=1 where pgeId like "MPGE1%"
++++++++++++++++++
Insert static input
++++++++++++++++++
insert_stat_mpge1.sh OPS
(current version does not Insert same granules used by PGE 7)
++++++++++++++++++
Insert dynamic input
++++++++++++++++++
Get Feng to Insert 9 L0 granules
ATT/EPH already Inserted for PGE 7
++++++++++++++++++
Insert tar file
++++++++++++++++++
##
File MPGE4#2.0_checklist
##
Generated by opscm using EcDpAtMkcfg on Sun Jan 11, 1998 15:03:40
###
DpAtMgr log initialization file.
parameter=value pairs:

```
#    DATABASE=path and filename
#    CHECKLIST=name of checklist
#    ITEM=checklist_item_name

DATABASE=/home/toma/green/MPGE4#2.0#02
```

11.18.4.8 Sample CHECKLIST for: MPGE4#2.0#02

```
ITEM=Install ESDTs
ITEM=Unpack Delivered Algorithm Package
ITEM=Determine path to new 32-bit IMSL library
ITEM=Edit SCF environment variables; change to DAAC values
ITEM=Edit SCF PCF; change file paths to DAAC values
ITEM=Examine SCF PCF; ensure paths to leapsec, utcpole, planetary eph files OK
ITEM=Examine SCF PCF; ensure LID for udunits file is 10302, not 11001
ITEM=Examine SCF PCF; ensure all output file paths have write permission
ITEM=Run PGE from command line; check against expected results
ITEM=Do test Insert of command line output .met file to Data Server
ITEM=Determine mapping of PCF logical IDs to ESDT Short Names
ITEM=Create PGE PDPS metadata ODL file template
ITEM=Edit PGE PDPS metadata ODL file to add ESDT Short Names, other info
ITEM=Create ESDT PDPS metadata ODL files
ITEM=Register SCIENCE metadata in PDPS database
ITEM=Register OPERATIONAL metadata in PDPS database
ITEM=Create .met files for PGE input file Inserts to Data Server
ITEM=Insert static files to Data Server
ITEM=Edit MISR_PGE.sh; replace MISR_PGE_start.pl with PCS_main
ITEM=Create -profile- file
ITEM=Create tar file of PGE executables, scripts, message files
ITEM=Insert tar file of PGE executables to Data Server
ITEM=Insert test dynamic files to Data Server
ITEM=Add UR of DataServer itself to PDPS database table with FixMcfGetScript
ITEM=Edit Toolkit template PCF;  Delete Toolkit ephemeris/attitude LIDs
ITEM=Examine Toolkit template PCF; ensure path to planetary eph files OK
ITEM=Examine Toolkit template PCF; ensure LID for udunits file 10302, not 11001
ITEM=Create DPR with Production Request Editor
ITEM=Create Plan and Kickoff PGE with Planning Workbench
ITEM=Invoke AutoSys to monitor PGE execution
ITEM=Place steps Execution (E) and Insert (I) steps On Hold in AutoSys
ITEM=Before Execution: Change PCF paths to leapsec and utcpole to MISR versions
ITEM=Before Execution: Ensure PCF output paths have write permission
ITEM=Before Execution: Change MCF CLASS value for PSAs with Data_Location=MCF
ITEM=Before Insert: Eliminate duplicate URs from InputPointer field in .met
ITEM=Before Insert: Truncate EquatorCrossingTime to 15 characters in .met
ITEM=Compare archived output file with expected test output
```

```
##################################################################################
```

11.18.4.9 howto_delete_misr_chain_granules_from_db

```
DELETING ALL MISR CHAIN GRANULES FROM PDPS DATABASE
Do this after deleting all DPRs.
These instructions are for deleting output granules:
references.
Ignore /* */ below
/*
select granuleId from PlDataGranule where dataTypeId like "MIB2GEOP%"
or dataTypeId like "MIANRCCH%"
or dataTypeId like "MIL1A%"
or dataTypeId like "MI1B2T%"
or dataTypeId like "MI1B2E%"
or dataTypeId like "MI1B1%"
or dataTypeId like "MIRCCM%"
or dataTypeId like "MISQA%"
or dataTypeId like "MIL2TCAL%"
*/

--select granuleId from DpPrGranuleLocation
/*
delete DpPrGranuleLocation
where granuleId like "MIB2GEOP%"
or granuleId like "MIANRCCH%"
or granuleId like "MIL1A%"
or granuleId like "MI1B2T%"
or granuleId like "MI1B2E%"
or granuleId like "MI1B1%"
or granuleId like "MIRCCM%"
or granuleId like "MISQA%"
or granuleId like "MIL2TCAL%"
*/
--select fileName from DpPrFile
/*
delete DpPrFile
where universalReference like "%MIB2GEOP%"
or universalReference like "%MIANRCCH%"
or universalReference like "%MIL1A%"
or universalReference like "%MI1B2T%"
or universalReference like "%MI1B2E%"
or universalReference like "%MI1B1%"
or universalReference like "%MIRCCM%"
or universalReference like "%MISQA%"
```

```
or universalReference like "%MIL2TCAL%"
*/
--select fileName from DpPrDiskAllocation
/*
delete DpPrDiskAllocation
where fileName like "%MIB2GEOP%"
or fileName like "%MIANRCCH%"
or fileName like "%MIL1A%"
or fileName like "%MI1B2T%"
or fileName like "%MI1B2E%"
or fileName like "%MI1B1%"
or fileName like "%MIRCCM%"
or fileName like "%MISQA%"
or fileName like "%MIL2TCAL%"
*/
/*
update PlRscDiskPartition
set diskUsage = (select sum(diskAllocationSize) from DpPrDiskAllocation)
*/
select * from PlRscDiskPartition

delete PlDataGranuleShort where dataTypeId like "MIB2GEOP%"
or dataTypeId like "MIANRCCH%"
or dataTypeId like "MIL1A%"
or dataTypeId like "MI1B2T%"
or dataTypeId like "MI1B2E%"
or dataTypeId like "MI1B1%"
or dataTypeId like "MIRCCM%"
or dataTypeId like "MISQA%"
or dataTypeId like "MIL2TCAL%"
```

11.18.4.10 howto_query_pdps_db_for_missing_urs

```
1> select a.granuleId, a.universalReference from PlDataGranuleShort a,
2> PlDprData b
3> where b.dprId like "%q4L%"
4> and a.granuleId = b.granuleId
5> and b.ioFlag = 0 and a.universalReference not like "UR%"
```

If anything is returned, it means this DPR has inputs that
do not have URs, which is a problem.1> select a.granuleId, a.universalReference
from PlDataGranuleShort a,

```
2> PlDprData b
3> where b.dprId like "%q4L%"
4> and a.granuleId = b.granuleId
```

5> and b.ioFlag = 0 and a.universalReference not like "UR%"
If anything is returned, it means this DPR has inputs that
do not have URs, which is a problem.
##

11.18.4.11 howto_run_in_pdps_drop4pX

rlogin odyssey -l cmts1
dce_login awhitele awhitele
setenv MODE TS1
cd /usr/ecs/${MODE}/CUSTOM/utilities
source EcCoEnvCsh
setenv DISPLAY ncdso7:0.0
setenv DISPLAY ncdso32:0.0
setenv DISPLAY ncdso8:0.0
setenv DISPLAY ncdso30:0.0
setenv DISPLAY ncdp11:0.0

****
Enter a Production Request using PR Editor
****
script your_scriptname_here
EcPlPRE_IFStart $MODE
Click PREdit tab
Select your PGE, click OK, wait 10-15 seconds
Enter start & stop time of your input data.
Click File menu, then Save As, type in a name to save your DPR.

After 15-30 seconds, a dialog box pops up.
If it says "1 DPR generated", you're in business!
If it says "0 DPRs generated", you're in trouble. Check the PREditor logs.

Exit PR Editor.
****
If PR fails to explode
****
Check error messages on screen.
After making fixes, delete entries from PlDataGranuleShort
for your PGE.
For example
isql -UpdpsUsers -Pwelcome -Silliad_srvr
1> use pdps_TS1
2> go
Rerun PREditor.
****

Bring up Planning Workbench
****
cd /usr/ecs/TS1/CUSTOM/utilities
EcPlAllStart $MODE 1
Wait 10-15 seconds for it to come up.

