625-EMD-006

EOSDIS Maintenance and Development Project

Training Material for the EMD Project Volume 6: Production Planning and Processing

Revision 02

July 2006

Raytheon Company
Upper Marlboro, Maryland

Training Material for the EMD Project
Volume 6: Production Planning and Processing

Revision 02

July 2006

Prepared Under Contract NAS5-03098
CDRL Item 23

RESPONSIBLE AUTHOR

July 27 2006

Lay'wan Gamble and Debra Stepp — Date
EOSDIS Maintenance and Development Project

RESPONSIBLE OFFICE

7/27/2006

Mary Armstrong, Program Manager — Date
EOSDIS Maintenance and Development Project

Raytheon Company
Upper Marlboro, Maryland

This page intentionally left blank.

Preface

This document is a formal contract deliverable. It requires Government review and approval within 45 business days. Changes to this document will be made by document change notice (DCN) or by complete revision.

Any questions should be addressed to:

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

Revision History

Document Number	Status/Issue	Publication Date	CCR Number
625-EMD-006	Original	July 2004	04-0384
625-EMD-006	Revision 01	July 2005	05-0320
625-EMD-006	Revision 02	July 2006	06-0377

This page intentionally left blank.

Abstract

This is Volume 6 of a series of lessons containing the training material for the Earth Observing System Data and Information System (EOSDIS) Maintenance and Development (EMD) Project. This lesson provides a detailed description of the process required for creating, modifying, and implementing production requests and production plans and monitoring the processing of data processing requests.

***Keywords**:* training, instructional design, course objective, production request, production plan, data processing request, production, planning, processing, Release 7.

This page intentionally left blank.

Contents

Preface

Abstract

Introduction

Related Documentation

Production Planning and Processing Overview

Production Planning and Processing

Production Requests

Logging in to System Hosts

Launching the Production Request Editor

Creating/Updating/Deleting a Production Request

Reviewing/Deleting Data Processing Requests

Launching Planning Workbench-Related GUIs

Creating a New Production Plan

Cleaning the PDPS Database and Science Processing Disks

Troubleshooting Production Planning Problems

Production Processing

Launching the AutoSys GUI Control Panel

Configuring AutoSys Screens/Displays

Monitoring/Controlling Job Processing

Tuning System Parameters

Troubleshooting Processing Problems

Launching the QA Monitor

Performing Science Product Quality Assurance

Regenerating Granules

Practical Exercise

Slide Presentation

This page intentionally left blank.

Introduction

Identification

Training Material Volume 6 is part of Contract Data Requirements List (CDRL) Item 23, which is a required deliverable under the Earth Observing System Data and Information System (EOSDIS) Maintenance and Development (EMD) Contract (NAS5-03098).

Scope

Training Material Volume 6 describes the procedures by which the production team prepares production plans and monitors production processing. This lesson is designed to provide the operations staff with sufficient knowledge and information to satisfy all lesson objectives.

Purpose

The purpose of this Student Guide is to provide a detailed course of instruction that forms the basis for understanding production planning and processing. Lesson objectives are developed and will be used to guide the flow of instruction for this lesson. The lesson objectives will serve as the basis for verifying that all lesson topics are contained within this Student Guide and slide presentation material.

Status and Schedule

This lesson module provides detailed information about training for the current baseline of the system. Revisions are submitted as needed.

Organization

This document is organized as follows:

Introduction:	The Introduction presents the document identification, scope, purpose, and organization.
Related Documentation:	Related Documentation identifies parent, applicable and information documents associated with this document.
Student Guide:	The Student Guide identifies the core elements of this lesson. All Lesson Objectives and associated topics are included.
Slide Presentation:	Slide Presentation is reserved for all slides used by the instructor during the presentation of this lesson.

This page intentionally left blank.

Related Documentation

Parent Documents

The parent documents are the documents from which the EMD Training Material's scope and content are derived.

423-41-01	Goddard Space Flight Center, EOSDIS Core System (ECS) Statement of Work
423-46-03	EMD Task 101 Statement of Work For ECS SDPS Maintenance
423-46-02	Contract Data Requirements Document for EMD Task 101 ECS SDPS Maintenance

Applicable Documents

The following documents are referenced within this EMD Training Material, or are directly applicable, or contain policies or other directive matters that are binding upon the content of this document:

420-05-03	Goddard Space Flight Center, Earth Observing System (EOS) Performance Assurance Requirements for the EOSDIS Core System (ECS)
423-41-02	Goddard Space Flight Center, Functional and Performance Requirements Specification for the Earth Observing System Data and Information System (EOSDIS) Core System (ECS) (ECS F&PRS)
423-46-01	Goddard Space Flight Center, Functional and Performance Requirements Specification for the Earth Observing System Data and Information System (EOSDIS) Core System (ECS) Science Data Processing System (EMD F&PRS)

Information Documents

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 EMD Training Material.

500-EMD-001	Terra Spacecraft Ephemeris and Attitude Data Preprocessing
500-EMD-002	Aqua Spacecraft Ephemeris and Attitude Data Preprocessing
500-EMD-003	Aura Spacecraft Ephemeris and Attitude Data Preprocessing
609-EMD-001	Release 7.11 Operations Tools Manual for the EMD Project

611-EMD-001	Release 7.11 Mission Operation Procedures for the EMD Project
910-TDA-022	Custom Code Configuration Parameters for ECS
505-41-33	Interface Control Document Between EOSDIS Core System (ECS) and Science Computing Facilities (SCF)

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 EMD Training Material.

305-EMD-001	Release 7.11 Segment/Design Specification for the EMD Project
311-EMD-001	Data Management Subsystem (DMS) Database Design and Database Schema Specifications for the EMD Project
311-EMD-002	INGEST (INS) Database Design and Schema Specifications for the EMD Project
311-EMD-003	Planning and Data Processing Subsystem Database Design and Schema Specifications for the EMD Project
311-EMD-004	Science Data Server Database Design and Schema Specifications for the EMD Project
311-EMD-005	Storage Management and Data Distribution Subsystems Database Design and Database Schema Specifications for the EMD Project
311-EMD-006	Subscription Server Database Design and Schema Specifications for the EMD Project
311-EMD-007	Systems Management Subsystem Database Design and Schema Specifications for the EMD Project
311-EMD-008	Registry Database Design and Schema Specifications for the EMD Project
311-EMD-009	Product Distribution Subsystem (PDS) Database Design and Database Schema Specifications for the EMD Project
311-EMD-010	NameServer Database Design and Schema Specifications for the EMD Project
311-EMD-011	Order Manager Database Design and Database Schema Specifications for the EMD Project
311-EMD-012	Spatial Subscription Server (SSS) Database Design and Schema Specifications for the EMD Project
311-EMD-013	Data Pool Database Design and Schema Specifications for the EMD Project
313-EMD-001	Release 7 Internal Interface Control Document for the EMD Project

508-EMD-001	ACRONYMS for the EOSDIS Maintenance and Development (EMD) Project
152-TP-003	Glossary of Terms for the EOSDIS Core System (ECS) Project

This page intentionally left blank.

Production Planning and Processing Overview

Lesson Overview

This lesson will provide you with the complete process by which the production team prepares production plans and monitors production processing. The processes described in the lesson apply primarily to Production Planners and Production Monitors. The procedures involved in production planning and processing include such tasks as preparing production requests, preparing production plans and monitoring data processing.

Lesson Objectives

Overall Objective - The overall objective of the Production Planning and Processing lesson is for maintenance and operations personnel to develop proficiency in the procedures that apply to production planning and production processing operations for the Earth Observing System (EOS) Data and Information System (EOSDIS) Core System (ECS).

Condition - The student will be given oral or written information and requirements for performing production planning and processing activities, access to the planning and production processing systems, a copy of 609-EMD-001, *Release 7.11 Operations Tools Manual for the EMD Project*, and a copy of 611-EMD-001, *Release 7.11 Mission Operation Procedures for the EMD Project*.

Standard - The student will perform production planning and processing activities in accordance with the prescribed procedures without error.

Specific Objective 1 - The student will describe the general functions and processes included in the Planning Subsystem and the Data Processing Subsystem (in the context of system operations).

Condition - The student will be given written or oral questions concerning the general functions and processes included in the Planning and Data Processing Subsystems.

Standard - The student will state without error the general functions and processes included in the Planning and Data Processing Subsystems in accordance with the lesson content.

Specific Objective 2 - The student will describe the purposes and general functions of production rules in the context of system operations.

Condition - The student will be given written or oral questions concerning the purposes and general functions of production rules.

Standard - The student will state without error the purposes and general functions of production rules in accordance with the lesson content.

Specific Objective 3 - The student will perform the steps involved in logging in to system hosts.

Condition - The student will be given a statement of the requirements for logging in to system hosts, access to the Planning Subsystem (through a workstation or X terminal), a copy of 609-EMD-001, *Release 7.11 Operations Tools Manual for the EMD Project*, and a copy of 611-EMD-001, *Release 7.11 Mission Operation Procedures for the EMD Project*.

Standard - In accordance with the lesson content, the applicable procedure, and the statement of requirements the student will access the command shell, set the DISPLAY environmental variable, and log in to the specified host using secure shell and the specified user ID.

Specific Objective 4 - The student will perform the steps involved in launching the Production Request Editor.

Condition - The student will be given a statement of the requirements for launching the Production Request Editor, access to the Planning Subsystem (through a workstation or X terminal), a copy of 609-EMD-001, *Release 7.11 Operations Tools Manual for the EMD Project*, and a copy of 611-EMD-001, *Release 7.11 Mission Operation Procedures for the EMD Project*.

Standard - In accordance with the lesson content, the applicable procedure, and the statement of requirements the student will log in to the appropriate host and start the Production Request Editor graphical user interface (GUI) in the specified mode.

Specific Objective 5 - The student will perform the steps involved in creating a new production request using the Production Request Editor GUI.

Condition - The student will be given a statement of the requirements for creating a new production request, access to the previously launched Production Request Editor GUI in the Planning Subsystem (through a workstation or X terminal), a copy of 609-EMD-001, *Release 7.11 Operations Tools Manual for the EMD Project*, and a copy of 611-EMD-001, *Release 7.11 Mission Operation Procedures for the EMD Project*.

Standard - In accordance with the lesson content, the applicable procedure, and the statement of requirements the student will select the PR Edit tab on the Production Request Editor GUI, prepare a new production request that is consistent with the written or stated requirements, and save the new production request.

Specific Objective 6 - The student will perform the steps involved in creating new production requests using the Production Request (PR) Generator (command-line interface).

Condition - The student will be given a statement of the requirements for creating new production requests, access to the Planning Subsystem (through a workstation or X terminal), a copy of 609-EMD-001, *Release 7.11 Operations Tools Manual for the EMD Project*, and a copy of 611-EMD-001, *Release 7.11 Mission Operation Procedures for the EMD Project*.

Standard - In accordance with the lesson content, the applicable procedure, and the statement of requirements the student will log in to the appropriate host, prepare an input file specifying the PGE ID (PgeId) and GEO ID (GEOId) values to be used in creating the production requests, start the PR Generator to create the new production requests, and check the PR Generator debug log to determine the results of running the PR Generator.

Specific Objective 7 - The student will perform the steps involved in modifying a production request.

Condition - The student will be given a statement of the requirements for modifying a production request, access to the previously launched Production Request Editor GUI in the Planning Subsystem (through a workstation or X terminal), a copy of 609-EMD-001, *Release 7.11 Operations Tools Manual for the EMD Project*, and a copy of 611-EMD-001, *Release 7.11 Mission Operation Procedures for the EMD Project*.

Standard - In accordance with the lesson content, the applicable procedure, and the statement of requirements the student will select the PR Edit tab on the Production Request Editor GUI, select the Production Request to be modified, make production request modifications consistent with the written or stated requirements, and save the modified production request.

Specific Objective 8 - The student will perform the steps involved in deleting a production request.

Condition - The student will be given a statement of the requirements for deleting a production request, access to the previously launched Production Request Editor GUI in the Planning Subsystem (through a workstation or X terminal), a copy of 609-EMD-001, *Release 7.11 Operations Tools Manual for the EMD Project*, and a copy of 611-EMD-001, *Release 7.11 Mission Operation Procedures for the EMD Project*.

Standard - In accordance with the lesson content, the applicable procedure, and the statement of requirements the student will select the PR List tab on the Production Request Editor GUI, select the production request to be deleted (from those listed), and delete the production request.

Specific Objective 9 - The student will perform the steps involved in reviewing data processing requests.

Condition - The student will be given a statement of the requirements for reviewing data processing requests, access to the previously launched Production Request Editor GUI in the Planning Subsystem (through a workstation or X terminal), a copy of 609-EMD-001, *Release 7.11 Operations Tools Manual for the EMD Project*, and a copy of 611-EMD-001, *Release 7.11 Mission Operation Procedures for the EMD Project*.

Standard - In accordance with the lesson content, the applicable procedure, and the statement of requirements the student will select the DPR List tab on the Production Request Editor GUI, select a Production Request from the list on the option button, select a DPR from the list displayed, open the DPR, and respond to questions concerning the characteristics of the DPR.

Specific Objective 10 - The student will perform the steps involved in deleting a data processing request.

Condition - The student will be given a statement of the requirements for deleting a data processing request, access to the previously launched Production Request Editor GUI in the planning system (through a workstation or X terminal), a copy of 609-EMD-001, *Release 7.11 Operations Tools Manual for the EMD Project*, and a copy of 611-EMD-001, *Release 7.11 Mission Operation Procedures for the EMD Project*.

Standard - In accordance with the lesson content, the applicable procedure, and the statement of requirements the student will select the DPR List tab on the Production Request Editor GUI, select the appropriate Production Request from the list on the option button, select the DPR to be deleted from the list displayed, and delete the DPR.

Specific Objective 11 - The student will perform the steps involved in launching the Production Strategies GUI.

Condition - The student will be given a statement of the requirements for launching the Production Strategies GUI, access to the Planning Subsystem (through a workstation or X terminal), a copy of 609-EMD-001, *Release 7.11 Operations Tools Manual for the EMD Project*, and a copy of 611-EMD-001, *Release 7.11 Mission Operation Procedures for the EMD Project*.

Standard - In accordance with the lesson content, the applicable procedure, and the statement of requirements the student will log in to the appropriate host using secure shell and start the Production Strategies GUI in the specified mode.

Specific Objective 12 - The student will perform the steps involved in launching the Planning Workbench-related GUIs.

Condition - The student will be given a statement of the requirements for launching the Planning Workbench-related GUIs, access to the Planning Subsystem (through a workstation or X terminal), a copy of 609-EMD-001, *Release 7.11 Operations Tools Manual for the EMD Project*, and a copy of 611-EMD-001, *Release 7.11 Mission Operation Procedures for the EMD Project*.

Standard - In accordance with the lesson content, the applicable procedure, and the statement of requirements the student will log in to the appropriate host using secure shell and start the Message Handler, Resource Model, System Name Server, Planning Workbench GUI, and the Planning Timeline in the specified mode.

Specific Objective 13 - The student will perform the steps involved in defining a production strategy.

Condition - The student will be given a statement of the requirements for defining a production strategy, access to the previously launched Production Strategies GUI in the Planning Subsystem (through a workstation or X terminal), a copy of 609-EMD-001, *Release 7.11 Operations Tools Manual for the EMD Project*, and a copy of 611-EMD-001, *Release 7.11 Mission Operation Procedures for the EMD Project*.

Standard - In accordance with the lesson content, the applicable procedure, and the statement of requirements the student will select priorities for the values for PR Type, User Type, and PGE Type, enter weights for the various attribute categories; enter a weight in the Production Request Editor field; normalize the weighting; and save the production strategy.

Specific Objective 14 - The student will perform the steps involved in creating a new production plan.

Condition - The student will be given a statement of the requirements for creating a new production plan, access to the previously launched Planning Workbench GUI in the Planning Subsystem (through a workstation or X terminal), a copy of 609-EMD-001, *Release 7.11 Operations Tools Manual for the EMD Project*, and a copy of 611-EMD-001, *Release 7.11 Mission Operation Procedures for the EMD Project*.

Standard - In accordance with the lesson content, the applicable procedure, and the statement of requirements the student will access the Planning Workbench; prepare a new production plan as directed; save the new production plan; activate the plan as directed; and shut down the Planning Master Timeline, Planning Workbench, Message Handler, Resource Model, and System Name Server.

Specific Objective 15 - The student will perform the steps involved in reviewing a production plan timeline.

Condition - The student will be given a statement of the requirements for reviewing a production plan timeline, access to the previously launched Production Planning Master Timeline in the Planning Subsystem (through a workstation or X terminal), a copy of 609-EMD-001, *Release 7.11 Operations Tools Manual for the EMD Project*, and a copy of 611-EMD-001, *Release 7.11 Mission Operation Procedures for the EMD Project*.

Standard - In accordance with the lesson content, the applicable procedure, and the statement of requirements the student will access the specified Planning Master Timeline, review the specified Production Planning Master Timeline, and respond to questions concerning the Production Planning Master Timeline.

Specific Objective 16 - The student will perform the steps involved in cleaning the PDPS database and science processing disks.

Condition - The student will be given a statement of the requirements for cleaning the PDPS database and science processing disks, access to the Planning Subsystem (through a workstation or X terminal), a copy of 609-EMD-001, *Release 7.11 Operations Tools Manual for the EMD Project*, and a copy of 611-EMD-001, *Release 7.11 Mission Operation Procedures for the EMD Project*.

Standard - In accordance with the lesson content, the applicable procedure, and the statement of requirements the student will log in to the appropriate host using secure shell, set environmental variables if necessary, and start the appropriate script(s) using suitable arguments.

Specific Objective 17 - The student will perform the steps involved in troubleshooting production planning problems.

Condition - The student will be given a statement of the requirements for troubleshooting production planning problems, access to the Planning, Data Processing, Data Server, and Communications Subsystems (through a workstation or X terminal), a copy of 609-EMD-001, *Release 7.11 Operations Tools Manual for the EMD Project*, and a copy of 611-EMD-001, *Release 7.11 Mission Operation Procedures for the EMD Project*.

Standard - In accordance with the lesson content, the applicable procedure, and the statement of requirements the student will review the trouble symptoms, check the status of relevant hosts/servers (as necessary), check log files (as necessary), take action to correct the problem(s), and respond to questions concerning the possible cause(s) of the trouble symptoms.

Specific Objective 18 - The student will perform the steps involved in launching the AutoSys GUI Control Panel.

Condition - The student will be given a statement of the requirements for launching the AutoSys GUI Control Panel, access to the Data Processing Subsystem (through a workstation or X terminal), a copy of 609-EMD-001, *Release 7.11 Operations Tools Manual for the EMD Project*, and a copy of 611-EMD-001, *Release 7.11 Mission Operation Procedures for the EMD Project*.

Standard - In accordance with the lesson content, the applicable procedure, and the statement of requirements the student will log-in to the appropriate host, set the necessary environmental variables, source the appropriate file, and start the GUI for the appropriate instance of AutoSys.

Specific Objective 19 - The student will perform the steps involved in configuring AutoSys runtime options.

Condition - The student will be given a statement of the requirements for configuring AutoSys runtime options, access to the previously launched AutoSys GUI Control Panel in the Data Processing Subsystem (through a workstation or X terminal), a copy of 609-EMD-001, *Release 7.11 Operations Tools Manual for the EMD Project*, and a copy of 611-EMD-001, *Release 7.11 Mission Operation Procedures for the EMD Project*.

Standard - In accordance with the lesson content, the applicable procedure, and the statement of requirements the student will access AutoSys functions as directed, select AutoSys runtime options as directed, and apply specified AutoSys runtime options.

Specific Objective 20 - The student will perform the steps involved in monitoring/controlling job processing.

Condition - The student will be given a statement of the requirements for monitoring/controlling job processing, access to the previously launched AutoSys GUI Control Panel in the Data Processing Subsystem (through a workstation or X terminal), a copy of 609-EMD-001, *Release 7.11 Operations Tools Manual for the EMD Project*, and a copy of 611-EMD-001, *Release 7.11 Mission Operation Procedures for the EMD Project*.

Standard - In accordance with the lesson content, the applicable procedure, and the statement of requirements the student will access AutoSys/Job Management Web Interfaces and monitor/control job processing, including responding to alarms, specifying job selection criteria, determining the ownership of an AutoSys job, sending an event to a job, canceling a sent event, performing job management client functions, reviewing job activity and job dependency reports, and defining and running monitors and browsers.

Specific Objective 21 - The student will perform the steps involved in troubleshooting processing problems.

Condition - The student will be given a statement of the requirements for troubleshooting processing problems, access to the Data Processing Subsystem (through a workstation or X terminal), a copy of 609-EMD-001, *Release 7.11 Operations Tools Manual for the EMD Project*, and a copy of 611-EMD-001, *Release 7.11 Mission Operation Procedures for the EMD Project*.

Standard - In accordance with the lesson content, the applicable procedure, and the statement of requirements the student will review the trouble symptoms, check the status of relevant hosts/servers (as necessary), check log files (as necessary), take action to correct the problem(s), and respond to questions concerning the possible cause(s) of the trouble symptoms.

Specific Objective 22 - The student will perform the steps involved in launching the Quality Assurance (QA) Monitor GUI.

Condition - The student will be given a statement of the requirements for QA Monitor GUI, access to the Data Processing Subsystem (through a workstation or X terminal), a copy of 609-EMD-001, *Release 7.11 Operations Tools Manual for the EMD Project*, and a copy of 611-EMD-001, *Release 7.11 Mission Operation Procedures for the EMD Project*.

Standard - In accordance with the lesson content, the applicable procedure, and the statement of requirements the student will log in to the appropriate host and start the QA Monitor GUI in the appropriate mode.

Specific Objective 23 - The student will perform the steps involved in updating quality assurance (QA) metadata of a science product granule.

Condition - The student will be given a statement of the requirements for updating QA metadata, access to the previously launched QA Monitor GUI in the Data Processing Subsystem (through a workstation or X terminal), a copy of 609-EMD-001, *Release 7.11 Operations Tools Manual for the EMD Project*, and a copy of 611-EMD-001, *Release 7.11 Mission Operation Procedures for the EMD Project*.

Standard - In accordance with the lesson content, the applicable procedure, and the statement of requirements the student will set up and query the database using the QA Monitor GUI, select the granule with QA metadata to be updated, set the operational and SCF quality flags to the appropriate values, and verify that the flags have actually been set in the database.

Specific Objective 24 - The student will perform the steps involved in regenerating granules in response to a loss of files from the archive.

Condition - The student will be given a statement of the requirements for regenerating granules in response to a loss of files from the archive, access to the Planning and Data Processing Subsystems (through a workstation or X terminal), a copy of 609-EMD-001, *Release 7.11 Operations Tools Manual for the EMD Project*, and a copy of 611-EMD-001, *Release 7.11 Mission Operation Procedures for the EMD Project*.

Standard - In accordance with the lesson content, the applicable procedure, and the statement of requirements the student will retrieve the Production History files (PH) for lost granules, create Production Requests for the generation of replacement granules, create and activate a Production Plan that includes the Production Requests for the generation of replacement granules, and prepare (if applicable) a "PDPS Residual Granules List."

Importance

This lesson applies to students who will be members of the production team (especially Production Planners and Production Monitors). The lesson will provide them with the knowledge and skills needed when performing their assigned tasks. Those tasks include (among other things) the following activities:

- Logging in to System Hosts
- Launching the Production Request Editor
- Creating/Updating/Deleting a Production Request
- Reviewing/Deleting Data Processing Requests
- Launching Planning Workbench-Related GUIs
- Creating a New Production Plan
- Defining/Deleting a Production Strategy
- Creating/Deleting a Production Plan
- Troubleshooting Production Planning Problems
- Launching the AutoSys GUI Control Panel
- Configuring AutoSys Screens/Displays
- Reviewing Hardware Status
- Monitoring/Controlling Job Processing
- Responding to Alarms
- Specifying Job Selection Criteria
- Determining the Ownership of an AutoSys Job
- Sending an Event to a Job

- Canceling a Sent Event
- Troubleshooting Processing Problems

The lesson describes why and how the activities are performed. The students will become aware of what tasks they will be performing on the job and how to accomplish those tasks.

This page intentionally left blank.

Production Planning and Processing

System Context

Production planning and processing processes are accomplished at the Distributed Active Archive Centers (DAACs). The people involved in production planning and processing activities are Production Planners and Production Monitors.

- The Production Planner performs planning functions; especially, using the Planning Subsystem (PLS) to create Data Processing Requests and specify which requests are to be processed as part of a particular Production Plan.
- The Production Monitor keeps track of operations in the Data Processing Subsystem, especially the execution of science data processing jobs (creation of data products).

The Context Diagram (Figure 1) shows the relationships among the Planning Subsystem, Data Processing Subsystem, Data Server Subsystem, and the other subsystems within the Science Data Processing component of the system. It is apparent that the interfaces the Planning and Data Processing Subsystems have with each other and that each has with the Data Server Subsystem (which manages access to the data archive) are critically important. Of course the context diagram shows a generalized (high-level) view of the system. The Planning Subsystem and Data Processing Subsystem Architecture diagrams (Figures 2 and 3 respectively) focus on the individual subsystems and their relationships with other subsystems.

The Planning Subsystem (Figure 2) provides a mechanism for accomplishing the following general functions:

- Defining DAAC production resources.
- Scheduling production resources for non-production-related activities.
- Defining data processing jobs to be performed at the DAAC.
- Generating efficient plans for scheduling defined data processing jobs.
- Coordinating production with the Data Server Subsystem and Data Processing Subsystem to achieve a highly automated production system.

The Data Processing Subsystem PRONG computer software configuration item (CSCI) shown in Figure 3 is involved in the following general functions:

- Managing the allocation of data processing jobs to the site's data processing resources.
- Managing, queuing, and executing data processing jobs to produce data products.
- Supporting preliminary processing of ancillary data granules.

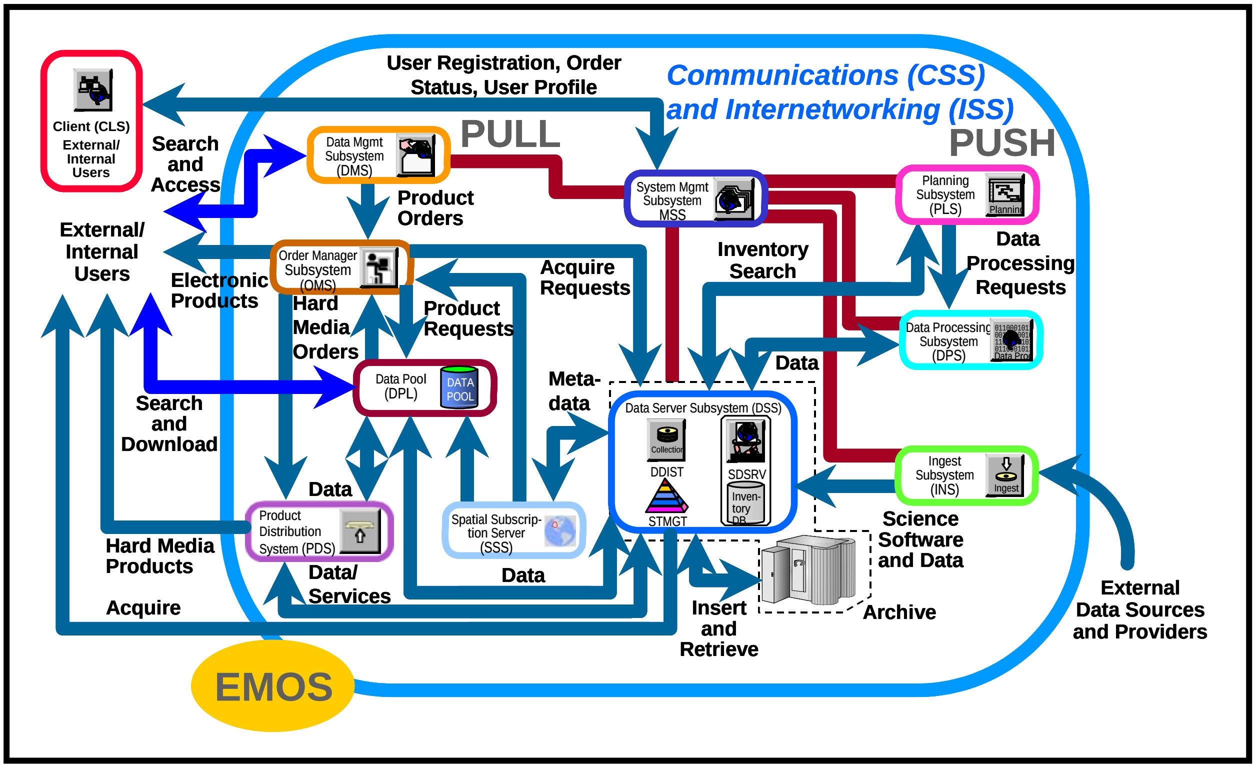

Figure 1. Context Diagram

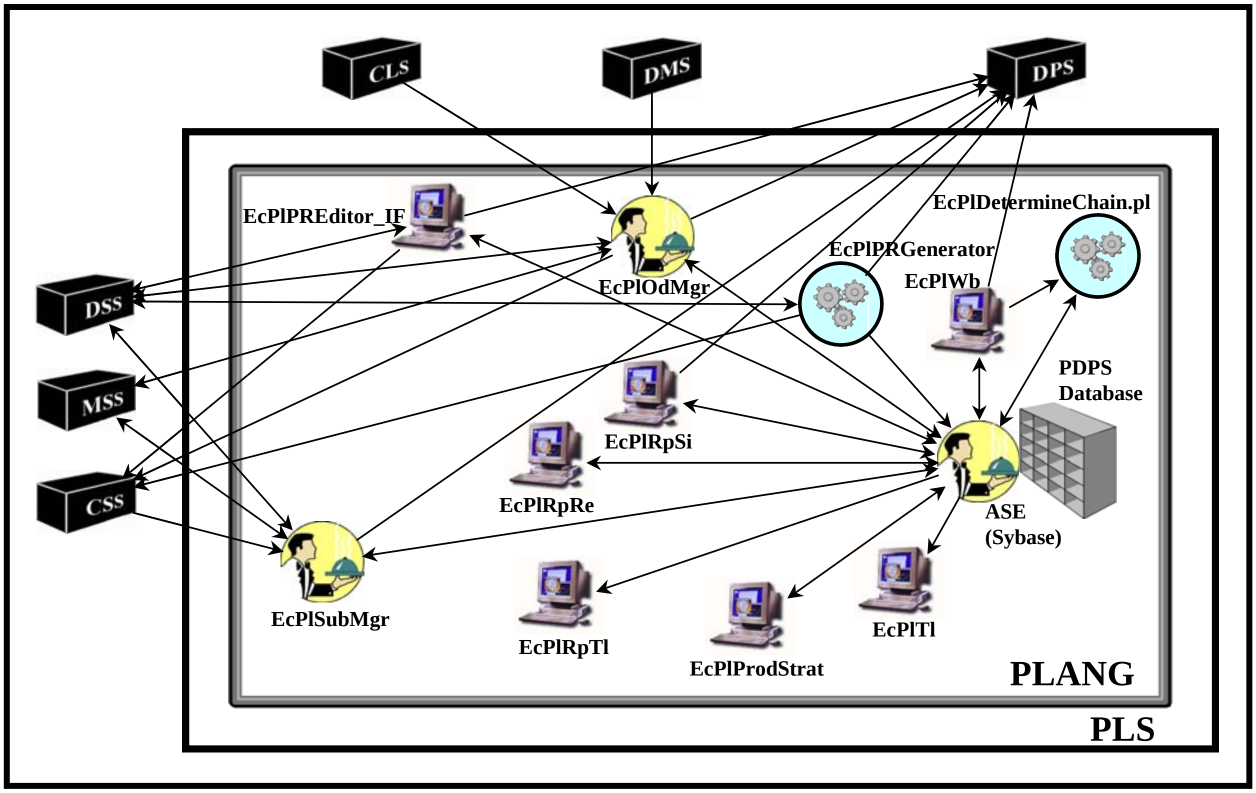

Figure 2. Planning Subsystem (PLANG CSCI) Architecture

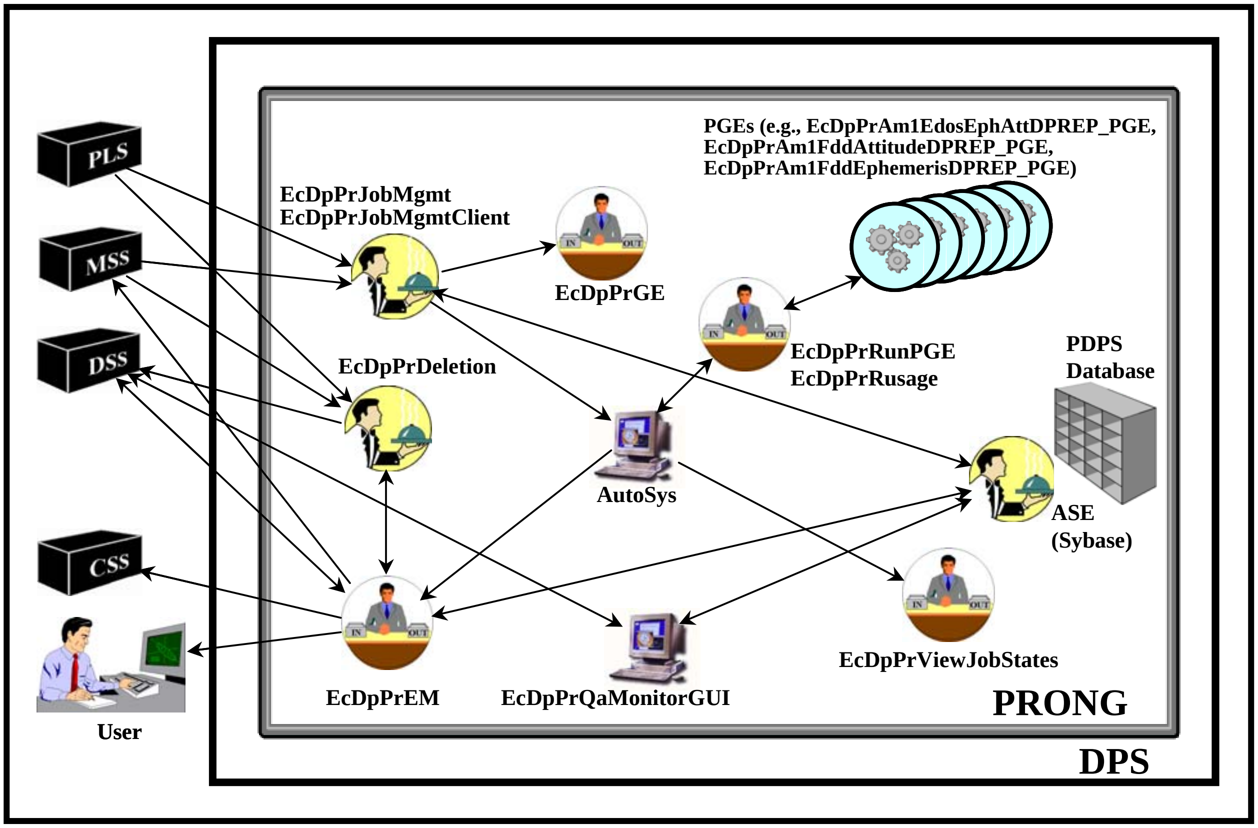

Figure 3. Data Processing Subsystem (PRONG CSCI) Architecture

Planning Subsystem

The Planning Subsystem (Figure 2) is the Science Data Processing subsystem that the Production Planner uses when developing production plans. The Production Planner has access to the Planning Subsystem primarily through the Production Request Editor and the Planning Workbench. The Production Request Editor (PRE) is used for creating or modifying production requests (PRs). The Planning Workbench is used for scheduling PRs.

The Planning Subsystem is composed of just one computer software configuration item (CSCI); i.e., PLANG. (The designation “PLANG” is derived from **PLAN**NIN**G**.) The subsystem has the following major components as shown in Figure 2:

- Resource Planning Workbench.
  - Resource Editor (EcPlRpRe) - Graphical user interface (GUI) for defining/editing the resources at the site.
  - Resource Scheduler (EcPlRpSi) - GUI for creating/approving/committing resource reservations for non-production-related events and preparing a site resource schedule.

- Resource Reservation Planning Master Timeline GUI (EcPlRpTl) - Graphical interface for displaying the resource schedule.

- Production Request Editor (EcPlPREditor).
  - GUI for submitting production requests that describe the data products to be produced; uses product generation executive (PGE) descriptions to generate the data processing requests (DPRs) necessary to fulfill the production requests.
- Production Planning Workbench.
  - Planning Workbench GUI (EcPlWb) - GUI for preparing and activating a site production schedule.
  - Production Strategies GUI (EcPlProdStrat) - GUI for defining production strategies (assign priorities for DPRs based on such characteristics as the type of production request, who is requesting processing, and the type of PGE to be run).
  - Planning Master Timeline GUI (EcPlTl) - Graphical interface for displaying production schedules, including resource reservations.
- On-Demand Manager (EcPlOdMgr).
  - Also known as the On-Demand Product Request Manager or ODPRM.
  - Server that receives on-demand product requests [from the EOS Data Gateway (EDG) web client via the V0 Gateway] from users.
  - Generates Production Requests needed to fill on-demand product requests and submits the PRs to the Data Processing Subsystem for processing.
- On-Demand Manager Client (EcPlOdMgrClient).
  - A utility program that manually provides some client functionality for EcPlOdMgr.
- Subscription Manager (EcPlSubMgr).
  - Server that manages receipt of subscription notifications (e.g., availability of input data needed for DPRs).
- Sybase Adaptive Server Enterprise (ASE) Server.
  - Commercial off-the-shelf (COTS) software application that handles interfaces with the Planning and Data Processing Subsystems’ (PDPS) shared database for planning and processing activities.
  - Performs the functions of a Structured Query Language (SQL) server for the PDPS database.

In addition to the preceding major components the Planning Subsystem includes the following components associated with both the resource planning applications and the production planning workbench:

- Message Handler (EcPlMsh).
  - GUI that displays various types of messages including warning messages and information messages.
- System Name Server (EcPlSns).
  - Handles interprocess communication.
- Resource Model (EcPlRpRm, EcPlRm).
  - Underlying resource data coordinators for the planning software.

The Message Handler, System Name Server, and Resource Model are associated with both the resource planning workbench and the production planning workbench applications.

Planning/production personnel use the following start-up and shutdown scripts (which are available in the /usr/ecs/*MODE*/CUSTOM/utilities directory on the Planning/Management Workstation) to either start or shut down many of the preceding applications:

- EcPlSomeStart.
  - Launches the System Name Server, Message Handler, and Resource Model needed by the Planning Workbench GUI.
  - Does not launch the Planning Workbench GUI.
- EcPlAllStart.
  - Launches the Planning Workbench and Planning Master Timeline as well as the Message Handler, System Name Server, and Resource Model.
- EcPlPRE_IFStart.
  - Launches the Production Request Editor.
- EcPlPRE_ReadOnlyStart.
  - Launches a read-only instance of the Production Request Editor.
- EcPlPRGeneratorStart.
  - Launches the Production Request Generator (command-line interface for creating production requests).
- EcPlProdStratStart.
  - Launches the Production Strategies GUI.

- EcPlRpAllStart.
  - Launches the Message Handler, System Name Server, and Resource Planning Resource Model needed by the Resource Editor and Resource Scheduler GUIs.
- EcPlRpReStart.
  - Launches the Resource Editor after the Message Handler, System Name Server, and Resource Planning Resource Model have been started (using EcPlRpAllStart).
- EcPlRpSiStart.
  - Launches the Resource Scheduler after the Message Handler, System Name Server, and Resource Planning Resource Model have been started (using EcPlRpAllStart).
- EcPlSubsEditStart.
  - Starts the Subscription Editor.
- EcPlTlStart.
  - Launches the Production Planning Master Timeline (assuming that the Message Handler, System Name Server, and Resource Model have been started).
- EcPlWbStart.
  - Launches the Planning Workbench if the Message Handler, System Name Server, and Resource Model have already been started.
- EcPlSlay.
  - Shuts down a Planning Subsystem executable that must be specified by name and application ID (message-service ID).
- EcPlSlayAll.
  - Shuts down the Planning Master Timeline, Message Handler, System Name Server, and Resource Model (and the Planning Workbench if it has not already been shut down).
- EcPlRpSlayAll.
  - Shuts down the Resource Planning Master Timeline, Message Handler, System Name Server, and Resource Model (and the Resource Scheduler and Resource Editor GUIs if they have not already been shut down).

The following start-up scripts are in the /usr/ecs/*MODE*/CUSTOM/utilities directory on the Queuing Server:

- EcPlOdMgrClientStart.
  - Starts the EcPlOdMgrClient.
- EcPlOdMgrStart.
  - Starts the PLS server EcPlOdMgr.
- EcPlPlanningAppStart.
  - Starts Planning Subsystem server group (i.e., Subscription Manager, On-Demand Manager).
- EcPlStart.
  - Starts Planning Subsystem server (i.e., Subscription Manager).
- EcPlSubMgrStart.
  - Starts the Subscription Manager.

The following start-up scripts in the /usr/ecs/*MODE*/CUSTOM/utilities directory on the Planning/Management Workstation are typically called by other applications and are not normally invoked directly by planning/production personnel:

- EcPlMshStart.
  - Starts the Message Handler individually.
- EcPlRmStart.
  - Starts the Resource Model individually.
- EcPlRpRmStart.
  - Starts the Resource Model individually.
- EcPlSnsStart.
  - Starts the System Name Server individually.
- EcPlStart.
  - Starts Planning Subsystem server (i.e., Subscription Manager).
- SweeperStart.
  - Starts the sweeper; called by EcCsIdPingServers.

In addition to the preceding start-up scripts the following scripts are available in the /usr/ecs/*MODE*/CUSTOM/utilities directory:

- EcLgLogCtrlStart.
  - Not applicable to log files for any Planning Subsystem or Data Processing Subsystem server. [Script for setting the Debug Log or ALOG Level to any value between 0 and 3 “on the fly” (without having to re-start the applicable server).]
- EcPlCdsPingServers.
  - Not currently functional due to the implementation of sockets and the retirement of the Distributed Computing Environment (DCE). [Uses DCE Cell Directory Service (CDS)].
- EcPlDbClean.
  - Cleans up some PDPS database tables by deleting records that meet specified criteria. [calls a stored procedure]
- EcPlDbCleanArchive.
  - Cleans Earth Science Data and Information System (ESDIS) Data Gathering and Reporting System (EDGRS) archive tables of granules older than seven days. [calls a stored procedure]
  - Intended to be run on a daily basis.
- EcPlDbBuild.
  - Supports installation of the PDPS database. Creates new database schema (builds tables and other database objects for the current release). Drops any previously existing objects, including any data in the tables. (The script automatically invokes additional scripts; such as EcPlDbDrop.) Designed to be run from ECS Assistant. The user must be Database Operator (DBO) with sso_role (Site Security Officer).
- EcPlDbDrop.
  - Drops all tables and related objects in the database. Designed to be run from ECS Assistant. Parameters are the same as for EcPlDbBuild. The user must be DBO.
  - EcPlDbDrop is called automatically by EcPlDbBuild. Due to complaints from some sites that obsolete tables were still present in their databases, EcPlDbDrop automatically drops all obsolete tables that still exist.

- EcPlDbDump.
  - Dumps the transaction log, the database, and the master database to a flat file that can be used for database recovery. Designed to be run from ECS Assistant. Creates files in the directory /usr/ecs/<*MODE*>/COTS/sybase/sybase_dumps with filenames pdps_tran_dmp.<timestamp>, pdps_dmp.<timestamp>, and master_dmp.<timestamp> respectively. Parameters are the same as for EcPlDbBuild. The user must be a System Administrator (SA) or at least DBO with OPER role.
- EcPlDbMigrate.
  - Migrates data from one version of the PDPS database to another.
- EcPlDbPatch.
  - Used in upgrading an existing PDPS database schema to the next valid database version level. Patches any modified or new database structures to the database without having to re-install the entire database. Allows for existing data to be maintained. Invoked through ECS Assistant.
  - Changes only those objects that have been modified since the last release, preserving existing data in the tables. Parameters are the same as for EcPlDbBuild. The user must be DBO.
- EcPlRpFetchBaseline.
  - [not used].
- EcPlDbReset.
  - Clears the data in the database tables and loads the values in the specified "saved database" file.
- EcPlDbList.
  - Provides a listing of saved databases.
- EcPlDbSave.
  - Saves the current database.
- fos_services.
  - Script used by the Sweeper executable (not normally invoked directly by planning/production personnel).

The following Planning Subsystem script is available in the /usr/ecs/*MODE*/CUSTOM/bin/PLS directory on the Planning/Management Workstation:

- EcPlDetermineChain.pl.
  - Called by the Planning Workbench GUI when a plan is activated.
  - The chainFlag values (in the PDPS database PlProductionRequest and PlDataProcessingRequest tables) for previously unflagged chain heads are set and a chainId (PlDataProcessingRequest table) is assigned to each DPR.

The following Communications Subsystem scripts (among others) are available in the /usr/ecs/*MODE*/CUSTOM/utilities directory on the Planning/Management Workstation:

- EcCsIdPingServers.
  - Pings servers using host manager (HST) sockets middleware.
  - Replacement for EcPlCdsPingServers, which does not work due to the implementation of sockets and the retirement of DCE.
- EcCsPerfLogProcessor.pl
  - Extracts performance information from log files.

The following Client Subsystem application is available in the /usr/ecs/*MODE*/CUSTOM/eosview directory on the Planning/Management Workstation:

- EOSView.
  - EOSView is an HDF-EOS viewer for visualizing science data. EOSView can take any HDF-EOS data file and perform visualization functions; however, it does not provide sophisticated data analysis functions.

Data Processing Subsystem

The Data Processing Subsystem is the Science Data Processing subsystem that the Production Monitor uses when monitoring data processing. The Production Monitor has access to the Data Processing Subsystem primarily through AutoSys and the Quality Assurance Monitor (QA Monitor). AutoSys is used for monitoring the processing of DPRs. The QA Monitor is used primarily for updating QA metadata flags.

The Data Processing Subsystem is composed of the following three computer software configuration items (CSCIs):

- PRONG (derived from PROCESSI**NG**).
  - Provides the services required to manage and monitor the Science Data Processing environment, which executes Science Software items (PGEs) and produces data products.

- Algorithm Integration & Test Tools (AITTL).
    - Set of tools used for test and integration of new science software, new versions of science software, and user methods into the Science Data Processing operational environment.
- Science Data Processing (SDP) Toolkit.
    - Provides a set of software libraries, which are used to integrate Science Software into the system environment.

The PRONG CSCI is the focus of this section. PRONG (shown in Figure 3) has the following major components:

- Job Management (EcDpPrJobMgmt).
    - Uses the AutoSys COTS product to create and initiate execution of PRONG administrative jobs for managing science processor hardware assets and for PGE execution.
    - Responsible for efficient AutoSys management so the maximum number of jobs possible can be continuously run using the product. (Controls the flow of jobs through AutoSys by only allowing jobs ready to run into the product and by removing jobs as they complete.)
    - Creates and starts execution of Ground Event jobs in AutoSys.
- Ground Event process (EcDpPrGE).
    - Initiated by the Job Management Server when the server gets a ground event request.
    - The ground event process starts at a specified time and runs a specified duration.
    - During the time the ground event process runs, it sets a computer resource [central processing unit (CPU), random-access memory (RAM), etc.] off-line and the computer resource is not available for running PGEs.
- Job Management Client (EcDpPrJobMgmtClient).
    - Used by programs that need access to the Job Management Server services to modify jobs in AutoSys.
        - Can be used to create, cancel, release, or change the priority of jobs (among other functions).
    - Used as a tool for testing the operation of the Job Management Server.

- AutoSys.
  - Event Processor (AutoSys daemon).
    - COTS job scheduling software application used to accomplish the execution of jobs that support PGE execution in an automated fashion.
  - Event Server.
    - Sybase database server for the AutoSys database.
  - AutoSys GUIs.
    - Allow human intervention in the AutoSys job stream.
    - Provide various mechanisms for monitoring and altering the job stream.
  - AutoSys Job Management Web Interface.
    - Provide graphical depictions of scheduled jobs, completed jobs, and jobs being processed.
    - Allow human intervention in the AutoSys job stream.
    - Provide various mechanisms for monitoring and altering the job stream.
- Execution Management (EcDpPrEM).
  - Initiates PGE execution (via AutoSys).
  - Supports the preparation activities prior to the execution of each PGE and activities subsequent to the execution of each PGE.
  - Provides status on On-Demand Processing Requests and sends e-mail to the originator in the event of a failure.
  - The Data Management library portion of EcDpPrEM (DpPrDM) manages the flow of science data to and from science processing resources including communication mechanisms to interface with the EcDsScienceDataServer.
    - Data Management manages data retention on science processing resources to support PGE executions.
- PGE Execution Manager (EcDpPrRunPGE).
  - Controls and monitors the execution of a PGE, including the creation of a Process Control File (PCF) and the growth of the output products.

  - Provides a buffer between AutoSys and the PGE. It serves as a wrapper to the PGE process, initiates the PGE execution and captures the PGE's exit status.
  - Monitors the PGE's computer resources. If the PGE's computer resources exceed its expected usage an alarm is sent to AutoSys.
- Resource Usage (EcDpPrRusage).
  - Measures the actual resources used by a PGE.
  - Reports unexpected resource usage to AutoSys.
- View Job States (EcDpPrViewJobStates).
  - Called by the EcDpPrDisplayJobStates script to generate reports that show which jobs have completed, the jobs that are executing, and the jobs awaiting execution.
- Data Preprocessing (DPREP).
  - Set of PGEs that use a statistical approach to convert Level 0 (L0) attitude and ephemeris ancillary data for a particular satellite (e.g., Terra, Aqua, or Aura) into SDP Toolkit native binary format without altering or modifying the scientific content of the granules.
  - Terra DPREP consists of the following PGEs:
    - EcDpPrAm1EdosEphAttDPREP_PGE.
    - EcDpPrAm1FddAttitudeDPREP_PGE.
    - EcDpPrAm1FddEphemerisDPREP_PGE.
    - EcDpPrDumpAttitudeDPREP.
    - EcDpPrDumpEphemerisDPREP.
  - Aqua DPREP consists of the following PGEs:
    - EcDpPrPm1FddEphemerisDPREP_PGE.
    - EcDpPrPm1AttitudeDPREP_PGE.
  - Aura DPREP consists of the following PGEs:
    - EcDpPrAuraEphemerisDPREP_PGE.
    - EcDpPrAuraAttitudeDPREP_PGE.
- Deletion Server (EcDpPrDeletion).
  - Notifies Science Data Server to remove interim granules via the execution management process once they are no longer needed.

  - Interim product is removed after the last PGE in the chain has used the interim product or a pre set time has expired after its last use.
  - Used by PLS to delete granules associated with a cancelled DPR.
- Deletion Client (EcDpPrDeletionClient).
  - Used by programs that need access to Deletion Server services to delete granules.
- Sybase ASE Server.
  - COTS product that acts as a SQL (database) server for the PDPS database.
- Quality Assurance Monitor (EcDpPrQaMonitorGUI).
  - Simple interface that allows DAAC operators to transfer science data from the archives, browse data images, and examine and update science metadata.
- EcDpPrLoadTable.pl.
  - Script that is run from the Job Management Client Tool to load the DpPrPgeLimits table.
  - If there were no record in the DpPrPge table for a particular PGE ID/computer combination that was scheduled, the DPR would not be prevented from running and there would be no limits placed on how many of those DPRs could run on the same virtual machine.
- EcDpPrAutocons.
  - A resource (configuration) parameter file that assigns labels to the user-defined (i.e., ECS/EMD project-defined) buttons on the AutoSys Job Activity Console and specifies the command associated with each button.
    - For example, Button 3 is labeled "Client Tool" and calls the command "/usr/ecs/TS2/CUSTOM/utilities/EcDpPrJobMgmtClientStart TS2 $JOB" that launches the Job Management Client.
- EcDpPrEMGetAncHeaders.
  - Extracts the header record of the Toolkit ancillary file to which the Logical Unit Number is pointing.
  - Results are written to the standard output.
- EcDpPrPREPQCConverterPGE.
  - Shell script for converting National Centers for Environmental Control (NCEP) PREPQC (Quality Controlled Observation Data) data products from the BUFR format to the HDF-EOS point format.

- Calls EcDpPrReadPREPQCData and EcDpPrWritePREPQCDataToHDFEOS in performing the conversion.

- EcDpPrReadPREPQCData.
  - PREPQC read program called by EcDpPrPREPQCConverterPGE when converting NCEP PREPQC data products from the BUFR format to the HDF-EOS point format.
- EcDpPrWritePREPQCDataToHDFEOS.
  - PREPQC write program called by EcDpPrPREPQCConverterPGE when converting NCEP PREPQC data products from the BUFR format to the HDF-EOS point format.
- EcDpPrSMFCopy.
  - Shell script that copies Status Message Facility (SMF) files from the toolkit message directory to a PGE directory.
    - 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 of sending log files, informational messages, and output data files to DAAC personnel or to remote users.

Production personnel use the following start-up scripts (which are available in the /usr/ecs/*MODE*/CUSTOM/utilities directory on the Queuing Server):

- EcDpPrAutosysStart.
  - Launches the AutoSys GUI Control Panel.
- EcDpPrDeletionClientStart.
  - Starts the Deletion Client.
- EcDpPrGarbageCollectorStart.
  - Starts the Deletion Client.
  - Differs from the EcDpPrDeletionClientStart script in the following three ways:
    - Does not open a separate xterm window.
    - Requires specification of a MACHINE_TO_COLLECT variable.
    - Includes retry logic in case of database deadlock.

The following start-up scripts in the /usr/ecs/*MODE*/CUSTOM/utilities directory on the Queuing Server are typically called by other applications and are not normally invoked directly by production personnel:

- EcDpPrDeletionStart.
  - Starts the Deletion Server.
- EcDpPrDisplayJobStates.
  - Invoked by AutoSys Job Activity Console buttons to generate either a "Jobs Waiting" or a "Jobs Completed" report.
  - Executes EcDpPrViewJobStates to generate the specified report.
- EcDpPrJobMgmtClientStart.
  - Starts the Job Management Client.
- EcDpPrJobMgmtStart.
  - Starts the Job Management Server.
- EcDpPrStart.
  - Starts the Data Processing Subsystem server group (i.e., Job Management Server and Deletion Server).
- EcDpProcessingAppStart.
  - Starts the Data Processing Subsystem server group (i.e., Job Management Server and Deletion Server).

The following start-up script is available in the /usr/ecs/*MODE*/CUSTOM/utilities directory on the Planning/Management Workstation:

- EcDpPrQaMonitorGUIStart.
  - Launches the Quality Assurance (QA) Monitor GUI.
  - The QA Monitor GUI is a Data Processing System GUI but it is installed on the Planning/Management Workstation with many of the Planning applications.

In addition to the preceding start-up scripts the following scripts are available in the /usr/ecs/*MODE*/CUSTOM/utilities directory on the Queuing Server:

- EcDpBusySystemClean.pl.
  - Performs a rough cleanup of some of the PDPS database tables based on information in the PlDataGranuleShort table.

- Removes from the run-time directory all files for which it is removing the file names from the DpPrFile database table.

- EcDpPrRestartFailedJobs.
  - Restarts all restartable jobs (i.e., not failed PGE jobs) for a specified mode/autosys instance.
- EcDpPrCleanMaintMachineFiles.pl.
  - Cleans out entries in various tables in the PDPS database for specified science processor(s) [deallocates resources on the specified processor(s)].
- EcDpPrRmFilesWOGranules.pl.
  - Ensures consistency between the file references in various tables in the PDPS database and the files actually staged on the science processing disks (removes files from the disks or references in the database tables as necessary).

Production Requests

Science Software and Production Requests

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

- Performs the actual data processing to create desired products.
- Is developed at Science Computing Facilities (SCFs).
- Is embodied in Product Generation Executives (PGEs) when the software is integrated into the system 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 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 Science Data Server (if the data have been previously ingested) or from internal predictions (if the data are expected to arrive in the future).

- Determines what data will be included in the DPR output so the system can make predictions concerning the availability of data for subsequent PGEs.

Figure 4 shows the relationships among the PGEs, PRs, and DPRs as they are accessed through the Production Request Editor GUI.

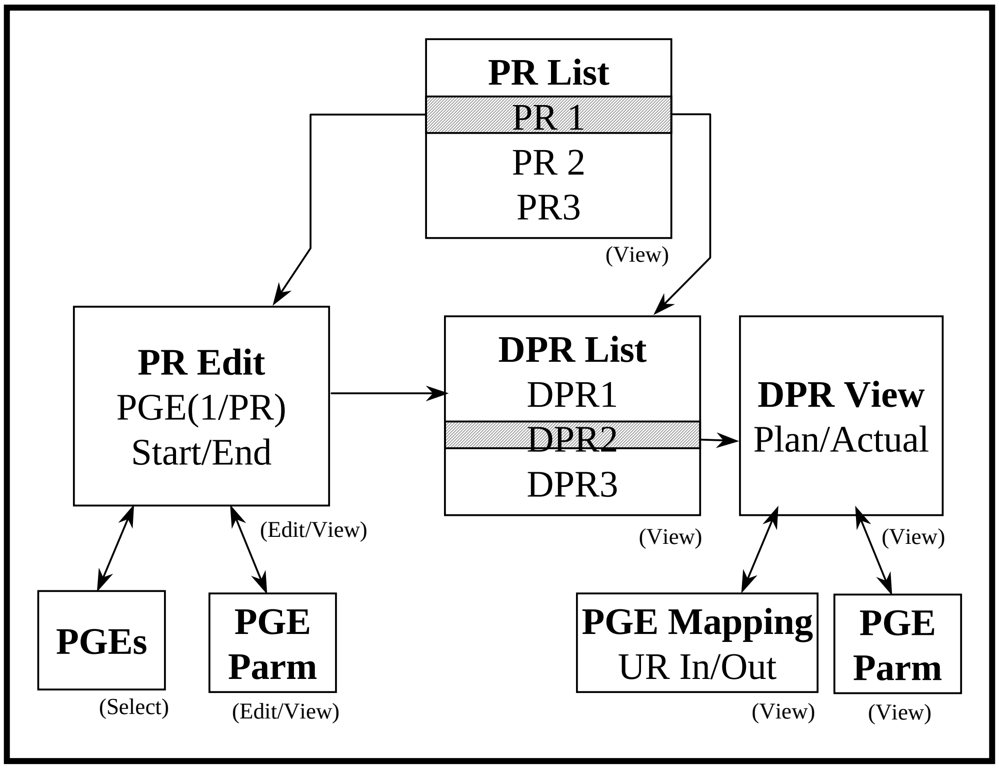

Figure 4. Production Request Editor Flow

Types of Processing

The system accommodates the following four general types of data processing:

- Routine Processing.
- Reprocessing.
- Regeneration.
- 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. The upgraded products are archived in addition to the versions that were previously archived.

Reprocessing is likely to be a large-scale operation, especially if several years worth of data are to be reprocessed. In fact the reprocessing load is expected to at least equal that of standard production. Consequently, the Production Planner schedules reprocessing in manageable quantities so the processing resources can accommodate routine and on-demand processing in addition to the reprocessing.

Regeneration is a type of reprocessing that is performed for the purpose of replacing a missing or damaged product. It is necessary when an existing product has been corrupted or deleted. If the damaged or missing product is needed for shipping or as input for additional processing, it must be recreated. To the extent possible the regenerated file is created using the same input, the same processing parameters, and the same algorithm as the original file.

On-demand processing involves ad-hoc processing initiated by 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. The ASTER researcher would use the EOS Data Gateway (EDG) web client to request the ASTER Level 2 product. The EDG would submit an on-demand request via the ECS V0 Gateway to have the product generated from a Level 1B product stored in the archive. The On-Demand Manager (EcPlOdMgr, also known as the ODPRM) in PLS would create the appropriate PRs and DPR(s) for processing and order(s) for distribution of the output product(s). The PLS would notify the DPS to initiate processing of the DPR(s) when the necessary input data became available.

Further details concerning the types of processing are provided in subsequent sections of this training lesson.

SCF Processing Requests

As specified in the *Interface Control Document Between EOSDIS Core System (ECS) and Science Computing Facilities (SCF)* (505-41-33) SCFs may request the DAACs to process or reprocess data produced using SCF-provided science software. The SCFs' processing/reprocessing request messages can include optional run-time parameters, in case of changes from the original processing parameters.

SCF staff send Processing Requests to the DAAC staff using one of the following three protocols:

- e-mail.
- X11/internet.
- X11/modem.

SCFs must use the e-mail interface unless the DAAC has authorized their use of the Processing Request X11 Interfaces. The specific protocol used in each instance will have been defined in an Operations Agreement (or lower-level type of document) between the SCF and the DAAC.

SCF Processing Requests provide the DAAC operations staff with the following types of information:

- PGE Name.
- PGE Version.
- Optional PGE Profile (number 1-99 defining how the PGE is to be run).
- The time window for which the product should be regenerated.
- Optional run-time parameters for Production Request Editor.
- Optional comment.

DAAC operations personnel acknowledge receipt of an SCF's Processing Request by replying to the SCF in an e-mail message that includes the SCF's original request and the following statement:

- "This is an acknowledgment of the receipt of the following Processing Request: . . . "

Production Rules

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 granules, 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, 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.
  - The following two pieces of information can be placed in a runtime parameter if so desired by the PGE:
    - Data Collection Start and Stop Times. The times are specified to the PGE via two logical IDs in the Process Control File for the PGE. Logical IDs 10258 and 10259 have the Start Data Collection and Stop Data Collection times for the current DPR run of the PGE. The date/times are only

accurate to the second and may not exactly match the times in the data supplied to the PGE because PDPS can match granules that are seconds (or even minutes) from the time set for the DPR.

- Year, Month and Day of the Data Collection Start. The year/month/day information can be placed under any Logical ID that the PGE desires. They are separate entries (each goes under a different logical ID) and can provide the PGE with the **year**, **month** and the **day** of input data collection.

Some (but not all) production rules can work with other production rules.

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 within a 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. Static inputs can be chosen through the following two production rules only:
  - Metadata Query.
  - Spatial Query.

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.

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 Science Metadata 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.

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.

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.

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, 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 two 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.

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) 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.

Production Rules

The following statements provide some simplified descriptions of the production rules:

- **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 Inputs** - PGE is run with different inputs based on the availability of various alternate input data sets.
- **Optional Inputs** - 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).
- **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.
- **Spatial Query/Spatial Pad** - Input granule selection is based on the spatial coverage of another input (i.e., the key input). Spatial Pad involves adding area to all sides of the key input's spatial shape. All granules that intersect the expanded area are retrieved.
- **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.
- **Multiple DPRs for Insertion Time -** Allows the creation of DPRs for multiple granules with the same insertion time (affects ASTER L1B routine processing only).
- **Tiling -** Input data is chosen on the basis of Instrument Team-defined tiles (geographic areas).

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 PGE for a different instrument 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.

Non-routine data can be used as an input to a Basic Temporal DPR. Such data must exist at the Science Data Server for the time period of the DPR when the DPR is created or it can be retrieved using the **Closest Granule** or **Optional Input** Production Rules. If no other Production Rule is used and the non-routine data is **not** available at Science Data Server, the DPR fails at creation time. Because there is no Boundary or Period associated with non-routine data, PDPS cannot predict it; consequently, it must be available at Production Request time.

Running with data that falls within the time of the DPR only is a special case that is applicable to some PGEs. Normally, PDPS would match any granules that intersect the time of the DPR. However, some PGEs require data that falls within the time of the DPR only. For example:

- A data set is composed of granules with the following time ranges:
  - 01:00-02:00
  - 02:00-03:00
  - 03:00-04:00
- The DPR has the following time range:
  - 01:30-03:30

Usually a PGE would want all three granules in the data set because they all intersect the time of the DPR. But if a PGE wants the data that falls within the DPR time only, the 02:00-03:00 granule would be the only one selected. Restriction of the data to the time of the DPR is accomplished using the **DPR Align Flag** (ALIGN_DPR_TIME_WITH_INPUT_TIME).

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.

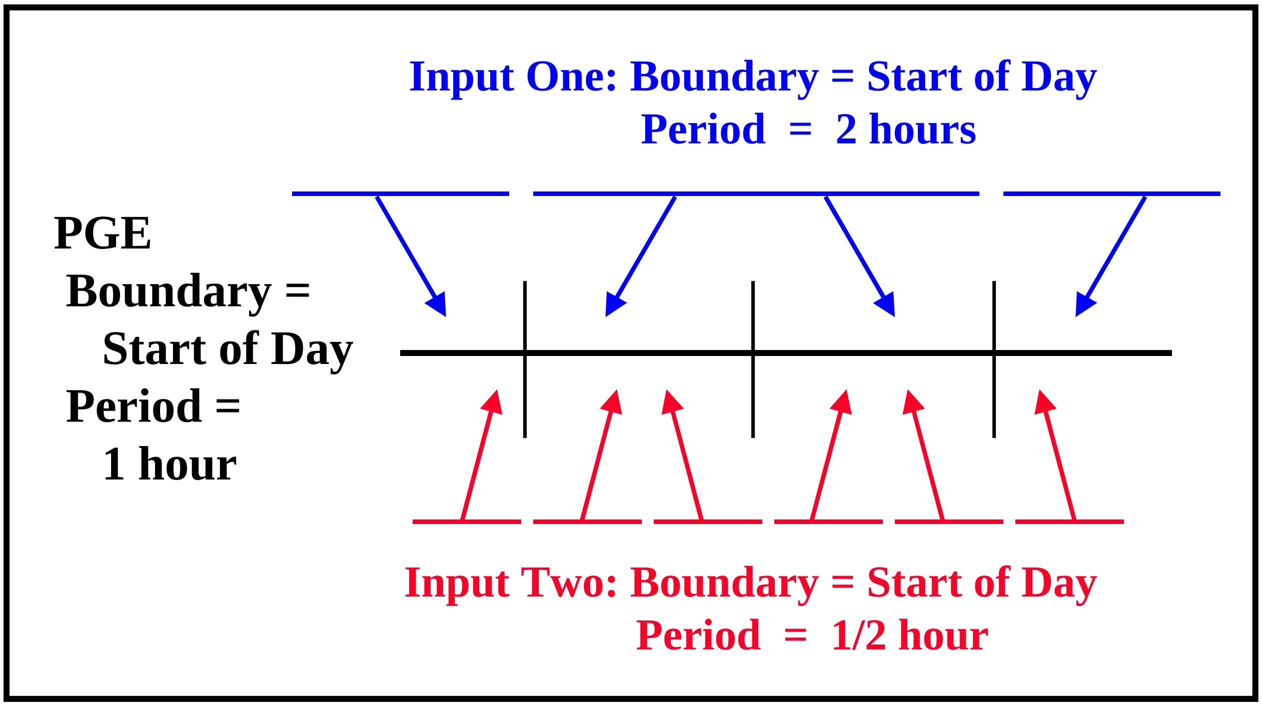

Figure 5. Example of the Basic Temporal Production Rule

In the example (Figure 5), 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.

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_OF_TIME" - PGE runs from the specified date/time in the form "START_OF_TIME=DD/MM/YYYY HH:MM:SS".
  - Can be used in conjunction with all types of PROCESSING_PERIOD.

There are other values for PROCESSING_BOUNDARY that apply to other production rules (as described in the applicable sections of the lesson).

Retrieving Data that Falls within the DPR Time Only

When the ALIGN_DPR_TIME_WITH_INPUT_TIME flag is set to "Y" (i.e., ALIGN_DPR_TIME_WITH_INPUT_TIME = "Y") PDPS uses only the granules for the input that fall within (instead of intersecting) the time of the DPR. If the flag is **not** set or is set to "N", granules for the input will be chosen on the basis of whether or not they intersect the time of the DPR.

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).
  - Earth Observing System (EOS) Data and Operations System (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.

The PREDICTION_METHOD parameter specifies the method of prediction that the Planning and Data Processing Systems use when creating DPRs that use the ESDT. The parameter has two possible values:

- "ROUTINE"
  - Data granules come in at regular intervals.
  - Boundary and Period are used when predicting the inputs to DPRs.
- "NONROUTINE"
  - Data granules come in at odd or random intervals.
  - DPRs that use the input can work off data that exists in the Science Data Server when the production request is created. This is because PDPS cannot predict the data.
  - When PREDICTION_METHOD = NONROUTINE, Boundary and Period are **not** used.

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 an 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 6 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.

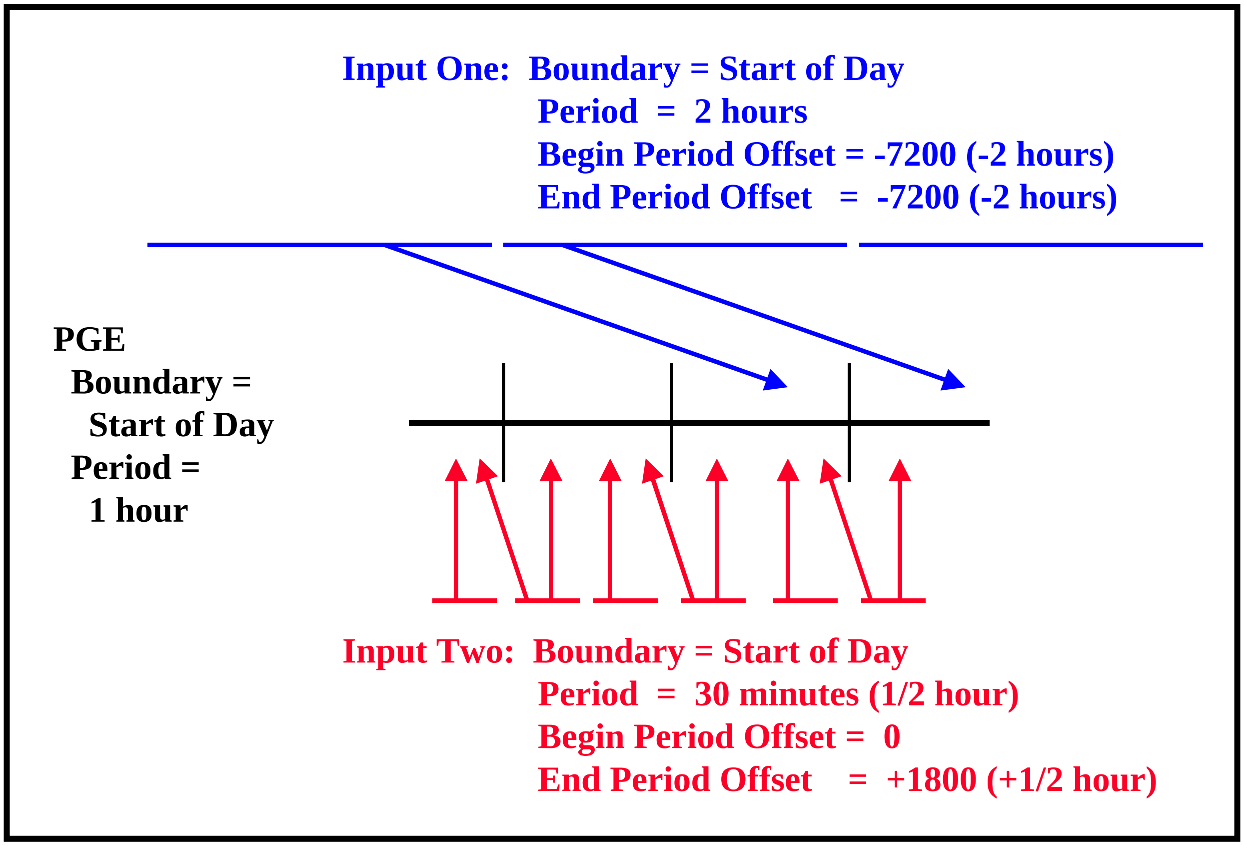

Figure 6. Example of the Advanced Temporal Production Rule

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 that fall within the time period of the DPR plus 1/2 hour would be staged. 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.

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.

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 were 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 7 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.

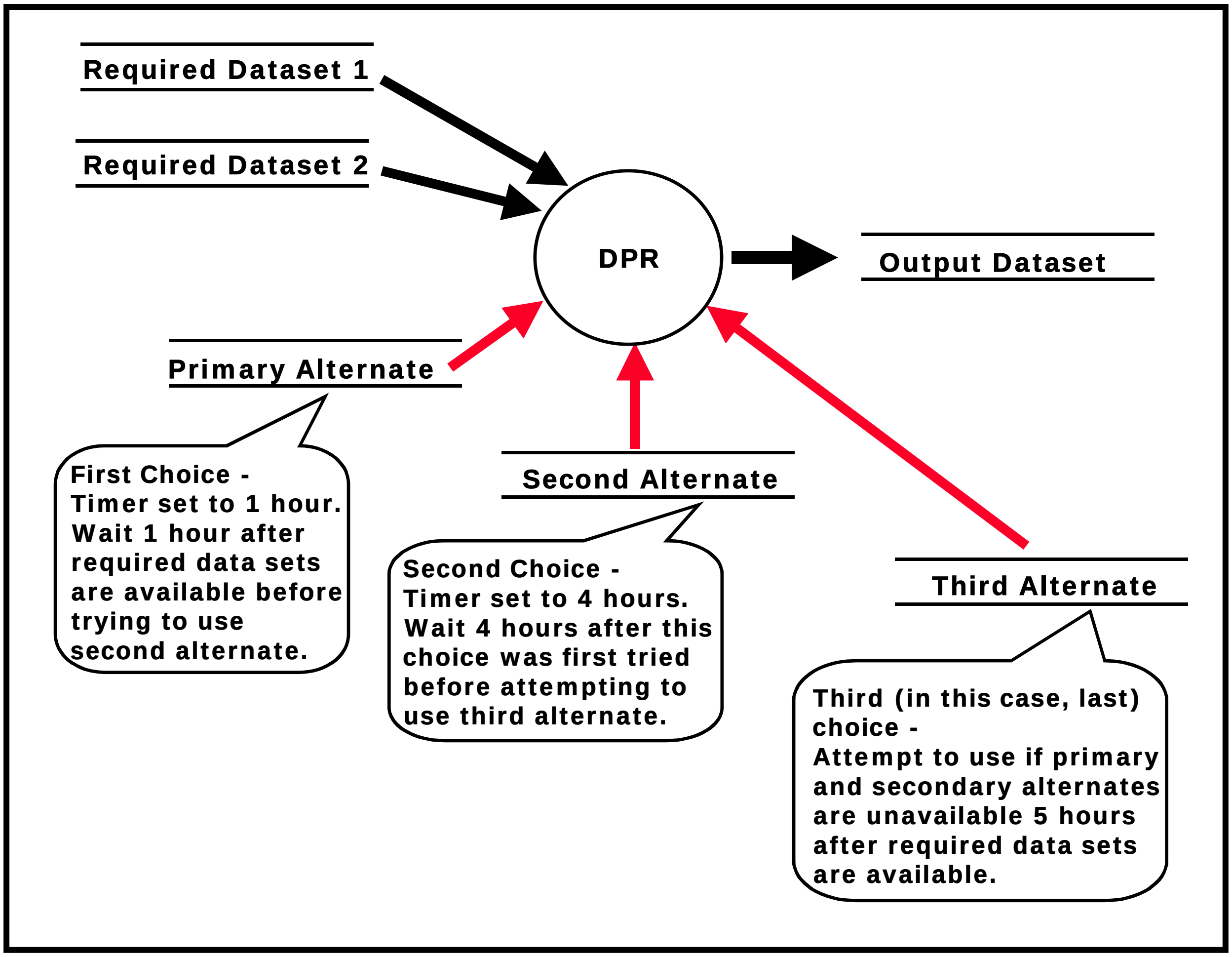

Figure 7. Example of the Alternate Input Production Rule

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.
- 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.

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 input:

- "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).

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.

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).

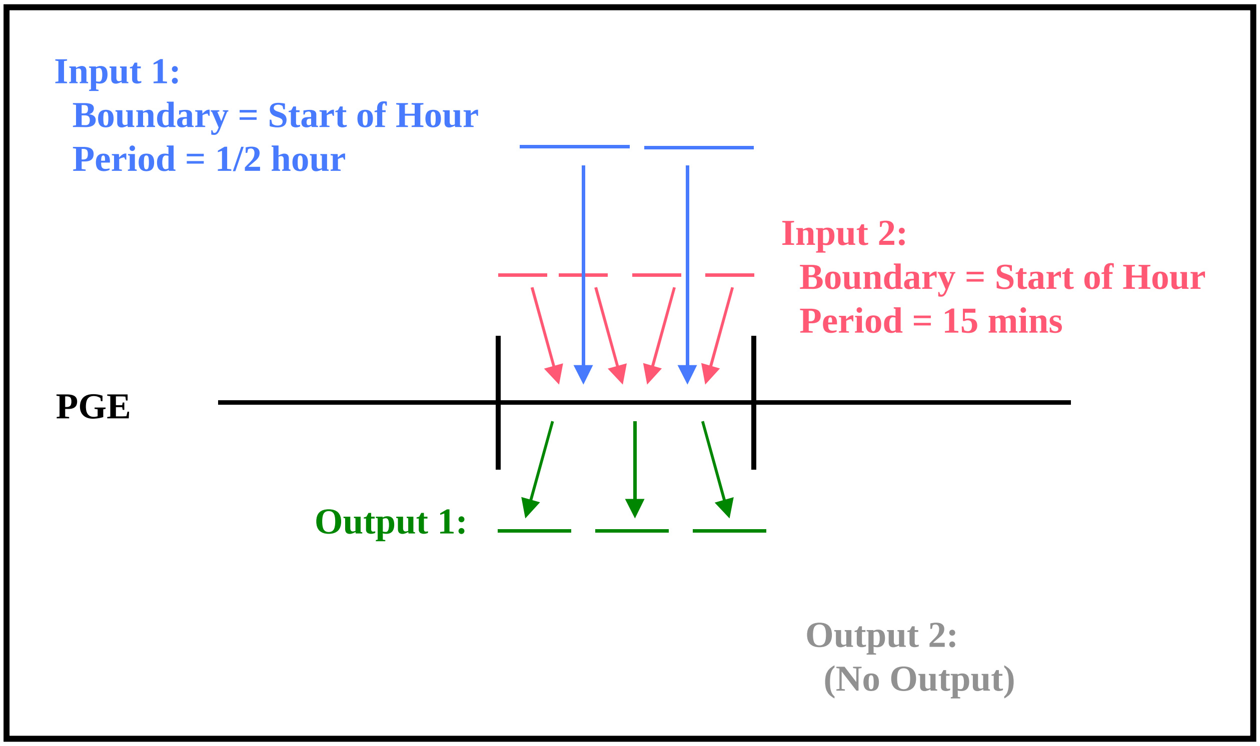

Figure 8. Example of the Minimum/Maximum Number of Granules Production Rule

- Input 2:
  - If 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).

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.

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).

NOTE: Key input applies to the Spatial Query Production Rule as well as the Optional DPRs Production Rule. Consequently, it is possible for the Optional DPRs and Spatial Query Production Rules to be used in combination only when they use the same key input.

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, the **insertion time** range (i.e., the time period when the desired data were archived) is used 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 9. 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.

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.

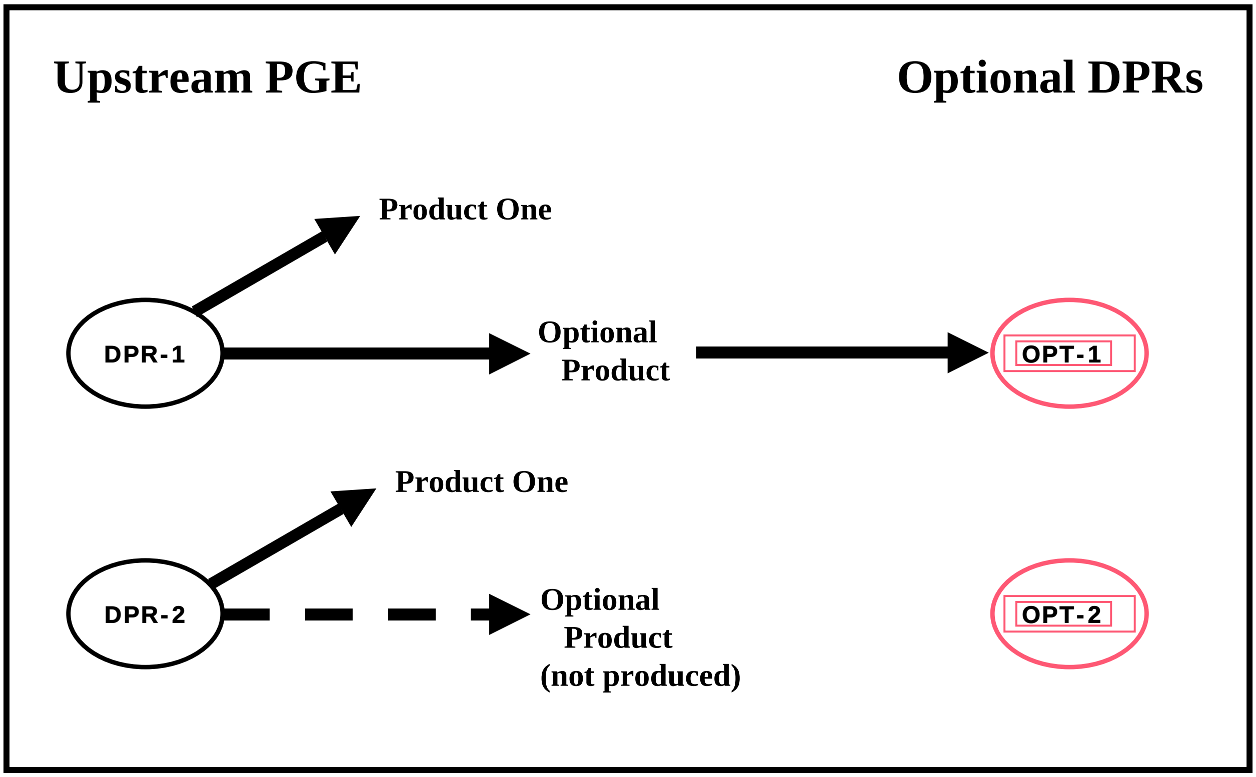

Figure 9. Example of the Optional DPRs Production Rule

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, 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.

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.

The Metadata Checks and Metadata Query Production Rules work differently depending on whether the Production Request is created for a time in the past or for a time in the future.

- Past data.
  - The Production Request Editor performs the Metadata Query or Metadata Check immediately.
  - Depending on what other production rules may be applicable, if no data is found to match the "query" (or data is found that does **not** match the "check"), the DPR fails to be created.
- Future data.
  - The Metadata Query/Metadata Check is put off until the DPR Data Collection Stop Time passes **plus** the value defined in the ODL for QUERY_DELAY.
  - The delay allows the Metadata Query/Metadata Check to be put off until it's likely that all matching data will be present.

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 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.

Data Day is actually an addition to the Metadata Query Production Rule involving **runtime parameter** values. Two settings (i.e., "Start Data Day" and "End Data Day") allow a PGE to perform a Metadata Query for the start of the Data Day and the end of the Data Day. The Start Data Day and End Data Day are calculated by subtracting twelve (12) hours from the starting day for which the PGE is executing and adding twelve (12) hours onto the ending day for which the PGE is running. So if the PGE is running from 00:00:00 on 07/04 to 00:00:00 07/05 then the Start Data Day = 07/03 12:00:00 and the End Data Day = 07/06 12:00:00.

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 10 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.

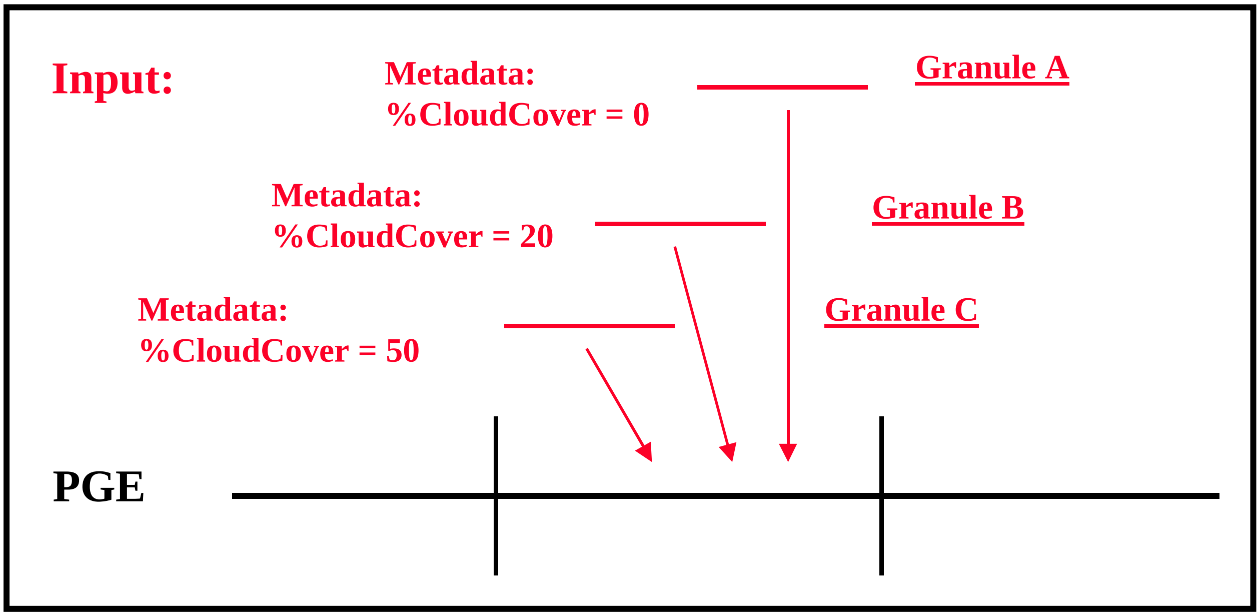

Figure 10. Example of the Metadata Checks and Query Production Rules

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 A 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.