If you get a message like:
Process EcPlSubMgr is already running in mode TS1 with PID ...

then check if someone else is running Planning Workbench.
(Only one copy may run at a time.)
If not, do
EcPlSlayAll $MODE
then retry EcPlStartAll.
****
Schedule a Production Request
****
Under FILE click NEW.
Type in a name for your Plan.
Click your PR in the Production Requests Unscheduled window.
Click Schedule.
Click Save in the dialog box.
Click Activate.
****
Check if job has started
****
isql_pdps
1> use pdps
2> go

1> select dprId,completionState from PlDataProcessingRequest
2> go
If your DPR is marked STARTED, th job has started.
If marked NULL, it has not.
If marked CQ_HOLD....check if your granules have a UR in the PDPS database.
1> select dataTypeId,universalReference from PlDataGranule where dataTypeId
like "CER%"
2> select dataTypeId,universalReference from PlDataGranule where dataTypeId
like "AM1%"
3> go
If all of your granules do not have UR of the form UR:10:DsShESDTUR...,
 go to "Delete a DPR and start over" below.
****
Cleanup and bounce PDPS

****
See howto_start_and_kill_servers_drop4.txt, use "EcPl"
as argument
****
Delete a DPR and start over
****
Do this if (for example) there is no UR for your dynamic granules
in PlDataGranule.
1> update PlDataProcessingRequest set completionState="NULL" where
dprId="CPGE1#1#010613970000"
1> delete PlPlans where planId like "CPGE1%"
Bring up Planning Workbench again
EcPlAllStart $MODE 1
Click DPRList tab
Click Production Request arrow and selct your PR.
Click on your DPR in the big window.
Click the Edit menu, select Delete.
Click the PRList tab. Your PR should be gone.
Go back to "Enter a Production Request using PR Editor" above.
****
Run in AutoSys
****
Once you have a plan scheduled:
rlogin illiad -l cmts1
illiad:> /usr/ecs/TS1/CUSTOM/utilities/EcDpPrAutosysStart TS1 TAL
Click JobScape
Scroll to your PGE
Buttons end in this character:
A -- Allocate
S -- Stage
P -- Preprocess (create runtime PCF)
E -- Execute
p -- Postprocess
I -- Insert to Data Server
D -- Destage (Insert output to Dsrv)

Look for runtime files under
/usr/ecs/TS1/CUSTOM/pdps/lasher/data/DpPrRm/lasher_disk/<pgeId>
If AutoSys turns red on Stage step:
look at file
/usr/ecs/RCCCO/CUSTOM/logs/<pgeId....>

****
Restart in AutoSys

****
First clean out log files from /usr/ecs/RCCCO/CUSTOM/logs .
In JobScape, click on the step that failed (in red).
Click Job Console.
Click Force Start Job.
Step changes to green.
##

11.18.4.12 howto_insert_dpprfile_entries _mpge1..

```
insert DpPrFile values
("MI1B2T#00101071996143741AssociatedSensorShortNameDA00000.met","lasher",
"UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[MDC:DSSDSRV]:18:SC:MI1B2T.001:12
33",
"/usr/ecs/OPS/CUSTOM/pdps/lasher/data//DpPrRm/lasher_disk",
10,0.032658,0,0,0,0,"08/31/1998  19:00:00",1200)

insert DpPrFile values
("MI1B2T#00101071996143741AssociatedSensorShortNameDA00000","lasher",
"UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[MDC:DSSDSRV]:18:SC:MI1B2T.001:12
33",
"/usr/ecs/OPS/CUSTOM/pdps/lasher/data//DpPrRm/lasher_disk",
10,516.909552,0,0,1,1,"08/31/1998  19:00:00",1201)

insert DpPrGranuleLocation
values("MI1B2T#00101071996143741AssociatedSensorShortNameDA0","lasher",2,NULL)

insert DpPrFile values
("MI1B2T#00101071996143741AssociatedSensorShortNameDA00000.met","lasher",
"UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[MDC:DSSDSRV]:18:SC:MI1B2T.001:12
33",
"/usr/ecs/OPS/CUSTOM/pdps/lasher/data//DpPrRm/lasher_disk",
10,0.032658,0,0,0,0,"08/31/1998  19:00:00",1200)

insert DpPrFile values
("MI1B2T#00101071996143741AssociatedSensorShortNameDA00000","lasher",
"UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[MDC:DSSDSRV]:18:SC:MI1B2T.001:12
33",
"/usr/ecs/OPS/CUSTOM/pdps/lasher/data//DpPrRm/lasher_disk",
10,516.909552,0,0,1,1,"08/31/1998  19:00:00",1201)

insert DpPrGranuleLocation
values("MI1B2T#00101071996143741AssociatedSensorShortNameDA0","lasher",2,NULL)
```
****

11.18.4.13 howto_run_from_cmdline_in_runtime_dir_drop4pL

```
lasher
cd /usr/ecs/OPS/pdps/DpPrRm/lasher_disk/MPGE8C#2.0a/MPGE8C_lasher
csh% sh
. /usr/ecs/OPS/CUSTOM/bin/DPS/auto.profile
. MPGE8C.Profile
grep PGS_PC_Shell.sh MPGE8C.Log
# Paste string after "COMMAND="
PGS_PC_Shell.sh
/usr/ecs/OPS/CUSTOM/pdps/lasher/data//DpPrRm/lasher_disk/MPGE8C#2.0a/PCS_main
1010
/usr/ecs/OPS/CUSTOM/pdps/lasher/data//DpPrRm/lasher_disk/MPGE8C#2.0a/MPGE8C_lasher
/MPGE8C.Pcf
50 -v -p
*****************************************************************************
```

11.18.4.14 howto_increase_stacksize_mpge8

Edy of MISR ssays that PGE 8C will coredump if the stacksize is
not increased to "unlimited".

```
lasher:>limit
cputime         unlimited
filesize        unlimited
datasize        unlimited
stacksize       65536 kbytes
coredumpsize    unlimited
memoryuse       1019024 kbytes
descriptors     200
vmemoryuse      unlimited
lasher:>limit stacksize unlimited
lasher:>limit
cputime         unlimited
filesize        unlimited
datasize        unlimited
stacksize       524288 kbytes
coredumpsize    unlimited
memoryuse       1019024 kbytes
descriptors     200
vmemoryuse      unlimited
datasize        unlimited
*****************************************************************************
```

11.18.4.15 howto_insert_test_metadata_only_in_sdsrv_drop4pL

rlogin texas

cd /usr/ecs/OPS/CUSTOM/utilities
setenv MODE OPS
source EcCoEnvCsh
/usr/ecs/OPS/CUSTOM/bin/DSS/insert_test ConfigFile
/usr/ecs/OPS/CUSTOM/cfg/EcDsScienceDataServerClient.CFG ecs_mode OPS
Respond to prompts:
your_short_name_here
1
your_metadata_filename_here
your_dummy_data_filename_here
y
NOTE: your_dummy_data_filename cannot be the same filename as
your_metadata_filename
[example for MISR]
Respond to prompts:
MISANCGM
/usr/ecs/OPS/CUSTOM/ssit/lasher/data/MISR/PGE1/2.0/ANC/
/usr/ecs/OPS/CUSTOM/ssit/lasher/data/MISR/PGE1/2.0/pdps/dum
y
If a UR is returned, test metadata Insert was successful, and metadata is
valid.
If not, check the string beginning "--CmdResults[" for error messages.
Edit .met file and retry.

Other tests
MISL0CF
/usr/ecs/OPS/CUSTOM/ssit/lasher/data/MISR/PGE1/2.0/fullswath/STP_CF_NEW_NOMINAL/
L0SIM/MISL0CF.met
/usr/ecs/OPS/CUSTOM/ssit/lasher/data/MISR/PGE1/2.0/fullswath/STP_CF_NEW_NOMINAL/
L0SIM/L0sim.cf.new.nominal.00.EDS
n
/usr/ecs/OPS/CUSTOM/ssit/lasher/data/MISR/PGE1/2.0/pdps/dum
y
MIANPP
/usr/ecs/OPS/CUSTOM/ssit/lasher/data/MISR/PGE1/2.0/fullswath/STP_CF_NEW_NOMINAL/
PP/cf.pp.met
/usr/ecs/OPS/CUSTOM/ssit/lasher/data/MISR/PGE1/2.0/fullswath/STP_CF_NEW_NOMINAL/
PP/cf.pp.met.0
y
MIRFOI
/usr/ecs/OPS/CUSTOM/ssit/lasher/data/MISR/PGE1/2.0/fullswath/STP_CF_NEW_NOMINAL/
ROI/cf.roi.met
/usr/ecs/OPS/CUSTOM/ssit/lasher/data/MISR/PGE1/2.0/fullswath/STP_CF_NEW_NOMINAL/
ROI/cf.roi.met.0

y
MI1B1
/usr/ecs/OPS/CUSTOM/ssit/lasher/data/MISR/PGE1/2.0/pdps/met0/
/usr/ecs/OPS/CUSTOM/ssit/lasher/data/MISR/PGE1/2.0/pdps/met0/dum
y
MI1B1
/usr/ecs/OPS/CUSTOM/ssit/lasher/data/MISR/PGE1/2.0/pdps/met/
/usr/ecs/OPS/CUSTOM/ssit/lasher/data/MISR/PGE1/2.0/pdps/met/dum
y
****

11.18.4.16 howto_delete_granules_frm_sdsrv_drop4pL

****
Delete granules from Sdsrv
****
rlogin texas -l cmops
To delete all granules for a given ESDT from the Data Server database:

MAKE SURE YOU USE THE RIGHT MODE!
else people may get mad at you
setenv DSQUERY journey_sqs222_srvr
setenv DBNAME relbdss
setenv DBUSERNAME sdsrv_role
setenv DBPASSWD popcorn11
setenv MODE OPS
/usr/ecs/${MODE}/CUSTOM/dbms/DSS/DsDbCleanGranules your_short_name_here 1
****
Delete granules from Stmgt
****
Once you have done this, there is no way for the system
to ever retrieve these granules from Stmgt again.
Therefore you can delete them from Stmgt:
rlogin raven -l cmops
cd /dss_amass/OPS
WARNING: BE VERY CAREFUL USING THIS COMMAND
If you type it incorrectly, you could delete *everyone's* granules!
rm *your_short_name_here*
****
Delete granules from Pdps
****
See howto_delete_single_file_granule_in_pdps.txt
****

11.18.4.17 howto_run_in_pdps_drop4pL

```
rlogin odyssey -l cmops
dce_login awhitele awhitele
setenv MODE OPS
cd /usr/ecs/${MODE}/CUSTOM/utilities
source EcCoEnvCsh
setenv DISPLAY ncdso7:0.0
setenv DISPLAY ncdso32:0.0
setenv DISPLAY ncdso8:0.0
setenv DISPLAY ncdso30:0.0
setenv DISPLAY ncdp11:0.0
*******************************************************************************
Enter a Production Request using PR Editor
*******************************************************************************
script your_scriptname_here
EcPlPRE_IFStart $MODE
Click PREdit tab
Select your PGE, click OK, wait 10-15 seconds
Enter start & stop time of your input data.
Click File menu, then Save As, type in a name to save your DPR.
After 15-30 seconds, a dialog box pops up.
If it says "1 DPR generated", you're in business!
If it says "0 DPRs generated", you're in trouble. Check the PREditor logs.
Exit PR Editor.
*******************************************************************************
If PR fails to explode
*******************************************************************************
Check error messages on screen.
After making fixes, delete entries from PlDataGranuleShort
for your PGE.
For example
isql -UpdpsUsers -Pwelcome -Silliad_srvr
1> use pdps_OPS
2> go
Rerun PREditor.
*******************************************************************************
Bring up Planning Workbench
*******************************************************************************
cd /usr/ecs/OPS/CUSTOM/utilities
EcPlAllStart $MODE 1
Wait 10-15 seconds for it to come up.
If you get a message like
Process EcPlSubMgr is already running in mode OPS with PID ...
then check if someone else is running Planning Workbench.
```

(Only one copy may run at a time.)
If not, do
EcPlSlayAll $MODE
then retry EcPlStartAll.
****
Schedule a Production Request
****
Under FILE click NEW.
Type in a name for your Plan.
Click your PR in the Production Requests Unscheduled window.
Click Schedule.
Click Save in the dialog box.
Click Activate.
****
Check if job has started
****
isql_pdps
1> use pdps
2> go
1> select dprId,completionState from PlDataProcessingRequest
2> go
If your DPR is marked STARTED, th job has started.
If marked NULL, it has not.
If marked CQ_HOLD....check if your granules have a UR in the PDPS database.
1> select dataTypeId,universalReference from PlDataGranule where dataTypeId
like "CER%"
2> select dataTypeId,universalReference from PlDataGranule where dataTypeId
like "AM1%"
3> go
If all of your granules do not have UR of the form UR:10:DsShESDTUR...,
go to "Delete a DPR and start over" below.
****
Cleanup and bounce PDPS
****

11.18.4.18 howto_start_and_kill_servers_drop4pL

use "EcPl" as argument
****
Delete a DPR and start over
****
Do this if (for example) there is no UR for your dynamic granules
in PlDataGranule.
1> update PlDataProcessingRequest set completionState="NULL" where
dprId="CPGE1#1#010613970000"

1> delete PlPlans where planId like "CPGE1%"

Bring up Planning Workbench again
EcPlAllStart $MODE 1
Click DPRList tab
Click Production Request arrow and selct your PR.
Click on your DPR in the big window.
Click the Edit menu, select Delete.
Click the PRList tab. Your PR should be gone.
Go back to "Enter a Production Request using PR Editor" above.
Alternative way to delete a DPR:
/home/opscm/pdpsScript/EcPlResetDpr.sh $MODE <dprId>
Now use PREditor to delete PRs as above.
To delete a PGE from the database:
/home/opscm/pdpsScript/EcPlResetPge.sh $MODE <pgeId>
****
Run in AutoSys
****
Once you have a plan scheduled:
rlogin illiad -l cmops
illiad:> /usr/ecs/OPS/CUSTOM/utilities/EcDpPrAutosysStart OPS TAL
Click JobScape
Scroll to your PGE
Buttons end in this character:
A -- Allocate
S -- Stage
P -- Preprocess (create runtime PCF)
E -- Execute
p -- Postprocess
I -- Insert to Data Server
D -- Destage (Insert output to Dsrv)
Look for runtime files under
/usr/ecs/OPS/CUSTOM/pdps/lasher/data/DpPrRm/lasher_disk/<pgeId>
If AutoSys turns red on Stage step:
look at file
/usr/ecs/RCCCO/CUSTOM/logs/<pgeId....>
****
Restart in AutoSys
****

First clean out log files from /usr/ecs/RCCCO/CUSTOM/logs .
In JobScape, click on the step that failed (in red).
Click Job Console.
Click Force Start Job.

Step changes to green.

11.18.4.19 Procedures to bring cdsbrowser up

telnet to msse5hp
#login in as yourself
login: rthamby
password: "yourpasswd"
cd /opt/dce/bin/
setenv DISPLAY
cdsbrowser &
#this would bring the cdsbrowser GUI
click on security
and dce login
login: awhitele
password: awhitele
next click on subsys icon,,,
then subsys/ecs
then subsys/ecs/server
now pick a server
****

dbbrowser:

telnet to texas
login: cmops
password: opsu$er
cd dbr
source dx.csh
%dbr
****

11.18.4.20 howto_insert static files

```
##########################################################################
# Make sure STMGT and SDSRV are up
##########################################################################
# Use either ECS Assist on texas *and* raven,
# Alternatively, use ps:
texas:> ps -ef | grep EcDsScienceDataServer
   sdsrv 24410     1  0 10:17:57 pts/3    0:33 /usr/ecs//OPS/CUSTOM/bin/DSS/
EcDsScienceDataServer ConfigFile /usr/ecs/OPS/CUSTOM/
raven:> ps -ef | grep  Server | grep OPS | grep ConfigFile
stmgt  3786     1  0   Dec 30 ?      10:39 /usr/ecs/OPS/CUSTOM/bin/DSS/
EcDsStStagingDiskServer ConfigFile /usr/ecs/OPS/CUS
stmgt  3803     1  0   Dec 30 ?      10:31 /usr/ecs/OPS/CUSTOM/bin/DSS/
```

```
EcDsStStagingMonitorServer ConfigFile /usr/ecs/OPS/
stmgt  3770     1  0   Dec 30 ?      16:18 /usr/ecs/OPS/CUSTOM/bin/DSS/
EcDsStArchiveServer ConfigFile /usr/ecs/OPS/CUSTOM/
stmgt  3815     1  0   Dec 30 ?      10:59 /usr/ecs/OPS/CUSTOM/bin/DSS/
EcDsStFtpDisServer ConfigFile /usr/ecs/OPS/CUSTOM/
# If any of these are not present, that server is down.
# You must arrange to have it up before continuing the install
##############################################################################
# Setup SSIT environment, login to DCE
##############################################################################
mini-daac% rlogin calahans
calahans% source /usr/ecs/OPS/CUSTOM/utilities/.buildrc
calahans% dce_login
Enter Principal Name: awhitele
Enter Password: awhitele
##############################################################################
```

11.18.4.21 Insert all static files needed for CERES SS1

```
##############################################################################
calahans% cd /usr/ecs/TS1/CUSTOM/ssit/lasher/data/CERES/ecs/ss1/pdps
calahans% Ceres1StaticInsert.sh
##############################################################################
# Insert a single static file
******************************************************************************
```

11.18.4.22 Insert Test Dynamic File (multi-file) using SSIT Manager

The Insert Test Dynamic File program
inserts test dynamic input files to the Data Server,
for use both during SSIT and in production.
Dynamic Files are files which change at each
instance of PGE processing; for example, Level 0 data files.
(An example of a static file is a calibration file.)
This tool performs a function that is normally done by Ingest. It allows the
user to insert dyanmic files to the Data Server for testing purposes.
Normally, Ingest will take in such files and insert them as part of its normal
processing.
Preconditions to running the Insert Test Dynamic File program
Before this program is run, the following must have occurred:
An ESDT for this data type must have been created in the Data Server.
A PGE that uses the Dynamic File(s) ESDT must have been Registered to
PDPS via the <A HREF="DpAtMgr_sciupdate.html">SSIT Science Metadata Update tool.
An ASCII Metadata File (.met) for this instance of the ESDT must have
been created in the format expected by the Data Server for Insert.
Both the test dynamic data file(s) in question and its corresponding

ASCII Metadata file must be accessible to the local machine.
The directory where the dynamic data file(s) and the ASCII Metadata File
exist must be cross mounted to the Data Server machines.
Running the Insert Test Dynamic File program from the SSIT
Manager
The Insert Test Dynamic File program is invoked from the SSIT Manager under
the TOOLS menu item Data Server,
sub-menu Insert Test.
The user prompts and explinations follow. Note that if there is a default
value for an entry (in most cases this will only occur if you run the
program more than once) it will appear at the end of the prompt line.
PGE Test Dynamic Input File Insertion
Configuration filename? (enter for default: ../../cfg/EcDpAtInsertTestFile.CFG)
In most cases hitting enter (for the default) is fine.
If not, enter the correct configuration filename including full path
ECS Mode of operations?
This is the mode (i.e. OPS, TS1) of operations.
In most cases this will be TS1
ESDT short name for the file(s) to Insert?
This is the ESDT short name (max 8 characters) for the data file(s).
ESDT Version for the file(s) to insert?
This is the ESDT version (an integer) for the data file(s).
Is there is more than one data file to this Dynamic Granule (Y = Yes, N = No)?
This indicates to the program if there is more than one file associated
with the dynamic input.
Most dynamic inputs are a single data file and a corresponding ASCII Metadata
File. Other dynamic inputs consist of more than one data file and a single
ASCII Metadata file (there is always only 1 ASCII Metadata File).
If there is only one data file enter N (or just hit enter for the default).
For multiple data files enter Y
If it is NOT a Dynamic Multi-File Granule (there is a single data file)
the following prompts will need to be answered
Single Filename to Insert (including FULL path)?
This is the name of the data file to insert to the Data Server.
Include the full path to the file so that it can be found by the tool
Associated ASCII Metadata Filename to Insert (including FULL path)?
<I>This is the name of the ASCII Metadata File (.met) associated with the data file.
Full path to the file must be included so that it can be found by the tool.
It can be created by using the <A HREF="DpAtMgr_getMCF.html">Get MCF</A> tool to get a
Metadata Control File which specifies the fields and type of data that is required for the ASCII
Metadata file
If it IS a Dynamic Multi-File Granule (more than 1 data file) the
following prompts will need to be answered
Directory where all data files and .met file exist (FULL path)?