PGE Science Metadata ODL File Parameters

At the main ODL level (not inside any objects) both the Metadata Checks and Metadata Query Production Rules use the following parameter to determine when to perform the check/query:

- QUERY_DELAY

The parameter is a positive integer (number of seconds) that defines the amount of time to wait before performing the Metadata Query or Metadata Check. When planning DPRs to run on future data, the Production Request Editor sets a timer for all Metadata Checks/Queries at the Data Collection Stop Time (end of the DPR) **plus** the value specified for the QUERY_DELAY parameter. The value adds time to the Query/Check so the PGE can be assured that all matching data will be present when the Query/Check is performed.

Both production rules use also need the following setting in the PCF_ENTRY to ensure that Planning and Data Processing consider metadata values when acquiring input data:

```
OBJECT = PCF_ENTRY
  .
  .
  QUERY_TYPE = "Metadata"
  .
END_OBJECT = PCF_ENTRY
```

QUERY_TYPE tells Planning and Data Processing how to retrieve the data for the input. For inputs based on metadata (whether Metadata Check or Metadata Query), the value is set to "Metadata".

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 or 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.

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 queries 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.

For Multi-Granule ESDT inputs the METADATA_QUERY object (consisting of entries for CLASS, PARM_NAME, OPERATOR, VALUE, DATABASE_QUERY, KEY_PARAMETER_NAME, and KEY_PARAMETER_VALUE) is replaced by the following parameter under the PCF_ENTRY object 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 addition, the input or output defined by the PCF_ENTRY must have a corresponding DISTINCT_PARAMETER entry in the appropriate ESDT science metadata ODL file.

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 and KEY_PARAMETER_VALUE are optional parameters that are used to specify the value for the container within a multi-container metadata group (i.e., the MeasuredParameters metadata group in most ESDTs). They allow PDPS to determine which container within the multi-container group is to be the object of the check or query. There must be a match between the KEY_PARAMETER_VALUE in both the ESDT science metadata ODL file and the PGE science metadata ODL file for any PCF_ENTRY that uses the ESDT.

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.

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.

Spatial Pad is an addition 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.

Due to the limitations in the current version of the SQS COTS package, the spatial region defined (i.e., the original size of the polygon combined with the size of the pad) must not exceed 60 degrees in latitude and longitude. If no spatial pad is specified or the pad is equal to zero, normal spatial query is used.

The **Spatial Query** Production Rule (unlike most time-base production rules) will work with both static and dynamic granules. Thus the Digital Elevation Models (DEMs) can be retrieved via a Spatial Query.

Figure 11 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.

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.

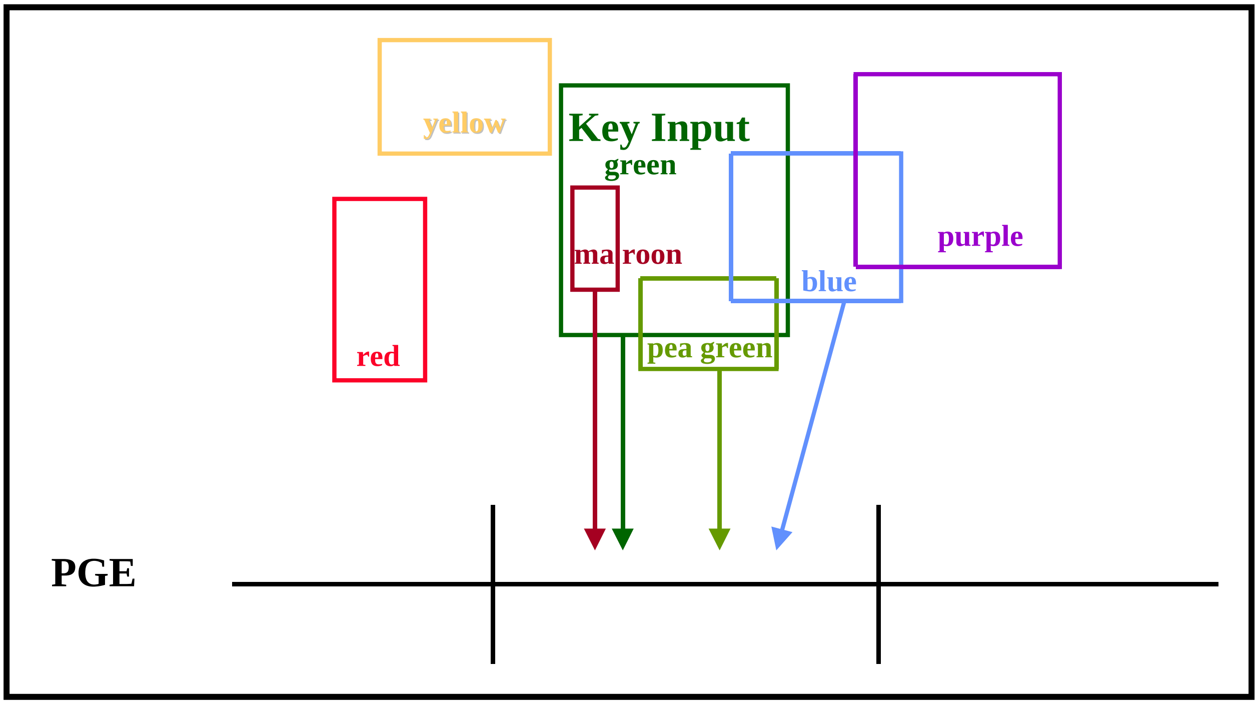

Figure 11. Example of the Spatial Query Production Rule

- 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.

Spatial Query with Spatial Pad

Figure 12 is an illustration of Spatial Query with Spatial Pad. As in Figure 11 the PGE has two input types, one of which is the key input (in green). The Spatial Pad value (the dotted line in Figure 12) is added around the key input to increase the area covered by the spatial coordinates. Then all inputs of the second ESDT are checked for overlap with the padded coordinates.

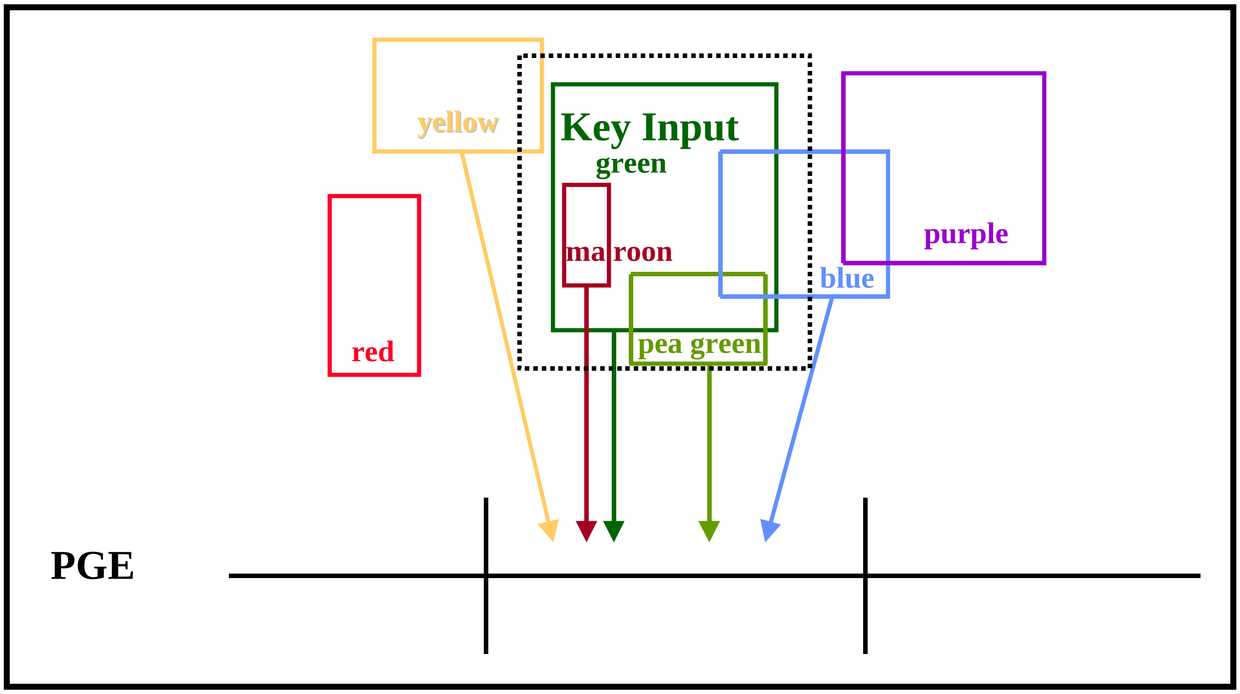

Figure 12. Example of the Spatial Query Production Rule with Spatial Pad

Assuming that all granules relate to the same time period, the granules are evaluated as follows:

- The “yellow” granule is retrieved as an input; although its spatial coordinates do not overlap with those of the key input, they do overlap the padded area (the dotted line).
- The “red” granule is not retrieved as an input because its spatial coordinates do not overlap with either the key input or the pad.
- The “blue” granule is retrieved as an input because its spatial coordinates overlap with 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 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 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 the key input. It does not matter that it overlaps with another input that is accepted, the “blue” granule.

PGE Science Metadata ODL File Parameters

Each input file is defined by a separate PCF_ENTRY object in the PGE science metadata ODL file. To mark a particular input as being subject to a Spatial Query the QUERY_TYPE parameter needs to be set to “Spatial” within that input’s PCF_ENTRY:

```
OBJECT = PCF_ENTRY
   .
   .
   .
   QUERY_TYPE = "Spatial"
   .
   .
   .
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 PCF_ENTRY object for the key input includes the following syntax:

```
OBJECT = PCF_ENTRY
   .
   .
```

KEY_INPUT = 'Y'
.
.
END_OBJECT = PCF_ENTRY

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.

If a pad is designated as an addition to the spatial constraints of the KEY_INPUT, the SPATIAL_PAD parameter must be set. The pad value is added to the key input's spatial constraints. The value for the SPATIAL_PAD parameter is set within the PCF_ENTRY defining the key input using the following syntax:

```
OBJECT = PCF_ENTRY
   .
   .
   KEY_INPUT = 'Y'
   SPATIAL_PAD =
   .
   .
END_OBJECT = PCF_ENTRY
```

The SPATIAL_PAD represents the number of kilometers (from 0.0 to 1000.0) that should be added to the Spatial Query. The value is a real number between 0 and 1000 and if not specified, it is set to 0.0 (and thus no pad is added to the spatial queries).

To allow for a temporal offset when performing a spatial query, the SPATIAL_TIME_DELTA parameter is specified to add or subtract time (in seconds) from the query. Consequently, the temporal range for the query may be different from the temporal range of the key input. The value for the SPATIAL_TIME_DELTA parameter is set within the PCF_ENTRY for the spatial input:

```
OBJECT = PCF_ENTRY
   .
   .
   .
   QUERY_TYPE = "Spatial"
   SPATIAL_TIME_DELTA =
   .
   .
END_OBJECT = PCF_ENTRY
```

The SPATIAL_TIME_DELTA adds or subtracts seconds to/from the start and stop times of the DPR when performing the spatial query for the input defined by the PCF_ENTRY. In effect it widens the temporal query. For example…

- DPR Time = 2:00:00 to 3:00:00
- SPATIAL_TIME_DELTA = 20 [seconds]
- Query time for spatial input: 1:50:40 to 3:00:20

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 optimal, 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 three values to determine what to query. The three values are concerned with the period of time to query, the direction of the query, and the number of queries allowed.

- Period (Offset).
  - Tells the PDPS software the query duration.
- Direction.
  - Indicates whether the query goes forward (positive) or backward (negative) in time.

- In the PlDataTypeReq table in the PDPS database, the Direction and the Period information are combined, so the sign of value (+ or -) indicates the Direction and the magnitude indicates the Period.

- Maximum Number of Queries.
  - Tells the PDPS software how many time periods (as defined by the Offset) to search (either forward or backward in time) for a 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 (in the specified direction) 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. However, there is a special case for the forward search; i.e., when the next search interval exceeds the current time, the search stops at the current time.

If the DPR is planned for a future time, the DPR is created using placeholder granules and a timer is activated. When the current time reaches the stop time of DPR, the timer invokes the Closest Granule method to search for the actual granules, which are then used to replace the dummy granules.

Figure 13 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 and a direction of backward, meaning that it queries back in time in six-hour intervals. The maximum number of queries 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.

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.

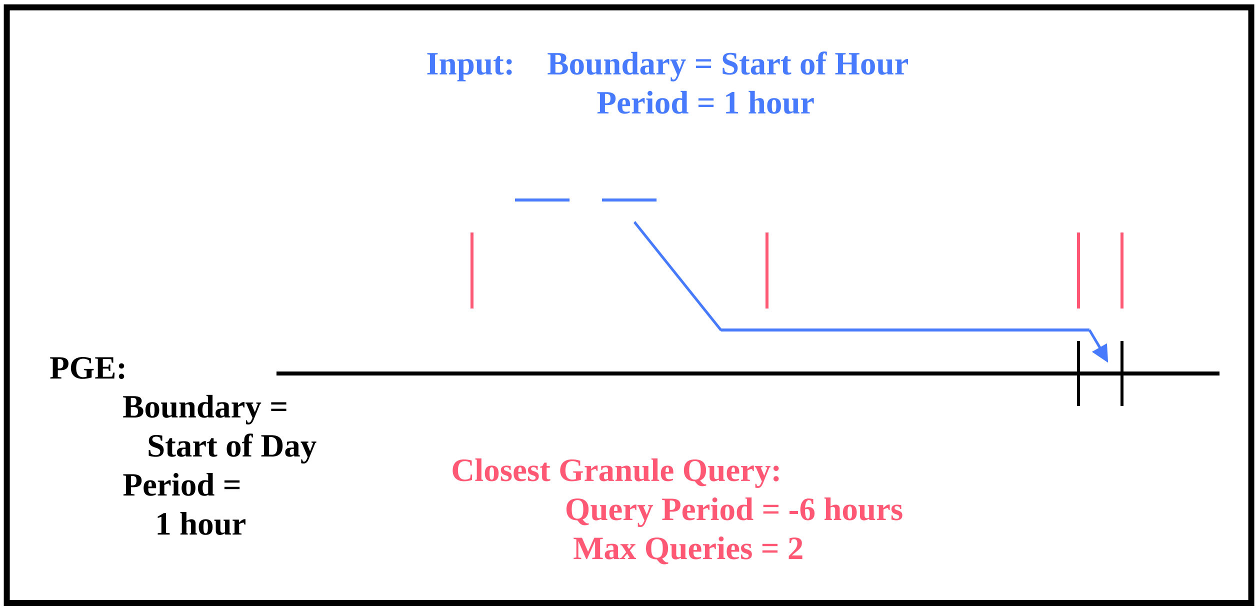

Figure 13. Example of Closest Granule Production Rule

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_DIRECTION.

As previously mentioned, if it is necessary to distinguish among granules with the same ESDT (e.g., among granules associated with different sensors for an instrument), the following parameter must be added to the PCF_ENTRY in the PGE science metadata ODL file:

- DISTINCT_VALUE.

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.

CLOSEST_QUERY_DIRECTION is the direction ("Forward" or "Backward") in time to search the CLOSEST_QUERY_OFFSET for a matching granule. The following “Direction” values are used in implementing the Closest Granule rule:

- "Forward"
  - Search forward in time (from the end of the DPR).
- "Backward"
  - Search backward in time (from the start of the DPR).

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.

The DISTINCT_VALUE (if used) must be set to the value of the metadata parameter that is used to differentiate granules within the Multi-Granule ESDT. In addition, the input or output defined by the PCF entry must have a corresponding entry identifying the metadata parameter in the ESDT science metadata ODL file.

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 Terra satellite.
  - The Terra 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.

Orbit model is a model of the satellite's orbits that allows PDPS to perform extrapolations for the Orbital Processing Production Rule. The model is a combination of a database table and a simple algorithm. The database table stores Orbit Number/Orbit Start Time/Orbital Period combinations. The algorithm uses the data to compute the same type of data relevant to subsequent orbits. For example, if Orbit Number 1000 is defined in the database table, the algorithm can calculate the start and end times for Orbits 1001, 1015, 2000, etc.

The Orbit Model works by extrapolation. It is unable to calculate data for any orbit that precedes the earliest entry in the database table. Data for the Orbit Model is specified in the Orbit Model ODL file, which is read only if the PGE requires orbital information.

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 ends within the given day.
- Granule Number within the Orbit.
  - The number of the granule within a given orbit.
  - It includes any granule that starts within the given orbit.

Figure 14 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.

In the example shown in Figure 14 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.

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.

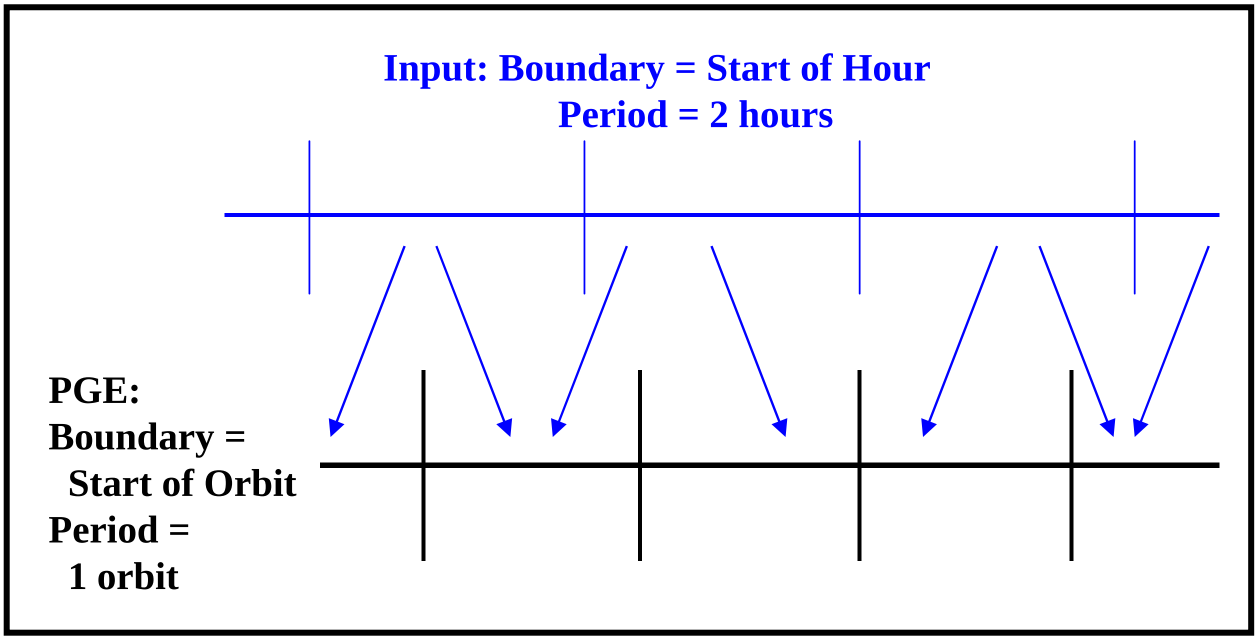

Figure 14. Example of the Orbital Processing Production Rule

- 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 is 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”.

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 ODL 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”.

Multiple DPRs for Insertion Time Production Rule

The Multiple DPRs for Insertion Time production rule allows the creation of DPRs for multiple granules with the same insertion time. It affects ASTER L1B routine processing only.

The Multiple DPRs for Insertion Time production rule is implemented when the Production Planner creates a production request using the Production Request Editor. The Production Planner enters the Duration information (Begin and End date and time) for the Insertion Time (versus Collection Time) and then ensures that the **Multiple DPRs** toggle button is depressed. If the **Multiple DPRs** toggle button is not depressed, the default rule is applied and when the production request is saved a DPR is created for the most recent granule with the specified insertion time.

The Multiple DPRs for Insertion Time production rule applies to the key input (e.g., ASTER L1B) only. It does not affect any ancillary data types (for which the most recent granule is selected).

When the Multiple DPRs for Insertion Time production rule is invoked, a DPR is generated for each key-data-type granule returned from the insertion-time query. Therefore, if multiple key-data-type granules are returned for the same collection period, multiple DPRs are generated. The DPR IDs are different in the last three characters, which are generated randomly.

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 15 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 15) 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.

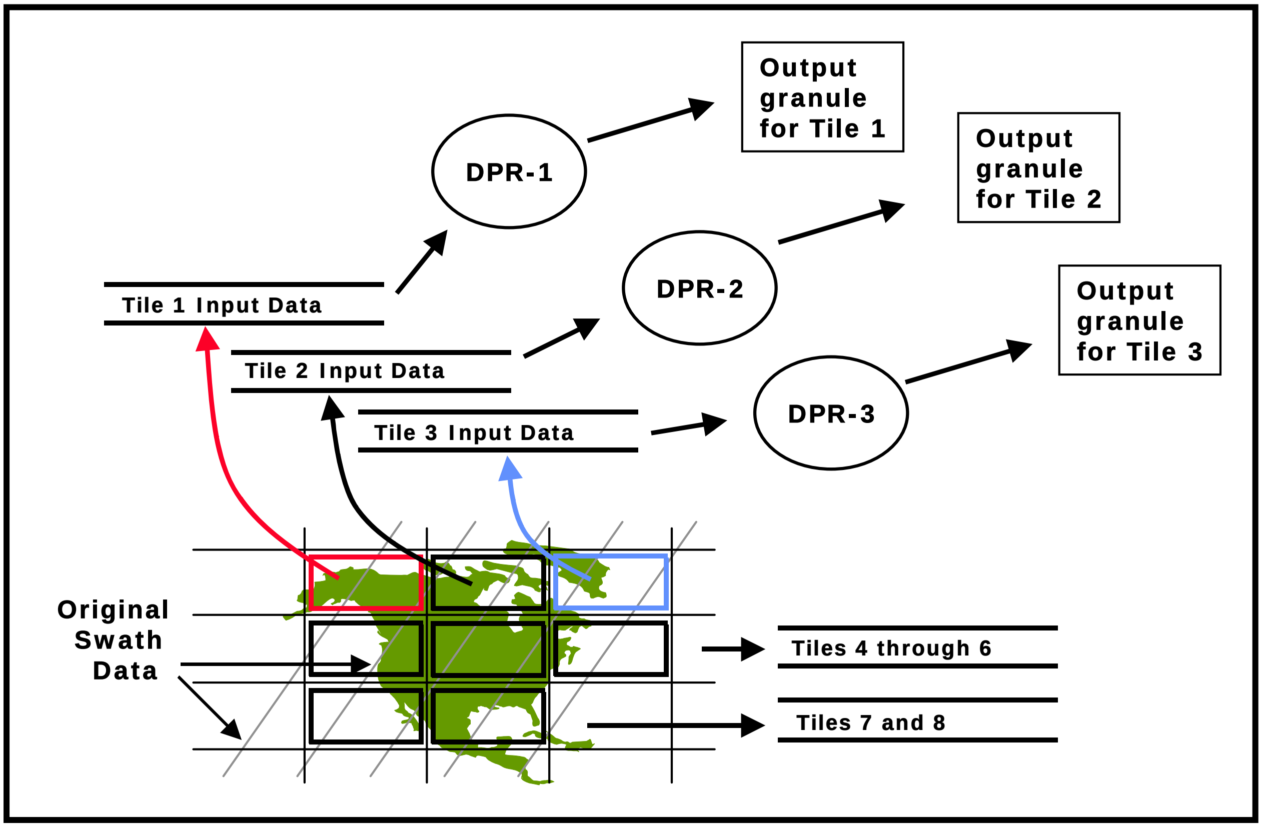

Figure 15. Example of the Tiling Production Rule

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.

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.

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-sided 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.

Production Planning Considerations

During normal operations it is expected that the Production Planner will have to add PR information to the PDPS database on a regular basis. The frequency of this activity is, to some extent, determined by the types of PGEs that run at the DAAC and by the requirements for supporting the SCF(s) responsible for the science software. PGEs vary greatly in the number of input and output files associated with them and the production rules associated with them. The amount of time it takes PLS to generate DPRs from a Production Request varies accordingly.

The following factors can affect how long it takes to create a DPR for a PGE and consequently how many DPRs should be considered for each PR:

- Number/size of input granules.
- Number/size of output granules.
- Number of times PGE should be run per day (number of DPRs per day).
- Number/characteristics of applicable Production Rules.

It may take a few minutes to generate the DPRs for a day's worth of data for one PGE; it may take several hours to generate the DPRs for a day's worth of data for another PGE. It takes some experience with creating Production Requests for a PGE to determine the optimum planning period for the PGE.

- 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 (time, snapshot, data, or orbit) 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 days or weeks, resulting in several hundred jobs being planned and run as the input data become available.
  - Early in a mission 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 DAAC of the need for any changes to the PR (e.g., discontinue the PR or modify input parameters).

- DAAC operations may have operational reasons for issuing processing requests for a limited time period.

- The Production Planner has to balance the various considerations when creating PRs.

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 changes to the resources available for use in processing.

The Production Planner should be aware of the anticipated processing requirements for the next week and month as well as the next day. The first step in the planning process is creating production requests using the Production Request Editor.

Chain Heads

A chain is a related set of DPRs in which the output of the first DPR in the chain (i.e., the "chain head") is used as input for at least one subsequent DPR. The output of the subsequent DPR(s) may be used as input for further DPR(s) and so on until the other end(s) of the chain is (are) reached. So chained DPRs must be processed in the proper sequence from the chain head to the chain's other end(s).

By default all DPRs in a chain are executed on a single computer. Consequently, most of the data for the chain are produced and consumed locally, so communication of data on network-mounted file systems is kept to a minimum.

The Production Request Editor provides a means by which the Production Planner can identify a PR as the first in a chain of PRs. The outputs of the DPR(s) that PLS creates from the chain-head PR are used as inputs to one or more subsequent DPR(s) specified in other PRs.

- Any downstream DPR (i.e., that requires the output of a preceding DPR as its input) is known as a "child" DPR.
- Any DPR that provides output for use as an input to a child DPR is a "parent" DPR.
  - Chain heads are parent DPRs.
- To completely deactivate chain processing, declare every PGE to be a chain head.
- To enable total chain processing, declare no PGE to be a chain head.
  - The "Determine Chain" script (i.e., EcPlDetermineChain.pl) finds chains starting with the DPREP PGEs.

In addition to designating chain heads in a PR, the Production Planner may select a particular virtual computer for running the chain-head DPRs. However, the selection of a virtual computer is optional. If no machine is specified, the system tries to schedule the DPR on the machine where the bulk of its accepted inputs (both static and dynamic inputs) are staged.

The implementation of chain heads includes the following factors:

- The following columns were added to the PlDataProcessingRequest and PlProductionRequest database tables in the PDPS database:
  - chainFlag.
  - scheduledMachine.
- The following column was added to the PlDataProcessingRequest database table in the PDPS database:
  - chainId.
- The Production Request Editor sets the chainFlag and scheduledMachine values based on Production Planner input when a PR is created.
- It is also possible to set the chainFlag value using the Planning Workbench GUI at plan activation time for DPRs that have no "parent" DPRs producing data used for their input.

The Planning Workbench GUI calls the "Determine Chain" script (i.e., /usr/ecs/*MODE*/CUSTOM/bin/PLS/EcPlDetermineChain.pl) when a plan is activated. At that time, chainFlag values for previously unflagged chain heads can be set and a chainId can be assigned to each DPR. If the Determine Chain script fails for some reason, the plan is not activated.

Intermittent Activation

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 Intermittent Activation 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 Intermittent Activation is illustrated in Figure 16:

- 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.
  - Sixth DPR runs.
  - Four DPRs (seventh through tenth) are skipped.
  - Eleventh DPR runs.

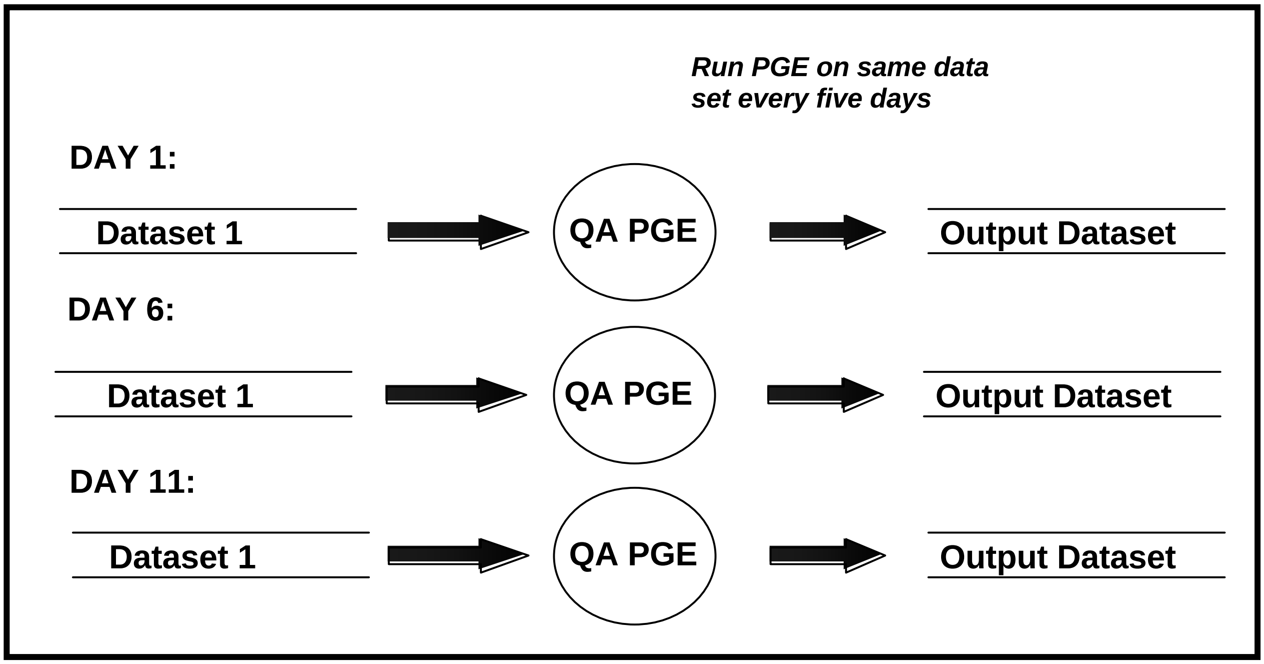

Figure 16. Example of Intermittent Activation

– Remaining three DPRs (twelfth through fourteenth) are skipped.

Data Preprocessing (DPREP)

DPREP (data preprocessing) consists of sets of PGEs that use a statistical approach to convert Level 0 (L0) ephemeris and attitude ancillary data for a particular satellite (e.g., Terra, Aqua, or Aura) into SDP Toolkit native binary format and HDF format without altering or modifying the scientific content of the granules.

Ephemeris data provide the following types of information:

- Spacecraft location: ephemeris (or orbit) data include: latitude, longitude, and height.

Attitude data provide the following types of information:

- Orientation of the satellite, including yaw, pitch, and roll angles; and angular rates about the three axes.

DPREP PGEs are supplied by the EMD Project, unlike most PGEs, which are provided by the Science Computing Facilities that EMD supports. Release 7 DPREP supports Terra, Aqua, and Aura operations.

Terra DPREP

Terra DPREP consists of the following three PGEs:

- EcDpPrAm1EdosEphAttDPREP_PGE.

- EcDpPrAm1FddAttitudeDPREP_PGE.
- EcDpPrAm1FddEphemerisDPREP_PGE.

Operationally, EcDpPrAm1EdosEphAttDPREP_PGE and EcDpPrAm1FddAttitudeDPREP_PGE are scheduled daily and run independently of one another. EcDpPrAm1FddEphemerisDPREP_PGE is scheduled and run on an as-needed basis.

The first PGE (EcDpPrAm1EdosEphAttDPREP_PGE) produces both *ephemeris* and *attitude* data. However, the *attitude* data from the first PGE is not sure to be of good enough quality for science data processing. The second PGE (EcDpPrAm1FddAttitudeDPREP_PGE) processes *attitude* data from FDD that is of good enough quality for science data processing. If the data quality analysis performed by the first PGE indicates that the quality of the *ephemeris* from the spacecraft is poor, DPREP initiates (by submitting a subscription) the request for replacement *ephemeris* data from FDD. The replacement ephemeris is processed by the third PGE, EcDpPrAm1FddEphemerisDPREP_PGE.

There are several sources of information on the Terra DPREP PGEs and how to run them:

- 500-EMD-001, Terra Spacecraft Ephemeris and Attitude Data Preprocessing.
- 611-EMD-001, Release 7.11 Mission Operation Procedures for the EMD Project, Chapter 26.
- Two files installed on the science processor hosts (e.g., e0spg11 or l0spg11) in the /usr/ecs/*MODE*/CUSTOM/data/DPS directory.
  - "AM1_DPREP_README"
  - "HowToRunAm1DPREP"

The Terra DPREP PGEs process Level 0 Terra (AM-1) spacecraft data (e.g., ESDT AM1ANC) provided by EDOS. In addition, the Terra DPREP PGEs use Terra FDD Attitude (AM1ATTF) data and may use Terra FDD Ephemeris (AM1EPHF) data. The output files/granules of the DPREP PGEs are subsequently used in the processing of science data from various instruments on the satellite. DPREP output granules provide valuable data concerning the location and orientation of the satellite when the data from the satellite's instruments were collected.

DPREP processing is granule-oriented; the processing interval selects data granules from the archive for DPREP to process. Then the granules get processed to completion. All data (i.e., ephemeris and attitude data) that are processed by Terra DPREP arrive in two-hour segments. Therefore, processing intervals are selected in multiples of two hours.

QA analysis is an important function of DPREP. It includes checking the continuity of the ephemeris and attitude data streams across data segments; i.e., the segment that is being processed and the *immediately* preceding and following segments. DPREP does this by performing consistency, limit, and data gap checks that bridge segment boundaries. The checks are initiated using data from the end of the *immediately* preceding segment and completed using data from the *immediately* following segment. When performing QA analysis on records close to a granule boundary, the QA window extends into the preceding or following granule as circumstances dictate. Consequently, Terra DPREP needs nominal access to the granules that immediately precede and follow the "current" granule.

Terra Ephemeris Processing

EcDpPrAm1EdosEphAttDPREP_PGE consists of a script that coordinates the following three DPREP executables:

- EcDpPrAm1EdosAncillary.
- EcDpPrAm1EdosEphemerisRepair.
- EcDpPrAm1ToolkitToHdf.

EDOS L0 Ancillary processing must run at least two hours behind "real time." As previously mentioned DPREP performs consistency checks across granule boundaries and requires the L0 ancillary granule that follows the "current" L0 ancillary granule in order to perform the consistency check. So EcDpPrAm1EdosAncillary reads in the "current" EDOS L0 Ancillary granule (ESDT AM1ANC) and the next AM1ANC granule. EcDpPrAm1EdosAncillary also reads in ephemeris and attitude data (ESDT AM1EPHN0 and ESDT AM1ATTN0) from the preceding run of the PGE to support QA analysis.

EcDpPrAm1EdosAncillary identifies data to be repaired in the ephemeris data stream. EcDpPrAm1EdosAncillary writes the ephemeris granules as temporary granules, not as Production Data Sets. If EcDpPrAm1EdosAncillary determines that the ephemeris data stream had repairable data quality problems, EcDpPrAm1EdosEphemerisRepair performs data repair on the temporary Toolkit-format ephemeris granule to produce the repaired granule. EcDpPrAm1ToolkitToHdf completes the process by generating an HDF-format EDOS ephemeris from the repaired Toolkit-format EDOS ephemeris granule. The granules that EcDpPrAm1EdosEphemerisRepair and EcDpPrAm1ToolkitToHdf produce are written as Production Data Sets. Temporary granules do not get archived, but do remain in the run-time directory for a short period of time in case they need to be examined.

EcDpPrAm1EdosEphAttDPREP_PGE produces Toolkit- and HDF-format attitude (ESDTs AM1ATTN0 and AM1ATTH0) and ephemeris (ESDTs AM1EPHN0 and AM1EPHH0) granules.

NOTE: EDOS-supplied ephemeris data are the primary source of ephemeris for Terra. However, EDOS-supplied attitude data are not the primary source of attitude for Terra. Attitude data supplied by the Flight Dynamics Division (FDD) are considered the primary source of Terra attitude.

EcDpPrAm1EdosAncillary performs a full complement of data quality analyses on the EDOS ephemeris data. In contrast EDOS-supplied attitude data are subject to minimal quality checks and never undergo data repair because EDOS-supplied attitude data are not considered good enough for science data processing. EDOS attitude data are "use at own risk" data; the data recommended for science data processing are the FDD attitude data routinely preprocessed by EcDpPrAm1FddAttitudeDPREP_PGE.

Terra FDD Attitude Processing

FDD attitude processing must run at least two hours behind “real time.” DPREP performs consistency checks across granule boundaries and requires the FDD Attitude granule that follows the “current” FDD Attitude granule to perform the consistency check. Accordingly, EcDpPrAm1FddAttitudeDPREP_PGE reads in both the “current” FDD attitude granule (AM1ATTF) and the next FDD attitude granule. It also reads in the attitude granule (AM1ATTNF) it produced with its last run. The output of the process is a native-format (Toolkit) attitude file (AM1ATTNF) and an HDF-format attitude file (AM1ATTHF). A metadata file is produced for each attitude data file.

Because the FDD attitude data are extensively preprocessed by FDD prior to processing by DPREP, DPREP performs minimal data quality checking on the FDD Attitude data stream (primarily to catch transmission errors). DPREP does not perform data repair of FDD attitude data.

Terra Data Repair

Data repair is performed on the EDOS-supplied *ephemeris* data stream only. If EcDpPrAm1EdosEphAttDPREP_PGE finds too many missing data points in the ephemeris data (e.g., AM1EPHH0 and AM1EPHN0 granules have gaps of 58 records or more - or about 60 seconds), DPREP requests an FDD replacement granule (AM1EPHF) by submitting a subscription to the Spatial Subscription Server. When the FDD ephemeris granule has been ingested, EcDpPrAm1FddEphemerisDPREP_PGE preprocesses the FDD Ephemeris granule to produce the FDD Toolkit ephemeris replacement granule. The granule replaces a Toolkit-format EDOS ephemeris granule directly within the EDOS Toolkit ephemeris data stream. Consequently, FDD ephemeris processing does not generate an ESDT that stands apart from the Toolkit-format EDOS ephemeris ESDT, but instead produces a different “flavor” of ephemeris data within the Toolkit-format EDOS ephemeris data stream. Furthermore, in addition to the Toolkit-format (AM1EPHN0) FDD ephemeris granule EcDpPrAm1FddEphemerisDPREP_PGE produces an HDF-format (AM1EPHH0) FDD ephemeris granule.

Because FDD performs preprocessing on replacement data, DPREP performs minimal replacement ephemeris data quality checking (primarily to catch transmission errors). DPREP does not perform data repair on replacement ephemeris.

Terra DPREP Profiles

As previously mentioned DPREP processing has data requirements beyond the current data segment. Data from the preceding and following segments are used for performing consistency checks on the ephemeris and attitude data streams when the data streams bridge segment boundaries. However, there is no guarantee that data from the preceding and following segments will always be available because adjacent segments may be in different granules. Consequently, four data processing profiles have been developed for each of the DPREP PGEs to accommodate the various permutations of data availability:

- Profile 1 is used when data are available from the preceding, current, and following segments.
- Profile 2 is used when data are available from the current and following segments only.
- Profile 3 is used when data are available from the preceding and current segments only.
- Profile 4 is used when data are available from the current segment only.

Profile 1 (data are available from the preceding, current, and following segments) is used for nominal DPREP operation. It is the profile of each DPREP step that is run on a routine basis. It is quite flexible in that it can often proceed even when there is no granule from the immediately preceding segment or from the following segment.

Profile 2 (no preceding data, but following data is available) is the boot-up process. It is used for initializing DPREP processing of the ephemeris and attitude data streams. After Profile 2 has been run on a data segment, Profile 1 (preceding and following data available) assumes processing responsibility on all data segments thereafter until data dropout or mission end is encountered.

Profile 3 (preceding data available, but no following data) processes the data segment that immediately precedes data dropout and, therefore, terminates processing on the ephemeris and attitude data streams.

Profile 4 is used for processing isolated data segments and is not likely to be scheduled operationally.

In the big picture of the mission, DPREP processing on the very first data segment would require running Profile 2 (boot-up). The next data segments would be processed using Profile 1 (nominal) processes. The very last data segment of the mission could be processed using Profile 3. However, given the processing flexibility of Profile 1, neither the scheduling of Profile 3 nor the scheduling of Profile 4 is envisioned operationally.

Aqua DPREP

Aqua DPREP consists of the following processes:

- EcDpPrPm1FddEphemerisDPREP_PGE - FDD Ephemeris Processing.

- EcDpPrPm1AttitudeDPREP_PGE - ECS Mission Operations Segment (EMOS) Attitude Processing.

There are several sources of information on the Aqua DPREP PGEs and how to run them:

- 500-EMD-002, Aqua Spacecraft Ephemeris and Attitude Data Preprocessing.
- 611-EMD-001, Release 7.11 Mission Operation Procedures for the EMD Project, Chapter 26.
- "HowToRunPm1DPREP" file installed on the science processor hosts (e.g., e0spg11 or l0spg11) in the /usr/ecs/*MODE*/CUSTOM/data/DPS directory.

Aqua DPREP includes the processing of FDD ephemeris and EMOS attitude data:

- Ephemeris data is received from the FDD in ephemeris data files.
  - Ephemeris data arrives at the DAAC daily about eight to ten hours after the end of the UTC day.
- The Ground-Based Attitude Determination (GBAD) data that is used in the processing of attitude data is received from EMOS in "carry-out" files.
  - GBAD carry-out files (PMCOGBAD) are used in conjunction with ephemeris data to prepare Aqua attitude data.

DPREP processing is granule-oriented; the processing interval selects data granules from the archive for DPREP to process. Then the granules get processed to completion. Ephemeris granules that are processed by Aqua DPREP consist of 24-hour segments while attitude granules processed by Aqua DPREP consist of two-hour segments.

Operationally, Aqua Ephemeris and Attitude PGEs are scheduled daily and run independently of one another. However, ephemeris is always run first on any given data segment because attitude processing depends on ephemeris data to complete its processing.

For both Aqua DPREP PGEs, the size of the QA window is three records. A window of three data points contains the record undergoing QA analysis and the immediately preceding and following records. When performing QA analysis on records close to a granule boundary in the ephemeris and attitude data streams, the QA window extends into the preceding or following data granule as circumstances dictate. Consequently, Aqua DPREP needs nominal access to the granules that immediately precede and follow the "current" granule.

Aqua Ephemeris Processing

Aqua ephemeris processing includes reformatting FDD ephemeris granules into Toolkit native format and HDF format. In addition, ephemeris metadata records are generated for the product granules.

The Aqua Ephemeris DPREP PGE (EcDpPrPm1FddEphemerisDPREP_PGE) processes Aqua FDD ephemeris granules (i.e., PM1EPHD). The output files/granules of

EcDpPrPm1FddEphemerisDPREP_PGE provide satellite ephemeris data and are subsequently used in the processing of Aqua satellite attitude data.

Because of the preprocessing that FDD performs on ephemeris data, DPREP performs minimal data quality checking. The data quality analyses include continuity, consistency, range, and data gap checking, which are intended mainly to catch transmission errors. DPREP does not perform limit (spike) checking or data repair on FDD ephemeris. Any failure of data quality analysis (e.g., data gap or consistency problem) detected in the FDD ephemeris triggers a request for a replacement granule. In such cases a subscription for the FDD ephemeris replacement granule is submitted to the Spatial Subscription Server.

For the sake of timely FDD ephemeris processing, Aqua DPREP typically foregoes gap checking and continuity checking of the FDD ephemeris data timeline/data stream between the segment being processed and the one immediately following. Rather than wait 24 hours for the FDD ephemeris granule from the following segment to become available, DPREP omits gap checking and continuity checking at the following segment boundary (i.e., it reverts to Profile 3 processing). Consequently, DPREP can detect such a data gap or ensure continuity only when processing of the *following* segment completes (e.g., during the next day's run of EcDpPrPm1FddEphemerisDPREP_PGE).

Note that there is no need to forgo gap checking or continuity checking during FDD ephemeris **reprocessing** because all granules on the timeline should be available in the archive.

During processing EcDpPrPm1FddEphemerisDPREP_PGE reads in the "current" FDD PM1EPHD ephemeris granule, the immediately preceding ephemeris granule, and (if available) the immediately following ephemeris granule. In addition, it reads in a previous PM1EPHND preprocessed Aqua platform ephemeris granule in native format. The outputs of the process are preprocessed Aqua platform ephemeris granules in native format (PM1EPHND) and HDF format (PM1EPHHD). A metadata file is produced for each output data file.

Aqua Attitude Processing

Aqua Attitude DPREP (EcDpPrPm1AttitudeDPREP_PGE) inputs are from ESDTs PMCOGBAD (EMOS-supplied Ground-Based Attitude Determination carry-out files) and PM1EPHND (preprocessed Aqua platform ephemeris data in native format). The outputs of Aqua Attitude DPREP are PM1ATTHR (Aqua attitude data in HDF-EOS format) and PM1ATTNR (Aqua attitude data in native format).

The attitude DPREP stream executes (nominally) twelve times per day at two-hour intervals. The PMCOGBAD data is in the form of two-hour granules (12 granules per day). Each PM1EPHND (ephemeris) granule represents 24 hours of data (one granule per day).

The EMOS-supplied attitude consists of two data streams that are subject to QA analyses:

- Attitude data stream.

- Guidance, Navigation, and Control (GN&C) Status Word 2 data stream.
  - Contains the mode that the on-board attitude system was in (i.e., mode zero, attitude hold, sun hold, fine point, earth point, or sun point) when the attitude of the Aqua platform was recorded (at eight-second intervals).

Because the attitude data are preprocessed by EMOS prior to processing by DPREP, DPREP performs minimal data quality checking on the EMOS attitude data stream. As with ephemeris data processing, the attitude data quality analyses include continuity, consistency, range, and data gap checking, which are intended mainly to catch transmission errors. DPREP does not perform limit (spike) checking or data repair. Any failure of data quality analysis (with the exception of long data gaps) results in entering a subscription for a replacement granule from EMOS. The presence of a long gap in the EMOS attitude data timeline does not cause DPREP to request replacement data from EMOS. The gap is flagged as a long gap and remains as such in the EMOS attitude data timeline.

In general EMOS attitude processing requires raw input from the preceding segment in order to complete continuity checks between the preceding and current data segments on the Status Word 2 data stream that is imbedded within the EMOS-supplied attitude granules. Raw input is also required from the following data segment in order to complete continuity checks between the current and following data segments on the attitude data stream as well as the Status Word 2 data stream.

Because DPREP's data quality analysis includes checks for continuity across granule boundaries, EMOS attitude processing must lag at least two hours behind "real time." However, due to exceptional Aqua DPREP processing requirements, the lag is usually much longer. DPREP expects the EMOS-supplied attitude granule that follows the segment being processed to be available for continuity checking (hence the processing lag).

Interpretation of the attitude data depends on the value of the GN&C Status Word 2 contained in the carry-out file GBAD data. In general if the value of Status Word 2 represents "fine point" mode, the attitude is acceptable for Aqua science processing. Any other value indicates inadequate attitude accuracy for Aqua science data processing.

During processing EcDpPrPm1AttitudeDPREP_PGE reads in the previous, current, and next EMOS-supplied attitude granules (PMCOGBAD). It reads in the ephemeris (PM1EPHND) granule for the same time period (and adjacent granule if near a granule boundary). It also reads in the attitude granule (PM1ATTNR) produced during the previous run. The output of the process is a native-format attitude file (PM1ATTNR) and an HDF-format attitude file (PM1ATTHR). A metadata file is produced for each output data file.

Aqua Data Repair

Because FDD and EMOS have preprocessed the ephemeris and attitude data streams, data repair has been deemed unnecessary and is not performed.

Aqua DPREP Profiles

As mentioned with regard to Terra DPREP processing Aqua DPREP data requirements extend beyond the current segment. Data from the preceding and following segments are used in performing consistency checks on the ephemeris and attitude data streams when the data streams bridge segment boundaries. However, there is no guarantee that data from the preceding and following segments will always be available because adjacent segments may be in different granules. Consequently, four data processing profiles have been developed for each of the Aqua DPREP PGEs to accommodate the various permutations of data availability. The Aqua DPREP profiles were developed in accordance with the same principles as the corresponding Terra DPREP profiles. (For further information refer to the section on **Terra DPREP Profiles**.)

Aura DPREP

Operationally, Aura DPREP is very similar to Aqua DPREP. Aura DPREP consists of the following processes:

- EcDpPrAuraEphemerisDPREP_PGE - FDD Ephemeris Processing.
- EcDpPrAuraAttitudeDPREP_PGE - EMOS Attitude Processing.

There are several sources of information on the Aura DPREP PGEs and how to run them:

- 500-EMD-003, Aura Spacecraft Ephemeris and Attitude Data Preprocessing.
- 611-EMD-001, Release 7.11 Mission Operation Procedures for the EMD Project, Chapter 26.
- “HowToRunAuraDPREP” file installed on the science processor hosts (e.g., e0spg11 or l0spg11) in the /usr/ecs/*MODE*/CUSTOM/data/DPS directory.

Aura DPREP includes the processing of FDD ephemeris and EMOS attitude data:

- Ephemeris data is received from the FDD in ephemeris data files.
  - Ephemeris data arrives at the DAAC daily about eight to ten hours after the end of the UTC day.
- The Ground-Based Attitude Determination (GBAD) data that is used in the processing of attitude data is received from EMOS in "carry-out" files.
  - GBAD carry-out files (AUCOGBAD) are used in conjunction with ephemeris data to prepare Aura attitude data.

DPREP processing is granule-oriented; the processing interval selects data granules from the archive for DPREP to process. Then the granules get processed to completion. Ephemeris granules that are processed by Aura DPREP consist of 24-hour segments while attitude granules processed by Aura DPREP consist of two-hour segments.

Operationally, Ephemeris and Attitude PGEs are scheduled daily and run independently of one another. However, ephemeris is always run first on any given data segment because attitude processing depends on ephemeris data to complete its processing.

For both Aura DPREP PGEs, the size of the QA window is three records. A window of three data points contains the record undergoing QA analysis and the immediately preceding and following records. When performing QA analysis on records close to a granule boundary in the ephemeris and attitude data streams, the QA window extends into the preceding or following data granule as circumstances dictate. Consequently, Aura DPREP needs nominal access to the granules that immediately precede and follow the "current" granule.

Aura Ephemeris Processing

Aura ephemeris processing includes reformatting FDD ephemeris granules into Toolkit native format and HDF format. In addition, ephemeris metadata records are generated for the product granules.

The Aura Ephemeris DPREP PGE (EcDpPrAuraEphemerisDPREP_PGE) processes Aura FDD ephemeris granules (i.e., AUREPHMF). The output files/granules of EcDpPrAuraEphemerisDPREP_PGE provide satellite ephemeris data and are subsequently used in the processing of Aura satellite attitude data.

Because of the preprocessing that FDD performs on ephemeris data, DPREP performs minimal data quality checking. The data quality analyses include continuity, consistency, range, and data gap checking, which are intended mainly to catch transmission errors. DPREP does not perform limit (spike) checking or data repair on FDD ephemeris. Any failure of data quality analysis (e.g., data gap or consistency problem) detected in the FDD ephemeris triggers a request for a replacement granule. In such cases a subscription for the FDD ephemeris replacement granule is submitted to the Spatial Subscription Server.

For the sake of timely FDD ephemeris processing, Aura DPREP typically foregoes gap checking and continuity checking of the FDD ephemeris data timeline/data stream between the segment being processed and the one immediately following. Rather than wait 24 hours for the FDD ephemeris granule from the following segment to become available, DPREP omits gap checking and continuity checking at the following segment boundary (i.e., it reverts to Profile 3 processing). Consequently, DPREP can detect such a data gap or ensure continuity only when processing of the *following* segment completes (e.g., during the next day's run of EcDpPrAuraEphemerisDPREP_PGE).

Note that there is no need to forgo gap checking or continuity checking during FDD ephemeris **reprocessing** because all granules on the timeline should be available in the archive.

During processing EcDpPrAuraEphemerisDPREP_PGE reads in the "current" FDD AUREPHMF ephemeris granule, the immediately preceding ephemeris granule, and (if available) the immediately following ephemeris granule. In addition, it reads in a previous AUREPHMN preprocessed Aura platform ephemeris granule in native format. The outputs of the process are preprocessed Aura platform ephemeris granules in native format (AUREPHMN) and HDF format (AUREPHMH). A metadata file is produced for each output data file.

Aura Attitude Processing

Aura Attitude DPREP (EcDpPrAuraAttitudeDPREP_PGE) inputs are from ESDTs AUCOGBAD (EMOS-supplied Ground-Based Attitude Determination carry-out files) and AUREPHMN (preprocessed Aura platform ephemeris data in native format). The outputs of Aura Attitude DPREP are AURATTH (Aura attitude data in HDF-EOS format) and AURATTN (Aura attitude data in native format).

The attitude DPREP stream executes (nominally) twelve times per day at two-hour intervals. The AUCOGBAD data is in the form of two-hour granules (12 granules per day). Each AUREPHMN granule represents 24 hours of data (one granule per day).

The EMOS-supplied attitude consists of two data streams that are subject to QA analyses:

- Attitude data stream.
- GN&C Status Word 2 data stream.

Because the attitude data are preprocessed by EMOS prior to processing by DPREP, DPREP performs minimal data quality checking on the EMOS attitude data stream. As with ephemeris data processing, the attitude data quality analyses include continuity, consistency, range, and data gap checking, which are intended mainly to catch transmission errors. DPREP does not perform limit (spike) checking or data repair. Any failure of data quality analysis (with the exception of long data gaps) results in entering a subscription for a replacement granule from EMOS. The presence of a long gap in the EMOS attitude data timeline does not cause DPREP to request replacement data from EMOS. The gap is flagged as a long gap and remains as such in the EMOS attitude data timeline.

In general EMOS attitude processing requires raw input from the preceding segment in order to complete continuity checks between the preceding and current data segments on the Status Word 2 data stream that is imbedded within the EMOS-supplied attitude granules. Raw input is also required from the following data segment in order to complete continuity checks between the current and following data segments on the attitude data stream as well as the Status Word 2 data stream.

Because DPREP's data quality analysis includes checks for continuity across granule boundaries, EMOS attitude processing must lag at least two hours behind "real time." However, due to exceptional Aura DPREP processing requirements, the lag is usually much longer. DPREP expects the EMOS-supplied attitude granule that follows the segment being processed to be available for continuity checking (hence the processing lag).

Interpretation of the attitude data depends on the value of the GN&C Status Word 2 contained in the carry-out file GBAD data. In general if the value of GN&C Status Word 2 represents "fine point," "attitude hold," or "earth point" mode, the attitude is acceptable for Aura science processing. Any other value indicates inadequate attitude accuracy for Aura science data processing.

During processing EcDpPrAuraAttitudeDPREP_PGE reads in the previous, current, and next EMOS-supplied attitude granules (AUCOGBAD). It reads in the ephemeris (AUREPHMN) granule for the same time period (and adjacent granule if near a granule boundary). It also reads in the attitude granule (AURATTN) produced during the previous run. The output of the process is a native-format attitude file (AURATTN) and an HDF-format attitude file (AURATTH). A metadata file is produced for each output data file.

Aura Data Repair

Because FDD and EMOS have preprocessed the ephemeris and attitude data streams, data repair has been deemed unnecessary and is not performed.

Aura DPREP Profiles

As mentioned with regard to Terra and Aqua DPREP processing Aura DPREP data requirements extend beyond the current segment. Data from the preceding and following segments are used in performing consistency checks on the ephemeris and attitude data streams when the data streams bridge segment boundaries. However, there is no guarantee that data from the preceding and following segments will always be available because adjacent segments may be in different granules. Consequently, four data processing profiles have been developed for each of the Aura DPREP PGEs to accommodate the various permutations of data availability. The Aura DPREP profiles were developed in accordance with the same principles as the corresponding Terra DPREP profiles. (For further information refer to the section on **Terra DPREP Profiles**.)

ASTER Expedited Data Processing

The system supports the ingest of ASTER expedited granules (EDSs) from the EOS Data and Operations System (EDOS) and their immediate availability to selected ASTER Scientists. In addition, the system supports the production of ASTER Level 1A/1B expedited data products from the ingested expedited granules. The higher-level expedited data products are produced at the LP DAAC.

The overall process for ASTER expedited data processing occurs as follows:

- ASTER expedited granules are selected and produced.
  - The ASTER instrument control center determines the set of scenes that are to be selected for expedited handling by the ground system.
  - The command loads to the spacecraft are created and uplinked to the spacecraft to identify the data that is subject to expedited handling.
  - The data are separated from the routine downlink data by EDOS, processed into expedited granules (EDSs) and routed to the Goddard Space Flight Center (GSFC) Earth Sciences (GES) DAAC ahead of the routine production granules (PDSs).
- ASTER EDS data granules are transferred to the Land Processes (LP) DAAC from the GES DAAC via the cross-DAAC transfer mechanism in response to a subscription for the data.

- The ASTER Ground Data System (GDS) produces a daily Observation Schedule File (OSF) (including the ASTER Observation Schedule and the ASTER One-Day Schedule), reflecting the planned operations of the ASTER instrument.
  - The OSF file is sent daily to the LP DAAC (outside the context of ECS at the DAAC).
  - The OSF is processed by the ASTER OSF Parser tool at the LP DAAC, again outside the ECS context.
    - The output of the processing is referred to as Parsed OSF (POSF) data files.
- At the LP DAAC, the POSF files and associated metadata are inserted into the ECS system via the Science Investigator-Led Processing Systems (SIPS) interface mechanism.
  - The ESDT shortname is AST_POSF.
- The Production Planner uses the Production Request Editor to delete all successfully completed L1AE and L1BE DPRs from PDPS.
- The Production Planner runs the ASTER Expedited Planning Script (i.e., "GetGranTimes.pl").
  - The script queries the SDSRV for the temporal metadata of any AST_EXP and AST_L1AE granules that were inserted to SDSRV since the last time the script was run.
  - The output of the script is a set of candidate DPR start/stop times.
  - The script writes the actual granule times discovered to a log file for further review if needed.
- The Production Planner uses the Production Request Editor to generate a set of Production Requests (PRs) for both the L1AE PGE and the L1BE PGE using the DPR start/stop times that were identified using the GetGranTimes.pl script.
  - ASTER Expedited Level 1A/1B Processing (L1AE/L1BE) PGEs must have been installed/configured previously at the LP DAAC. L1AE/L1BE PGEs are configured to run with Collection Time (rather than Insertion Time) production rules.
  - If the matching AST_POSF granule has not been inserted, a L1AE PR is still generated with a subscription for AST_POSF.
  - The L1BE PGEs being planned are not those corresponding to the L1AE PGEs being planned.
    - The L1AE PGEs must be finished and their DPRs deleted from PDPS prior to planning the corresponding L1BE PGEs.

- The Production Planner uses the Planning Workbench to create a new production plan, merging the newly created L1AE and L1BE PRs with any other PRs from the currently active plan.
- The Production Planner activates the new plan, allowing the L1AE and L1BE DPRs to be released for execution.
  - The DAAC may find it appropriate to assign higher priority to the expedited PGEs.
- Subsequent processing, archiving and access activities are the same as for other data types.

ASTER On-Demand Processing

The system supports requests for on-demand high-level production. The requests can be divided into the following categories:

- Pre-defined (standard) ASTER high-level products.
- High-level products with nonstandard input parameters.
- Digital Elevation Model (DEM) products.

In any case, the requester selects the granule to be used and/or the additional non-standard parameters to generate the high-level product. Then the requester submits a request for the system to create the high-level product.

ASTER Standard or Non-Standard On-Demand Processing

The system provides support for the processing of ASTER L1A and L1B data to higher information levels via requests for on-demand processing. A request for on-demand processing may require a sequence of algorithms to be run on the specified data. Granules produced by on-demand processing are not permanently archived.

- The requester uses the EOS Data Gateway (EDG) web client to submit ASTER On-Demand Product Requests via the ECS V0 Gateway.
- The On-Demand Manager (EcPlOdMgr) in PLS manages on-demand orders received from the EDG (via the V0 Gateway).
  - On-Demand Manager is also known as the On-Demand Product Request Manager or ODPRM.
  - Upon receipt of an order, the On-Demand Manager determines the order type. The order type can be a standard higher-level product order, a non-standard L1B, or a Digital Elevation Model (DEM).
  - Once the order type is determined, the On-Demand Production Request class verifies that the inputs provided by the EDG are valid.

- The ODPRM periodically checks each order it is tracking to determine if all of the required inputs are available. If all required inputs for a given order are present, the ODPRM marks all of the corresponding DPRs as “completed processing.”
- If the order is for a standard higher-level product, the ODPRM creates the PRs and DPRs necessary to fill the order. If all input data are available, the production request is added to the list of on-demand requests maintained by the manager and the DPR(s) are submitted to the Data Processing Subsystem.

- Job Management (EcDpPrJobMgmt) gets the activated on-demand DPRs from PLS.
  - If the on-demand DPRs are waiting for data to arrive or if there are not enough processing resources, Job Management places them in the On-Demand Queue.
  - When all input data are available and there are adequate processing resources, the on-demand DPRs are released to AutoSys for processing.
- Execution Management (EcDpPrEM) sends updates to MSS concerning the status of the on-demand jobs.

ASTER On-Demand DEM Production

Requests for ASTER on-demand DEM production are submitted to the system through the EDG and are forwarded to the ASTER DEM SIPS for processing. The ASTER DEM SIPS is located in close proximity to the LP DAAC.

When the ASTER DEM SIPS has created the requested product, ECS ingests it and distributes the product to the requester.

- The ASTER scientist submits a request for ASTER on-demand DEM production to ECS through the EDG.
- Upon receipt of an on-demand order, ODPRM determines the order type.
- The ODPRM submits a subscription (to the subscription server) to be notified when the ASTER DEM product has been inserted in the archive.
- The ODPRM sends an e-mail message to the ASTER DEM Operator.
- The ASTER DEM Operator creates the DEM in accordance with the order information.
- When the requested product has been created, the ASTER DEM operator puts a PDR file on the PDR server.
  - The PDR file identifies the location of the ASTER DEM data file and the corresponding metadata file.
- ECS ingests the DEM product using the polling-with-delivery-record interface.

- Upon successful insertion of the DEM product in the archive, the science data server triggers the appropriate "insert event" qualified with the universal reference (UR) of the inserted granule.
- The subscription server queries the subscription database to determine which subscriptions need to be activated (fired).
- The status of the on-demand request is updated to “Processing Complete.”
- ODPRM receives subscription notification and sends a "user acquire" request to the science data server to have the on-demand product sent to the requester.
- The DEM product is sent to the requester via the requested medium.
  - The data may be sent via ftp or on a physical medium (e.g., tape).

Reprocessing

The term "reprocessing" is used to indicate the following two situations:

- A PGE has been run in the past and produced a product and for some unspecified reason (e.g., a file has been corrupted/deleted) the PGE needs to be run again to reproduce the exact same outputs.
- A PGE that was run in the past over a given time period or set of data has been improved (software or static inputs) by the Instrument Team, who would like to have the modified PGE run on the data from the previously processed time period/granule.

The second case is the more important. It leads to the requirement that the system support a workload for reprocessing that is equal in size to the current or first-time processing load.

PGE changes leading to reprocessing may occur for any of the following reasons:

- An error that must be corrected may have been discovered in the software.
- An improved algorithm may be found based on an improved understanding of the instrument or physical phenomena.
- Static files, such as calibration data, may need to be updated for several reasons including compensation for instrument degradation.
- Changes to software design may be required to incorporate new or different ancillary data files.
- Changes to production rules may be needed.
- Changes to or addition of run-time parameters may be needed.

Note also that changes to an upstream PGE resulting in a new or improved product very likely mean that a downstream PGE has to be modified if for no other reason than to insure that the correct, updated version of the upstream product is picked up for processing.

Operations concepts associated with reprocessing include the following considerations:

- A PGE that is being used for reprocessing represents a new version of a PGE.
  - The new version of the PGE must go through the SSI&T process.
  - PGEs may be updated several times in the course of the mission.
- A reprocessing PGE may specify new static or dynamic inputs rather than new PGE software.
  - The new version of the PGE must go through the SSI&T process and must be identified with a new PGE version tag.
  - The PGE appears to PDPS as a completely new PGE with a distinct profile.
- Output granules from reprocessing PGEs are represented under the existing ESDT used for the previous generation or version of the data granules.
- The distinction between generations of the data can be made through the ReprocessingActual attribute that is associated with each granule.
  - Other mechanisms for distinguishing reprocessing levels include the ProductionDateTime and PGEVersion attributes.
  - The instrument team associated with the PGE must supply the value for the ReprocessingActual attribute.
- DPRs specifying the exact same PGE may be scheduled for both reprocessing and routine processing in the same production plan.
  - For example, first-time processing may have been carried out for the time period from Day 1 to Day N, at which point an improved PGE is introduced.
  - The improved PGE is used for processing from Day N+1 forward, and is also used concurrently in reprocessing from Day 1 to Day N.
- It is expected that in a typical planning session the Production Planner will plan time ranges for reprocessing that are comparable in extent with the routine processing.
  - For example, for routine processing the Production Planner will plan the processing for the data acquired during the most recent 24-hour period.
  - The Production Planner will plan the processing for the next 24-hour period from the time span of data that needs to be reprocessed.
  - During later phases of the mission (e.g., Launch + 24 months), the operator might plan the next 48-hour period from the time span of data that needs to be reprocessed.

- DPRs created via planning are organized and managed in separate queues for processing and reprocessing.
  - A fixed number of processing "slots" are reserved for…
    - On-Demand processing.
    - Routine processing.
    - Reprocessing.
  - DPRs are dispatched for processing from either queue against the appropriate slot when the slot becomes available, i.e., as a job of that slot's type finishes and the input data become available.
  - Operations personnel can configure the number of slots of each type.
  - Operations personnel can change the number of slots of each type (and as a result the total number of concurrent processing slots) at run time.
  - Operations personnel have to change the number of slots of each type; there is no dynamic slot allocation (e.g., based upon unused slots for some period of time).

Regeneration

The reason for regenerating granules is to produce replacements for previously generated granules that have been lost or corrupted due to failure in the archive. The overall process involves the following general operations:

- Retrieval of the Production History files (PH) for lost granules to determine parameters for the generation of replacement granules.
- Creating Production Requests for the generation of replacement granules.
- Creating and activating a Production Plan that includes the Production Requests for the generation of replacement granules.
- Preparing (if applicable) a "PDPS Residual Granules List," which identifies granules that either cannot or should not be regenerated at the DAAC.
  - Some granules do need not be reproduced; e.g., if there is a more recent version of the product available.

The regeneration process is initiated when the Production Planner receives a list of "Granules for PDPS Re-Generation." The list contains information about the granules to be regenerated and Universal References (URs) for the associated Production History tar files. The list is the product of a Science Data Server (SDSRV) procedure concerning SDSRV Retrieval of Granule Production History Metadata.

The following considerations apply to the regeneration of granules in response to the loss of files from the archive:

- When regenerating lost granules, all outputs of the PGE [not just those equivalent to the lost granule(s)] are to be produced and archived.
- There is no guarantee that when a PGE is re-run it will use the same inputs as were used during the original execution of the PGE; consequently, the output may be different from the original granule(s).
  - The variability of: Optional/Alternate inputs, Metadata Checks, Metadata Query and other production rules affects PGE output.
  - It is possible that at the time of the original run of the PGE, certain optional/alternate inputs were not available, which became available later. During the re-run of the PGE use of those additional or other optional inputs cannot be avoided. However, it can be assumed that an equivalent or better product than the original will be produced as a result.
- PDPS maintains a minimal amount of granule-level versioning. By design, only the latest version of the granule is used.
  - If the PGE to be re-run uses inputs that have more than one granule-level version, PDPS uses only the latest version of those inputs.
  - However, if references to those granules have been deleted from the PDPS database (a delete script, which runs periodically, cleans up unused database entries), PDPS chooses the first one returned from SDSRV. SDSRV does not guarantee any sort of ordering in this case but PDPS selects the latest granule from those returned.
- At Production Request time, the default values for metadata checks can be overridden. The new values used are stored in the PDPS database but not in the Production History. If at the time a PGE is re-run the references to the PGE have been deleted from PDPS database, the default metadata checks are used.
  - It is possible that default metadata check values would cause the DPR not to be run; e.g., if the metadata checks are more restrictive than those used in the original run.
  - If changes to metadata checks were required in order to get DPRs to run originally, it is assumed that the values were saved as part of the PGE profile.
- For reasons of production timing or updated QA values, during regeneration a PGE subject to a metadata query could have input which is different from that used in the original processing.
  - The assumption is that regeneration will result in a better product.

- Other production rules (e.g., spatial query) could make it impossible to reproduce identical granules.
- If a PGE (PGE name, version and profile) has to support lost granule regeneration, the PGE should not be deleted from the PDPS database.
  - In the SSIT **Operational Metadata** GUI, the delete flag for the PGE should not be checked.

The procedure for **Regenerating Granules in Response to Loss of Files from the Archive** is described in a subsequent section of this lesson.

Logging in to System Hosts

Logging in to System Hosts

Logging in to system hosts is accomplished from a UNIX command line prompt. It is an initial set of steps that is performed when accomplishing many other Production Planning and Processing tasks.

Logging in to system hosts starts with the assumption that the applicable hosts are operational and the Production Planner has logged in to a workstation or X-term that has access to the applicable network in the system.

Logging in to System Hosts

NOTE: Commands in Steps 1 and 2 are typed at a UNIX system prompt.

1 At the UNIX command line prompt type **setenv DISPLAY *clientname*:0.0** then press the **Return/Enter** key.

- Use either the X terminal/workstation IP address or the machine-name for the client name.
- When using secure shell, the DISPLAY variable is set just once, before logging in to remote hosts. If it were to be reset after logging in to a remote host, the security features would be compromised.

2 In the terminal window (at the command line prompt) start the log-in to the appropriate host by typing **/tools/bin/ssh *hostname*** then press **Return/Enter**.

- The **-l** option can be used with the ssh command to allow logging in to the remote host (or the local host for that matter) with a different user ID. For example, to log in to x0pls02 as user cmops enter:

  /tools/bin/ssh -l cmops x0pls02

- Depending on the set-up it may or may not be necessary to include the path (i.e., /tools/bin/) with the ssh command. Using ssh alone is often adequate. For example:

  ssh x0pls02

  \- or -

  ssh -l cmops x0pls02

- Examples of Planning/Management Workstation host names include **e0pls03** and **l0pls02.**

- Examples of Science Processor host names include **e0spg11** and **l0spg11**.
- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.
- Examples of Access/Process Coordinators (APC) Server host names include **e0acg11** and **l0acg02**.
- Examples of Ingest Server host names include **e0icg11** and **l0acg02**.
- Examples of Sun Consolidation External Server host names include **e0ins01** and **l0ins01**.
- Examples of Sun Consolidation Internal Server host names include **e0acs11** and **l0acs03**.
- If you receive the message, **“Host key not found from the list of known hosts. Are you sure you want to continue connecting (yes/no)?”** enter **yes** (“y” alone will not work).
- If you have previously set up a secure shell passphrase and executed **sshremote**, a prompt to **Enter *passphrase* for RSA key '<*user@localhost*>'** appears; continue with Step 3.
- If you have not previously set up a secure shell passphrase, go to Step 4.

3 If a prompt to **Enter passphrase for RSA key '<*user@localhost*>'** appears, type your ***passphrase*** then press **Return/Enter**.

- If a command line prompt is displayed, log-in is complete.
- If the passphrase is unknown, press **Return/Enter**, which should cause a **<user@remotehost>’s password:** prompt to appear (after the second or third try if not after the first one), then go to Step 4.
- If the passphrase is entered improperly, a **<user@remotehost>’s password:** prompt should appear (after the second or third try if not after the first one); go to Step 4.

4 If a prompt for **<*user@remotehost*>’s password:** appears, type your ***password*** then press **Return/Enter**.

- A command line prompt is displayed.
- Log-in is complete.

Launching the Production Request Editor

Launching the Production Request Editor

The following software applications are associated with the Production Request Editor:

- Production Request Editor.
- Subscription Manager.
- Sybase ASE Server for the PDPS database.

Access to the Production Planning and Processing tools is gained through the use of UNIX commands.

Launching Production Request Editor applications starts with the assumption that the applicable servers are running and the Production Planner has logged in to the system.

Launching the Production Request Editor

1 Access a terminal window logged in to the Planning/Management Workstation host.

- Examples of Planning/Management Workstation host names include **e0pls03** and **l0pls02.**
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 Type **setenv ECS_HOME /usr/ecs/** then press the **Return/Enter** key.

- When logging in as a system user (e.g., cmshared), the ECS_HOME variable may be set automatically so it may not be necessary to perform this step.

3 Type **cd /usr/ecs/*MODE*/CUSTOM/utilities** then press **Return/Enter**.

- Change directory to the directory containing the production planning startup scripts (e.g., EcPlPRE_IFStart).
- The ***MODE*** will most likely be one of the following operating modes:
  - OPS (for normal operation).
  - TS1 (for Science Software Integration and Test (SSI&T)).
  - TS2 (new version checkout).

- Note that the separate subdirectories under /usr/ecs apply to (describe) different operating modes.

4 Type **EcPlPRE_IFStart** ***MODE*** then press **Return/Enter** to launch the **Production Request Editor** GUI.

- The **Production Request Editor** graphical user interface (GUI) introductory window (Figure 17) is displayed.

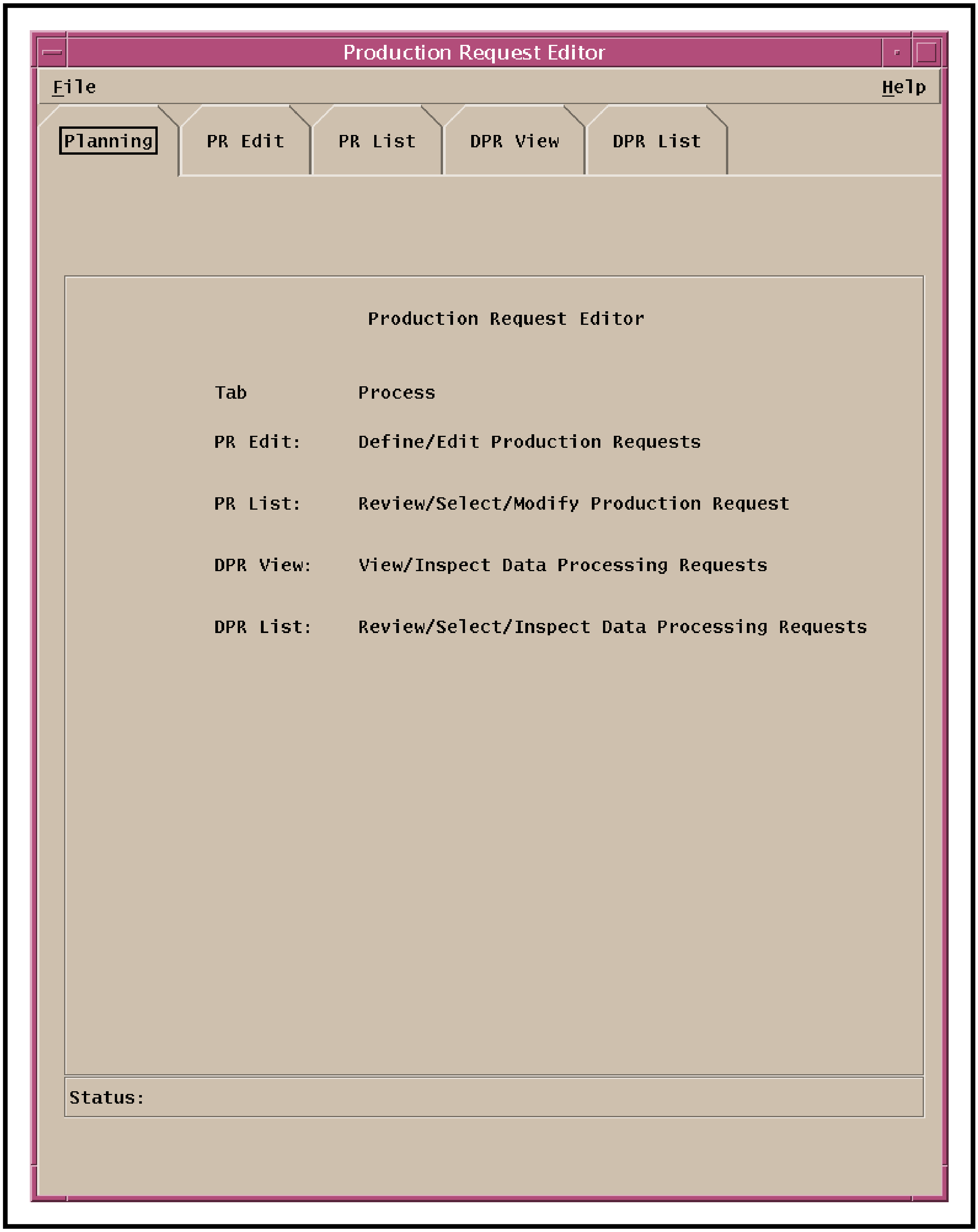

Figure 17. Production Request Editor Introductory GUI

This page intentionally left blank.

Creating/Updating/Deleting a Production Request

Creating a New Production Request Using the Production Request Editor GUI

The new Production Request process begins when the Production Planner starts the Production Request Editor graphical user interface (GUI) from a UNIX prompt. The Production Planner enters the Request Definition, PGE parameters, duration, production rule related information, and comments for the new Production Request.

Before creating the new PR the Production Planner must be prepared to provide the following types of information:

- Name of the PR.
- Priority of the PR.
- PGE to be used in processing the PR.
- Start Date.
- Start Time.
- End Date.
- End Time.
- Beginning orbit (if orbit-based).
- Ending orbit (if orbit-based).
- Production rules that affect the PGE and applicable values to be entered (if any).
- Comments (if applicable).

The procedure for creating a new production request using the Production Request Editor GUI starts with the assumption that all applicable servers and the **Production Request Editor** GUI are currently running and the **Production Request Editor** Introductory GUI (Figure 17) is being displayed.

Creating a New Production Request Using the Production Request Editor GUI

1 Click on the **PR Edit** tab.

- The **PR Edit** GUI (Figure 18) is displayed.

NOTE: The **Help** buttons at the bottom of various the **PR Edit GUI** pages are non-functional (do not work).

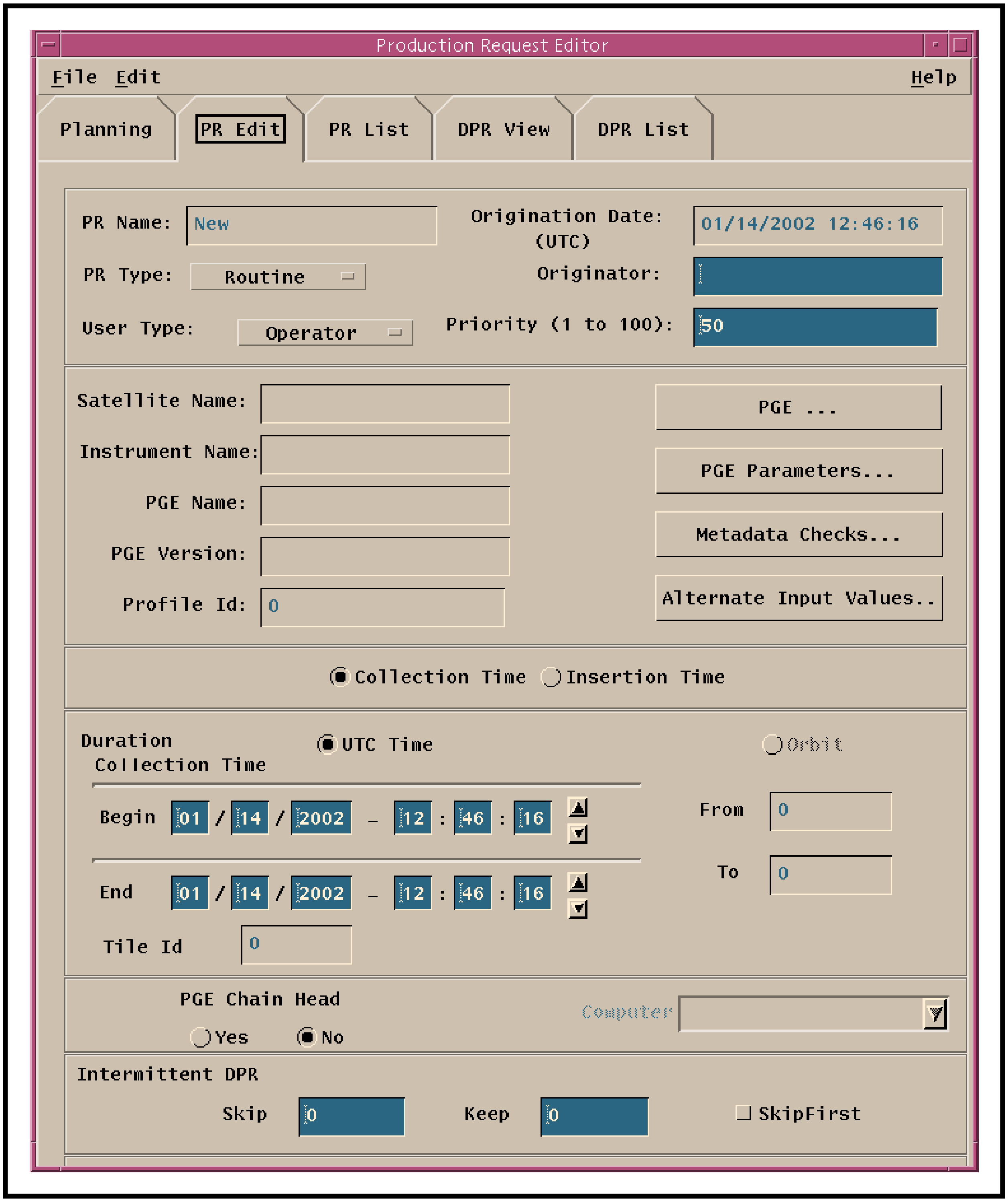

Figure 18. PR Edit GUI

2 Click and hold the **PR Type** option button to display a menu of types of production requests, move the mouse cursor to the desired selection (highlighting it), then release the mouse button.

- The following production request types are listed:
  - **Routine.**
  - **On-Demand** [not currently available for selection].
  - **Reprocessing.**

3 Click and hold the **User Type** option button to display a menu of types of users, move the mouse cursor to the desired selection (highlighting it), then release the mouse button.

- The following user types are listed:
  - **Operator.**
  - **DAAC Manager.**
  - **Scientist.**
  - **Researcher.**

NOTE: The **PR Name** and **Origination Date** fields will be filled automatically when the Production Request is saved at the end of the procedure. (You do not need to fill in these fields.)

4 Type the identification of the person creating the production request in the **Originator** field.

- Either UserID or actual name of a person may be used, depending on DAAC policy (if applicable).

5 Type the priority for the PR in the **Priority** field.

- Enter a number in the range of one (1) to ten (10).
  - One (1) has the highest priority; ten (10) has the lowest priority.
- The **Priority** field specifies the Production Request Editor priority to be included in the Data Processing Request(s) that result(s) from the Production Request.
- The Production Request Editor **Priority** is subsequently weighted according to the value specified in the Production Strategy selected from the Planning Workbench when a Production Plan is created using the Production Request.

6 Click on the **PGE...** button.

- The **PGE Selection** GUI (Figure 19) is displayed.

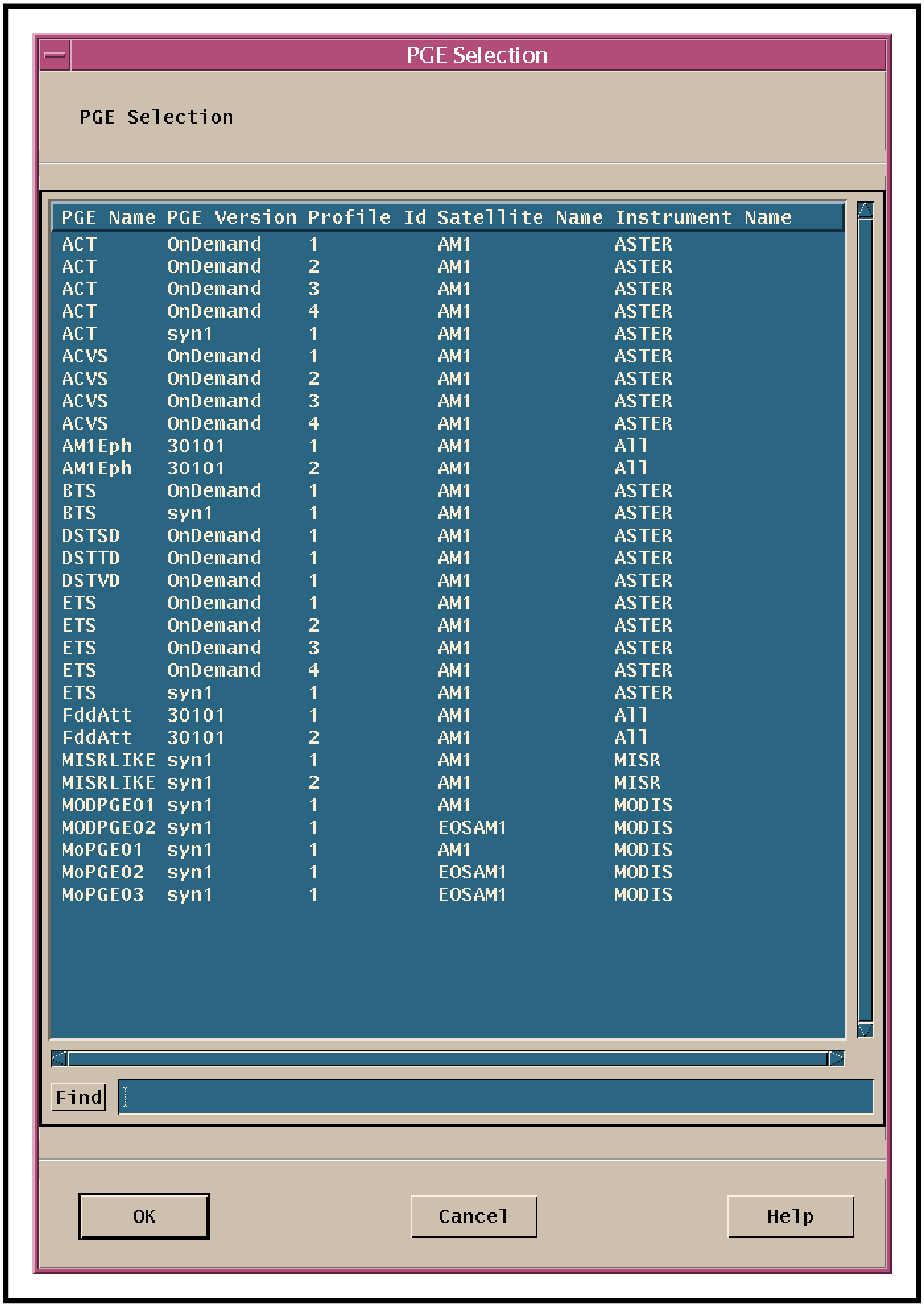

Figure 19. PGE Selection GUI

7 Select the desired PGE from the list by clicking on the appropriate row in the table.

- The PGE list is scrollable. (If there are items on the list in addition to those currently visible in the window, the additional items can be viewed by clicking on the arrows associated with the scrollbars.)
- The **Find** button provides a means of performing a keyword search of the **PGE Selection** table.
  - A keyword search is performed by typing relevant text in the text entry box to the right of the **Find** button then clicking on the **Find** button.
    - The first entry in the **PGE Selection** table that has the search text is highlighted.
    - Click on the **Find** button again (as necessary) to highlight additional occurrences of the search text.

8 Click on the appropriate button from the following selections:

- **OK** - to approve the selected PGE and dismiss the **PGE Selection** page.
  - The **Production Request - PR Edit** GUI (Figure 18) is displayed.
  - The following fields are automatically filled in on the **Production Request - PR Edit** GUI:
    - **Satellite Name.**
    - **Instrument Name.**
    - **PGE Name.**
    - **PGE Version.**
    - **Profile ID.**
- **Cancel** - to return to the **Production Request - PR Edit** GUI without specifying a PGE.
  - The **Production Request - PR Edit** GUI (Figure 18) is displayed.

9 Click on the **PGE Parameters…** button.

- The **PGE Parameter Mappings** GUI page (Figure 20) is displayed.
- The **PGE Parameter Mappings** GUI has a table that lists the following information:
  - **Parameter Name.**
  - **Logical Id.**
  - **Default Value.**

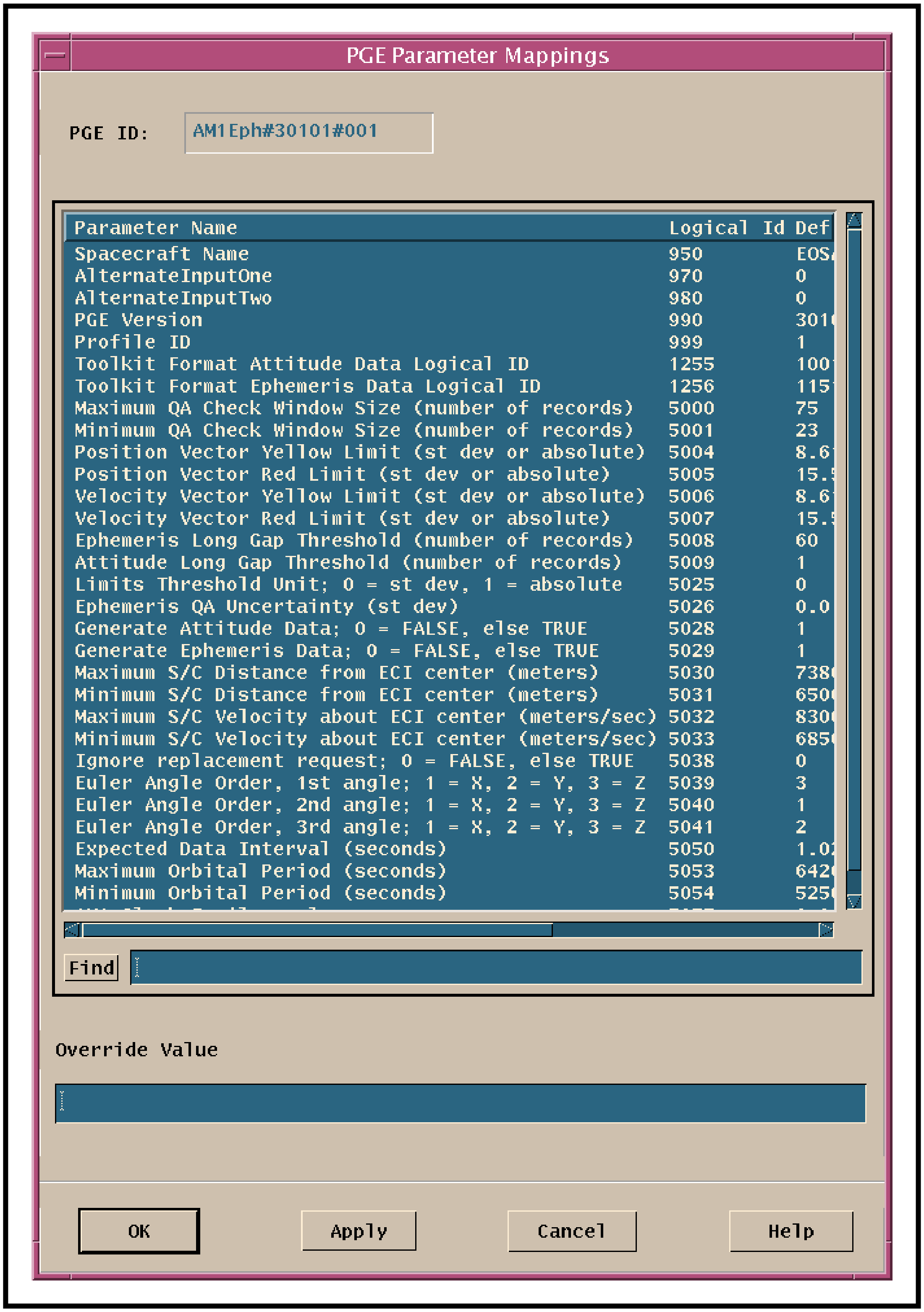

Figure 20. PGE Parameter Mappings GUI

- **Override Value.**
- **Description.**

- The PGE parameters (if any) listed in the table are the parameters relevant to the particular PGE. (Different PGEs may have different parameters.) The parameters and their default values are defined during the Science Software Integration and Test (SSI&T) process. They are included with the PGE information stored in the PDPS database. The Production Request Editor retrieves the parameter information from the database.

10 If any PGE parameter(s) should be changed, first select (highlight) a parameter to be changed by clicking on the corresponding row in the list of parameters.

- The parameter row is highlighted.
- If you do not change the parameters, the values in the **Default** column will be used.
- Whenever Profile 2 or 4 DPRs are scheduled in Terra DPREP PGE EcDpPrAm1EdosEphAttDPREP_PGE or EcDpPrAm1FddEphemerisDPREP_PGE, modify the **Initial Orbit Number** parameter (the orbit from which orbit number counting is to resume).
  - Failure to do so causes DPREP to abort with a complaint that it encountered an invalid initial orbit number (-1), which is the default initial orbit number.
- In general modify parameter values when and as directed by the customer (e.g., MODAPS) only.
  - The SCF will provide notification if PGE parameters other than the default values should be used.
- The PGE parameter mappings list is scrollable.

11 Type the desired value in the **Override Value** field.

12 Click on the **Apply** button at the bottom of the **PGE Parameter Mappings** window.

- The value in the **Override Value** column is updated.

13 If any other parameter is to be changed, repeat Steps 10 through 12.

14 Click on the appropriate button from the following selections:

- **OK** - to approve the change(s) and dismiss the **PGE Parameter Mappings** window.
  - The **Production Request - PR Edit** GUI (Figure 18) is displayed.
- **Cancel** - to return to the **Production Request - PR Edit** GUI without saving the new value(s).
  - The **Production Request - PR Edit** GUI (Figure 18) is displayed.

15 If the PGE is subject to a metadata-based production rule and the value(s) to be checked need(s) to be changed, click on the **Metadata Checks…** button.

- If the **Metadata Checks…** button has been selected, perform Steps 16 through 20 as applicable; otherwise go to Step 21.
- The **MetadataChecks** GUI page (Figure 21) is displayed.
- The **MetadataChecks** GUI has an **InputDataType** window that lists input data types.
- In addition, the **MetadataChecks** GUI has a metadata checks (**MetaDataField-Operator-Value-Type**) window in which there is a table that lists the following information concerning each metadata check:
  - **MetaDataField.**
  - **Operator.**
  - **Value.**
  - **Type.**
- Initial values for metadata checks are entered during SSI&T; however, it is possible to modify the values using the **MetadataChecks** GUI when creating a production request.

16 If it is necessary to change any value(s) for metadata checks, first select (highlight) an input data type with a value to be changed by clicking on the corresponding row in the **InputDataType** window.

- The input data type row is highlighted.
- The metadata check information for the highlighted input data type is displayed in the **MetaDataField-Operator-Value-Type** window.
- The **Find** button provides a means of performing a keyword search of the **InputDataType** window.
  - A keyword search is performed by typing relevant text in the text entry box to the right of the **Find** button then clicking on the **Find** button.
    - The first entry in the **InputDataType** window that has the search text is highlighted.
    - Click on the **Find** button again (as necessary) to highlight additional occurrences of the search text.

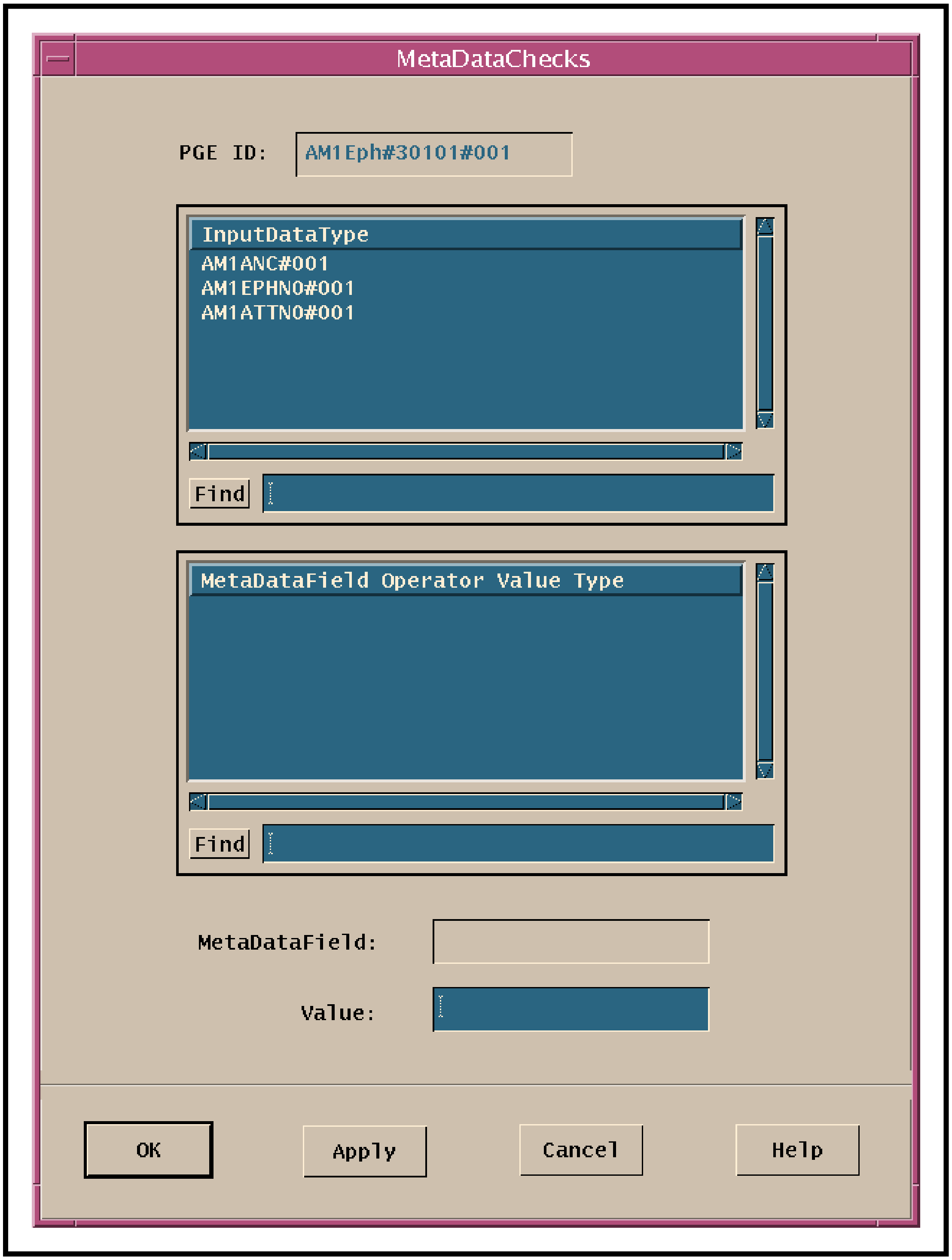

Figure 21. MetadataChecks GUI

17 Select (highlight) a metadata field with a comparison value to be changed by clicking on the corresponding row in the **MetaDataField-Operator-Value-Type** window.

- The metadata field row is highlighted in the **MetaDataField-Operator-Value-Type** window.
- The identity of the metadata field is displayed in the **MetaDataField** window.
- The **Find** button provides a means of performing a keyword search of the **MetaDataField-Operator-Value-Type** window.
  - A keyword search is performed by typing relevant text in the text entry box to the right of the **Find** button then clicking on the **Find** button.
    - The first entry in the **MetaDataField-Operator-Value-Type** window that has the search text is highlighted.
    - Click on the **Find** button again (as necessary) to highlight additional occurrences of the search text.

18 Type the new value for the metadata check in the **Value** field.

19 Click on the appropriate button from the following selections:

- **OK** - to approve the new value and dismiss the **MetadataChecks** page.
  - The **Production Request - PR Edit** GUI (Figure 18) is displayed.
  - Go to Step 21.
- **Apply** - to approve the new value without dismissing the **MetadataChecks** page.
  - Go to Step 20.
- **Cancel** - to return to the **Production Request - PR Edit** GUI without saving the new value.
  - The **Production Request - PR Edit** GUI (Figure 18) is displayed.
  - Go to Step 21.

20 If any additional value(s) to be checked need to be changed, repeat Steps 16 through 19 as necessary.

21 If the PGE is subject to the Alternate Inputs Production Rule and the timer settings or the order of alternate inputs need to be changed, click on the **Alternate Input Values…** button.

- If the **Alternate Input Values…** button has been selected, perform Steps 22 through 27 as applicable; otherwise go to Step 28.
- The **AlternateInputValues** GUI page (Figure 22) is displayed.

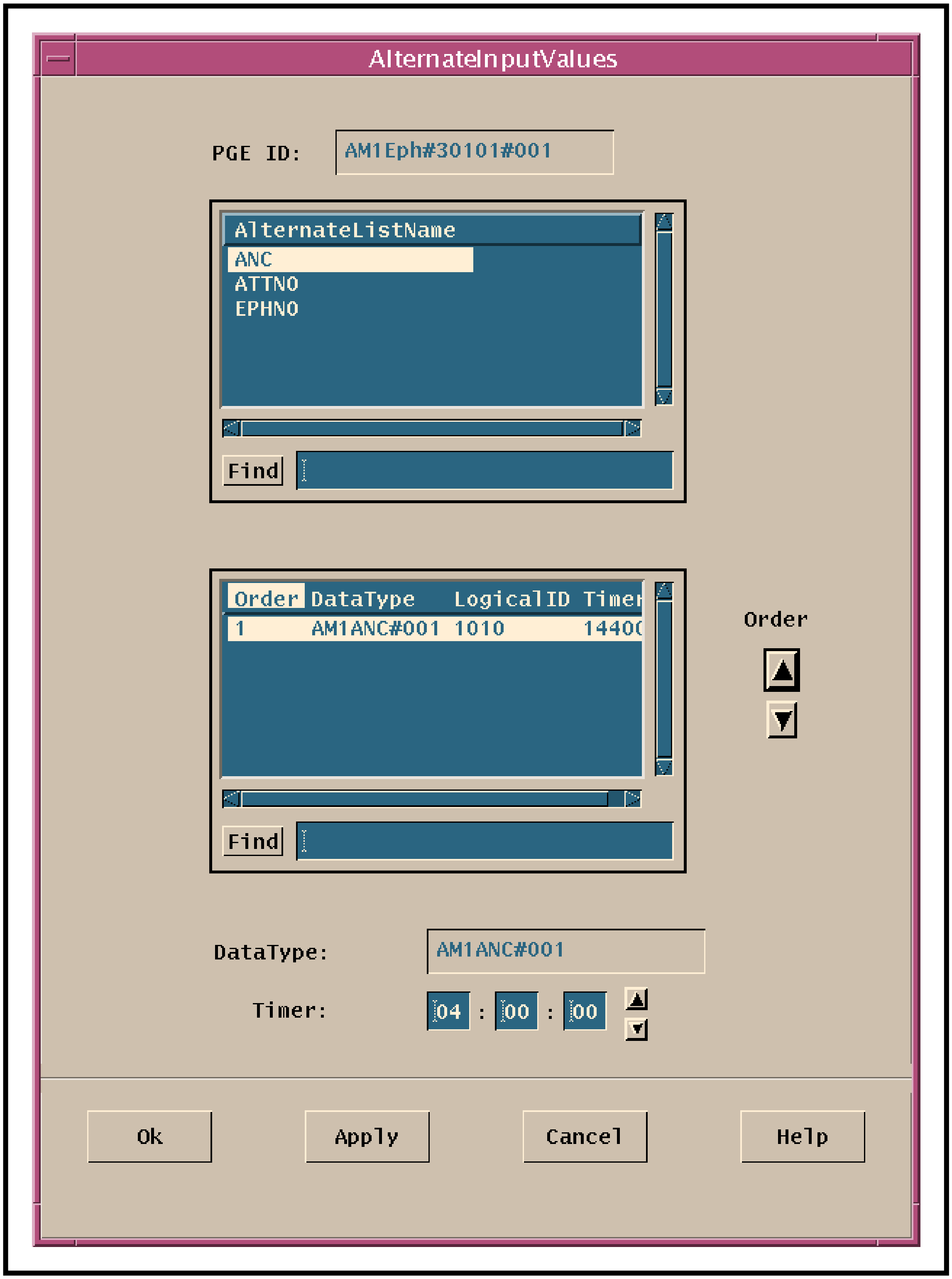

Figure 22. AlternateInputValues GUI

- The **AlternateInputValues** GUI has an **AlternateListName** window that lists the applicable alternate inputs.
- In addition, the **AlternateInputValues** GUI has an alternate input (**Order-DataType-LogicalID-Timer**) window in which there is a table that lists the following information concerning each alternate input:
  - **Order.**
  - **DataType.**
  - **LogicalID.**
  - **Timer.**
- The initial set-up for alternate inputs is entered during SSI&T; however, it is possible to modify the set-up using the **AlternateInputValues** GUI when creating a production request.

22 If it is necessary to change timer settings or the order of alternate inputs, first select (highlight) an alternate input to be changed by clicking on the corresponding row in the **AlternateListName** window.

- The alternate input row is highlighted.
- Information concerning the highlighted alternate input is displayed in the **Order-DataType-LogicalID-Timer** window.
- The **Find** button provides a means of performing a keyword search of the **AlternateListName** window.
  - A keyword search is performed by typing relevant text in the text entry box to the right of the **Find** button then clicking on the **Find** button.
    - The first entry in the **AlternateListName** window that has the search text is highlighted.
    - Click on the **Find** button again (as necessary) to highlight additional occurrences of the search text.

23 Select (highlight) an alternate input with timer settings or the order of alternate inputs to be changed by clicking on the corresponding row in the **Order-DataType-LogicalID-Timer** window.

- The alternate input row is highlighted in the **Order-DataType-LogicalID-Timer** window.
- The data type of the alternate input is displayed in the **DataType** field.

- The **Find** button provides a means of performing a keyword search of the **Order-DataType-LogicalID-Timer** window.
  - A keyword search is performed by typing relevant text in the text entry box to the right of the **Find** button then clicking on the **Find** button.
    - The first entry in the **Order-DataType-LogicalID-Timer** window that has the search text is highlighted.
    - Click on the **Find** button again (as necessary) to highlight additional occurrences of the search text.

24 If it is necessary to change the order of alternate inputs, click on the up/down arrow buttons adjacent to the **Order-DataType-LogicalID-Timer** window as necessary until the highlighted alternate input has the proper order listed in the **Order** column of the window.

- If necessary, repeat Steps 23 and 24 to change the order of additional alternate inputs.

25 If the timer setting for an alternate input is to be modified, verify that the alternate input with the timer setting to be changed has been highlighted then type the new timer setting in the **Timer** fields.

- Another method of changing timer settings (other than typing the numbers) is to click in each of the timer fields in turn and click on the up/down buttons adjacent to the **Timer** fields until the correct time is indicated.

26 Click on the appropriate button from the following selections:

- **OK** - to approve the new alternate input setting(s) and dismiss the **AlternateInputValues** GUI.
  - The **Production Request - PR Edit** GUI (Figure 18) is displayed.
  - Go to Step 28.
- **Apply** - to approve the new alternate input setting(s) without dismissing the **AlternateInputValues** GUI.
  - Go to Step 27.
- **Cancel** - to return to the **Production Request - PR Edit** GUI without saving the new alternate input setting(s).
  - The **Production Request - PR Edit** GUI (Figure 18) is displayed.
  - Go to Step 28.

27 If any additional alternate input setting(s) need to be changed, repeat Steps 22 through 26 as necessary.

NOTE: Depending on the specific PGE chosen, either the **Collection Time** or **Insertion Time** button is automatically selected. **Collection Time** (time when the data were collected by the instrument on the satellite) is used for most PGEs. The **Insertion Time** option is available primarily for ASTER processing to allow the generation of DPRs for all data contained on an ASTER tape received from the ASTER Ground Data System (GDS). If the **Insertion Time** option is selected, the **Multiple DPRs** toggle button appears on the **PR Edit** GUI as shown in Figure 23.

28 Click on either the **UTC Time** (Coordinated Universal Time) button or the **Orbit** button, depending on whether data to be processed is specified by time or orbit.

- If **UTC Time** is selected, perform Steps 29 through 31 (as applicable).
- If **Orbit** is selected go to Step 32.

29 Type the desired data start date and time (in ***MM/DD/YYYY hh:mm:ss*** format) in the **Begin** fields.

- As data are typed in each field the cursor automatically advances to the next field.
- Another method of entering date and time (other than typing the numbers) is to click in each of the date/time fields in turn and click on the up/down buttons adjacent to the date/time fields until the correct date/time is indicated.

30 Type the desired data end date and time (in ***MM/DD/YYYY hh:mm:ss*** format) in the **End** fields.

31 If the Multiple DPRs for Insertion Time Production Rule applies, ensure that the **Multiple DPRs** toggle button (Figure 23) is depressed.

- If necessary, click on the **Multiple DPRs** toggle button.
- If the Multiple DPRs for Insertion Time Production Rule does not apply, go to Step 32.

32 If the Orbital Processing Production Rule applies, type the number of the first orbit of data to be processed in the **From** field.

33 If the Orbital Processing Production Rule applies, type the number of the last orbit of data to be processed in the **To** field.

34 If the Tiling Production Rule applies, type the number of the appropriate tile in the **Tile Id** field.

35 Click on the appropriate button from the following selections under **PGE Chain Head**:

- **Yes** - to designate the DPRs resulting from the PR as chain heads.
  - If designated as chain heads, the DPRs produce outputs that are to be used as inputs to downstream DPRs.

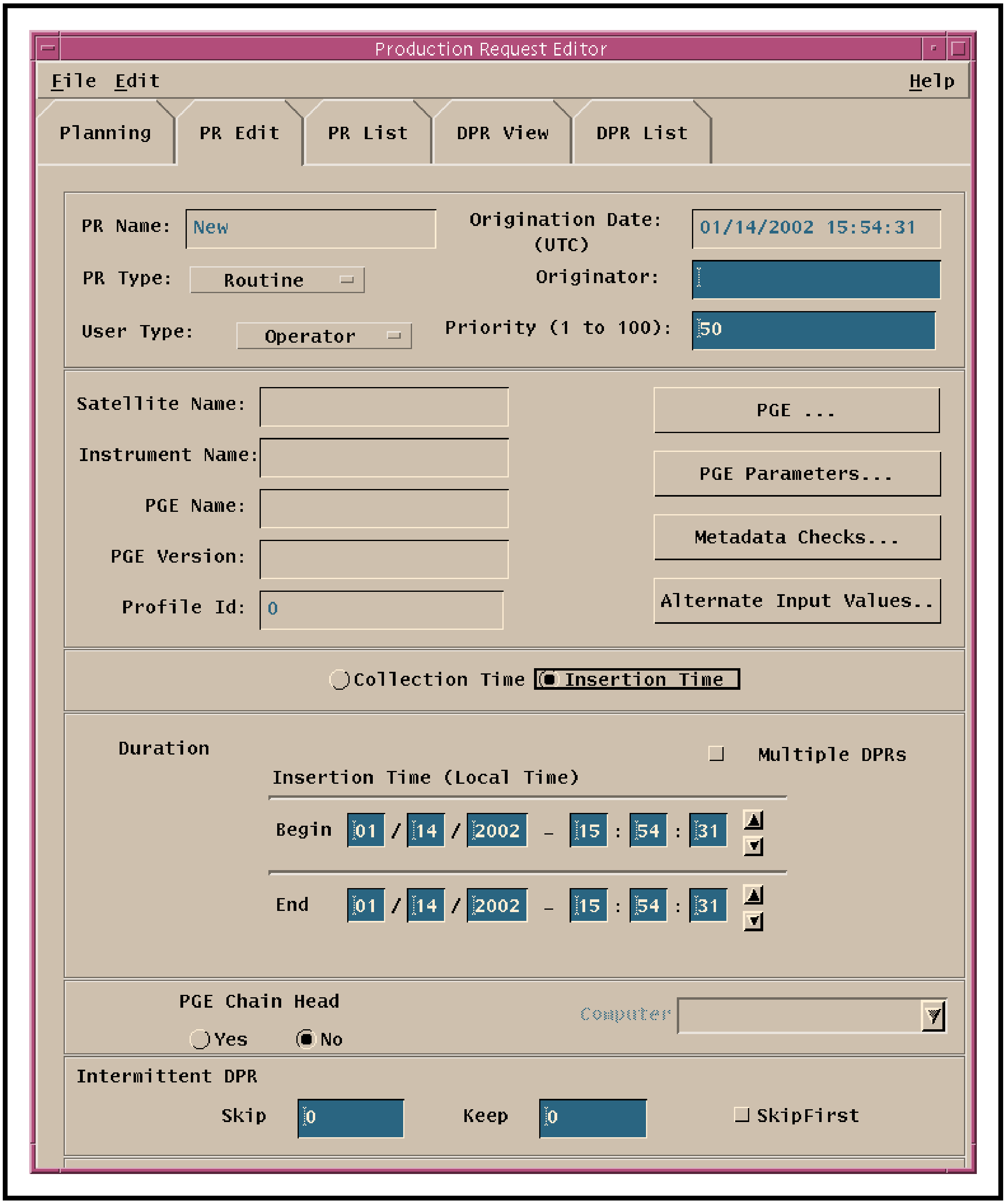

Figure 23. PR Edit GUI with Multiple DPRs Toggle Button

- **No -** to indicate that the DPRs resulting from the PR are not chain heads.

36 If the PGE specified in the PR is at the head of a chain and should be run on a particular virtual computer, click and hold the button adjacent to the **Computer** field to display a menu of available virtual computers, move the mouse cursor to the desired selection (highlighting it), then release the mouse button.

- If a particular virtual computer is selected, each DPR will be scheduled to run on the corresponding machine.
- If no virtual computer is selected, the system will try to schedule the DPRs on the machine where the bulk of the DPR's accepted inputs (both static and dynamic inputs) are already staged.

37 If Intermittent Activation applies, type the number of DPRs to skip in the **Skip** field.

- If Intermittent Activation applies, perform Steps 38 and 39.
- If Intermittent Activation does not apply, go to Step 40.

38 Type the number of DPRs to keep in the **Keep** field.

39 If the first DPR is to be skipped, click on the **SkipFirst** button.

40 Type any relevant comments in the **Comments** field.

41 Select **Save As** from the **File** pull-down menu (**File** → **Save As**).

- The **File Selection** window (Figure 24) is displayed.

42 Type a file name for the production request in the **Selection** field.

43 Click on the **OK** button to save the production request.

- Eventually a **Production Request Explosion into DPRs** dialogue box (similar to Figure 25) is displayed.
  - It may take several minutes or even hours for the process to complete.
  - If the explosion into DPRs is successful, the production request and corresponding DPR(s) are saved in the PDPS database and the Production Request **PR Name** and **Origination Date** fields are automatically updated.
  - If the explosion into DPRs is not successful, the dialogue box contains a message to that effect and it will be necessary to troubleshoot the problem. (This lesson contains a section on troubleshooting production planning problems.)

44 Click on the **OK** button to dismiss the **Production Request Explosion into DPRs** dialogue box.

- The dialogue box is dismissed.

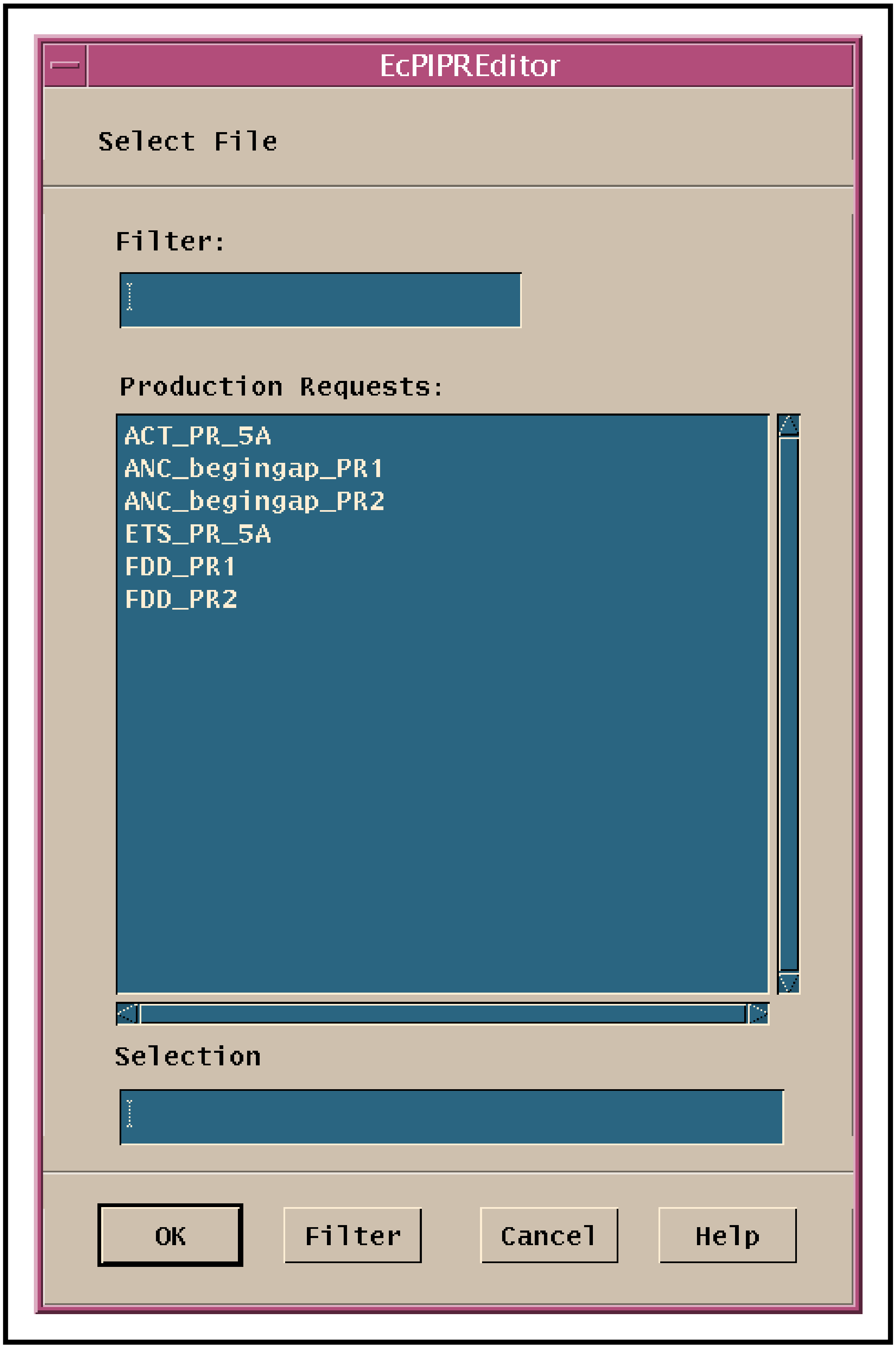

Figure 24. Production Request Editor File Selection Window

Figure 25. Production Request Explosion into DPRs Dialogue Box

45 Select **File** → **New** to clear the entries on the **Production Request Editor** GUI.

- Return to Step 2 to create another new PR.

46 To start the process of exiting from the **Production Request Editor** GUI select **File** → **Exit** from the pull-down menu.

- A **Do you really want to exit?** dialogue box is displayed.

47 Click on the appropriate button from the following selections:

- **OK** - to exit from the **Production Request Editor** GUI.
  - The **Production Request - PR Edit** GUI (Figure 18) is dismissed.
- **Cancel** - to return to the **Production Request - PR Edit** GUI.

It is also possible to create a new PR by editing or modifying an existing PR and renaming it. This is particularly quick and useful if there are only minor differences between the existing PR and the new one.

Creating New Production Requests Using the Production Request Generator (Command-Line Interface)

The process of creating new Production Requests using the Production Request (PR) Generator (EcPlPRGenerator) begins with the Production Planner preparing an input file for the PR Generator. Then the Production Planner starts the PR Generator, referencing the input file, and the PR Generator generates the appropriate PRs.

Before creating the new PRs the Production Planner must be prepared to provide the PGE ID (PgeId) and GEO ID (GEOId) values to be used in preparing the PRs. PgeId identifies a PGE. GEOId is the portion of a granule UR that identifies granule type, subtype, and database ID in the format ***type:subtype:dbID***. For example, SC:AST_ANC.001:14754 is the GEOId for a science granule of ASTER ancillary data with a database ID of 14754.

The procedure for creating new production requests using the Production Request Generator (command-line interface) starts with the assumption that all applicable servers are currently running.

Creating New Production Requests Using the Production Request Generator (Command-Line Interface)

---

1 Access a terminal window logged in to the Planning/Management Workstation host.

- Examples of Planning/Management Workstation host names include **e0pls03** and **l0pls02.**
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 Type **setenv ECS_HOME /usr/ecs/** then press the **Return/Enter** key.

- When logging in as a system user (e.g., cmshared), the ECS_HOME variable may be set automatically so it may not be necessary to perform this step.

3 Type **cd /*path*** then press **Return/Enter**.

- ***/path*** is the path (e.g., **/usr/ecs/OPS/CUSTOM/data/PLS**) specified as the value for the EcPlPRGenerator's **InfoPath** configuration parameter.
  - Refer to the EcPlPRGenerator configuration parameters in the Registry database or the EcPlPRGenerator.CFG (EcPlPRGenerator.CFG.rgy) file.

4 Type **vi *FileName*** then press **Return/Enter**.

- A new file with the specified ***FileName*** is opened.
- For example:

  x0pls01{cmops}[10]->vi PRfile

  ~

  ~

  ~

  […]

  "PRfile" [New file]

  - Many blank lines have been deleted from the example.
- The new file will specify the PgeId and GEOId values to be used in generating new Production Requests.

- Although this procedure has been written for the **vi** editor, any UNIX editor can be used to create the file.

5 Using vi editor commands create a file that specifies the PgeId and GEOId values to be used in generating new Production Requests.

- For example the following entries are included in **PRfile** (the preceding example):

  ACT#syn1#001 SC:AST_L1B.001:76677
  BTS#syn1#001 SC:AST_L1B.001:76677

  – **ACT#syn1#001** and **BTS#syn1#001** are PgeId values.

  – **SC:AST_L1B.001:76677** is a GEOId.

- The following vi editor commands are useful:

  – **h** (move cursor left).

  – **j** (move cursor down).

  – **k** (move cursor up).

  – **l** (move cursor right).

  – **a** (append text).

  – **i** (insert text).

  – **r** (replace single character).

  – **x** (delete a character).

  – **dw** (delete a word).

  – **dd** (delete a line).

  – ***n*dd** (delete *n* lines).

  – **u** (undo previous change).

  – **Esc** (switch to command mode).

6 Press the **Esc** key.

7 Type **ZZ**.

- **vi** exits and the new file is saved.

  – To exit **vi** without saving the new entries in the file type **:q!** then press **Return/Enter**.

- UNIX prompt is displayed.

8 Type **cd ../../utilities** then press **Return/Enter**.

- Change directory to the directory containing the production planning startup scripts (e.g., EcPlPRE_IFStart).

9 Type **EcPlPRGeneratorStart** ***MODE FileName*** then press **Return/Enter** to start the PR Generator.

10 When the PR Generator has completed its operation, type **cd ../logs** then press **Return/Enter**.

- Change directory to the logs directory.

11 Type **pg EcPlPRGeneratorDebug.log** then press **Return/Enter**.

- The first page of the PR Generator debug log is displayed.
- Although this procedure has been written for the **pg** command, any UNIX editor or visualizing command (e.g., **more**, **view, vi**) can be used to review the log file.

12 Review the log file to determine the results of running the PR Generator.

- For example (extract from EcPlPRGeneratorDebug.log):

  *********** THE STATUS OF PRODUCTIONREQUESTS: ************

  ACT#syn1#001 SC:AST_L1B.001:76677
  ********* The above entries in Input file Failed because of bad GEOID's*********

- To exit from **pg** at the **:** prompt enter:

  q

  – The command line prompt is displayed.

Editing or Modifying a Production Request

Editing or modifying a Production Request is typically a means of using an existing Production Request as a template for a new request.

- Examples:
  – A Production Request for processing Tuesday's data using PGE A may be used as a template for a Production Request for processing Wednesday's data using PGE A.
    · Only the beginning and ending dates would have to be changed.

- A Production Request for processing Tuesday's data using PGE A may be used as a template for a Production Request for processing Tuesday's data using PGE B.
    - Only the PGE would have to be changed.
- Whenever a Production Request is successfully saved, DPRs are generated.
  - The DPRs are consistent with the Production Request's specifications.
- It is not possible to change existing DPRs so saving a modified Production Request results in the generation of new DPRs.

To edit/modify a new Production Request, execute the procedure steps that follow. Perform only those steps of the procedure that are applicable to the changes you want to make. You do not have to go through all of the fields in the PR to successfully modify it. However, you must save the modified PR (Steps 44 - 46) to make the changes effective.

The procedure for editing/modifying a production request starts with the assumption that all applicable servers and the **Production Request Editor** GUI are currently running and the **Production Request Editor** Introductory GUI (Figure 17) is being displayed.

Editing/Modifying a Production Request

---

1 Click on the **PR Edit** tab.

- The **PR Edit** GUI (Figure 18) is displayed.

2 Select **File** → **Open** from the pull-down menu to display a list of Production Requests from which to select the PR to be edited/modified.

- The **File Selection** window (Figure 24) is displayed.

3 Select (highlight) the PR to be edited/modified by clicking on the corresponding PR name in the list of PRs.

- The selected PR is displayed in the **Selection** field.

4 Click on the **OK** button.

5 If the type of production request shown on the **PR Type** option button is not the desired type, click and hold the **PR Type** option button to display a menu of types of production requests, move the mouse cursor to the desired selection (highlighting it), then release the mouse button.

- The following production request types are available:
  - **Routine.**

- **On-Demand** [not currently available for selection].
- **Reprocessing.**

6 If the type of user shown on the **User Type** option button is not the desired type, click and hold the **User Type** option button to display a menu of types of users, move the mouse cursor to the desired selection (highlighting it), then release the mouse button.

- The following user types are listed:
  - **Operator.**
  - **DAAC Manager.**
  - **Scientist.**
  - **Researcher.**

NOTE: The **PR Name** and **Origination Date** fields will be filled automatically when the Production Request is saved at the end of the procedure. (You do not need to fill in these fields.)

7 If the identification of the person editing/modifying the production request is not already displayed in the **Originator** field, type it there.

- Either UserID or actual name of a person may be used, depending on DAAC policy (if applicable).

8 If the PR's priority is to be changed, type the new priority in the **Priority** field.

- Enter a number in the range of one (1) to ten (10).
  - One (1) has the highest priority; ten (10) has the lowest priority.
- The **Priority** field specifies the Production Request Editor priority to be included in the Data Processing Request(s) that result(s) from the Production Request.
- The Production Request Editor **Priority** is subsequently weighted according to the value specified in the Production Strategy selected from the Planning Workbench when a Production Plan is created using the Production Request.

9 If the PGE is to be changed, click on the **PGE...** button.

- The **PGE Selection** GUI (Figure 19) is displayed.
- Perform Steps 10 and 11 to designate the new PGE.

10 Select the desired PGE by clicking on the appropriate row in the table.

11 Click on the **OK** button.

- The **Production Request - PR Edit** GUI page (Figure 18) is displayed.

- The following fields are automatically filled:
  - **Satellite Name.**
  - **Instrument Name.**
  - **PGE Name.**
  - **PGE Version.**
  - **Profile ID.**

12 If any PGE parameter(s) should be changed, click on the **PGE Parameters...** button.

- The **PGE Parameter Mappings** GUI (Figure 20) is displayed.
- Modify parameter values when and as directed by the customer (e.g., MODIS) only.
  - The SCF will provide notification if PGE parameters other than the default values should be used.
- Perform Steps 13 through 15 to designate the new PGE parameter(s).

13 Select (highlight) a parameter to be changed by clicking on the corresponding row in the list of parameters.

- The parameter row is highlighted.
- If you do not change the parameters, the values in the **Default** column will be used.
- The **Find** button provides a means of performing a keyword search of the PGE Parameters table.
  - A keyword search is performed by typing relevant text in the text entry box to the right of the **Find** button then clicking on the **Find** button.
    - The first entry in the PGE Parameters table that has the search text is highlighted.
    - Click on the **Find** button again (as necessary) to highlight additional occurrences of the search text.

14 Type the desired value in the **Override Value** field.

15 Click on the **Apply** button at the bottom of the **PGE Parameter Mappings** window.

- The value in the **Override Value** column is updated.

16 If any other parameter is to be changed, repeat Steps 13 through 15.

17 Click on the **OK** button at the bottom left of the **PGE Parameter Mappings** window to approve the changes and dismiss the window.

- The **Production Request - PR Edit** GUI (Figure 18) is displayed.

18 If any metadata check value needs to be changed, click on the **Metadata Checks…** button.

- The **MetadataChecks** GUI page (Figure 21) is displayed.
- Perform Steps 19 through 23 to change the metadata check value(s).

19 Select (highlight) an input data type with a value to be changed by clicking on the corresponding row in the **InputDataType** window.

- The input data type row is highlighted.
- The metadata check information for the highlighted input data type is displayed in the **MetaDataField-Operator-Value-Type** window.

20 Select (highlight) a metadata field with a comparison value to be changed by clicking on the corresponding row in the **MetaDataField-Operator-Value-Type** window.

- The metadata field row is highlighted in the **MetaDataField-Operator-Value-Type** window.
- The identity of the metadata field is displayed in the **MetaDataField** window.

21 Type the new value for the metadata check in the **Value** field.

22 Click on the appropriate button from the following selections:

- **OK -** to approve the new value and dismiss the **MetadataChecks** GUI.
  - The **Production Request - PR Edit** GUI (Figure 18) is displayed.
- **Apply -** to approve the new value without dismissing the **MetadataChecks** GUI.
- **Cancel** - to return to the **Production Request - PR Edit** GUI without saving the new value.
  - The **Production Request - PR Edit** GUI (Figure 18) is displayed.

23 Repeat Steps 20 through 22 as necessary to change any additional value(s) to be checked.

24 If any alternate input timer settings or the order of alternate inputs should be changed, click on the **Alternate Input Values…** button.

- The **AlternateInputValues** GUI page (Figure 22) is displayed.
- Perform Steps 25 through 30 as applicable to change alternate input timer settings or the order of alternate inputs.

25 Select (highlight) an alternate input to be changed by clicking on the corresponding row in the **AlternateListName** window.

- The alternate input row is highlighted.
- Information concerning the highlighted alternate input is displayed in the **Order-DataType-LogicalID-Timer** window.

26 Select (highlight) an alternate input with timer settings or the order of alternate inputs to be changed by clicking on the corresponding row in the **Order-DataType-LogicalID-Timer** window.

- The alternate input row is highlighted in the **Order-DataType-LogicalID-Timer** window.
- The data type of the alternate input is displayed in the **DataType** field.

27 To change the order of alternate inputs click on the up/down arrow buttons adjacent to the **Order-DataType-LogicalID-Timer** window as necessary until the highlighted alternate input has the proper order listed in the **Order** column of the window.

- Repeat Steps 26 and 27 as necessary to change the order of additional alternate inputs.

28 To modify the timer setting for an alternate input verify that the alternate input with the timer setting to be changed has been highlighted then type the new timer setting in the **Timer** fields.

29 Click on the appropriate button from the following selections:

- **OK** - to approve the new alternate input setting(s) and dismiss the **AlternateInputValues** GUI.
  - The **Production Request - PR Edit** GUI (Figure 18) is displayed.
- **Apply** - to approve the new alternate input setting(s) without dismissing the **AlternateInputValues** GUI.
- **Cancel** - to return to the **Production Request - PR Edit** GUI without saving the new alternate input setting(s).
  - The **Production Request - PR Edit** GUI (Figure 18) is displayed.

30 Repeat Steps 24 through 29 as necessary to change any additional alternate input setting(s).

NOTE: Depending on the specific PGE chosen, either the **Collection Time** or **Insertion Time** button is automatically selected. If the **Insertion Time** option is selected, the **Multiple DPRs** toggle button appears on the **PR Edit** GUI as shown in Figure 23.

31 If **UTC Time** is to be changed, type the desired data start date and time (in ***MM/DD/YYYY hh:mm:ss*** format) in the **Begin** fields.

32 If **UTC Time** is to be changed, type the desired data end date and time (in ***MM/DD/YYYY hh:mm:ss*** format) in the **End** fields.

33 If the Multiple DPRs for Insertion Time Production Rule applies, ensure that the **Multiple DPRs** toggle button (Figure 23) is depressed.

- If necessary, click on the **Multiple DPRs** toggle button.

34 If the start of the Orbit range is to be changed, type the number of the first orbit of data to be processed in the **[Orbit] From** field.

35 If the end of the Orbit range is to be changed, type the number of the last orbit of data to be processed in the **[Orbit] To** field.

36 If the tile ID is to be changed, type the number of the appropriate tile in the **Tile Id** field.

37 If the DPRs resulting from the PR are to be designated as chain heads, click on the **Yes** button under **PGE Chain Head**.

38 If the DPRs resulting from the PR are not to be designated as chain heads, click on the **No** button under **PGE Chain Head**.

39 If the PGE specified in the PR is at the head of a chain and should be run on a particular virtual computer, click and hold the button adjacent to the **Computer** field to display a menu of available virtual computers, move the mouse cursor to the desired selection (highlighting it), then release the mouse button.

40 If the number of DPRs to be skipped is to be changed, type the number of DPRs to skip in the **Skip** field.

41 If the number of DPRs to be kept is to be changed, type the number of DPRs to keep in the **Keep** field.

42 If the state of the **SkipFirst** button is to be changed, click on the **SkipFirst** button.

43 Type any relevant comments in the **Comments** field.

44 Select **Save As** from the **File** pull-down menu (**File** → **Save As**).

- The **File Selection** window (Figure 24) is displayed.

45 Type a file name for the production request in the **Selection** field.

46 Click on the **OK** button to save the production request.

- Eventually a **Production Request Explosion into DPRs** dialogue box (similar to Figure 25) is displayed.

47 Click on the **OK** button to dismiss the **Production Request Explosion into DPRs** dialogue box.

- The dialogue box is dismissed.

48 Select **File → New** to clear the entries on the **Production Request Editor** GUI.

- Return to Step 1 to modify another new PR.

49 To start the process of exiting from the **Production Request Editor** GUI select **File → Exit** from the pull-down menu.

- A **Do you really want to exit?** dialogue box is displayed.

50 Click on the appropriate button from the following selections:

- **OK** - to exit from the **Production Request Editor** GUI.
  - The **Production Request - PR Edit** GUI (Figure 18) is dismissed.
- **Cancel** - to return to the **Production Request - PR Edit** GUI.

Deleting a Production Request

To delete a Production Request, execute the procedural steps that follow. The procedure starts with the assumption that all applicable servers are currently running and the **Production Request Editor** Introductory GUI (Figure 17) is being displayed.

Deleting a Production Request

---

1 Select the list of Production Requests by clicking on the **PR List** tab.

- The **PR List** GUI page (Figure 26) is displayed.
- A list of Production Requests is displayed.

2 If filtering of the list of Production Requests is desired, click and hold the **PR Type** option button to display a menu of types of Production Requests, move the mouse cursor to the desired selection (highlighting it), then release the mouse button.

- The following Production Request types are available:
  - **All** [default selection].
  - **Routine.**
  - **Reprocessing.**

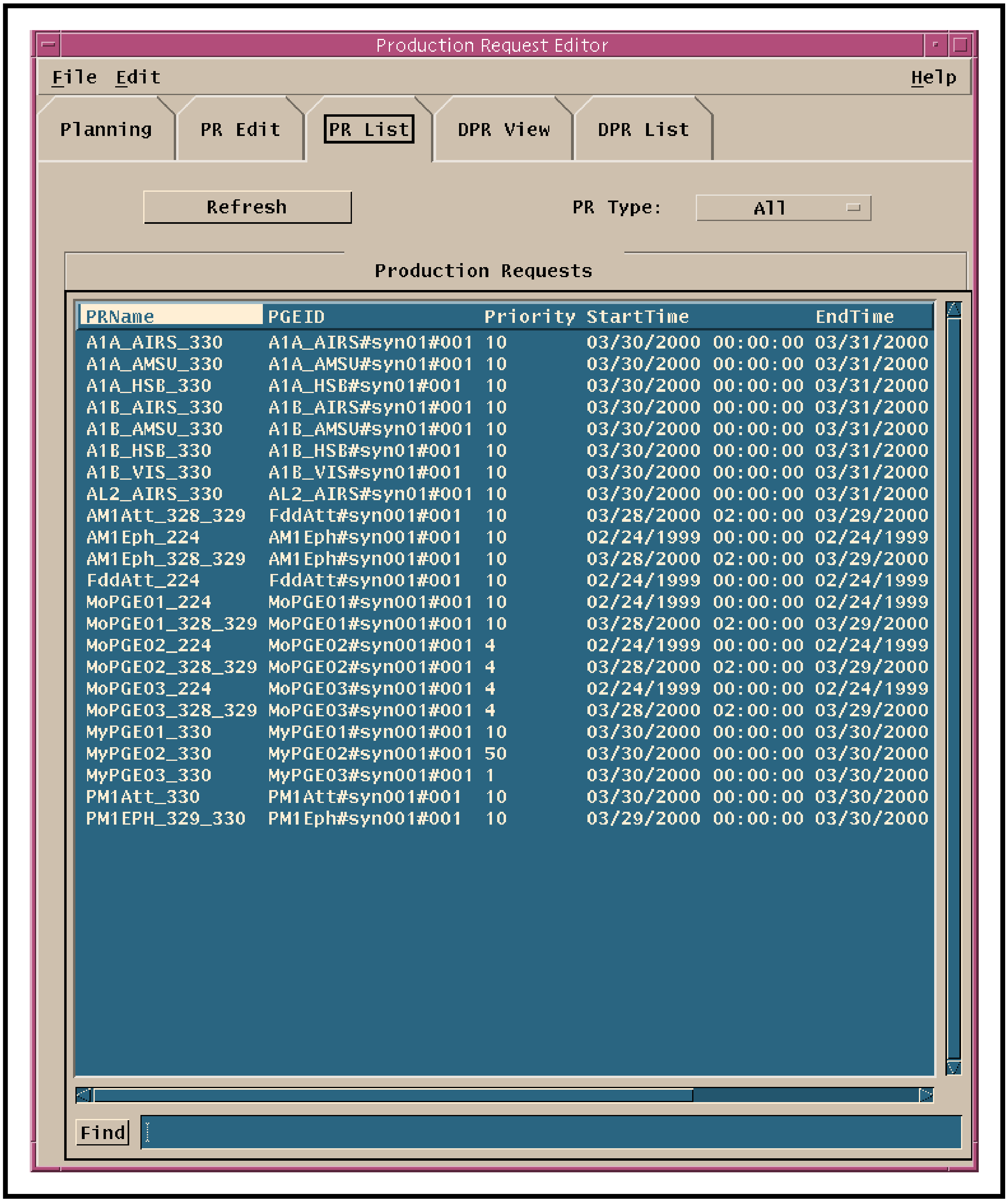

Figure 26. PR List GUI

- **OnDemand.**

- A list of Production Requests of the specified type is displayed.
- A **Refresh** button is available for refreshing the list if necessary.
- The **Find** button provides a means of performing a keyword search of the **PR List** GUI page.
  - A keyword search is performed by typing relevant text in the text entry box to the right of the **Find** button then clicking on the **Find** button.
    - The first entry in the **PR List** GUI page that has the search text is highlighted.
    - Click on the **Find** button again (as necessary) to highlight additional occurrences of the search text.

3 Click on the Production Request to be deleted.

- The Production Request to be deleted is highlighted.

4 Select **Edit** → **Delete**.

- A dialogue box is displayed requesting confirmation of the decision to delete the Production Request.

5 Select **OK** to delete the Production Request.

- If there are no dependencies among Production Requests, a confirmation dialogue box similar to Figure 27 is displayed.
- If there are dependencies among Production Requests, a "List Of Orphan DPRs" dialogue box similar to Figure 28 is displayed.
- The **Find** button provides a means of performing a keyword search of the "List Of Orphan DPRs" dialogue box.
  - A keyword search is performed by typing relevant text in the text entry box to the right of the **Find** button then clicking on the **Find** button.
    - The first entry in the "List Of Orphan DPRs" dialogue box that has the search text is highlighted.
    - Click on the **Find** button again (as necessary) to highlight additional occurrences of the search text.

6 Click on the appropriate button from the following selections:

- **OK** - to confirm deletion of the Production Request(s) and dismiss the dialogue box.
  - A deletion completed dialogue box (similar to Figure 29) is displayed.

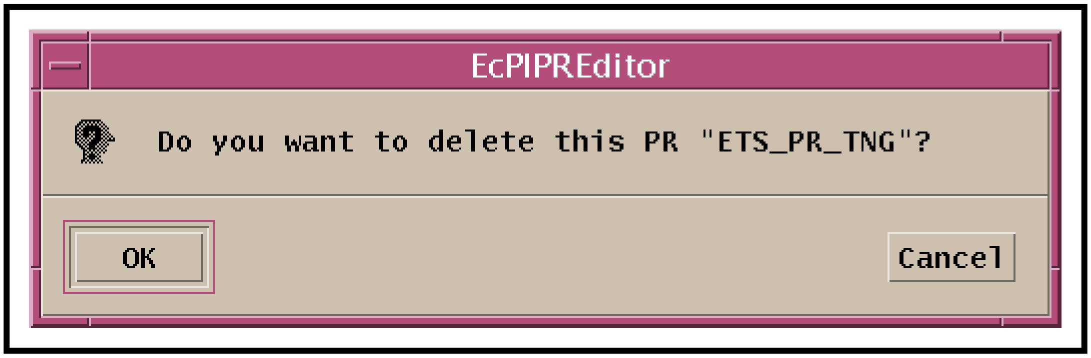

Figure 27. Production Request Deletion Confirmation Dialogue Box

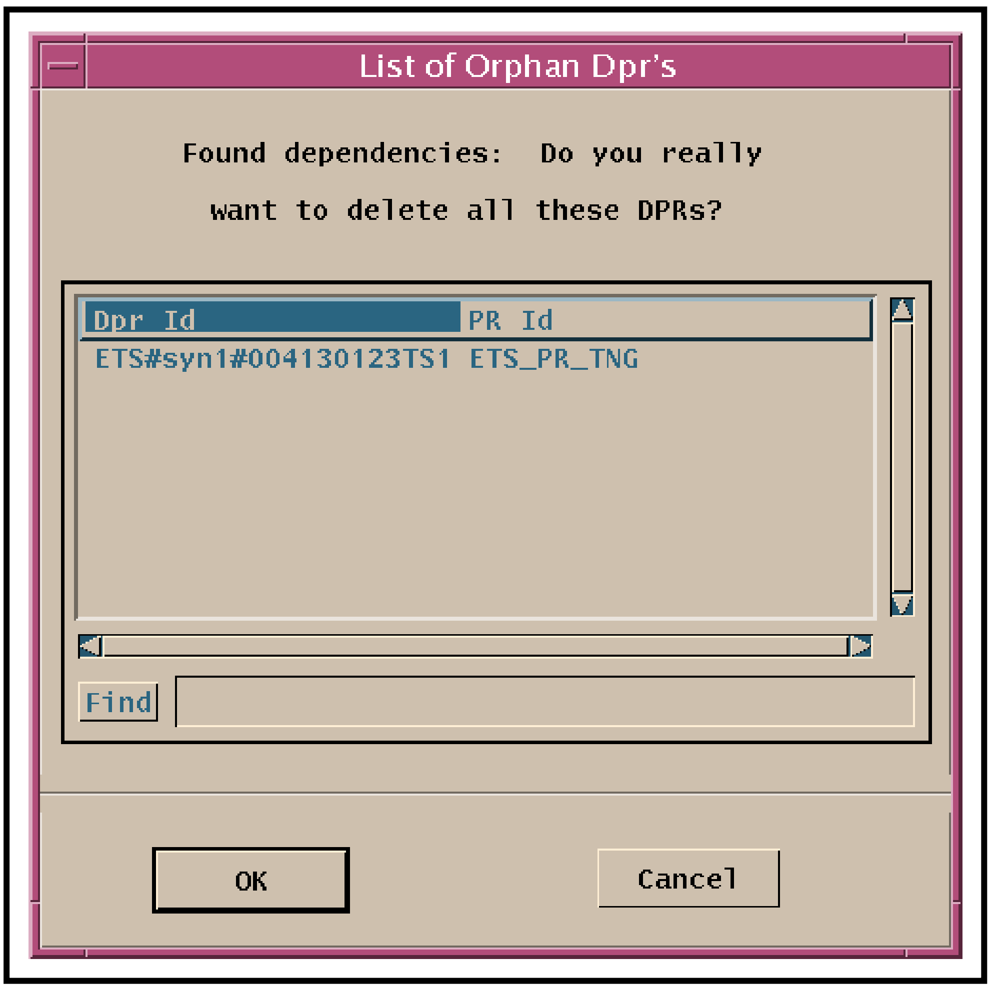

Figure 28. Production Request List of Orphan DPRs Dialogue Box

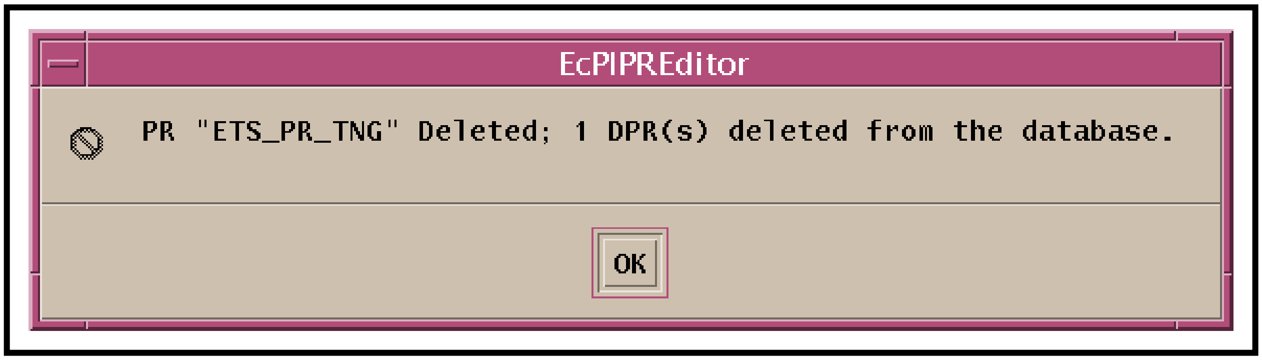

Figure 29. Production Request Deletion Completed Dialogue Box

  - The Production Request(s) is/are deleted.
- **Cancel** - to preserve the Production Request(s) and dismiss the dialogue box.
  - The dialogue box is dismissed.
  - The Production Request(s) is/are not deleted.

7 Select **OK** to dismiss the dialogue box.

- The deletion completed dialogue box is dismissed.

8 To start the process of exiting from the **Production Request Editor** GUI select **File** → **Exit** from the pull-down menu.

- A **Do you really want to exit?** dialogue box is displayed.

9 Click on the appropriate button from the following selections:

- **OK** - to exit from the **Production Request Editor** GUI.
  - The Production Request Editor GUI is dismissed.
- **Cancel** - to return to the **Production Request Editor** GUI.

Reviewing/Deleting Data Processing Requests

Data Processing Requests

Data Processing Requests (DPRs) are generated automatically by the Planning Subsystem.

- DPR is generated automatically from…
  - A PR (which specifies a PGE).
  - Information concerning the PGE acquired during the Science Software Integration and Test (SSI&T) process.
- DPRs are complex, reflecting the complexity of the PGEs.
- DPRs contain information that is used by the Data Processing Subsystem and the AutoSys production scheduling software.

The Production Planner can review DPRs.

- DPRs may provide useful information.
- DPR fields cannot be edited by Operations personnel.
  - Modifications to DPR information would have undesirable side-effects.

Reviewing Data Processing Requests

The Production Request Editor can display DPR information in two ways:

- List of all DPRs applicable to a particular Production Request.
- Data concerning an individual DPR.

The following characteristics pertain to the DPR List:

- The DPR list is filtered, so that only DPRs associated with a particular PR are displayed.
- Each line of the Data Processing Request table represents a DPR, i.e., a job that when made part of an active production plan will be run, assuming all data and resource needs have been satisfied.
- For each DPR the table includes the DPR identification, relevant PGE, name of the corresponding PR, data start date and time, etc.

The Production Planner can select a particular DPR and obtain data on that DPR, including the following characteristics:

- PGE parameters.
  - Displayed in the same format as the **PGE Parameter Mappings** GUI (Figure 20) used in creating a PR.
- UR File Mappings.
  - Displays the input and output files for a particular DPR on the **UR File Mappings** GUI (Figure 30)
  - GUI displays one line of information for each file that may be used by or be produced by the PGE, including the following data:
    - Logical ID - The identification (ID) or tag used within the PGE to access the file.
    - Granule ID - The reference used to identify the granule uniquely.
    - Start/Stop Time - The start and stop date and time for the data contained in this file.
    - Availability - Data and time when the data file is expected to be accessible for use in processing.
    - UR (Universal Reference) - The unique identifier for each individual data granule.

The procedure for reviewing data processing requests starts with the assumption that all applicable servers and the **Production Request Editor** GUI are currently running and the **Production Request Editor** Introductory GUI (Figure 17) is being displayed.

Reviewing Data Processing Requests

1 Select the **Data Processing Requests** list by clicking on the **DPR List** tab.

- The **DPR List** GUI page (Figure 31) is displayed.
- The table shown on the GUI provides the following information on the DPRs:
  - **DPR Id** - identification (name) of the DPR.
  - **PGE Id** - identification of the PGE specified in the DPR.
  - **PR Name** - name of the PR that led to the creation of the DPR.
  - **Tile Id –** identification of the tile (if applicable).
  - **Data Start Time (UTC)** - DPR's start date and time.
  - **Data Stop Time (UTC)** - DPR's stop date and time.

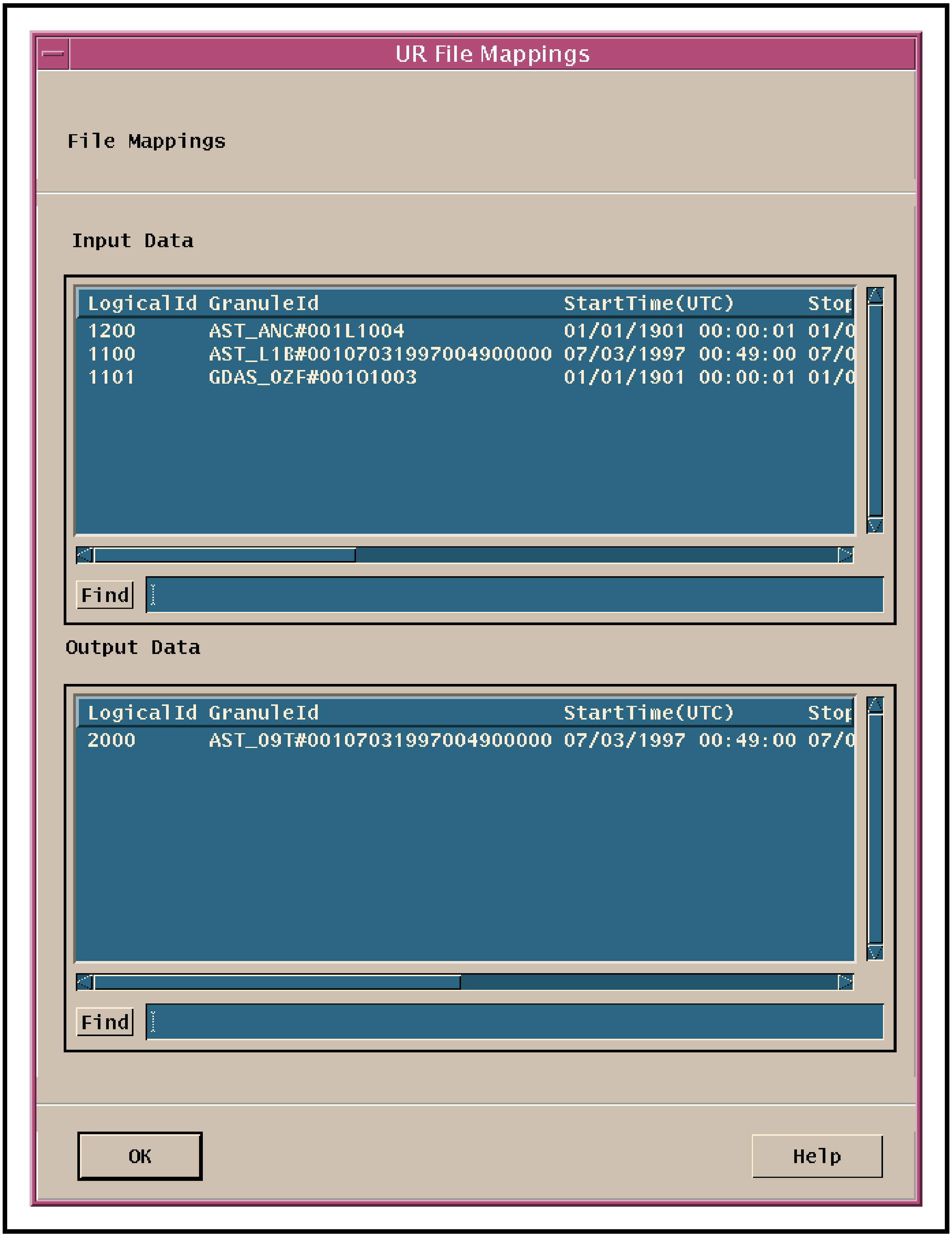

Figure 30. UR File Mappings GUI

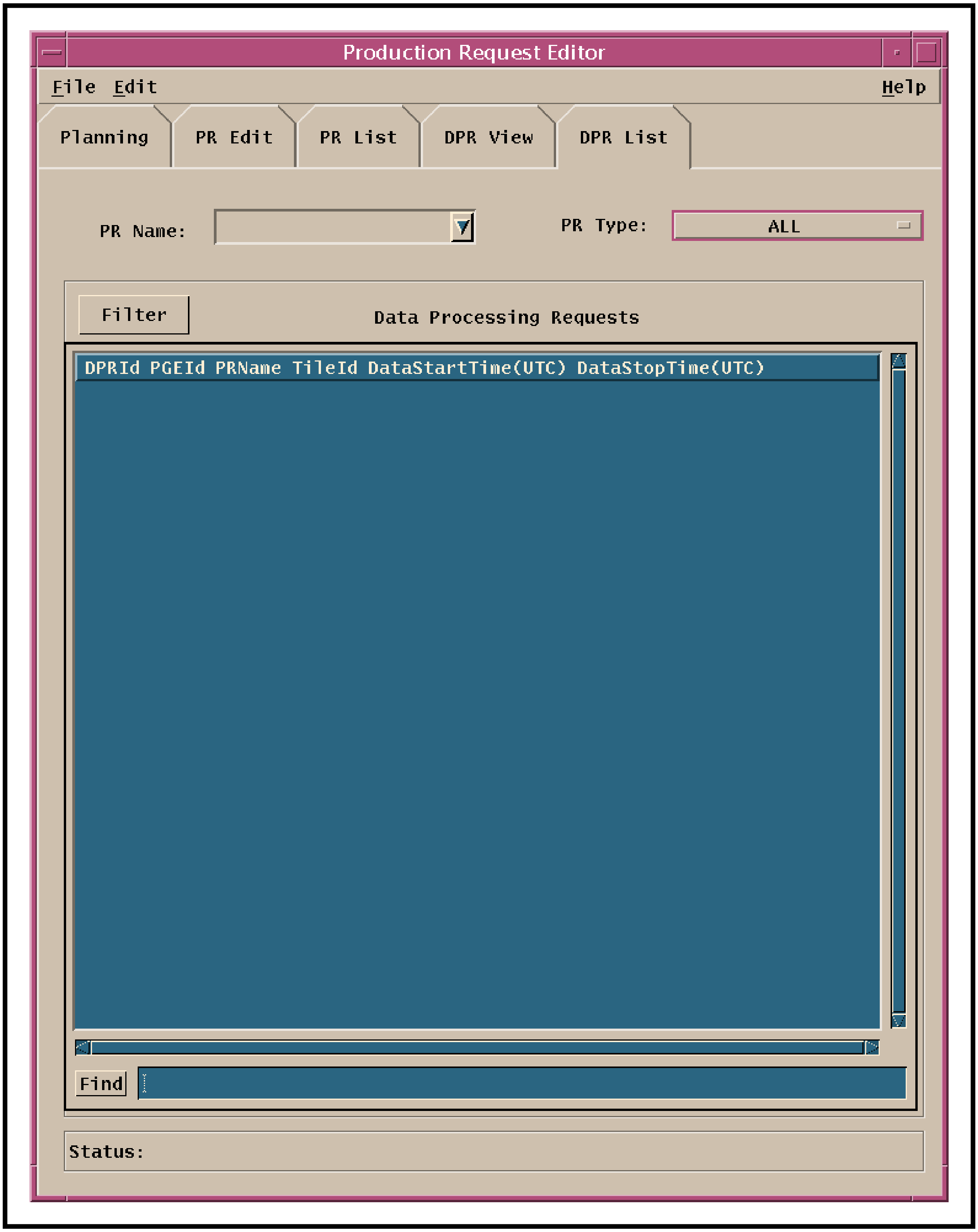

Figure 31. DPR List GUI

– **DPR State** - DPR's state.

2 If filtering of the list of Production Requests is desired, click and hold the **PR Type** option button to display a menu of types of Production Requests, move the mouse cursor to the desired selection (highlighting it), then release the mouse button.

- The following Production Request types are available:
  - **ALL** [default selection].
  - **Routine.**
  - **Reprocessing.**
  - **OnDemand.**

3 To list the DPRs associated with a particular Production Request, first click on the option button associated with the **PR Name** field.

- An option menu of Production Requests is displayed.

4 Click on the Production Request that includes the DPR to be reviewed.

- The Production Request is displayed in the **PR Name** field.

5 Click on the **Filter** button.

- The DPRs for the specified PR only are shown in the **Data Processing Requests** table.

6 Click on the DPR to be reviewed from those shown in the **Data Processing Requests** table.

- The DPR to be reviewed is highlighted.
- The **Find** button provides a means of performing a keyword search of the **Data Processing Requests** table.
  - A keyword search is performed by typing relevant text in the text entry box to the right of the **Find** button then clicking on the **Find** button.
    - The first entry in the **Data Processing Requests** table that has the search text is highlighted.
    - Click on the **Find** button again (as necessary) to highlight additional occurrences of the search text.

7 Select **File** → **Open** from the pull-down menu.

8 Click on the **DPR View** tab.

- The **DPR View** GUI (Figure 32) is displayed.

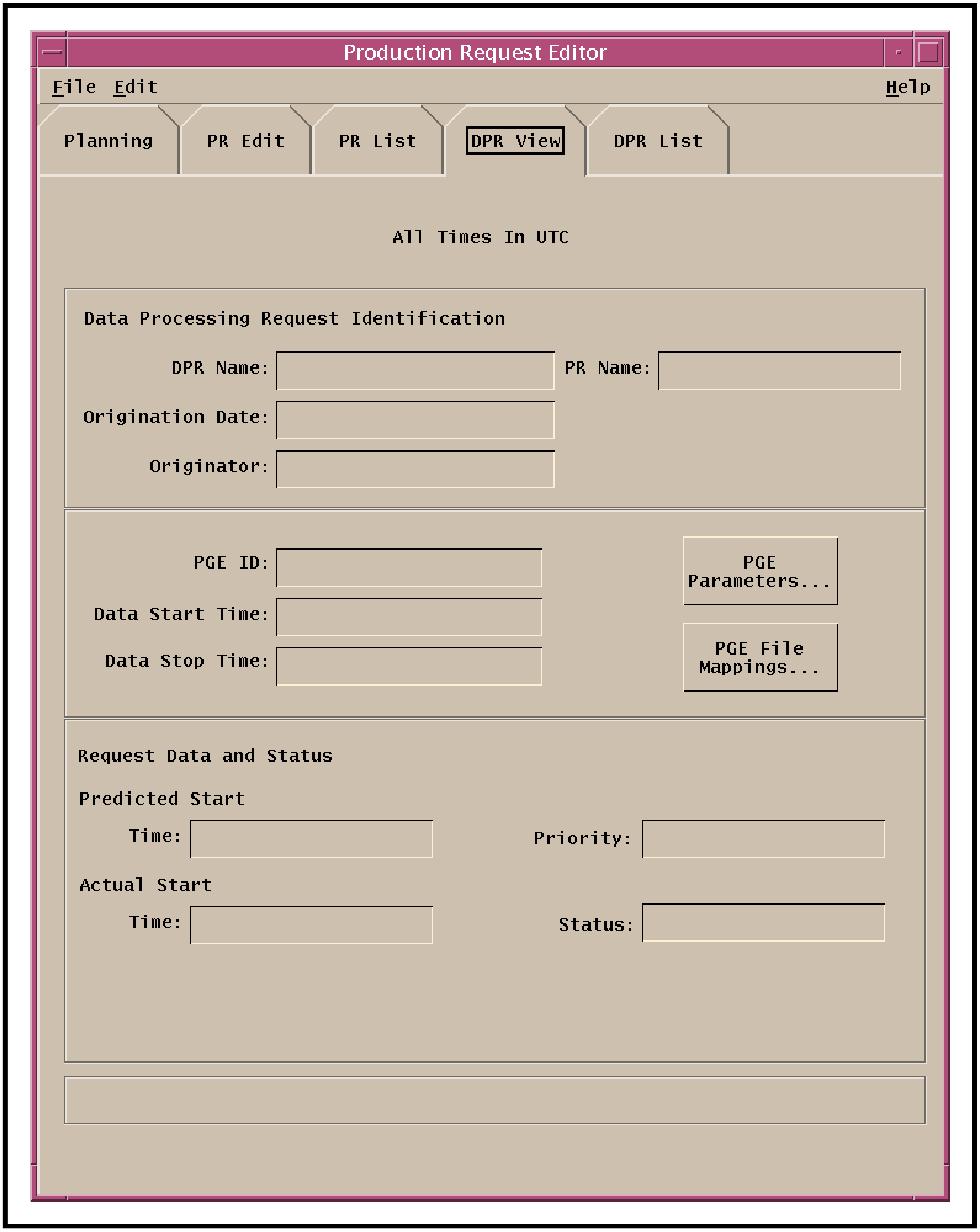

Figure 32. DPR View GUI

- The **DPR View** GUI shows the following information concerning the selected DPR:
  - **DPR Name.**
  - **PR Name.**
  - **Origination Date.**
  - **Originator.**
  - **PGE Id.**
  - **Data Start Time.**
  - **Data Stop Time.**
  - **Predicted Start Time** (date and time)**.**
  - **Actual Start Time** (date and time)**.**
  - **Priority.**
  - **Status.**

9 To obtain information concerning the PGE parameters (of the PGE associated with the DPR), click on the **PGE Parameters...** button.

- The **PGE Parameters** GUI (same format as Figure 20 except there is no **Parameter Mapping** field) is displayed.
- The **Find** button provides a means of performing a keyword search of the PGE Parameters table.
  - A keyword search is performed by typing relevant text in the text entry box to the right of the **Find** button then clicking on the **Find** button.
    - The first entry in the PGE Parameters table that has the search text is highlighted.
    - Click on the **Find** button again (as necessary) to highlight additional occurrences of the search text.

10 When finished reviewing the PGE parameters, click on the **OK** button to return to the **DPR View** GUI.

- The **DPR View** GUI (Figure 32) is displayed.

11 Click on the **PGE File Mappings...** button.

- The **UR File Mappings** GUI (Figure 30) is displayed.

- The **UR File Mappings** GUI displays one line of information for each file that may be used by or be produced by the PGE associated with the selected DPR:
  - **Logical Id.**
  - **Granule Id.**
  - **Start Time (UTC)** (date and time).
  - **Stop Time (UTC)** (date and time).
  - **Availability** (date and time).
  - **UR** (granule universal reference).
- The PGE's input data and output data are displayed in separate areas.
- The **Find** buttons provides a means of performing keyword searches of the **Input Data** and **Output Data** tables.
  - A keyword search is performed by typing relevant text in the text entry box to the right of the **Find** button then clicking on the **Find** button.
    - The first entry in the **Input Data** or **Output Data** table (as applicable) that has the search text is highlighted.
    - Click on the **Find** button again (as necessary) to highlight additional occurrences of the search text.

12 When finished reviewing the **UR File Mappings**, click on the **OK** button to return to the **DPR View** GUI.

- The **DPR View** GUI is displayed.

13 Review the **Data Start Time** and **Data Stop Time**.

14 Review the **Request Data and Status** area of the GUI.

- The **DPR View** GUI displays the following information concerning the selected DPR:
  - **Predicted Start Time** (date and time).
  - **Actual Start Time** (date and time).
  - **Priority.**
  - **Status.**
- **Predicted Start Date/Time** and **Actual Start Date/Time** are displayed only if the Production Request has been scheduled.

15 When finished reviewing information concerning the DPR, repeat Steps 1 through 14 to review additional DPRs.

16 To start the process of exiting from the **Production Request Editor** GUI select **File → Exit** from the pull-down menu.

- A **Do you really want to exit?** dialogue box is displayed.

17 Click on the appropriate button from the following selections:

- **OK** - to exit from the **Production Request Editor** GUI.
  - The **Production Request Editor** GUI is dismissed.
- **Cancel** - to return to the **Production Request Editor** GUI.

Deleting a Data Processing Request

To delete a DPR, execute the procedure steps that follow. The procedure starts with the assumption that all applicable servers are currently running and the **Production Request Editor** Introductory GUI (Figure 17) is being displayed.

Deleting a Data Processing Request

---

1 Select the DPR List by clicking on the **DPR List** tab.

- The **DPR List** GUI page (Figure 31) is displayed.

2 If filtering of the list of Production Requests is desired, click and hold the **PR Type** option button to display a menu of types of Production Requests, move the mouse cursor to the desired selection (highlighting it), then release the mouse button.

- The following Production Request types are available:
  - **ALL** [default selection].
  - **Routine.**
  - **Reprocessing.**
  - **OnDemand.**

3 Click on the **PR Name** option button.

- A list of Processing Requests is displayed.

4 Click on the Production Request with which the DPR to be deleted is associated.

- The Production Request is displayed in the **PR Name** field.

5 Click on the **Filter** button.

- A list of the DPRs associated with the selected Production Request is displayed.

6 Click on the DPR to be deleted.

- The DPR to be deleted is highlighted.
- The **Find** button provides a means of performing a keyword search of the **Data Processing Requests** table.
  - A keyword search is performed by typing relevant text in the text entry box to the right of the **Find** button then clicking on the **Find** button.
    - The first entry in the **Data Processing Requests** table that has the search text is highlighted.
    - Click on the **Find** button again (as necessary) to highlight additional occurrences of the search text.

7 Select **Edit→Delete** from the pull-down menu.

- A DPR deletion confirmation dialogue box (similar to Figure 33) is displayed requesting confirmation of the decision to delete the DPR.

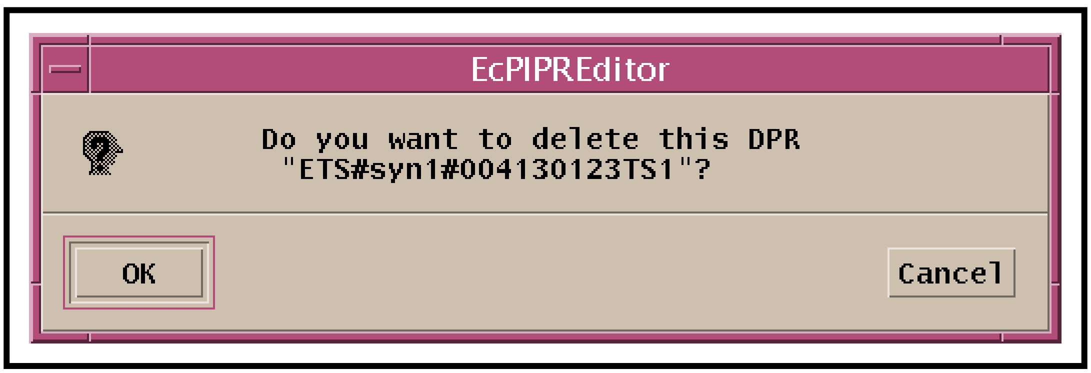

Figure 33. DPR Deletion Confirmation Dialogue Box

8 Click on the appropriate button from the following selections:

- **OK** - to confirm deletion of the DPR and dismiss the dialogue box.
  - A deletion dialogue box (similar to Figure 34) is displayed.

– The DPR is deleted.

– If the DPR was the only DPR in the Production Request, the PR is deleted too.

- **Cancel** - to preserve the DPR and dismiss the dialogue box.

  – The deletion confirmation dialogue box is dismissed.

  – The DPR is not deleted.

9 Select **OK** to dismiss the dialogue box.

- The dialogue box is dismissed.

10 To start the process of exiting from the **Production Request Editor** GUI select **File** → **Exit** from the pull-down menu.

- A **Do you really want to exit?** dialogue box is displayed.

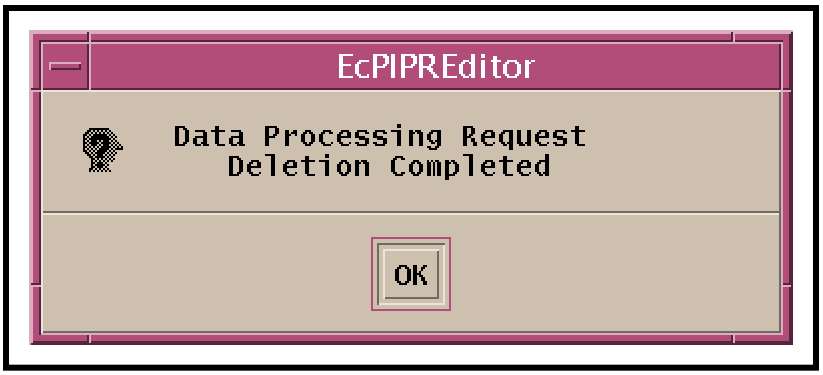

Figure 34. DPR Deletion Completed Dialogue Box

11 Click on the appropriate button from the following selections:

- **OK** - to exit from the **Production Request Editor** GUI.

  – The **Production Request Editor** GUI is dismissed.

- **Cancel** - to return to the **Production Request Editor** GUI.

Deleting or Aborting an On-Demand Processing Request

The Planning Subsystem detects changes in the status of ASTER on-demand processing requests, whether the changes were made by the system or by operations personnel. However, PLS is not able to determine why operations personnel have canceled or aborted a request. Consequently, if it is necessary to cancel or abort an on-demand processing request, notify User Services personnel so they can send an e-mail message to the requester explaining why the request was canceled or aborted.

Launching Planning Workbench-Related GUIs

Launching the Production Strategies GUI

The following software applications are associated with the Production Strategies GUI:

- Production Strategies GUI.
- Sybase ASE Server for the PDPS database.

Production Strategies are high-level sets of priorities that the Production Planner makes available to the Planning Workbench for determining the priorities and preferences in the processing of DPRs. The values included in the selected strategy are read by the Planning Workbench when prioritizing the DPRs in a production plan.

Access to the Production Strategies GUI is gained through the use of UNIX commands. Launching the Production Strategies GUI starts with the assumption that the applicable servers are running and the Production Planner has logged in to the system.

Launching the Production Strategies GUI

1 Access a terminal window logged in to the Planning/Management Workstation host.

- Examples of Planning/Management Workstation host names include **e0pls03** and **l0pls02.**
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 Type **setenv ECS_HOME /usr/ecs/** then press the **Return/Enter** key.

- When logging in as a system user (e.g., cmshared), the ECS_HOME variable may be set automatically so it may not be necessary to perform this step.

3 Type **cd /usr/ecs/*MODE*/CUSTOM/utilities** then press **Return/Enter**.

- Change directory to the directory containing the production planning startup scripts (e.g., EcPlPRE_IFStart).
- The ***MODE*** will most likely be one of the following operating modes:
  - OPS (for normal operation).
  - TS1 (for SSI&T).
  - TS2 (new version checkout).

- Note that the separate subdirectories under /usr/ecs apply to (describe) different operating modes.

4 Type **EcPlProdStratStart** ***MODE*** then press **Return/Enter** to launch the **Production Strategies** GUI.

- The **Production Strategies** GUI (Figure 35) is displayed.

Launching Planning Workbench-Related GUIs

The following software applications are associated with the Planning Workbench:

- System Name Server.
- Message Handler.
- Resource Model.
- Planning Workbench.
- Production Timeline.
- Sybase ASE Server for the PDPS database.

Access to the Planning Workbench GUIs is gained through the use of UNIX commands. Launching Planning Workbench-related GUIs starts with the assumption that the applicable servers are running and the Production Planner has logged in to the system.

Launching Planning Workbench-Related GUIs

---

1 Access a terminal window logged in to the Planning/Management Workstation host.

- Examples of Planning/Management Workstation host names include **e0pls03** and **l0pls02.**
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 Type **setenv ECS_HOME /usr/ecs/** then press the **Return/Enter** key.

- When logging in as a system user (e.g., cmshared), the ECS_HOME variable may be set automatically so it may not be necessary to perform this step.

3 Type **cd /usr/ecs/*MODE*/CUSTOM/utilities** then press **Return/Enter**.

- Change directory to the directory containing the production planning startup scripts (e.g., EcPlPRE_IFStart).

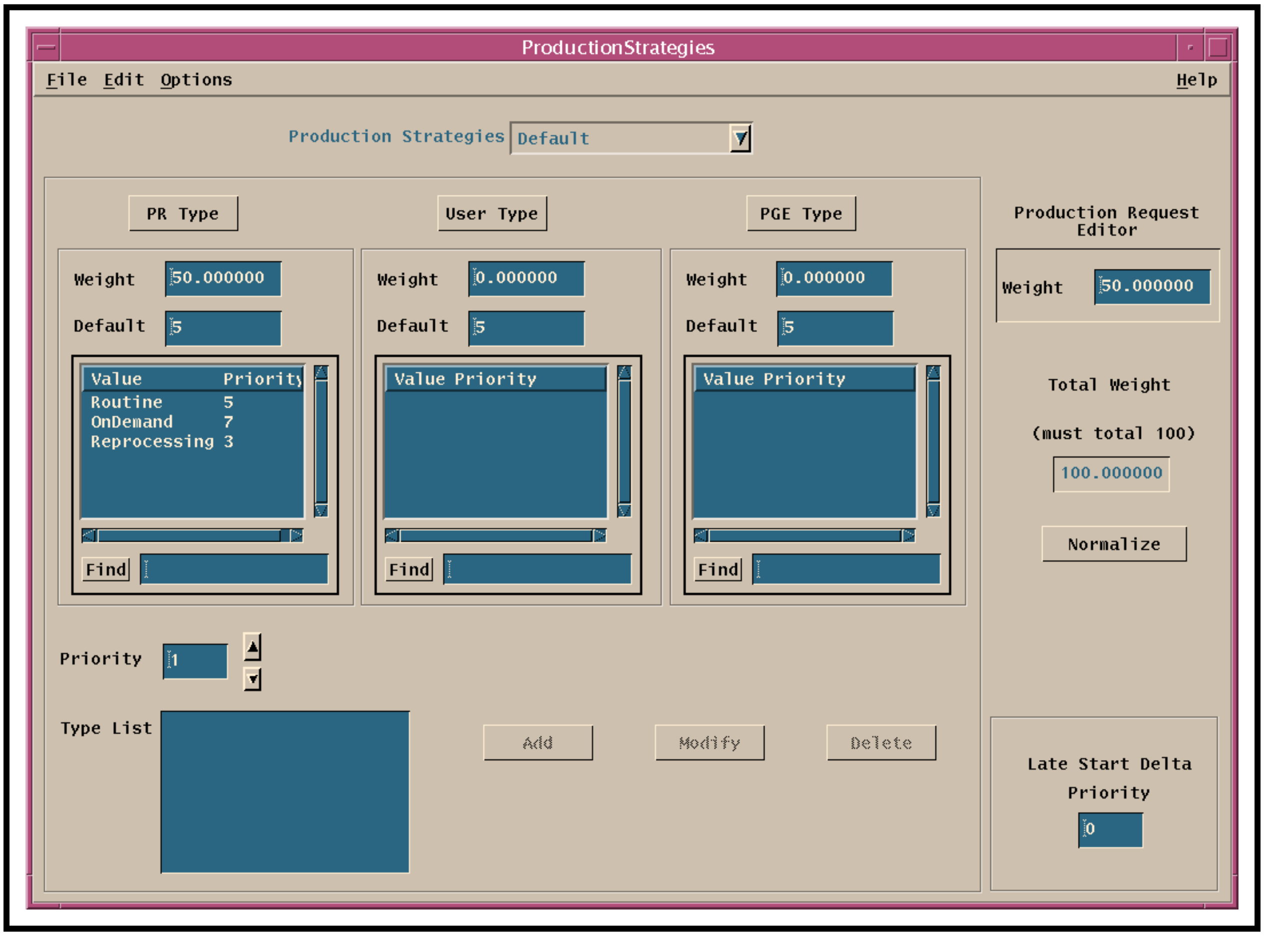

Figure 35. Production Strategies GUI

- The ***MODE*** will most likely be one of the following operating modes:
  - OPS (for normal operation).
  - TS1 (for SSI&T).
  - TS2 (new version checkout).
- Note that the separate subdirectories under /usr/ecs apply to (describe) different operating modes.

NOTE: The selection of a start-up script for launching the **Planning Workbench** GUI depends on the number of DPRs in the PDPS database. If the number is very high (~4000), the **EcPlSomeStart** script is used to start the underlying processes (Message Handler, System Name Server, and Resource Model). Then additional scripts are used to start the **Planning Workbench** GUI and the **Production Planning Master Timeline** GUI. If the number of DPRs in the PDPS database is not very high (less than about 4000), the **EcPlAllStart** script is used to start the underlying processes, **Planning Workbench** GUI and **Production Planning Master Timeline** GUI.

4 If the number of DPRs in the PDPS database is not large (less than about 4000), type **EcPlAllStart** ***MODE ApplicationID*** then press **Return/Enter** to launch the Message Handler, System Name Server, Resource Model, **Planning Workbench** GUI, and **Production Planning Master Timeline** GUI.

- The **Message Handler** GUI (Figure 36) is displayed.
- Eventually, the **Planning Workbench** GUI (Figure 37) is displayed. Then the **Production Planning Master Timeline** GUI (Figure 38) is displayed.
  - The **Production Planning Master Timeline** GUI usually occupies the entire screen when it is initially displayed.
- The **Message Handler** GUI displays messages of the following types:
  - Engineering.
  - Information.
  - Warning.
  - Internal.
- The **System Name Server** (SNS) handles interprocess communication.
- The **Resource Model** is an underlying resource data coordinator for the planning software.

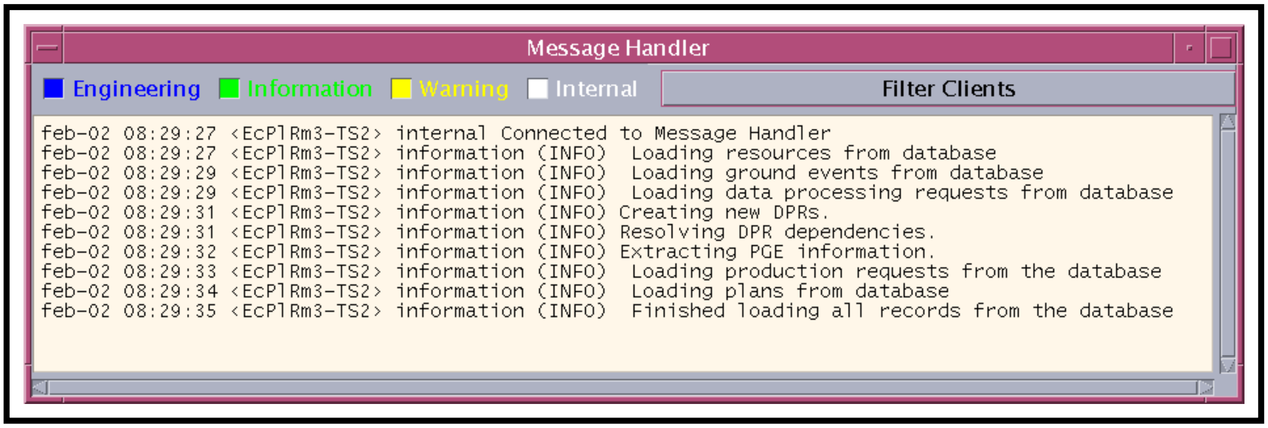

Figure 36. Message Handler GUI

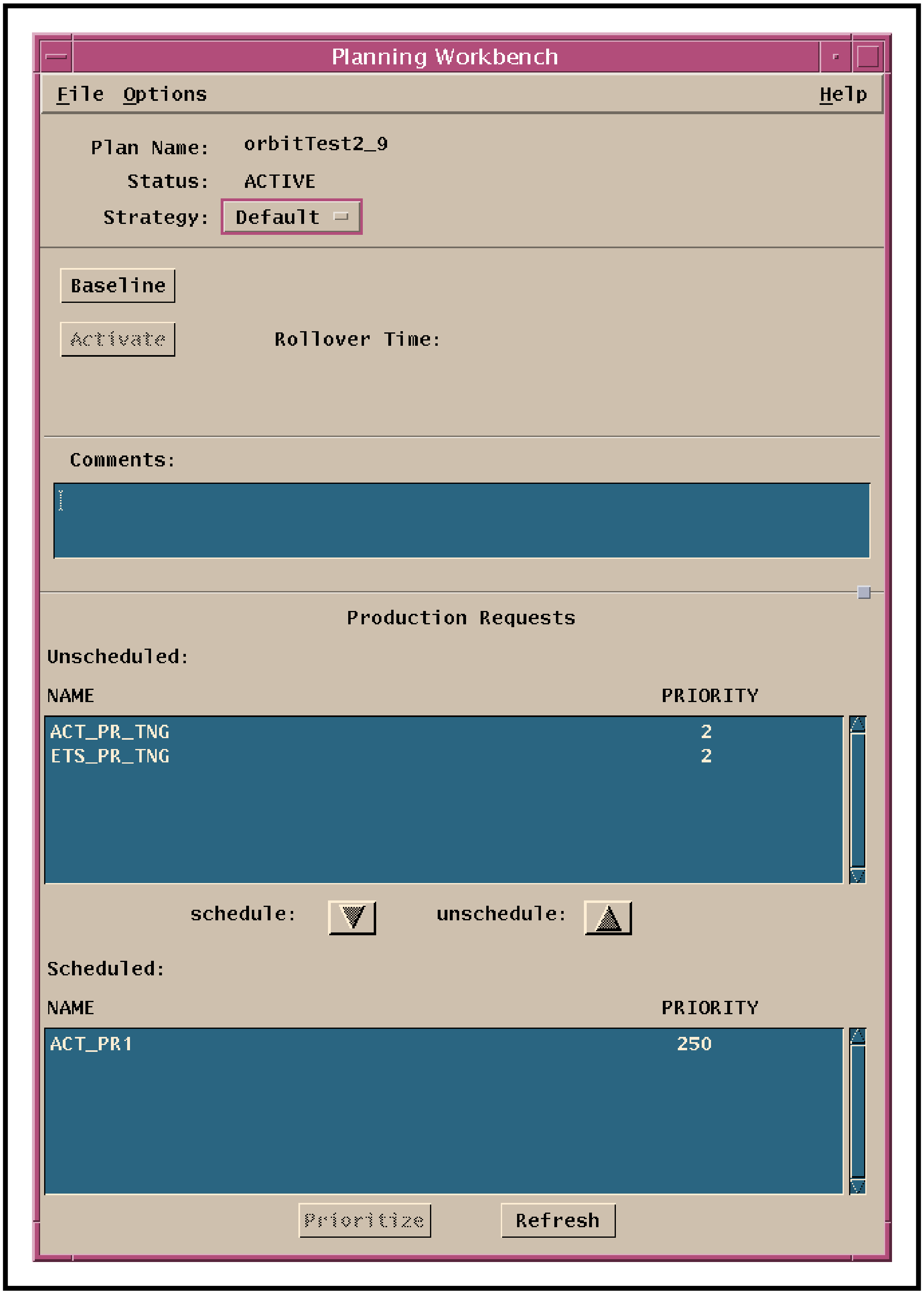

Figure 37. Planning Workbench GUI

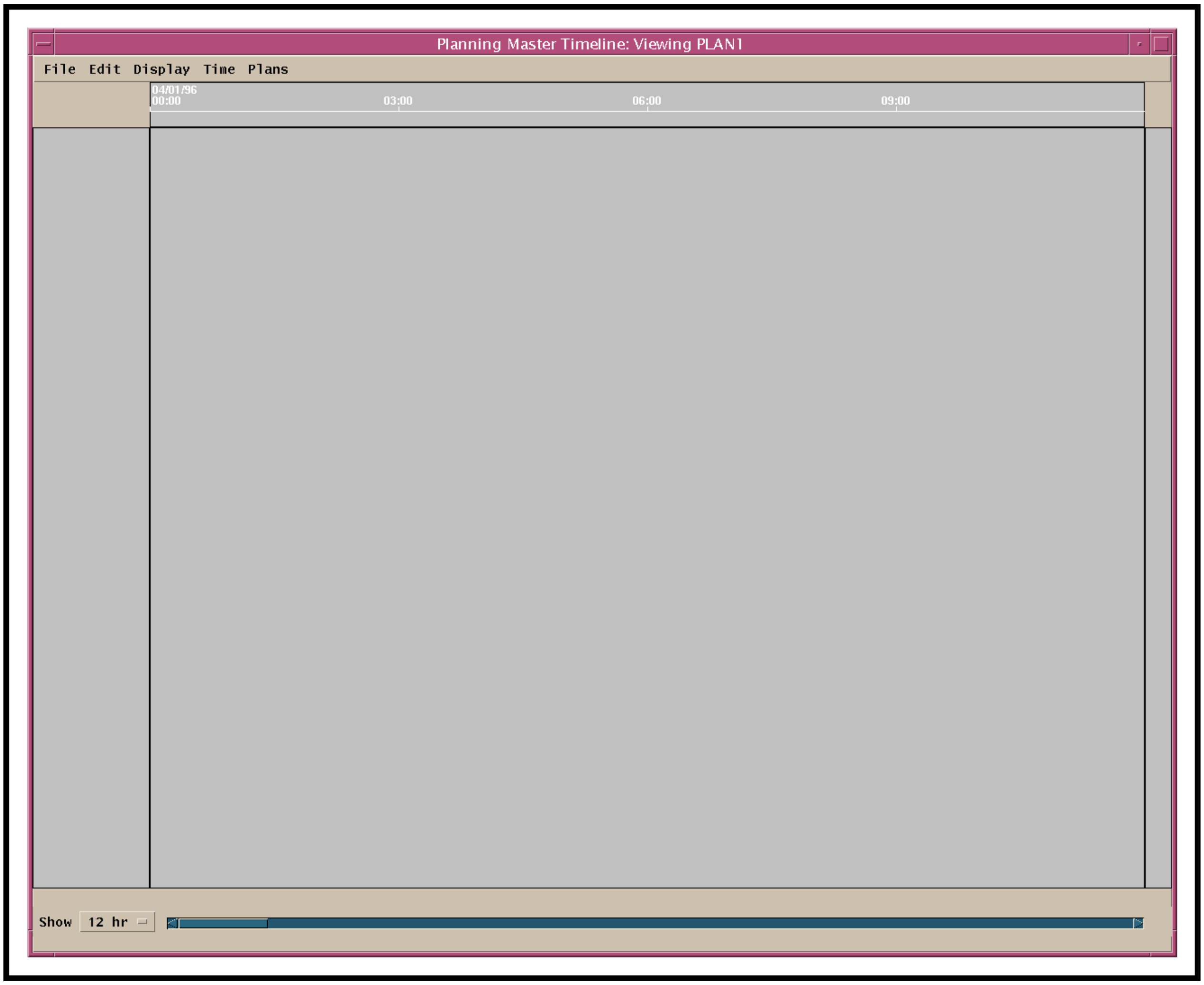

Figure 38. Production Planning Master Timeline GUI

- The ***ApplicationID*** is any number from 1 to 5. It identifies the message service in use so messages can be directed to the proper message handler GUI. Consequently, it is a good idea to use the same ApplicationID (also called MSGSRV_ID) consistently during a planning session.
- Go to Step 8.

5 If the number of DPRs in the database is large (~4000 or more), type **EcPlSomeStart *MODE ApplicationID*** then press **Return/Enter** to launch the Message Handler, System Name Server, and Resource Model.

- The **Message Handler** GUI (Figure 36) is displayed.
- Wait until the Resource Model is completely up before proceeding with starting the **Planning Workbench** GUI.

6 If the number of DPRs in the database is large (~4000 or more), type **EcPlWbStart *MODE ApplicationID*** then press **Return/Enter** to launch the **Planning Workbench** GUI.

- The **Planning Workbench** GUI (Figure 37) is displayed.

7 If the number of DPRs in the database is large (~4000 or more), type **EcPlTlStart *MODE ApplicationID*** then press **Return/Enter** to launch the **Production Planning Master Timeline** GUI.

- The **Production Planning Master Timeline** GUI (Figure 38) is displayed.

8 If the **Production Planning Master Timeline** GUI is occupying the whole screen, either click on the "minimize" icon in the upper right corner of the GUI or adjust the window size and the view of the timeline as necessary using the mouse.

- Grab a corner of the timeline window with the cursor and resize the window as desired.

This page intentionally left blank.

Creating a New Production Plan

Production Plan

The planning process involves the Production Planner preparing monthly and weekly production plans as well as a daily production schedule from the most current weekly plan. Much of the planning occurs off line; however, the Production Planner uses the Planning Workbench GUI to create and activate the current or active plan.

Each DAAC has created its own policy and process for creating, reviewing, publishing and distributing production plans. Daily plans may be posted on a Worldwide Web page or distributed in hardcopy form or via e-mail.

The most important factor in production planning is meeting the customers' needs. The ultimate customers are the Science Computing Facilities that have provided science software to be run on the system on a routine basis. They expect timely generation and archiving of products using their science software. However, there may be other considerations as well, so the Production Planner typically has to consider the following types of factors when creating production plans:

- Keeping up to date with routine processing of products using the SCFs' science software.
- Performing reprocessing of data using new versions of science software.
- Regenerating products (as needed) that have been damaged or lost from the archive.
- Taking into account the effects of ASTER on-demand processing, which circumvents the normal planning process but uses processing resources.

Although production planning varies from DAAC to DAAC, the following guidelines are generally applicable:

- Monthly plans.
  - Developed for the coming month and one or two months in advance.
  - Produced, reviewed, updated, published and distributed approximately two weeks before the beginning of the month.
  - Plan for the coming month is used to establish a baseline against which production targets can be measured.
- Weekly plans.
  - Produced, reviewed, updated, published and distributed approximately five days before the beginning of the coming week.

- Used to produce a baseline for comparison of planned vs. actual production results.

- Daily plan or schedule.
  - Produced each day for the next processing day.
  - Developed from the current weekly plan, adjusted to reflect the actual processing accomplished and the actual resources available at the time the daily schedule is generated.

- Current (active) plan.
  - Activated as needed (one or more plans a day, depending on circumstances) using the Planning Workbench.
  - Implements the daily plan (or some portion of the daily plan).

During normal processing, when reasonably accurate predictions of the processing time for the PGEs are available, the processing schedule should result in a reasonably accurate prediction of when data products will be generated. However, during abnormal situations (e.g., hardware or software failure), what is actually accomplished could depart significantly from the plan. In addition, a daily plan may be divided into several segments to be activated separately. In such situations, the Production Planner is likely to develop a new current plan to continue processing or to change the processing priorities. This process is known as "replanning." Further information concerning replanning is provided in a subsequent section of this lesson.

The Production Planner uses the Planning Workbench when creating a current (active) plan for production data processing at the DAAC. The Planning Workbench provides the means by which the Production Planner selects and schedules specific PRs whose DPRs are to be run.

The Planning Master Timeline tool provides a forecast of the start and completion times of the jobs based upon experience in running the PGEs during the SSI&T process. When the generated plan is "activated" through the Planning Workbench, information in the plan is transferred to Job Management in the Data Processing Subsystem. Job Management creates and releases jobs into the Platinum AutoSys tool, where production processing is managed.

Figure 39 shows the general flow of production requests/data processing requests from the Production Request Editor through the Planning Workbench to the AutoSys production management tool.

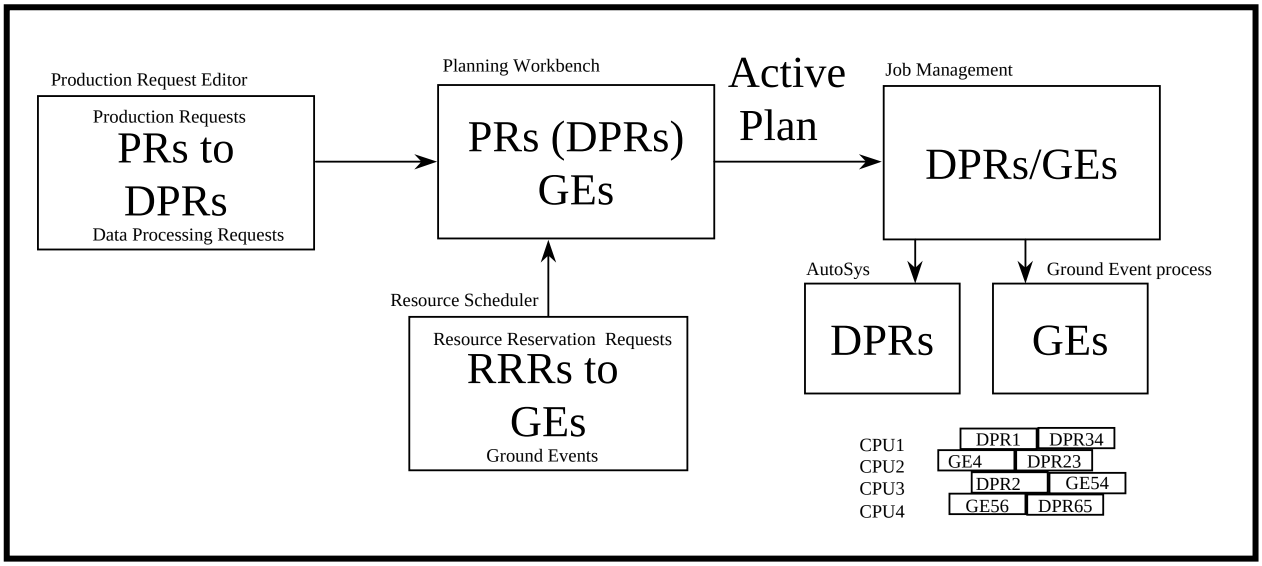

Figure 39. Planning Workbench Flow

Production Strategy

As previously mentioned Production Strategies are high-level sets of priorities created by the Production Planner. The strategies are available for selection on the Planning Workbench. The Production Planner's choice of a strategy for a current production plan determines the priorities and preferences in the processing of associated DPRs.

Production Strategies work on two levels. First, the Production Planner can update lists of DPR attributes so that each value an attribute can have is tied to a particular priority. For example, the DPR attribute “PR Type” has three values that may have their default priority of 2 changed as follows (on a scale of 1 to 10):

- Routine 6
- On-Demand 10
- Reprocessing 4

Next, the Production Planner can change the weight that each attribute's priority is given. For example, weights (from 1 to 100) might be assigned to the DPR attributes as follows:

- PR Type 45
- User Type 15
- PGE Type 20

A weight is also given to the priority specified when the user (e.g., the Production Planner) created the Production Request as shown in the following example:

- Production Request Editor 20

The total weights assigned to PR Type, User Type, PGE Type and Production Request Editor [Priority] must equal 100.

The values included in the selected strategy are read by the Planning Workbench when prioritizing the DPRs in a production plan. Figure 40 provides an illustration of how priority is calculated for a DPR.

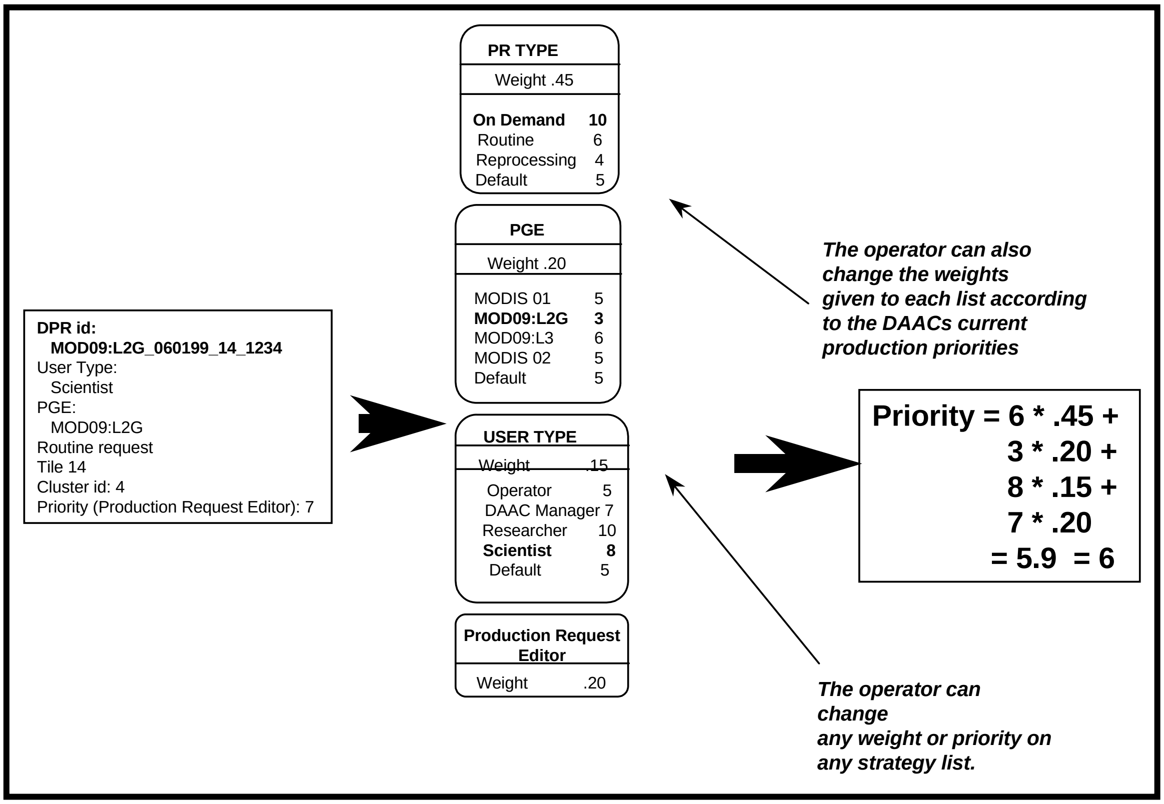

Figure 40. Example of Calculating Priority for a DPR

Finally, there is a Late Start Delta that can be used to increase the priority of all jobs that have been waiting in the Production Queue for more than a day.

Defining a Production Strategy

The Production Planner uses the Production Strategies GUI to develop Production Strategies. The procedure that follows describes how to define or modify a Production Strategy.

The procedure for defining a production strategy starts with the assumption that all applicable servers are currently running and the **Production Strategies** GUI (Figure 35) is being displayed.

Defining a Production Strategy

---

1 If defining a new production strategy, select **File → New** from the pull-down menu.

- The fields of the **Production Strategies** GUI (Figure 35) are reset.

2 If modifying an existing production strategy, first click on the option button associated with the **Production Strategies** field, then highlight (in the option menu) the name of the production strategy to be modified.

- Data pertaining to the selected production strategy are displayed in the applicable fields of the **Production Strategies** GUI (Figure 35).
- Alternatively, it is possible to select **File → Open** from the pull-down menu, select the desired production strategy from the list on the **Open** window (Figure 41), and click on the **Ok** button to open the production strategy.

NOTE: The **Help** button at the bottom of **Production Strategies Open** window is non-functional (does not work).

3 If changing the default priority for PR Type, click in the **Default** field below the **PR Type** button and type the desired default value.

- The range for the default is from 1 to 10.

4 If changing the default priority for User Type, click in the **Default** field below the **User Type** button and type the desired default value.

5 If changing the default priority for PGE Type, click in the **Default** field below the **PGE Type** button and type the desired default value.

6 If defining or modifying a priority for a type of production request, first click on the **PR Type** button.

- The different types of production requests are displayed in the **Type List** field at the bottom left of the GUI.

7 If defining a priority for a type of production request **not** currently listed in the **PR Type Value-Priority** list, click on that PR type in the **Type List** field.

- The PR type is highlighted.
- It is possible to highlight multiple PR types (by clicking on each one in turn) if they are all going to be assigned the same priority.

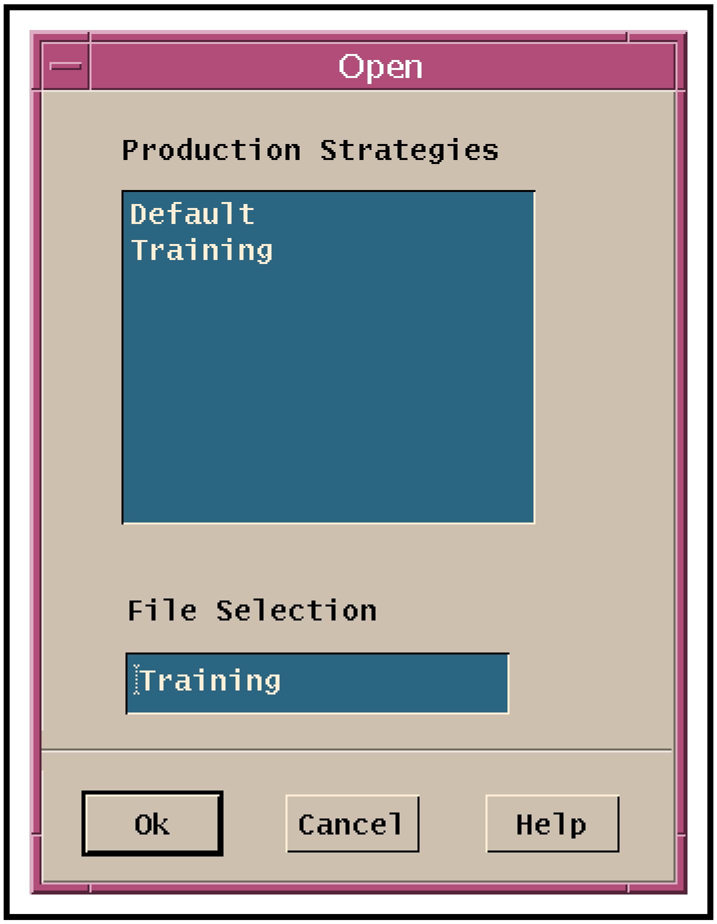

Figure 41. Production Strategies "Open" Window

8 If redefining or deleting a priority for a type of production request **already** listed in the **PR Type Value-Priority** list, click on that PR type in the **Value-Priority** list.

- It is possible to highlight multiple PR types (by clicking on each one in turn while holding down either the **Shift** key or the **Ctrl** key) if the same action is going to be taken with respect to all of them.
- The PR type is highlighted.
- The **Find** button provides a means of performing a keyword search of the **PR Type Value-Priority** list.
  - A keyword search is performed by typing relevant text in the text entry box to the right of the **Find** button then clicking on the **Find** button.
    - The first entry in the **PR Type Value-Priority** list that has the search text is highlighted.

- Click on the **Find** button again (as necessary) to highlight additional occurrences of the search text.

9 If defining or modifying (not deleting) a priority, click on the up/down arrow buttons to the right of the **Priority** field until the desired priority value is displayed in the **Priority** field.

- An alternative method of entering the priority is to type the desired priority value in the **Priority** field.
- The acceptable range for the priority is from 1 to 10.

10 Click on the appropriate button from the following selections:

- **Add** - to approve a priority for an additional PR type and display the selected PR type and priority in the **PR Type Value-Priority** list at the left center of the GUI.
- **Modify** - to approve a revised priority for the selected PR type and display the PR type and modified priority in the **PR Type Value-Priority** list.
- **Delete** - to delete the priority for the selected PR type and remove the PR type and priority from the **PR Type Value-Priority** list.

11 Repeat Steps 6 through 10 as necessary until all PR Type priorities (as shown in the **PR Type Value-Priority** field) are correct.

12 If defining or modifying a priority for a type of user, first click on the **User Type** button.

- The different types of users are displayed in the **Type List** field at the bottom left of the GUI.

13 If defining a priority for a type of user **not** currently listed in the **User Type Value-Priority** list, click on that user type in the **Type List** field.

- The user type is highlighted.
- It is possible to highlight multiple user types (by clicking on each one in turn).

14 If redefining or deleting a priority for a user type **already** listed in the **User Type Value-Priority** list, click on that user type in the **Value-Priority** list.

- The user type is highlighted.
- It is possible to highlight multiple user types (by clicking on each one in turn while holding down either the **Shift** key or the **Ctrl** key).

- The **Find** button provides a means of performing a keyword search of the **User Type Value-Priority** list.
  - A keyword search is performed by typing relevant text in the text entry box to the right of the **Find** button then clicking on the **Find** button.
    - The first entry in the **User Type Value-Priority** list that has the search text is highlighted.
    - Click on the **Find** button again (as necessary) to highlight additional occurrences of the search text.

15 If defining or modifying (not deleting) a priority, click on the up/down arrow buttons to the right of the **Priority** field until the desired priority value is displayed in the **Priority** field.

- An alternative method of entering the priority is to type the desired priority value in the **Priority** field.
- The acceptable range for the priority is from 1 to 10.

16 Click on the appropriate button from the following selections:

- **Add** - to approve a priority for an additional user type and display the selected user type and priority in the **User Type Value-Priority** list near the center of the GUI.
- **Modify** - to approve a revised priority for the selected user type and display the user type and modified priority in the **User Type Value-Priority** list.
- **Delete** - to delete the priority for the selected user type and remove the user type and priority from the **User Type Value-Priority** list.

17 Repeat Steps 12 through 16 as necessary until all user type priorities (as shown in the **User Type Value-Priority** field) are correct.

18 If defining a priority for a type of PGE, first click on the **PGE Type** button.

- The different types of PGEs are displayed in the **Type List** field at the bottom left of the GUI.

19 If defining a priority for a type of PGE **not** currently listed in the **PGE Type Value-Priority** list, click on that PGE type in the **Type List** field.

- The PGE type is highlighted.
- It is possible to highlight multiple PGE types (by clicking on each one in turn).

20 If redefining or deleting a priority for a PGE type **already** listed in the **PGE Type Value-Priority** list, click on that PGE type in the **Value-Priority** list.

- The PGE type is highlighted.
- It is possible to highlight multiple PGE types (by clicking on each one in turn while holding down either the **Shift** key or the **Ctrl** key).
- The **Find** button provides a means of performing a keyword search of the **PGE Type Value-Priority** list.
  - A keyword search is performed by typing relevant text in the text entry box to the right of the **Find** button then clicking on the **Find** button.
    - The first entry in the **PGE Type Value-Priority** list that has the search text is highlighted.
    - Click on the **Find** button again (as necessary) to highlight additional occurrences of the search text.

21 If defining or modifying (not deleting) a priority, click on the up/down arrow buttons to the right of the **Priority** field until the desired priority value is displayed in the **Priority** field.

- An alternative method of entering the priority is to type the desired priority value in the **Priority** field.
- The acceptable range for the priority is from 1 to 10.

22 Click on the appropriate button from the following selections:

- **Add** - to approve a priority for an additional PGE type and display the selected PGE type and priority in the **PGE Type Value-Priority** list near the center of the GUI.
- **Modify** - to approve a revised priority for the selected PGE type and display the PGE type and modified priority in the **PGE Type Value-Priority** list.
- **Delete** - to delete the priority for the selected PGE type and remove the PGE type and priority from the **PGE Type Value-Priority** list.

23 Repeat Steps 18 through 22 as necessary until all PGE type priorities (as shown in the **PGE Type Value-Priority** field) are correct.

24 Click in the **Weight** field below the **PR Type** button and type the desired weight.

- The acceptable range for weights is from 1 to 100.
- The **Total Weight** field displays updated totals of all weighting factors as they are entered.

- When entering weights for the PR Type, User Type, PGE Type, and Production Request Editor [Priority] factors, relative values can be typed in without regard to whether the values in the four categories add up to 100. The **Normalize** button provides a means of eventually ensuring that the total of all four categories equals 100.
- The assigned weight in each category is multiplied by the priority for each type. To maintain a high priority (low number, such as one), assign a low weight; to ensure a low priority, assign a relatively high weight.

25 Click in the **Weight** field below the **User Type** button and type the desired weight.

26 Click in the **Weight** field below the **PGE Type** button and type the desired weight.

27 Click in the **Production Request Editor** field and type the desired weight.

- The priority to which the Production Request Editor weight is applied is the priority assigned using the Production Request Editor when a production request is created.

28 Click on the **Normalize** button.

- The Planning Subsystem adjusts all weighting factors to produce a total weight of 100 (as displayed in the **Total Weight** field).

29 If it is necessary to change the priority of jobs that have been waiting in the Production Queue for more than a day, click in the **Late Start Delta Priority** field and type the desired value.

- The range for the Late Start Delta Priority is from 1 to 100.

30 Select **File → Save As** from the pull-down menu.

- A **Save As** window (Figure 42) is displayed.

31 Type the desired file name for the new production strategy in the **Save As** field.

32 Click on the appropriate button from the following selections:

- **OK** - to accept the file name in the **Save As** field.
  - The **Save As** window is dismissed.
  - The production strategy is saved with the specified file name.
- **Cancel** - to dismiss the **Save As** window without saving the production strategy.

NOTE: The **Help** button at the bottom of **Production Strategies Save As** window is non-functional (does not work).

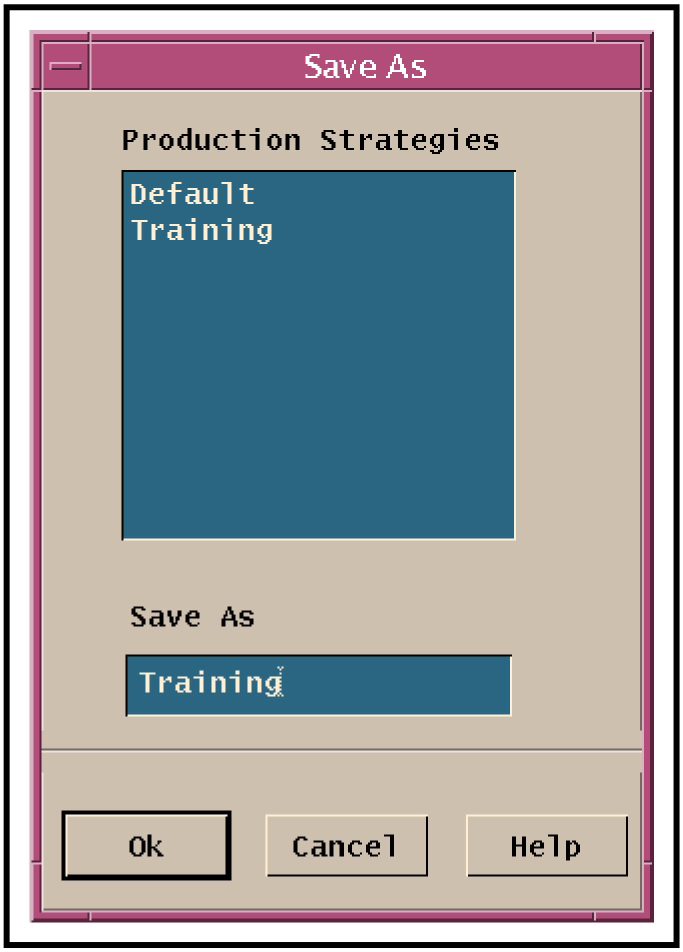

Figure 42. Production Strategies "Save As" Window

33 To start the process of modifying or creating another production strategy select **File** → **New** to clear the entries on the **Production Strategies** GUI.

- Return to Step 1 to create another new production strategy.
- Return to Step 2 to modify an existing production strategy.

34 To start the process of exiting from the **Production Strategies** GUI select **File** → **Exit** from the pull-down menu.

- A **Do you really want to exit?** dialogue box is displayed.

35 Click on the appropriate button from the following selections:

- **OK** - to exit from the **Production Strategies** GUI.
  - The **Production Strategies** GUI is dismissed.

- **Cancel** - to return to the **Production Strategies** GUI.

Reviewing the Current Active Strategy

The Production Planner uses the Production Strategies GUI to review the production strategy applied to the currently active production plan. The procedure that follows describes how to review the current active strategy.

The procedure for reviewing the current active strategy starts with the assumption that all applicable servers are currently running and the **Production Strategies** GUI (Figure 35) is being displayed.

Reviewing the Current Active Strategy

1 Select **Options** → **activeStrategy** from the pull-down menu.

- The **Active Production Strategy** window (Figure 43) is displayed.

2 Review the data displayed in the **Active Production Strategy** window:

- PR Type.
  - Weight.
  - Default [value].
  - [Individual PR types and values if any are listed.]
- User Type.
  - Weight.
  - Default [value].
  - [Individual user types and values if any are listed.]
- PGE Type.
  - Weight.
  - Default [value].
  - [Individual PGE types and values if any are listed.]
- Production Request Editor Weight.
- Late Start Delta Priority.

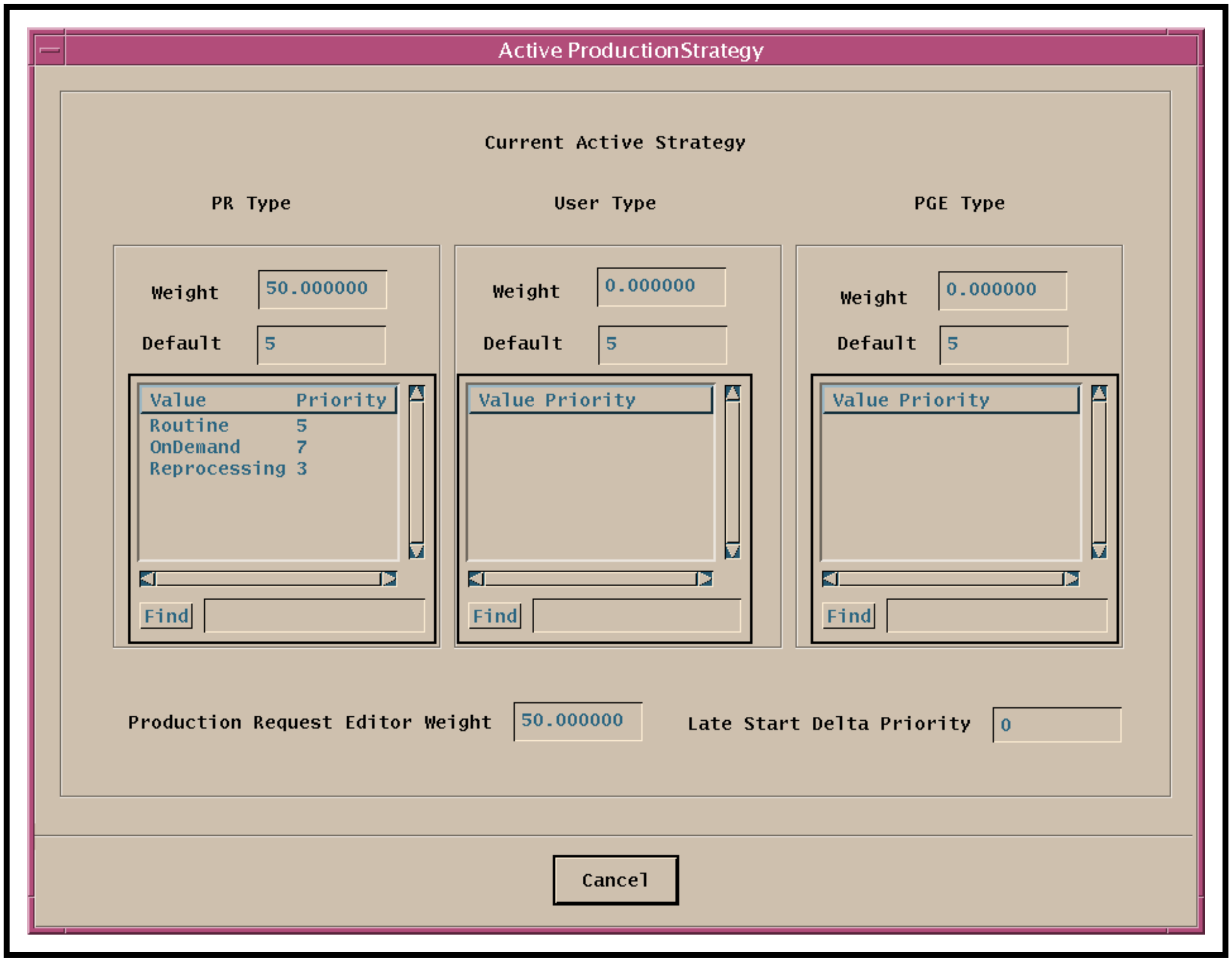

Figure 43. Active Production Strategy Window

3 To exit from the **Active Production Strategy** window and return to the **Production Strategies** GUI click on the **Cancel** button.

- The Active Production Strategy window is dismissed.
- The **Production Strategies** GUI is displayed.

Deleting a Production Strategy

The Production Planner uses the Production Strategies GUI to delete production strategies that are no longer needed. The procedure that follows describes how to delete a production strategy.

The procedure for deleting a production strategy starts with the assumption that all applicable servers are currently running and the **Production Strategies** GUI (Figure 35) is being displayed.

Deleting a Production Strategy

---

1 Click on the option button associated with the **Production Strategies** field, then highlight (in the option menu) the name of the production strategy to be deleted.

- Data pertaining to the selected production strategy are displayed in the applicable fields of the **Production Strategies** GUI.
- Alternatively, it is possible to select **File** → **Open** from the pull-down menu, select the desired production strategy from the list on the **Open** window (Figure 41), and click on the **Ok** button to open the production strategy.

2 Select **Edit** → **Delete** from the pull-down menu.

- A production strategy deletion confirmation dialogue box (Figure 44) is displayed requesting confirmation of the decision to delete the production strategy.
- The **Active Production Strategy** window is displayed.

Figure 44. Production Strategy Deletion Confirmation Dialogue Box

3 Click on the appropriate button from the following selections:

- **OK** - to confirm deletion of the production strategy and dismiss the dialogue box.
  - The deletion confirmation dialogue box is dismissed.
  - The production strategy is deleted.
- **Cancel** - to preserve the production strategy and dismiss the dialogue box.
  - The deletion confirmation dialogue box is dismissed.

– The production strategy is not deleted.

4 To start the process of exiting from the **Production Strategies** GUI select **File → Exit** from the pull-down menu.

- A **Do you really want to exit?** dialogue box is displayed.

5 Click on the appropriate button from the following selections:

- **OK** - to exit from the **Production Strategies** GUI.
  - The **Production Strategies** GUI is dismissed.
- **Cancel** - to return to the **Production Strategies** GUI.

Production Plans

Creating a New Production Plan

The Production Planner creates a production 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.

Before creating a new production plan the Production Planner must be prepared to provide the following information:

- Name of the plan.
- PRs to be included in the new production plan.
- Comments (if any).

The Production Planner uses the Planning Workbench GUI to prepare Production Plans. The procedure that follows describes how to create a new Production Plan. The procedure starts with the assumption that all applicable servers are currently running and the **Planning Workbench** GUI (Figure 37) is being displayed.

Creating a New Production Plan

---

1 Select **File → New** on the **Planning Workbench** GUI.

- The **New Plan** window (Figure 45) is displayed.

2 Type the ***Plan Name*** in the **Plan Names** field.

- Name is displayed in **Plan Names** field.

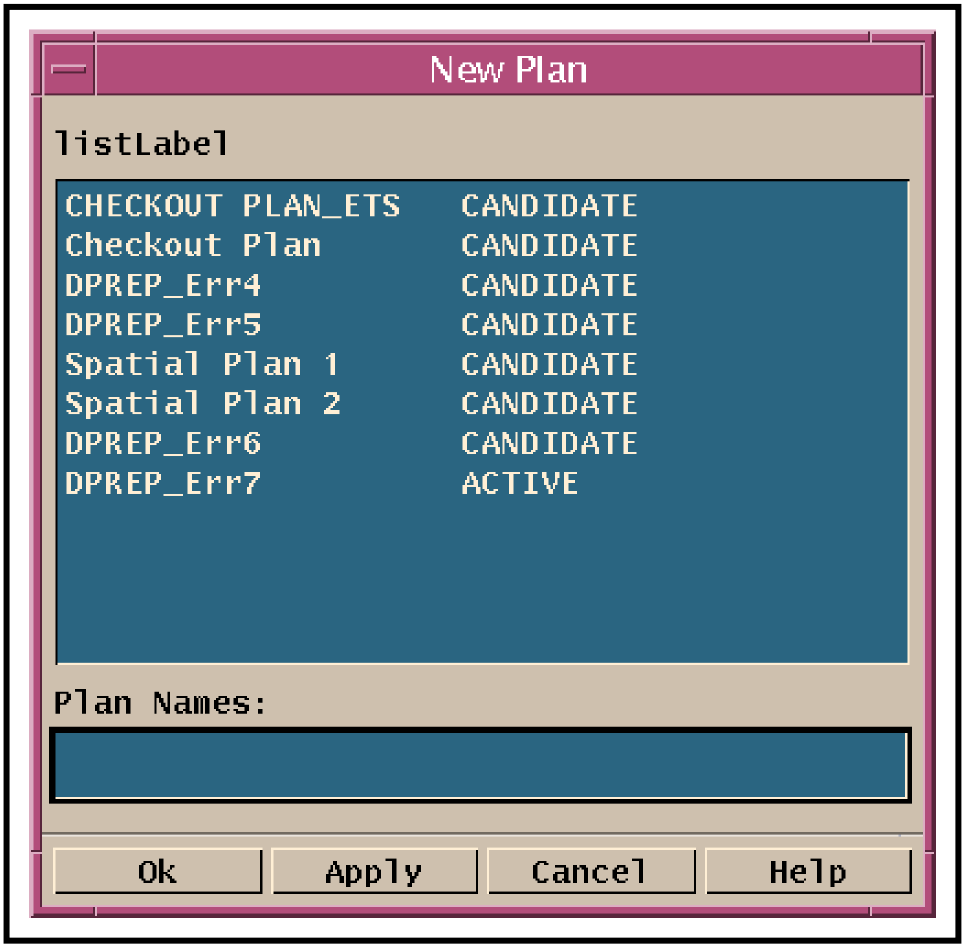

Figure 45. Planning Workbench New Plan Window

- Status is **Candidate**.

3 Click on the appropriate button from the following selections:

- **Ok** - to accept the file name in the **Plan Names** field.
  - The **New Plan** window is dismissed.
  - The production plan is saved with the specified file name.
- **Apply** - to save the production plan without dismissing the **New Plan** window.
  - The production plan is saved with the specified file name.
- **Cancel** - to dismiss the **New Plan** window without saving the production plan.

NOTE: The **Help** buttons at the bottom of various **Planning Workbench** windows are non-functional (do not work).

4 If applicable, click on the option button associated with the **Strategy** field, then highlight the desired production strategy in the option menu.

5 Move PRs between the **Unscheduled** and **Scheduled** lists as necessary by highlighting (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.
- In the **Scheduled** list, items with the prefix "GE_" are resource reservations (also called "ground events").
  - Ground events are resource reservations for non-production-related purposes, including such activities as testing, preventive maintenance, or system upgrades.
  - Ground events are scheduled through the resource planning process.
- If processing of the currently active plan is to be continued when the new plan is activated, include the PR(s) for the currently active plan in the new plan.

6 Type any relevant comments (up to 255 characters) in the **Comments** field.

7 If the priority of any PR in the **Scheduled** list needs to be changed, perform Steps 8 through 12; otherwise go to Step 13.

8 Click on the PR entry in the **Scheduled** list to highlight it.

9 Click on the **Prioritize** button.

- The **Priority popup** window (Figure 46) is displayed.

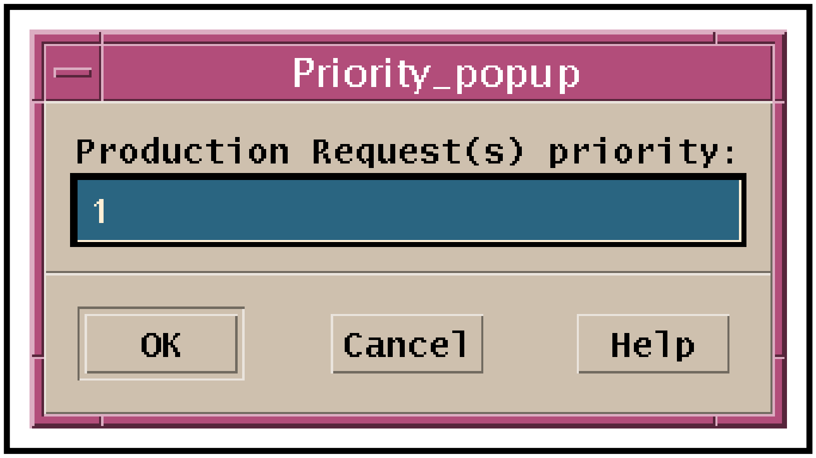

Figure 46. Priority Popup Window

10 Type the new priority in the **Production Request(s) priority** field.

11 Click on the appropriate button from the following selections:

- **OK** - to accept the new priority in the **Production Request(s) priority** field.
  - The **Priority popup** window is dismissed.
  - The new priority for the PR is saved in the database.
- **Cancel** - to dismiss the **Priority popup** window without saving the new priority.

12 Repeat Steps 8 through 11 for any additional PR(s) needing a change of priority.

13 Select **File** → **Save As** from the pull-down menu.

- The **Save Plan** window is displayed.
- Except for the title, the **Save Plan** window looks the same as the **New Plan** window (Figure 45).

14 If the ***Plan Name*** for the production plan is not displayed in the **Plan Names** field, type the ***Plan Name*** in the **Plan Names** field.

15 Click on the appropriate button from the following selections:

- **Ok** - to accept the file name in the **Plan Names** field.
  - The **Save Plan** window is dismissed.
  - The production plan is saved with the specified file name.
  - The **Planning Workbench** GUI (Figure 37) is displayed.
  - The **Plan Name** is displayed.
  - The **Status** displayed is **Candidate**.
- **Apply** - to save the production plan without dismissing the **Save Plan** window.
  - The production plan is saved with the specified file name.
- **Cancel** - to dismiss the **Save Plan** window without saving the production plan.

16 If the production plan is to be activated immediately, perform Steps 17 and 18; otherwise, go to Step 19.

17 Click on the **Activate** button.

- A **Confirm Activation** dialogue box (Figure 47) is displayed.

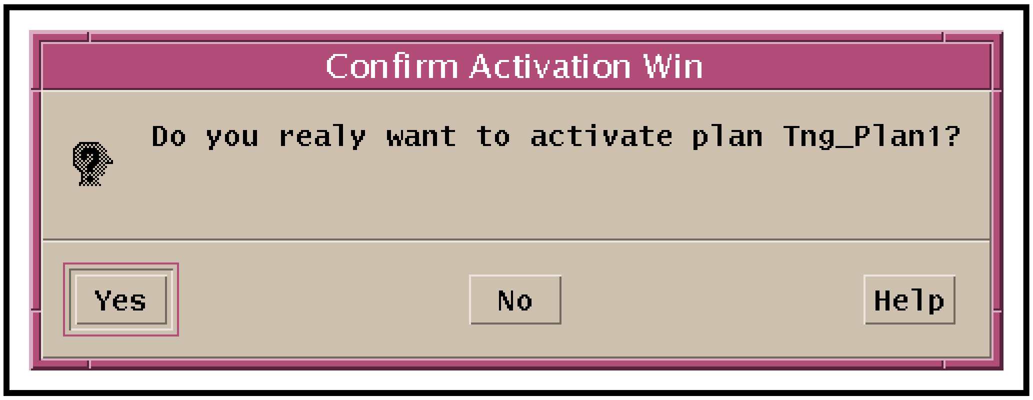

Figure 47. Confirm Activation Dialogue Box

18 Click on the appropriate button from the following selections:

- **Yes** - to activate the plan.
  - The **Confirm Activation** dialogue box is dismissed.
  - The new plan is activated.
  - The time of plan activation is displayed next to **Rollover Time** on the **Planning Workbench** GUI.
- **No –** to dismiss the **Confirm Activation** dialogue box without activating the plan.

19 If the production plan is to be used as a baseline plan, perform Steps 20 through 24; otherwise, go to Step 25.

20 Click on the **Baseline** button.

- A confirmation dialogue box containing the message **The current plan is *Plan Name*. Do you wish to baseline it?** is displayed.
- Clicking on the **Baseline** button records the plan and the time of baselining.
- A baseline plan can be used as a point of comparison with which to compare future plans and results.

21 Click on the appropriate button from the following selections:

- **Yes** - to baseline the plan.
  - The confirmation dialogue box containing the message **The current plan is *Plan Name*. Do you wish to baseline it?** is dismissed.
  - The plan is baselined.
- **No –** to dismiss the confirmation dialogue box containing the message **The current plan is *Plan Name*. Do you wish to baseline it?** without baselining the plan.

22 Select **File → Save As** from the pull-down menu.

- The **Save Plan** window is displayed.

23 If the ***Plan Name*** for the production plan is not displayed in the **Plan Names** field, type the ***Plan Name*** in the **Plan Names** field.

24 Click on the appropriate button from the following selections:

- **Ok** - to accept the file name in the **Plan Names** field.
  - The **Save Plan** window is dismissed.
  - The production plan is saved with the specified file name.
- **Apply** - to save the production plan without dismissing the **Save Plan** window.
  - The production plan is saved with the specified file name.
- **Cancel** - to dismiss the **Save Plan** window without saving the production plan as a baseline plan.

25 Repeat Steps 1 through 24 to perform additional production planning activities.

26 To view the Planning Master Timeline perform the procedure for **Reviewing a Plan Timeline** (subsequent section of this lesson).

27 To start the process of exiting from the **Planning Workbench** GUI select **File → Exit** from the pull-down menu.

- A **Do you really want to exit?** dialogue box is displayed.

28 Click on the appropriate button from the following selections:

- **OK** - to exit from the **Planning Workbench** GUI.
  - The **Planning Workbench** GUI is dismissed.
- **Cancel** - to return to the **Planning Workbench** GUI.

29 After quitting the **Planning Workbench** GUI click in the UNIX window used to start the **Planning Workbench** GUI.

- The Message Handler, System Name Server, and Resource Model should be shut down to eliminate unneeded processes and allow other operators to gain access to the Planning Workbench if necessary.

30 Type **EcPlSlayAll** ***MODE ApplicationID*** then press **Return/Enter** to shut down the Planning Master Timeline, Message Handler, System Name Server, and Resource Model (and the Planning Workbench if it has not already been shut down).

- The **Message Handler** GUI (Figure 36) disappears.

31 Type **ps -ef | grep** ***MODE*** then press **Return/Enter** to obtain a list of active processes in the specified mode.

- A list of active processes in the specified mode is displayed.
- If an error message is received when **ps -ef | grep** ***MODE*** is entered, type **ps -auxwww | grep** ***MODE*** then press **Return/Enter**.

32 Examine the list of processes running in the specified mode to determine whether the Message Handler, System Name Server, and Resource Model processes have actually been shut down.

- None of the following processes should be active:
  - EcPlWb
  - EcPlTl
  - EcPlMsh
  - EcPlSns
  - EcPlRm

33 If any of the specified processes [especially the Message Handler, System Name Server, and/or Resource Model process(es)] is/are still active, type **kill -15** ***process_id1 [process_id2] [process_id3] […]*** to terminate the active process(es).

34 Repeat Steps 31 through 33 as necessary.

Reactivation/Replanning

"Reactivation" or "replanning" assumes that there is a current plan. The current plan may or may not have DPRs associated with it.

- If there are no DPRs in a plan, it may be because all the DPRs that were in the plan have run to completion.
- Alternatively, there may never have been any DPRs associated with the plan because an empty plan was activated to cancel DPRs from a previously activated plan.

Reactivation/replanning occurs in the following three types of situations:

- No DPR in old plan but DPR(s) in new plan.
  - A current active plan without DPRs (e.g., the DPRs have finished processing) can be re-planned/reactivated as a new plan with DPRs.
- DPR(s) in old plan but not in new plan.
  - A current active plan with DPRs in the Job Management queue can be re-planned/reactivated as a new plan without DPRs (e.g., to cancel the DPRs that have been released to Data Processing but that have not started processing yet).
- DPR(s) in both old plan and new plan.
  - A current active plan with DPRs can be re-planned/reactivated as a new plan with those same DPRs (e.g., to add DPRs to a plan without canceling DPRs already in the plan).

No DPR in Old Plan but DPR(s) in New Plan

The overall process for replanning when there are no DPRs in the old plan but there will be DPRs in the new plan occurs as follows:

- Enter a new plan name.
  - The plan is created using the **Planning Workbench** GUI.
- Select the PRs to be included.
- Schedule the PRs to be activated.
- Save the new plan.
- Activate the new plan.
  - The new plan is activated and the new DPRs are ready to run.
- Check on the state of the jobs in AutoSys.
  - New DPRs should be in AutoSys and should begin to run.

DPR(s) in Old Plan but Not in New Plan

The overall process for replanning when there are DPRs in the old plan but there will be no DPRs in the new plan occurs as follows:

- Verify that DPRs in the active plan are in the Job Management queue.
  - The current DPRs should be in the queue and can be viewed by pressing the **Jobs Waiting** button on the AutoSys **Job Activity Console (Ops Console)**.

- Create a new plan without PRs/DPRs to replace the old plan (with DPRs).
  - Enter a new plan name.
    - The plan is created using the **Planning Workbench** GUI.
  - Do not schedule any PRs to be activated.
  - Save the new plan.
- Activate the new plan.
- Verify the state of the new plan's DPRs.
  - Use **JobScape** to verify the state of the new jobs.
- Verify the state of the old (existing) plan's DPRs.
  - Use the **Jobs Waiting** button on the AutoSys **Job Activity Console (Ops Console)** to verify the cancellation of old DPR jobs.
  - Only jobs that are in the Job Management queue are cancelled.
  - Jobs that have already been released to AutoSys are not deleted during the replan (they continue processing to termination).

DPR(s) in Both Old Plan and New Plan

The overall process for replanning when there are DPRs in both the old plan and the new plan occurs as follows:

- Verify that DPRs in the active plan are in the queue.
  - The current DPRs should be in the Job Management queue and can be viewed using the **Jobs Waiting** button on the AutoSys **Job Activity Console (Ops Console)**.
- Enter a new plan name.
  - The plan is created using the **Planning Workbench** GUI.
- Select the PRs to be included.
  - Include relevant PRs used in the old plan; i.e., those PRs with DPRs to be re-prioritized that are in the Job Management queue.
- Schedule the PRs to be activated.
- Save the new plan.
- Activate plan.

- Verify the priorities of the plan's DPRs in the Job Management queue.
  - The new DPR(s) must be in the Job Management queue.
  - The old DPR(s) must be cancelled.
  - If priorities differ between old and new DPR, the old DPR should have been cancelled and a new DPR created.
  - Only jobs that are in the Job Management queue have their priorities changed.
  - Jobs that have already been released to AutoSys are not changed during the replan because their priorities are no longer relevant.

Deleting a Production Plan

The Production Planner uses the Planning Workbench GUI to delete production plans that are no longer needed. The procedure that follows describes how to delete a production plan.

The procedure for deleting a production plan starts with the assumption that all applicable servers are currently running and the **Planning Workbench** GUI (Figure 37) is being displayed.

Deleting a Production Plan

---

1 Select **File → Delete** on the **Planning Workbench GUI**.

- The **Delete Plan** window (Figure 48) is displayed.
- A list of Production Plans is displayed in the **Delete Plan** window.

2 Select (highlight) the production plan to be deleted by clicking on the corresponding name in the list of plans.

- The plan name is displayed in the **Plan Names** field of the **Delete Plan** window.

3 Click on the **OK** button.

- The **Delete Plan** window is dismissed.
- The production plan is deleted.

Reviewing a Plan Timeline

It is possible to display a graphic, timeline-oriented depiction of a production plan, as shown in Figure 38. The timeline application becomes available when the Production Planning Workbench is initiated. The display shows a set of processing equipment, arranged along the left side of the GUI, and some period of time as indicated across the top edge of the GUI.

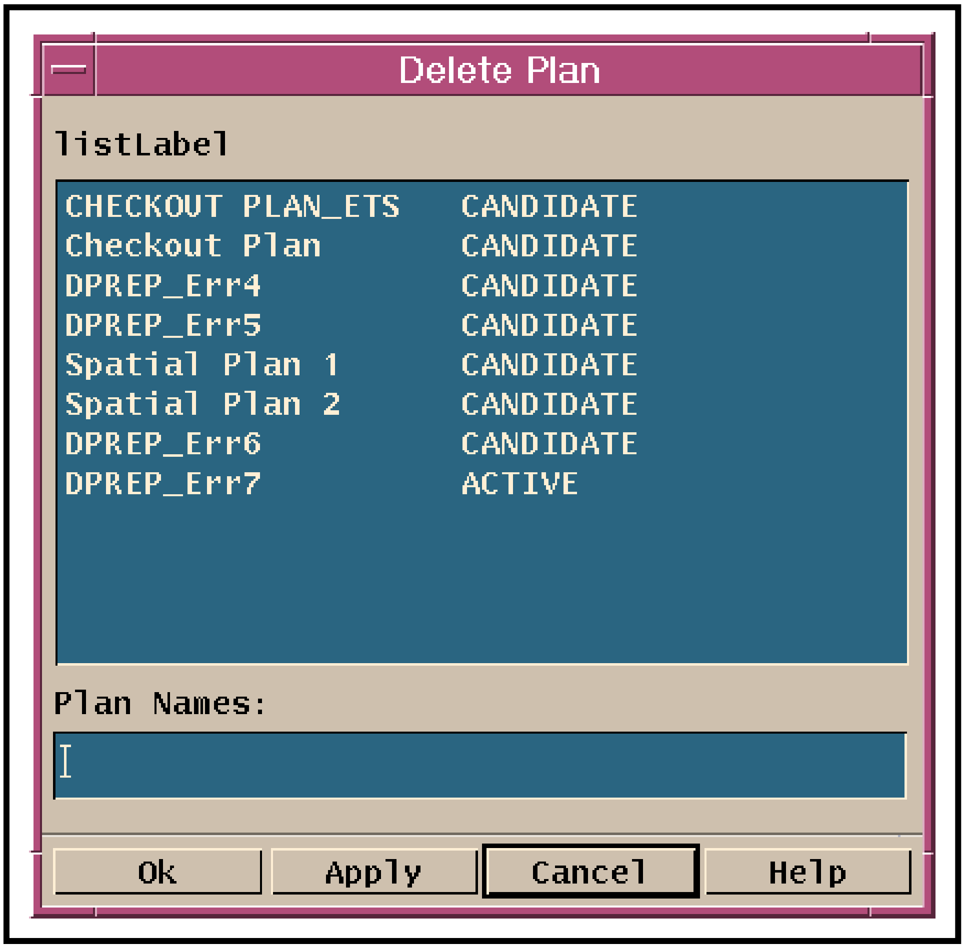

Figure 48. Delete Plan Window

The execution of DPRs on the processing equipment over a period of time is represented by several DPR bars across the GUI for that equipment. In addition, there may be bars that represent resource reservations for non-production-related purposes, which are also called "ground events." They are scheduled through the resource planning process. Ground events include such activities as testing, preventive maintenance, or system upgrades.

- A bar represents a time period during which a DPR is to be processed or a resource reservation has been planned.
- Each bar bears the name of a DPR or a resource reservation. Given the selection of a light enough color for the bar and a time span that allows a long enough bar, the name of the DPR or resource reservation can be seen on the bar.
  - Placing the cursor on a bar causes the name of the DPR (or resource reservation), its description, and its start and end dates/times to appear near the bottom of the timeline GUI.
  - Resource reservations are identified by the prefix "GE_".

The procedure for reviewing a production plan timeline starts with the assumption that all applicable production planning servers are running and the **Production Planning Master Timeline** GUI (Figure 38) is being displayed.

Reviewing a Plan Timeline

---

1 Adjust the **Production Planning Master Timeline** window size and the view of the timeline as necessary using the mouse.

- Grab a corner of the timeline window with the cursor and resize the window as desired.
- Scroll up or down through the full list of equipment.
- Scroll left or right to go backward or forward in time.

2 If a different plan is to be viewed (other than the one currently being displayed), select **File → Open Plan** from the pull-down menu.

- The **Open Plan** window (Figure 49) is displayed.
- The available plans are listed.

3 Select (highlight) the plan to be reviewed by clicking on the corresponding plan name in the list of plans.

4 Click on the appropriate button from the following selections:

- **OK** - to open the selected plan and dismiss the **Open Plan** window.
  - The timeline for the specific plan is displayed on the **Production Planning Master Timeline**.
  - Name is displayed in the Title bar.
- **Apply** - to open the selected plan without dismissing the **Open Plan** window.
  - The timeline for the specific plan is displayed on the **Production Planning Master Timeline**.
  - Name is displayed in the Title bar.
- **Cancel** - to dismiss the **Open Plan** window without opening any plan.

NOTE: The **Help** buttons at the bottom of various **Production Planning Master Timeline** windows are non-functional (do not work).

5 If a different time scale (start and end dates and times) is desired, perform Steps 6 through 8; otherwise, go to Step 9.

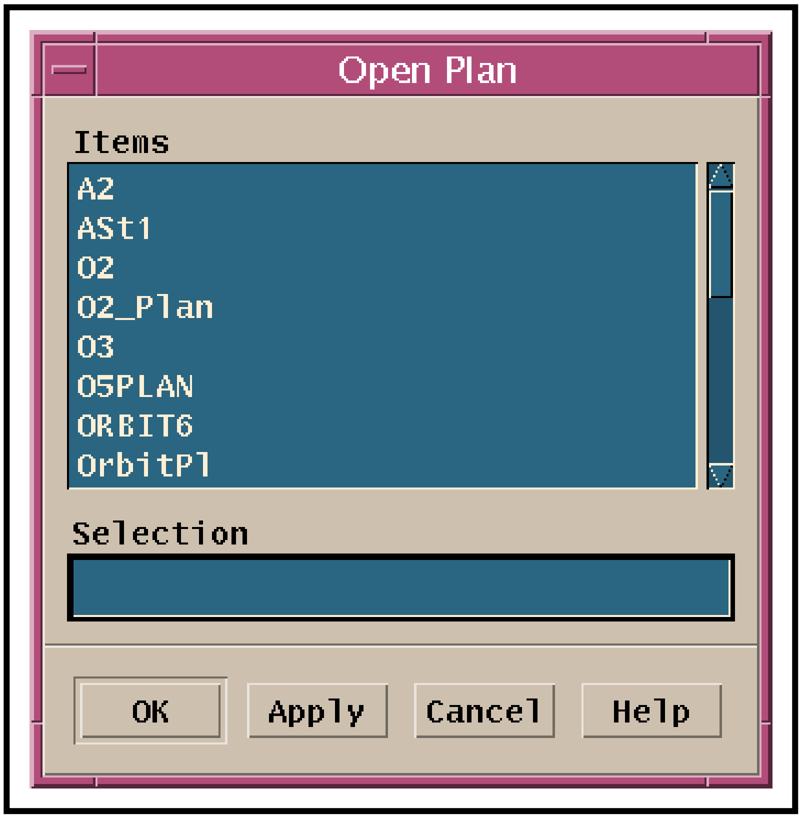

Figure 49. Open Plan Window

6 Select **Time** → **Change Plan Window** from the pull-down menu:

- The **plan window edit** window (Figure 50) is displayed.

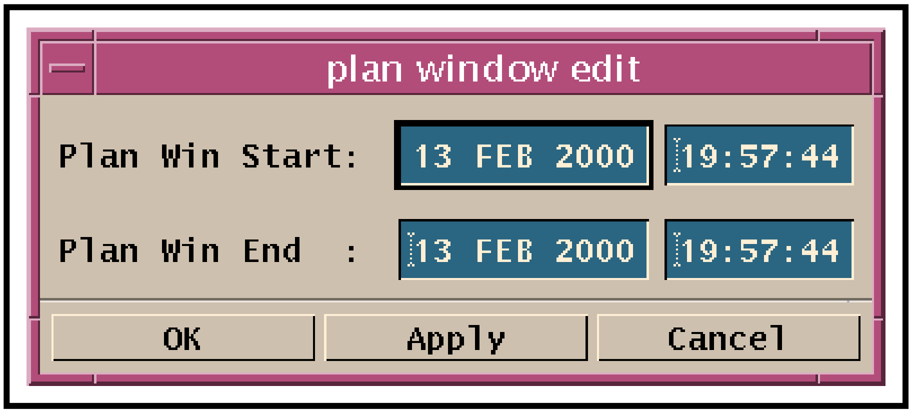

Figure 50. Plan Window Edit Window

7 Type date and time for the desired start and end times (in ***DD MMM YYYY hh:mm:ss*** format) in the **Plan Win Start** and **Plan Win End** fields of the **plan window edit** window.

8 When the appropriate date and time have been entered, click on the appropriate button from the following selections:

- **OK** - to accept the changes and dismiss the **plan window edit** window.
- **Apply** - to accept the changes without dismissing the **plan window edit** window.
- **Cancel** - to cancel the changes and dismiss the **plan window edit** window.

9 If a different time span is desired, click and hold on the **Show** option button and select (highlight then release the mouse button) the desired time span from the option menu that is displayed:

- **5 min**
- **10 min**
- **30 min**
- **45 min**
- **1 hr**
- **2 hr**
- **4 hr**
- **6 hr**
- **12 hr**
- **24 hr**
- **168 hr**
- **other**

10 If no resources are displayed on the GUI or if different resources should be displayed, perform Steps 11 through 15; otherwise, go to Step 16.

11 Select **Display** → **Change resources** from the pull-down menu:

- The **Resource edit** window (Figure 51) is displayed.

12 If adding resource(s) from the **Available Resources** list to the **Viewed Resources** list, select (highlight) the resource(s) to be added, then click on the **Add** button to move the resource(s) to the **Viewed Resources** list.

- Highlighted resource(s) appear(s) on the **Viewed Resources** list.

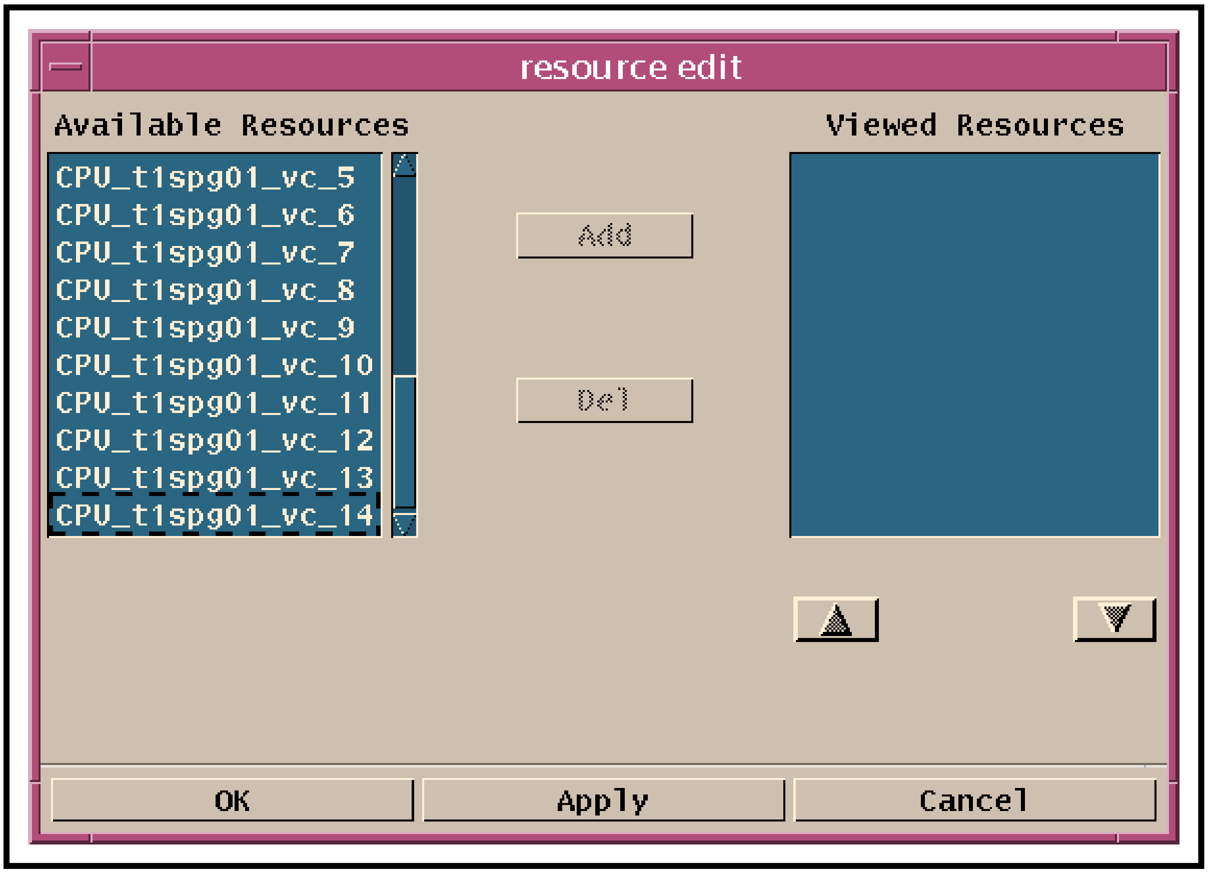

Figure 51. Resource Edit Window

13 If deleting resource(s) from the **Viewed Resources** list, select (highlight) the resource(s) to be removed, then click on the **Del** button to remove the resource(s) from the **Viewed Resources** list.

- Highlighted resource(s) disappear(s) from the **Viewed Resources** list.

14 If changing the order in which resources are listed in the **Viewed Resources** list, select (highlight) the resource to be moved, then click on the up or down arrow as necessary to reposition the selected resource.

- Highlighted resource changes position in the **Viewed Resources** list.

15 When the **Viewed Resources** list contains the desired set of resources, click on the appropriate button from the following selections:

- **OK** - to accept the changes and dismiss the **Resource edit** window.
- **Apply** - to accept the changes without dismissing the **Resource edit** window.
- **Cancel** - to cancel the changes and dismiss the **Resource edit** window.

16 If different color-coding of the timeline is desired, perform Steps 17 through 21; otherwise, go to Step 22.

17 Select **Display** → **Change colors** from the pull-down menu:

- The Color Selections window (Figure 52) is displayed.

18 Click on the name of one of the DPRs or resource reservations to be recolored.

- The DPR or resource reservation is highlighted.

19 Click on the desired color (in the color palette) to be applied to the highlighted DPR or resource reservation.

20 Repeat Steps 18 and 19 as necessary.

21 When the appropriate color changes have been made, click on the appropriate button from the following selections:

- **OK** - to accept the changes and dismiss the **Color Selections** window.
- **Apply** - to accept the changes without dismissing the **Color Selections** window.
- **Cancel** - to cancel the changes and dismiss the **Color Selections** window.

22 Observe the production scheduling information displayed on the **Production Planning Master Timeline** GUI.

23 Repeat the previous steps as necessary.

24 To start the process of exiting from the **Production Planning Master Timeline** GUI select **File** → **Exit** from the pull-down menu.

- A **Do you really want to exit?** dialogue box is displayed.

25 Click on the appropriate button from the following selections:

- **OK** - to exit from the **Production Planning Master Timeline** GUI.
  - The **Production Planning Master Timeline** GUI is dismissed.
- **Cancel** - to return to the **Production Planning Master Timeline** GUI.

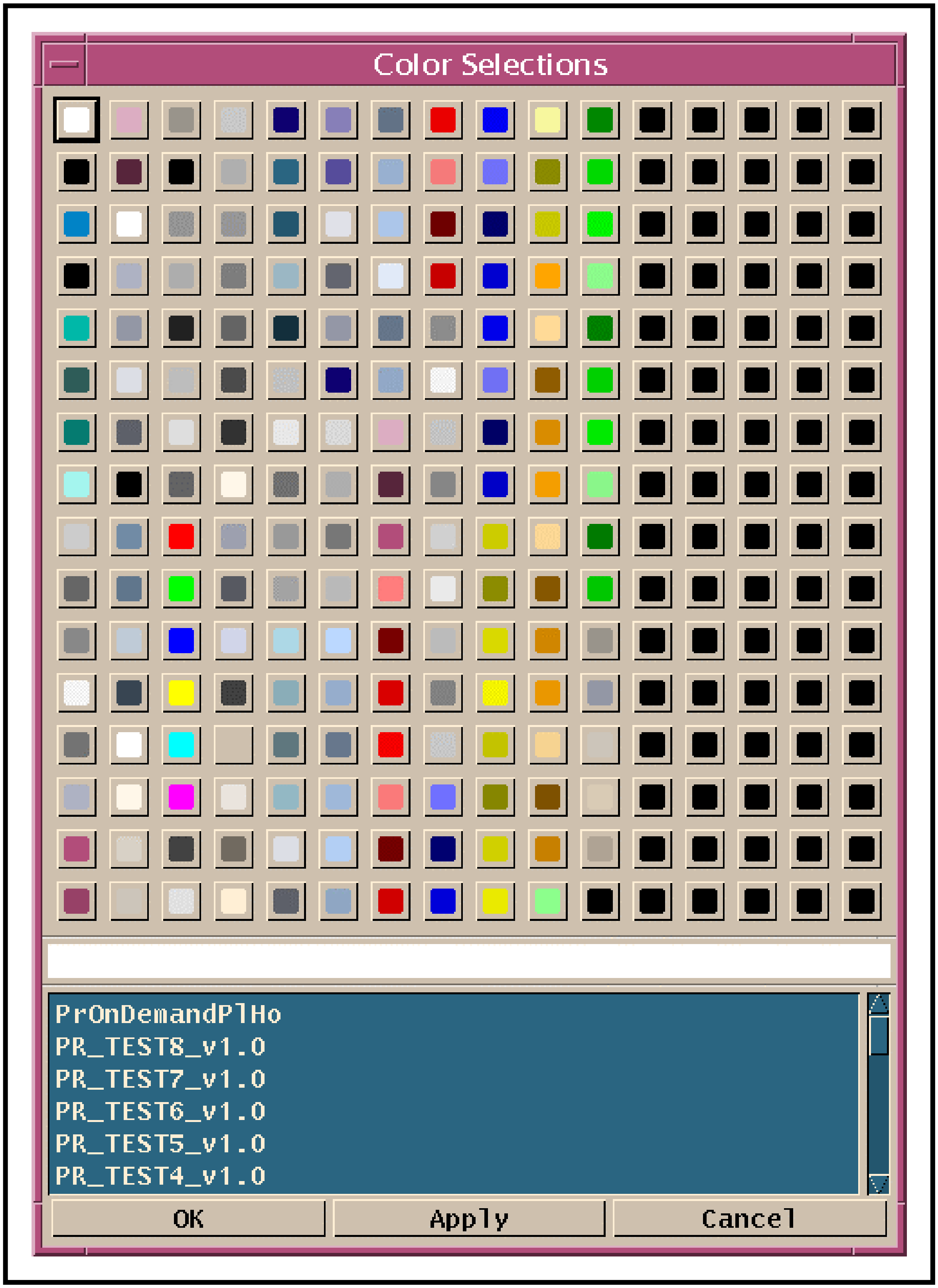

Figure 52. Color Selections Window

This page intentionally left blank.

Cleaning the PDPS Database and Science Processing Disks

Cleaning the PDPS Database and Science Processing Disks

At DAACs with a heavy data processing load it is essential to perform PDPS database cleanup and virtual-computer “garbage collection” on a regular basis. The frequency of cleanup is determined by each DAAC’s needs. Failure to perform cleanup is likely to lead to performance problems with the Planning and Data Processing Subsystems.

It is recommended that the database cleanup and “garbage collection” processes be run in the following order:

- Database cleanup.
- “Garbage collection” on the virtual computer(s).
- Database cleanup (again).

If not already done, it is possible to create a script to run the preceding processes. If it is necessary to run particular scripts individually, refer to the sections that follow.

NOTE: To the extent possible the scripts mentioned in this section should be run when the system is relatively quiet; i.e., when no PRs/DPRs are being created and no jobs are running in AutoSys.

Cleaning the PDPS Database

In the /usr/ecs/*MODE*/CUSTOM/utilities directory on the Planning/Management Workstation there is a script (i.e., EcPlDbClean) that can be run to clean up some tables in a PDPS database. When it runs, the script tries to delete applicable records in the following order:

- Data Processing Requests based on timeStamp and completionState(SUCC_DEL) (deletes all DPR records with a completionState value of SUCC_DEL that are older than the specified number of months/days).
- Production Requests that have no associated DPRs (e.g., deletes the Production Requests that are older than the specified number of months/days and for which all of the DPRs were successfully completed and subsequently deleted).
- Dynamic data granules that are not used by any DPR or by the Data Processing Subsystem (i.e., cleans up the PlDataGranuleShort table of entries for dynamic granules that are older than the specified number of months/days, are no longer needed by any DPR, and are not being used by DPS).

- PGEs that are marked with a deleteFlag [deletes PGE information in the PlPgeMaster table for any PGE for which the deleteFlag is set (not equal to zero) and assuming that there are no DPRs associated with the PGE(s)].
- Science Software that has no associated PGE [cleans up the PlResourceRequirement table if there is no PGE that has a particular sswId [science software ID] (e.g., the associated PGE(s) was/were previously deleted)].

In addition to deleting the previously mentioned database entries, the script compiles a list of data granules that are not deleted because the Data Processing Subsystem currently needs to use them.

The procedure for cleaning the PDPS database starts with the assumption that the applicable servers are running and the Production Planner has logged in to the system.

Cleaning the PDPS Database

---

1 Access a terminal window logged in to the Planning/Management Workstation host.

- Examples of Planning/Management Workstation host names include **e0pls03** and **l0pls02.**
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 Type **setenv ECS_HOME /usr/ecs/** then press the **Return/Enter** key.

- When logging in as a system user (e.g., cmshared), the ECS_HOME variable may be set automatically so it may not be necessary to perform this step.

3 Type **cd /usr/ecs/*MODE*/CUSTOM/utilities** then press **Return/Enter**.

- Change directory to the directory containing the production planning startup scripts (e.g., EcPlPRE_IFStart).
- The ***MODE*** will most likely be one of the following operating modes:
  - OPS (for normal operation).
  - TS1 (for Science Software Integration and Test (SSI&T)).
  - TS2 (new version checkout).
- Note that the separate subdirectories under /usr/ecs apply to (describe) different operating modes.

4 Type **EcPlDbClean** ***MODE dbuser dbpassword dbserver months days*** then press **Return/Enter** to start the database cleaning process.

- ***dbuser*** is the user name for logging in to interactive structured query language (isql).

- ***dbpassword*** is the password for isql login.
- ***dbserver*** refers to the name of the PDPS database server (e.g., x0sps02_srvr).
- ***months*** is a number specifying the removal of records that are older than that number of months.
- ***days*** is an optional argument. It is a number that specifies the removal of records that are older than that number of days.
- Both ***months*** and ***days*** are taken into account by the cleaning script.
- When the process has finished, the following type of message is displayed:

  Log is written to /usr/ecs/OPS/CUSTOM/logs/EcPlDbClean.log

5 When the EcPlDbClean process has finished, type **cd ../logs** then press **Return/Enter**.

- Change directory to the directory containing the production planning log files (including EcPlDbClean.log).

6 Type **pg** ***filename*** then press **Return/Enter**.

- ***filename*** refers to the log file to be reviewed (e.g., EcPlDbClean.log).
- The first page of the log file is displayed.
- Although this procedure has been written for the **pg** command, any UNIX editor or visualizing command (e.g., **more**, **view, vi**) can be used to review the log file.

7 Review the log file to determine the results of running the script.

- To exit from **pg** at the **:** prompt enter:

  q

  – The command line prompt is displayed.

Performing Garbage Collection

In the /usr/ecs/*MODE*/CUSTOM/utilities directory on the Queuing Server there is a script (i.e., EcDpPrGarbageCollectorStart) that can be run to delete unneeded files from the science processing disks and update the PDPS database accordingly.

The actual executable invoked by the EcDpPrGarbageCollectorStart script is EcDpPrDeletionClient. However, the EcDpPrGarbageCollectorStart script differs from the EcDpPrDeletionClientStart script in the following three ways:

- Does not open a separate xterm window.
- Requires specification of a MACHINE_TO_COLLECT variable.

- Includes retry logic in case of database deadlock.

The procedure for performing garbage collection starts with the assumption that the applicable servers (especially EcDpPrDeletion) are running and the Production Planner has logged in to the system.

Performing Garbage Collection

---

1 Access a terminal window logged in to the Queuing Server host.

- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 Type **setenv ECS_HOME /usr/ecs/** then press the **Return/Enter** key.

- When logging in as a system user (e.g., cmshared), the ECS_HOME variable may be set automatically so it may not be necessary to perform this step.

3 Type **cd /usr/ecs/*MODE*/CUSTOM/utilities** then press **Return/Enter**.

- Change directory to the directory containing the production processing startup scripts (e.g., EcDpPrAutosysStart).
- The ***MODE*** will most likely be one of the following operating modes:
  - OPS (for normal operation).
  - TS1 (for Science Software Integration and Test (SSI&T)).
  - TS2 (new version checkout).
- Note that the separate subdirectories under /usr/ecs apply to (describe) different operating modes.

4 Type **EcDpPrGarbageCollectorStart *MODE machine retries interval*** then press **Return/Enter** to start the garbage collection process.

- ***machine*** refers to the name of the virtual computer (e.g., x0spg01_vc) for which garbage collection is being requested.
  - The value entered is assigned to a MACHINE_TO_COLLECT variable.
- ***retries*** is the number of retries in case of database deadlock.
  - The default number of retries is five (5) [times].
- ***interval*** is the amount of time (in minutes) between retries.
  - The default interval is 30 [minutes].

5 Observe the results as the script runs.

- A UNIX command line prompt is displayed when the script has finished.

Running the Deletion Server Client

In the /usr/ecs/*MODE*/CUSTOM/utilities directory on the Queuing Server there is a script (i.e., EcDpPrDeletionClientStart) that can be run to delete unneeded files from the science processing disks and update the PDPS database accordingly. The executable invoked by the script is EcDpPrDeletionClient.

The procedure for running the Deletion Server Client starts with the assumption that the applicable servers (especially EcDpPrDeletion) are running and the Production Planner has logged in to the system.

Running the Deletion Server Client

1 Access a terminal window logged in to the Queuing Server host.

- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 Type **setenv ECS_HOME /usr/ecs/** then press the **Return/Enter** key.

- When logging in as a system user (e.g., cmshared), the ECS_HOME variable may be set automatically so it may not be necessary to perform this step.

3 Type **cd /usr/ecs/*MODE*/CUSTOM/utilities** then press **Return/Enter**.

- Change directory to the directory containing the production processing startup/utility scripts (e.g., EcDpPrAutosysStart).
- The ***MODE*** will most likely be one of the following operating modes:
  - OPS (for normal operation).
  - TS1 (for Science Software Integration and Test (SSI&T)).
  - TS2 (new version checkout).
- Note that the separate subdirectories under /usr/ecs apply to (describe) different operating modes.

4 Type **EcDpPrDeletionClientStart *MODE*** then press **Return/Enter** to start the Deletion Server Client process.

5 Observe the results as the script runs.

- A UNIX command line prompt is displayed when the script has finished.

Resolving PDPS Database and Science Processing Disk Content Discrepancies

In the /usr/ecs/*MODE*/CUSTOM/utilities directory on the Queuing Server there is a script (i.e., EcDpPrRmFilesWOGranules.pl) that can be run to ensure consistency between the file references in various tables in the PDPS database and the files actually staged on the science processing disks. When it runs, the script performs the following functions:

- Generates a list of files with consistent references among various tables in the PDPS database for the specified mode.
- Checks for files on the disk that are not included in the list of files referenced in the PDPS database and either lists the inconsistent files or generates a script to delete them (as specified by the person running the script).
- Checks to determine whether the disk partitions referenced in the PDPS database actually exist on the disk(s).
  - If the disk partitions are not on the disk(s), the script removes references to them from the database.
- Removes all file references in the PDPS database that are not included in the list of files.
  - The script resets the PlRscDiskPartition according to the remaining DpPrDiskPartition entries (if specified by the person running the script).

The procedure for resolving PDPS database and science processing disk content discrepancies starts with the assumption that the applicable servers (including the PDPS database server) are running and the Production Planner has logged in to the system.

NOTE: The **EcDpPrRmFilesWOGranules.pl** script should be run when the system is relatively quiet; i.e., when no jobs are running in AutoSys.

Resolving PDPS Database and Science Processing Disk Content Discrepancies

---

1 Access a terminal window logged in to the Queuing Server host.

- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 Type **cd /usr/ecs/*MODE*/CUSTOM/utilities** then press **Return/Enter**.

- Change directory to the directory containing the production processing utility/startup scripts (e.g., EcDpPrAutosysStart).
- The ***MODE*** will most likely be one of the following operating modes:
  - OPS (for normal operation).
  - TS1 (for Science Software Integration and Test (SSI&T)).
  - TS2 (new version checkout).
- Note that the separate subdirectories under /usr/ecs apply to (describe) different operating modes.

3 Type **EcDpPrRmFilesWOGranules.pl *dbuser dbpassword MODE dbserver* fix** [or **nofix**] then press **Return/Enter** to start the DPS disk cleaning process.

- ***dbuser*** is the user name for logging in to interactive structured query language (isql).
- ***dbpassword*** is the password for isql login.
- ***dbserver*** refers to the name of the PDPS database server (e.g., x0sps02_srvr).
- **fix** (or **nofix**) specifies whether or not inconsistent files are removed from the science processing disks.
- An option is to add **>& fix.log** or **>& /usr/ecs/*MODE*/CUSTOM/logs/fix.log** to the end of the command line to direct the output of the script to a log file.
- For example, the following command:

  EcDpPrRmFilesWOGranules.pl pdps_role password1 OPS x0sps02_srvr nofix

  produced the following results:

  server = x0sps02_srvr; username = pdps_role; mode = OPS;
  fix = nofix

  This script will do the following:

  a) Get a consistent set of files from database pdps for mode OPS

  b) If the staging disk(s) is(are) readable :

  i) Check for files that do not belong to the consistent set; List them in nonfix option, and in fix option generate a delete script called EcDpPrRmFilesWOGranulesDelScript

  ii) Check the consistent set to ensure that the entries pertaining to the readable partitions exist on disk; List them in nonfix, and

remove them from the consistent set in both cases

c) Finally remove all entries in the PDPS database that dont belong to this consistent set; If in fix option, reset PlRscDiskPartition according to the DpPrDiskPartition entries remaining.Found the following partitions on your database pdps:

partitionName : partitionId
x0spg01_disk : 1
x0aqg02_disk : 2

Getting consistent database entries.. please wait
DpPrDiskAllocation done
DpPrFile done
Created temporary table
Synced with DpPrDiskAllocation
Added met files

Processing data returned from SQL

Reading in filenames from disk

Files on disk having no consistent entry in the database for partition x0spg01_disk
Path of staging disk :
/usr/ecs/OPS/CUSTOM/pdps/x0spg01/data//DpPrRm/x0spg01_disk

0 total invalid files found

Files absent on disk but with consistent entries in the database for partition x0spg01_disk
fileName:AIRXADCM#001O20000000; granuleId:AIRXADCM#001O2000

1 total invalid dB entries found

no path for partition x0aqg02_disk

Analyzing PDPS database.. please wait

Looking for invalid entries