This is the directory location where all the data files and the ASCII
Metadata file are located.
For Multi-File Dynamics all files and Metadata must reside in the same
directory
Name of MFG file (enter to end list)?
This is the name of one of the data files for a Dynamic Multi-File
Granule.
Enter a file at the prompt and the prompt will then be repeated, allowing
for the next file to be named. When all data files have been entered, just
hit the enter key at the next prompt (thus entering nothing).
Associated ASCII Metadata Filename to Insert?
This is the name of the ASCII Metadata File (.met) associated with the data file.
No path is needed because it is assumed that the file resides in the directory
specified above.
It can be created by using the <A HREF="DpAtMgr_getMCF.html">Get MCF</A> tool to get a
Metadata Control File which specifies the fields and type of data that is required for the ASCII
Metadata file
Any success or error messages will then be displayed followed by:
Hit Enter to run again, 'q <Enter>' to quit:
This allows the user to enter another Dyanmic File (just hit enter) or
quit the program (q).
Running the Insert Test Dynamic File program from the
command line
There are two methods to run the program from the UNIX command line:
EcDpAtInsertTest
This is the same shell script that the SSIT Manager GUI kicks off and
quieries the user for all neccessary data (as specified in the section above)
for the program that will actually insert the test data file.
Or the program itself can be run directly as follows:
EcDpAtInsertTestFile ConfigFile EcDpAtInsertTestFile.CFG ecs_mode MODE
SHORT_NAME VERSION {FILENAME METADATA_FILE} {DIRECTORY FILENAME ...
FILENAME METADATA_FILE}
The arguments are described as follows:
EcDpAtInsertTestFile - name of the program. Substitute full path if
running from a different location.
ConfigFile - delimiter that tells the program that the configuration file
is the next item in the input list. Note that it must follow the program name.
EcDpAtInsertTestFile.CFG - name of the configuation file. Again, substitute full path if it is in
another directory.
ecs_mode - delimiter that tells the program that the Mode
is the next item in the input list. Note that it must follow the
configuration file name.
MODE substitute the Mode of operations (i.e. TS1).

SHORT_NAME the ESDT short name for the file(s) to insert. Must already be installed at the Data Server.
VERSIONversion of the ESDT. An integer value.
{FILENAME METADATA_FILE- the pairing of inputs for a single file dynamic input.
FILENAME the name of the file (including Full path) to insert for single file dynamic granules.
METADATA_FILE- the name of the ASCII Metadata file (including Full path).
{DIRECTORY FILENAME ... FILENAME METADATA_FILE- the pairing of inputs for a multi-file dynamic granule.
DIRECTORY- the directory where all the files and the associated
ASCII Metadata File is located.
FILENAME- the name of the file (assumed to be in the directory specified above) to insert for a multi-file dynamic granule. Repeat with as many filenames as needed to complete the granule. Separate with spaces.
METADATA_FILE- the name of the ASCII Metadata file (assumed to be in the directory specified above).
After this program runs successfully, the specified file(s) have been stored
at the Data Server and can be acquired by PDPS when executing PGEs. Also,
any subscriptions made on the ESDT of the insert file(s) will trigger, and
could (via Subscription Manager) cause PGE waiting on this data to start execution.
Problems?
If you get an error message like
Config file EcDpAtInserTestFile.CFG is unreadable
then the name or path of the Process Framework configuration file
for this program is invalid.
Change the name and path of the .CFG file to point to
a valid Process Framework configuration file.
If you get an error message like
Could not get DSS from Adv.
then there is a problem getting the location for the Data Server from
Advertising (IOS). This is either because the Advertising server is down or
the ESDT has not been properly installed. Check that the Advertising server
is up and then verify the installation of the ESDT for the insert.
If you get an error message like
Submit to the Data Server failed
then it is likely that the you did not specify the full path for either
the file or the metadata file, or the directory in which the files reside
is not NFS mounted to the DSS machine. Try again with the full path name, or
find a directory on the local machine that also appears on the DSS machine.
****
****

11.18.4.23 howto_insert_test_metadata_only_in_sdsrv_drop4/MISR

rlogin texas
/usr/ecs/TS1/CUSTOM/bin/DSS/insert_test ConfigFile

/usr/ecs/TS1/CUSTOM/cfg/EcDsScienceDataServerClient.CFG ecs_mode TS1
Respond to prompts:
your_short_name_here
your_metadata_filename_here
your_dummy_data_filename_here
y
NOTE: your_dummy_data_filename cannot be the same filename as
your_metadata_filename
[example for MISR]
Respond to prompts:
MISANCGM
/usr/ecs/TS1/CUSTOM/ssit/lasher/data/MISR/PGE1/2.0/ANC/
/usr/ecs/TS1/CUSTOM/ssit/lasher/data/MISR/PGE1/2.0/pdps/dum
y
If a UR is returned, test metadata Insert was successful, and metadata is
valid.
If not, check the string beginning "--CmdResults[" for error messages.
Edit .met file and retry.
Other tests
MISL0CF
/usr/ecs/TS1/CUSTOM/ssit/lasher/data/MISR/PGE1/2.0/fullswath/STP_CF_NEW_NOMINAL/
L0SIM/MISL0CF.met
/usr/ecs/TS1/CUSTOM/ssit/lasher/data/MISR/PGE1/2.0/fullswath/STP_CF_NEW_NOMINAL/
L0SIM/L0sim.cf.new.nominal.00.EDS
n
/usr/ecs/TS1/CUSTOM/ssit/lasher/data/MISR/PGE1/2.0/pdps/dum
y
MIANPP
/usr/ecs/TS1/CUSTOM/ssit/lasher/data/MISR/PGE1/2.0/fullswath/STP_CF_NEW_NOMINAL/
PP/cf.pp.met
/usr/ecs/TS1/CUSTOM/ssit/lasher/data/MISR/PGE1/2.0/fullswath/STP_CF_NEW_NOMINAL/
PP/cf.pp.met.0
y
MIRFOI
/usr/ecs/TS1/CUSTOM/ssit/lasher/data/MISR/PGE1/2.0/fullswath/STP_CF_NEW_NOMINAL/
ROI/cf.roi.met
/usr/ecs/TS1/CUSTOM/ssit/lasher/data/MISR/PGE1/2.0/fullswath/STP_CF_NEW_NOMINAL/
ROI/cf.roi.met.0
y
****

11.18.4.24 howto_insert test dynamic files

##
Make sure STMGT and SDSRV are up

```
##########################################################################
# Use either ECS Assist on texas *and* raven,
# Alternatively, use ps:
texas:> ps -ef | grep EcDsScienceDataServerdsrv 24410     1  0 10:17:57 pts/3    0:33
/usr/ecs//TS1/CUSTOM/bin/DSS/EcDsScienceDataServer ConfigFile /usr/ecs//TS1/CUS
raven:> ps -ef | grep  Server | grep ConfigFile
   stmgt  3786     1  0   Dec 30 ?      10:39 /usr/ecs/TS1/CUSTOM/bin/DSS/EcD
sStStagingDiskServer ConfigFile /usr/ecs/TS1/CUS
   stmgt  3803     1  0   Dec 30 ?      10:31 /usr/ecs/TS1/CUSTOM/bin/DSS/EcD
sStStagingMonitorServer ConfigFile /usr/ecs/TS1/
   stmgt  3770     1  0   Dec 30 ?      16:18 /usr/ecs/TS1/CUSTOM/bin/DSS/EcD
sStArchiveServer ConfigFile /usr/ecs/TS1/CUSTOM/
   stmgt  3815     1  0   Dec 30 ?      10:59 /usr/ecs/TS1/CUSTOM/bin/DSS/EcD
sStFtpDisServer ConfigFile /usr/ecs/TS1/CUSTOM/c
# If any of these are not present, that server is down.
# You must arrange to have it up before continuing the install
##########################################################################
# Setup SSIT environment, login to DCE
##########################################################################
mini-daac% rlogin calahans
calahans% source /usr/ecs/TS1/CUSTOM/bin/DPS/.buildrc
calahans% dce_login
Enter Principal Name: awhitele
Enter Password: awhitele
##########################################################################
# Insert the test file
##########################################################################
cd /usr/ecs/TS1/CUSTOM/ssit/lasher/data/CERES/ecs/ss1/pdps
# Bring up InsertStatic window -- leave it up until you
# are completely done inserting all static files –
**************************************************************************
```

11.18.4.25 howto_insert_multifile_granules.scr

```
rlogin calahans -l sdsrv
dce_login
cd /home/mtheobal/tmp
source ld_path
Script started on Thu Jan 08 10:05:37 1998
sourcing /usr/ecs/TS1/CUSTOM/utilities/libpath.csh
/usr/ecs/TS1/CUSTOM/utilities/libpath.csh: No such file or directory
calahans{sdsrv}1: cd /home/mtheobal/tmp
calahans(sdsrv):/home/mtheobal/tmp<2>source ld_path
calahans(sdsrv):/home/mtheobal/tmp<3>subacquire ConfigFile
EcDsScienceDataServerClient.
```

```
CFG ecs_mode TS1
Warning: could not open message catalog "oodce.cat"
SDSRV cell is not in configfile, using local cell
Client G1 Path: /.:/subsys/ecs/TS1/EcDsScienceDataServerG1
1       Insert Data (w/multiple file)
2       Test Selective Acquirer
3       Test Subsetted Acquirer
4       Search for Granule
5       Exit
Please make selection=> 1
Executing insert .....
trying to connect to /.:/subsys/ecs/TS1/EcDsScienceDataServerG1
Enter data type=> ER00AF       ERDIAF
Enter data metafile name(full path)=>
/usr/ecs/TS1/CUSTOM/ssit/lasher/data/CERES/ecs/ss1/instrument/data/input/TRMM_G056_LZ
_1997-06-12T01-07-34Z_V01.DAT1.met
Enter number of datafile names=> 2
Inserting MetadataFile
[/usr/ecs/TS1/CUSTOM/ssit/lasher/data/CERES/ecs/ss1/instrument/data/input/TRMM_G056_L
Z_1997-06-12T01-07-34Z_V01.DAT1.met]
Enter datafile #1 name(full path)=>
/usr/ecs/TS1/CUSTOM/ssit/lasher/data
/CERES/ecs/ss1/instrument/data/input/TRMM_G056_LZ_1997-06-12T01-07-34Z_V01.DAT1
Inserting
DataFile #1
[/usr/ecs/TS1/CUSTOM/ssit/lasher/data/CERES/ecs/ss1/instr
ument/data/input/TRMM_G056_LZ_1997-06-12T01-07-34Z_V01.DAT1]
Enter datafile #2 name(full path)=>
/usr/ecs/TS1/CUSTOM/ssit/lasher/data
/CERES/ecs/ss1/instrument/data/input/TRMM_G056_LZ_1997-06-13T04-19-59Z_V01.DAT1
Inserting
DataFile #2
  [/usr/ecs/TS1/CUSTOM/ssit/lasher/data/CERES/ecs/ss1/instr
ument/data/input/TRMM_G056_LZ_1997-06-13T04-19-59Z_V01.DAT1]
Insert science data only …Success.
UR(UR:10:DsShESDTUR:UR:15:DsShSciServerUR:7:dssdsrv:20:Science:ERDIAF:1694)
1       Insert Data (w/multiple file)
2       Test Selective Acquirer
3       Test Subsetted Acquirer
          4   Search for Granule
          5   Exit
*****************************************************************************
```

11.18.4.26 Insert Static File (multi_file) using SSIT Manager

The Insert Static File program
inserts static input file(s) to the Data Server,
for use both during SSIT and in production.
Static Files are files which rarely change between
instances of PGE processing; for example, calibration files.
(An example of a dynamic file is a Level 0 data file.)
Preconditions to running the Insert Static File program
Before this program is run, the following must have occurred:
An ESDT for this data type must have been created in the Data Server.
A PGE that uses the Static File must have been Registered to PDPS via
the <A HREF="DpAtMgr_sciupdate.html">SSIT Science Metadata Update tool.
An ASCII Metadata File (.met) for this ESDT must have been created,
in the format expected by the Data Server for Insert.
Both the static data file(s) in question and its corresponding ascii
Metadata file must be accessible to the local machine.
The directory where the static data file(s) and the ASCII Metadata File
exist must be cross mounted to the Data Server machines
Running the Insert Test Dynamic File program from the SSIT Manager
The Insert Static File program is invoked from the SSIT Manager under
the TOOLS menu item Data Server,
sub-menu Insert Static.
The user prompts and explinations follow. Note that if there is a default
value for an entry (in most cases this will only occur if you run the
program more than once) it will appear at the end of the prompt line.

PGE Static Input File Insertion Script

Configuration filename? (enter for default: ../../cfg/EcDpAtInsertStaticFile.CFG)
In most cases hitting enter (for the default) is fine.
If not, enter the correct configuration filename including full path
ECS Mode of operations?
This is the mode (i.e. OPS, TS1) of operations.
In most cases this will be <B>TS1.
ESDT short name for the file(s) to Insert?
This is the ESDT short name (max 8 characters) for the data file(s)
ESDT Version for the file(s) to insert?
This is the ESDT version (an integer) for the data file(s).
Science Group for Static file (one of {C, L, D, O} followed by a 3 digit number)?
This is the Science Group for this Static File(s).
It is defined in the <A HREF="DpAtMgr_PGEMetadata.html">PGE Metadata ODL File for
each PGE that uses this static file.
C = Coefficent File(s), L = Lookup Table/file(s), D = Database File(s), O = Other Files. Note
that using the wrong letter for the file(s) (say L for a Coefficent File) will not result in an error.
Is there is more than one data file to this Static (Y = Yes, N = No)?

This indicates to the program if there is more than one file associated
with the static input.
Most static inputs are a single data file and a corresponding ASCII Metadata
File. Other static inputs consist of more than one data file and a single
ASCII Metadata file (there is always only 1 ASCII Metadata File).
If there is only one data file enter N (or just hit enter for the default).
For multiple data files enter Y.
If it is NOT a Static Multi-File Granule (there is a single data file)
the following prompts will need to be answered:
Single Filename to Insert (including FULL path)?
This is the name of the data file to insert to the Data Server.
Include the full path to the file so that it can be found by the tool.
Associated ASCII Metadata Filename to Insert (including FULL path)?
This is the name of the ASCII Metadata File (.met) associated with the data file.
Full path to the file must be included so that it can be found by the tool.
It can be created by using the <A HREF="DpAtMgr_getMCF.html">Get MCF</A> tool to get a
Metadata Control File which specifies the fields and type of data that is required for the ASCII
Metadata file.
If it IS a Static Multi-File Granule (more than 1 data file) the following prompts will need to be
answered
Directory where all data files and .met file exist (FULL path)?
This is the directory location where all the data files and the ASCII
Metadata file are located.
For Multi-File Statics all files and Metadata must reside in the same
directory.
Name of MFG file (enter to end list)?
This is the name of one of the data files for a Static Multi-File
Granule.
Enter a file at the prompt and the prompt will then be repeated, allowing
for the next file to be named. When all data files have been entered, just
hit the enter key at the next prompt (thus entering nothing).
Associated ASCII Metadata Filename to Insert?
This is the name of the ASCII Metadata File (.met) associated with the data file.
No path is needed because it is assumed that the file resides in the directory
specified above.
It can be created by using the <A HREF="DpAtMgr_getMCF.html">Get MCF</A> tool to get a
Metadata Control File which specifies the fields and type of data that is required for the ASCII
Metadata file
Any success or error messages will then be displayed followed by:
Hit Enter to run again, 'q <Enter>' to quit:
This allows the user to enter another Static File (just hit enter) or
quit the program (q).
Running the Insert Test Dynamic File program from the
command line

There are two methods to run the program from the UNIX command line:
EcDpAtInsertStatic
This is the same shell script that the SSIT Mgr GUI kicks off and
quieries the user for all neccessary data (as specified inthe section above)
for the program that will actually insert the static file.
Or the program itself can be run directly as follows:
EcDpAtInsertStaticFile ConfigFile EcDpAtInsertStaticFile.CFG ecs_mode MODE
SHORT_NAME VERSION SCIENCE_GROUP {FILENAME METADATA_FILE}
{DIRECTORY FILENAME ... FILENAME METADATA_FILE}
The arguments are described as follows:
EcDpAtInsertStaticFile - name of the program. Substitute full path if
running from a different location.
ConfigFile - delimiter that tells the program that the configuration file
is the next item in the input list. Note that it must follow the program name.
EcDpAtInsertStaticFile.CFG - name of the configuation file. Again, substitute full path if it is in another directory.
ecs_mode - delimiter that tells the program that the Mode
is the next item in the input list. Note that it must follow the
configuration file name.
MODE- substitute the Mode of operations (i.e. TS1).
SHORT_NAME- the ESDT short name for the file(s) to insert. Must already be installed at the Data Server.
VERSION- version of the ESDT. An integer value.
SCIENCE_GROUP- the grouping that defines this input to PGEs
that use it. It is one of {C, L, D, O} followed by a 3-digit number.
{FILENAME METADATA_FILE- the pairing of inputs for a single file static
input.
FILENAME- the name of the file (including <B>Full</B> path) to insert for single file static granules.
METADATA_FILE</EM> - the name of the ascii Metadata file (including Full path).
{DIRECTORY FILENAME ... FILENAME METADATA_FILE}- the pairing of inputs for a multi-file static granule.
DIRECTORY- the directory where all the files and the associated
ASCII Metadata File is located.
FILENAME- the name of the file (assumed to be in the directory specified above) to insert for a multi-file static granule. Repeat with as
many filenames as needed to complete the granule. Separate with spaces.
METADATA_FILE - the name of the ASCII Metadata file (assumed to be in the directory specified above).
After this program runs successfully,
the PDPS database has been updated with the Data Server Universal Reference
(UR) of the file inserted.
This UR is read by the Planning and Processing systems at runtime. so that the file can be acquired and used as need during production.

Problems
If you get an error message like
“Config file EcDpAtInsertStaticFile.CFG is unreadable”,
then the name or path of the Process Framework configuration file
for this program is invalid.
Change the name and path of the .CFG file to point to a valid Process Framework configuration file.
If you get an error message like “Could not get DSS from Adv.”
then there is a problem getting the location for the Data Server from
Advertising (IOS). This is either because the Advertising server is down or
the ESDT has not been properly installed. Check that the Advertising server
is up and then verify the installation of the ESDT for the insert.
If you get an error message like “Submit to the Data Server failed”
then it is likely that the you did not specify the full path for either
the file or the metadata file, or the directory in which the files reside
is not NFS mounted to the DSS machine. Try again with the full path name, or
find a directory on the local machine that also appears on the DSS machine.