```
------ DpPrFile ------
 fileName                                         partitionId
 ------------------------------------------------ -----------
 AIRXADCM#001O20000000                                      1
 AIRXADCM#001O20000000.met                                  0
 AM1ATTF#0011231199922000000.FDD                            1
 AM1ATTF#00112311999220000000000.met                        0
```

AM1ATTF#00101012000000000000.FDD	**1**
AM1ATTF#001010120000000000000000.met	**0**
AM1ATTF#00101012000020000000.FDD	**1**
AM1ATTF#001010120000200000000000.met	**0**
MOD021KM#0010101200000400000000000.met	**0**
MOD021KM#001010120000040000000000	**1**
MOD021KM#0010101200000450000000000.met	**0**
MOD021KM#001010120000045000000000	**1**
MOD021KM#0010101200000500000000000.met	**0**
MOD021KM#001010120000050000000000	**1**
MOD021KM#0010101200000550000000000.met	**0**
MOD021KM#001010120000055000000000	**1**
MOD021KM#0010101200001000000000000.met	**0**
MOD021KM#001010120000100000000000	**1**
MOD021KM#0010101200001050000000000.met	**0**
MOD021KM#001010120000105000000000	**1**
MOD021KM#0010101200001100000000000.met	**0**
MOD021KM#001010120000110000000000	**1**
MOD021KM#0010101200001450000000000.met	**0**
MOD021KM#001010120000145000000000	**1**
MOD021KM#0010101200001400000000000.met	**0**
MOD021KM#001010120000140000000000	**1**
MOD021KM#0010101200001500000000000.met	**0**
MOD021KM#001010120000150000000000	**1**
MOD29#00101012000000500000000000.met	**0**
MOD29#0010101200000050000000000	**1**
MOD29#00101012000000150000000000.met	**0**
MOD29#0010101200000015000000000	**1**
MOD29#00101012000000100000000000.met	**0**
MOD29#0010101200000010000000000	**1**
MOD29#00101012000000200000000000.met	**0**
MOD29#0010101200000020000000000	**1**
MOD29#00101012000000250000000000.met	**0**
MOD29#0010101200000025000000000	**1**
MOD29#00101012000000300000000000.met	**0**
MOD29#0010101200000030000000000	**1**
MOD29#00101012000000350000000000.met	**0**
MOD29#0010101200000035000000000	**1**
MOD29#00101012000000400000000000.met	**0**
MOD29#0010101200000040000000000	**1**
MOD29#00101012000000450000000000.met	**0**

MOD29#0010101200000450000000000 1
MOD29#0010101200000500000000000.met 0
MOD29#0010101200000500000000000 1
MOD29#0010101200000550000000000.met 0
MOD29#0010101200000550000000000 1
MOD29#0010101200001000000000000.met 0
MOD29#0010101200001000000000000 1
MOD29#0010101200001050000000000.met 0
MOD29#0010101200001050000000000 1
MOD29#0010101200001450000000000.met 0
MOD29#0010101200001450000000000 1
MOD29#0010101200001500000000000.met 0
MOD29#0010101200001500000000000 1
AM1ATTF#00101012000000000000.FDD 1
AM1ATTF#00101012000020000000.FDD 1
AM1ATTF#00112311999220000000.FDD 1
AM1ATTF#00101012000000000000.FDD 1
AM1ATTF#00101012000020000000.FDD 1
AM1ATTF#00112311999220000000.FDD 1
AM1ATTNF#001010120000000000000000 1
AM1ATTNF#001010120000000000000000.met 0
AM1ATTHF#001010120000000000000000 1
AM1ATTHF#001010120000000000000000.met 0
68 Invalid entries in DpPrFile

------ DpPrDiskAllocation ------
fileName partitionId
-- -----------
GDAS_0ZF#001O10000000 1
GDAS_0ZF#001O10010000 1
AST_ANC#001L10010000 1
AIRXADCM#001O20000000 1
AST_ANC#001O980010000 1
AST_04#001072119932312120000000 1
AST_ANC#001L10000000 1
AST_ANC#001O970040000 1
AST_ANC#001O300030000 1
AST_ANC#001L20020000 1
AST_ANC#001L10010000 1
AST_ANC#001O980050000 1
AST_04#001072119932312120010000 1
AST_ANC#001O970080000 1

AST_ANC#001O300070000	**1**
AST_ANC#001L20060000	**1**
AST_ANC#001O980090000	**1**
AST_ANC#001L2700000000	**1**
AST_ANC#001L2710000000	**1**
AST_ANC#001L2720000000	**1**
AST_ANC#001L2730000000	**1**
AST_ANC#001L2740000000	**1**
AST_ANC#001L2750000000	**1**
AST_ANC#001L2760000000	**1**
AST_ANC#001L2770000000	**1**
AST_ANC#001L2780000000	**1**
AST_06VD#00107211993231212000000 0	**1**
AST_L1B#0010721199323121200000000	**1**
AST_06SD#0010721199323121200000000	**1**
AST_06TD#0010721199323121200000000	**1**
AST_L1B#0010721199323121200000000	**1**
AST_L1B#0010721199323121200000000	**1**
AST_ANC#001L2200100000	**1**
AST_05#0010721199323121200000000	**1**
AST_08#0010721199323121200000000	**1**
AST_09T#0010721199323121200300000	**1**
AST_05#0010721199323121200100000	**1**
AST_08#0010721199323121200100000	**1**
AST_09T#0010721199323121200400000	**1**
AST_05#0010721199323121200200000	**1**
AST_08#0010721199323121200200000	**1**
AST_09T#0010721199323121200500000	**1**
AST_ANC#001L2800000000	**1**
AST_ANC#001L2810000000	**1**
AST_ANC#001L2820000000	**1**
AIRHBRAD#0011105199608302500000000	**1**
AIRHBRAD#0011105199608302500000000	**1**
AIRHBRAD#0011105199608302500000000	**1**
AIRABRAD#0011105199608302500000000	**1**
AIRIBRAD#0011105199608302400000000	**1**
AIRVBRAD#0011105199608302500000000	**1**
AST_L1B#0010721199323121200000000	**1**
GDAS0ZFH#0010721199318000000000000	**1**
AIRABRAD#0011105199608302500000000	**1**
AIRIBRAD#0011105199608302400000000	**1**

AIRVBRAD#00111051996083025000000	**1**
AST_L1B#00107211993231212000000	**1**
AST_ANC#001L10000000	**1**
OZ_DLYH#001072119931200000000000	**1**
AST_ANC#001L279000.00R00	**1**
AST_ANC#001L200000.00R00	**1**
AST_ANC#001L201000.00R00	**1**
AST_ANC#001L202000.00R00	**1**
AST_ANC#001L203000.00R00	**1**
AST_ANC#001L204000.00R00	**1**
AST_ANC#001L205000.00R00	**1**
AST_ANC#001L206000.00R00	**1**
AST_ANC#001L207000.00R00	**1**
AST_ANC#001L208000.00R00	**1**
AST_ANC#001L209000.00R00	**1**
AST_ANC#001L210000.00R00	**1**
AST_ANC#001L211000.00R00	**1**
AST_ANC#001L212000.00R00	**1**
AST_ANC#001L213000.00R00	**1**
AST_ANC#001L214000.00R00	**1**
AST_ANC#001L215000.00R00	**1**
AST_ANC#001L216000.00R00	**1**
AST_ANC#001L217000.00R00	**1**
AST_ANC#001L218000.00R00	**1**
AST_ANC#001L219000.00R00	**1**
AST_ANC#001L220000.00R00	**1**
AST_ANC#001L221000.00R00	**1**
AST_ANC#001L2220000000	**1**
AST_ANC#001L223000.00R00	**1**
AST_ANC#001L224000.00R00	**1**
AST_ANC#001L225000.00R00	**1**
AST_ANC#001L226000.00R00	**1**
AST_ANC#001L227000.00R00	**1**
AST_ANC#001L228000.00R00	**1**
AST_ANC#001L229000.00R00	**1**
AST_ANC#001L230000.00R00	**1**
AST_ANC#001L231000.00R00	**1**
AST_ANC#001L232000.00R00	**1**
AST_ANC#001L233000.00R00	**1**
AST_ANC#001L234000.00R00	**1**
AST_ANC#001L235000.00R00	**1**

AST_ANC#001L236000.00R00	**1**
AST_ANC#001L237000.00R00	**1**
AST_ANC#001L238000.00R00	**1**
AST_ANC#001L239000.00R00	**1**
AST_ANC#001L240000.00R00	**1**
AST_ANC#001L241000.00R00	**1**
AST_ANC#001L242000.00R00	**1**
AST_ANC#001L243000.00R00	**1**
AST_ANC#001L244000.00R00	**1**
AST_ANC#001L245000.00R00	**1**
AST_ANC#001L246000.00R00	**1**
AST_ANC#001L247000.00R00	**1**
AST_ANC#001L248000.00R00	**1**
AST_ANC#001L249000.00R00	**1**
AST_ANC#001L250000.00R00	**1**
AST_ANC#001L251000.00R00	**1**
AST_ANC#001L252000.00R00	**1**
AST_ANC#001L253000.00R00	**1**
AST_ANC#001L254000.00R00	**1**
AST_ANC#001L255000.00R00	**1**
AST_ANC#001L256000.00R00	**1**
AST_ANC#001L257000.00R00	**1**
AST_ANC#001L258000.00R00	**1**
AST_ANC#001L2590000000	**1**
AST_ANC#001L2600000000	**1**
AST_ANC#001L2610000000	**1**
AST_ANC#001L2620000000	**1**
AST_ANC#001L2630000000	**1**
AST_ANC#001L2640000000	**1**
AST_ANC#001L2650000000	**1**
AST_ANC#001L2660000000	**1**
AST_ANC#001L2670000000	**1**
AST_ANC#001L2680000000	**1**
AST_ANC#001L2690000000	**1**
AM1EPHN0#00107011998000000000000	**1**
AM1ATTNF#00107011998000000000000	**1**
ActSched#0010701199816401400000000	**1**
AM1ATTF#001123119992200000000.FDD	**1**
AM1ATTF#0010101200000000000000.FDD	**1**
AM1EPHN0#00101012000000000010000	**1**
MOD29#0010101200000050000000000	**1**

MOD29#001010120000010000000000 1
MOD29#001010120000015000000000 1
MOD29#001010120000020000000000 1
MOD29#001010120000025000000000 1
MOD29#001010120000030000000000 1
MOD29#001010120000035000000000 1
MOD29#001010120000040000000000 1
MOD29#001010120000045000000000 1
MOD29#001010120000050000000000 1
MOD29#001010120000055000000000 1
MOD29#001010120000100000000000 1
MOD29#001010120000105000000000 1
MOD29#001010120000145000000000 1
MOD29#001010120000150000000000 1
AM1EPHN0#001070119980000000000000 1
AM1ATTNF#001070119980000000000000 1
ActSched#001070119981640140000000 1
MIB2GEOP#001070119981640140000000 1
MOD01SS#001081419970050000000000 1
MOD01#00108141997005000000.hdf 1
AM1ATTN0#001010120000000000010000 1
MOD03#001010120000000000000000 1
MOD03#001010120000005000000000 1
MOD03#001010120000010000000000 1
MOD03#001010120000015000000000 1
MOD03#001010120000020000000000 1
MOD03#001010120000025000000000 1
MOD03#001010120000030000000000 1
MOD03#001010120000035000000000 1
MOD03#001010120000040000000000 1
MOD03#001010120000045000000000 1
MOD03#001010120000050000000000 1
MOD03#001010120000055000000000 1
MOD03#001010120000100000000000 1
MOD03#001010120000105000000000 1
MOD03#001010120000110000000000 1
MOD03#001010120000115000000000 1
MOD03#001010120000120000000000 1
MOD03#001010120000125000000000 1
MOD03#001010120000130000000000 1
MOD03#001010120000135000000000 1

MOD03#001010120000140000000000 1
MOD03#001010120000145000000000 1
MOD03#001010120000150000000000 1
MOD03#001010120000155000000000 1
MOD021KM#001010120000115000000000 1
MOD021KM#001010120000120000000000 1
MOD021KM#001010120000125000000000 1
MOD021KM#001010120000130000000000 1
MOD021KM#001010120000135000000000 1
MOD29#001010120000115000000000 1
MOD29#001010120000120000000000 1
MOD29#001010120000125000000000 1
MOD29#001010120000130000000000 1
AM1ATTF#00101012000000000000.FDD 1
AM1ATTF#00101012000020000000.FDD 1
AM1ATTF#00112311999220000000.FDD 1
AM1ATTF#00101012000000000000.FDD 1
AM1ATTF#00101012000020000000.FDD 1
AM1ATTF#00112311999220000000.FDD 1
AM1ATTF#00101012000020000000.FDD 1
AST_L1B#00107041997130123000.hdf 1
199 Invalid entries in DpPrDiskAllocation

------ DpPrGranuleLocation ------
granuleId
--
AIRXADCM#001O2000
1 Invalid entries in DpPrGranuleLocation

------ PlDataGranuleShort ------
granuleId
--
ACT#syn1#004130123OPS::PH#001
AIRXADCM#001O2001
AIRXAGEO#001O6000
AIRXARYL#001O5001
AIRXATCM#001O3001
AIRXATCS#001O4001
AST_ANC#001L1000
AST_ANC#001L1002
AST_ANC#001L1004
AST_ANC#001L200000

AST_ANC#001L201000
AST_ANC#001L202000
AST_ANC#001L203000
AST_ANC#001L204000
AST_ANC#001L205000
AST_ANC#001L206000
AST_ANC#001L207000
AST_ANC#001L208000
AST_ANC#001L209000
AST_ANC#001L210000
AST_ANC#001L211000
AST_ANC#001L212000
AST_ANC#001L213000
AST_ANC#001L214000
AST_ANC#001L215000
AST_ANC#001L216000
AST_ANC#001L217000
AST_ANC#001L218000
AST_ANC#001L219000
AST_ANC#001L220000
AST_ANC#001L221000
AST_ANC#001L222000
AST_ANC#001L223000
AST_ANC#001L224000
AST_ANC#001L225000
AST_ANC#001L226000
AST_ANC#001L227000
AST_ANC#001L228000
AST_ANC#001L229000
AST_ANC#001L230000
AST_ANC#001L231000
AST_ANC#001L232000
AST_ANC#001L233000
AST_ANC#001L234000
AST_ANC#001L235000
AST_ANC#001L236000
AST_ANC#001L237000
AST_ANC#001L238000
AST_ANC#001L239000
AST_ANC#001L240000
AST_ANC#001L241000

AST_ANC#001L242000
AST_ANC#001L243000
AST_ANC#001L244000
AST_ANC#001L245000
AST_ANC#001L246000
AST_ANC#001L247000
AST_ANC#001L248000
AST_ANC#001L249000
AST_ANC#001L250000
AST_ANC#001L251000
AST_ANC#001L252000
AST_ANC#001L253000
AST_ANC#001L254000
AST_ANC#001L255000
AST_ANC#001L256000
AST_ANC#001L257000
AST_ANC#001L258000
AST_ANC#001L259000
AST_ANC#001L260000
AST_ANC#001L261000
AST_ANC#001L262000
AST_ANC#001L263000
AST_ANC#001L264000
AST_ANC#001L265000
AST_ANC#001L266000
AST_ANC#001L267000
AST_ANC#001L268000
AST_ANC#001L279000
MIANCAGP#001L0001002
MICNFG#001C7001004
MICNFG#001C7001006
MICNFG#001C7101003
MICNFG#001C7101005
MISANCGM#001C0001000
PGE71#2.1.14003000OPS::PH#001
86 Invalid entries in PlDataGranuleShort

4 Observe the results as the script runs.

- A UNIX command line prompt is displayed when the script has finished.

Saving and/or Resetting the PDPS Database

The utilities directory (/usr/ecs/*MODE*/CUSTOM/utilities) contains the following three scripts that are used in resetting the database:

- EcPlDbReset.
- EcPlDbList.
- EcPlDbSave.

Saving the database using the EcPlDbSave script produces one ASCII file (with a ".dat" extension) for each database table and writes each file to a specified directory. The saved data can be restored to the database by running the EcPlDbReset script with the name of the saved database as an argument.

Resetting the database involves clearing ("wiping out") the data in the database tables and loading values from a specified "saved database" file. It is important to take into consideration the consequences of resetting the database before performing the procedure. Resetting the database removes and replaces **all** Resource Definitions, Resource Reservations (Ground Events), Production Requests, Data Processing Requests, and Production Plans. There should be coordination with all affected parties, including the Resource Planner, Production Planner, and Production Monitors concerning the effects of resetting the database as well as its after-effects (e.g., recreating resource definitions, resource reservations, and production requests).

Whenever the PDPS database is reset (not including database cleanup by running the EcPlDbClean script) it is also necessary to remove all PLS subscriptions in the Communications Subsystem (CSS) Subscription Server database (where the subscriber is Subscription Manager). Production personnel can remove the subscriptions using the Subscription Server GUI (EcSbSubServerGUI) if they have access to the GUI. Otherwise, they can request User Services personnel to remove the subscriptions.

As a result of removing the subscriptions, no subscription notification will come through for existing jobs in the newly loaded database. Only new jobs generated using the Production Request Editor will work normally with regard to subscriptions.

The procedure for saving and resetting the PDPS database starts with the assumption that the applicable servers are running and the Production Planner has logged in to the system.

Saving and/or Resetting the PDPS Database

NOTE: It is important to log in as a user who has "write" permission in the saved_dumps directory (/usr/ecs/*MODE*/CUSTOM/utilities/saved_db/saved_dumps); otherwise it will not be possible to save database contents.

1 Access a terminal window logged in to the Planning/Management Workstation host.

- Examples of Planning/Management Workstation host names include **e0pls03** and **l0pls02.**
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 Type **setenv ECS_HOME /usr/ecs/** then press the **Return/Enter** key.

- When logging in as a system user (e.g., cmshared), the ECS_HOME variable may be set automatically so it may not be necessary to perform this step.

3 Type **cd /usr/ecs/*MODE*/CUSTOM/utilities** then press **Return/Enter**.

- Change directory to the directory containing the production planning startup scripts (e.g., EcPlPRE_IFStart).
- The ***MODE*** will most likely be one of the following operating modes:
  - OPS (for normal operation).
  - TS1 (for Science Software Integration and Test (SSI&T)).
  - TS2 (new version checkout).
- Note that the separate subdirectories under /usr/ecs apply to (describe) different operating modes.

4 If saving the current database is desired, type **EcPlDbSave *Saved_DB_Name*** then press **Return/Enter** to start the process of saving the database.

- ***Saved_DB_Name*** is the desired name for the directory into which the data files will be written.
- If there is not already a directory with the specified ***Saved_DB_Name***, a **"Making directory..."** message is displayed.
- If there is already a directory with the specified ***Saved_DB_Name***, the following prompt is displayed:

  Are you sure you wish you overwrite previously saved version of *Saved_DB_Name*? (Y/N)

5 If the prompt to **Are you sure you wish you overwrite previously saved version of *Saved_DB_Name*? (Y/N)** appears, type the appropriate response (**Y** or **N**) then press the **Return/Enter** key.

- If **y** is entered, the data are saved in the specified directory, replacing the data that had previously been saved there.
- If **n** is entered, an **"Exiting without saving database..."** message is displayed.

6 If obtaining a listing of saved databases is desired, type **EcPlDbList** then press **Return/Enter** to generate a listing of the saved databases.

- A listing of saved databases is displayed.
- The listing is a useful tool to display the options for resetting the database.

7 If resetting the current database is desired, type **EcPlDbReset *Saved_DB_Name*** then press **Return/Enter** to start the database reset process.

- ***Saved_DB_Name*** refers to the name of the directory containing the data files to be loaded into the database.
- The data in the database are replaced with the data from the specified directory.
- When the database has been reset a **"Successfully reset the database to *Saved_DB_Name*"** message is displayed.

Troubleshooting Production Planning Problems

Trouble Symptoms

Troubleshooting is a process of identifying the source of problems on the basis of observed trouble symptoms. One common source of problems involves connections with other subsystems for the transmission of messages or data. Like many other operational areas in the system, Planning has interfaces with many other subsystems. Consequently, problems with processing can be traced to either the Planning Subsystem or one of many other subsystems, including (but not necessarily limited to) those in the following list:

- Data Processing Subsystem (DPS).
- Data Server Subsystem (DSS).
- Communications Subsystem (CSS).

Fault Recovery

Each request that crosses a client/server boundary is assigned a system-unique identifier referred to as an RPC ID. (RPC refers to Remote Procedure Call, the mechanism by which requests are submitted from client to server.) The RPC ID facilitates the automatic fault recovery events that occur whenever there is a client or server failure.

- As a request propagates through the system, each associated client/server exchange is assigned a unique RPC ID.
  - The RPC ID for each interaction is derived from the previous RPC ID received by the client for the request.
    - Consequently, all RPC IDs associated with a given request have a common portion that relates the various client/server calls to one another.
  - Given the previous RPC ID, clients consistently reproduce the same RPC ID that was submitted to the server on the subsequent event.
- The concept of reproducible RPC IDs is central to the system fault recovery capability.
  - When requests are retried from client to server, they are always submitted with the same RPC ID that was used in the original submission of the request, even if either the client or server has crashed between retries.
- The RPC ID is also central to the check-pointing aspect of fault recovery.

- As requests arrive at fault recovery-enabled servers, they are recorded in a persistent store (typically a database), tagged with the RPC ID, which identifies the request.
- As the request is serviced, check-pointing state information may be updated in the persistent store, up to and including the completion status of the request.
- This allows the servers to resume servicing from the last check-pointed state, particularly upon resubmission from a client.

PLANG and PRONG components checkpoint the following types of information:

- **EcPlOdMgr** - Request information.
- **EcPlSubMgr** - Unprocessed subscription notifications.
- **EcDpPrDeletion** - Interim Delete Requests.
- **EcDpPrEM** - Queued and activated jobs.

Fault Handling

Failure events are classified according to the following three severity levels:

- Fatal error.
  - Returned when a request cannot be serviced, even with operator intervention.
  - For example, if a request is made to distribute data via ftp to a non-existent host, the request is failed with a fatal error.
- Retry error.
  - Potentially recoverable error.
  - Normally, a retry error would be returned to the client only when the server cannot recover from the error automatically.
  - A retry error may require operator assistance during recovery.
    - For example, when using the Production Request Editor GUI, the Production Planner would enter a new name for a production request after being notified that a previously entered name contained too many characters.
- Warning.
  - Provided when operations can proceed without interruption, but an unexpected circumstance was detected.
  - For example, if a client requests a file to be removed but the file does not exist, there is no error per se; however, a warning is generated to caution the client that the file to be removed did not exist in the first place.

Transient errors (such as network errors) are always retry errors.

- In general, clients and servers that experience transient retry errors first attempt to recover by retrying the operation automatically.
- One special case of this is “rebinding,” which refers to the process by which a client automatically attempts to re-establish communication with a server in the event communication is disrupted.
  - The disruption may be caused by transient network failure, or by the server crashing or being brought down.
  - In any case, the client automatically attempts to reconnect to the server for a configurable period of time on a client-by-client basis.

System processes encountering an error or receiving an error from a server request can either pass the error back to a higher-level client or present it to the operator for operator intervention. The specific fault handling policies for PLANG and PRONG client processes are shown in Table 1.

Table 1. PLANG and PRONG Fault Handling Policies

Client Process	Fault Handling Policy
EcPlSubMgr	**Retry errors:** All Subscription processing errors are retried a configurable number of times and for a configurable time period. After the configurable number of times (or time period) the subscription is lost. **Fatal errors:** N/A.
EcPlPREditor_IF **EcPlWb**	**Retry errors:** Since these are GUI applications, errors are reported to the operator and it is the operator’s responsibility to retry the request. **Fatal errors:** Errors are reported to the operator.
EcPlOdMgr	**Retry errors:** Retries errors from the Science Data Server and the Subscription Server. **Fatal errors:** Logs errors and stops current on demand requests.
EcDpPrEM	**Retry errors:** Errors are retried a configurable number of times, then the job is failed and it is up to the Production Monitor to restart the job through AutoSys. **Fatal errors:** A fatal error message is logged.
EcDpPrJobMgmt	**Retry errors:** If a DPR cannot be assigned to a machine or created in AutoSys, it is left in a PENDING state and the assignment is retried after DpPrPendingThreadWaitInterval seconds. **Fatal errors:** N/A.
EcDpPrDeletion	**Retry errors:** No retries are implemented. Status from DSS is not checked. **Fatal errors:** N/A.

Client Crash and Restart

When a client of a PLANG or PRONG server crashes, the server (i.e., EcPlSubMgr, EcDpPrJobMgmt, or EcDpPrDeletion) continues to service the requests that were in process at the time of the client's crash.

When a client restarts in the system, it sends a restart notification to each server with which it interacts.

- Clients notify servers that they have come up either "cold" or "warm."
- Generally, the notification temperature sent to the server matches the temperature at which the client process is restarted.

The default server behavior in response to startup notification from a client is as follows:

- Warm Notification.
  - Outstanding requests for the restarted clients remain available in the persistent store.
  - The outstanding requests may be resubmitted by the client, and are serviced to completion upon resubmission.
  - Associated resources are left allocated until the requests are completed.
- Cold Notification.
  - All outstanding requests for the restarted client are cancelled.
  - If the client resubmits any cancelled request using the same RPC ID (e.g., by pressing the Retry button from an operator GUI), it is failed with a fatal error due to the client cold startup notification.
  - Any resources associated with the cancelled requests are released and reclaimed by the system.

Server Crash and Restart

When a server crashes, clients cannot continue to submit requests for processing.

- Synchronous requests in progress result in a Distributed Computing Environment (DCE) exception being thrown back to the client process, which enters a rebinding failure recovery mode (as previously mentioned).
- Attempts to submit requests while the server is down result in the client blocking until a communication timeout has been reached.
- Although DCE has been replaced by socket-based library calls (i.e., CCS Middleware), the DCE exception code is handled by the CCS Middleware.

When a server restarts, it may perform various resynchronization activities in order to recover from an unexpected termination.

- In the event of a server cold start or cold restart, the server typically cancels all outstanding requests and reclaims all associated resources.
- In general, existing request queues are retained for warm restarts and cleared for cold starts or cold restarts.
- **EcPlOdMgr**-specific activities upon start/restart:
  - **Warm Restart:**
    - Any requests that have not been processed are read from the database and processed.
  - **Cold Start or Cold Restart:**
    - N/A.
- **EcPlSubMgr**-specific activities upon start/restart:
  - **Warm Restart:**
    - Any subscriptions that have not been processed are read from checkpoint file and processed.
  - **Cold Start or Cold Restart:**
    - N/A.
- **EcDpPrDeletion**-specific activities upon start/restart:
  - **Warm Restart:**
    - Interim granules marked for deletion are read from the database and are deleted when time-out occurs.
  - **Cold Start or Cold Restart:**
    - N/A.
- **EcDpPrJobMgmt**-specific activities upon start/restart:
  - **Warm Restart:**
    - Jobs in AutoSys and jobs waiting in the queue are read from the database.
    - Any jobs that are ready are placed into AutoSys from the queue (if there are processing slots available).
  - **Cold Start or Cold Restart:**
    - N/A.

Request Resubmission

Upon restarting a crashed client or server, requests are typically resubmitted. If the restarted process was started warm, the fault-recovery capabilities permit the server to resume processing of the request from its last check-pointed state. This prevents needless repetition of potentially time-consuming activities.

- **EcDpPrJobMgmt**- and **EcDpPrDeletion**-specific activities upon resubmission of a request:
  - Requests are submitted synchronously.
  - If the entire request is resubmitted by a client, then only that part of the resubmitted request that has not been completed is reprocessed.

Troubleshooting a Production Planning Failure

Table 2 describes actions to be taken in response to some common Planning problems. If the problem cannot be identified and fixed without help within a reasonable period of time, the appropriate response is to call the help desk and submit a trouble ticket in accordance with site Problem Management policy.

Table 2. Troubleshooting Production Planning Problems (1 of 3)

Symptom	Response
Unable to log in to the Planning Subsystem host (e.g., e0pls03).	Check with the Operations Controller/System Administrator to ensure that the host is "up."
GUI not displayed when the start-up script has been properly invoked.	Ensure that the DISPLAY variable was set properly. [For detailed instructions refer to the applicable procedure, either **Launching the Production Request Editor** or **Launching Planning Workbench-Related GUIs** (previous sections of this lesson).]
Error message indicating that SNS (System Name Server) and/or Resource Model is/are in use using the selected Application ID.	1. Use another Application ID if working in a different mode from the person using the selected Application ID. 2. If working in the same mode as the other user, coordinate use of Planning applications with the other user and/or the System Administrator. [For detailed instructions refer to the procedure for **Launching Planning Workbench-Related GUIs** (previous section of this lesson).]
Error message associated with the Production Request Editor.	Refer to Table 3, Production Request Editor User Messages (adapted from the corresponding table in 609-EMD-001, *Release 7.11 Operations Tools Manual for the EMD Project*).
Error message associated with the Production Strategies GUI.	Refer to Table 4, Production Strategy User Messages (adapted from the corresponding table in 609-EMD-001, *Release 7.11 Operations Tools Manual for the EMD Project*).

Table 2. Troubleshooting Production Planning Problems (2 of 3)

Symptom	Response
Error message associated with the Planning Workbench.	Refer to Table 5, Planning Workbench User Messages (adapted from the corresponding table in 609-EMD-001, *Release 7.11 Operations Tools Manual for the EMD Project*).
Production Request fails (DPR generation fails).	1. Ensure that it is possible to connect to the necessary hosts and servers (listed in Table 6). [For detailed instructions refer to the section on **Checking Connections to Hosts/Servers** (subsequent section of this lesson).] 2. If hosts/servers are all “up,” perform the procedure for **Handling a Failure to Generate DPRs** (subsequent section of this lesson). 3. Retry generating DPRs by resaving the Production Request. [For detailed instructions refer to the section on **Editing/Modifying a Production Request** (previous section of this lesson).]
PR or DPR deletion hangs.	1. Ensure that enough time has passed to allow DPR deletion (deleting a DPR can require as much time as creating a DPR). 2. Ensure that it is possible to connect to the necessary hosts and servers (listed in Table 6). (Both the Job Management Server and Deletion Server are called to clean up all PDPS database tables associated with the DPR or PR.) [For detailed instructions refer to the section on **Checking Connections to Hosts/Servers** (subsequent section of this lesson).] 3. If hosts/servers are all “up,” check for a database lock or resource lock in the PDPS database. [For detailed instructions refer to the section on **Responding to PR or DPR Deletion that Hangs** (subsequent section of this lesson).]
DPR deletion fails.	1. Ensure that enough time has passed to allow DPR deletion (deleting a DPR can require as much time as creating a DPR). 2. Ensure that it is possible to connect to the necessary hosts and servers (listed in Table 6). (Both the Job Management Server and Deletion Server are called to clean up all PDPS database tables associated with the DPR or PR.) [For detailed instructions refer to the section on **Checking Connections to Hosts/Servers** (subsequent section of this lesson).] 3. If hosts/servers are all “up,” check the Deletion Server Debug log (EcDpPrDeletionDebug.log). [For detailed instructions refer to the section on **Responding to DPR Deletion that Fails** (subsequent section of this lesson).]

Table 2. Troubleshooting Production Planning Problems (3 of 3)

Symptom	Response
DPR scheduling fails (DPR is not passed to Data Processing).	1. Ensure that it is possible to connect to the necessary hosts and servers (listed in Table 6). [For detailed instructions refer to the section on **Checking Connections to Hosts/Servers** (subsequent section of this lesson).] 2. If hosts/servers are all “up,” perform the procedure for **Handling a DPR Scheduling Failure** (subsequent section of this lesson). 3. If necessary, delete the affected DPRs. [For detailed instructions refer to the section on **Creating a New Production Plan** (previous section of this lesson).] 4. If affected DPRs were deleted, recreate the DPRs. [For detailed instructions refer to the section on **Editing/Modifying a Production Request** (previous section of this lesson).] 5. If affected DPRs were recreated, create a new production plan. [For detailed instructions refer to the section on **Creating a New Production Plan** (previous section of this lesson).]
Other problems.	1. Ensure that it is possible to connect to the necessary hosts and servers (listed in Table 6). [For detailed instructions refer to the section on **Checking Connections to Hosts/Servers** (subsequent section of this lesson).] 2. If hosts/servers are all “up,” check the log files (e.g., EcPlPREditor.ALOG, EcPlPREditorDebug.log, EcPlWb.ALOG, EcPlWbDebug.log, and EcPlTl.ALOG) in the /usr/ecs/*MODE*/CUSTOM/logs directory for error messages. [For detailed instructions refer to the procedure for **Checking Log Files** (subsequent section of this lesson).]

Table 3. Production Request Editor User Messages (1 of 6)

Message Text	Impact	Cause and Corrective Action
Data Processing Request not open, Data Processing Request to be open must be selected.	Unable to open a DPR.	1. Select (highlight) a DPR from the DPR List. 2. Select **File** → **Open** from the pull-down menu to open the DPR. [For detailed instructions refer to the procedure for **Reviewing Data Processing Requests** (previous section of this lesson).]
Do you want to delete this DPR <DPR ID>?	Requires operator input (confirmation of deletion) before proceeding.	1. Select **Yes** to delete the DPR. 2. Select **No** to keep the DPR.
Do you want to delete this PR <PR NAME>?	Requires operator input (confirmation of deletion) before proceeding.	1. Select **Yes** to delete the PR. 2. Select **No** to keep the PR.
DPR Delete Failed.	Selected DPR was not deleted from the PDPS database.	Check the log files (e.g., EcPlPREditor.ALOG and EcPlPREditorDebug.log) in the /usr/ecs/*MODE*/CUSTOM/logs directory for error messages. [For detailed instructions refer to the procedure for **Checking Log Files** (subsequent section of this lesson).]
Dpr Generation Incomplete for PR <PRNAME>. Do you want to complete Dpr explosion?	Requires operator input before proceeding.	1. Select **Yes** to complete the DPR generation. 2. Select **No** to display another message that will show how many DPRs have been generated so far.
Environment variable PL_NEW not set.	**Production Request Editor** GUI is not started	1. Submit a request to the Database Administrator to check the value assigned to **PL_NEW** (associated with EcPlPREditor) in the Configuration Registry. 2. If no value is specified for **PL_NEW**, submit a request to the Database Administrator to add the value **PL_NEW = New** for the EcPlPREditor in the Configuration Registry.
Invalid Time is entered or End time is less than Begin Time.	Unable to create Data Processing Requests because the collection/insertion time(s) entered are invalid.	Ensure that the **Begin** and **End** dates/times entered on the **PR Edit** screen are valid.
Must select a PGE for Production Request before saving.	Unable to proceed with Data Processing Request creation until a PGE is entered on the **PR Edit** screen.	1. Open the **PGE** screen by clicking the mouse on the **PGE** button of the **PR Edit** screen. 2. Select (highlight) a PGE from the PGE list. 3. Click on the **OK** button.

Table 3. Production Request Editor User Messages (2 of 6)

Message Text	Impact	Cause and Corrective Action
<number> Dpr(s) have previously been generated. Are you sure you want to delete this PR?	Requires operator input before proceeding.	1. Select **Yes** to delete the PR. 2. Select **No** to clear the message.
Please select a Dpr you want to delete from the DPR List.	DPR will not be deleted from the PDPS database until a DPR is selected from DPR List.	1. Select (highlight) a DPR from the DPR List. 2. Delete the DPR using either **Edit** → **Delete** from the pull-down menu or the keyboard **Ctrl-D**. [For detailed instructions refer to the procedure for **Deleting a Data Processing Request** (previous section of this lesson).]
Please select a PR you want to delete from the PR List.	Cannot delete a Production Request until it is selected from the PR List.	1. Select a PR from the PR List. 2. Delete the PR using either **Edit** → **Delete** from the pull-down menu or the keyboard **Ctrl-D**. [For detailed instructions refer to the procedure for **Deleting a Production Request** (previous section of this lesson).]
Please specify production request to filter by.	Unable to filter Production Request IDs.	**Filter** button acts on the pattern specified in the filter text entry area. Here, the user can filter all the production requests by the specified pattern. 1. Select the appropriate PR type on the **PR Type** option button. 2. Select the appropriate PR in the **PR Name** field. 3. Click on the **Filter** button. [For detailed instructions refer to the procedure for **Reviewing Data Processing Requests** (previous section of this lesson).]
PR Failed - Need to include 1st orbit in the Orbit model.	Data Processing Request creation fails because 1st orbit information is incorrect.	1. Enter isql commands for checking the PDPS database PlPgeOrbitModel table to determine whether the orbit information for the first orbit is in the table. [For detailed instructions refer to the procedure for **Using ISQL to Check Database Tables** (subsequent section of this lesson).] 2. If the orbit information for the first orbit is not in the PlPgeOrbitModel table, notify the SSI&T team.
PR Failed - Read failure from PlOrbitModel.	Data Processing Request creation fails because orbit information cannot be read from the PDPS database.	Unable to read the PlPgeOrbitModel table from the PDPS database. 1. Check the database connections. [For detailed instructions refer to the procedure for **Checking Database Connections** (subsequent section of this lesson).] 2. If the problem recurs, call the help desk and submit a trouble ticket in accordance with site Problem Management policy.

Table 3. Production Request Editor User Messages (3 of 6)

Message Text	Impact	Cause and Corrective Action
Production Request <PR NAME> Deleted; <number> DPR(s) deleted from the database.	Production Request(s) and dependent DPR(s) are successfully deleted from the PDPS database.	For information only. The message indicates how many DPRs associated with the PR were deleted.
Production Request Delete Failed.	Selected Production Request(s) is (are) not deleted.	Unable to delete a PR from the database. Check the log files (e.g., EcPlPREditor.ALOG and EcPlPREditorDebug.log) in the /usr/ecs/*MODE*/CUSTOM/logs directory for error messages. [For detailed instructions refer to the procedure for **Checking Log Files** (subsequent section of this lesson).]
Production Request Explosion into DPR(s) Failed.	Data Processing Request creation failed.	Check the log files (e.g., EcPlPREditor.ALOG and EcPlPREditorDebug.log) in the /usr/ecs/*MODE*/CUSTOM/logs directory for error messages. [For detailed instructions refer to the procedure for **Checking Log Files** (subsequent section of this lesson).]
Production Request Explosion into DPRs Failed, zero DPRs Generated.	Data Processing Request creation failed.	Check the log files (e.g., EcPlPREditor.ALOG and EcPlPREditorDebug.log) in the /usr/ecs/*MODE*/CUSTOM/logs directory for error messages. [For detailed instructions refer to the procedure for **Checking Log Files** (subsequent section of this lesson).]
Production Request Explosion into DPRs ok. <number> DPR(s) Generated.	Data Processing Requests successfully created and stored in the PDPS database.	For information only. The message indicates how many DPR(s) were generated during the PR Explosion.
Production Request Explosion into DPR(s) ok. <number> more DPR(s) Generated.	Data Processing Requests successfully created and stored in the PDPS database.	For information only. The message indicates how many more DPRs were generated for the PR.
Production Request not open, Production Request to be open must be selected.	Production Request is not opened on the **PR Edit** screen until a PR is selected from the PR List.	1. Select a PR from the PR List. 2. Open the PR using either **File** → **Open** from the pull-down menu or the keyboard **Ctrl-O**. [For detailed instructions refer to the procedure for **Editing/Modifying a Production Request** (previous section of this lesson).]

Table 3. Production Request Editor User Messages (4 of 6)

Message Text	Impact	Cause and Corrective Action
Production Request not saved, Production Request already exists.	Data Processing Request creation failed because the Production Request name already exists in the PDPS database.	1. Enter a new name for the production request. 2. Save the PR. [For detailed instructions refer to the procedure for **Editing/Modifying a Production Request** (previous section of this lesson).]
Production Request not saved, Production Request must have a name before it is saved.	Unable to proceed with Data Processing Request creation because no name has been entered for the Production Request.	1. Enter a new name for the production request. 2. Save the PR. [For detailed instructions refer to the procedure for **Creating a New Production Request Using the Production Request Editor GUI** (previous section of this lesson).]
Production Request not saved, Production Request must have a name shorter than 20 characters before it is saved	Unable to proceed with Data Processing Request creation because the name chosen for the Production Request exceeds the maximum number of characters allowed.	1. Enter a PR name with fewer than 20 characters. 2. Save the PR. [For detailed instructions refer to the procedure for **Editing/Modifying a Production Request** (previous section of this lesson).]
Production Request not saved, save Production Request first.	Unable to open a PR.	1. Save the modified PR. 2. Open the new PR. [For detailed instructions refer to the procedure for **Editing/Modifying a Production Request** (previous section of this lesson).]
SECURITY VIOLATION: no write permission.	Unable to proceed with Data Processing Request creation because Production Request Editor does not have proper "write" permission on the PDPS database (database cannot be updated).	Operator does not have permission to save a production request. 1. Exit from the Production Request Editor (**File** → **Exit** from the pull-down menu). 2. Log in as a user with "write" permission. [Contact the System Administrator for assistance if necessary.] 3. Launch the Production Request Editor. [For detailed instructions refer to the procedure for **Launching the Production Request Editor** (previous section of this lesson).] 4. Create the Production Request. [For detailed instructions refer to the procedure for **Creating a New Production Request Using the Production Request Editor GUI** (previous section of this lesson).]

Table 3. Production Request Editor User Messages (5 of 6)

Message Text	Impact	Cause and Corrective Action
Unable to Initialize PlDpr Pool.	**Production Request Editor** GUI is not started (database cannot be accessed).	Unable to read the DPR table from the database. 1. Check the database connections. [For detailed instructions refer to the procedure for **Checking Database Connections** (subsequent section of this lesson).] 2. If the problem recurs, call the help desk and submit a trouble ticket in accordance with site Problem Management policy.
Unable to Initialize PlPge Pool.	**Production Request Editor** GUI is not started (database cannot be accessed).	Unable to read the PGE table from the database. 1. Check the database connections. [For detailed instructions refer to the procedure for **Checking Database Connections** (subsequent section of this lesson).] 2. If the problem recurs, call the help desk and submit a trouble ticket in accordance with site Problem Management policy.
Unable to Initialize PlProductionRequest Pool.	**Production Request Editor** GUI is not started (database cannot be accessed).	Unable to read the PR table from the database. 1. Check the database connections. [For detailed instructions refer to the procedure for **Checking Database Connections** (subsequent section of this lesson).] 2. If the problem recurs, call the help desk and submit a trouble ticket in accordance with site Problem Management policy.
Write to Database of Production Request Failed.	Data Processing Request creation fails because the Production Request information cannot be saved in the PDPS database.	Unable to write the data to the database. 1. Check the database connections. [For detailed instructions refer to the procedure for **Checking Database Connections** (subsequent section of this lesson).] 2. If the problem recurs, call the help desk and submit a trouble ticket in accordance with site Problem Management policy.

Table 3. Production Request Editor User Messages (6 of 6)

Message Text	Impact	Cause and Corrective Action
Zero DPR Generated. Production Request Not Saved.	Data Processing Request creation failed.	The proposed Production Request might result in the creation of a duplicate DPR. 1. Compare the characteristics of the proposed PR and existing DPRs to determine whether the Production Request would result in the creation of a duplicate DPR. [For detailed instructions refer to the procedure for **Reviewing Data Processing Requests** (previous section of this lesson).] 2. Ensure that it is possible to connect to the necessary hosts and servers (listed in Table 6). [For detailed instructions refer to the section on **Checking Connections to Hosts/Servers** (subsequent section of this lesson).] 3. If hosts/servers are all "up," perform the procedure for **Handling a Failure to Generate DPRs** (subsequent section of this lesson). 4. Retry generating DPRs by resaving the Production Request. [For detailed instructions refer to the section on **Editing/Modifying a Production Request** (previous section of this lesson).] 5. Check the log files (e.g., EcPlPREditor.ALOG and EcPlPREditorDebug.log) in the /usr/ecs/*MODE*/CUSTOM/logs directory for error messages. [For detailed instructions refer to the procedure for **Checking Log Files** (subsequent section of this lesson).]

Table 4. Production Strategy User Messages (1 of 2)

Message Text	Impact	Cause and Corrective Action
Invalid StrategyID. Please enter a StrategyID.	Unable to save the Production Strategy.	The strategy ID text field is blank. Enter a valid Strategy ID. [For detailed instructions refer to the procedure for **Defining a Production Strategy** (previous section of this lesson).]
No StrategyID selected. Please select one.	Unable to open or delete a Strategy.	No strategy is selected from the Production Strategy list. Select a Strategy ID from the list.
This StrategyID already exists.	Unable to save the Production Strategy.	The Strategy ID entered already exists. Enter a new name for the Strategy ID.

Table 4. Production Strategy User Messages (2 of 2)

Message Text	Impact	Cause and Corrective Action
Total Weight must be 100 – Normalize weights.	Unable to save the Production Strategy.	The combined weights for the different "Types" exceed 100. Click on the **Normalize** push button.
Value out of range (0-10).	Unable to save the Production Strategy.	Default value entered for the selected "Type" is out of range and is reset to this strategy's default value for this "Type". Enter a value in the range of 0 – 10.
Value out of range (0-100).	Unable to save the Production Strategy.	Weight for the selected "Type" is out of range and is reset to this strategy's default weight for this "Type". Enter a value in the range of 0 – 100.

Table 5. Planning Workbench User Messages

Message Text	Impact	Cause and Corrective Action
The following dpr jobs were failed when you activated the plan <plan name> <list of failed dprs, one per line> Do you want to recover the failed dpr jobs?	Planning Workbench detects that Job Management has not received the DPRs correctly. The operator needs to be aware that there is (are) failed job(s).	Click on **Recover**; the failed jobs will be resubmitted.
You cannot reschedule with the current active plan. Do you want to create new plan to schedule production request(s) and to reactivate the plan?	Cannot replan under the same plan name.	1. Select **Yes** to create a new plan that can be used for replanning. 2. Select **No** and use an existing plan to activate and replan.
You have already submitted the plan, <plan name> and this plan is currently active. Do you want to create a new plan?	If the operator has already submitted this plan for activation, the Planning Workbench will not allow it to be submitted twice.	1. Select **Yes** to create a new plan that can be used for replanning. 2. Select **No** and use an existing plan to activate and replan.
You need to save the current plan before you activate the plan!!!	Insures consistency between the database active plan and what actually gets activated.	Select **Yes** to save the plan. The next time activation is attempted there will be no error message.

Table 6. Hosts, Servers, Clients and Other Software Relevant to Production Planning and Processing

HOST	SERVER/CLIENT/OTHER SOFTWARE
Planning/Management Workstation (e.g., x0pls01)	Production Request Editor (EcPlPREditor_IF) Planning Workbench GUI (EcPlWb) Production Strategies GUI (EcPlProdStrat) Production Planning Master Timeline (EcPlTl) Message Handler (EcPlMsh) System Name Server (EcPlSns) Resource Model (EcPlRm) Production Request (PR) Generator (EcPlPRGenerator)
Queuing Server (e.g., x0sps04)	Job Management Server (EcDpPrJobMgmt) Deletion Server (EcDpPrDeletion) Execution Management (EcDpPrEM) AutoSys Event Processor (event_demon) AutoSys Event Server (Sybase server) (e.g., x0sps02_srvr) On-Demand Manager (EcPlOdMgr) Subscription Manager (EcPlSubMgr) PDPS database Sybase server (e.g., x0sps02_srvr)
Science Processor (e.g., x0spg11)	PGE Management (EcDpPrRunPGE) Resource Usage (EcDpPrRusage) PGEs
Access/Process Coordinators (APC) Server (e.g., x0acg01)	Archive Server (EcDsStArchiveServer) FTP Server (EcDsStFtpServer) Cache Manager Server (EcDsStCacheManagerServer) Staging Disk Server (EcDsStStagingDiskServer) Pull Monitor Server (EcDsStPullMonitorServer)
Ingest Server (e.g., x0icg01)	Name Server (EcCsIdNameServer) Registry Server (EcCsRegistry)
Sun Consolidation External Server (e.g., x0ins01)	Data Dictionary (EcDmDictServer)
Sun Consolidation Internal Server (e.g., x0acs11)	Science Data Server (EcDsScienceDataServer) Subscription Server (EcSbSubServer)

NOTE: Depending on the installation, software may be loaded on hosts other than the examples provided.

Production Planning Troubleshooting Procedures

The following procedures for correcting Production Planning problems are provided in this section:

- Checking Connections to Hosts/Servers.
- Checking the Production Request Editor ALOG File.
- Using ISQL to Check Database Tables.
- Checking the PDPS Database for Causes of Failure to Generate DPRs.
- Checking the Production Request Editor Debug File for Evidence of Metadata Queries.
- Checking for Database Deadlocks.
- Checking for Resource Locks in the PDPS Database.
- Responding to DPR Deletion that Fails.
- Responding to a "DPR Validation Failed" Error.
- Responding to an "information (INFO) Production Request {Production Request Id} has unschedulable DPR {DPR Id}" Error.
- Checking Log Files.
- Checking Database Connections.

Checking Connections to Hosts/Servers

The procedure for checking connections to hosts/servers is normally performed as part of troubleshooting problems.

The procedure for checking connections to hosts/servers starts with the assumption that the operator has logged in to the system and the Planning/Management Workstation (e.g., e0pls03 or l0pls02).

Checking Connections to Hosts/Servers

---

1 Access a terminal window logged in to the Planning/Management Workstation.

- Examples of Planning/Management Workstation host names include **e0pls03** and **l0pls02.**
- Most other system hosts are acceptable for checking connections.
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 At the command line prompt type **cd /usr/ecs/*MODE*/CUSTOM/utilities** then press **Return/Enter**.

- Change directory to the directory containing the utility scripts.

3 At the command line prompt type **EcCsIdPingServers *MODE*** then press **Return/Enter**.

- The following type of response is displayed (only a few representative lines are shown):

  /usr/ecs/TS2/CUSTOM/bin/CSS/Sweeper -nsh x0icg01 -nsp 18202 FoSwSweeper application started...

  We made a connection with EntryId =x0acs06:38709:23057 --- EcSrTransportSubServer

  We made a connection with EntryId =x0acs06:38712:23057 --- EcSrTransportSubEventServer

  We made a connection with EntryId =x0acs06:33379:17033 --- DsShQuitIDL

  We made a connection with EntryId =x0wkg01:11959:41838305 --- EcDsHdfEosServer_3_G3

  […]

4 Observe the results displayed on the screen to determine whether connections can be made with the necessary hosts and servers.

- The necessary hosts and servers are listed in Table 6. Hosts, Servers, Clients and Other Software Relevant to Production Planning and Processing.

5 If pinging the servers (Step 3) indicated a problem with any connection, ping the servers again (at the command line prompt type **EcCsIdPingServers *MODE*** then press **Return/Enter**).

6 Observe the results displayed on the screen to determine whether connections can be made with the necessary hosts and servers.

7 If it is not possible to connect to any needed host(s)/server(s), notify the Operations Controller/System Administrator to check the hosts/servers and bring them back up if necessary.

8 Return to the procedure that recommended checking connections to hosts.

Handling a Failure to Generate DPRs

There are several possible reasons for the Planning Subsystem to fail to generate DPRs. A failure to generate DPRs could be a normal consequence of one of the production rules. For example, if the Closest Granule production rule applies, there may be times when no acceptable input granules are found. However, a failure to generate DPRs is most likely due to one of the following problems:

- Error creating a subscription for input data.
- Production Request Editor does not query DSS.
- Too many granules meet the criteria for input granules for a particular DPR.

A failure due to “too many granules” is caused when too many granules meet the criteria for input granules for a particular DPR.

- At PGE registration, the number of granules expected for each input ESDT is defined.
  - The definition includes the minimum number and the maximum number of granules expected.
- If the number of granules found is not between the minimum and maximum number, the request fails.
  - The failure occurs either when trying to save the production request or when the PCF file is generated (AutoSys pre-processing step).

Whether the failure to generate DPRs is due to an error creating a subscription for input data or “too many granules,” the first step in investigating the problem is to check the Production Request Editor application log (ALOG) file.

The following procedures may be involved in investigating a failure to generate DPRs:

- Checking the Production Request Editor ALOG File.
- Using ISQL to Check Database Tables.
- Checking the PDPS Database for Causes of Failure to Generate DPRs.
- Checking for DPR Explosion Failure Because the Production Request Editor Does Not Query DSS for Data.
- Checking the Production Request Editor Debug File for Evidence of Metadata Queries.

Checking the Production Request Editor ALOG File

The procedure for checking the Production Request Editor ALOG file starts with the assumption that the operator has logged in to the system and the Planning/Management Workstation (e.g., e0pls03 or l0pls02).

Checking the Production Request Editor ALOG File

1 Access a terminal window logged in to the Planning/Management Workstation host.

- Examples of Planning/Management Workstation host names include **e0pls03** and **l0pls02.**
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 Type **cd /usr/ecs/*MODE*/CUSTOM/logs** then press **Return/Enter**.

- Change directory to the directory containing the planning log files (e.g., EcPlPREditor.ALOG).

3 Type **pg *filename*** then press **Return/Enter**.

- ***filename*** refers to the log file to be reviewed (e.g., EcPlPREditor.ALOG).
- The first page of the log file is displayed.
- Although this procedure has been written for the **pg** command, any UNIX editor or visualizing command (e.g., **vi**, **view**, **more**) can be used to review the log file.

4 Review the log file to determine whether the log contains an error message that indicates why the DPR generation failed.

- For example:

  Msg: PlPge::GetInputForDpr - Extr input to process DPR MoPGE01#2007081600OPS, for data type id MOD000#001, with logical id 599001. PlDataTypeReq has a scienceGroup of for this datatype. Expected 2 max inputs, but got 3. Priority : 2 Time : 07/09/99 17:10:52

  – In the example the Production Request Editor queried the PDPS database for granules that would satisfy the data needs for the DPR and found three granules instead of the two it expected (i.e., “Expected 2 max inputs, but got 3”).

  – Consequently, the Production Request failed due to “too many granules.”

5 If there is an error in the log file indicating that too many granules meet the criteria for input granules for a particular DPR, go to the procedure for **Checking the PDPS Database for Causes of Failure to Generate DPRs** (subsequent section of this lesson).

6 If there is an error in the log file indicating that an entry could not be found for a specific ESDT in the appropriate database, notify the Science Data Specialist or whoever else installs ESDTs to verify that the corresponding ESDT(s) has/have been properly installed.

- If no entry can be found for the ESDT in question it indicates either of the following conditions:
  - The ESDT was not installed.
  - The ESDT was not installed properly.

7 If there is an error in the log file indicating a subscription error, notify the Science Data Specialist or whoever else installs ESDTs to verify that the corresponding ESDT(s) has/have been properly installed.

- If subscription errors exist in the log, it is most likely that the ESDT in question (for the input) was either not installed or improperly installed.
- Notify the Science Data Specialist or whoever else installs ESDTs to verify that the corresponding ESDT(s) has/have been properly installed.

8 If there is an error in the log file indicating communication problems, notify the Operations Controller/System Administrator of the suspected communication problems.

9 If there is an error in the log file indicating database problems, access a terminal window logged in to the Queuing Server host.

- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

10 If there is an error in the log file indicating database problems, type **ps –ef | grep sybase** then press **Return/Enter**.

- The following type of message is displayed:

  x0sps02{cmshared}43: ps -ef | grep sybase

  sybase 317 316 0 Apr 04 ? 0:48 /usr/ecs/OPS/COTS/sybase/bin/backupserver -Sx0sps02_backup_srvr -e/usr/ecs/OPS/

  sybase 314 311 25 Apr 04 ? 1944:49 /usr/ecs/OPS/COTS/sybase/bin/dataserver -d/dev/rdsk/c0t2d0s4 -sx0sps02_srvr -e/

  sybase 311 1 0 Apr 04 ? 0:00 /bin/sh /usr/ecs/OPS/COTS/sybase/install/RUN_x0sps02_srvr

sybase 316 1 0 Apr 04 ? 0:00 /bin/sh /usr/ecs/OPS/COTS/sybase/install/RUN_x0sps02_backup_srvr

sybase 324 1 0 Apr 04 ? 0:00 /usr/ecs/OPS/COTS/sybase/bin/monserver -Sx0sps02_srvr -Mx0sps02_monitor_srvr -U

cmshared 15516 15405 0 14:20:33 pts/6 0:00 grep sybase

- In the example the entry that includes **"/usr/ecs/OPS/COTS/sybase/bin/dataserver -d/dev/rdsk/c0t2d0s4 -sx0sps02_srvr -e/"** indicates that the PDPS database server is running.
- If the PDPS database server were **not** running, the following type of message only would be displayed:

cmshared 15516 15405 0 14:20:33 pts/6 0:00 grep sybase

11 If there is an error in the log file indicating database problems and the database servers are running, go to the procedure for **Checking the PDPS Database for Causes of Failure to Generate DPRs** (subsequent section of this lesson).

- Check for lack of (or corruption of) data in the database.

12 If the database servers are **not** running, notify the Database Administrator of the problem.

13 If there is an error in the log file indicating lack of disk space, type **cd /*path*** then press **Return/Enter**.

- Change directory to a directory on the full disk from which files can be deleted.

14 If there is an error in the log file indicating lack of disk space, type **rm *filename1 … filenamex*** then press **Return/Enter**.

- ***filename1 … filenamex*** represent unnecessary files to be removed.
- Use wild cards if appropriate.
  - For example:

    rm *.hdf

    would prompt the system to remove all files with a suffix of .hdf.

15 If removing files from a full disk, respond appropriately to system prompts.

- For example:

rm: remove trashfile.rs (yes/no)? y

16 If there is a recurring disk space problem, notify the Operations Controller/System Administrator of the problem.

17 If the log contains an error message which indicates that a configuration item in another subsystem may be the source of the problems, consult with the relevant technician (e.g., Distribution Technician) to request assistance in isolating the problem.

18 If the log does **not** contain an error message that indicates why the DPR generation failed, perform the procedure for **Checking the PDPS Database for Causes of Failure to Generate DPRs** (subsequent section of this lesson).

19 When the problem has been corrected, use the procedure for **Creating a New Production Request Using the Production Request Editor GUI** (previous section of this lesson) to re-create the Production Request that led to discovery of the problem.

Using ISQL to Check Database Tables

The PDPS database is the repository of data concerning PGEs, Production Requests, Data Processing Requests, Production Strategies, Production Plans and other production-related data. The Subscription Server (SubsSrv) database contains data concerning subscriptions.

The data stored in databases can be checked using either a database browser or isql commands. The procedure in this section describes how to check the tables using isql commands.

The procedure for using isql to check database tables starts with the assumption that the operator has logged in to the system.

Using ISQL to Check Database Tables

---

1 Access a terminal window logged in to the appropriate host.

- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.
- Examples of Subscription Server host (Sun Consolidation Internal Server host) names include **e0acs11** and **l0acs03**.
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 Type **isql -U *userID* -S *DBServer*** then press **Return/Enter**.

- For example:

  isql –U pdps_role –S x0sps02_srvr

3 At the **Password:** prompt type ***dbpassword*** then press **Return/Enter**.

- The ***dbpassword*** is the password for logging in to the database using the specified ***userID***.

4 Type **use** ***dbname*** at the **1>** prompt then press **Return/Enter**.

- The ***dbname*** is likely to be one of the following names:
  - **pdps** [OPS mode].
  - **pdps_TS1** [TS1 mode].
  - **pdps_TS2** [TS2 mode].

5 Type **go** at the **2>** prompt then press **Return/Enter**.

6 Type **select * from** ***TableName*** at the **1>** prompt then press **Return/Enter**.

- Alternatively, type **select** ***columnName*** **from** ***TableName*** at the **1>** prompt then press **Return/Enter**.
- Another alternative is to type **select** ***columnName1,columnName2[,columnName3,…]*** **from** ***TableName*** at the **1>** prompt then press **Return/Enter**.

7 Type **go** at the **2>** prompt then press **Return/Enter**.

- Table contents are displayed.
  - If * was specified, all entries in the table are displayed.
  - If specific columnNames were entered, the data associated with those columns only are displayed.

8 To exit from isql type **quit** at the **1>** prompt then press **Return/Enter**.

Checking the PDPS Database for Causes of Failure to Generate DPRs

The PDPS database is a useful resource for troubleshooting a failure to generate DPRs. Certain values must be entered in tables in the PDPS database during the DPR generation process in order for the DPRs to be successfully generated.

- The subscriptionFlag values (PlDataTypeMaster table) for all the data types (ESDTs) needed for the PGE are set.
- The Science Data Server UR is entered in the dataServUrString column (PlDataTypeMaster table) for all data types (ESDTs) needed for the PGE.
- If the PGE specified in the Production Request requires static files, the URs for the static files are included in PlDataGranuleShort table (universalReference column).

The procedure to check the PDPS database for causes of failure to generate DPRs starts with the assumption that the operator has logged in to the system.

Checking the PDPS Database for Causes of Failure to Generate DPRs

1 Access a terminal window logged in to the Planning/Management Workstation host.

- Examples of Planning/Management Workstation host names include **e0pls03** and **l0pls02.**
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 Log in to the PDPS database as described in Steps 1 through 5 of the procedure for **Using ISQL to Check Database Tables** (previous section of this lesson).

3 Type **select dataTypeId,subscriptionFlag from PlDataTypeMaster** at the **1>** prompt then press **Return/Enter**.

- Prepare a request to view PlDataTypeMaster table subscriptionFlag column values for all data types (dataTypeId column).

4 Type **go** at the **2>** prompt then press **Return/Enter**.

- Contents of specified columns of the **PlDataTypeMaster** table are displayed.
- For example:

```
1> select dataTypeId,subscriptionFlag from PlDataTypeMaster
2> go
 dataTypeId          subscriptionFlag
 -------------------- ----------------
 AM1ATTNF#001                      40
 AM1EPHN0#001                      39
 AP#001                     1
 ActSched#001                  38
 DAP#001                     1
 MI1B1#001                    0
 MI1B2E#001                    0
 MI1B2T#001                    0
 MIANCAGP#001                    0
 MIANCARP#001                    0
 MIANCSSC#001                   0
 MIANPP#001                    0
 MIANRCCH#001                    0
 MIB1LM#001                    0
 MIB2GEOP#001                   42
 MICNFG#001                    0
 MIL1A#001                    0
```

```
PGEEXE#001                    1
PH#001                   1
SSAPC#001                  1

(20 rows affected)
```

- During DPR generation, the DPR executable should turn the subscriptionFlag for all the data types (ESDTs) needed for the PGE from zero to non-zero.
- If the subscriptionFlag values for the data types did not turn to non-zero, subscription trouble is indicated.

5 If the subscriptionFlag value for any ESDT needed for the PGE is **zero**, make a note of the fact for subsequent reporting of the problem.

6 Type **select dataTypeId,dataServUrString from PlDataTypeMaster** at the **1>** prompt then press **Return/Enter**.

- Prepare a request to view PlDataTypeMaster table dataServUrString column values for all data types.

7 Type **go** at the **2>** prompt then press **Return/Enter**.

- Contents of specified columns of the **PlDataTypeMaster** table are displayed.
- For example:

```
1> select dataTypeId,dataServUrString from PlDataTypeMaster
2> go
 dataTypeId
        dataServUrString
 --------------------
        ---------------------------------------------------------
 AM1ATTNF#001
        UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]
 AM1EPHN0#001
        UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]
 AP#001
        NULL
 ActSched#001
        UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]
 DAP#001
        NULL
 MI1B1#001
        UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]
 MI1B2E#001
        UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]
```

MI1B2T#001
UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]
MIANCAGP#001
UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]
MIANCARP#001
UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]
MIANCSSC#001
UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]
MIANPP#001
UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]
MIANRCCH#001
UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]
MIB1LM#001
NULL
MIB2GEOP#001
UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]
MICNFG#001
UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]
MIL1A#001
UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]
PGEEXE#001
UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]
PH#001
NULL
SSAPC#001
NULL

(20 rows affected)

- During DPR generation, the DPR executable should turn the dataServUrString for all the ESDTs needed for the PGE from "NULL" to the UR value for Science Data Server (e.g., UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]).

8 If the dataServUrString value for any ESDTs needed for the PGE does not have the UR value for the Science Data server (e.g., is NULL), make a note of the fact for subsequent reporting of the problem.

9 If the PGE specified in the Production Request requires static files, type **select dataTypeId,universalReference from PlDataGranuleShort** at the **1>** prompt then press **Return/Enter**.

- Prepare a request to view PlDataGranuleShort table universalReference column for the URs of the necessary static files.

- If there is data on many data types in the PlDataGranuleShort table, it may be advisable to limit the search, for example:

  1> select dataTypeId,universalReference from PlDataGranuleShort where dataTypeId = "AM1ATTNF#001"

  – In the example **AM1ATTNF#001** is the data type being checked.

10 Type **go** at the **2>** prompt then press **Return/Enter**.

- Contents of specified columns of the **PlDataGranuleShort** table are displayed.

- For example:

```
1> select dataTypeId,universalReference from PlDataGranuleShort where
dataTypeId = "AM1ATTNF#001"
2> go
 dataTypeId
        universalReference
 --------------------
        ----------------------------------------------------------------------------------------------------
---------------------------------

 AM1ATTNF#001
        UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]:20:
SC:AM1ATTNF.001:1449

 (1 row affected)
```

- If the PGE requires static files, the URs for the static files must be included in the PlDataGranuleShort table in order for a DPR to be successfully generated.

- For dynamic granules, the corresponding UR values may become available during DPR generation; however, if the dynamic granules' URs do not become available during DPR generation, there is no effect on DPR generation.

11 If the PGE requires static files, and the URs for the static files are **not** in the PlDataGranuleShort table, make a note of the fact for subsequent reporting of the problem.

12 To exit from isql type **quit** at the **1>** prompt then press **Return/Enter**.

13 If any problems were noted in the PDPS database (e.g., subscriptionFlag value zero or no static file UR), report the problem(s) to the SSI&T team and/or call the help desk and submit a trouble ticket in accordance with site Problem Management policy.

- There may be a problem with PGE registration (especially if no other PGEs are affected) or other problem associated with the SSI&T process.

DPR Explosion Fails Because Production Request Editor does Not Query DSS for Data

If the Production Request Editor reports a failure creating a DPR that is subject to the Metadata Query production rule, the process for responding to the problem consists of the following activities:

- Check the "debug" log for evidence that the Science Data Server was queried for metadata values.
- If there is no message in the debug log indicating that the Science Data Server was queried for metadata values, the SSI&T team should check for an error in the appropriate PGE science metadata ODL file.

Checking the Production Request Editor Debug File for Evidence of Metadata Queries

The procedure for checking the Production Request Editor Debug file starts with the assumption that the operator has logged in to the system.

Checking the Production Request Editor Debug File for Evidence of Metadata Queries

---

1 Access a terminal window logged in to the Planning/Management Workstation host.

- Examples of Planning/Management Workstation host names include **e0pls03** and **l0pls02.**
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 Type **cd /usr/ecs/*MODE*/CUSTOM/logs** then press **Return/Enter**.

- Change directory to the directory containing the planning log files (e.g., EcPlPREditorDebug.log).

3 Type **pg *filename*** then press **Return/Enter**.

- ***filename*** refers to the log file to be reviewed (e.g., EcPlPREditorDebug.log).
- The first page of the log file is displayed.
- Although this procedure has been written for the **pg** command, any UNIX editor or visualizing command (e.g., **vi**, **view**, **more**) can be used to review the log file.

4 Review the log file to determine whether the log contains a message that indicates why the DPR generation failed.

- When the Production Request Editor queries the Science Data Server to search for the PGE's input(s), the query should include the metadata value(s) desired by the PGE.
- Note that events are recorded in the debug log in chronological order and are preceded by a time and date stamp.
- If there is no message which indicates that the Science Data Server was queried for the metadata value(s) desired by the PGE, there may be an error in the PGE Metadata ODL File for the PGE.

5 If there was no message in the debug log indicating that the Science Data Server was queried for metadata values, submit a request to the SSI&T team to check for an error in the appropriate PGE science metadata ODL file and re-register the PGE.

- After the SSI&T team has edited the ODL file and re-registered the PGE it should be possible to successfully create DPRs for the PGE.

Responding to PR or DPR Deletion that Hangs

When deleting DPRs or PRs from the Production Request Editor, the Job Management Server and Deletion Server (both in DPS) are called to clean up all PDPS database tables associated with the DPR or PR. It is possible that any failure or "hung" condition can be attributed to the Production Request Editor itself or the Deletion or Job Management Server. If the servers are functioning properly, there are two other possible causes:

- The database is locked.
- There may be resource locks in the PDPS database.

Accordingly, the following procedures may be involved in responding to PR or DPR deletion that hangs:

- Checking for Database Deadlocks.
- Checking for Resource Locks in the PDPS Database.

Checking for Database Deadlocks

A deadlock occurs when a database transaction locks a record that another transaction needs and the second transaction locks the record that first transaction needs. Each program must wait until the other completes. However, neither can complete (because each is waiting for the other) so both end up waiting indefinitely.

The procedure for checking for database deadlocks starts with the assumption that the operator has logged in to the system and the Planning/Management Workstation (e.g., e0pls03 or l0pls02).

Checking for Database Deadlocks

---

1 Log in to the PDPS database as described in Steps 1 through 5 of the procedure for **Using ISQL to Check Database Tables** (previous section of this lesson).

2 Type **sp_lock** at the **1>** prompt then press **Return/Enter**.

3 Type **go** at the **2>** prompt then press **Return/Enter**.

- The following type of result is displayed:

```
--
 spid   locktype                  table_id    page       dbname
        class
 ------ ---------------------------- ----------- ----------- ---------------
        ------------------------------
     17 Sh_intent                  197575742          0 pdps
        Non Cursor Lock
     20 Sh_intent                  197575742          0 pdps
        Non Cursor Lock
     23 Sh_intent                  448004627          0 master
        Non Cursor Lock
     24 Ex_intent-blk                85575343          0 pdps
        Non Cursor Lock
     24 Ex_page                   85575343       1225 pdps
        Non Cursor Lock
     24 Ex_page                   85575343       1257 pdps
        Non Cursor Lock
     24 Ex_table-blk                85575343          0 pdps
        Non Cursor Lock
     24 Ex_intent                 197575742          0 pdps
        Non Cursor Lock
     24 Ex_page-blk                197575742       1313 pdps
        Non Cursor Lock

 (9 rows affected, return status = 0)
```

- In the preceding example the…
  - **spid** column shows the process id. The database user that owns a process can be determined using the **sp_who** isql command.

– **locktype** column indicates a problem if the entry starts with "Ex_" (exclusive).

– **table_id** column identifies the table that the corresponding spid has locked. The name of the table can be determined using the **select** command [i.e., **select object_name (table_id)**].

- In the example the locks of concern are related to spid 24, Table IDs 197575742 and 85575343.

4 Type **select object_name (*table_id*)** at the **1>** prompt then press **Return/Enter**.

- For example, to check the exclusive locks related to spid 24, table ID 197575742 in the preceding example, type:

  1> select object_name (197575742)

5 Type **go** at the **2>** prompt then press **Return/Enter**.

- The following type of result is displayed:

  PlDprData

  (1 row affected)

  – Table ID 197575742 refers to the PlDprData table in the database.

- To check the exclusive locks related to spid 24, table ID 85575343 in the preceding example, type:

  1 select object_name (85575343)
  2 go

  – The following type of result is displayed:

    PlDataGranuleShort

    (1 row affected)

  – Table ID 85575343 refers to the PlDataGranuleShort table in the database.

6 Type **sp_who** at the **1>** prompt then press **Return/Enter**.

7 Type **go** at the **2>** prompt then press **Return/Enter**.

- The following type of result is displayed:

```
spid   status      loginame                    hostname   blk
      dbname                     cmd
 ------ ------------ ----------------------------- ---------- -----
      ----------------------------- ----------------
     1 recv sleep   EcPlSubMgr                          0
      pdps_TS2                   AWAITING COMMAND
     8 recv sleep   EcPlSubMgr                          0
      pdps_TS2                   AWAITING COMMAND
     9 recv sleep   EcDpPrDeletion                      0
      pdps_TS2                   AWAITING COMMAND
    10 recv sleep   EcDpPrJobMgmt                       0
      pdps_TS2                   AWAITING COMMAND
    11 recv sleep   EcPlSubMgr                          0
      pdps_TS1                   AWAITING COMMAND
    12 recv sleep   EcPlSubMgr                          0
      pdps_TS1                   AWAITING COMMAND
    13 recv sleep   EcDpPrDeletion                      0
      pdps_TS1                   AWAITING COMMAND
    14 recv sleep   EcDpPrJobMgmt                       0
      pdps_TS1                   AWAITING COMMAND
    15 lock sleep   EcDpPrDeletion                      24
      pdps                  SELECT
    16 recv sleep   EcDpPrJobMgmt                       0
      pdps                  AWAITING COMMAND
    17 lock sleep   EcPlSubMgr                          24
      pdps                  SELECT
    18 recv sleep   EcPlSubMgr                          0
      pdps                  AWAITING COMMAND
    19 recv sleep   pdps_role               l0ais01     0
      pdps_TS2                   AWAITING COMMAND
    20 lock sleep   EcDpPrEM                            24
      pdps                  SELECT
    21 recv sleep   EcDpPrJobMgmt                       0
      pdps                  AWAITING COMMAND
    22 lock sleep   EcDpPrDm                            24
      pdps                  SELECT
    23 running      EcPlWb                  l0pls01     0
      pdps                  SELECT
```