11.18.4.27 howto_cleanup_dce_on_hp

rlogin msse5hp -l cmops
setenv DISPLAY mojave:0.0
/opt/dce/bin/cdsbrowser &
Login to DCE from Security menu, click DCE Login...

Acquire failure: DDIST

If the AutoSys Staging step fails, and you get message in the
PDPS *.err file like
Data Distribution Submit failed
Then check the EcDsDistributionServer.ALOG on the DDIST
machine (sorccess in the mini-DAAC).
If you see this message
Msg: Exception: <DsStResourceProviderProxy.C:245>
(DsStResourceProviderProxy.C:245)
DCE Rebinding Area Error Priority: 2 Time : 07/29/98 13:24:44
Then you need to cdsping all servers, e.g. using EcsAssist.

Cannot Acquire big files

Very large files cause a core dump in the StagingMonitorServer
on the Stmgt machine, when Acquire is attempted by PDPS
in AutoSys.
Workaround is to copy over the Drop 4P-Landsat executable
for StagingMonitorServer.

You must also manually copy the files from the archive
to the staging disk directory, e.g.
raven:> cp /dss_amass/TS1/SC:MIANPP.001:1282:1.HDF-EOS
/usr/ecs/TS1/CUSTOM/drp/raven/data/staging/user1
(Note that files in the user1 directory are blown away when the
Stmgt servers are cold started -- so it is best to avoid cold
starts if you have to do this workaround.)
At this writing the MIANPP granules (size 945 MB) are
the only ones causing this problem.
The file size limit which causes the core dump is not known;
the largest size file Acquired so far without this
problem is the MIANCAGP granule of 240 MB.
NCR 16425 was written for this, and is fixed in 4P-Landsat.

If execution fails: Switched order of L0 files

If you see a bunch of error messages in the Toolkit
LogStatus file that start with
  PGS_IO_L0_FileVersionInfo():PGSIO_E_L0_MEM_ALLOC:11824
  Error occurred during memory allocation
then the problem is that the L0 construction record
file and the L0 data file are in the wrong order
in the PCF.
This can happen if at Ingest the files were named incorrectly,
since PDPS sorts them in alphabetical order before
placing them in PCF.
To fix it at Ingest, make sure the files have consistent names.
To fix it in PDPS, flip the order of the files in the PCF,
and also change the sequence numbers (last field in PCF
entry) so that sequence number 2 appears first in the PCF.

If execution fails: Bad L0 construction record

If you see a bunch of error messages in the Toolkit
LogStatus file that start with
  PGS_IO_L0_FileVersionInfo():PGSIO_E_L0_BAD_FILEINFO:11830
  corrupt construction record, invalid EDS start time
then the problem is that the construction record was
not done correctly.
This may happen because the delivered construction record
file delivered by MISR needs to be changed in order to go
through Ingest.
Solution is to get the correct construction record file
through Ingest, and rerun the Planning Workbench,

or to copy over the correct file to the PDPS runtime directory
and force start Execution.
POC: Feng Huang

Cleaning up PDPS tables

If the AutoSys job fails for some external reason such
as DDIST or a Stmgt server problem, and one of those
servers needs to be cold started, then you must delete
all relevant entries from the PDPS database table
DpPrRpcID before continuing.
(See "Cleaning up PDPS database after DDIST cold restart"
above.)
It is usually not necessary to also delete the entries
in tables DpPrFile and DpPrGranuleLocation.
The StageState flag in DpPrGranuleLocation is set to 1
if the Acquire failed, so PDPS knows about this.
It is particularly important that you not delete
these tables arbitrarily if you are doing many
DPRs at once (e.g., 9 cameras for PGE 1).
This is because it will screw up the value
of the numberOfUsage flag in DpPrFile, causing
it to go to 0, which will cause Allocate and
Insert to fail.
The only time you might need to delete from these two tables
is if you had a problem with duplicate filenames or something
similar.
Developer POC: Naga Hariharan

Cleanup servers
****
Double click /.:/subsys/ecs/OPS to fold them back,
then click /.:/subsys/ecs/servers.
For each server D machine in the following list:
(huckfinn),
Double click the D entry of the machine.
Double click each D entry under the machine.
If it has no sub-entries, skip to next one.
If it has S sub entries:
Select each OPS MODE server (marked O or S, doesn't matter).
(Select more than one using Shift-Click)
WARNING: DO NOT SELECT OTHER MODES -- PEOPLE WILL GET MAD AT YOU.
Click Right Mouse Button. Select Delet Entry. Click OK.
Do this until no more sub-entries for OPS on this machine.

(Exception: DO not try to delete MsSubAgent servers.
You will not have permission anyway.)

Cold Start Servers

By default, ECS Assist does a warm start on DDIST
(and all servers).
This means it will try to recreate all the requests
that happened before it crashed.
If you kill & restart using ECSAssist, check that
it came up correctly by
sorccess:/usr/ecs/TS1/CUSTOM/logs> grep Monit EcDsDistributionServerDebug.log
Success means you will see the line
  End of StartMonitoring
If you don't, there is trouble.
This often happens if the requests are old.
Do a cold start instead.
COLD START
To do a cold start, which cleans the database out of pending requests:
sorccess:/usr/ecs/TS1/CUSTOM/utilities> EcDsDistributionServerStart TS1
StartTemperature cold
Make sure it came up correctly as above.

11.18.4.28 howto_insert_MISR_multifile_granules

rlogin texas -l **sdsrv**
% **cd /usr/ecs/TS1/CUSTOM/bin/DSS**
% **dsses**
Respond to prompts:
[MDC:DSSDSRV]
2
MISCALAA
2
/home/dps/SSIT/MISR/PGE4/SSTEST1/TestIn/MISCALAA.met
/home/dps/SSIT/MISR/PGE4/SSTEST1/TestIn/E0420350AAAAAAAAAAAAAA97350153021
200.EDS
/home/dps/SSIT/MISR/PGE4/SSTEST1/TestIn/E0420350AAAAAAAAAAAAAA97350153021
201.EDS
<CR>

11.18.4.29 howto_insert_multifile_granules

rlogin calahans -l **sdsrv**
dce_login awhitele awhitele
cd /home/mtheobal/tmp
source ld_path
subacquire ConfigFile EcDsScienceDataServerClient.CFG ecs_mode TS1

[answer prompts for example as follows:]
1
CER00AF
/usr/ecs/TS1/CUSTOM/ssit/lasher/data/CERES/ecs/ss1/instrument/data/input/TRMM_G054_LZ
_1
997-06-12T00-00-02Z_V01.DAT1.met
2
/usr/ecs/TS1/CUSTOM/ssit/lasher/data/CERES/ecs/ss1/instrument/data/input/TRMM_G054_LZ
_1
997-06-12T00-00-02Z_V01.DAT1
/usr/ecs/TS1/CUSTOM/ssit/lasher/data/CERES/ecs/ss1/instrument/data/input/TRMM_G054_LZ
_1
997-06-13T00-00-03Z_V01.DAT1
[Success mesages:]
Insert science data only ...
Success.
UR(UR:10:DsShESDTUR:UR:15:DsShSciServerUR:7:dssdsrv:20:Science:CER00AF:1826)
****

11.18.4.30 howto_lookat_dss_granules

mini-daac% rlogin **texas**
Data server database entry for this granule
texas% isql -UsdsrvApp -Pwelcome -Sjourney_sqs222_srvr
1> use relbdss_TS1
2> go
1> select * from DsMdGranules where ShortName="CER00AF"
2> go
You will see as many entries as there are granules for this ESDT.
Look for one close to the current time, since insert insertTime
is one of the fields
1> select dbID,ShortName,insertTime from DsMdGranules where ShortName="CER00AF"
2> go
1> select
dbID,RangeBeginningDate,RangeBeginningTime,RangeEndingDate,RangeEndingTime fr
om DsMdGranules where ShortName="CER00AF"
2> Go
****

11.18.4.31 howto_install_esdt/CER

(for example, if your ESDTs all contain the string "CER")
/usr/ecs/TS1/CUSTOM/data/ESS/*CER*
Select your ESDT from the list, click OK
Click second File.., then edit Filter so it gets your ESDTs:
(for example, if your ESDTs all contain the string "CER")

```
/usr/ecs/TS1/CUSTOM/lib/ESS/*CER*
Select your ESDT from the list, click OK
Type in:
Archive ID: DRP1_TS1
Click OK
ESDT begins to install
If you get a "Failure to Add Datatype" message,
go to SDSRV logs and examine them
cd /usr/ecs/OPS/CUSTOM/logs
vi EcDsScienceDataServer.ALOG
vi EcDsScienceDataServerDebug.log
# Go to end of log file
:$
# Now look for your ShortName, then read error messages if any
# Make sure you are looking at recent error messages by comparing
# to the current time
##########################################################################
# Check that ESDT is in SDSRV database
##########################################################################
# Invoke ISQL to look at database
texas% isql -UsdsrvApp -Pwelcome -Sjourney_sqs222_srvr
# Get correct Dsrv database
1> use relbdss
2> go
# See your ShortName
1> select ShortName from DsMdCollections where ShortName="<shortname>"
# Substitute your ShortName for <shortname>
2> go
# Your ShortName should appear on the screen
##########################################################################
# Check that ESDT was advertised
##########################################################################
# Open URL http://incagold:4900 with netscape
# Click "Search All"
# Enter ShortName in "Search for" box
# Click "Service entry" box
# Click "Search" button at bottom of page
# You should see a list of items, all for your ShortName
# Make sure Insert Service is one of them, so you can do Inserts
# Alternatively, run a script to see all the services
mini-daac% rlogin incagold -l ios
# passwd Mak!thpn
cd /home/ios/scripts
listESDT.csh <shortname>
```

passwd welcome
You should see a bunch of lines with your ShortName in them
&&&

Same changes as for PGE 4, except leapsec.dat and utcpole.dat
are unchanged from the DAAC versions.
++
End COMMAND LINE RUNS
++
++
PDPS RUNS
++
****
PDPS ODL files
****
A copy of the Landover versions are supplied separately.

P1. Change PERIOD, DURATION, BOUNDARY in AM1EPHMN and AM1ATTN ESDT ODL files
---
AM1EPHMN and AM1ATTN ESDT ODL files must change from PGE 4 to PGE 7,
because PGE 4 test eph/att files are of length 1 day each,
whereas PGE 7 tes eph/att file is only 1 orbit.
This means there is a possible conflict if running both of these PGEs at once
or more accurately, if both are registered in the PDPS
database concurrently.
Correct values for PGE 7 are
PERIOD = "ORBITS=1"
DURATION = "ORBITS=1"
BOUNDARY = "START_OF_ORBIT"
These changes are already done in the supplied PGE 7 ODL files.
Just don't use the ones from PGE 4.
****
End PDPS ODL files
****
Changes to PDPS database
****
none
****
Changes to Toolkit Template PCF used by PDPS
****
This is the file PDPS starts from in constructing the runtime PCF.
It has been copied from the Toolkit; it does not reside in the
Toolkit directory structure.
Its path is defined in PDPS config files on the DPS machine:
EcDpPrEM.CFG
EcDpPrGE.CFG
EcDpPrJobMgmt.CFG

These files reside in directory /usr/ecs/<mode>/CUSTOM/cfg .
For example, on taltos in mode TS1 in the miniDAAC,
DpPrEM_MASTER_PCF =
 /usr/ecs//TS1/CUSTOM/data/DPS/template_PCF//PCF.relB0.template
The DAAC path will be different.
Some changes may be necessary to this file.

P2. Delete Toolkit ephemeris/attitude LIDs

If the PDPS template PCF contains the following lines, delete them:
10501|INSERT_EPHEMERIS_FILES_HERE|||||1
10502|INSERT_ATTITUDE_FILES_HERE|||||1
This is necessary because it causes confusion when a DPR
is created. Too many entries appear in the runtime PCF.

P3. Fix LID for udunits file

11001|udunits.dat|~/database/common/CUC||||1
10302|udunits.dat|~/database/common/CUC||||1
****
Changes to Data Server MCF to be made after stopping execution
****
The MCF staged by Data Server as PGE input must be edited before Execution.
In the miniDAAC this file is located in directory
/usr/ecs/TS1/CUSTOM/pdps/lasher/data/DpPrRm/lasher_disk/MPGE7#2.0/MPGE7#_lasher
where MPGE7#2.0 is the SswId and MPGE7# is the DPR Id.
The DAAC location will be different.

P4. Change CLASS value of AssociatedPlatformInstrumentSensorContainer
attributes

A bug prevents values of attributes specified in the Data Server MCF from being
passed to the output .met file at runtime.
To fix this, edit MIB2GEOP#0.MCF .
For all of the attributes for OBJECT =
AssociatedPlatformInstrumentSensorContainer
with Data_Location = "MCF" (i.e., AssociatedPlatformShortName,
OperationMode, and AssociatedInstrumentShortName)
change the line
CLASS = "M"
to
CLASS = "1" .
NCR ECSed12896 has been written for this problem.
****

END Changes to Data Server MCF to be made after stopping execution
****
Changes to output .met file to be made after stopping Insert
****
The metdata file (".met") output from the PGE must be edited during
a PDPS run, by placing the Insert (I) step On Hold in Autosys.
When all the changes below have been made, Force Restart for this step
in Autosys.

P5. Delete Gring Polygon data

There are 2 problems with the Gring Polygon
a. Data Server rejects duplicate points
b. Data Server rejects data with range in longitude of more than 180 degrees
These issues are being worked internally by ECS.
NCR 14223 was written for this problem.
Workaround for now is to delete the entire
SPATIALDOMAINCONTAINER group which encloses
the Gring Polygon data.
Downstream MISR processing is not affected by this;
it only affects end users who want to retrieve the
data by doing spatial searches.

P6. Add missing ASSOCIATEDPLATFORMINSTRUMENTSENSORCONTAINER attributes
---
The PGE generates 9 ASSOCIATEDPLATFORMINSTRUMENTSENSORCONTAINER
objects,
each with a ASSOCIATEDSENSORSHORTNAME object.
However, Data Server requires that there be 3 other objects in
this Container.
The fix to the MCF file described in P4 above allows the
first of the 9 ASSOCIATEDPLATFORMINSTRUMENTSENSORCONTAINER objects
to correctly include the 3 other objects.
However, the other 8 still incorrectly do not include them.
Edit the .met file to add the following to each of the 8
ASSOCIATEDPLATFORMINSTRUMENTSENSORCONTAINERs, after
ASSOCIATEDSENSORSHORTNAME:

```
    OBJECT          = ASSOCIATEDPLATFORMSHORTNAME
      CLASS         = "2"
      NUM_VAL       = 1
      VALUE         = "AM-1"
    END_OBJECT      = ASSOCIATEDPLATFORMSHORTNAME
    OBJECT          = OPERATIONMODE
      CLASS         = "2"
```

```
  NUM_VAL          = 1
  VALUE          = "Normal"
END_OBJECT          = OPERATIONMODE
OBJECT          = ASSOCIATEDINSTRUMENTSHORTNAME
  CLASS          = "2"
  NUM_VAL          = 1
  VALUE          = "MISR"
END_OBJECT          = ASSOCIATEDINSTRUMENTSHORTNAME
```

Replace the CLASS value with the same value as ASSOCIATEDSENSORSHORTNAME for that Container (2,3,...9).
When finished, you should have 9 identical ASSOCIATEDPLATFORMINSTRUMENTSENSORCONTAINERs, except for different values of ASSOCIATEDSENSORSHORTNAME and different CLASS values.
NCR ECSed12896 also applies for this problem.

P7. If MIB2GEOP granule still fails to Insert....

Check the Data Server log EcDsScienceDataServer.ALOG .
If you see a message

```
SybaseError<Attempt to insert duplicate key row
in object 'DsMdGranulePlatformXref'
with unique index 'PK_DSMDGRANULEPLATFORMXREF'
```

then you have not yet had a late fix installed.
The issue is a bug in Data Server which caused it to reject .met files that contain more than 1 instance of object ASSOCIATEDPLATFORMINSTRUMENTSENSORCONTAINER, with the same values for one
 or more of its attributes.
There are 9 such objects for the MIB2GEOP .met file,
each of which has a different value of ASSOCIATEDSENSORSHORTNAME,
but the same values for the other 3 attributes

ASSOCIATEDPLATFORMSHORTNAME
OPERATIONMODE
ASSOCIATEDINSTRUMENTSHORTNAME

It is the latter fact that causes the problem.
NCR 14212 was written for this problem.
The fix was verified as being part of the PDPS patch;
however, at this writing (20-May-98) it appears doubtful if
this fix actually made it in.
I wrote NCR 14978 for the missing fix.

****

End Changes to output .met file to be made after stopping Insert

****
++

```
End PDPS RUNS
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
*****************************************************************************
Summary of Suggested Changes for MISR to Make to PGE 7 at JPL
*****************************************************************************
Remove leading zeroes from SP_AM_PATH_NO values in metadata (see below).
  GROUP              = ADDITIONALATTRIBUTES
    OBJECT             = ADDITIONALATTRIBUTESCONTAINER
      CLASS            = "1"
      OBJECT             = ADDITIONALATTRIBUTENAME
        CLASS            = "1"
        NUM_VAL            = 1
        VALUE            = "SP_AM_PATH_NO"
      END_OBJECT           = ADDITIONALATTRIBUTENAME
      GROUP              = INFORMATIONCONTENT
        CLASS            = "1"
        OBJECT             = PARAMETERVALUE
          NUM_VAL            = 1
          CLASS            = "1"
/* 29-Apr-98 TWA  Leading 0 causes error in PDPS/Dsrv */
/*        VALUE              = "027" */
          VALUE            = "27"
        END_OBJECT           = PARAMETERVALUE
      END_GROUP            = INFORMATIONCONTENT
    END_OBJECT           = ADDITIONALATTRIBUTESCONTAINER
  END_GROUP            = ADDITIONALATTRIBUTES
*****************************************************************
End Changes to MISR PGE 7 Delivery
*****************************************************************
```

11.18.4.33 howto_run_from_cmdline_in_runtime_dir

```
lasher
cd
/usr/ecs/TS1/CUSTOM/pdps/lasher/data/DpPrRm/lasher_disk/MISRISH#syn2/MISRIS0000_las
her
csh% sh
. /usr/ecs/TS1/CUSTOM/bin/DPS/auto.profile
. MISRIS0000.Profile
grep PGS_PC_Shell MISRIS0000.Log
# Paste string after "COMMAND="
PGS_PC_Shell.sh
/usr/ecs//TS1/CUSTOM/pdps/lasher/data//DpPrRm/lasher_disk/MISRISH#syn2/synpge_sgi6n32
1111
```

/usr/ecs//TS1/CUSTOM/pdps/lasher/data//DpPrRm/lasher_disk/MISRISH#syn2/MISRIS0000_lasher/MISRIS0000.Pcf
50 -v –p
****

11.18.4.34 howto_setup_orbits_and_pathmaps

ORBITS & PATHMAPS
The Path is a an orbit swath, defined for Landsat-7 ("WRS"),
which MISR uses in its processing.
Because the earth rotates under it, the path the satellite
traverses for its next orbit is not path 2, but path 17.
This mapping of orbit number to path number is found in the
PATHMAP ODL file.
In that file ABSOLUTE_PATH is what we call orbit number here,
and MAPPED_PATH is the path number.
Now, this mapping is fixed, and never changes.
What does change is the time each orbit starts.
This is because the orbit may drift and be subject to
maneuvers.
Periodically (say, every 2 weeks), the Flight Dynamics
Facility (FDF) at GSFC issues a new Orbit Start Time,
with corresponding Orbit Number.
When this happens, the PDPS ORBIT ODL must be updated,
with a new ORBIT_MODEL object, containing the new
ORBIT_START and corresponding ORBIT_NUMBER.
The new ORBIT_PATH_NUMBER is determined manually by
the operator, using the lookup table in the PATHMAP ODL file.
1. Example (MISR PGE7 test data)
FDF issues a bulletin stating that imaginary platform MPGE7
orbit number 27 starts at 14:37:39Z 07-Jan-96.
SSIT operator receives the bulletin, looks up in
the corresponding PATHMAP_WRS7.odl file to find that
ABSOLUTE_PATH XX corresponds to MAPPED_PATH XX.
NEED TO FIX THIS
The SSIT operator then creates a new ORBIT_MODEL object in the
ORBIT ODL file as follows:

```
OBJECT = ORBIT_MODEL
    CLASS = 2
    ORBIT_NUMBER = 27
    ORBIT_PERIOD = "SECS=5932"
    ORBIT_START = "01/07/1996 14:37:39Z"
    ORBIT_PATH_NUMBER = 90
END_OBJECT = ORBIT_MODEL
```

****

11.18.4.35 howto_register_pge

```
# First prepare PGE and ESDT ODL files
# Update database
xterm -sb -sl 256 -bg maroon -fg "papaya whip" -cr "papaya whip" -fn
"*l*s*type*b*r*140
*" -T 'SSIT: Science Metadata Database Update' -n 'Science Metadata Update' -e
/usr/ecs
//OPS/CUSTOM/bin/DPS/EcDpAtDefinePGE &
# Update performance info
/usr/ecs//OPS/CUSTOM/bin/DPS/EcDpAtOpDbGui ConfigFile
/usr/ecs//OPS/CUSTOM/cfg/EcDpAtOp
DbGui.CFG ecs_mode OPS &
*****************************************************************************
```

11.18.4.36 howto_register_dpr

```
#############################################################################
# Make sure STMGT and SDSRV are up
#############################################################################
# Use either ECS Assist on texas *and* raven,
# Alternatively, use ps:
texas:> ps -ef | grep EcDsScienceDataServer | grep TS1
   sdsrv 24410     1  0 10:17:57 pts/3    0:33
/usr/ecs//TS1/CUSTOM/bin/DSS/EcDsScience
DataServer ConfigFile /usr/ecs//TS1/CUS
raven:> ps -ef | grep  Server | grep TS1 | grep ConfigFile
   stmgt  3786     1  0   Dec 30 ?      10:39
/usr/ecs/TS1/CUSTOM/bin/DSS/EcDsStStaging
DiskServer ConfigFile /usr/ecs/TS1/CUS
   stmgt  3803     1  0   Dec 30 ?      10:31
/usr/ecs/TS1/CUSTOM/bin/DSS/EcDsStStaging
MonitorServer ConfigFile /usr/ecs/TS1/
   stmgt  3770     1  0   Dec 30 ?      16:18
/usr/ecs/TS1/CUSTOM/bin/DSS/EcDsStArchive
Server ConfigFile /usr/ecs/TS1/CUSTOM/
   stmgt  3815     1  0   Dec 30 ?      10:59
/usr/ecs/TS1/CUSTOM/bin/DSS/EcDsStFtpDisS
erver ConfigFile /usr/ecs/TS1/CUSTOM/c
# If any of these are not present, that server is down.
# You must arrange to have it up before continuing
*****************************************************************************
```

Using Production Request Editor and Planning Workbench

PRE:
telnet odyssey, cmops, dce_login

ops, cd utilities,
then invoke EcplStartPRE_IF OPS 3 &
PWB: telnet odyssey, cmops, dce_login
ops, cd utilities, invoke EcPlSlayAll prior to using PWB.
then invoke EcPlStartAll OPS 3 & or 2
times used for plan activation: 01/01/1990 time 00:00:00
AutoSys- To Monitor DPS , telnet to: taltos_srvr
##

11.18.4.37 howto_Make a Production Request

##
rlogin odyssey -l pls
dce_login
awhitele
cd /usr/ecs/TS1/CUSTOM/utilities
set path = (/usr/bin)
source .buildrc
setenv DISPLAY mojave:0.0
Make sure no one else is running PREditor, then run it
/usr/ucb/ps -auxwww | grep -i pre | grep TS1
EcPlStartPRE_IF TS1 1
Click PREdit
Click PGE, select it, OK
Set time to input, e.g. (CPGE1)
06 12 1997 00 00 00
06 14 1997 00 00 00
****

11.18.4.38 Additional SSI&T Howto_procedures and Production rules taken from

PDPS tests documented in the following files last updated 12/01/98:
SCF: cd /home/PDPS/docs - look through list of 'howto_s'
production_rules_wp.doc*
Drop3Scen2.txt
HowToCompileAndRunSyntheticPGE
HowToDealWithDPSDeadLocks
HowToFakeSubscriptionNotification
HowToInstallOnComanche
HowToStartAutosysAndViewJobStates
HowToActivateAPlan
HowToDeleteAndRecreateJobs
HowToSetUpPlanning
Tiling.txt
HowToRunProductionStrategyGUI
SpatialQuery.txt

HowToInsertMultiFileGranules
HowToStartDPS
ModisDataList
HowToUseSubscriptionServerDriverToSendSubscriptionNotification
HowToReactivateReplan
VerifiedList
HowToStartQaMonitor
HowToTestFaultRecovery
HowToTestGroundEventJobs
HowToCleanAutoSys
HowToInstall-General
HowToInsertData
MergeList
HowToEnterGroundEvents
HowToTestDynamicMetaDataQuery
HowToReacquireAGranule
ReacquireGranule.sql
HowToAdHocReprocessPRs
HowToTestDatabaseCleanupScript
HowToDoProfiling
HowToRunMulti-GranuleESDTsTest
HowToRunSSITAcquireTool#
HowToRunResourcePlanning
HowToCreateDprepTarFile*HowToRunBROWSETest~
HowToRunBROWSETest
HowToRunSpatial
mailfile
HowToRunDataDay-InterimFilesTest~
HowToTest
HowToInstallPlanning~
HowToRunDataDay-InterimFilesTest
HowToInstallPlanning
HowToRunSSIT
HowToRunOptionalDPRs~
EndToEnd
HowToRunASTER
HowToRunOptionalDPRs
HowToRunTiling~
HowToRunMostRecentGranule~
HowToRunMostRecentGranule
HowToRunTiling
HowToTestAlternateInput~
HowToTestAlternateInput
HowToRunASTER#

HowToRunModis
HowToRunMISRLIKE~
HowToRunModisPlus
HowToRunMISRLIKE
HowToRunASTERRoutine
HowToRunDPREP
HowToRunMostRecentGranule#

11.18.4.39 Technical Notice concerning Leap Second/DPREP

Notice of Change to Toolkit Ephemeris and
Attitude Interpolation

March 15, 1999

Effective with a late patch to Drop 4PY, the time interval over which the Toolkit will interpolate spacecraft ephemeris or attitude is reduced from 121 seconds to 60 seconds. (The normal interval between packets for AM1 is 1.024 seconds, except for FDD replacement ephemeris, which will use 1 second time intervals.)

Impact of this Change

For reasons explained below, the only impact on real data processing would be to avoid the possibility of a less-than-ideal interpolation.

The impact on I&T would be that test data should always be generated or obtained with packet interval 60 seconds or less. There are a few inputs for the Toolkit test drivers, which are not a deliverable, but are occasionally supplied to Toolkit users, which (to save file space) are set up to work with 120 second packets. These lie in the file "orbsim.in" (which is, as explained, a PDPS add-on which is unsupported software when it is provided to users). To conduct tests without new and annoying failures, if you use this input file for "orbsim", you should edit it to replace "120" by "60" or less, globally. Results will not change significantly from the expected test results.

Rationale for this Change

The SDP Toolkit uses an analytic, cubic polynomial method to interpolate ephemeris that is extremely fast, but which could generate undesirable and spurious variations if applied across too large a time interval, or to data that are not smooth.

DPREP, which will process all incoming ephemeris data, has a robust method for patching gaps up to 60 seconds, based on a least squares quartic fit, component by component, to position and velocity. Although this method is undesirably slow for repeated use within the Toolkit, it is ideal for DPREP, which runs only once on a data set. The existence of any gaps not repairable by DPREP (i.e. > 60 seconds) will trigger procedures to obtain replacement data files from FDD.

It is therefore unwise to allow the Toolkit to interpolate gaps which DPREP cannot fill, so it is being changed. The same limit is set for attitude. We expect no gaps whatever in FDD attitude, and the attitude is not of much use without the ephemeris, so this change was made consistently.

This page intentionally left blank.