```
24 recv sleep   EcPlSubMgr                  0
   pdps                 AWAITING COMMAND
25 recv sleep   EcPlSubMgr                  0
   pdps                 AWAITING COMMAND
```

(25 rows affected, return status = 0)

- The **blk** column shows the spid of the process that is doing the blocking.
- The **cmd** column shows the command that the blocked process is trying to complete.

8 Analyze the results of the request.

- In the example shown, EcPlSubMgr (spid 24) is blocking SELECT transactions by the following database users:
  - EcDpPrEM (spid 20, spid 22).
  - Itself (spid 17 - presumably another thread).
  - EcDpPrDeletion (spid 15).

9 To exit from isql type **quit** at the **1>** prompt then press **Return/Enter**.

10 If there is a deadlock in the database, ask the Operations Controller to bounce the server that is causing the deadlock.

- In the preceding example, the Subscription Manager (EcPlSubMgr) should be bounced.

11 If there is no deadlock, perform the procedure for **Checking for Resource Locks in the PDPS Database** (subsequent section of this lesson).

Checking for Resource Locks in the PDPS Database

Resource locks used to occur if there was an attempt to delete DPRs/PRs while their corresponding jobs were still running in AutoSys or jobs had been explicitly killed before the DPRs/PRs were deleted. However, resource locking has been removed for all Resource Management calls (e.g., for allocating CPUs and disk space). The locks have been replaced with the following features:

- Sybase stored procedures that use transactions.
- Database triggers.

Resource locking is still used for disk space reclamation. Momentary system interruptions occur during the process of disk space reclamation. The interruptions may happen several times a day. The system may look like it is "hung" during such periods. The procedure that follows should be performed to verify that disk space reclamation is proceeding normally.

Although the procedure for checking for resource locks in the PDPS database includes the use of isql commands, an acceptable alternative is to use a database browser to check the contents of the DpPrResourceLock table. The procedure for checking for resource locks in the PDPS database starts with the assumption that the operator has logged in to the system and the Planning/Management Workstation (e.g., e0pls03 or l0pls02).

Checking for Resource Locks in the PDPS Database

---

1 Log in to the PDPS database as described in Steps 1 through 5 of the procedure for **Using ISQL to Check Database Tables** (previous section of this lesson).

2 Type **select * from DpPrResourceLock** at the **1>** prompt then press **Return/Enter**.

- Prepare a request to view the contents of the DpPrResourceLock table.

3 Type **go** at the **2>** prompt then press **Return/Enter**.

- The contents of the **DpPrResourceLock** table are displayed.
- For example:

  jobId priority ecsUnit attempts
  state pid queuePosition
  -------------------------------------- ----------- ----------- -----------
  ----------- ----------- -------------------

  (0 rows affected)

- There are no resource locks in the example.

4 Analyze the results of the request.

- A jobId with a state <> 0 would indicate a resource lock.
  - If there are entries in the **DpPrResourceLock** table and there are no other jobs running in AutoSys, all entries in the table need to be deleted before the DPR/PR deletion can complete.
  - If other jobs (DPRs) are currently being executed in AutoSys and the other jobs should not be deleted, the entries in the table that need to be deleted are those related to the job to be deleted only. The entries concerning the other (running) jobs must be left in the table.
- If there is no evidence of a resource lock, go to Step 8.

5 If all entries in the DpPrResourceLock table are to be deleted, type **delete DpPrResourceLock** at the **1>** prompt then press **Return/Enter**.

- Go to Step 7.

6 If some (but not all) entries in the DpPrResourceLock table are to be deleted, type **delete DpPrResourceLock where jobId like "*jobId*"** at the **1>** prompt then press **Return/Enter**.

- ***jobId*** specifies the job whose entries are to be deleted.

7 Type **go** at the **2>** prompt then press **Return/Enter**.

- Entries in the **DpPrResourceLock** table are deleted.
- The DPR/PR deletion that was delayed by the resource lock should go to completion.

8 To exit from isql type **quit** at the **1>** prompt then press **Return/Enter**.

- If entries were deleted from the DpPrResourceLock table the procedure is finished; otherwise, continue with Step 9.

9 Access a terminal window logged in to the Queuing Server host.

- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

10 Type **cd /usr/ecs/*MODE*/CUSTOM/logs** then press **Return/Enter**.

- Change directory to the directory containing the data processing log files (e.g., EcDpPrJobMgmt.ALOG, EcDpPrDeletion.ALOG).

11 Type: **tail -f *jobId*.err** then press **Return/Enter**.

- ***jobId*.err** refers to the data processing log file to be reviewed.

12 Observe the log file to determine whether entries are being made in the file.

- If entries are being made in the log file, there is probably no resource lock.

13 To quit tailing the log in the terminal window type **Ctrl-c**.

- A command line prompt is displayed in the terminal window.

14 Ensure that it is possible to connect to the necessary hosts and servers.

- For detailed instructions refer to the procedure for **Checking Connections to Hosts/Servers** (previous section of this lesson).

15 If no there is no database deadlock or resource lock and the Data Processing Subsystem servers (especially Deletion Server and Job Management Server) are up, call the help desk and submit a trouble ticket in accordance with site Problem Management policy.

Responding to DPR Deletion that Fails

When deleting DPRs or PRs from the Production Request Editor, both the Job Management Server and Deletion Server (both DPS servers) are called to clean up all PDPS database tables associated with the DPR or PR. It is possible that any failure or "hung" condition can be attributed to the Production Request Editor itself or to the Job Management Server or Deletion Server.

The procedure for responding to DPR deletion that fails starts with the assumption that the operator has logged in to the system and the Queuing Server host (e.g., e0sps04 or l0sps03).

Responding to DPR Deletion that Fails

---

1 Access a terminal window logged in to the Queuing Server host.

- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 Type **cd /usr/ecs/*MODE*/CUSTOM/logs** then press **Return/Enter**.

- Change directory to the directory containing the planning log files (e.g., EcDpPrDeletionDebug.log).

3 If there is a Deletion Server Debug log, type **pg *filename*** then press **Return/Enter**.

- ***filename*** refers to the data processing log file to be reviewed (e.g., EcDpPrDeletionDebug.log).
- The first page of the log file is displayed.
- Although this procedure has been written for the **pg** command, any UNIX editor or visualizing command (e.g., **vi**, **view**, **more**) can be used to review the log file.

4 If there is a Deletion Server Debug log, review the log file for error messages.

- For example:

  Could not make database interface to DpPrFile
  Aborting: will report error to client.

- The message in the example indicates a compilation problem in Deletion Server so Deletion Server cannot communicate with the PDPS database.
  - When Deletion Server was compiled the RogueWave database (RWDB) libraries were not linked correctly.

- The most likely cause is the fact that the Deletion Server Imakefile was changed so that the DPS make.options was included.

5 If an error message of the type shown in the previous step was present in the log, notify the Operations Controller/System Administrator of the problem.

Handling a DPR Scheduling failure

There are three principal sources of information concerning failures to activate DPRs from the Planning Workbench:

- The Planning Workbench GUI itself may provide an error message.
  - The message would indicate either success or failure.
  - It would not indicate why the request failed.
- The Planning Workbench Message Handler (the window that comes up before the Planning Workbench and to which the Workbench logs various messages).
  - Any error encountered should be logged to the Message Handler.
- The Planning Workbench logs (debug log and ALOG).
  - The logs are located in the /usr/ecs/*MODE*/CUSTOM/logs directory and begin with “EcPlWb” (e.g., EcPlWbDebug.log, EcPlWb.ALOG).
  - The logs can further refine the cause of the error.

If job activation fails from the Planning Workbench, one of the following errors is likely to be reported:

- "DPR Validation Failed"
- "information (INFO) Production Request {Production Request Id} has unschedulable DPR {DPR Id}"

Accordingly, the following procedures may be involved in handling a DPR scheduling failure:

- Responding to a "DPR Validation Failed" Error.
- Responding to an "information (INFO) Production Request {Production Request Id} has unschedulable DPR {DPR Id}" Error.

Responding to a "DPR Validation Failed" Error

There are two conditions that may cause a "DPR Validation Failed" error to be reported:

- Performance data for the PGE are missing from the PDPS PlPgePerformance database table.
- Resource information for the PGE is missing from the PDPS PlResourceRequirement database table.

Consequently, it is necessary to check the appropriate PDPS database tables to determine whether the necessary PGE information is in the database.

Although the procedure for responding to a "DPR Validation Failed" error includes the use of isql commands, an acceptable alternative is to use a database browser to check the contents of the PlPerformance and PlPgePerformance tables. The procedure for responding to a "DPR Validation Failed" error starts with the assumption that the operator has logged in to the system and the Planning/Management Workstation (e.g., e0pls03 or l0pls02).

Responding to a "DPR Validation Failed" Error

---

1 Log in to the PDPS database as described in Steps 1 through 5 of the procedure for **Using ISQL to Check Database Tables** (previous section of this lesson).

2 Type **select * from PlPgePerformance** at the **1>** prompt then press **Return/Enter**.

- Prepare a request to view the contents of the PlPgePerformance table.

3 Type **go** at the **2>** prompt then press **Return/Enter**.

- The contents of the **PlPgePerformance** table are displayed.
- For example:

```
1> select * from PlPgePerformance
2> go
 pgeId           cpuTime     pgeElapsedTime dprElapsedTime
      maxMemory          faults     swaps      blockInputOperation
      blockOutputOperation runCpuTime  runMaxMemory        runPgeElapsed
      runDprElapsed runFaults   runSwaps    runBlockInOperation
      runBlockOutOperation sharedMemory        runSharedMemory
 ----------------- ----------- -------------- --------------
      -------------------- ----------- ----------- -------------------
      -------------------- ----------- -------------------- -------------
      ------------- ----------- ----------- -------------------
      -------------------- -------------------- --------------------
 ACT#syn1#01            550         1800         1800
           10.000000       100        100             100
                100        550          10.000000        1800
```

```
                 1800        100        100              100
                      100          0.000000          0.000000
ETS#syn1#01               55         1800         1800
                  10.000000       100        100              100
                      100        55          10.000000        1800
                 1800        100        100              100
                      100          0.000000          0.000000
```

(2 rows affected)

- – The PGEs in the example have appropriate entries for performance data.

- For the PGE(s) that is (are) not schedulable, verify there are non-zero values for the entries in the table.
  - – As shown in the example, zero may be appropriate for some columns but not for all.

4 Type **select * from PlResourceRequirement** at the **1>** prompt then press **Return/Enter**.

- Prepare a request to view the contents of the PlResourceRequirement table.

5 Type **go** at the **2>** prompt then press **Return/Enter**.

- The contents of the **PlResourceRequirement** table are displayed.
- For example:

```
1> select * from PlResourceRequirement
2> go
 sswId           string
      numOfCPUs
      computer
      diskSpace          topLevelShellName
      exeTarFileDiskSpace  mcfName
      ramSize             exeUntarFileDiskSpace
      exeTarUR
      pgeId          toolkitArchitecture  pgeCommands
 ----------------- ----------------------------------------------------------
      -----------
      ----------------------------------------------------------
      -------------------- ------------------------------
      -------------------- ------------------------------
      -------------------- ---------------------
      -----------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------
---
```

```
---------------------------------------------------
----------------- ------------------- -----------
 ACT#syn1         x0spg01_string
                 1
        NULL
                  10.000000 synpge_sgi6n32
                   1.787376 NULL
                   0.000000            0.000000

UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]:18:LM:P
GEEXE.001:3747
        ACT#syn1#01      sgi32              1010
 ETS#syn1         x0spg01_string
                 1
        NULL
                  10.000000 synpge_sgi6n32
                   1.787376 NULL
                   0.000000            0.000000

UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]:18:LM:P
GEEXE.001:3753
        ETS#syn1#01      sgi32              1010

(2 rows affected)
```

- The PGEs in the example have appropriate entries for resource data.

• For the PGE(s) that is (are) not schedulable, verify that there are non-zero values for the entries in the table.

  - As shown in the example, zero may be appropriate for some columns but not for all.

6 Exit from isql. (Type **quit** at the **1>** prompt then press **Return/Enter**.)

7 If the entries in either the **PlPgePerformance** or **PlResourceRequirement** table are not appropriate, make a request to the SSI&T team to have the correct values entered.

8 Delete the affected DPR(s) as described in the procedure for **Deleting a Data Processing Request** (previous section of this lesson).

9 When the SSIT team has completed updating the PGE information in the PDPS database, create a new Production Request to replace the affected DPR(s) as described in the procedure for **Creating a New Production Request Using the Production Request Editor GUI** (previous section of this lesson).

10 Create and activate a new Production Plan (to send the replacement DPR(s) to data processing) as described in the procedure for **Creating a New Production Plan** (previous section of this lesson).

Responding to an "information (INFO) Production Request {Production Request Id} has unschedulable DPR {DPR Id}" Error

When an "information (INFO) Production Request {Production Request Id} has unschedulable DPR {DPR Id}" error is reported on the Message Handler, one of two problems is likely:

- A predecessor DPR exited in a "bad" state.
  - The unschedulable DPR depends on the predecessor DPR for input data.
  - The predecessor DPR is in either a completed or an error status, or was not included in the activated plan.
- The current DPR has a "QA_FAILURE."
  - A metadata check or query for the DPR failed so the DPR cannot be scheduled.

Consequently, it is necessary to check the appropriate PDPS database tables and possibly the Planning Workbench debug log to localize the problem.

Although the procedure for responding to an "information (INFO) Production Request {Production Request Id} has unschedulable DPR {DPR Id}" error includes the use of isql commands, an acceptable alternative is to use a database browser to check the contents of the appropriate tables. The procedure starts with the assumption that the operator has logged in to the system and the Planning/Management Workstation (e.g., e0pls03 or l0pls02).

Responding to an "information (INFO) Production Request {Production Request Id} has unschedulable DPR {DPR Id}" Error

---

1 Log in to the PDPS database as described in Steps 1 through 5 of the procedure for **Using ISQL to Check Database Tables** (previous section of this lesson).

2 Type **select * from PlDataProcessingRequest** at the **1>** prompt then press **Return/Enter**.

- Prepare a request to view the contents of the PlDataProcessingRequest table.

- If there is data on many DPRs in the PlDataProcessingRequest table, it may be advisable to limit the search, for example:

  1> select * from PlDataProcessingRequest where dprId like "AM1Eph#30002282200TS2"

  – In the example **AM1Eph#30002282200TS2** is the predecessor DPR being checked.

3 Type **go** at the **2>** prompt then press **Return/Enter**.

- The contents of the **PlDataProcessingRequest** table are displayed.
- For example:

```
1>  select * from PlDataProcessingRequest
2> go
 dprId                        productionRequestId  priority
        predictedStart           actualStart
        completionState
        pgeId          baselineTime
        name                    dataStartTime
        dataStopTime            sswId          dprCollectionId
        alarmTime               lateMessagesSent autosysId
        tileId     timeStamp
 ---------------------------- -------------------- -----------
       ------------------------- ------------------------- ---------------
       ----------------- -------------------------
       ---------------------------- -------------------------
       ------------------------- ----------------- ---------------
       ------------------------- ---------------- --------------------
       ----------- -------------------------
 ACT#syn1#004130123TS2        Aster_Pr                  250
            Jan 18 2000  7:39PM       Jan 18 2000  7:39PM STARTED
        ACT#syn1#01            Jan 18 2000  7:39PM
        ACT#syn1#004130123TS2              Jul  4 1997  9:01AM
            Jul  4 1997  9:01AM ACT#syn1         NULL
            Jan 18 2000  7:39PM              0 VAT
              0       Jan 18 2000  7:51PM
 AM1Eph#30002282200TS2        AM1Eph_PR_NonLeap_1         250
            Jan 12 2000  2:00PM       Jan 12 2000  2:00PM SUCCESS
        AM1Eph#3000102         Jan 12 2000  2:00PM
       AM1Eph#30002282200TS2              Feb 28 2000  5:00PM
            Feb 28 2000  7:00PM AM1Eph#30001      NULL
            Jan 12 2000  2:00PM              0 VAT
              0       Jan 12 2000  2:28PM
```

- For the predecessor DPR to the unschedulable DPR, verify that the **completionState** is **SUCCESS**.

4 Exit from isql. (Type **quit** at the **1>** prompt then press **Return/Enter**.)

5 If the **completionState** for the predecessor DPR to the unschedulable DPR is **not SUCCESS** (for example, if completionState is FAILEDPGE or CANCELED), perform Steps 6 through 8; otherwise, go to Step 9.

6 Delete the unschedulable DPR(s) and its predecessor(s) as described in the procedure for **Deleting a Data Processing Request** (previous section of this lesson).

7 Create new Production Request(s) to replace the affected DPR(s) as described in the procedure for **Creating a New Production Request Using the Production Request Editor GUI** (previous section of this lesson).

8 Create and activate a new Production Plan (to send the replacement DPR(s) to data processing) as described in the procedure for **Creating a New Production Plan** (previous section of this lesson).

9 If the **completionState** for the predecessor DPR to the unschedulable DPR is **SUCCESS**, type **cd /usr/ecs/*MODE*/CUSTOM/logs** then press **Return/Enter**.

- For ***MODE*** type the desired operating mode.
- Change directory to the directory containing the appropriate log files (e.g., EcPlWbDebug.log).

10 Type **pg** ***filename*** then press **Return/Enter**.

- ***filename*** refers to the data processing log file to be reviewed (e.g., EcPlWbDebug.log).
- The first page of the log file is displayed.
- Although this procedure has been written for the **pg** command, any UNIX editor or visualizing command (e.g., **vi**, **view**, **more**) can be used to review the log file.

11 Review the log file to determine whether the log contains an error message (concerning the unschedulable DPR) that indicates why the DPR generation failed.

- Look for an entry for the unschedulable DPR that contains either “QA_FAILURE” or “INVALID.”
  - "QA_FAILURE" indicates a problem with a metadata check/query so the DPR cannot be scheduled.
  - “INVALID” indicates that DPR validation failed.

12 If the log contains an error message that includes the term "INVALID", perform the procedure for **Responding to a "DPR Validation Failed" Error** (previous section of this lesson).

13 If the log contains a "QA_FAILURE" error message, log in to the PDPS database as described in Steps 1 through 4 of the procedure for **Using ISQL to Check Database Tables** (previous section of this lesson).

14 Type **select * from PlDprData** at the **1>** prompt then press **Return/Enter**.

- Prepare a request to view the contents of the PlDprData table.
- If there is data on many DPRs in the PlDprData table, it may be advisable to limit the search, for example:

  1> select * from PlDprData where dprId like "ACT#syn1#004130123TS2"

  – In the example **ACT#syn1#004130123TS2** is the DPR being checked.

15 Type **go** at the **2>** prompt then press **Return/Enter**.

- The contents of the **PlDprData** table are displayed.
- For example:

```
1> select * from PlDprData
2> go
 dprId
        granuleId
        logicalId   primaryType         accepted    theOrder    type
        temporalFlag timeWait    ioFlag      timerExp timerStart numNeeded
        waitForFlag linkId      minGranReq
 -----------------------------
        ----------------------------------------------------------------------------------------------
---------------------------------
        ----------- -------------------- ----------- ----------- -----------
        ------------ ----------- ----------- -------- ---------- -----------
        ----------- ----------- -----------
 ACT#syn1#004130123TS2
        AST_09T#00107041997130123000
             2000 NULL                       0          1
0
                0          0          1        0         0
1
               1       NULL          0
```

```
 ACT#syn1#004130123TS2
       AST_ANC#001L1000
             1200 1200                    1         1
0
                 0          0          0       0         0
1
                1       NULL          1
 ACT#syn1#004130123TS2
       AST_L1B#00107041997130123000
             1100 1100                    1         1
0
                 0          0          0       0         0
1
                1       NULL          1
 ACT#syn1#004130123TS2
       GDAS_0ZF#001O1000
             1101 1101                    1         1
0
                 0          0          0       0         0
1
                1       NULL          1
```

(4 rows affected)

- Note that for each DPR there are separate entries for the different data types.
- For all entries for the unschedulable DPR, verify that the values in the **accepted** column are **not** equal to 2.
  - A value of 2 in the **accepted** column indicates that the metadata check/query for the corresponding input (**granuleId** column) failed.

16 Type **select * from PlEsdtParam** at the **1>** prompt then press **Return/Enter**.

- Prepare a request to view the contents of the PlEsdtParam table.
- If there is data on many data types in the PlEsdtParam table, it may be advisable to limit the search, for example:

  1> select * from PlEsdtParam where dataTypeId like "AST_L1B#001"

  - In the example **AST_L1B#001** is the data type being checked.

17 Type **go** at the **2>** prompt then press **Return/Enter**.

- The contents of the **PlEsdtParam** table are displayed.

- For example:

```
1> select * from PlEsdtParam
2> go
 dataTypeId           paramName
         paramType
         containerName
         secondParm
         secondValue
 ------------------- --------------------------------------
        -------------------
        ---------------------------------------------------------------------------------------------
---
        --------------------------------------
        ----------------------------------------------------------------------------

 (0 rows affected)
```

  – Verify that there are entries for the appropriate **dataTypeId** in the PlEsdtParam table.
  – Note that in the example there are no entries for any data type.

18 If no entry for the unschedulable DPR has a value equal to 2 in the **accepted** column, type **select * from PlEsdtParmValues** at the **1>** prompt then press **Return/Enter**.

- Prepare a request to view the contents of the PlEsdtParmValues table.

19 Type **go** at the **2>** prompt then press **Return/Enter**.

- The contents of the **PlEsdtParmValues** table are displayed.
- For example:

```
1> select * from PlEsdtParmValues
2> go
 esdtParmName
         esdtParmVal
         granuleId
         secondParm
         secondValue
 --------------------------------------
        ----------------------------------------------------------------------------
        ---------------------------------------------------------------------------------------------
-------------------------------
        --------------------------------------
        ----------------------------------------------------------------------------
```

(0 rows affected)

- Verify that there are entries for the appropriate data type (in the **granuleId** column) in the PlEsdtParmValues table.
- Note that in the example there are no entries for any data type.

20 If there are entries for the appropriate data type in the **PlEsdtParam** and **PlEsdtParmValues** tables, type **select * from PlMetadataChecks** at the **1>** prompt then press **Return/Enter**.

- Prepare a request to view the contents of the PlMetadataChecks table.
- If there is data on many data types in the PlMetadataChecks table, it may be advisable to limit the search, for example:

  1> select * from PlMetadataChecks where dataTypeId like "AST_L1B#001"

  - In the example **AST_L1B#001** is the data type being checked.

21 Type **go** at the **2>** prompt then press **Return/Enter**.

- The contents of the **PlMetadataChecks** table are displayed.
- For example:

```
1> select * from PlMetadataChecks
2> go
 pgeId           dataTypeId
       paraName                              logicalId   type  operator
       value
       ioFlag     queryFlag   queryType
       secondParm
       secondValue
 ----------------- --------------------
      -------------------------------------- ----------- ----- --------
      ------------------------------------------------------------------------------
      ----------- ----------- -----------
      ---------------------------------------
      ------------------------------------------------------------------------------

(0 rows affected)
```

- Verify that the entries for the appropriate data type in the PlEsdtParam and PlEsdtParmValues tables match the expected values in the PlMetadataChecks table.
- Note that in the example there are no entries for any data type.

22 If an entry for the unschedulable DPR has a value equal to 2 in the **accepted** column of the **PlDprData** table (Step 15), make a record of the fact that the PGE cannot be run on the specified input data.

- The metadata check/query for the corresponding input failed.
- It may be possible to either reingest or reprocess the input data that failed the metadata check or query.
  - Steps 17 through 21 can give indications as to which applies, based on the data type(s) involved.

23 If no problem is identified in the preceding steps, call the help desk and submit a trouble ticket in accordance with site Problem Management policy.

Checking Log Files

Log files can provide indications of the following types of problems:

- Communication problems.
- Database problems.
- Lack of disk space.

The procedure for checking log files starts with the assumption that the operator has logged in to the system.

Checking Log Files

---

1 Access a terminal window logged in to the Planning/Management Workstation, Queuing Server, or PDPS DBMS Server as applicable.

- In general Planning Subsystem applications are installed on the Planning/ Management Workstation (e.g., e0pls03 or l0pls02) and Data Processing Subsystem (PRONG) applications are installed on the Queuing Server (e.g., e0sps04 or l0sps03). Note the following exceptions:
  - QA Monitor is on the Planning/Management Workstation.
  - Subscription Manager is on the Queuing Server (e.g., e0sps04 or l0sps03).
- Examples of Planning/Management Workstation host names include **e0pls03** and **l0pls02.**
- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.

- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 Type **cd /usr/ecs/*MODE*/CUSTOM/logs** then press **Return/Enter**.

- Change directory to the directory containing the production planning log files (e.g., EcPlPREditor.ALOG, EcPlPREditorDebug.log, EcPlWb.ALOG, EcPlWbDebug.log, or EcPlTl.ALOG).

3 Type **pg *filename*** then press **Return/Enter**.

- ***filename*** refers to the production planning log file to be reviewed (e.g., EcPlPREditor.ALOG, EcPlPREditorDebug.log, EcPlWb.ALOG, EcPlWbDebug.log, EcPlTl.ALOG, EcPlProdStrat.ALOG).
- The first page of the log file is displayed.
- Although this procedure has been written for the **pg** command, any UNIX editor or visualizing command (e.g., **more**, **view, vi**) can be used to review the log file.

4 Review the log file to identify problems that have occurred.

5 Respond to problems as follows:

- Production Planning-related problems.
  – Perform the appropriate procedure(s) from the list near the beginning of the **Production Planning Troubleshooting Procedures** section.
- Communication problems.
  – Notify the Operations Controller/System Administrator of suspected communication problems.
- Database problems.
  – Verify that relevant database servers are running.
  – Check for lack of (or corruption of) data in the database using either a database browser or isql commands.
  – Notify the Database Administrator of suspected database problems.
- Lack of disk space.
  – Remove unnecessary files.
  – Notify the Operations Controller/System Administrator of recurring disk space problems.

Checking Database Connections

If applications (including the GUIs) are unable to connect to the database, data cannot be retrieved or (in the case of the GUIs) displayed. Consequently, if a GUI does not display data or if the display does not refresh, checking the database connections is a logical step in trying to isolate the problem.

The procedure for checking database connections starts with the assumption that the operator has logged in to the system and the Planning/Management Workstation (e.g., e0pls03 or l0pls02).

Checking Database Connections

---

1 Submit a request to the Database Administrator to identify the values for parameters associated with the appropriate application.

- The following parameters are should be requested:
  - **DBName.**
  - **DBServer.**
  - **DBMaxConnections.**
- The preceding parameters are associated with the following applications:
  - EcPlPREditor.
  - EcPlProdStrat.
  - EcPlRm.
  - EcPlSubsEdit.
  - EcPlTl.
  - EcPlWb.

2 Access a terminal window logged in to the Queuing Server host.

- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

3 Type **isql -U *userID* -S *DBServer*** then press **Return/Enter**.

- For example:

  isql –U pdps_role –S x0sps02_srvr

4 At the **Password:** prompt type ***dbpassword*** then press **Return/Enter**.

- The ***dbpassword*** is the password for logging in to the database using the specified ***userID***.

5 Type **sp_who** at the **1>** prompt then press **Return/Enter**.

6 Type **go** at the **2>** prompt then press **Return/Enter**.

- A listing similar to the following one is displayed (some lines have been deleted):

```
spid   status      loginame                    hostname   blk
     dbname                      cmd
------ ----------- ---------------------------- ---------- -----
     ---------------------------- ----------------
     1 recv sleep   EcDpPrDeletion                          0
       pdps_TS1                   AWAITING COMMAND
     2 sleeping     NULL                                    0
       master                     NETWORK HANDLER
     3 sleeping     NULL                                    0
       master                     DEADLOCK TUNE
     4 sleeping     NULL                                    0
       master                     MIRROR HANDLER
     5 sleeping     NULL                                    0
       master                     HOUSEKEEPER
     6 sleeping     NULL                                    0
       master                     CHECKPOINT SLEEP
     7 sleeping     NULL                                    0
       master                     AUDIT PROCESS
     8 recv sleep   EcPlSubMgr                              0
       pdps                       AWAITING COMMAND
     9 recv sleep   EcPlSubMgr                              0
       pdps                       AWAITING COMMAND
    10 recv sleep   EcDpPrDeletion                          0
       pdps                       AWAITING COMMAND
    11 recv sleep   EcDpPrJobMgmt                           0
       pdps                       AWAITING COMMAND
    12 recv sleep   pdps_role                   x0ais03     0
       master                     AWAITING COMMAND
    13 recv sleep   EcDpPrJobMgmt                           0
       pdps                       AWAITING COMMAND
    14 recv sleep   EcDpPrDeletion                          0
       pdps                       AWAITING COMMAND
    15 recv sleep   pdps_role                   x0sps02     0
```

```
    master                   AWAITING COMMAND
  16 recv sleep   EcDpPrJobMgmt                    0
    pdps_TS1                  AWAITING COMMAND
  17 recv sleep   EcDpPrJobMgmt                    0
    pdps_TS1                  AWAITING COMMAND
  18 recv sleep   pdps_role              x0ais03    0
    tempdb                   AWAITING COMMAND
  19 recv sleep   EcDpPrDeletion                  0
    pdps_TS1                  AWAITING COMMAND
  20 recv sleep   pdps_role              x0sps02    0
    tempdb                   AWAITING COMMAND
  21 recv sleep   pdps_role              x0sps02    0
    pdps_TS1                  AWAITING COMMAND
  22 recv sleep   EcPlRm                         0
    pdps_TS1                  AWAITING COMMAND
  23 recv sleep   EcPlWb                         0
    pdps_TS1                  AWAITING COMMAND
  24 recv sleep   EcPlSubMgr                     0
    pdps_TS1                  AWAITING COMMAND
  25 recv sleep   EcPlSubMgr                     0
    pdps_TS1                  AWAITING COMMAND
  26 recv sleep   EcPlSubMgr                     0
    pdps_TS1                  AWAITING COMMAND
  27 recv sleep   EcPlSubMgr                     0
    pdps_TS1                  AWAITING COMMAND
  28 running      pdps_role              x0sps02    0
    pdps                   SELECT

(28 rows affected)
(return status = 0)
```

7 Type **sp_configure "user connections"** at the **1>** prompt then press **Return/Enter**.

8 Type **go** at the **2>** prompt then press **Return/Enter**.

- A listing similar to the following one is displayed:

```
Parameter Name              Default     Memory Used Config Value
    Run Value
 ------------------------------ ----------- ----------- ------------
      -----------
 number of user connections            25       7503        100
100
```

(1 row affected)
(return status = 0)

9 Type **quit** at the **1>** prompt then press **Return/Enter**.

10 Compare the number of actual connections (results of **sp_who**) with the number of connections for which the database has been configured (results of **sp_configure "user connections"**).

11 If the number of actual connections is very close to the number of connections for which the database has been configured, notify the Database Administrator of the fact.

12 If the number of actual connections is **not** very close to the number of connections for which the database has been configured, compare the number of actual connections with the value for DBMaxConnections that the Database Administrator specified (Step 1).

13 If the number of actual connections is very close to the value for DBMaxConnections, notify the Database Administrator of the fact.

- It may be advisable to increase the value assigned to the DBMaxConnections parameter in the Configuration Registry.

Production Processing

Processing Considerations

DPR Chaining

The Job Management Server in the DPS schedules chains of DPRs through the use of stored procedures in the PDPS database. In addition, when managing chains of DPRs, Job Management uses data from the following two tables in the PDPS database:

- DpPrPgeLimits.
- DpPrClassSchedulingLimits.

DpPrPgeLimits imposes restrictions on the number of chain-head DPRs of a particular PGE that can run simultaneously on the same virtual machine. A database record defines each PGE-virtual computer combination that will be run.

The DpPrPgeLimits table has the following columns:

- pgeId – PGE ID.
- computerName – virtual computer name.
- maxConcurrent - defines the maximum number of chain heads with a particular PGE ID that can run at the same time on a specific virtual computer.
- numConcurrent - shows how many chain heads with a particular PGE ID are running at the same time on a specific virtual computer.
- numScheduled - shows how many chain heads with a particular PGE ID are currently scheduled on a specific virtual computer.

If there were no record in the DpPrPgeLimits table for a particular PGE-computer combination that was scheduled, there would be no limits on the DPRs for that PGE.

- The PGE would be able to run and there would be no limits placed on how many DPRs for that PGE could run on the same virtual machine.
  - DPRs would run on the machine specified by the Production Planner in the Production Request (if the Production Planner designated a machine when creating the Production Request).
  - If no machine were specified in the Production Request, the machine would be determined from the PlResourceRequirement table (data entered during PGE registration).

DpPrClassSchedulingLimits limits the classes of DPRs that can run at any point in time. The classes correspond to the following types of processing that the system supports:

- Routine.
- On Demand.
- Reprocessing.

Consequently, the DpPrClassSchedulingLimits table has three records, one for each type of processing. Each record has the following fields:

- dprClass - assigned value identifies the type of processing.
  - 0 = Routine Processing.
  - 1 = On-Demand Processing.
  - 2 = Reprocessing.
- maxDprs - defines the maximum number of jobs of the type (specified in dprClass) that are allowed to run on the system.
- minDprs - currently not used.
- currentDprs - the number of jobs of the type (specified in dprClass) that are currently running.

If the DpPrClassSchedulingLimits table has no record for a particular type of processing, DPRs of that type are not allowed into AutoSys.

Values for the maxDprs and minDprs columns in the DpPrClassSchedulingLimits table are loaded at Job Management Server startup using data from the following two configuration parameters:

- DpPrMaxConcurrentDPRs - maximum allowed jobs
- DpPrMinConcurrentDPRs - minimum allowed jobs

Each parameter has three integer values; the first for routine processing; the second for on-demand processing; and the third for reprocessing jobs.

- For example, the Configuration Registry may have the following entries:

  DpPrMaxConcurrentDPRs = 100 60 40
  DpPrMinConcurrentDPRs = 0 0 0

  - In this case the maximum allowed jobs is 100 for routine processing, 60 for on-demand processing, and 40 for reprocessing.
  - The minimum allowed jobs is 0 for each type of processing.

The total number of completed jobs allowed in AutoSys is defined by the following configuration parameter:

- DpPrAutoSysMaxDPRs

Within the restrictions of the DpPrClassSchedulingLimits and DpPrPgeLimits database tables classes of DPR chains are scheduled by DPS with chains of DPRs (identified by their chain heads) being scheduled on the same machine whenever possible. When scheduling a chain-head DPR, attempts are made to schedule it on the machine that has the highest number of accepted inputs if accepted inputs are found. The chain-head DPR's children should be scheduled on the same machine.

Load balancing is done among virtual computers on the same string when selecting a computer for a DPR chain. If a DPR does not have its PGE identified in the DpPrPgeLimits table, that does not prevent it from being scheduled on a particular machine if most of the DPR's inputs are staged on that machine.

If there is more than one chain head for a set of DPRs, the DPRs are combined into a single chain.

- For example:
  - DPR#1 and DPR#2 are both parents of DPR#3.
  - Marking DPRs#1 and #2 as chain heads results in a single chain consisting of all three DPRs.
- Another example (MODIS processing):
  - If the Production Planner defines both the MODPGE01 PR and the MODPGE02 PR as chain heads, they are combined in one chain with the MODPGE01 DPR ID as the chainId.
  - MODPGE01 is designated the chain head because its outputs are used as inputs to MODPGE02 and MODPGE08.
  - MODPGE02 outputs are used as input to MODPGE08.

The PGEs that will be used as chain heads should be identified before installation. In selecting potential chain heads, it is recommended that PGEs which create substantial (i.e., large or many) inputs for other PGEs be so designated. MODPGE01 is an example of an appropriate chain head. DPREP PGEs are not considered good candidates for chain heads although their outputs are used as input to other PGEs.

Copy on Demand Feature

The term "copy on demand" describes a feature whereby the DPS code stages granules locally; i.e., granule files are staged to the processor on which the PGE is running. The copy on demand feature is specified in the following locations:

- onDemandCopy column in the PlDataTypeMaster database table (PDPS database):
  - The value of the onDemandCopy flag is used by the system to determine whether input files need to be locally staged.
- ON_DEMAND_COPY flag in the ESDT ODL files after DATA_TYPE and DATA_TYPE_VERSION.
  - For example:

    ON_DEMAND_COPY = “Y”
  - The ON_DEMAND_COPY flag indicates whether or not the DPS code should always stage the granule locally.
  - Whenever a PGE that uses the ESDT is registered, the value assigned to ON_DEMAND_COPY is put in the onDemandCopy column in the PlDataTypeMaster table of the PDPS database.

DPR Output Files Immediately Available as Input

Output files are immediately available as input when generated as output of a DPR. There is no waiting for Subscription Server notification before the output of a parent DPR is used as input to a child DPR.

AutoSys Production Scheduling Tool

The Planning and Data Processing Subsystems provide a batch processing environment to support the generation of data products. They manage, queue and execute Data Processing Requests (DPR) on the processing resources at a DAAC. The DPRs are submitted from the Planning Subsystem. The Planning and Data Processing Subsystems provide the operational interfaces needed to monitor the execution of science software PGEs specified in the DPRs.

The AutoSys software is a production scheduling tool intended to support the operational activities surrounding production processing. It assists with the following activities (among others):

- Job monitoring.
- Job scheduling.
- Fault notification.

- Job restart.
- Determining the effects of failure of a DPR.
- Determining the cause and actions to be taken due to the failure of a DPR.

AutoSys recognizes the following three categories of jobs:

- Box jobs.
- Command jobs.
- File-watcher jobs.

A box job is a collection of other jobs. It performs no processing action other than providing an organizational structure for a group of jobs that should be run within the same time period. Box jobs can be nested; i.e., box jobs can be included in other box jobs. Box jobs are particularly useful for organizing, managing, and administering large numbers of jobs, which are interrelated or have complex logic flows.

Box jobs are subject to the following rules:

- If no other starting conditions are specified at the job level, a job within a box job runs as soon as the starting conditions for the box are satisfied.
- If there are no job-level starting conditions for some of the jobs in a box, those jobs will run in parallel.
  - The jobs will run only once even if multiple start times are specified for some of the individual jobs.
  - Consequently, jobs in boxes will not be run several times inadvertently.
- Whenever any job in a box changes state, all jobs in the box are checked to see if they are eligible to be run.

A command job is the type most commonly thought of as a “job.” The “command” can be a shell script, the name of an executable program, a file transfer, or any other command that causes execution of a UNIX command on a client machine. When all of the starting conditions for the particular job have been met, AutoSys performs the following functions:

- Executes the command (runs the job).
- Captures the exit code at job completion.
- Sends the exit event (success or failure) and code back to the relational database management system (RDBMS).

A file-watcher job functions in a manner that is similar to a command job; however, it has a special purpose, i.e., to monitor the creation and size of a particular operating system file. When the file has reached a specified minimum size and is no longer increasing in size, the file-watcher job sends AutoSys an event indicating that the file has arrived. The file-watcher job allows AutoSys to know the status of external files that are needed in the processing of command jobs or box jobs.

When determining whether to start a job of any type, AutoSys evaluates the job with respect to the following the following starting parameters:

- Date and time scheduling parameters are met.
- Starting Conditions specified in the job definition evaluate to "true."
- For jobs in a box, the box must be in the RUNNING state.
- The current status of the job is not ON_HOLD or ON_ICE.

Every time there is an event that changes the truth of any of the preceding parameters, AutoSys finds all jobs that may be affected by the change and determines whether or not to start them.

Each DPR generated by the Planning Subsystem defines a box job for AutoSys. Every DPR/box job is composed of three command jobs that run in the following order:

- Preprocessing (EcDpPrEM)
- Execution (EcDpPrRunPGE)
- Postprocessing (EcDpPrEM)

The number of command jobs in an AutoSys Job box was reduced from seven to three in order to reduce overhead. The change reduces the load on the AutoSys Event Processor by about one half. It has no effect on the DPS Queuing Server CPU loading.

To create the new Preprocessing job the following old command jobs were combined:

- Allocation.
- Staging.
- Pre-processing.

To create the new Postprocessing job the following old command jobs were combined:

- Post-processing.
- Destaging.
- Deallocation.

The value in the database column dprState correlates to the seven old job states.

In general job names indicate the mode in which the DPR was generated and the stage of processing. The following list of job names for a DPREP DPR provides an illustration:

- AM1Eph#2.012302200TS2

- AM1Eph#2.012302200TS2R
- AM1Eph#2.012302200TS2E
- AM1Eph#2.012302200TS2P

The first job name in the list is a DPR/box job-level name. The first few characters identify the PGE (i.e., AM1Eph - Step 1 DPREP). The last three characters of the DPR/box job-level name (i.e., TS2) indicate the mode in which the DPR was generated. The last four characters of the remaining (command) job names in the list indicate the mode (i.e., TS2) and the stage of processing. The job name that ends in "R" is the preprocessing job; the job name that ends in "E" is the execution job; and the job name that ends in "P" is the postprocessing job.

Execution and postprocessing command jobs are dependent on successful completion of the command job that precedes them. For example, execution does not start until preprocessing has been successfully completed. The start of preprocessing depends on the box having started.

For postprocessing to start, the execution job must have completed. Postprocessing does not depend on execution having been "successful," just on being done. If execution failed, postprocessing handles failed PGE processing.

Just as most of the command jobs within a box job are dependent on the completion of other jobs, a DPR/box job itself may be dependent on the successful completion of some other box job(s). Such dependencies usually involve a need for the output of another DPR as input.

The following rules apply to DPR dependencies:

- Any DPRs that depend on data which are not yet available are kept in a "held" state by Job Management until their data availability subscriptions are fulfilled.
- The Subscription Manager software, which is part of the Planning Subsystem, receives subscription notifications for the DPRs and informs the DPS (Job Management) to release jobs to AutoSys after all data subscriptions for a given DPR are fulfilled.
- The Data Processing Subsystem (as monitored by the AutoSys Job Scheduling engine) runs the PGEs and associated jobs as the resources required for the tasks become available.
- The procedure continues until all scheduled DPRs have completed.

Each mode has multiple Job Management queues (i.e., one for each type of processing) and there is a maximum number of jobs allowed in AutoSys for each type of processing. For example, in each mode a DAAC would have queues for the following types of processing:

- Routine Processing.
- On-Demand Processing.
- Reprocessing

The DAAC Production Monitor uses AutoSys/Job Management Web Interface when monitoring and controlling job processing. That includes the following functions:

- Deleting/suspending/resuming jobs as required.
- Monitoring and providing processing status upon request.

The Production Monitor can configure some aspects of AutoSys, including the runtime options.

Launching the AutoSys GUI Control Panel

Launching the AutoSys GUI Control Panel

The following software applications are associated with Production Processing (excluding Science Software Integration and Test (SSI&T) and Science Data Processing Toolkit applications):

- Subscription Manager.
- Job Management.
- Execution Management.
- PGE Management.
- Deletion Server.
- AutoSys.
  - Event Processor (AutoSys daemon).
  - Event Server (Sybase database server for the AutoSys database).
  - AutoSys GUIs.
  - Job Management Web Interface.
- QA Monitor.
- Sybase ASE Server for the PDPS database.

Access to the Production Processing tools must be gained through the use of UNIX commands.

Launching the AutoSys GUI Control Panel starts with the assumption that the applicable servers are running and the Production Monitor has logged in to the system.

Launching the AutoSys GUI Control Panel

1 Access a terminal window logged in to the Queuing Server host.

- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 Type **setenv ECS_HOME /usr/ecs/** then press the **Return/Enter** key.

- When logging in as a system user (e.g., cmshared), the ECS_HOME variable may be set automatically so it may not be necessary to perform this step.

3 Type **cd /*path*** then press **Return/Enter**.

- Change directory to the directory (e.g., /usr/ecs/*MODE*/COTS/autotreeb/autouser, /usr/ecs/*MODE*/COTS/autotree/autouser, /data1/SHARED/COTS/autotree/autouser) containing the set-up files (e.g., FMR.autosys.csh.e0sps04).
- The particular path to be typed may vary from site to site.

4 Type **source *AUTOSYSINSTANCE*.autosys.csh.*hostname*** then press **Return/Enter**.

- An ***AUTOSYSINSTANCE*** (also called an AUTOSERV instance) is installed as part of the Data Processing Subsystem and is identified by three capital letters.
  - Examples of AUTOSYS (AUTOSERV) instances at DAACs include **FMR** and **SPG.**
  - Multiple AUTOSYS instances may be installed at a DAAC.

5 Type **cd /usr/ecs/*MODE*/CUSTOM/utilities** then press **Return/Enter**.

- Change directory to the directory containing the AutoSys start script (e.g., EcDpPrAutosysStart).
- The ***MODE*** will most likely be one of the following operating modes:
  - OPS (for normal operation).
  - TS1 (for SSI&T).
  - TS2 (new version checkout).
- Note that the separate subdirectories under /usr/ecs apply to (describe) different operating modes.

6 Type **EcDpPrAutosysStart *MODE AUTOSYSINSTANCE*** then press **Return/Enter**.

- The **AutoSys GUI Control Panel** (Figure 53) is displayed.

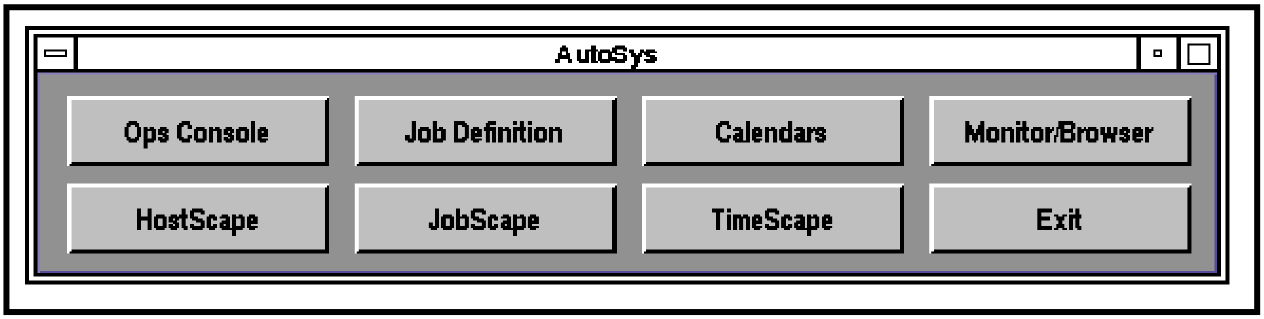

Figure 53. AutoSys GUI Control Panel

This page intentionally left blank.

Configuring AutoSys Screens/Displays

Configuring AutoSys Runtime Options

This section explains how to configure AutoSys runtime options. The Production Monitor can define the following runtime options:

- Refresh Interval.
  - Determines how often the View Region (the area on the right side of the GUI display where data are presented) will be updated.
- Inches/Hr.
  - Indicates how much information is displayed on the screen.

There are default values that apply to the runtime options until the Production Monitor modifies them.

Table 7 lists the runtime options available for TimeScape and JobScape. Not all options are available for both interfaces.

Table 7. Runtime Options Table

Interface	Refresh Interval	Inches/Hour
TimeScape	X	X
JobScape	X	

The procedure for configuring AutoSys runtime options assumes that AutoSys has been launched and the **AutoSys GUI Control Panel** (Figure 54) is being displayed. Perform only the steps that are applicable to the changes to be made (as specified in Table 7). Note that if the Freeze Frame feature is enabled, any changes entered will not take place until Freeze Frame has been disabled.

Configuring AutoSys Runtime Options

1 Click on either the **TimeScape** or **JobScape** button as applicable.

- The interface corresponding to the selected button is displayed (Figures 55-56).

2 Select **Options** → **Edit Runtime Options** from the pull-down menu.

- The **Runtime Options** dialogue box is displayed.

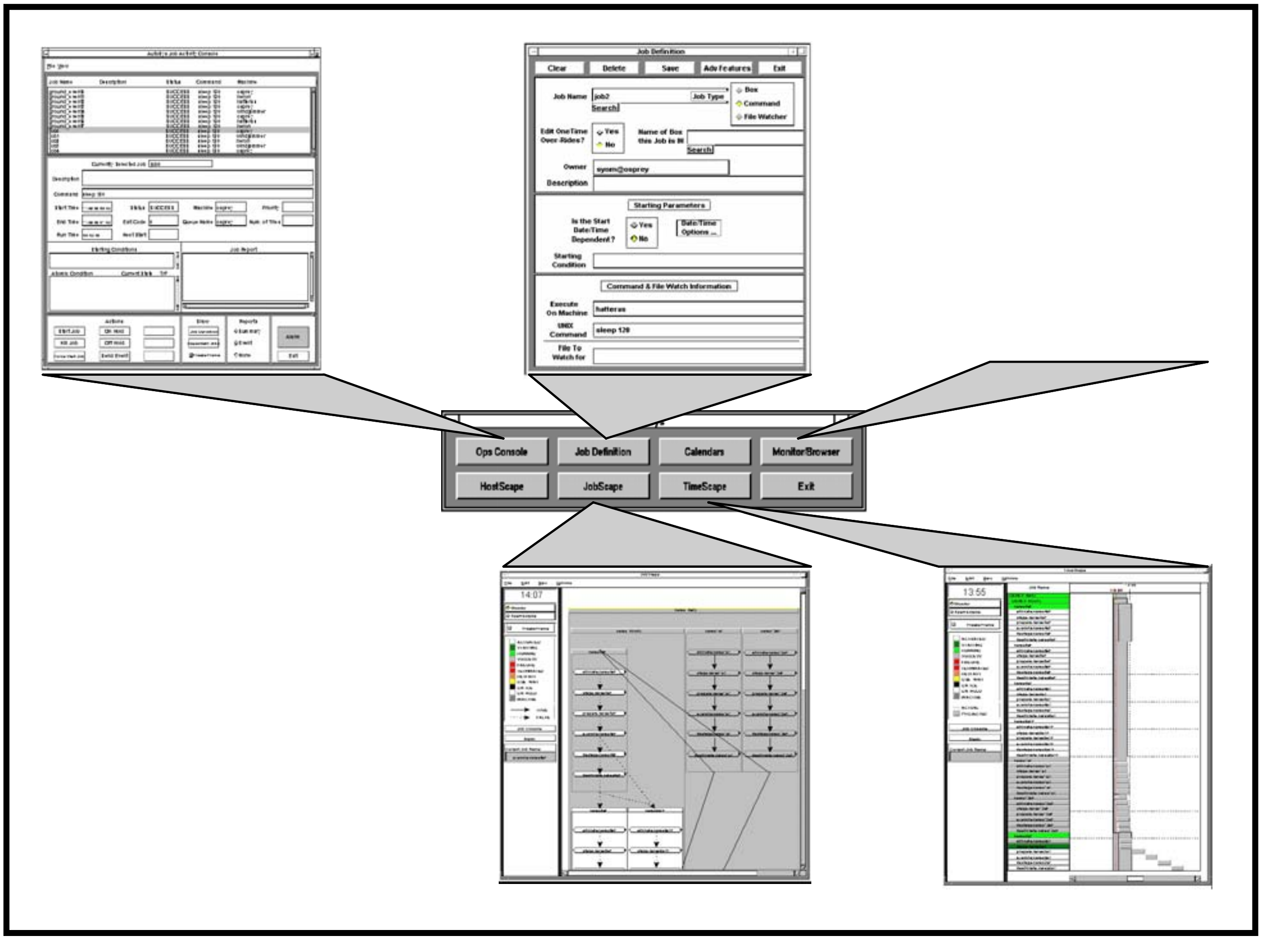

Figure 54. AutoSys GUI Control Panel

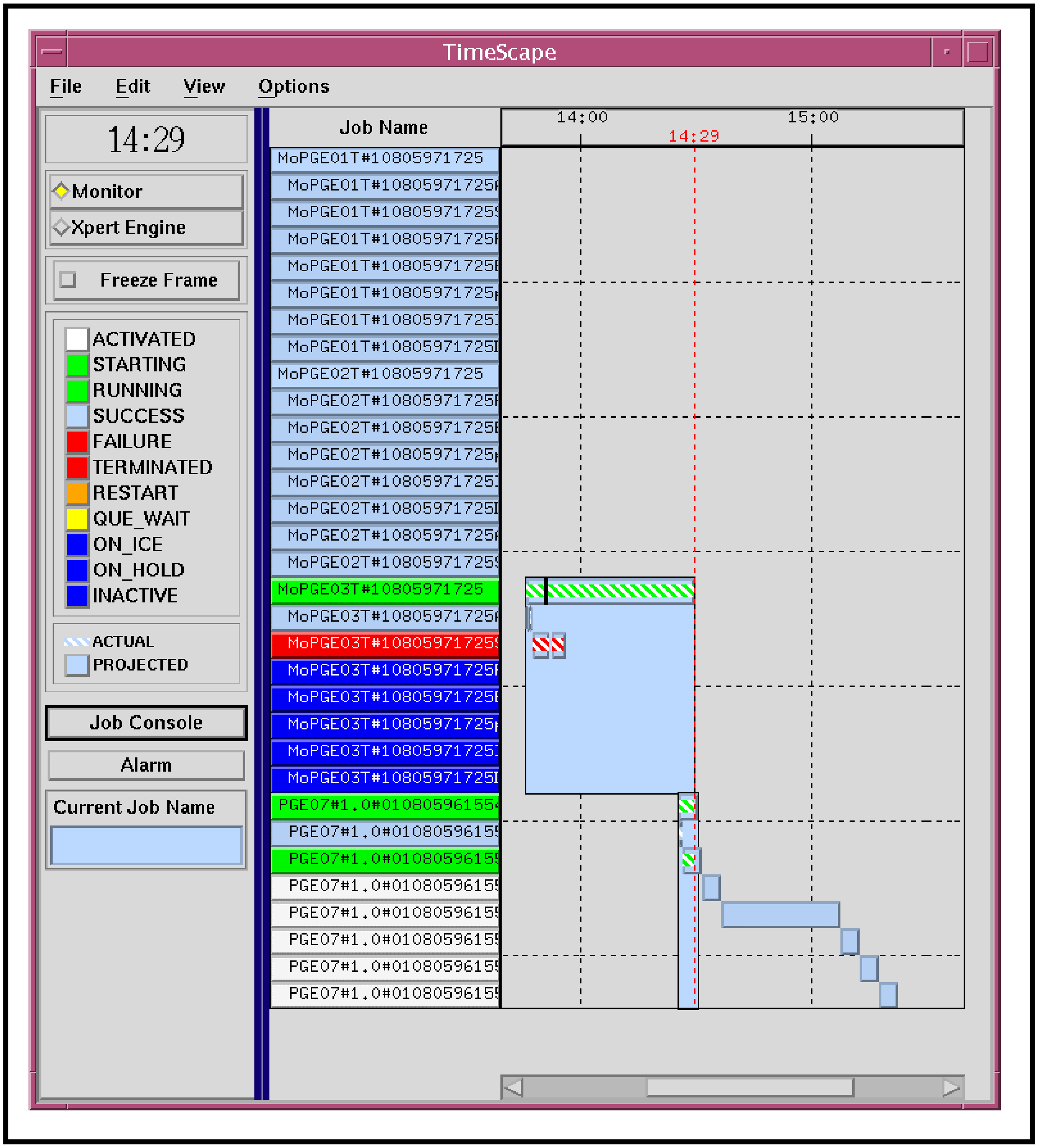

Figure 55. TimeScape Interface

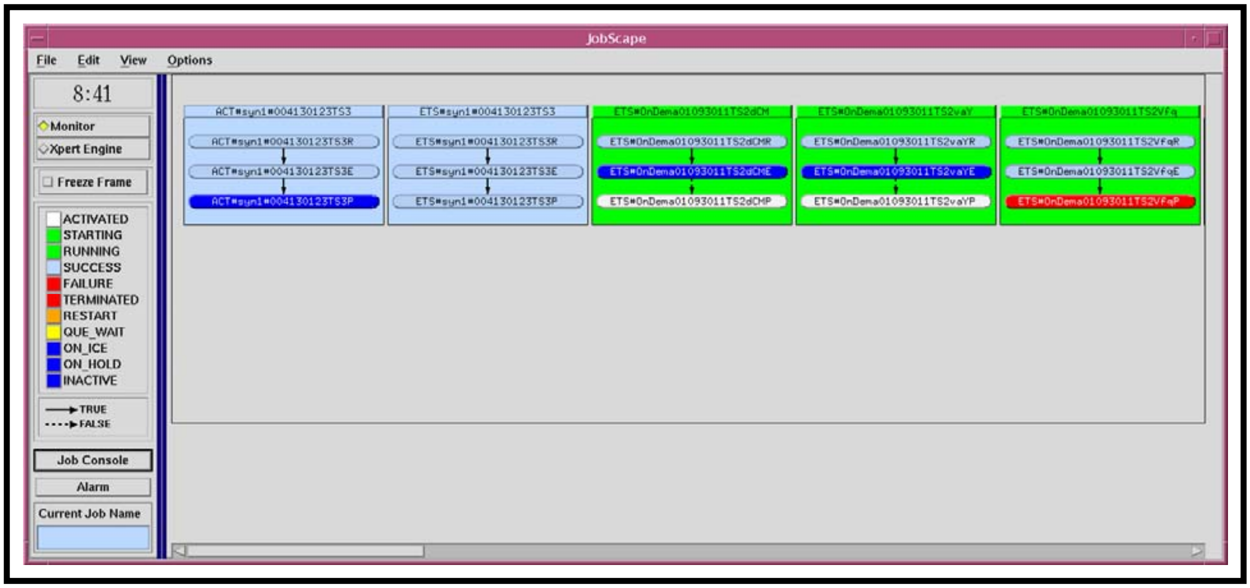

Figure 56. JobScape Interface

- Figure 57 shows the **Runtime Options** dialogue box for **JobScape**.

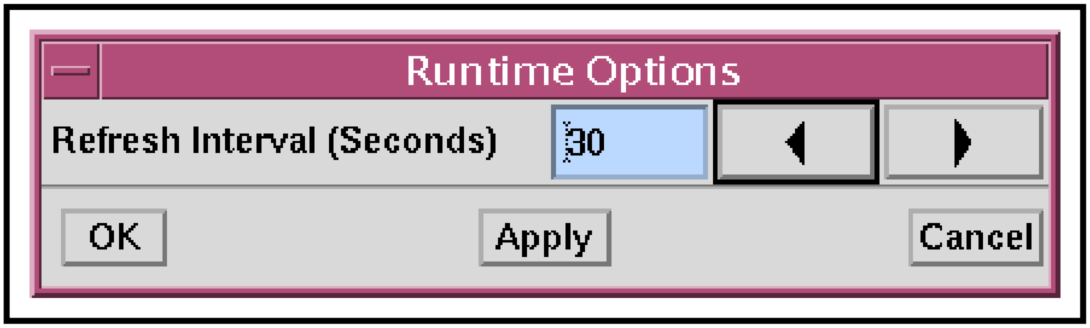

Figure 57. Runtime Options Dialogue Box for JobScape

3 If the refresh interval is to be modified, click on **Refresh Interval (Seconds)** and either type in a value between **1** and **99999** or click on the **<|** and **|>** keys as necessary to decrease or increase the current numerical value until the desired value is reached.

- Default value is **30 seconds**.
- The **Reloading Job Data** window reappears as often as specified in the **Refresh Interval (Seconds)** field.

4 If the number of inches/hour is to be modified, click on **Inches/Hr (inches)** and either type in a value or click the **<|** and **|>** keys as necessary to decrease or increase the current numerical value until the desired value is reached.

- Default value is **2 inches/hr**.

5 When all desired modifications have been entered, click on the **OK** button.

- The runtime options are set.
- The dialogue box closes.

6 If another of the interfaces needs to have its runtime options configured, repeat Steps 2 through 5 for the next interface.

7 To start the process of exiting from either of the interfaces (i.e., **JobScape** or **TimeScape**) select **File** → **Exit** from the pull-down menu.

- An **Exit** dialogue box is displayed.

8 Click on the appropriate button from the following selections:

- **OK** - to exit from the interface.
  – The interface is dismissed.
- **Cancel** - to return to the interface.

9 To start the process of exiting from AutoSys click on the **Exit** button of the **AutoSys GUI Control Panel**.

- An **AutoSys GUIs Exit** dialogue box is displayed.

10 Click on the appropriate button from the following selections:

- **OK** - to exit from the AutoSys GUIs.
  – The **AutoSys GUI Control Panel** is dismissed.
- **Cancel** - to keep the **AutoSys GUI Control Panel**.

Selecting Jobs to Be Shown on AutoSys Displays

This section explains how to select jobs to be shown on AutoSys displays. The Production Monitor can select jobs on the basis of the following criteria:

- Job Name.
- Job Status.
- Machine.

The following default values apply to the job selection criteria until the Production Monitor modifies them:

- All Jobs.
- All Statuses.
- All Machines.

The procedure for selecting jobs to be displayed on AutoSys displays assumes that AutoSys has been launched and the **AutoSys GUI Control Panel** (Figure 53) is being displayed.

Selecting Jobs to Be Displayed on AutoSys Displays

---

1 Click on either the **TimeScape** or **JobScape** button as applicable.

- The interface corresponding to the selected button is displayed (Figure 55 or Figure 56).

2 Select **View → Select Jobs to Display** from the pull-down menu.

- **Job Selection** dialogue box (Figure 58) is displayed.
- **Job selection** has the following default settings:
  - **All Jobs** for **Select by Name**.
  - **All Statuses** for **Select by Status**.
  - **All Machines** for **Select by Machine**.
- If the default settings are the desired settings, proceed to Step 10.

3 To have all jobs displayed verify that the **All Jobs** toggle button is selected (click on the **All Jobs** button to change state from unselected to selected or vice versa).

- When the **All Jobs** option is selected, the **All Jobs** button color is yellow.
- Leave the **Box Hierarchies: Show Number of Levels** set at **all**.
- Proceed to Step 7.

4 To select a particular job or set of jobs by name first verify that the **All Jobs** button is unselected (click on the **All Jobs** button to change state from selected to unselected or vice versa).

5 To select a particular job or set of jobs by name type the appropriate name(s) in the **Name Matching Patterns** fields.

- The asterisk (*) wildcard character can be used for entering a partial job name (e.g., type ***OPS*** to select jobs with “OPS” in their name).

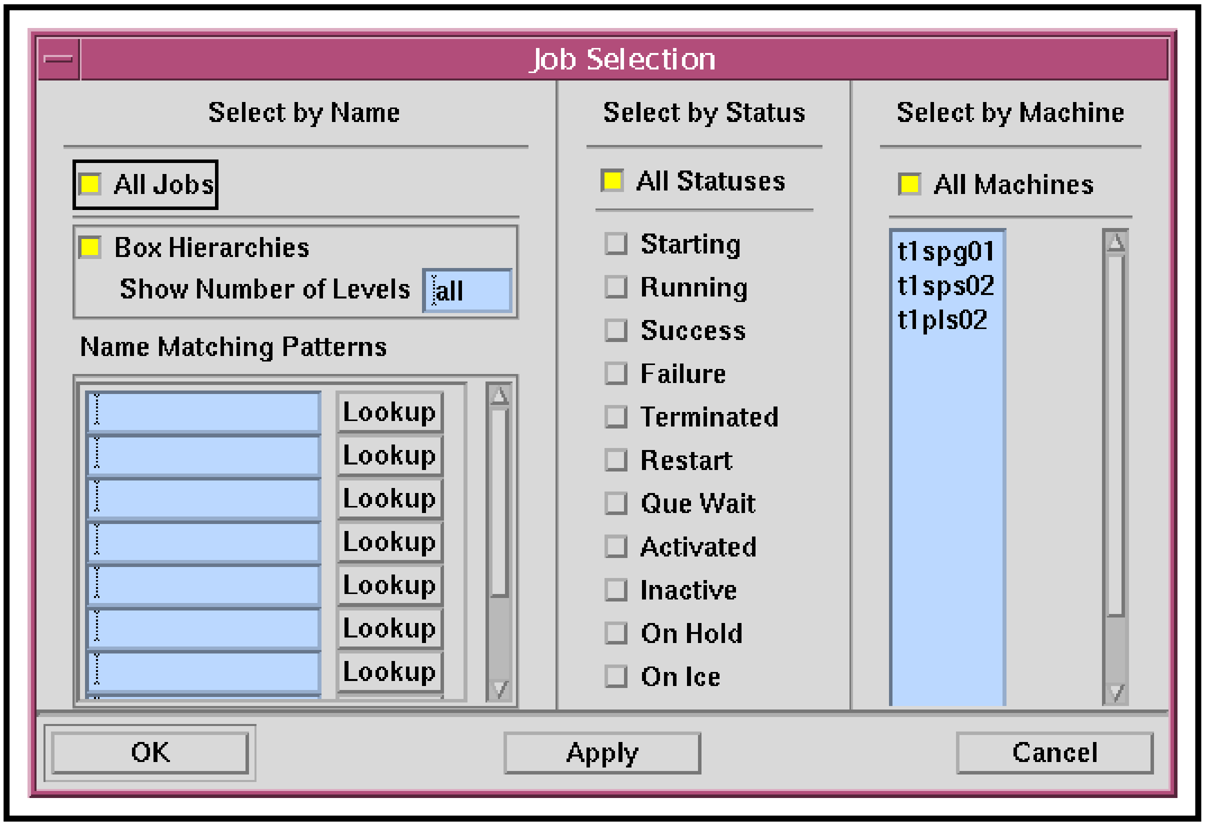

Figure 58. Job Selection Dialogue Box

6 To have jobs displayed regardless of status verify that the **All Statuses** toggle button is selected (click on the **All Statuses** button to change state from unselected to selected or vice versa).

- When the **All Statuses** option is selected, the **All Statuses** button color is yellow.

7 To select jobs by status click on whichever of the following **Select by Status** toggle buttons properly describe(s) the status(es) to be selected:

- **All Statuses.**
- **Starting**.
- **Running**.
- **Success**.
- **Failure**.
- **Terminated**.
- **Restart**.

- **Que Wait**.
- **Activated**.
- **Inactive**.
- **On Hold**.
- **On Ice**.

8 To select jobs on all machines verify that the **All Machines** toggle button is selected (click on the **All Machines** button to change state from unselected to selected or vice versa).

- When the **All Machines** option is selected, the **All Machines** button color is yellow.

9 To select jobs on a particular machine or set of machines click on the name(s) of the desired machine(s) in the **Select by Machine** list.

- To select multiple machines press and hold either the **Ctrl** key or the **Shift** key while clicking on individual machines in the **Select by Machine** list.
- Alternatively, to select multiple machines press and hold either the **Ctrl** key or the **Shift** key then click on the first machine and drag the cursor to the name of the last machine to be selected and release the mouse button.
  - Selected machine(s) is (are) highlighted.

10 When the **Job Selection** dialogue box contains the desired set of job selection criteria, click on the appropriate button from the following selections:

- **OK** - to accept all specified job selection criteria and dismiss the **Job Selection** dialogue box.
  - Original interface is displayed.
  - Jobs are displayed on the interface based on the specified selection criteria.
- **Apply** - to accept all specified job selection criteria without dismissing the **Job Selection** dialogue box.
  - Repeat Steps 3 through 10 as necessary to specify additional job selection criteria.
- **Cancel** - to dismiss the **Job Selection** dialogue box without accepting any job selection criteria.
  - Original interface is displayed

Setting the Current Job on AutoSys Displays

This section explains how to set the “current job” on AutoSys displays. Setting the current job causes the job name to be displayed in the **Current Job Name** field in the Control Region of the display. Subsequently clicking on the **Job Console** button on the display causes the **Job Activity Console** GUI (also known as the **Ops Console** GUI) to be displayed with information concerning the current job.

Either of the following two methods can be used to set the current job:

- Click on the name of a job shown on an AutoSys display.
- Set the current job using the pull-down menu.

The procedure for setting the current job on an AutoSys display using the pull-down menu assumes that AutoSys has been launched and the **AutoSys GUI Control Panel** (Figure 53) is being displayed.

Setting the Current Job on AutoSys Displays Using the Pull-Down Menu

---

1 Click on either the **TimeScape** or **JobScape** button as applicable.

- The interface corresponding to the selected button is displayed (Figures 55-56).

2 Select **View → Set Current Job** from the pull-down menu.

- **Set Current Job** dialogue box (Figure 59) is displayed.

3 Type the ***jobname*** or a portion of the ***jobname*** in the **Filter** field.

- The asterisk (*) wildcard character can be used for entering a partial job name (e.g., type ***AM1*** to list all jobs with “AM1” in their name).

4 Click on the **Filter** button.

- All jobs that meet the criteria specified in the **Filter** field are displayed in the **Jobs** field.

5 Select (highlight) the job to be designated the “current job” from the jobs listed in the **Jobs** field.

- The name of the selected job is displayed in the **Selected Job** field of the **Set Current Job** dialogue box.

6 When the **Selected Job** field of the **Set Current Job** dialogue box contains the desired job name, click on the appropriate button from the following selections:

- **OK -** to accept the selected job and dismiss the **Set Current Job** dialogue box.
  - Selected job is displayed in the **Current Job Name** field in the Control Region of the display.

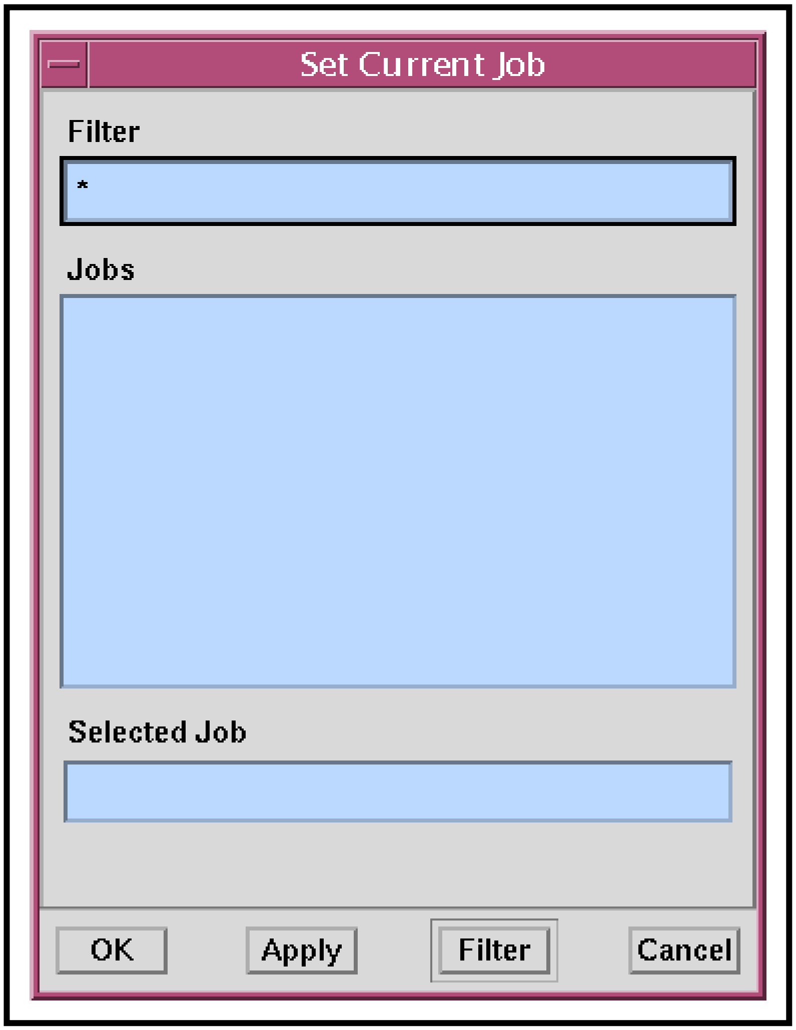

Figure 59. Set Current Job Dialogue Box

  – Original interface is displayed.

- **Apply** - to accept the selected job without dismissing the **Set Current Job** dialogue box.
  – Selected job is displayed in the **Current Job Name** field in the Control Region of the display.
- **Cancel** - to dismiss the **Set Current Job** dialogue box without setting a "current job."
  – Original interface is displayed

Monitoring/Controlling Job Processing

Monitoring/Controlling Job Processing

There are three primary tools for monitoring and controlling job processing:

- **JobScape**.
- **TimeScape**.
- AutoSys **Job Activity Console** GUI (also known as also known as the **Ops Console** GUI).

JobScape presents a Pert-like graphical view of job processing from a logical (or job dependency) point of view. JobScape depicts both command jobs and box jobs. In addition, it depicts the nesting of jobs within boxes and the dependencies among jobs within a box.

JobScape can be used for monitoring and controlling job flow in real-time. It allows the Production Monitor to identify potential problems, try to prevent them from becoming actual problems, put problem jobs on hold in favor of letting good jobs run, restart jobs after correcting problems with them, etc.

TimeScape presents a Gantt-like graphical view of a job processing from a temporal (time-related) point of view. TimeScape depicts both command jobs and box jobs. It also depicts the nesting of jobs within boxes and the duration of time it will take for jobs to complete. TimeScape is used for monitoring job flow in real-time.

AutoSys defines job status in the terms listed in Table 8. The different states are color-coded on JobScape and TimeScape displays. However, the codes can be changed. The color codes listed in the table are the default values.

Table 8. Job States (1 of 2)

Status	Color Code	Meaning
ACTIVATED	white	The top-level box that the job is in is now in the “running” state but the job itself has not started yet.
STARTING	green	The Event Processor has initiated the start procedure with the Remote Agent. The job is in the process of “coming up.”
RUNNING	green	The job is running. If the job is a box job, “running” means that the jobs within the box may be started (other conditions permitting). If the job is a command job, “running” means that the process is actually running on the remote machine.

Table 8. Job States (2 of 2)

Status	Color Code	Meaning
SUCCESS	light blue	When the job had completed running, it had an exit code equal to or less than the "maximum exit code for success." By default, only the exit code "0" is interpreted as "success." However, a range of values up to the "maximum exit code for success" may be reserved for each job to be interpreted as success. If the job is a box job, "success" means that all jobs within the box had exit codes indicating "success" (default) or the "exit condition for box success" was "true."
FAILURE	red	When the job had completed running, it had an exit code greater than the "maximum exit code for success." The default is any non-zero exit code. If the job is a box job, "failure" means that at least one job within the box had an exit code greater than zero (the default meaning) or the "exit condition for box failure" was "true."
TERMINATED	red	The job terminated while in the "running" state. Termination may be the result of an operator sending a "killjob" event, or a job may have been terminated because the job itself (or the box it is in) failed. If the job itself fails, it has a "failure" status rather than a "terminated" status.
RESTART	orange	The job was unable to start due to hardware or application problems and has been scheduled to restart.
QUE_WAIT	yellow	The job can logically run (i.e., all starting conditions have been met); however, there are not enough machine resources available to allow it to run.
ON_ICE	dark blue	The job is removed from all conditions and logic but is still defined to AutoSys. Operationally it is as though the job had been deactivated. The job remains "on_ice" until it receives the "job_off_ice" event. Downstream dependent jobs behave as though the "on_ice" job ran successfully. A job that is "starting" or "running" cannot be put "on_ice."
ON_HOLD	dark blue	The job is on hold and will not run until it receives the "job_off_hold" event. Downstream jobs will not run until the job is taken off hold. A job that is "starting" or "running" cannot be put "on_hold."
INACTIVE	dark blue	The job has not yet been processed. Either the job has never been run or its status was intentionally altered to "turn off" its previous completion status.

The AutoSys **Job Activity Console** GUI (Figure 60) is a text-based interface for monitoring jobs that have been defined for AutoSys. It displays information on the job's start time (and date), end time (and date), run time, status, exit code (if completed), host, priority, and other attributes. It provides a means of evaluating job starting conditions, which can be useful in determining what "upstream" job may be preventing the currently selected job from running. It provides summary and event reports that can be used in identifying problems with processing a particular job.

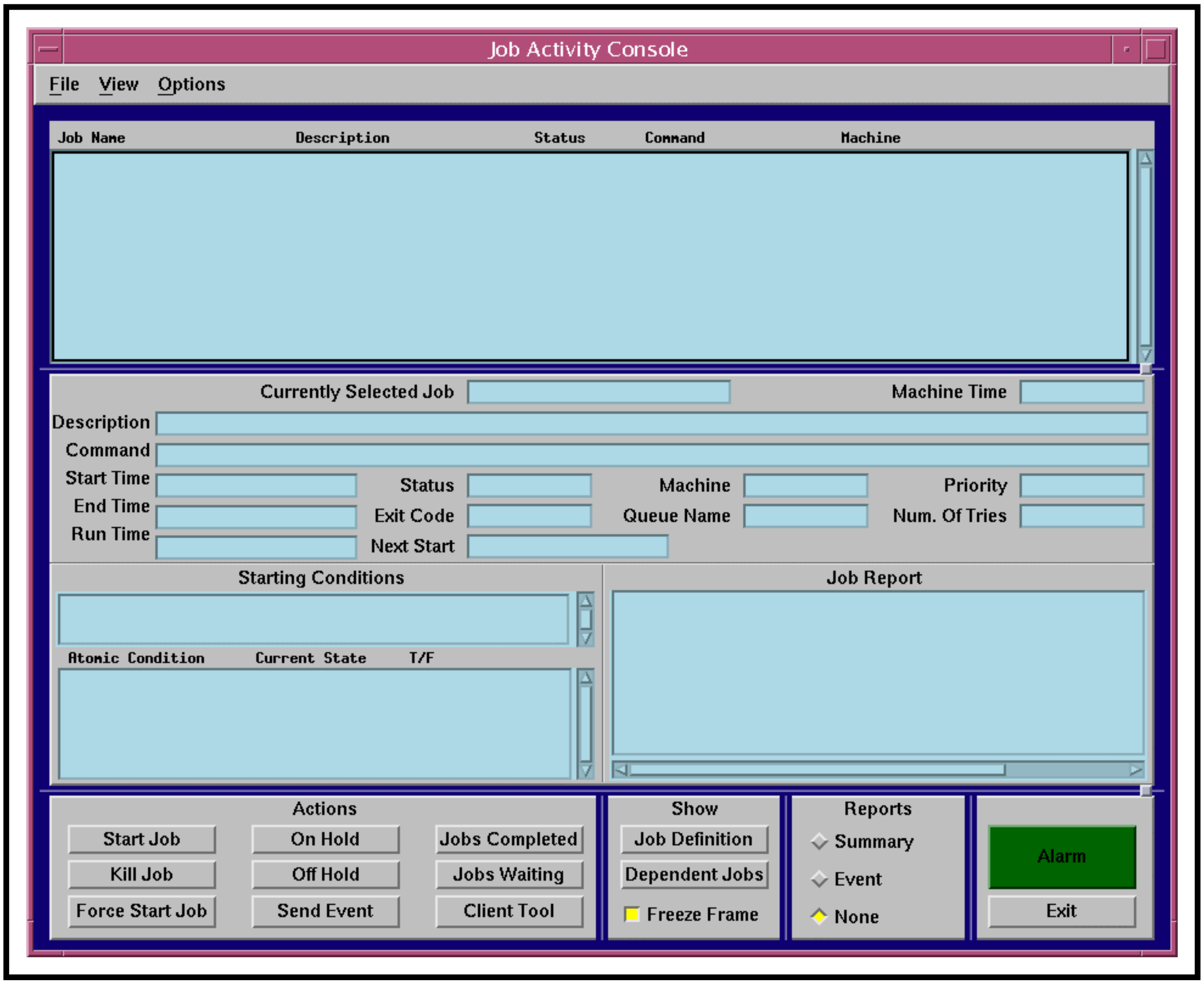

Figure 60. Job Activity Console (Ops Console) GUI

At times the Production Monitor may need to modify a particular job in any of the following ways:

- Start the job.
- Kill the job.
- Force the job to start.
- Place the job on hold.
- Take the job off hold.

The Production Monitor has the option of the following three methods for making those types of modifications to a particular job:

- Buttons in the **Actions** region of the **Job Activity Console (Ops Console)**.
- Menu accessed by clicking the **right** mouse button on the relevant job name on either **JobScape** or **TimeScape**.
- AutoSys **Send Event** GUI.

In AutoSys terms a control action such as starting or killing a job is accomplished by sending an "event" to the job. An event is basically a message. For example, clicking on the **Start Job** button on the AutoSys **Job Activity Console** begins the process by which AutoSys sends a "start" message to the **Currently Selected Job**.

In addition to the previously mentioned modifications to job status, the buttons in the **Actions** region of the **Job Activity Console (Ops Console)** allow the Production Monitor to generate one of the following types of reports:

- Jobs Completed (Figure 61).
- Jobs Waiting (Figure 62).

The menu accessed using the right mouse button on one of the AutoSys displays allows the Production Monitor to initiate either of the following actions (in addition to the previously mentioned modifications to job status):

- Put the job on ice.
- Take the job off ice.

The **Send Event** GUI (Figure 63) allows the Production Monitor to initiate a very broad range of actions, including any of the following items:

- Start the job.
- Kill the job.
- Force the job to start.
- Force the job to start.
- Place the job on hold.
- Take the job off hold.
- Change the job's status.
- Change the job's queue priority.
- Put the job on ice.
- Take the job off ice.

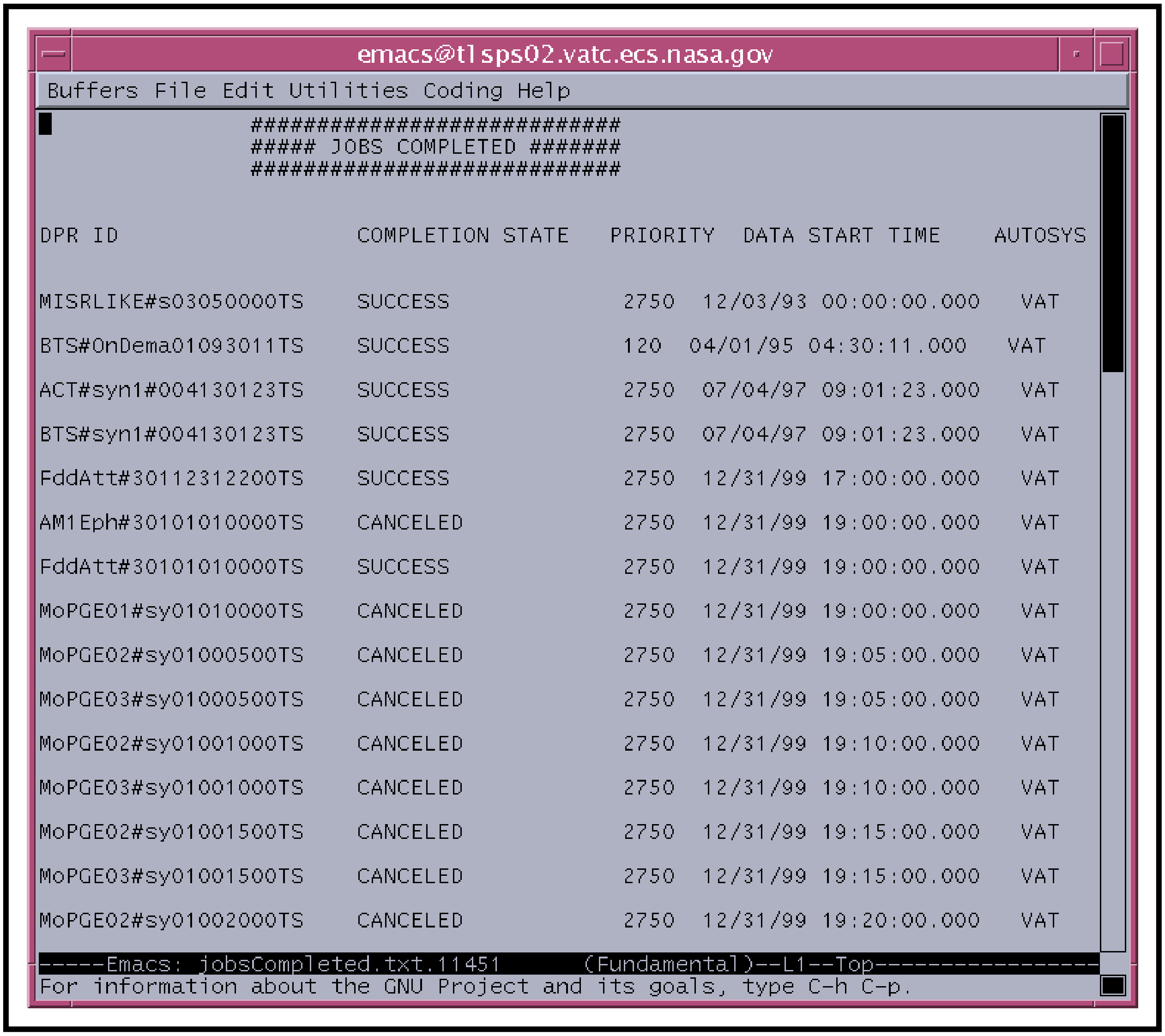

Figure 61. Jobs Completed Report

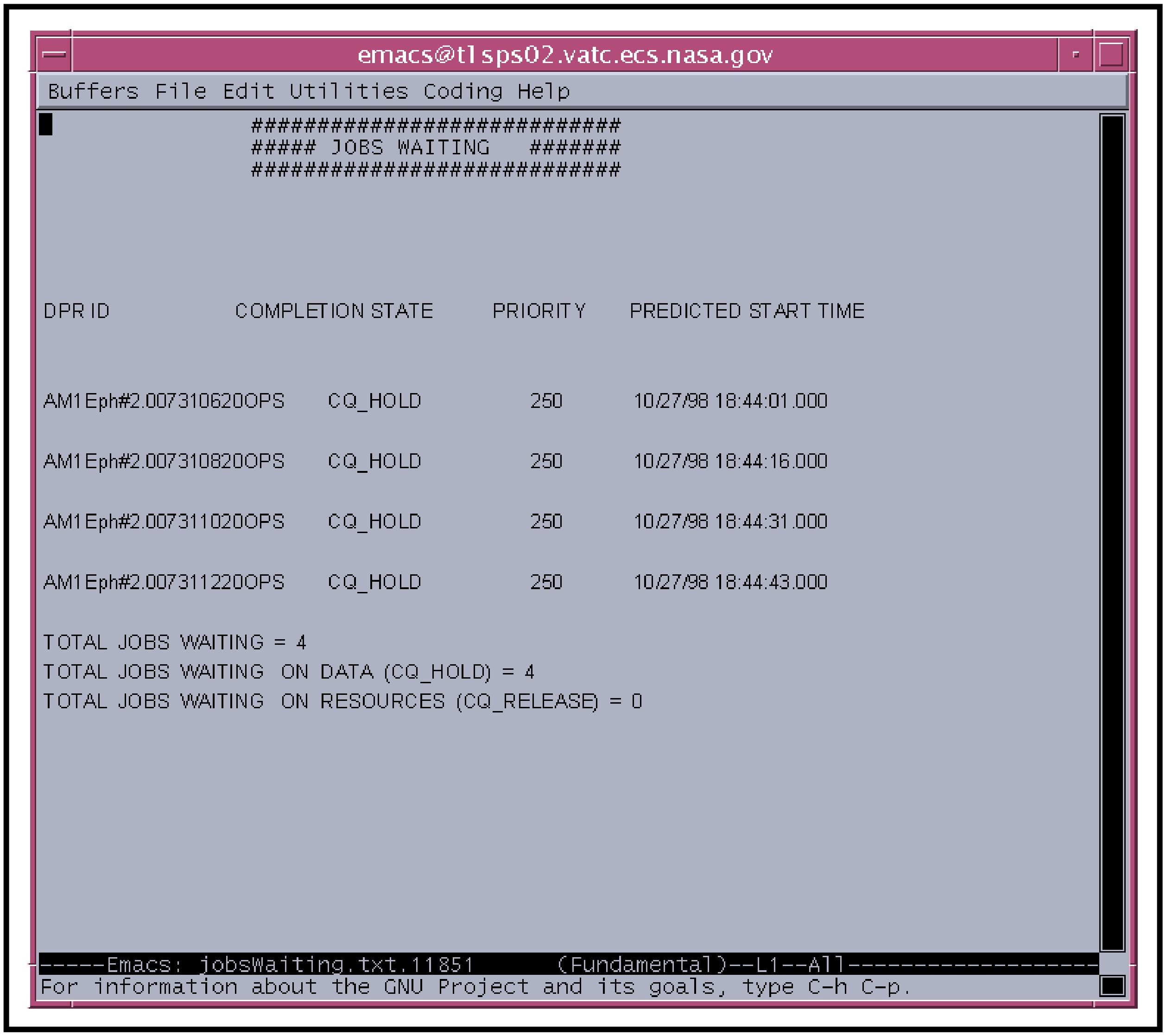

Figure 62. Jobs Waiting Report

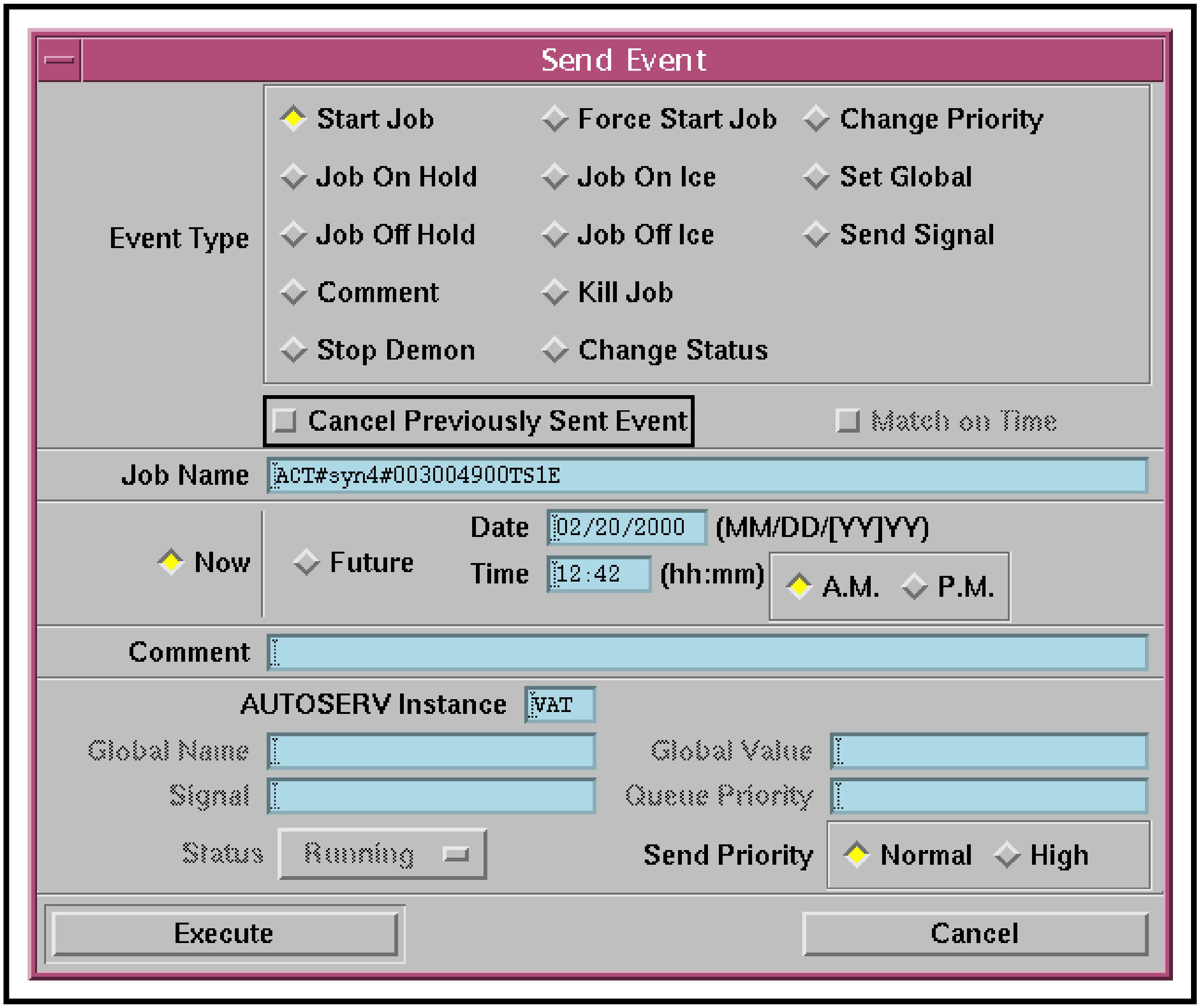

Figure 63. Send Event GUI

- Stop the daemon (stop the Event Processor in an emergency).
- Set a global value.
- Send a signal concerning the job.
- Make a comment (for example, why a job start was forced).

Guidelines for Reporting Unsuccessful Completion of On-Demand Jobs

- Under any of the following circumstances involving an on-demand job notify User Services of the problem in accordance with the applicable local policy:
  - Job is killed.

- Job terminates and cannot be restarted.
- A FAILPGE granule is created.

- The DAAC is obliged to send an e-mail message to the requester of an unsuccessful on-demand job to explain why the request cannot be fulfilled.

Guidelines for Putting Jobs “On Ice” or “On Hold”

- Put jobs on “on hold” rather than “on ice” unless there is a compelling reason to put a job on ice.
- Ensure that the job to be put either “on hold” or “on ice” is not already in a “starting” or “running” state. (A job that is either “starting” or “running” cannot be put “on hold” or “on ice.”)

Guidelines for Force-Starting Jobs

- Force-start command jobs (e.g., preprocessing or postprocessing) only; do not attempt to force-start a box job.
  - The software does not support box job force-starts. (Although it may work fine in some cases, it can cause the PDPS database to get out of sync and prevent the DPR (and possibly other DPRs) from running successfully.)
  - If a box job were force-started, the allocation portion of the preprocessing job would run again. Allocation might choose a different science processor than was chosen the previous time the job ran. Using a different science processor could cause failure of the job.
  - After each job (and often within each job) the state of the DPR is tracked in various tables in the database. Box job force-starts lack the code needed to check the state of the box and perform the cleanup activities necessary for starting over.
- Ensure that the GUI has refreshed and the job to be force-started is not already running before trying to force-start a job. (If a job is already running, it should not be force-started.)
  - It should not be possible to force-start jobs that are already running.
- If any command job other than execution fails, force-start the job that failed only. Do not force start any preceding or succeeding jobs in the box.
- If execution fails, it is not safe to restart it unless the postprocessing job had been put on hold and the failure was detected before postprocessing started running.

- If execution fails and the failure was not detected before postprocessing started running, the DPR must run to completion as a failed PGE and the DPR must be deleted and recreated.

In any case the Production Monitor may implement certain changes of job status only when the Production Monitor "owns" the job affected by the modification.

The procedure for monitoring and controlling job processing starts with the assumption that all applicable servers are currently running and the **AutoSys GUI Control Panel** (Figure 53) is being displayed.

Monitoring/Controlling Job Processing

---

1 To display **JobScape** click on the **JobScape** button on the **AutoSys GUI Control Panel**.

- **JobScape** (Figure 56) is displayed.

2 To display **TimeScape** click on the **TimeScape** button on the **AutoSys GUI Control Panel**.

- **TimeScape** (Figure 55) is displayed.

3 To display the **Job Activity Console** GUI click on the **Job Activity Console** button on the **AutoSys GUI Control Panel**.

- The **Job Activity Console** GUI, also known as the **Ops Console** GUI, (Figure 60) is displayed.

4 Configure runtime options for **JobScape** and/or **TimeScape** as described in the procedure for **Configuring AutoSys Runtime Options** (preceding section of this lesson).

5 If necessary, select jobs to be displayed on **JobScape** and/or **TimeScape** as described in the procedure for **Selecting Jobs to Be Displayed on AutoSys Displays** (preceding section of this lesson).

6 If necessary, set the current job on **JobScape** and/or **TimeScape** as described in the procedure for **Setting the Current Job on AutoSys Displays Using the Pull-Down Menu** (preceding section of this lesson).

7 If necessary, generate a list of jobs to be displayed on the **Job Activity Console** GUI by performing the procedure for **Specifying Job Selection Criteria** (subsequent section of this lesson).

- Job list based on the specified selection criteria is displayed in the **Job List** region of the **Job Activity Console**.

8 Observe information displayed on **JobScape**, **TimeScape**, and/or **Job Activity Console** GUI.

- **JobScape** (Figure 56) presents a Pert-like graphic view of job processing from a logical (or job dependency) point of view.
  - The **Control Region** (left side of display) of **JobScape** has the legend for symbols displayed in the **View Region** (right side of the display).
  - Arrows in the **View Region** of **JobScape** indicate the status of job dependencies:
    - Solid arrow represents **True** (job dependencies have been met).
    - Dashed arrow represents **False** (job dependencies have **not** been met).
    - Dependency arrows indicate only that a job dependency exits for a job; they do not define time-related starting conditions, nor do they describe the type of job dependency; e.g., “success,” “started,” or “running.”
  - Colors in the **View Region** of **JobScape** indicate the status of jobs:
    - Default colors representing job statuses are listed in Table 8.
  - Shapes in the **View Region** of **JobScape** indicate the types of jobs:
    - Rectangle depicts **Box Job**.
    - Ellipse depicts **Command Job**.
- **TimeScape** (Figure 55) presents a Gantt-like graphic view of a job processing from a temporal (time-related) point of view.
  - The **Control Region** (left side of display) of **TimeScape** has the legend for symbols displayed in the **View Region** (right side of the display).
  - Time is listed along the horizontal axis of the **View Region** (right side of the display) of **TimeScape**. (Current time is indicated in red and as a red dashed vertical line.)
  - Jobs are listed along the vertical axis of the **View Region** of **TimeScape**.
  - Bars in the **View Region** of **TimeScape** indicate projected and actual time involved in job processing (refer to Figure 64.):
    - Solid bar represents **Projected** completion time (average job completion time).

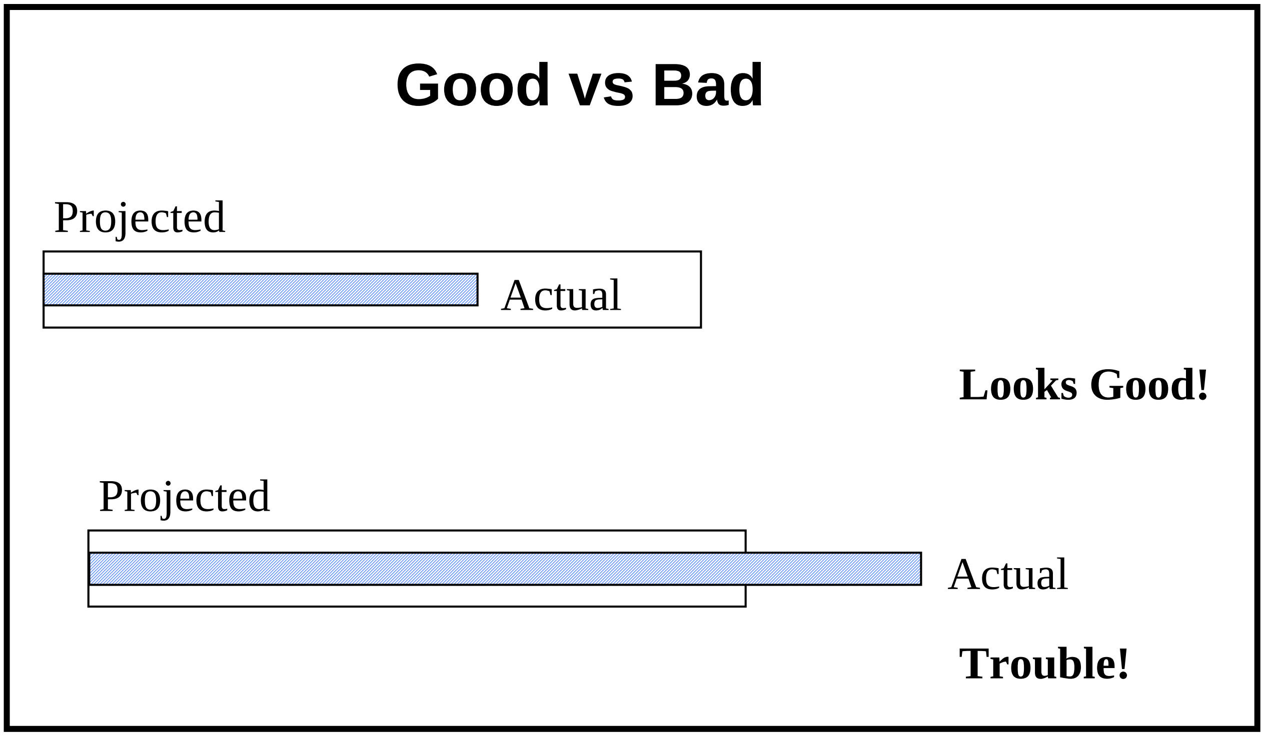

Figure 64. Evaluating Actual versus Projected Job Processing Time

- Striped bar represents **Actual** time taken.

– Colors in the **View Region** indicate the status of jobs:

- Default colors representing job statuses are listed in Table 8.

- The **Job Activity Console** GUI, also known as the **Ops Console** GUI, (Figure 60) provides a text view of each individual job.

– In the **Job List** region of the **Job Activity Console** GUI the following job characteristics are displayed in a table:

- **Job Name.**
- **Description.**
- **Status.**
- **Command.**
- **Machine.**

– The following job details are displayed in the **Currently Selected Job** region of the **Job Activity Console**:

  - ***jobname* (Currently Selected Job)**.
  - **Machine Time** (current time or time at which the frame was frozen)**.**
  - Description.
  - Command.
  - **Start Time** (and date).
  - **End Time** (and date).
  - **Run Time.**
  - **Status.**
  - **Exit Code.**
  - **Next Start.**
  - **Machine.**
  - **Queue Name.**
  - **Priority.**
  - **Num. of Tries.**

– The following job starting conditions are displayed in the **Starting Conditions** region of the **Job Activity Console**:

  - Overall starting conditions (including all atomic conditions).
  - Identification of each **Atomic Condition**.

    Current State.

    T/F (true or false).

  - The starting conditions can be useful in determining what "upstream" job may be preventing the currently selected job from running.
  - An **Atomic Condition** is one of the most basic components of an overall starting condition; for example, if SUCCESS(JOB_X) and SUCCESS(JOB_Y) define the overall starting condition for a job, there are two atomic conditions, one of which is SUCCESS(JOB_X) and the other of which is SUCCESS(JOB_Y).

- The **T/F** (true/false) flag indicates whether the corresponding atomic condition has been satisfied.
- Clicking on one of the **Atomic Conditions** causes the job associated with that condition to become the currently selected job, with its details displayed in the **Currently Selected Job** region of the display. By checking the atomic conditions, it is possible to check the path of upstream dependencies to determine which job (if any) is preventing a particular job from running.
- Figure 65 shows how atomic conditions relate to job dependencies as displayed using JobScape. (In this case JOBOPSR is an atomic condition for JOBOPSE.).
- Note that clicking on one of the atomic conditions listed on the **Job Activity Console** does **not** actually cause **JobScape** to be displayed.

– The following types of reports can be displayed in the **Event Report** region of the **Job Activity Console**:

- **Summary**, which shows the result of the last execution of the job including the following types of information:

  Job Name.

  Last Start.

  Last End.

  Status.

  Pri/Xit.

  Run.

- **Event**, which lists all events from the last execution of the job including the following types of information:

  Status [Event].

  Time.

  Ntry [number of tries].

  EventState [e.g., “Processed”].

  ProcessTime.

  Machine.

- **None.**

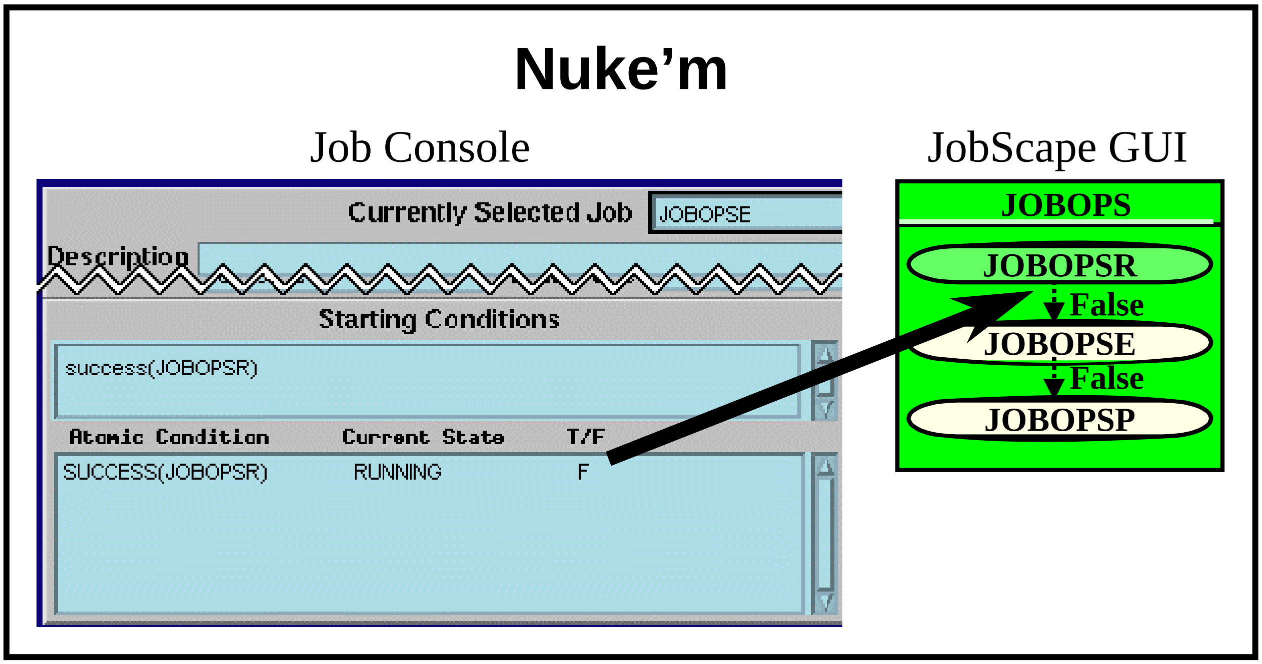

Figure 65. Atomic Conditions and Upstream Dependencies

- Clicking anywhere on a job row in the **Job List** region of the **Job Activity Console** causes detailed information for that job to be displayed in the **Currently Selected Job** region of the display.
- Clicking on a type of report in the **Reports** list of the **Job Activity Console** causes the report to be displayed in the **Event Report** region.
  - The selected report is displayed. The color of the button corresponding to the selected report changes to yellow.
  - For a better view of a report, it is possible to expand the size of the GUI by grabbing a corner of the GUI with the mouse cursor and resizing as desired.
  - An example of a Summary Report is shown in Figure 66.
- The freeze-frame feature prevents the interfaces (**JobScape**, **TimeScape**, or **Job Activity Console** GUI) from being updated, which can disrupt the display.
  - The **Freeze Frame** toggle button is yellow when the freeze-frame feature is activated.
  - To change the state of the freeze-frame feature click on the **Freeze Frame** toggle button.
  - Deactivating the freeze-frame feature allows the display to be updated with new information.

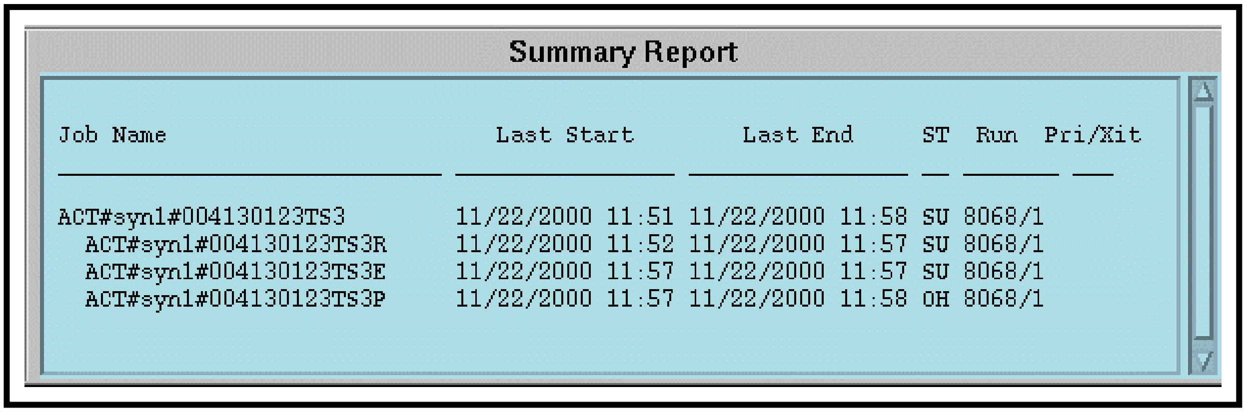

Figure 66. Example of a Summary Report on the Job Activity Console

- Horizontal and vertical scroll bars appear when necessary to allow viewing data that are not readily visible in GUI windows.
- AutoSys GUIs have Alarm buttons that are red when there is an unacknowledged alarm in the alarm list.
  - To display and acknowledge alarms perform the procedure for **Responding to Alarms** (subsequent section of this lesson).

9 If it becomes necessary to perform any of the following actions, go to the corresponding procedure (subsequent sections of this lesson):

- **Determining the Descendants of a Job** [to determine relationships among jobs].
- **Changing the JobScape View Using the Pull-Down Menu** [to change the level of detail displayed for each job shown in the View Region of **JobScape**].
- **Responding to Alarms** [to display and acknowledge alarms involving failures of job processing or other errors in data processing].
- **Determining the Ownership of an AutoSys Job** [to determine which user ID has "edit" privileges and can make changes to the status of a job].
- **Sending an Event to a Job from JobScape or TimeScape** [to modify a particular job using **JobScape** or **TimeScape**].
- **Sending an Event to a Job from the Job Activity Console** [to modify a particular job using the **Job Activity Console**].
- **Sending an Event to a Job from the Send Event GUI** [to modify a particular job using the **Send Event** GUI].
- **Canceling a Sent Event** [to cancel a sent event].

- **Performing Job Management Client Functions** [to perform certain actions using the Job Management Client user interface].
  - Create DPR Job.
  - Release DPR Job.
  - Cancel DPR Job.
  - Change DPR ID.
  - View Job Management DPR Queue.
  - Create Ground Event Job.
  - Cancel Ground Event Job.
  - Change Max Concurrent Jobs for PGE Limits Table.
  - Cancel Max/Min DPRs for Job Class.
  - Trigger Release of Unreleased Ready-to-Run DPRs.
- **Reviewing a Job Activity Report** [to determine job states (e.g., running, completed, or in the AutoSys queue) using the AutoSys **autorep** command].
- **Reviewing a Job Dependency Report** [to determine the current state of a job, its job dependencies, the dependencies and nested hierarchies (for boxes) as specified in the job definition, etc. using the AutoSys **job_depends** command].
- **Running Monitors/Browsers from the Monitor/Browser GUI** [to monitor (using the **Monitor/Browser** GUI) a limited set of AutoSys events or determine the eventual status of jobs run during the preceding shift or day (e.g., which jobs were successful, which jobs failed, and which jobs are still running)].
- **Running Monitors/Browsers Using UNIX Commands** [to monitor (using UNIX commands) a limited set of AutoSys events or determine the eventual status of jobs run during the preceding shift or day (e.g., which jobs were successful, which jobs failed, and which jobs are still running)].

NOTE: When all events for all jobs should be monitored, do *not* run a monitor. Instead, display the Event Processor log in real time (using the command **autosyslog -e**). Running a monitor adds another connection to the database and establishes an additional process that is continually polling the database. That has a significant impact on system performance.

10 Repeat Steps 4 through 9 as necessary to monitor/control jobs.

11 If it becomes necessary to exit from **JobScape** or **TimeScape**, select **File → Exit** from the pull-down menu.

- An exit dialogue box (i.e., **JobScape Exit** or **TimeScape Exit** dialogue box) is displayed.

12 If exiting from the **JobScape** or **TimeScape**, click on the appropriate button from the following selections:

- **OK** - to exit from the interface.
  - The interface is dismissed.
- **Cancel** - to return to the applicable interface.

13 If it becomes necessary to exit from the **Job Activity Console (Ops Console)** GUI, click on the **Exit** button.

- An **AutoSys JAC Exit** dialogue box is displayed to confirm the decision to quit the display.

14 If exiting from the **Job Activity Console (Ops Console)** GUI, click on the appropriate button from the following selections:

- **OK** - to exit from the **Job Activity Console (Ops Console)** GUI.
  - The **Job Activity Console (Ops Console)** GUI is dismissed.
- **Cancel** - to return to the **Job Activity Console (Ops Console)** GUI.

Determining the Descendants of a Job

This section explains how to determine the descendants of a job on either **JobScape** or **TimeScape**. The procedure starts with the assumption that AutoSys has been launched and at least one of the appropriate interfaces (i.e., **JobScape** or **TimeScape**) is being displayed.

Determining the Descendants of a Job

---

1 Select the job by placing the **mouse cursor** on the job and clicking with the **left** mouse button.

- Color of the border around the selected job changes to **yellow**.
- Name of the job appears in the **Current Job Name** area of the Control Region.

2 Place the **mouse cursor** on the job and click and hold the **right** mouse button.

- Pop-up menu appears. It has the following entries:
  – ***jobname*.**
  – **Show Children** [grayed out if not applicable].
  – **Show All Descendants** [grayed out if not applicable].
  – **Hide All Descendants** [grayed out if not applicable].
  – **Show Job Arrows** [**JobScape** only].
  – **Hide Job Arrows** [**JobScape** only].
  – **Show Box Arrows** [**JobScape** only].
  – **Hide Box Arrows** [**JobScape** only].
  – **Job Definition.**
  – **View Dependencies.**
  – **Set Simulation Overrides** [grayed out].
  – **Start Job.**
  – **Kill Job.**
  – **Force Start Job.**
  – **On Hold.**
  – **Off Hold.**
  – **On Ice.**
  – **Off Ice.**

3 If applicable, select (highlight) **Show Children** from the pop-up menu (release the right mouse button).

- Job's first-level Command and Box Jobs appear.
- Repeat Steps 1 and 2 to select a different job.

4 If applicable, select **Show All Descendants** from the pop-up menu.

- Job's Command and Box Jobs appear for all levels.

5 If applicable, select **Hide All Descendants** from the pop-up menu.

- Default view is displayed.

- All descendants are hidden.

Changing the JobScape View Using the Pull-Down Menu

This section explains how to change the view on **JobScape**. Changing the view affects the level of detail displayed for each job shown in the **View Region** of the GUI.

As previously mentioned the view can be changed in some ways by simply clicking with the **right** mouse button on the name of a job shown on an AutoSys display and selecting the desired option from the pop-up menu. The following options related to changing the view and display levels are displayed on the menu:

- **Show Children.**
- **Show All Descendants.**
- **Hide All Descendants.**
- **Show Job Arrows.**
- **Hide Job Arrows.**
- **Show Box Arrows.**
- **Hide Box Arrows.**

Another method for changing the view on **JobScape** involves using the **View** pull-down menu. Many of the same choices plus some additional options can be selected using the pull-down menu.

The procedure for changing the view on **JobScape** using the pull-down menu assumes that AutoSys has been launched and **JobScape** (Figure 56) is being displayed.

Changing the JobScape View Using the Pull-Down Menu

---

1 Select **View** → **Set View** from the pull-down menu.

- The following menu options are displayed:
  - **Normal Text View.**
  - **Small Text View.**
  - **No Text View.**
  - **Show Arrows.**
  - **Hide Arrows.**

– **View by <u>I</u>d.**

– **View by <u>N</u>ame** [grayed out].

2 Select the desired option from the pull-down menu.

- **Normal Text View** is the default view (Figure 56).
- **Small Text View** (Figure 67) is similar to **Normal Text View** but the text and graphics are smaller.
- No text is displayed in the **No Text View** (Figure 68), which provides a global or big-picture view of the jobs currently in processing without specifically identifying them by name.
- **Show Arrows** displays the lines/arrows between jobs.
  – Is characteristic of the default view.
- **Hide Arrows** removes the lines/arrows between jobs from the display.
- **View by Id** changes the display to provide a sequential reference number for each job rather than showing the job name.
- **View by Name** changes the display to show job names rather than reference numbers.
  – Is characteristic of the default view.
  – Is accessible only when the current display is by **Id** number.

3 Select **<u>V</u>iew → Set Display <u>L</u>evels** from the pull-down menu.

- The following menu options are displayed:
  – **<u>1</u>.**
  – **<u>2</u>.**
  – **<u>3</u>.**
  – **<u>4</u>.**
  – **<u>5</u>.**
  – **<u>A</u>ll.**

4 Select the desired option from the pull-down menu.

- **<u>A</u>ll** is the default type of view (Figure 56).
- Selecting **<u>1</u>** provides a display of the box level only, as shown in Figure 69.
- If any other selection (i.e., **<u>2</u>**, **<u>3</u>**, **<u>4</u>**, **<u>5</u>**, or **<u>A</u>ll**) is made, the boxes and command jobs with the boxes are displayed.

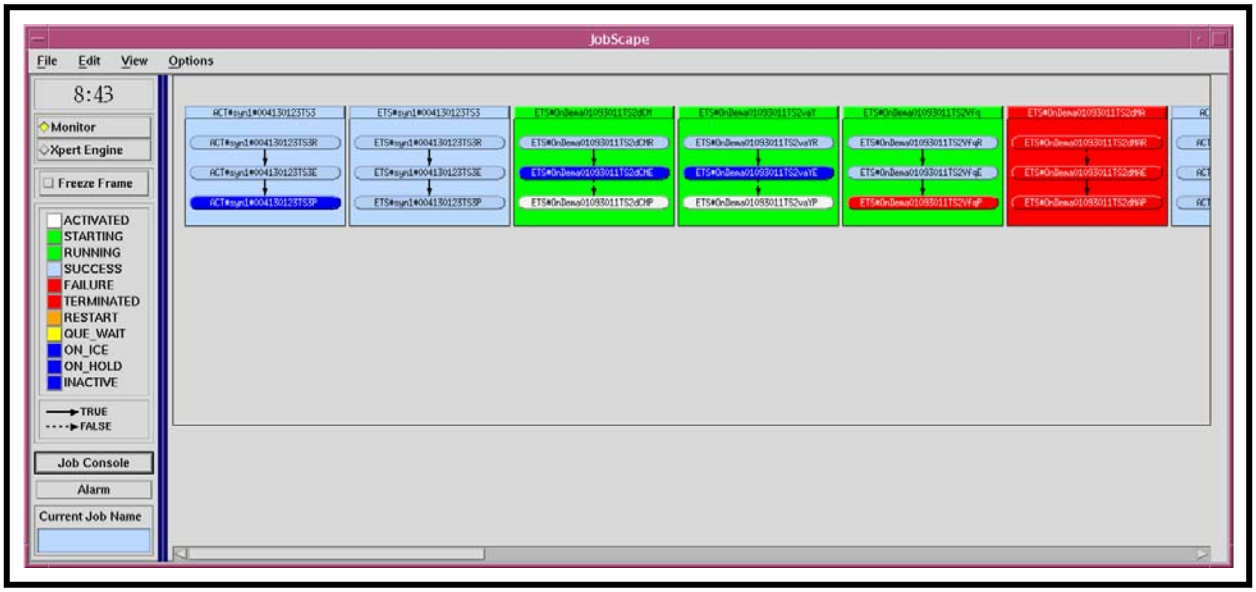

Figure 67. JobScape "Small Text" View

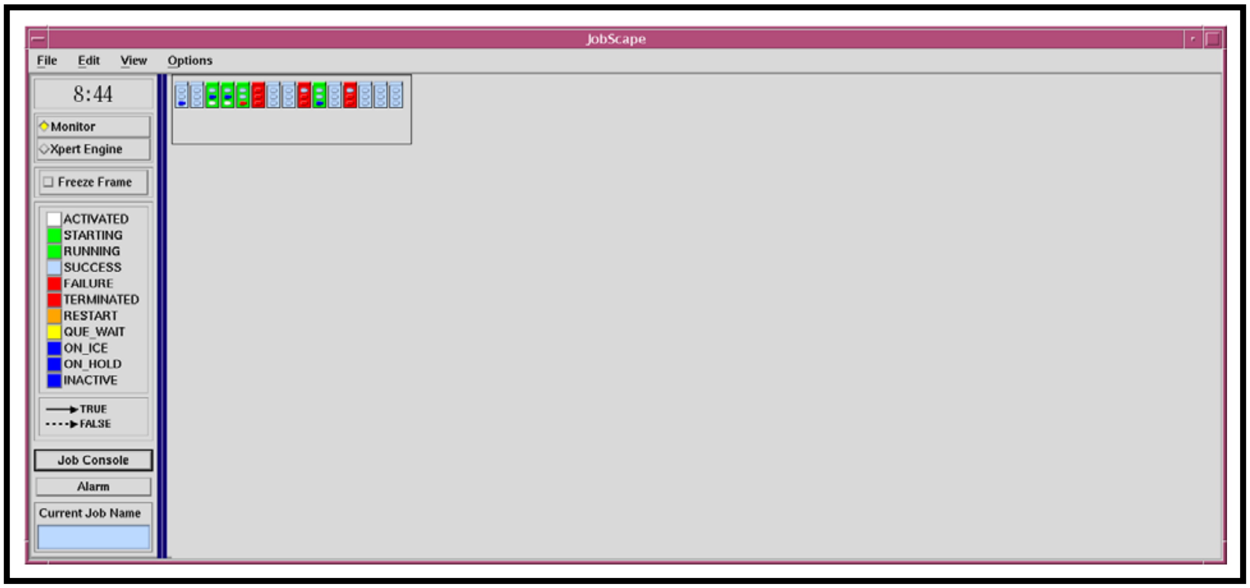

Figure 68. JobScape "No Text" View

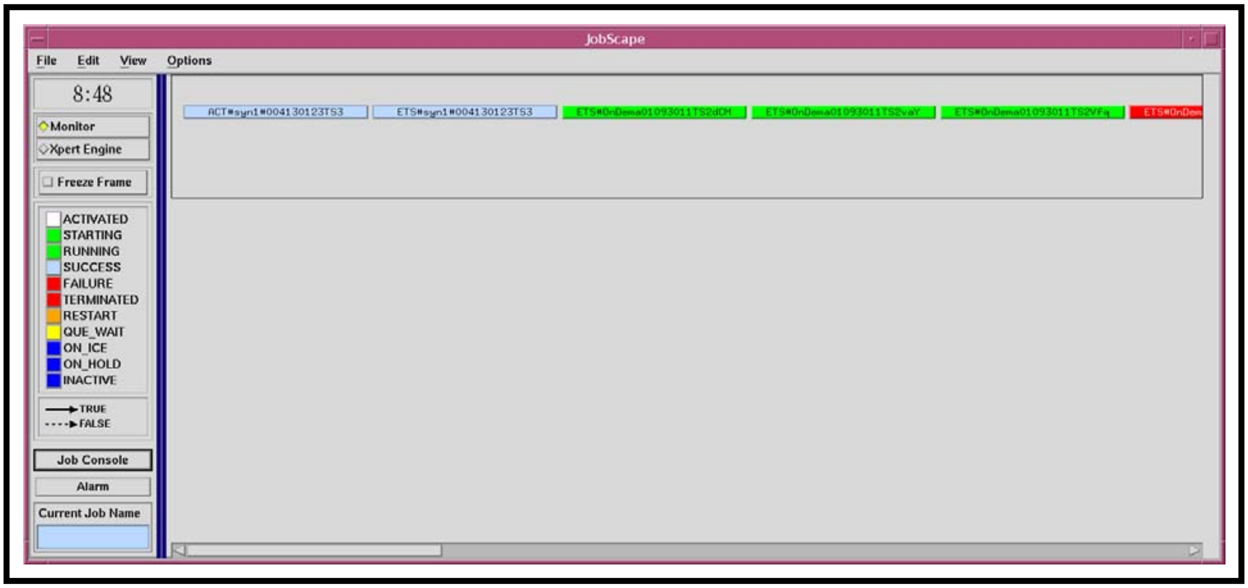

Figure 69. JobScape "One Level" View

Responding to Alarms

Alarms indicate problems with job processing. They may involve a failure of job processing, a database problem, a communication problem, hardware or software failure or some other error in the data processing system.

The Production Monitor responds to alarms using the AutoSys Alarm Manager. The Alarm Manager allows the Production Monitor to perform the following functions:

- View alarms as they arrive.
- Provide a response to an alarm.
- Change alarm status.

The Production Monitor can configure the Alarm Manager to display certain types of alarms only. The Production Monitor may wish to see only certain types of alarms (e.g., job failure alarms) or only those alarms that are open (have not yet been acknowledged) or only the alarms that have occurred within the last thirty minutes.

The Production Monitor can select alarms to be displayed based on any or all of the following three criteria:

- Type of alarm
- Alarm state
- Time of the alarm

The procedure for responding to alarms starts with the assumption that all applicable servers are currently running, and the **AutoSys GUI Control Panel** (Figure 53) and the applicable AutoSys interface(s) (e.g., **JobScape**, **TimeScape**, and/or **Job Activity Console** GUI) is (are) being displayed.

Responding to Alarms

---

1 Click on the **Alarm** button.

- The **Alarm Manager** GUI (Figure 70) is displayed.
- Alarms are displayed in reverse order of occurrence; i.e., the most recent alarm appears at the top of the list.

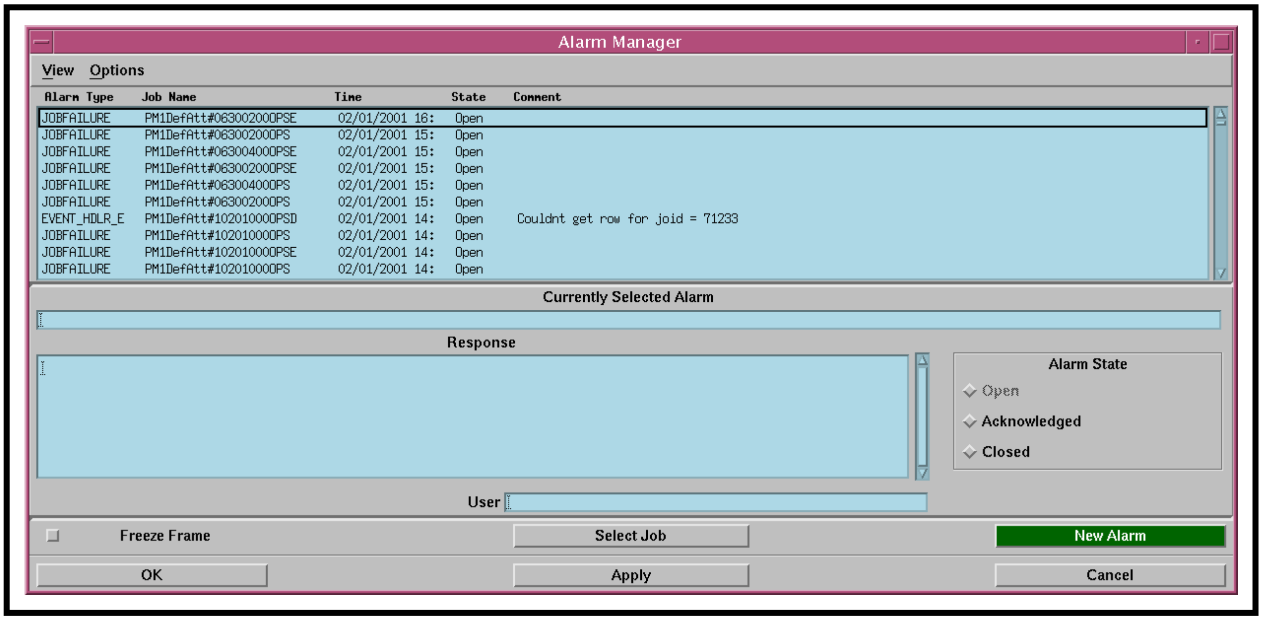

Figure 70. Alarm Manager GUI

- The following information is displayed:
  - **Alarm Type.**
  - **Job Name.**
  - **Time.**
  - **State.**
  - **Comment.**

2 Configure alarm selection as described in the procedure for **Configuring Alarm Selection** (subsequent section of this lesson).

3 If desired, verify that the freeze-frame feature of the **Alarm Manager** GUI is activated (click on the **Freeze Frame** button if necessary).

- The freeze-frame feature prevents the **Alarm Manager** from being updated, disrupting the display.

4 Click on an alarm in the **Alarm List**.

- Alarm is displayed in detail in the **Currently Selected Alarm** region of the display.
- Refer to the *AutoSys® Reference Guide for UNIX* for descriptions of AutoSys alarms.
  - The *AutoSys® Reference Guide for UNIX* and the *AutoSys® User Guide for UNIX* can be downloaded from the Computer Associates Technical Support website but require an account and login. Contact the DAAC COTS software representative for assistance.

5 If a response is to be documented, click the **Response** edit box and type in a response, then press the **Tab** key on the keyboard.

- Response is updated on the GUI (but not yet recorded).

6 Update the **Alarm State** by clicking on the radio button that appropriately describes the **Alarm State**.

- The following **Alarm State** radio buttons are available:
  - **Open.**
  - **Acknowledged.**
  - **Closed.**
- The **Alarm State** is updated on the GUI (but not yet recorded).

7 Click on the appropriate button from the following selections:

- **OK -** to enter all alarm responses and dismiss the **Alarm Manager** GUI.
  - **Alarm Manager** GUI is dismissed.
- **Apply** - to enter all alarm responses without dismissing the **Alarm Manager** GUI.
  - Repeat Steps 4 through 7 as necessary to review and update additional alarms.
- **Cancel** - to return (without entering any alarm responses) to the interface from which the **Alarm Manager** GUI was started.
  - **Alarm Manager** GUI is dismissed.

NOTE: Information concerning a job for which there is/was an alarm can be reviewed by performing the procedure for **Monitoring/Controlling Job Processing** (previous section of this lesson).

By configuring the AutoSys Alarm Manager the Production Monitor can control which alarms are displayed. Alarms can by selected by type, state, or time.

The procedure for configuring the Alarm Manager starts from the assumption that the **Alarm Manager** is currently running.

Configuring Alarm Selection

---

1 Select **View → Select Alarms…** from the pull-down menu of the **Alarm Manager** GUI.

- **Alarm Selection** GUI (Figure 71) is displayed.
- **Alarm Selection** has the following defaults:
  - **All Types** for **Select by Type**.

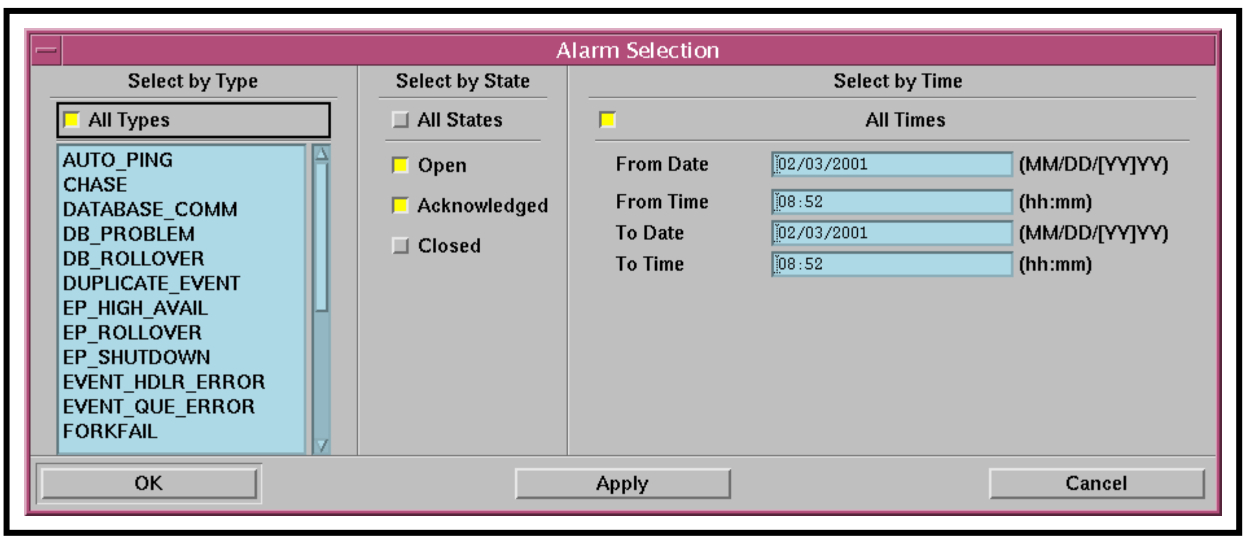

Figure 71. Alarm Selection GUI

  - **Open** and **Acknowledged** for **Select by State**.
  - **All Times** for **Select by Time**.
- If the default settings are the desired settings, proceed to Step 13.

2 If all types of alarms are to be displayed on the **Alarm Manager** GUI, verify that the **All Types** toggle button is selected in the **Select by Type** area.

- Click on the **All Types** button to change state from unselected to selected or vice versa.
  - When the **All Types** option is selected, the **All Types** button color is yellow.
- Proceed to Step 4.

3 If selecting a particular type of alarm or set of alarm types, click on the name(s) of the desired alarm(s) in the **Select by Type** list.

- To select multiple types of alarms press and hold either the **Ctrl** key or the **Shift** key while clicking individual alarms in the **Alarm List**.
- Alternatively, to select multiple types of alarms press and hold either the **Ctrl** key or the **Shift** key, then click on the first type of alarm and drag the cursor to the last type of alarm to be selected and release the mouse button.
  - Selected alarm(s) is (are) highlighted.
- Refer to the *AutoSys® Reference Guide for UNIX* for descriptions of AutoSys alarms.
  - The *AutoSys® Reference Guide for UNIX* and the *AutoSys® User Guide for UNIX*, can be downloaded from the Computer Associates Technical Support website but require an account and login. Contact the DAAC COTS software representative for assistance.

4 If all alarm states are to be displayed on the **Alarm Manager** GUI, verify that the **All States** toggle button is selected in the **Select by State** area.

- Click on the **All States** button to change state from unselected to selected or vice versa.
  - When the **All States** option is selected, the **All States** button color is yellow.
- Proceed to Step 6.

5 If selecting a particular alarm state or set of alarm states to be displayed on the **Alarm Manager** GUI, click on the name(s) of the desired alarm state(s) in the **Select by State** list.

- The following options are available:
  - **Open.**
  - **Acknowledged.**
  - **Closed.**
- Any or all buttons can be selected.

- Button turns yellow when selected.

6 To select alarms by time perform Steps 7 through 12 (as applicable); otherwise, go to Step 13.

7 If alarms at all times are to be displayed on the **Alarm Manager** GUI, verify that the **All Times** toggle button is selected in the **Select by Time** area.

- Click on the **All Times** button to change state from unselected to selected or vice versa.
  - When the **All Times** option is selected, the **All Times** button color is yellow.
- Proceed to Step 13.

8 If selecting a particular date/time range for alarms to be displayed on the **Alarm Manager** GUI, first verify that the **All Times** toggle button is **unselected**.

- Click on the **All Times** button to change state from unselected to selected or vice versa.

9 Type the starting date (in *MM/DD/YYYY* format) in the **From Date** field and press the **Tab** key on the keyboard to advance to the next field.

- Starting date is entered.

10 Type the starting time (in *hh:mm* format) in the **From Time** field and press the **Tab** key on the keyboard to advance to the next field.

- Starting time is entered.

11 Type the end date (in *MM/DD/YYYY* format) in the **To Date** field, and press the **Tab** key on the keyboard to advance to the next field.

- End date is entered.

12 Type the end time (in *hh:mm* format) in the **To Time** field.

- End time is entered.

13 When the **Alarm Selection** GUI contains the desired set of alarm display criteria, click on the appropriate button from the following selections:

- **OK** - to accept all specified alarm selections and dismiss the **Alarm Selection** GUI.
  - **Alarm Manager** GUI is displayed.

- **Apply** - to accept all specified alarm selections without dismissing the **Alarm Selection** GUI.
  - Repeat Steps 2 through 13 as necessary to specify additional alarm selection criteria.
- **Cancel** - to dismiss the **Alarm Selection** GUI without accepting any alarm selections.
  - **Alarm Manager** GUI is displayed.

14 If alarm sound is desired, select **Options → Sound On** from the pull-down menu of the **Alarm Manager** GUI.

- **Sound On** toggle button is yellow when the sound is on.

Specifying Job Selection Criteria

The Production Monitor reviews job activities using the AutoSys **Job Activity Console** as described in the section of the lesson that follows this one. The AutoSys **Job Selection** GUI is used for specifying (filtering) the jobs to be reviewed, including setting the criteria for displaying jobs by name, status and/or machine.

The procedure for specifying job selection criteria starts with the assumption that all applicable servers are currently running and the **Job Activity Console (Ops Console)** GUI, (Figure 60) is being displayed.

Specifying Job Selection Criteria

1 Select **View → Select Jobs** from the pull-down menu on the **Job Activity Console (Ops Console)**.

- The **Job Selection** GUI (Figure 72) is displayed.
- **Job Selection** has the following default values:
  - **All Jobs/Job Name** for **Select by Name**.
  - **All Statuses** for **Select by Status**.
  - **All Machines** for **Select by Machine**.
  - **Unsorted** for **Sort Order**.
- If the default settings are the desired settings, proceed to Step 9.

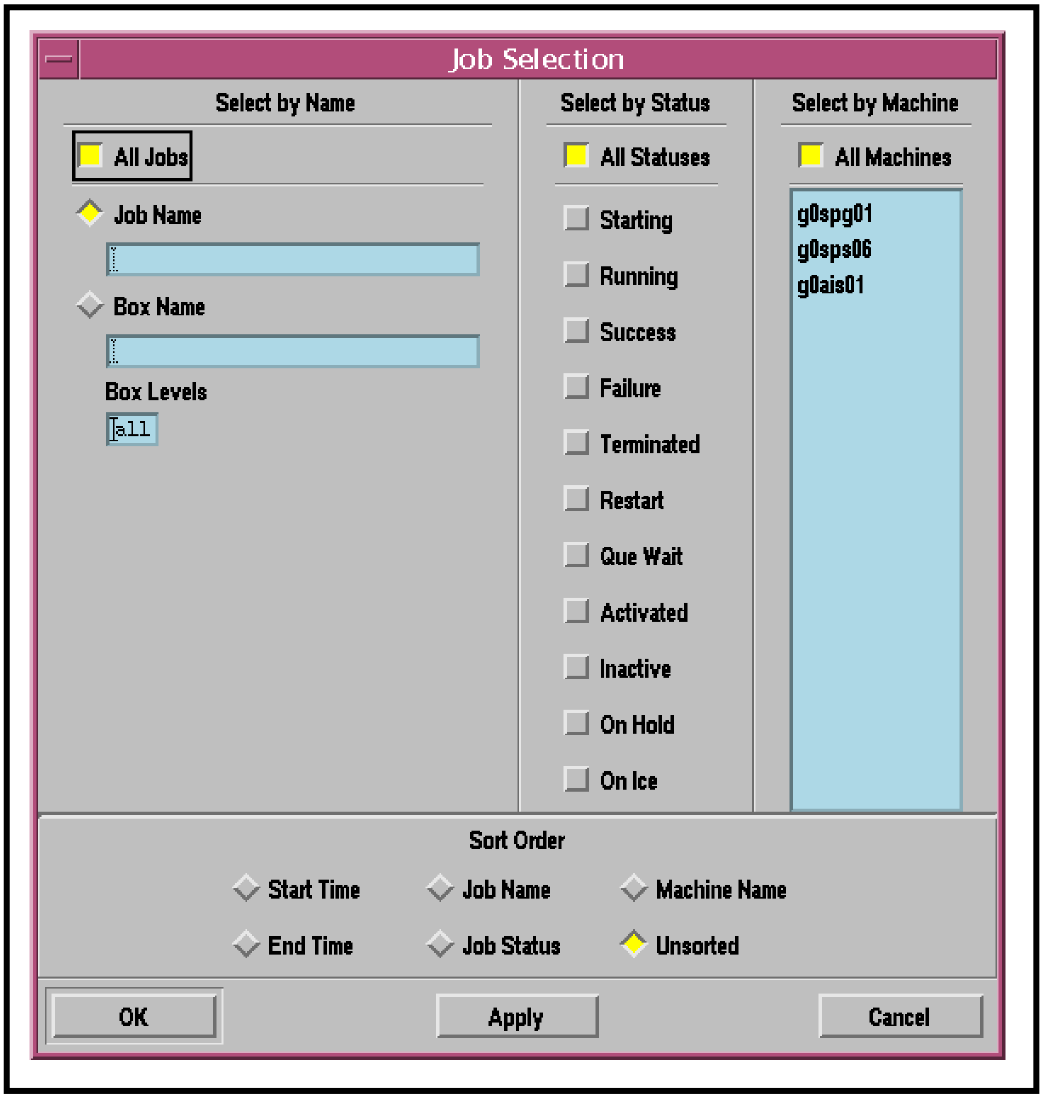

Figure 72. Job Selection GUI

2 To select all jobs click on the **All Jobs** button.

- When the **All Jobs** option is selected, the **All Jobs** button color changes to yellow.

3 To select a particular job by name, type the name of the desired job in the **Job Name** field.

- When typing in either the **Job Name** field or the **Box Name** field, the corresponding toggle button is automatically turned on. (You do not have to click on the button, just start typing in the desired field.)
- The asterisk (*) wildcard character can be used for entering a partial job or box name (e.g., *AST*).

4 To select a particular box by name, type the name of the desired box in the **Box Name** field then type in the **Box Levels** field how many levels of nesting you want to view for the box job.

- In the **Box Levels** field any valid positive integer can be entered or the word "all."
  - 0 - indicates that only the top-level box specified in the **Box Name** field is to be displayed.
  - 1 - indicates that the specified top-level box and all direct descendant boxes and enclosed jobs are to be displayed.
  - all - indicates that all jobs in the box are to be displayed.

5 To select jobs by status click on whichever of the following **Select by Status** toggle buttons properly describe(s) the status(es) to be selected:

- **All Statuses.**
- **Starting**.
- **Running**.
- **Success**.
- **Failure**.
- **Terminated**.
- **Restart**.
- **Que Wait**.
- **Activated**.
- **Inactive**.
- **On Hold**.
- **On Ice**.

6 To select all machines click on the **All Machines** button.

- **All Machines** button turns yellow.
- Proceed to Step 8.

7 To select jobs by the particular machine click on the name of the desired machine in the **Select by Machine** list.

- To select multiple machines press and hold either the **Ctrl** key or the **Shift** key while clicking on individual machines in the **Select by Machine** list.

- Alternatively, to select multiple machines press and hold either the **Ctrl** key or the **Shift** key then click on the first machine and drag the cursor to the name of the last machine to be selected and release the mouse button.
  - Selected machine(s) is (are) highlighted.

8 Click on whichever of the following **Sort Order** toggle buttons properly describes the **Sort Order** to be selected:

- **Start Time.**
- **End Time.**
- **Job Name.**
- **Job Status.**
- **Machine Name.**
- **Unsorted.**

9 When the **Job Selection** GUI contains the desired set of job selection criteria, click on the appropriate button from the following selections:

- **OK** - to accept all specified job selection criteria and dismiss the **Job Selection** GUI.
  - **Job Activity Console** GUI is displayed.
  - Job list based on the specified selection criteria is displayed in the **Job List** region of the **Job Activity Console**.
- **Apply** - to accept all specified job selection criteria without dismissing the **Job Selection** GUI.
  - Repeat Steps 2 through 9 as necessary to specify additional job selection criteria.
- **Cancel** - to dismiss the **Job Selection** GUI without accepting any job selection criteria.
  - **Job Activity Console** GUI is displayed.

Determining the Ownership of an AutoSys Job

AutoSys is very much ownership-aware. Only the “owner” of a job has “edit” privileges and can make changes to the status of an owned job.

AutoSys recognizes ownership in terms of two factors:

- UserID.
- Machine where the operator (user) logged in.

For example, **cmshared@e0sps04** identifies the Production Monitor who logged in as "cmshared" at machine e0sps04. Any operator who logs in as "cmshared" at another machine (e.g., e0pls11) would not be able to change the status of a job "owned" by **cmshared@e0sps04**. Consequently, to have any real effect on a job first it is necessary to log in as the job's owner and launch the AutoSys GUIs as that owner.

The procedure for determining the ownership of a job starts with the assumption that all applicable servers are currently running and at least one of the appropriate GUIs (i.e., **JobScape** or **TimeScape**) is being displayed.

Determining the Ownership of an AutoSys Job

---

1 Place the mouse cursor on the relevant job (on **JobScape** or **TimeScape**) and click and hold the **right** mouse button.

- Pop-up menu appears. It has the following entries:
  - ***jobname*.**
  - **Show Children** [grayed out if not applicable].
  - **Show All Descendants** [grayed out if not applicable].
  - **Hide All Descendants** [grayed out if not applicable].
  - **Show Job Arrows** [**JobScape** only].
  - **Hide Job Arrows** [**JobScape** only].
  - **Show Box Arrows** [**JobScape** only].
  - **Hide Box Arrows** [**JobScape** only].
  - **Job Definition.**
  - **View Dependencies.**
  - **Set Simulation Overrides** [grayed out].
  - **Start Job.**
  - **Kill Job.**
  - **Force Start Job.**

- **On Hold.**
- **Off Hold.**
- **On Ice.**
- **Off Ice.**

2 Select (highlight) **Job Definition** from the pop-up menu (release the right mouse button).

- The **Job Definition** GUI (Figure 73) is displayed.
- If the current UserID does not "own" (have edit permissions on) the job, a **Job Security MESSAGE** window (Figure 74) is displayed.

3 If a **Job Security MESSAGE** window (Figure 74) is displayed, click on the Ok button.

- The **Job Security MESSAGE** window is dismissed.

4 Review the entry in the **Owner** field of the **Job Definition** GUI.

- Job owner is identified in the **Owner** field of the **Job Definition** GUI.
- Job name is listed in the **Job Name** field of the **Job Definition** GUI.

NOTE: Jobs should **not** be deleted using the AutoSys **Job Definition** GUI because it does not communicate with the PDPS database.

5 To exit from the **Job Definition** GUI, click on the **Exit** button.

Sending an Event to a Job

As previously mentioned there are three methods for making certain types of modifications (e.g., start or kill) to a particular job:

- Menu accessed by clicking the **right** mouse button on the relevant job name on either **JobScape** or **TimeScape**.
- Buttons in the **Actions** region of the **Job Activity Console (Ops Console)**.
- AutoSys **Send Event** GUI.

Sending an Event to a Job from JobScape or TimeScape

The procedure for sending an event to a job from **JobScape** or **TimeScape** starts with the assumption that all applicable servers are currently running and at least one of the appropriate interfaces is being displayed.

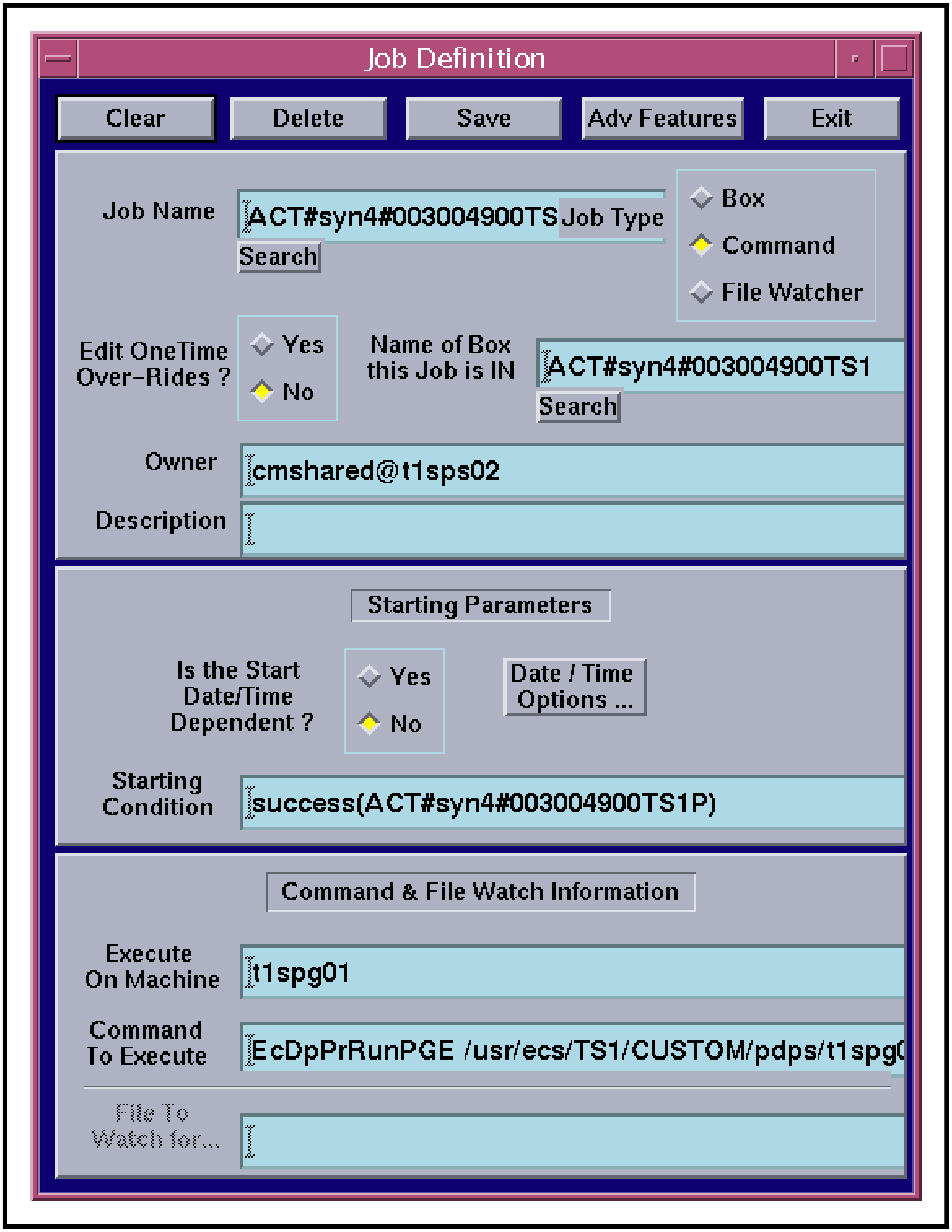

Figure 73. Job Definition GUI

Figure 74. Job Security MESSAGE Window

Sending an Event to a Job from JobScape or TimeScape

---

1 Place the mouse cursor on the relevant job (on **JobScape** or **TimeScape**) and **click** and **hold** the **right** mouse button.

- Pop-up menu appears.
- Of the options displayed on the pop-up menu the following choices are relevant:
  - **Start Job.**
  - **Kill Job.**
  - **Force Start Job.**
  - **On Hold** (Put job on hold).
  - **Off Hold** (Take job off hold).
  - **On Ice** (Put job on ice).
  - **Off Ice** (Take job off ice).

2 Select the event (e.g., **Force Start Job**) to be sent to the job from the pop-up menu (release the right mouse button).

- A confirmation dialogue box similar to Figure 75 is displayed requesting permission to proceed with sending the event.
- If there is no option corresponding to the desired action, modify job status from either the **Job Activity Console** or the **Send Event** GUI as described in the procedure for **Sending an Event to a Job from the Job Activity Console** or the procedure for **Sending an Event to a Job from the Send Event GUI** (subsequent sections of this lesson).

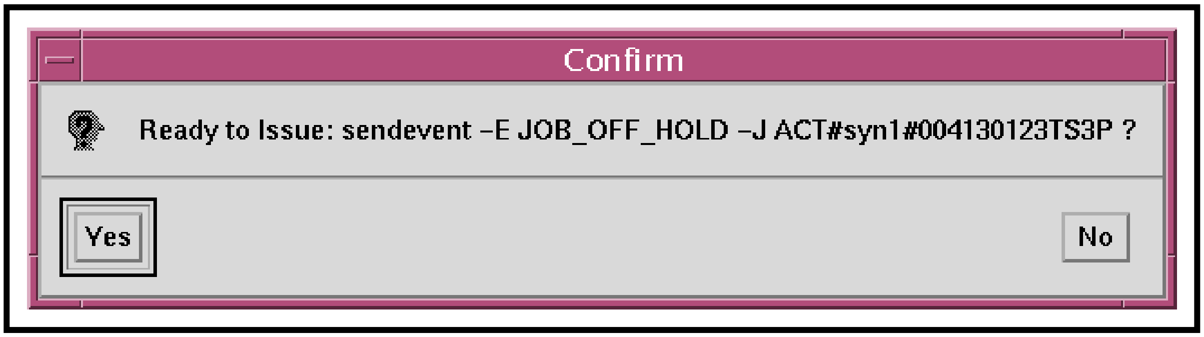

Figure 75. Confirmation Dialogue Box

3 Click on the appropriate button from the following selections:

- **Yes** - to send the event to the job.
  - – The confirmation dialogue box is dismissed.
  - – The specified action is taken.
- **No –** to dismiss the confirmation dialogue box without sending the event to the job.

Sending an Event to a Job from the Job Activity Console

The procedure for sending an event to a job from the **Job Activity Console (Ops Console)** starts with the assumption that all applicable servers are currently running and the **Job Activity Console (Ops Console)** (Figure 60) is being displayed.

Sending an Event to a Job from the Job Activity Console

---

1 Specify job selection criteria for the AutoSys **Job Activity Console** by performing the procedure for **Specifying Job Selection Criteria** (preceding section of this lesson)

2 Verify that the job with the status to be modified is listed in the **Currently Selected Job** field of the **Job Activity Console** (**Ops Console**).

- Click on the job row in the **Job List** region of the **Job Activity Console** if necessary.
  - – Information concerning the selected job is displayed in the **Currently Selected Job** region of the **Job Activity Console**.

3 Click on the button corresponding to the desired action to be taken with respect to the selected job (if there is a corresponding button in the **Actions** region of the **Job Activity Console**).

- The following options are displayed:
  - **Start Job.**
  - **Kill Job.**
  - **Force Start Job.**
  - **On Hold** (Put job on hold).
  - **Off Hold** (Take job off hold).
- A confirmation dialogue box similar to Figure 75 is displayed requesting permission to proceed with sending the event.
- If there is no button corresponding to the desired action, modify job status using the **Send Event** GUI as described in the procedure for **Sending an Event to a Job from the Send Event GUI** (subsequent section of this lesson).

4 Click on the appropriate button from the following selections:

- **Yes** - to send the event to the job.
  - The confirmation dialogue box is dismissed.
  - The specified action is taken.
- **No** – to dismiss the confirmation dialogue box without sending the event to the job.

Sending an Event to a Job from the Send Event GUI

The procedure for sending an event to a job using the **Send Event** GUI starts with the assumption that all applicable servers are currently running and the **Job Activity Console (Ops Console)** (Figure 60) is being displayed.

CAUTION

Once an event has been sent from the **Send Event** dialogue, it may not be possible to cancel or modify it.

Sending an Event to a Job from the Send Event GUI

1 Specify job selection criteria for the AutoSys **Job Activity Console** by performing the procedure for **Specifying Job Selection Criteria** (preceding section of this lesson)

2 In the **Job List** region of the **Job Activity Console** click on the job row corresponding to the job with the status to be modified.

- Information concerning the selected job is displayed in the **Currently Selected Job** region of the **Job Activity Console**.

3 Click on the **Send Event** button in the **Actions** Region of the **Job Activity Console**.

- The **Send Event** GUI (Figure 63) is displayed.
- **Send Event** has the following defaults:
  - **Start Job** for **Event Type**.
  - **Now** for **Time**.
  - **Normal** for **Send Priority**.
- If the default settings are the desired settings, proceed to Step 18.

4 Verify that the correct job is listed in the **Job Name** field of the **Send Event** GUI.

- If not, click on the **Cancel** button and select the correct job (return to Step 2).

5 Click on the **Event Type** to be sent to the job in AutoSys.

- The following options are displayed:
  - **Start Job.**
  - **Job On Hold.**
  - **Job Off Hold.**
  - **Comment.**
  - **Stop Demon.**
  - **Force Start Job.**
  - **Job On Ice.**
  - **Job Off Ice.**
  - **Kill Job.**

- **Change Status.**
- **Change Priority.**
- **Set Global.**
- **Set Signal.**

- Remember that a job with status of either “starting” or “running” cannot be put “on hold” or “on ice.”
- Note that the GUI has an option to **Cancel Previously Sent Event**.

6 To select a future time for sending the event to the job click on the **Future** button.

- If **Now** (the default value) is desired, proceed to Step 10.
  - Current date and time are default values.

7 Type the date (in *MM/DD/YYYY* format) for future execution in the **Date** field.

8 Type the time (in *hh:mm* format) for future execution in the **Time** field.

9 Click on either the **A.M.** or **P.M.** button as applicable.

10 If **Comment** was selected as the **Event Type**, type the appropriate comment in the **Comment** field.

- **Comment** is a free-form field for entering text to be sent to the specified job.

11 Verify that the entry in the **AUTOSERV Instance** field is the current AutoSys instance identifier.

- Type the correct **AUTOSERV Instance** in the **AUTOSERV Instance** field if necessary.
- **AUTOSERV Instance** field specifies the instance of AutoSys to which the event will be sent. (You can send events to instances of AutoSys other than the one you are running.)
- The current AutoSys instance should be displayed by default in the **AUTOSERV Instance** field.

12 If **Set Global** was selected as the **Event Type**, type the appropriate name in the **Global Name** field.

- **Global Name** and **Global Value** are accessible only if **Set Global** was selected in the **Event Type** region.
- The name in the **Global Name** field identifies a variable that is made available to all jobs in AutoSys; consequently, it is a “global” variable.

13 If **Set Global** was selected as the **Event Type**, type the appropriate value (for the variable specified in the **Global Name** field) in the **Global Value** field.

14 If either **Send Signal** or **Kill Job** was selected as the **Event Type**, in the **Signal** field type the number corresponding to the UNIX signal to be sent to the job.

- The **Signal** field is accessible only if **Send Signal** or **Kill Job** was selected in the **Event Type** region.
- Numbers corresponding to UNIX signals are shown in Table 9.

Table 9. UNIX Signals (1 of 2)

NAME	VALUE	DEFAULT	EVENT
HUP	1	Exit	Hangup.
INT	2	Exit	Interrupt.
QUIT	3	Core	Quit.
ILL	4	Core	Illegal Instruction.
TRAP	5	Core	Trace/Breakpoint Trap.
ABRT	6	Core	Abort.
EMT	7	Core	Emulation Trap.
FPE	8	Core	Arithmetic Exception.
KILL	9	Exit	Killed.
BUS	10	Core	Bus Error.
SEGV	11	Core	Segmentation Fault.
SYS	12	Core	Bad System Call.
PIPE	13	Exit	Broken Pipe.
ALRM	14	Exit	Alarm Clock.
TERM	15	Exit	Terminated.
USR1	16	Exit	User Signal 1.
USR2	17	Exit	User Signal 2.
CHLD	18	Ignore	Child Status Changed.
PWR	19	Ignore	Power Fail/Restart.
WINCH	20	Ignore	Window Size Change
URG	21	Ignore	Urgent Socket Condition.
POLL	22	Exit	Pollable Event.
STOP	23	Stop	Stopped (signal).
TSTP	24	Stop	Stopped (user).
CONT	25	Ignore	Continued.
TTIN	26	Stop	Stopped (tty input).
TTOU	27	Stop	Stopped (tty output).
VTALRM	28	Exit	Virtual Timer Expired
PROF	29	Exit	Profiling Timer Expired.

Table 9. UNIX Signals (2 of 2)

NAME	VALUE	DEFAULT	EVENT
XCPU	30	Core	CPU time limit exceeded.
XFSZ	31	Core	File size limit exceeded.
WAITING	32	Ignore	Concurrency signal reserved by threads library
LWP	33	Ignore	Inter-LWP signal reserved by threads library.
FREEZE	34	Ignore	Check point Freeze
THAW	35	Ignore	Check point Thaw
CANCEL	36	Ignore	Cancellation signal reserved by threads library.
RTMIN	*	Exit	First real time signal
(RTMIN+1)	*	Exit	Second real time signal
(RTMAX-1)	*	Exit	Second-to-last real time signal.
RTMAX	*	Exit	Last real time signal

*The symbols RTMIN through RTMAX are evaluated dynamically in order to permit future configurability.

15 If **Change Status** was selected as the **Event Type**, click on the **Status** option menu button and select the desired status from the pick list.

- **Status** can be changed only if **Change Status** was selected in the **Event Type** region.
- **Status** has the following options.
  - **Running.**
  - **Success.**
  - **Failure.**
  - **Terminated.**
  - **Starting.**
  - **Inactive.**

16 If **Change Priority** was selected as the **Event Type**, type the new priority in the **Queue Priority** field.

- Queue priority can be changed only if **Change Priority** was selected in the **Event Type** region.
- The queue priority is not relevant to box jobs (DPR-level jobs).

17 If sending the event to the job is due to an emergency condition, click on the **High** button in the **Send Priority** area.

- **Send Priority** refers to the priority for sending the selected event to the job (not the job priority).
- **Send Priority** has the following options.
  - **Normal.**
  - **High.**
- **High** priority is reserved for emergencies.

18 Click on the **Execute** button.

- A confirmation dialogue box similar to Figure 76 is displayed requesting permission to proceed with sending the event.

19 Click on the appropriate button from the following selections:

- **yes** - to send the event to the job.
  - The confirmation dialogue box and the **Send Event** GUI are dismissed.
  - The selected event is sent to the specified job.
  - Once an event has been sent from the **Send Event** dialogue, it may not be possible to cancel or modify it.
- **no** - to dismiss the dialogue box and return to the **Send Event** GUI without sending the event to the job.

Canceling a Sent Event

It may be possible to cancel an event sent to an AutoSys job, especially if the event was previously scheduled for *sometime in the future*.

The Production Monitor uses the AutoSys **Send Event** GUI to cancel an event sent to a job. The procedure starts with the assumption that all applicable servers are currently running and the **Job Activity Console** GUI (Figure 60) is being displayed.

Canceling a Sent Event

---

1 Click on the **Send Event** button in the **Actions** Region of the **Job Activity Console**.

- The **Send Event** GUI (Figure 63) is displayed.

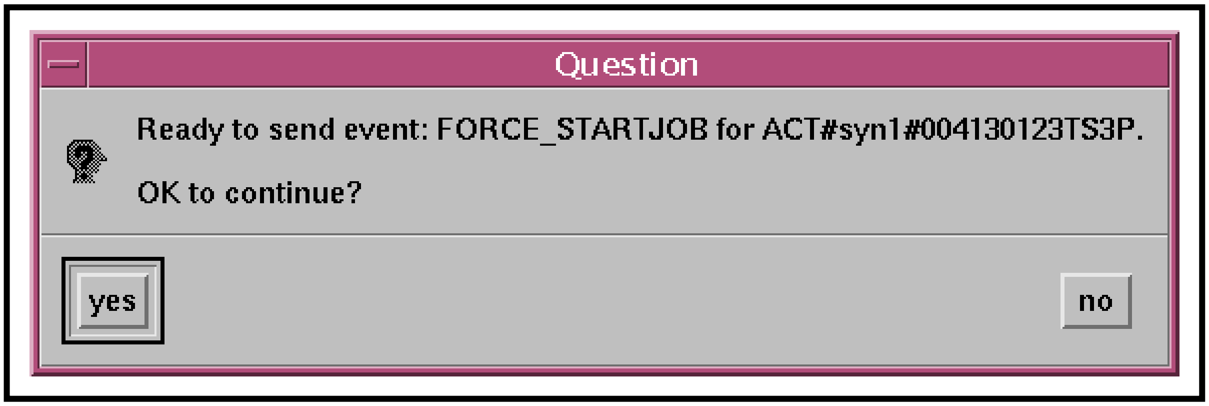

Figure 76. Send Event GUI Confirmation Dialogue Box

2 Click on **Event Type** to select the type of event that was sent to the job and is to be canceled.

3 Click on the **Cancel Previously Sent Event** radio button.

4 Verify that the correct job is listed in the **Job Name** field of the **Send Event** GUI.

- Type the job name in the **Job Name** field if necessary.
- **Not** necessary for the following types of events:
  - SET_GLOBAL.
  - STOP_DEMON.
  - ALARM.
  - COMMENT.

5 Click on the **Execute** button.

- A confirmation dialogue box similar to Figure 76 is displayed requesting permission to proceed with canceling the event.

6 Click on the appropriate button from the following selections:

- **yes** - to send the request to cancel the event.
  - The confirmation dialogue box and the **Send Event** GUI are dismissed.
  - The event is cancelled.
- **no** - to dismiss the dialogue box and return to the **Send Event** GUI without sending the request to cancel the event.

Performing Job Management Client Functions

The Job Management Client tool is a set of utility programs intended primarily for use by software developers. However, if necessary, it is possible to gain access to the following Job Management Client functions from AutoSys by clicking on the **Client Tool** button in the **Actions** region of the **Job Activity Console** (Figure 60):

- Create DPR Job.
- Release DPR Job.
- Cancel DPR Job.
- Change DPR ID.
- View Job Management DPR Queue.
- Create Ground Event Job.
- Cancel Ground Event Job.
- Change Max Concurrent Jobs for PGE Limits Table.
- Cancel Max/Min DPRs for Job Class.
- Trigger Release of Unreleased Ready-to-Run DPRs.

The Production Monitor starts the process of performing Job Management Client functions from the AutoSys **Job Activity Console** (**Ops Console**). The procedure starts with the assumption that all applicable servers are currently running and the AutoSys **Job Activity Console** (Figure 60) is being displayed.

Performing Job Management Client Functions

---

1 Verify that a box job (e.g., a box job with status to be modified) is listed in the **Currently Selected Job** field of the **Job Activity Console** (**Ops Console**) .

- Click on a job row in the **Job List** region of the **Job Activity Console** if necessary.
  - Information concerning the selected job is displayed in the **Currently Selected Job** region of the **Job Activity Console** (Figure 60).

2 Click on the **Client Tool** button in the **Actions Region** of the **Job Activity Console**.

- A confirmation dialogue box similar to Figure 75 is displayed.

3 Click on the **yes** button.

- The dialogue box closes.

- The **Job Activation User Interface** window (Figure 77) is displayed.
- The following menu options are displayed:

  0) Exit
  1) Create Dpr Job
  2) Release Dpr Job
  3) Cancel Dpr Job
  4) Change Dpr Id
  5) View Job Management Dpr Queue
  6) Create Ground Event Job
  7) Cancel Ground Event Job
  8) Change Max Concurrent Jobs for PGE Limits table
  9) Cancel Max/Min Dprs for Job Class
  a) Trigger release of unreleased ready-to-run Dprs

4 Type the number or letter corresponding to the desired function at the **enter an option** prompt then press the **Return/Enter** key.

- For example, to trigger the release of unreleased ready-to-run DPRs, type **a** then press the **Return/Enter** key.
  - The **a) Trigger release of unreleased ready-to-run Dprs** option orders the Job Management Server to check information in the limits tables and determine the next job to be placed into AutoSys.
  - The **a) Trigger release of unreleased ready-to-run Dprs** option should be used whenever **8) Change Max Concurrent Jobs for PGE Limits table** or **9) Cancel Max/Min Dprs for Job Class** has been used.

5 If applicable, type a response to the Job Management Client prompt then press the **Return/Enter** key.

6 Repeat Steps 4 and 5 as necessary.

7 To quit the Job Management Client type **0** at the **enter an option** prompt then press the **Return/Enter** key.

- Job Management Client is dismissed.

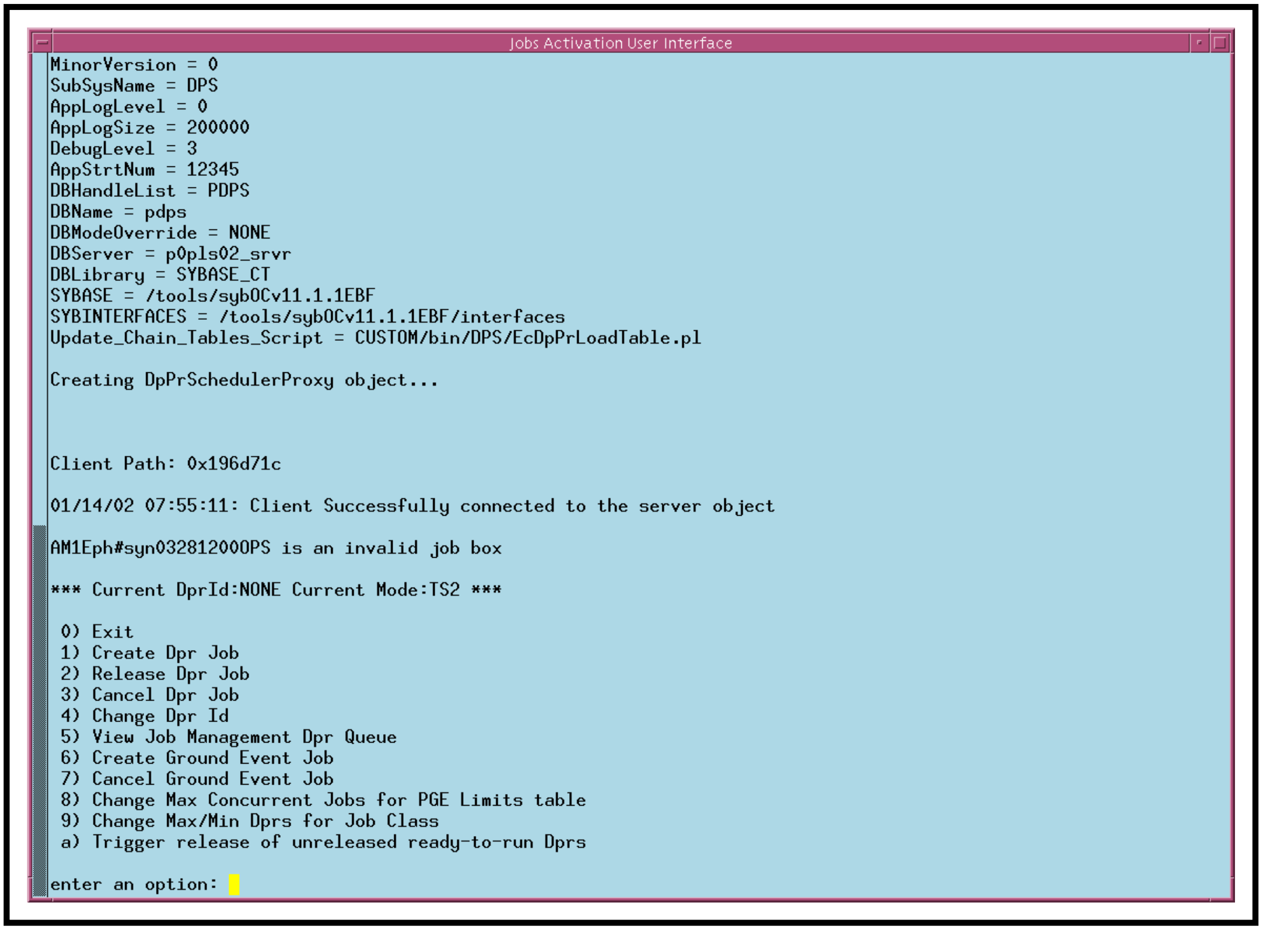

Figure 77. Job Activation User Interface Window

Reviewing Activity and Job Dependency Reports

Reviewing a Job Activity Report

The Production Monitor reviews the job activity report to obtain the following types of information:

- Which jobs are currently in the AutoSys queue.
- Which jobs have been completed.
- The completion status of jobs that have been completed.
- Which jobs are currently running.

The Production Monitor reviews the job activity report using the AutoSys **autorep** command. The **autorep** command reports information about a job, jobs within boxes, machines, and machine status. Figure 78 shows a sample job activity report.

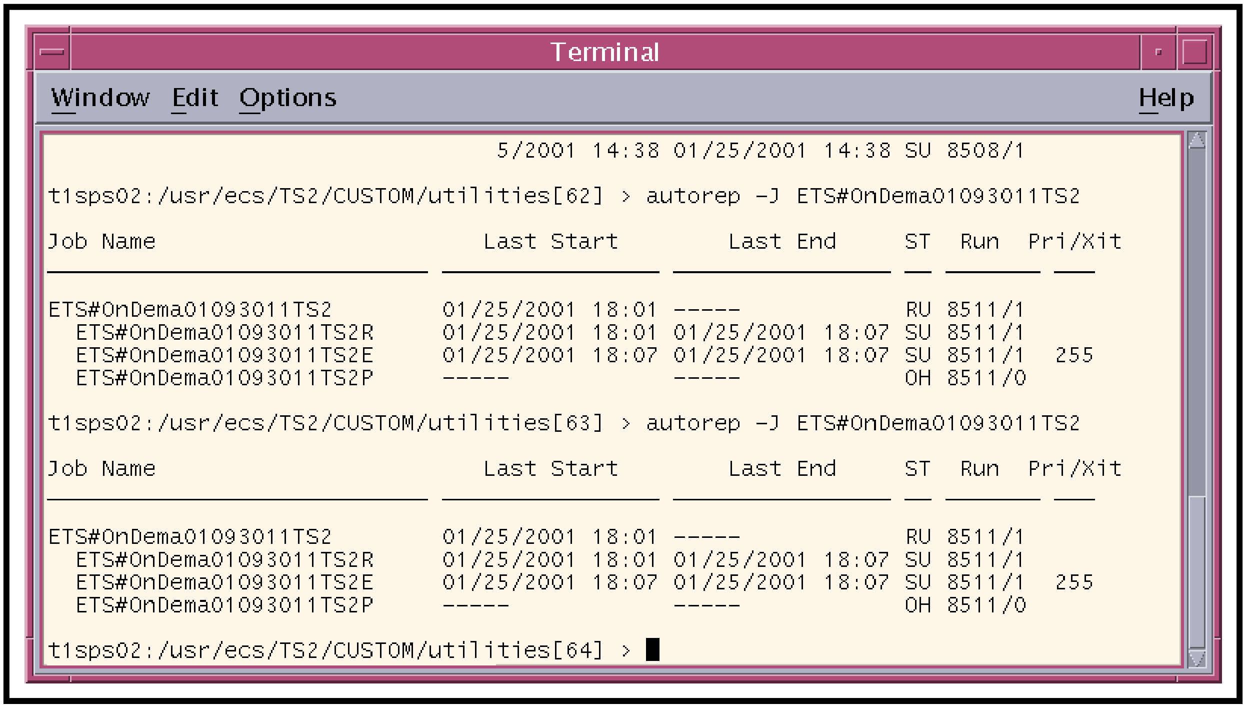

Figure 78. Sample Job Activity Report

The procedure starts with the assumption that the Production Monitor has logged in to the system.

Reviewing a Job Activity Report

---

1 Access a terminal window logged in to the Queuing Server host.

- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 Type **cd /*path*** then press **Return/Enter**.

- Change directory to the directory (e.g., /usr/ecs/*MODE*/COTS/autotreeb/autouser, /usr/ecs/*MODE*/COTS/autotree/autouser, /data1/SHARED/COTS/autotree/autouser) containing the set-up files (e.g., FMR.autosys.csh.e0sps04).

- The particular path to be typed may vary from site to site.

3 Type **source *AUTOSYSINSTANCE*.autosys.csh.*hostname*** then press **Return/Enter**.

- The **source** command sets the environment variables.

4 Type **autorep -J ALL** unless the command needs to be modified in one of the following ways:

- To specify a particular job, type the job name instead of **ALL.**
- To obtain a machine report, type **-M *machine_name*** after either **ALL** or the job name.
- To obtain a summary report, type **-s** after either **ALL** or the job name.
- To obtain a detailed report, type **-d** after either **ALL** or the job name.
- To obtain a query report, type **-q** after either **ALL** or the job name.
- To display the document one page at a time, type either **| pg** or **| more** after typing the code for whichever of the preceding options are desired.
- To print the document, type **| lp** after typing the code for whichever of the preceding options are desired.
- To save the report in a file, type **> /*path*/*filename*** after either **ALL** or the job name.

5 Press the **Return/Enter** key on the keyboard to obtain access to the **Job Activity Report**.

- The **Job Activity Report** (Figure 78) is displayed.
- If **| lp** was typed on the command line, the **Job Activity Report** is printed.
- If **> /*path*/*filename*** was typed on the command line, the **Job Activity Report** has been saved under the specified ***filename***.

6 Review the **Job Activity Report** to determine job states.

- Completed.
- Currently running.
- In the AutoSys queue.

Reviewing a Job Dependency Report

The Production Monitor reviews a job dependency report using the AutoSys **job_depends** command. The **job_depends** command reports information about the dependencies and conditions of jobs. The command can be used for determining the current state of a job, its job dependencies, the dependencies and nested hierarchies (for boxes) as specified in the job definition, and a forecast of what jobs will run during a given period of time.

The procedure starts with the assumption that the Production Monitor has logged in to the system.

Reviewing a Job Dependency Report

---

1 Set up **AutoSys** as described in Steps 1 through 3 of the procedure for **Reviewing a Job Activity Report**.

2 Type **job_depends -c -J ALL** unless the command needs to be modified in one of the following ways:

- To specify a particular job, type the job name instead of **ALL**.
- To obtain the current condition status, type **-c** before **-J**.
- To obtain the dependencies only, type **-d** before **-J**.
- To obtain the time dependencies, type **-t** before **-J**.
- To display the document one page at a time, type either **| pg** or **| more** after typing the code for whichever of the preceding options are desired.
- To print the document, type **| lp** after typing the code for whichever of the preceding options are desired.
- To save the report in a file, type **>** ***/path/filename*** after either **ALL** or the job name.

3 Press the **Return/Enter** key on the keyboard to obtain access to the **Job Dependency Report**.

- The **Job Dependency Report** (Figure 79) is displayed.
- If **| lp** was typed on the command line, the **Job Dependency Report** is printed.
- If **>** ***/path/filename*** was typed on the command line, the **Job Dependency Report** has been saved under the specified ***filename***.

4 Review the **Job Dependency Report** to determine job status, including the status of atomic conditions.

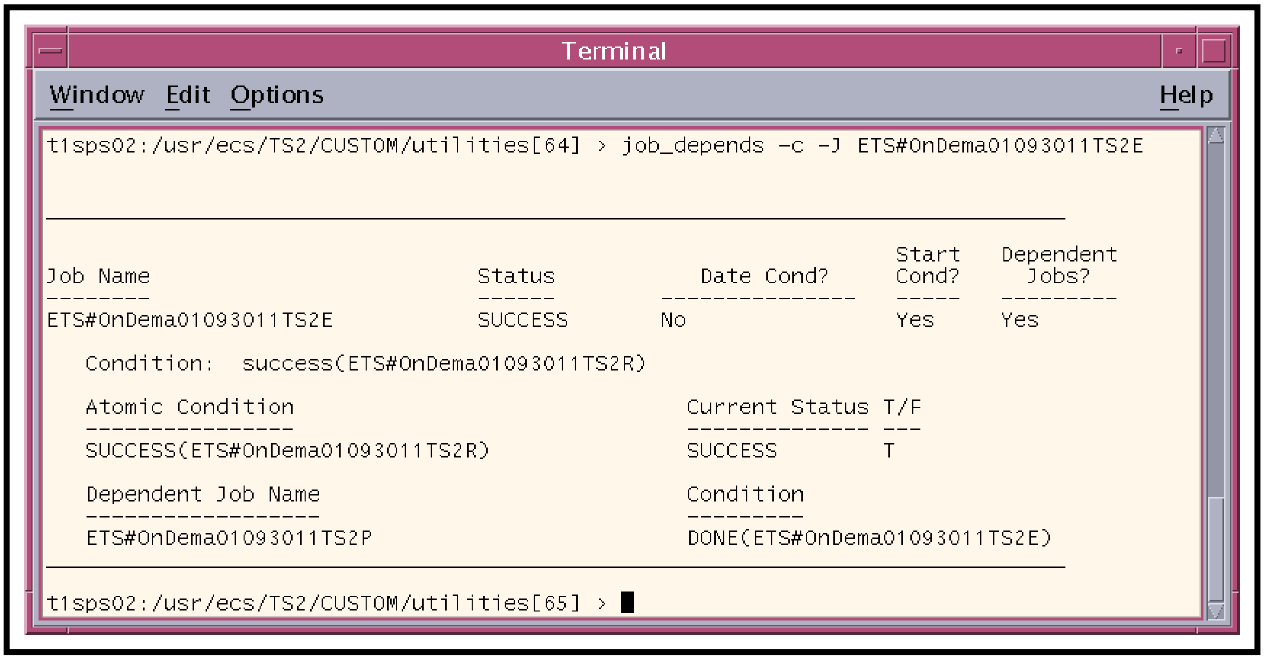

Figure 79. Sample Job Dependency Report

Defining and Running Monitors/Browsers

Defining Monitors/Browsers

The current edition of the *Release 7.11 Operations Tools Manual for the EMD Project* (609-EMD-001) indicates that the project does not support the AutoSys monitor/browser capabilities. However, they are functional and the Production Monitor can use them (with no expectation of project support if problems are encountered).

Although some Production Monitors may wish to monitor all events, it is more likely that they will prefer to limit monitoring to alarms and changes of job status (e.g., from "running" to "success" or "failure"). The browser function (Figure 80) is particularly useful for determining the eventual status of jobs run during the preceding shift or day; for example, which jobs were successful, which jobs failed, and which jobs are still running.

NOTE: When all events for all jobs should be monitored, do *not* run a monitor. Instead, display the Event Processor log in real time (using the command **autosyslog -e**). Running a monitor adds another connection to the database and establishes an additional process that is continually polling the database. That has a significant impact on system performance.

The procedure for defining monitors or browsers starts with the assumption that all applicable servers are currently running, AutoSys has been launched, and the **AutoSys GUI Control Panel** (Figure 53) is being displayed.

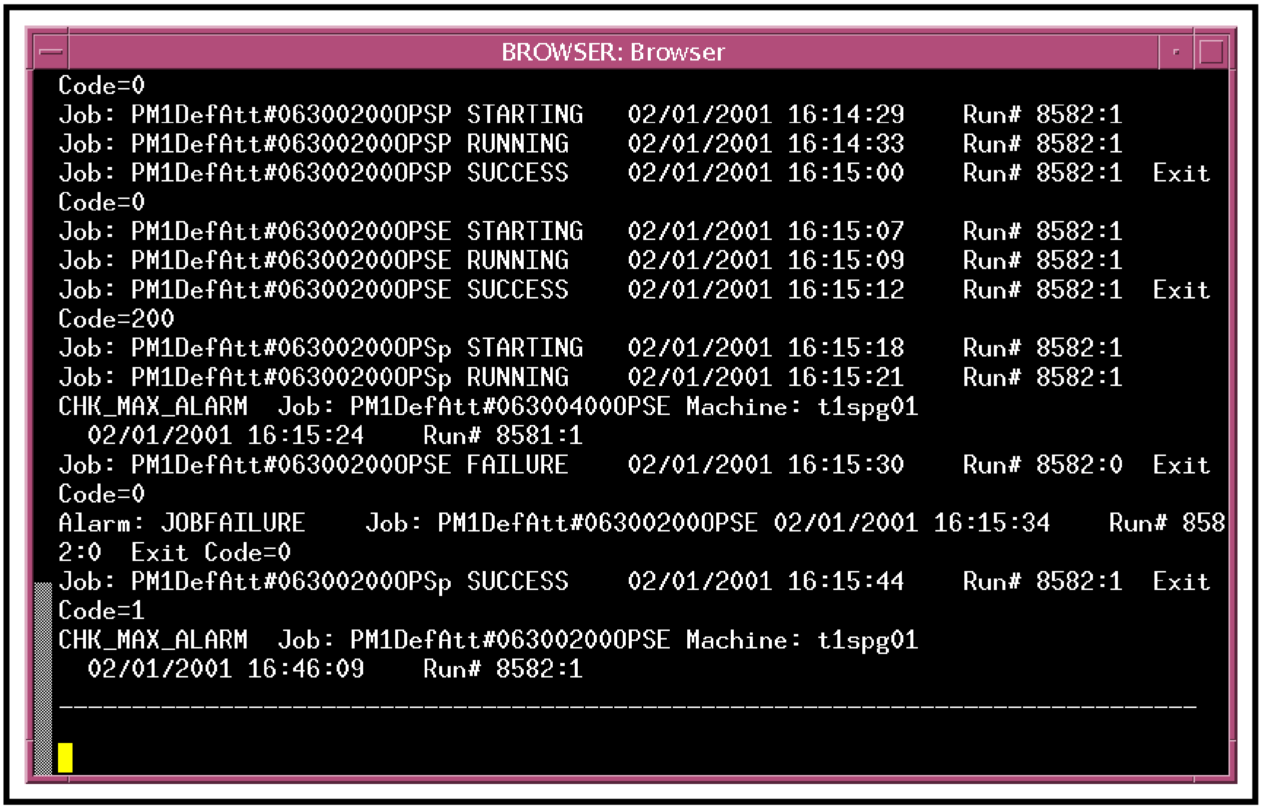

Figure 80. Sample Browser Screen

Defining Monitors/Browsers

---

1 Click on the **Monitor/Browser** button on the **AutoSys GUI Control Panel**.

- The **Monitor/Browser** GUI (Figure 81) is displayed.

2 Type a name for the monitor or browser in the **Name** field near the top of the GUI.

- Name must be in valid file-name format.
- If a pre-defined monitor or browser is desired, use the **Search** button under the **Name** field to call it up.

3 Click on either the **Alarms** button or the **ALL EVENTS** button for **Types of Events.**

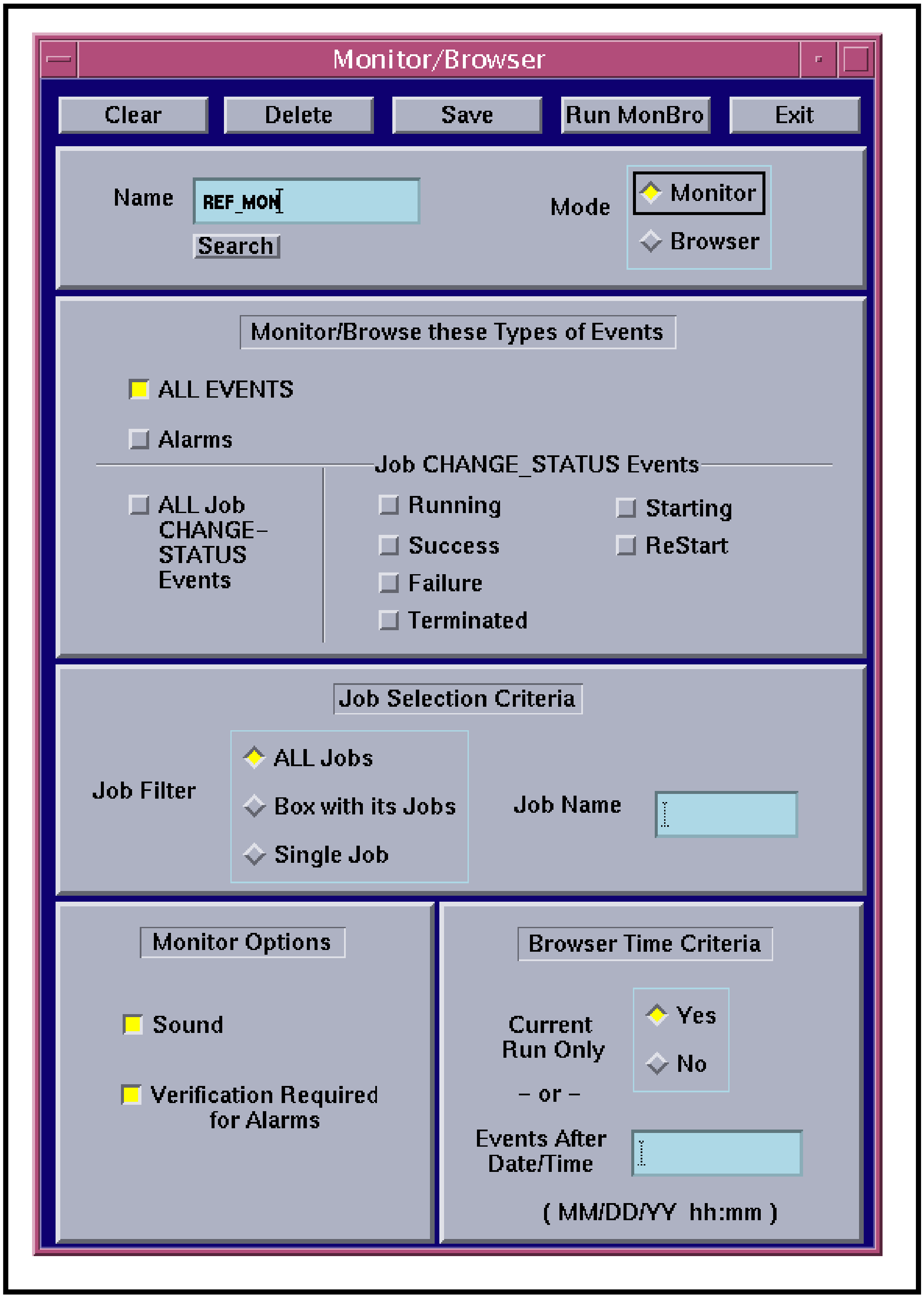

Figure 81. Monitor/Browser GUI

4 Click on either **ALL Job Status Events** or the corresponding toggle button(s) to select individual **Job Status Events.**

- Any or all of the following **Job Status Events** can be selected:
  - **Running.**
  - **Success.**
  - **Failure.**
  - **Terminated.**
  - **Starting.**
  - **ReStarting.**
  - **On Ice.**
  - **On Hold.**

5 Click on the corresponding toggle button to select the desired **Job Selection Criteria.**

- Job selection criteria options are as follows:
  - **All Jobs.**
  - **Box with its Jobs.**
  - **Single Job.**

6 If **Single Job** was selected in the previous step, type the job name in the **Job Name** field.

7 Click on the corresponding toggle button to select the desired **Monitor Options.**

- **Monitor Options** refers to one of the following choices:
  - **Sound.**
  - **Verification Required for Alarms.**

8 Click on **Yes** or **No** to select the desired **Current Run Time** and/or **Events After Date/ Time**, which are the **Browser Time Criteria**.

9 If **Events After Date/Time** was selected in the previous step, type the starting date and time (in *MM/DD/YY hh:mm* formats) in the **Events After Date/Time** field.

10 Click on the corresponding toggle button to select the desired **Mode**.

- The following options are available:
  - **Monitor.**
  - **Browser.**

- If **Monitor** is selected, settings are defined for a monitor.
- If **Browser** is selected, settings are defined for a report.

11 Click on the **Save** button.

- The monitor or browser definition is saved to the database.
- Before running a monitor or browser you must **Save** the monitor/browser definition first.

12 Click on the **Run MonBro** button to run the monitor/browser that has just been defined.

13 Review the monitor/browser results.

14 To exit from a browser or monitor type **Ctrl-C** in the monitor/browser window.

Running Monitors/Browsers

There are two procedures for running monitors/browsers. Monitors/browsers may be run from the **Monitor/Browser** GUI or they may be run using UNIX commands. In either case, the procedure starts with the assumptions that all applicable servers are currently running, AutoSys has been launched, and the **AutoSys GUI Control Panel** (Figure 53) is being displayed.

Running Monitors/Browsers from the Monitor/Browser GUI

1 Click on the **Monitor/Browser** button on the **AutoSys GUI Control Panel**.

- The **Monitor/Browser** GUI (Figure 81) is displayed.

2 If the desired monitor or browser has not been previously defined, define the monitor or browser as described in the procedure for **Defining Monitors/Browsers** (preceding section of this lesson).

- After defining the monitor or browser go to Step 7.

3 If the name of the monitor or browser is known exactly, type the name in the **Name** field.

- Go to Step 7.

4 If the name of the monitor or browser is **not** known exactly, type **%** (percent sign wild card) in the **Name** field.

5 Click on the **Search** button.

- A dialogue box containing a list of previously defined monitors and browsers is displayed.

6 If the name of the desired monitor or browser is displayed in the dialogue box, double-click on the name to retrieve the desired monitor/browser definition.

7 Click on the **Run MonBro** button.

- Monitor/browser is displayed in a separate window.

8 Click on the **Exit** button to exit from the **Monitor/Browser** GUI.

9 Review the monitor/browser results.

10 To exit from the browser or monitor type **Ctrl-C** in the browser/monitor window.

Running Monitors/Browsers Using UNIX Commands

---

1 If the desired monitor or browser has not been previously defined, define the monitor or browser as described in the procedure for **Defining Monitors/Browsers** (preceding section of this lesson).

2 Access a terminal window logged in to the Queuing Server host.

- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

3 Type **cd /*path*** then press **Return/Enter**.

- Change directory to the directory (e.g., /usr/ecs/*MODE*/COTS/autotreeb/autouser, /usr/ecs/*MODE*/COTS/autotree/autouser, /data1/SHARED/COTS/autotree/autouser) containing the set-up files (e.g., FMR.autosys.csh.e0sps04).
- The particular path to be typed may vary from site to site.

4 Type **source *AUTOSYSINSTANCE*.autosys.csh.*hostname*** then press **Return/Enter**.

- The **source** command sets the environment variables.

5 Type **monbro -N *name* &** then press the **Return/Enter** key on the keyboard to run the previously defined monitor/browser.

- The monitor or report (browser) must have been previously defined and saved under an appropriate file ***name*** using the **Monitor/Browser** GUI.
- The report is displayed.

- Refer to the *AutoSys® Reference Guide for UNIX* for all options and displays for **monbro** reports.
  - The *AutoSys® Reference Guide for UNIX* and the *AutoSys® User Guide for UNIX* can be downloaded from the Computer Associates Technical Support website but require an account and login. Contact the DAAC COTS software representative for assistance.

6 Review the monitor/browser results.

7 To exit from the browser or monitor type **Ctrl-C**.

Tuning System Parameters

Tuning System Configuration Parameters

The values assigned to system parameters affect the functioning and performance of the system. When certain parameters are modified, the system operates differently. Changes to some other parameters may not appear to affect the system although there may in fact be subtle effects. In any case before system parameters are modified it is essential to understand what will happen to system functioning and performance.

Many system parameters may be subject to control by Configuration Management (CM). When making or requesting a change to system parameters, the CM process at the particular site must be followed (if applicable).

Values are assigned to Data Processing Subsystem and Planning Subsystem parameters in the following databases:

- Configuration Registry database.
- PDPS database.

Parameters in the Configuration Registry Database

The Configuration Registry Server provides a single interface (via a Sybase server) for retrieving configuration attribute-value pairs for system servers from the Configuration Registry database. When system servers are started, they access the Configuration Registry Database to obtain needed configuration parameters.

The Database Administrator has access to a Configuration Registry GUI for viewing and editing configuration data in the database. Therefore, it is necessary to coordinate with the Database Administrator when changes to configuration parameters are needed. Also, as previously mentioned, changes to configuration-controlled parameters are subject to approval through the site CM process.

Default and adjusted values assigned to system parameters vary from site to site. For guidance concerning the assignment of values to parameters included in the Configuration Registry refer to document 910-TDA-022, *Custom Code Configuration Parameters for ECS*. The document is available at http://cmdm-ldo.raytheon.com/baseline/ under “Technical Documents.”

The following parameters are examples of parameters whose values may be modified to enhance system functioning or performance:

- AppLogSize [parameter applies to all servers].
  - Maximum size of the application log (ALOG) file for a particular application.

- Recommended size varies considerably depending the nature of the application for which the file is being written.

- AppLogLevel [parameter applies to all servers].
  - Level of detail provided in the ALOG file for a particular application.
  - Acceptable values are 0, 1, 2, or 3.
  - A setting of “0” provides the most data.
- DebugLevel [parameter applies to all servers].
  - Level of detail provided in the debug log file for a particular application.
  - Normally acceptable values are 0, 1, 2, or 3.
  - A setting of "0" turns off logging; a setting of “3” provides a significant amount of data.
- DpPr_MAX_RETRIES [EcDpPrEM and EcDpPrDeletion parameter (also EcDpPrQaMonitorGUI and several Science Software Integration and Test programs)].
  - Number of retries (e.g., 30) to the Science Data Server for acquires/inserts before giving up.
- DpPr_WAIT_PERIOD [EcDpPrEM and EcDpPrDeletion parameter (also EcDpPrQaMonitorGUI and several Science Software Integration and Test programs)].
  - Time in seconds (e.g., 120) to wait between retries to the Science Data Server.
- DpPrRM_MAX_RETRIES [EcDpPrEM, EcDpPrGE, EcDpPrJobMgmt, EcDpPrDeletion parameter].
  - Maximum number (e.g., 100) of attempts to allocate a computer resource.
- DpPrRM_RETRY_PERIOD [EcDpPrEM, EcDpPrGE, EcDpPrJobMgmt, EcDpPrDeletion parameter].
  - Number of seconds (e.g., 120) between retries when trying to allocate a resource.
- DpPrMaxConcurrentDPRs [EcDpPrJobMgmt parameter].
  - Maximum allowed jobs.
  - Three integer values (e.g., 100 100 100) are assigned to DpPrMaxConcurrentDPRs; the first for routine processing; the second for on-demand processing; and the third for reprocessing jobs.

- DpPrMinConcurrentDPRs [EcDpPrJobMgmt parameter].
  - Minimum allowed jobs.
  - Three integer values (e.g., 0 0 0) are assigned to DpPrMaxConcurrentDPRs; the first for routine processing; the second for on-demand processing; and the third for reprocessing jobs.
  - Minimum number of concurrent DPRs for each job class (i.e., routine, on demand, reprocessing) NOT CURRENTLY USED.
- DpPrAutoSysMaxDPRs [EcDpPrJobMgmt parameter].
  - Maximum number of completed DPRs (i.e., in SUCCESS or FAILEDPGE state) in AutoSys.
  - When the maximum number of completed DPRs is in AutoSys, the next DPR that succeeds or fails causes the oldest completed DPR to be deleted from AutoSys.
  - If the value assigned to DpPrAutoSysMaxDPRs is too low, completed jobs are swept out of AutoSys very quickly, which may not allow the operator enough time to see that the job was completed.
- DpPrDeleteFailedPGEJobs [EcDpPrJobMgmt parameter].
  - If TRUE, failed PGE Jobs are removed by Job Management, as necessary, when space is needed for another job that is ready to run. This is recommended to keep job management straightforward. However, this may be confusing for the operator, since they may not get a chance to see the failure if the system is busy.
  - If FALSE (the usual value), failed PGE Jobs are left in AutoSys. They must not be removed manually from AutoSys, however, since they will be removed by the Production Request Editor when a Production Request or DPR is cancelled.
- DBConnections [EcPoConnections (includes EcPlSubMgr, EcPlOdMgr, EcDpPrDeletion, EcDpPrJobMgmt and EcDpPrJobMgmtClient) parameter].
  - Number of connections needed by a particular application (e.g., 10 for EcPlOdMgr).
  - Optional parameter that specifies the number of connections to maintain in the connection pool.
  - The parameter is a list of positive integers. There must be one entry for each DbHandle in the DbHandleList.
  - Generally it should be set to the maximum number of connections that are expected to be used simultaneously in a process. If one connection per thread is used, this will be the same as the number of concurrent threads expected to

execute. When the pool is used up there is a performance penalty to allocate and deallocate connections on the fly.

  – If this parameter is not specified or is given as “NONE”, it defaults to 1.

- SleepDelayForFailures [EcPlSubMgr parameter].
  – Amount of time in seconds (e.g., 60) to wait before reprocessing failed notifications. If the specified value were less than 60, a default value of 60 seconds would be assumed.
  – Duration of the sleep delay used by the failed notification thread in seconds.
  – Less frequent checking can increase speed for the other threads.
- SleepDelayForTimers [EcPlSubMgr parameter].
  – Amount of time in seconds (e.g., 60) the Subscription Manager should sleep between checking for expired timers. It should be set to the minimum amount of time a timer will be set for at this DAAC. The minimum it can be set to is 60 seconds.
  – Duration of sleep delay used by the timer checking thread in seconds.
  – Less frequent checking can increase speed for the other threads.
- SleepDelayForExp [EcPlOdMgr parameter].
  – Sleep delay for expiration thread in seconds (e.g., 86400).
  – Should be considerably greater than the sleep delay for completion threads (SleepDelayForCmp).
- SleepDelayForCmp [EcPlOdMgr parameter].
  – Sleep delay for completion threads in seconds (e.g., 300).
  – Should be considerably less than the sleep delay for expiration threads (SleepDelayForExp).
- SocketLimit [EcDpPrDeletion, EcDpPrJobMgmt, EcPlOdMgr, EcPlSubMgr parameter].
  – Number of connections (e.g., 200) to a server through the Hubble Space Telescope (HST) sockets middleware.
  – Too low a number misses connections.
  – Too high a number may adversely affect the memory of the server's host.

When the value assigned to a parameter has been changed and saved in the Configuration Registry, the modified value does not take effect until the affected server has been restarted. For example, if the debug level for the Subscription Manager log has been changed from “2” to “3” in the Configuration Registry, the modification does not affect the recording of data in the log until after a warm restart of the Subscription Manager (at which time the server would read the parameters in the Configuration Registry).

Parameters in the PDPS Database

The following two tables in the PDPS database have significant effects on the running of DPRs:

- DpPrPgeLimits - controls where DPRs run.
- DpPrClassSchedulingLimits - controls how many DPRs run at a time.

DpPrPgeLimits imposes restrictions on the number of DPRs of a particular PGE that can run simultaneously on the same virtual computer. A database record defines each pgeId/computerName (PGE/virtual computer) combination that will be run and how many jobs (DPRs) associated with the particular combination can run at the same time.

The DpPrPgeLimits table has the following columns:

- pgeId – PGE ID.
- computerName – virtual computer name.
- maxConcurrent - maximum number of jobs (DPRs) associated with a particular pgeId/computerName combination that can run at the same time.
- numConcurrent - number of jobs (DPRs) currently running in AutoSys for a particular pgeId/computerName combination.
- numScheduled - shows how many jobs (DPRs) are currently scheduled on a specific pgeId/computerName combination.

Unless a particular host is specified (using the Production Request Editor) when a Production Request is created, all jobs in a chain are scheduled to run on the machine(s) [virtual computer(s)] specified for the PGE in the DpPrPgeLimits table in the PDPS database. However, if no machine is specified in either the Production Request or in the DpPrPgeLimits table, the jobs run on the computer entered in the PlResourceRequirement table during PGE registration.

An easy way to balance the load on two or more virtual computers is to specify an equal number of pgeIds to run on each virtual computer. If the number is large (e.g., 10,000), potentially all ready-to-run DPRs specifying the PGE can run and the number is balanced on the valid computers. If the number is small (e.g., two per machine), the number of DPRs using the PGE can be throttled, with the excess DPRs being queued.

Now, if controlling the total number of DPRs that can run at any one time is considered necessary, the DpPrClassSchedulingLimits table is involved. The table controls the total number of concurrent DPRs scheduled for Routine, Reprocessing and On-demand processing. When a slot is free, all ready-to-run DPRs that have empty slots in DpPrPgeLimits are considered and the DPR with the oldest time stamp in the PlDataProcessingRequest table is selected.

As previously mentioned the DpPrClassSchedulingLimits table controls the total number of concurrent DPRs scheduled for the following classes of processing:

- Routine.
- Reprocessing.
- On Demand.

Consequently, the DpPrClassSchedulingLimits table has three records, one for each type of processing. Each record has the following fields:

- dprClass - assigned value identifies the type of processing.
  - 0 = Routine Processing.
  - 1 = On-Demand Processing.
  - 2 = Reprocessing.
- maxDprs - maximum number of jobs (DPRs) of the type (specified in dprClass) that are allowed to run on the system.
- minDprs - currently not used.
- currentDprs - number of jobs (DPRs) of the type (specified in dprClass) that are currently running.

If the DpPrClassSchedulingLimits table has no record for a particular type of processing, DPRs of that type are not allowed into AutoSys.

Values for the maxDprs and minDprs columns in the DpPrClassSchedulingLimits table are loaded at Job Management Server startup using data from the following two configuration parameters:

- DpPrMaxConcurrentDPRs - maximum allowed jobs
- DpPrMinConcurrentDPRs - minimum allowed jobs

Each parameter has three integer values; the first for routine processing; the second for on-demand processing; and the third for reprocessing jobs.

- For example, the Configuration Registry may have the following entries:

  DpPrMaxConcurrentDPRs = 100 60 40
  DpPrMinConcurrentDPRs = 0 0 0

  - In this case the maximum allowed jobs is 100 for routine processing, 60 for on-demand processing, and 40 for reprocessing.
  - The minimum allowed jobs is 0 for each type of processing.

Modifying the DpPrPgeLimits and DpPrClassSchedulingLimits Tables (PDPS Database)

Either the DpPrPgeLimits table or the DpPrClassSchedulingLimits table can be loaded by running the EcDpPrLoadTable.pl script from the Job Management Client tool (using the appropriate option).

- The Job Management Client tool is accessed through the AutoSys Job Activity Console.
- The EcDpPrLoadTable.pl script loads values from an input data file.
- Instructions for using the script are available in the EcDpPrLoadTable.README file in the /usr/ecs/*MODE*/CUSTOM/data/DPS directory on the Queuing Server host.
  - The same directory has a template for constructing the necessary input data file.

The Job Management Client tool has the following options for modifying the DpPrPgeLimits table or the DpPrClassSchedulingLimits table:

8) Change Max Concurrent Jobs for PGE Limits table

- For DpPrPgeLimits table modifications.

9) Change Max/Min Dprs for Job Class

- For DpPrClassSchedulingLimits table modifications.

For detailed instructions on modifying the DpPrPgeLimits table or the DpPrClassSchedulingLimits table using the Job Management Client tool refer to the procedure for **Performing Job Management Client Functions** (preceding section of this lesson).

An alternative method of modifying the DpPrPgeLimits table or the DpPrClassSchedulingLimits table is to create one's own load script using SQL statements. It is acceptable to add pgeId entries for a machine, add new machines to the DpPrPgeLimits table, or change the maximum number of DPRs that can concurrently execute in DpPrClassSchedulingLimits. However, values for the number of currently scheduled or running DPRs in the tables must not be changed.

Also, note that the DpPrPgeLimits table can be empty but DpPrClassSchedulingLimits must be fully populated. As previously mentioned, default values for the maxDprs and minDprs columns in the DpPrClassSchedulingLimits table are loaded at Job Management Server startup using data from configuration parameters in the Registry database.

Monitoring the Load on Processing Resources

The Production Planner and Production Monitor should work with the Resource Planner to make optimum use of processing resources. The Resource Planner allocates the disk partitions, CPUs, and RAM available for processing among the active modes (e.g., OPS, TS1, or TS2). The Production Planner and Production Monitor monitor the load on the processing resources.

The Resource Planner assigns the bulk (typically 60% - 80%) of the processing resources to the OPS mode. The remainder of the processing assets is divided among the modes used for SSI&T and new version software checkout.

The Production Planner and Production Monitor monitor the load on the processing resources to identify whether the actual load is appropriately distributed among modes. They inform the Resource Planner of under- or over-use of resources as allocated.

When monitoring the load on the processing resources, the Production Planner and Production Monitor should take the following considerations into account:

- Disk space allocated to OPS mode is likely to be used to capacity.
- Disk space assigned to the other two modes may not fill up.
- There is no one-to-one mapping of CPU allocation with actual CPUs on the science processor.
- The operating system (OS) takes care of true CPU and RAM allocation.
  - Actual CPU usage during processing is limited by the OS.
  - If ten CPUs have been specified for a particular mode, only ten Data Processing Requests (DPRs) can be running the Execute job at a given time.
  - What is really being defined is the maximum number of DPRs that will execute at a given time.
- CPUs can be over-allocated or under-allocated as necessary to get the most out of the CPUs on each science processor.
- If monitoring indicates that the processor is underused when OPS mode is at full processing capacity, the number of CPUs allocated to OPS mode could probably be increased.
- If the science processor is at full capacity when OPS mode is at full processing capacity (and the processor may be overworked) the number of CPUs allocated to OPS mode should be reduced.
- Random-access memory (RAM) is subject to the same considerations as CPUs.
  - RAM can be over-allocated or under-allocated as necessary to get the most out of the memory on each science processor.

Strategies for Tuning

A scenario that demonstrates how DPRs might be processed under a particular set of conditions and some strategies for tuning the system are presented in the paragraphs that follow. The processing conditions include the following types of items:

- The total number of jobs allowed into AutoSys.
- The number of CPUs available for processing.
- Characteristics of the PGEs to be processed.

The total number of jobs allowed into AutoSys is controlled by the DpPrPgeLimits table in the PDPS database. An example of some of the types of data maintained in the DpPrPgeLimits table is shown in Table 10.

Table 10. Example of PDPS Database DpPrPgeLimits Table Contents (Selected Columns)

ComputerName [Virtual Computer]	pgeId	maxConcurrent [DPRs]
A	1	20
B	1	20
A	2	20
B	2	20

The scenario assumes that each of the virtual computers (i.e., A and B) listed in Table 10 has 16 CPUs. (There are 32 CPUs total.)

Relevant PGE characteristics are shown in Table 11.

Table 11. PGE Characteristics

PGE	# CPUs Used	Average Execution Time	Average Stage Time	Destage Time
1	1	5 minutes	5 minutes	5 minutes
2	1	60 minutes	5 minutes	5 minutes

Assuming that 100 DPRs of each type (i.e., PGE 1 and PGE 2 - 200 DPRs total) are ready to run and are released at once into AutoSys, the following actions occur:

- Eighty (80) DPRs enter AutoSys. The remaining 120 are queued, with their assignments already made:
  - Machine (Virtual Computer) A.
    - 20 PGE 1s start staging.
    - 30 PGE 1s are queued on Machine A.
    - 20 PGE 2s start staging.
    - 30 PGE 2s are queued on Machine A.
  - Machine (Virtual Computer) B.
    - 20 PGE 1s start staging.
    - 30 PGE 1s are queued on Machine B.
    - 20 PGE 2s start staging.
    - 30 PGE 2s are queued on Machine B.
- After about five (5) minutes, all 80 DPRs that were staging have finished staging and are ready for execution. However, only 32 CPUs are available.
- The first 32 DPRs that ask for CPUs get them and start running [sixteen (16) on Machine A and sixteen (16) on Machine B]. Forty-eight (48) DPRs are waiting.
  - Assuming that parameters in the Registry database are set as follows:
    - DpPrRM_RETRY_PERIOD = 120 seconds.
    - DpPrRM_MAX_RETRIES = 100.

    the waiting DPRs keep trying every two minutes for up to 100 times each before timing out (after 200 minutes).
  - Note that in this example timing out is a real possibility.
- The quick jobs complete processing after five (5) minutes, freeing up sixteen (16) CPUs. In the current example, the sixteen (16) CPUs are subsequently occupied with about eight (8) five-minute PGEs and eight (8) 60-minute PGEs because CPUs are given randomly to whichever DPR gets back first to asking for them after waiting for the retry period (i.e., 120 seconds). Priorities are not used.
  - At first, there was a 50:50 ratio of fast:slow DPRs, now there is a 25:75 ratio of fast:slow. After another five (5) minutes, the ratio becomes 12.5:87.5 fast:slow, so 87.5 % of the CPUs are occupied by 60-minute DPRs.

- Apparently, the 60-minute DPRs tend to dominate the CPUs. After one (1) hour the first batch of sixteen (16) 60-minute PGEs vacates the CPUs to be replaced by eight (8) five-minute PGEs and eight (8) 60-minute PGEs, but the five-minute PGEs become extinguished again by the slow ones.
  - If the staging and destaging times were not the same (so the DPRs didn't have the same opportunity to hit the execution stage at the same time) the scenario would proceed differently.

Various strategies can be employed to tune the system:

- Limit the number of DPRs through the use of the DpPrPgeLimitsTable.
  - In the preceding example if the number of slow DPRs allowed into AutoSys is less than the number of CPUs, there is always a channel for the fast jobs to squeeze through.
  - The big disadvantage to this approach is that the slow jobs are also being prevented from staging.
- Increase the declared number of CPUs for the processors to more than the actual number (overallocate CPUs).
  - This approach allows more of each type of PGE into the science processors.
  - The disadvantage is that it could overwhelm the science computers. However, they are kept busy.
- Create new virtual computers (assigning CPUs on the processors to them) and assign (via the DpPrPgeLimits table) PGEs to run on the new virtual computers.
  - This approach is another way to guarantee bandwidth (CPUs) to PGEs.
  - The disadvantage of this approach is that some CPUs could remain idle, not being seen by one of the virtual computers.
  - In the past, there may have also been some code problems with supporting this, but those difficulties should have been resolved.

Probably some combination of the first two of the preceding strategies is best; i.e., increase the number of declared CPUs to be more than the total number of slow jobs allowed into AutoSys, always leaving some CPUs for a channel of fast jobs. The total number of faster-moving jobs should be increased to make sure that there is always be a queue of them available to get their channel occupied.

The staging and destaging times have to be accounted for and this could change things in terms of using the DpPrPgeLimits table and the number of CPUs per processor to tune the job flow.

Also, it is important to perform regular garbage collection on all of the virtual computers. Procedures for cleaning the PDPS database and DPS disks (i.e., “garbage collection”) are provided in a previous section of this lesson.

Changing the AutoSys Database Maintenance Time

Once a day the Event Processor (also known as the AutoSys daemon) goes into an internal database maintenance cycle. During this time, it does not process any events and it waits for the maintenance activities to be completed before resuming normal operations. The time of day for start-up of the maintenance cycle is pre-set to 3:30 AM. The database maintenance cycle takes approximately one minute. If it is necessary to change the time when the maintenance cycle occurs, whoever has "write" access to the configuration file can reset it, preferably to a time when there is minimal activity.

The procedure for changing the AutoSys database maintenance time starts with the assumption that the Production Monitor has logged in to the system.

Changing the AutoSys Database Maintenance Time

---

1 Access a terminal window logged in to the Queuing Server host.

- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 Type cd /*path* then press Return/Enter.

- Change directory to the directory (e.g., /usr/ecs/*MODE*/COTS/autotreeb/autouser, /usr/ecs/*MODE*/COTS/autotree/autouser, /data1/SHARED/COTS/autotree/autouser) containing the **config.*AUTOSYSINSTANCE*** file.
- The particular path to be typed may vary from site to site.

3 Type **vi config.*AUTOSYSINSTANCE*** then press **Return/Enter**.

- The configuration file is displayed by the vi text editor.
- Although this procedure has been written for the **vi** command, any UNIX editor can be used to edit the configuration file.

4 Using vi editor commands find **DBMaintTime=** and replace the existing time with the desired time in 24 hour format (hh:mm).

- The time may already have been changed to some value other than 03:30 (e.g., **DBMaintTime=04:00**).
- The following vi editor commands are useful:
  - **h** (move cursor left).
  - **j** (move cursor down).
  - **k** (move cursor up).

- **l** (move cursor right).
- **a** (append text).
- **i** (insert text).
- **r** (replace single character).
- **x** (delete a character).
- **dw** (delete a word).
- **dd** (delete a line).
- ***n*dd** (delete *n* lines).
- **u** (undo previous change).
- **Esc** (switch to command mode).

5 Press the **Esc** key.

6 Type ZZ.

- New database maintenance time is entered and saved in the configuration file.
- UNIX prompt is displayed.

This page intentionally left blank.

Troubleshooting Processing Problems

Trouble Symptoms

Troubleshooting is a process of identifying the source of problems on the basis of observed trouble symptoms. One common source of problems involves connections with other subsystems for the transmission of messages or data. Like many other operational areas in the system, processing has interfaces with many other subsystems. Consequently, problems with processing can be traced to either the Data Processing Subsystem or one of many other subsystems, including (but not necessarily limited to) those in the following list:

- Planning Subsystem (PLS).
- Data Server Subsystem (DSS).
- Communications Subsystem (CSS).

Fault Recovery

Refer to the **Fault Recovery** topic in the section on **Troubleshooting Production Planning Problems** (previous section of this lesson).

Troubleshooting a Processing Failure

Table 12 describes actions to be taken in response to some common Processing problems. If the problem cannot be identified and fixed without help within a reasonable period of time, the appropriate response is to call the help desk and submit a trouble ticket in accordance with site Problem Management policy.

Table 12. Troubleshooting Processing Problems (1 of 2)

Symptom	Response
Unable to log in to the Queuing Server host (e.g., e0sps04).	Check with the Operations Controller/System Administrator to ensure that the host is “up."
GUI not displayed when the start-up script has been properly invoked.	Ensure that the DISPLAY variable was set properly. [For detailed instructions refer to the procedure for **Launching the AutoSys GUI Control Panel** (previous section of this lesson).]

Table 12. Troubleshooting Processing Problems (2 of 2)

Symptom	Response
Entire processing system hangs (no jobs change state over time).	1. Ensure that it is possible to connect to the necessary hosts and servers (listed in Table 6). [For detailed instructions refer to the section on **Checking Connections to Hosts/Servers** (subsequent section of this lesson).] 2. If hosts/servers are all “up,” perform the procedure for **Responding to Hanging of the Processing System** (subsequent section of this lesson).
Jobs are activated but do not get started in AutoSys.	Refer to the procedure for **Responding to Failure of Jobs to Start in AutoSys** (subsequent section of this lesson).
AutoSys box job hangs (does not change state over time).	Refer to the procedure for **Handling a Box Job that is Hanging in AutoSys** (subsequent section of this lesson).
“Preprocess” function fails (job either does not change state over time or has turned red on JobScape or TimeScape).	Refer to the procedure for **Handling a Hanging or Failed Preprocessing Job** (subsequent section of this lesson).
“Execute” job hangs (job has turned orange or oscillates between orange and green on JobScape or TimeScape).	Refer to the procedure for **Handling a Hanging Execution Job** (subsequent section of this lesson).
“Execute” job fails (job has turned red on JobScape or TimeScape).	Refer to the procedure for **Handling a Failed Execution Job** (subsequent section of this lesson).
“Postprocess” job fails (job has turned red on JobScape or TimeScape).	Refer to the procedure for **Handling a Failed Postprocessing Job** (subsequent section of this lesson).
Both the “Execute” and “Postprocess” jobs fail (jobs have both turned red on JobScape or TimeScape).	Refer to the procedure for **Handling Failure of Both Execution and Postprocessing Jobs** (subsequent section of this lesson).
On-Demand Processing Request fails.	Refer to the procedure for **Handling a Failed On-Demand Processing Request** (subsequent section of this lesson).
Other problems.	Check the log files (e.g., EcDpPrJobMgmt.ALOG, EcDpPrJobMgmt.Debug.log, EcDpPrDeletion.ALOG, *DPR#*.ALOG, *DPR#*.err, etc.) in the /usr/ecs/*MODE*/CUSTOM/logs directory for error messages. [For detailed instructions refer to the procedure for **Checking Log Files** (subsequent section of this lesson).]

Production Processing Troubleshooting Procedures

The following procedures for correcting Production Processing problems are provided in this section:

- Checking AutoSys Status
- Checking the AutoSys Log
- Checking Job Management Server Status
- Checking to Determine Whether the DPR Is Waiting in the AutoSys Queue
- Checking to Determine Whether AutoSys Is Full
- Responding to a Condition Where a DPR Was Released But Failed Due to a JIL Failure
- Handling Subscription Server Problems
- Responding to a DPR That Was Released But Failed Due to an AutoSys ID Failure
- Responding to a DPR That Was Released But Failed Due to Invalid DPR
- Responding to a DPR That Was Released But Failed to Be Received by Job Management Server
- Handling a Hanging Allocation Function
- Running Execution Management Outside of AutoSys
- Handling a Failed Allocation Function
- Force-Starting a Job
- Responding to a Restart of a Job That Fails Although All Known Problems Have Been Corrected
- Handling a Failed Staging Function
- Cleaning Up the DPS File Tables
- Handling a Failed Preprocessing Job
- Responding to Execution Job That Has Failed and the DPR Has Gone into "Failed-PGE" Processing
- Handling a Failed Postprocessing Job
- Handling Failure of Both Execution and Postprocessing Jobs
- Handling a Failed Insertion Function

- Handling a Failed Deallocate Function
- Responding to a DPR that Failed in OdMgr because the PGE ID Could Not Be Found
- Checking Log Files

Processing Problems

This section contains information concerning the following processing problems:

- The entire processing system is hanging
- Jobs are activated, but do not get started in AutoSys.
- Single job has failed or is hanging.
- An On-Demand Processing Request fails.

Responding to Hanging of the Processing System

If the entire processing system is hanging (if no jobs are changing state), it is probably due to one of the following conditions:

- AutoSys is not functional (e.g., the AutoSys event processor is not running).
- Database is deadlocked (preceding section of this lesson).

The response is to perform the following two procedures (as necessary):

- Checking AutoSys Status.
- Checking the AutoSys Log.

Checking AutoSys Status

Like any other program AutoSys can crash or experience connectivity problems between its server and its clients.

The procedure for checking AutoSys status starts with the assumption that the applicable servers are running and the Production Monitor has logged in to the system.

Checking AutoSys Status

---

1 Access a terminal window logged in to the Queuing Server host.

- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 If the AutoSys .csh file has not already been sourced, type **cd /*path*** then press **Return/Enter**.

- Change directory to the directory (e.g., /usr/ecs/*MODE*/COTS/autotreeb/autouser, /usr/ecs/*MODE*/COTS/autotree/autouser, /data1/SHARED/COTS/autotree/autouser) containing the set-up files (e.g., FMR.autosys.csh.x0sps02).
- The particular path to be typed may vary from site to site.

3 Type **source *AUTOSYSINSTANCE*.autosys.csh.*hostname*** then press **Return/Enter**.

- An ***AUTOSYSINSTANCE*** (also called an AUTOSERV instance) is installed as part of the Data Processing Subsystem and is identified by three capital letters.
  - Examples of AUTOSYS (AUTOSERV) instances at DAACs include **FMR** and **SPG.**
  - Multiple AUTOSYS instances may be installed at a DAAC.

4 Type **autoping -m ALL** then press **Return/Enter**.

- The following type of message is displayed:

  AutoPinging Machine [x0spg11]
  AutoPing WAS SUCCESSFUL!

  AutoPinging Machine [x0spg07]
  AutoPing WAS SUCCESSFUL!

  AutoPinging Machine [x0sps06]
  AutoPing WAS SUCCESSFUL!

  - In the example the statements **"AutoPing WAS SUCCESSFUL!"** indicate that the server and client machines are properly configured and are communicating successfully.
- To check all machines and verify their database access, type **autoping -m ALL -D** then press **Return/Enter**.
  - If successful, the following type of message is displayed:

    AutoPinging Machine [x0spg11] AND checking the Remote Agent's DB Access.
    AutoPing WAS SUCCESSFUL!

    […]

5 If the results of the **autoping** command (Step 4) indicated anything other than a SUCCESSFUL response from any machine(s) [host(s)], notify the Operations Controller/System Administrator to have the affected host(s) brought back up.

6 Type **chk_auto_up** then press **Return/Enter**.

- The following type of message is displayed:

  Attempting (1) to Connect with Database: x0sps02_srvr:FMR

  ***** Have Connected successfully with Database: x0sps02_srvr:FMR. *****

  Connected with Event Server: x0sps02_srvr:FMR

  Checking Machine: x0sps02
  Primary Event Processor is RUNNING on machine: x0sps02

  Checking Machine: x0spg01
  No Event Processor is RUNNING on machine: x0spg01

  Checking Machine: x0ais01
  No Event Processor is RUNNING on machine: x0ais01

  – In the example the statements **"Have Connected successfully with Database: x0sps02_srvr:FMR"** and "**Connected with Event Server: x0sps02_srvr:FMR**" indicate that the AutoSys Event Server (database server) is running and a connection has been made with the appropriate AutoSys database.

  – In the example the statement **"Primary Event Processor is RUNNING on machine: x0sps02"** indicates that the Primary Event Processor is running on the Queuing Server (as it should).

7 If the Primary Event Processor is **not** running, either notify the Operations Controller/System Administrator to have it brought back up or (if authorized to do so) type **eventor** then press **Return/Enter**.

- The AutoSys **eventor** command starts the Primary Event Processor.

  – First **eventor** ensures that there is no other Event Processor of the same instance (e.g., FMR) running on the machine where the instance is being started.

- Then **eventor** runs the **chase** command, which inspects the database to determine which jobs are supposed to be running and checks each machine to verify that the jobs are there.
  - If it detects problems, **chase** sends alarms and/or failure events (depending on the options specified) for any missing jobs.
  - If the missing jobs can be restarted, they are automatically restarted.

- If the Primary Event Processor does not stay up (e.g., it is brought up and it goes down right away) one of the following problems may be occurring:
  - It may be possible that too many events were queued up to AutoSys while it was down. If AutoSys detects a certain number of events in a short time period, it brings itself down. The only way to handle this is to keep bringing AutoSys back up. Each time it will work through a few of the events before it detects "too many" and shuts down. Eventually the events will be cleared out and AutoSys will stay up.
  - It may be that the Sybase ASE server for AutoSys (the Event Server) is not up.
    - The **chk_auto_up** command would determine the status of the Event Server.

8 If the Event Processor is running, check for database-related error messages in the AutoSys log as described in the procedure for **Checking the AutoSys Log** (subsequent section of this lesson) or when attempting to bring up **JobScape**.

- For example:

  Couldn't create DBPROCES
  Unable to get encoded and plaintext passwords for x0sps02_srvr:FMR

  - The error messages in the example indicate that the AutoSys Event Server (database server) may not be up.

9 If the results of the **chk_auto_up** command (Step 6) indicated that multiple Primary Event Processors were running (for the same AUTOSERV instance), either notify the Operations Controller/System Administrator to have the Primary Event Processors stopped and a fresh instance of the Primary Event Processor started or (if authorized to do so) type **sendevent -E STOP_DEMON** then press **Return/Enter**.

- An alternative method of sending a STOP_DEMON event to an Event Processor is to use the **Send Event** GUI.
  - For detailed instructions refer to the procedure for **Sending an Event to a Job from the Send Event GUI** (subsequent section of this lesson).

10 If a STOP_DEMON event was sent to stop the Event Processor(s) and if authorized to do so, type **eventor** then press **Return/Enter**.

- The AutoSys **eventor** command starts the Primary Event Processor.
- If not authorized to send the **eventor** command, wait for the Operations Controller/System Administrator to do so.

11 If the results of the **chk_auto_up** command (Step 6) indicated that the Event Server (database server) is **not** running, notify the Database Administrator to have it brought back up.

12 If the Event Processor or Event Server had to be started or restarted, return to Step 4.

13 If no Event Processor or Event Server problems were detected, check the AutoSys log.

- For detailed instructions refer to the procedure for **Checking the AutoSys Log** (subsequent section of this lesson).

Checking the AutoSys Log

The AutoSys event demon log tells how a DPR has progressed through AutoSys, showing failures and force-starts of jobs.

The procedure that follows describes the use of the UNIX **grep** command on the DPR Id in the event demon log file. An alternative is to use the **vi** command to view the full log, which contains timestamps. Another alternative is to request either a Summary Report or an Event Report from the Job Activity Console (Ops Console) as described in the procedure for **Monitoring/Controlling Job Processing** (previous section of this lesson).

The procedure for checking the AutoSys event demon log starts with the assumption that the applicable servers are running and the Production Monitor has logged in to the system.

Checking the AutoSys Log

---

1 Access a terminal window logged in to the Queuing Server host.

- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 Type **cd /*path*** then press **Return/Enter**.

- Change directory to the directory (e.g., /usr/ecs/*MODE*/COTS/autotreeb/autouser/out, /usr/ecs/*MODE*/COTS/autotree/autouser/out, /data1/SHARED/COTS/autotree/autouser/out) containing the event demon log file (e.g., event_demon.FMR).
- The AutoSys event demon log is named event_demon.*AutoSys_Instance*.
  - A typical *AutoSys_Instance* at a DAAC is **FMR**.
- The particular path to be typed may vary from site to site.

3 Type **grep *jobname* event_demon.*AutoSys_Instance*** then press **Return/Enter**.

- If there were no problems, the results should appear as follows:

  grep MoPGE02#2014193500OPS event_demon.FMR

  EVENT: STARTJOB JOB: MoPGE02#2014193500OPS
  EVENT: CHANGE_STATUS STATUS: RUNNING JOB: MoPGE02#2014193500OPS
  EVENT: CHANGE_STATUS STATUS: STARTING JOB: MoPGE02#2014193500OPSR
  EVENT: CHANGE_STATUS STATUS: RUNNING JOB: MoPGE02#2014193500OPSR
  EVENT: CHANGE_STATUS STATUS: SUCCESS JOB: MoPGE02#2014193500OPSR
  EVENT: CHANGE_STATUS STATUS: STARTING JOB: MoPGE02#2014193500OPSE
  EVENT: CHANGE_STATUS STATUS: RUNNING JOB: MoPGE02#2014193500OPSE
  EVENT: CHANGE_STATUS STATUS: SUCCESS JOB: MoPGE02#2014193500OPSE
  EVENT: CHANGE_STATUS STATUS: STARTING JOB: MoPGE02#2014193500OPSP
  EVENT: CHANGE_STATUS STATUS: RUNNING JOB: MoPGE02#2014193500OPSP
  EVENT: CHANGE_STATUS STATUS: SUCCESS JOB: MoPGE02#2014193500OPSP
  EVENT: CHANGE_STATUS STATUS: STARTING JOB:
  EVENT: CHANGE_STATUS STATUS: SUCCESS JOB: MoPGE02#2014193500OPS

- When there are no problems, each command job goes through the following changes of status: STARTING, RUNNING, SUCCESS.
- If there are problems, something similar to the following results may be obtained:

  grep MoPGE02#2014193500OPS event_demon.FMR

  EVENT: STARTJOB JOB: MoPGE02#2014193500OPS
  EVENT: CHANGE_STATUS STATUS: RUNNING JOB: MoPGE02#2014193500OPS
  EVENT: CHANGE_STATUS STATUS: STARTING JOB: MoPGE02#2014193500OPSR
  EVENT: CHANGE_STATUS STATUS: RUNNING JOB: MoPGE02#2014193500OPSR
  EVENT: CHANGE_STATUS STATUS: FAILURE JOB: MoPGE02#2014193500OPSR
  EVENT: ALARM ALARM: JOBFAILURE JOB: MoPGE02#2014193500OPSR
  EVENT: FORCE_STARTJOB JOB: MoPGE02#2014193500OPSR
  EVENT: CHANGE_STATUS STATUS: STARTING JOB: MoPGE02#2014193500OPSR
  EVENT: CHANGE_STATUS STATUS: RUNNING JOB: MoPGE02#2014193500OPSR
  EVENT: CHANGE_STATUS STATUS: SUCCESS JOB: MoPGE02#2014193500OPSR
  ......

- The preceding job had some failures and a force-start.

4 If the AutoSys event log does not indicate any problems, perform the procedure for **Checking for Database Deadlocks** (preceding section of this lesson).

Responding to Failure of Jobs to Start in AutoSys

The following events trigger the Job Management Server to check its queue, check information in the limits tables (i.e., DpPrPgeLimits and DpPrClassSchedulingLimits), and ascertain which job should be placed in AutoSys next:

- DPR is released by the Planning Workbench.
- Subscription Manager sets a DPR completionState (PlDataProcessingRequest table in the PDPS database) to PENDING.
- Job finishes in AutoSys.

- Job is cancelled through the Production Request Editor.

On very rare occasions, it is possible for DPS processing to freeze up because no triggering events can occur. In such cases the Production Monitor can use the Job Management Client to "wake up" (trigger) the Job Management Server.

Subscriptions are processed and cause jobs to be released into AutoSys by means of the following process:

- An ESDT is "registered" to the Science Data Server (SDSRV). The ESDT information includes three events (insert, delete, and update metadata) and a datatype. It is possible to enter a subscription for any of the events.
- The Production Request Editor (PRE) sends the datatype short name and version ID, the action type (e.g., insert) and the Subscription Manager (SubsMgr) name to the Subscription Server (SubsSrv) in order to register a subscription. The value in the subscriptionFlag column (PlDataTypeMaster table in the PDPS database) for the data type is updated.
- When an “insert” event (or a “delete” event or an “update metadata” event) occurs, SDSRV sends notification to SubsSrv, which sends the Subscription ID to the PLS Subscription Manager. The Subscription Manager is identified by name in the SubsSrv database in connection with the Subscription ID.
- Subscription Manager gets the UR for the inserted granule from the SDSRV and updates the UR information in the PlDataGranule table in the PDPS database. In the simple case (e.g., for Production Requests that do not require optional inputs or alternate inputs) SubsMgr checks to see if all of the datatypes in PlDprData that have an ioFlag of 0 (input) for the DPR are present in the data archive and can be acquired. If this is the case, SubsMgr sends a ReleaseDprJob request to the Job Management Server to release the job into AutoSys.

Jobs that are activated may not get started in AutoSys for any of the following reasons:

- Job Management Server is down.
- DPR is waiting in the AutoSys queue (never got released).
- DPR was released but failed due to an AutoSys ID failure.
- DPR was released but failed due to invalid DPR.
- DPR was released but was not received by the Job Management Server.

To determine whether the Job Management Server is down or a DPR is waiting in the AutoSys queue (having never been released), perform the following procedures (as necessary):

- **Checking Job Management Server Status**.
- **Checking to Determine Whether the DPR Is Waiting in the AutoSys Queue**.

- **Using ISQL to Check Database Tables**.
- **Checking to Determine Whether AutoSys Is Full**.
- **Responding to a Condition Where a DPR Was Released But Failed Due to a JIL Failure**.
- **Handling Subscription Server Problems**.

To determine whether a DPR was released but failed due to an AutoSys ID failure, perform the procedure for **Responding to a DPR That Was Released But Failed Due to an AutoSys ID Failure**.

To determine whether a DPR was released but failed due to an invalid DPR, perform the procedure for **Responding to a DPR That Was Released But Failed Due to Invalid DPR**.

To determine whether a DPR was released but was not received by the Job Management Server, perform the procedure for **Responding to a DPR That Was Released But Failed to Be Received by Job Management Server**.

Checking Job Management Server Status

If jobs that are activated do not get started in AutoSys, it may be because the Job Management Server is down. Consequently, one of the first steps in investigating why jobs do not get started in AutoSys is to check the status of the Job Management Server.

The procedure for checking Job Management Server status starts with the assumption that the Production Monitor has logged in to the system.

Checking Job Management Server Status

---

1 Access a terminal window logged in to the Queuing Server host.

- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 Ensure (e.g., by typing **ps -ef | grep EcDpPrJobMgmt**) that the Job Management Server (EcDpPrJobMgmt) is "up."

- The following type of response is displayed:

  cmshared 110 1 0 09:21:30 ? 0:05
  /usr/ecs/TS2/CUSTOM/bin/DPS/EcDpPrJobMgmt ConfigFile
  /usr/ecs/TS2/CUSTOM/cfg/Ec

cmshared 3594   1 0   Jun 17 ?   3:02 /usr/ecs/OPS/CUSTOM/bin/DPS/EcDpPrJobMgmt ConfigFile /usr/ecs/OPS/CUSTOM/cfg/Ec

cmshared 16104 15434 0 13:08:36 pts/13   0:00 grep EcDpPrJobMgmt

– The preceding example indicates that the Job Management Server is running in TS1 mode and OPS mode.

3 If the server has gone down, notify the Operations Controller/System Administrator to have the server brought back up.

4 If the Job Management Server (EcDpPrJobMgmt) is "up," continue with the procedure for **Checking to Determine Whether the DPR Is Waiting in the AutoSys Queue** (subsequent section of this lesson).

Checking to Determine Whether the DPR Is Waiting in the AutoSys Queue (Never Got Released)

The Job Management Server may have never received a ReleaseDprJob command from the PLS Subscription Manager. As a result the job would wait in the AutoSys queue and would not be able to start processing.

The procedure for checking to determine whether the DPR is waiting in the AutoSys queue starts with the assumption that the Production Monitor has logged in to the system.

Checking to Determine Whether the DPR Is Waiting in the AutoSys Queue

---

1 Manually trigger the release of DPRs.

- For detailed instructions refer to the procedure for **Performing Job Management Client Functions** (preceding section of this lesson).

2 Check for job activation in AutoSys.

- For detailed instructions refer to the procedure for **Monitoring/Controlling Job Processing** (preceding section of this lesson).
- End of procedure if the job was activated in AutoSys.

3 If the job was not activated in AutoSys, access a terminal window logged in to the Queuing Server host.

- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.

- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

4 If the job was not activated in AutoSys, perform the procedure for **Using ISQL to Check Database Tables** (previous section of this lesson) or use a database browser to check for an entry for the job (by searching on the **dprId** column) in the **DpPrCreationQueue** table in the applicable PDPS database.

- For example:

```
x0sps02{cmshared}43: isql -U pdps_role –S x0sps02_srvr
Password:
1> use pdps
2> go
1> select * from DpPrCreationQueue
2> go
 dprId                        autosysId            priority    hold
 ---------------------------- -------------------- ----------- ----
 ETS#syn1#014020000OPS        FMR                          250    1
 ETS#syn1#014020010OPS        FMR                          250    1
 ETS#syn1#014020020OPS        FMR                          250    1
 ETS#syn1#014020030OPS        FMR                          250    1
 ETS#syn1#014020040OPS        FMR                          250    1
 ETS#syn1#014020050OPS        FMR                          250    1
 ETS#syn1#014020100OPS        FMR                          250    1
 ETS#syn1#014020110OPS        FMR                          250    1
 ETS#syn1#014020130OPS        FMR                          250    1

 (9 rows affected)
```

- If the job is listed in the **DpPrCreationQueue** table, it probably never got a ReleaseDprJob command from the PLS Subscription Manager (unless AutoSys is full). (Refer to the procedure for **Checking to Determine Whether AutoSys Is Full,** which is a subsequent section of this lesson.)

5 In the terminal window logged in to the Queuing Server (e.g., e0sps04 or l0sps03) type **cd /usr/ecs/*MODE*/CUSTOM/logs** then press **Return/Enter**.

- Change directory to the directory containing the log files (e.g., EcDpPrJobMgmt.ALOG, EcDpPrDeletion.ALOG, *DPR#*.ALOG, *DPR#*.err, EcPlSubMgrDebug.log, EcDpPrJobMgmtDebug.log).

6 Type **pg EcDpPrJobMgmtDebug.log** then press **Return/Enter**.

- The first page of the log file is displayed.

- Although this procedure has been written for the **pg** command, any UNIX editor or visualizing command (e.g., **vi**, **view**, **more**) can be used to review the log file.

7 Review the Job Management debug log file for an entry indicating that Job Management received the ReleaseDprJob command for the job.

- For example:

  03/30/00 11:38:09: ---

  DpPrScheduler_1_0_Mgr::ReleaseDprJob(dpr_id dpr) CALLED.
  ETS#syn1#004130123OPS

  03/30/00 11:38:09: In DpPrScheduler::ReleaseDprJob, dpr=
  ETS#syn1#004130123OPS
  03/30/00 11:38:09: DpPrCreationQueue::SetDprHoldStatus, dprId=
  ETS#syn1#004130123OPS
  DpPrCreationQueue::SetDprHoldStatus, autoSysId= VAT
  DpPrCreationQueue::SetDprHoldStatus, holdStatus= 0
  03/30/00 11:38:09: DpPrCreationQueue::HasAutosysId, autosysId= 0xee4534d8
  03/30/00 11:38:09: DpPrCreationQueue::HasAutosysId, autosysId= 0xee4534d8
  03/30/00 11:38:09: removed ETS#syn1#004130123OPS
  there are now 0entries on this queue
  queue priority of this node is now 250

- If Job Management received the ReleaseDprJob command for the job, there may have been a JIL (AutoSys Job Information Language) processor problem. [Refer to **Responding to a Condition Where a DPR Was Released But Failed Due to a JIL Failure** (subsequent section of this lesson).]

- If there is no evidence that Job Management received the ReleaseDprJob command for the job, the PLS Subscription Manager did not send the command.

  – Subscription Manager does not send the ReleaseDprJob command unless it thinks that all of the DPR's required inputs have been received.

8 If the DPR is a regular one (e.g., with no alternate or optional inputs), perform the procedure for **Using ISQL to Check Database Tables** (previous section of this lesson) or use a database browser to check the **PlDprData** table and find all of the **granuleId** column entries with an **ioFlag** column entry of 0 (an input granule) for the DPR (**dprId** column).

- For example:

  x0sps02{cmshared}43: isql -U pdps_role –S x0sps02_srvr
  Password:

```
1> use pdps
2> go
1> select dprId,granuleId,ioFlag from PlDprData where dprId like
"ETS#syn1#004130123OPS"
2> go
 dprId
        granuleId
        ioFlag
 -----------------------------
       ----------------------------------------------------------------------------------------------------
---------------------------------
       -----------
 ETS#syn1#004130123OPS
        AST_05#00102141998020120000
                1
 ETS#syn1#004130123OPS
        AST_08#00102141998020120000
                1
 ETS#syn1#004130123OPS
        AST_09T#00102141998020120000
                0
 ETS#syn1#004130123OPS
        AST_ANC#001L1004
                0

(4 rows affected)
```

- In the preceding example there are four **granuleId** column entries for the DPR (ETS#syn1#004130123OPS); two have an **ioFlag** column entry of 0 (an input granule) and two have an **ioFlag** column entry of 1 (output granule).

9 In the **PlDataGranule** table observe the **universalReference** column for each applicable **granuleId**.

- If all of the input granules have URs (as opposed to granuleId), the Subscription Manager *should* have sent a ReleaseDprJob command to Job Management.
- To check the preceding example observe the entries for the following granuleId:
  - AST_09T#00102141998020120000
  - AST_ANC#001L1004

- For example:

  1> select universalReference from PlDataGranule where granuleId like "AST_09T#00102141998020120000"
  2> go

  universalReference

  --

  UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]:19:SC:AST_09T.001:5672

  (1 row affected)

- In the example there is a universalReference for the granuleId. The same search should be performed for the other granuleId.

10 Perform the procedure for **Checking Log Files** (subsequent section of this lesson) to search the Subscription Manager debug log file (EcPlSubMgrDebug.log) for an entry indicating that Subscription Manager sent the ReleaseDprJob command for the job to Job Management.

- Subscription Manager log files are on the Queuing Server host (e.g., e0sps04 or l0sps03).

- For example:

  DpPrSchedulerProxy::ReleaseDprJob :
  ETS#syn1#004130123OPS
  03/30/00 11:37:07: Destroying DpPrSchedulerProxy object

11 Search the Subscription Manager debug log file for subscription notification from the Subscription Server concerning dynamic data that the DPR needs.

- For example:

  03/30/00 11:36:58: ***** Begining of PlSubMsgCb::HandleCbMsg() *******
  03/30/00 11:36:58: Entire message = Subscription Notification:: UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]:19:SC:AST_09T.001:5478

  ESDT Information: AST_09T.001:INSERT

  User Information: SubsMgr

  EventID: 805

Subscription ID: 82

Qualifier List:
UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]:19:SC:AST_09T.001:5478 AsterGranule 03/30/2000 16:36:27 Day This is a quality flag Passed 30 20 GuruTej 1 AST_09T
UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]:19:SC:AST_L1B.001:5400
UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]:19:SC:AST_ANC.001:5369 (((90.0000, -180.0000), (90.0000, 180.0000), (-90.0000, 180.0000), (-90.0000, -180.0000))) 13:01:23.000000Z 07/04/1997 1.0 5478 SC AST_09T.001 AST_09T#00107041997130123000000
:SC:AST_09T.001:5478:1.HDF-EOS 0 40367 DRP1_OPS:AST_09T.001 1 None 0.04036699980497360
…

- The preceding example shows subscription notification for a granule of AST_09T (input for ETS) that has been inserted into the archive.

12 If there is no Subscription Server notification to Subscription Manager or if it seems likely that all of the necessary input files for the DPR have been inserted by another DPR, there may be Subscription Server problems. [Refer to the procedure for **Handling Subscription Server Problems** (subsequent section of this lesson).]

13 If there are no Subscription Server Problems, all of the input granules for the DPR have URs, and/or Subscription Manager received notification for all dynamic granules, notify the Operations Controller/System Administrator that there may be a problem with the Subscription Manager.

Checking to Determine Whether AutoSys Is Full

This is an unlikely problem and would occur only when the DPR completionState in the PlDataProcessingRequest database table is CQ_RELEASE.

The procedure for checking to determine whether AutoSys is full starts with the assumption that the Production Monitor has logged in to the Queuing Server (e.g., e0sps04 or l0sps03) and started the **AutoSys GUI Control Panel** and **JobScape**.

Checking to Determine Whether AutoSys Is Full

1 Perform the procedure for **Using ISQL to Check Database Tables** (previous section of this lesson) or use a database browser to determine the value in the **completionState** column for the job (by searching on the **dprId** column) in the **PlDataProcessingRequest** table in the applicable PDPS database.

- For example:

```
x0sps02{cmshared}43: isql -U pdps_role –S x0sps02_srvr
Password:
1> use pdps
2> go
1> select dprId,completionState from PlDataProcessingRequest where dprId
like "ETS#syn1#014020000OPS"
2> go
 dprId                    completionState
 ---------------------------- ---------------
 ETS#syn1#014020000OPS        CQ_HOLD

 (1 row affected)
```

2 If the value in the **completionState** column for the DPR in the PDPS database **PlDataProcessingRequest** table is "CQ_RELEASE" wait for a DPR to finish, so that the next waiting one can be put into AutoSys.

- The Job Management Server got the command from Subscription Manager to release the job but AutoSys cannot accommodate any more jobs at present.

3 If the value in the **completionState** column for the DPR in the PDPS database PlDataProcessingRequest table is "JIL_FAILUR," go to the procedure for **Responding to a Condition Where a DPR Was Released But Failed Due to a JIL Failure**.

Responding to a Condition Where a DPR Was Released But Failed Due to a JIL Failure

A "JIL Failure" means that the Job Management Server had some problem placing the DPR in AutoSys. The Job Interface Language (JIL) processor rejected the “create job” command sent to it by the Job Management Server. The principal reasons for a JIL failure are as follows:

- There is already a job with an identical name in AutoSys.

- The AutoSys event processor is down. (Refer to **Checking AutoSys Status** (preceding section of this lesson).)
- The job had a problem when it was loaded into AutoSys and a malformed or mutant job box is the result.

The procedure for responding to a condition where a DPR was released but failed due to a JIL failure starts with the assumption that the Production Monitor has logged in to the Queuing Server (e.g., e0sps04 or l0sps03) and started the **AutoSys GUI Control Panel**.

Responding to a Condition Where a DPR Was Released But Failed Due to a JIL Failure

---

1 Check whether there is already a job with an identical name in AutoSys using the procedure for **Monitoring/Controlling Job Processing** (preceding section of this lesson).

- When specifying job selection criteria (refer to the procedure for **Specifying Job Selection Criteria** (preceding section of this lesson)) type a portion of the job name in the "Job Name" box, bracketed by the "*" or "%" wildcard character.

2 If there is a job with an identical name already in AutoSys, either request the Production Planner to delete it using the Production Request Editor or delete the job using the Job Management Client tool (refer to the procedure for **Performing Job Management Client Functions** (preceding section of this lesson)).

- Jobs should **not** be deleted using the AutoSys **Job Definition** GUI because it does not communicate with the PDPS database.

3 If there is not a job with an identical name already in AutoSys, observe the characteristics of the job box in **JobScape** using the procedure for **Monitoring/Controlling Job Processing** (preceding section of this lesson).

- If the job box is malformed or mutant, it will stay dark blue (meaning that it was not activated) and may be missing one of the three job steps.

4 If the job box is malformed or mutant, in **JobScape** place the mouse cursor on the job and click and hold the **right** mouse button.

- Pop-up menu appears.

5 If the job box is malformed or mutant, select (highlight) **Job Definition** from the pop-up menu (release the right mouse button).

- The **Job Definition** GUI (Figure 73) is displayed.

6 If the job box is malformed or mutant, click on the **Delete** button.

NOTE: In general, it is bad practice to delete a job from AutoSys using the AutoSys **Job Definition** GUI because the AutoSys database and PDPS database lose their synchronization. However, there is no other solution in this case and the PDPS database must be updated manually.

7 To exit from the **Job Definition** GUI, click on the **Exit** button.

- The **Job Definition** GUI is dismissed.

8 Log in to the appropriate PDPS database as described in Steps 1 through 5 of the procedure for **Using ISQL to Check Database Tables** (previous section of this lesson).

9 Update the **completionState** of the DPR in the **PlDataProcessingRequest** table of the PDPS database for which the mutant job box was created.

- Set **completionState** equal to NULL using the update command.
- For example:

  1> update PlDataProcessingRequest set completionState=”” where dprId like "MoPGE02#2014193500OPS"
  2> go

10 Request the Production Planner to delete (using the procedure for **Deleting a Data Processing Request**) the DPR that maps to the job then recreate the DPR and any subsequent DPRs.

- Only the DPR that had the mutant job box and any DPRs that depend on it have to be deleted. It may not be necessary to delete entire production requests.

Handling Subscription Server Problems

Handling Subscription Server problems involves determining whether the Subscription Manager is getting notification from Subscription Server after a dynamic granule has been inserted. If no notification is received, the Subscription Manager does not send a ReleaseDprJob request to the Job Management Server to release the affected job(s) into AutoSys. So the job(s) is (are) not processed.

The procedure for handling Subscription Server problems starts with the assumption that the Production Monitor has logged in to the system.

1 Access a terminal window logged in to the Subscription Server host.

- Examples of Subscription Server host (Sun Consolidation Internal Server host) names include **e0acs11** and **l0acs03**.
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 Log in to the appropriate Subscription Server database as described in Steps 1 through 5 of the procedure for **Using ISQL to Check Database Tables** (previous section of this lesson).

- For example:

  x0acs06{cmshared}45: isql -U css_role –S x0acs06_srvr
  Password:
  1> use SubServer
  2> go

- If the Subscription Manager debug log file was previously searched for subscription notification from the Subscription Server concerning dynamic data that the DPR needs, the following types of information will have been discovered (if not for the specific granule required, at least for the datatype):
  - ESDT Information (data type and event).
  - User (i.e., Subscription Manager).
  - Event ID.
- For example:

  03/30/00 11:36:58: Entire message = Subscription Notification:: UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]:19:SC:AST_09T.001:5478

  ESDT Information: AST_09T.001:INSERT

  User Information: SubsMgr

  EventID: 805

3 Type **select * from EcSbSubscription where eventID=*n*** at the **1>** prompt then press **Return/Enter**.

- The ***n*** represents the relevant **eventID** (e.g., from the Subscription Manager debug log file).

4 Type **go** at the **2>** prompt then press **Return/Enter**.

- Table contents are displayed.
- For example:

```
1> select * from EcSbSubscription where eventID=805
2> go
 subID       eventID     userID
         expDate
         object
 ----------- ----------- ------------------------------
        --------------------------
        -----------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------
-------
          82         805 SubsMgr
                Jan 18 2001 12:00AM
         82
805
7
"SubsMgr"26
"Subscription\x20Notification:"0
""17
"SubscriptionQueue"4
"XDC"2451563
2451928
13
"EcSbGenActio
n":16386
\[9
"UnnamedPL"0
"":32808
\[0
\]\]:16386
\[9
```

"UnnamedPL"0
"":32808
\[0
\]\]

(1 row affected)

- In the example note that **subID** 82 is entered for **eventID** 805 (AST_09T.001:INSERT) and the **userID** is SubsMgr.

5 To exit from isql type **quit** at the **1>** prompt then press **Return/Enter**.

6 Perform the procedure for **Checking Log Files** (subsequent section of this lesson) to search the .err file for the DPR that inserted data (ACT#syn1#004130123OPS.err) for an entry indicating that the data were in fact inserted.

- The .err log files containing output file insertion information are on the Queuing Server (e.g., e0sps04 or l0sps03).

- For example:

03/30/00 11:38:02:
DpPrDSSInterface::CheckStatusParameters The request results returned from request.GetStatus() is :
-ReqUpdate[CmdCount(1) ReqSuccess(1)]
03/30/00 11:38:02: Request status indicates success
03/30/00 11:38:02: DpPrDSSInterface::CheckResultParameters The request results returned from request.GetResults() is :
-ReqResults[
--CmdResults[
---Insert results[

DATAFILEGROUP[userDataFile(/usr/ecs/OPS/CUSTOM/pdps/x0spg01/data//DpPrRm/x0spg01_disk/AST_09T#00107041997130123000000) ESDTStatus(1) archiveDescription(None)]
UR(UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]:19:SC:AST_09T.001:5478)] CmdSuccess(1)]]
03/30/00 11:38:02: DpPrDSSInterface::RequestOK, The request results returned from this method is :
-ReqResults[
--CmdResults[
---Insert results[

DATAFILEGROUP[userDataFile(/usr/ecs/OPS/CUSTOM/pdps/x0spg01/data//DpPrRm/x0spg01_disk/AST_09T#00107041997130123000000) ESDTStatus(1) archiveDescription(None)]
UR(UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]:19:SC:AST_09T.001:5478)] CmdSuccess(1)]]
03/30/00 11:38:02: ~~~~ RPC ID completed ACT#syn1#004130123OPSAST_09T
03/30/00 11:38:02: DpPrDSSInterface::~DpPrDSSInterface()
03/30/00 11:38:02: inserted UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]:19:SC:AST_09T.001:5478 into ursVector =
ursVector.length()= 1
UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]:19:SC:AST_09T.001:5478
03/30/00 11:38:02: About to update PlDataGranule granuleId = AST_09T#00107041997130123000
03/30/00 11:38:02: Successfully updated PlDataGranule granuleId = AST_09T#00107041997130123000

7 In the terminal window logged in to the Subscription Server host (Sun Consolidation Internal Server host; e.g., x0acs11) type **cd /usr/ecs/*MODE*/CUSTOM/logs** then press **Return/Enter**.

- Change directory to the directory containing the log files (e.g., EcSbSubServer.ALOG, EcSbSubServerDebug.log).

8 Type **pg EcSbSubServer.ALOG** then press **Return/Enter**.

- The first page of the log file is displayed.
- Although this procedure has been written for the **pg** command, any UNIX editor or visualizing command (e.g., **vi**, **view**, **more**) can be used to review the log file.

9 Correlate the data insertion time (as specified in the .err file for the DPR that inserted data) with activity in the Subscription Server ALOG file:

- For example, the following entries are in the Subscription Server ALOG file around AST_09T data insertion time of 11:38 (as discovered in the ACT#syn1#004130123OPS.err log file):

Msg: Getting event for EventID = 805 Priority: 0 Time : 03/30/00 11:38:07
PID : 25042:MsgLink :0 meaningfulname :DsDbInterface::Connect()
Msg: Connectted to server: Priority: 2 Time : 03/30/00 11:38:07
PID : 25042:MsgLink :0 meaningfulname :EcSbTriggerEventRequestTriggerTrigger
Msg: Triggering event for EventID = 805 Priority: 0 Time : 03/30/00 11:38:08

PID : 25042:MsgLink :0 meaningfulname :EcSbTriggerEventRequestTriggerTrigger0
Msg: Firing subscriptions for event #805 Priority: 0 Time : 03/30/00 11:38:08
PID : 25042:MsgLink :0 meaningfulname :DsDbInterface::Connect()
Msg: Connectted to server: Priority: 2 Time : 03/30/00 11:38:08
PID : 25042:MsgLink :0 meaningfulname :EcSbSubscriptionExecuteExecute
Msg: No action specified. Priority: 0 Time : 03/30/00 11:38:08
PID : 25042:MsgLink :0 meaningfulname :EcSbSubscriptionExecuteExecute2
Msg: Email notification sent Priority: 0 Time : 03/30/00 11:38:08
PID : 25042:MsgLink :0 meaningfulname :EcSbSubscriptionExecuteExecute
Msg: No action specified. Priority: 0 Time : 03/30/00 11:38:08
PID : 25042:MsgLink :0 meaningfulname :EcSbSubscriptionExecuteExecute2
Msg: Email notification sent Priority: 0 Time : 03/30/00 11:38:08
PID : 25042:MsgLink :0 meaningfulname :EcMpMsgQueueOutInvokeInvoke
Msg: DCE Exception: Object not found (dce / rpc) Priority: 2 Time : 03/30/00 11:38:09
PID : 25042:MsgLink :0 meaningfulname :EcMpMsgQueueOutInvoke2
Msg: Exception: Unknown Priority: 2 Time : 03/30/00 11:38:09
PID : 25042:MsgLink :0 meaningfulname :EcSbGetEventRequestGetEventDataGetEventData
Msg: Getting event for EventID = 8 Priority: 0 Time : 03/30/00 11:38:16
PID : 25042:MsgLink :0 meaningfulname :DsDbInterface::Connect()
Msg: Connectted to server: Priority: 2 Time : 03/30/00 11:38:16
PID : 25042:MsgLink :0 meaningfulname :EcSbTriggerEventRequestTriggerTrigger
Msg: Triggering event for EventID = 8 Priority: 0 Time : 03/30/00 11:38:16
PID : 25042:MsgLink :0 meaningfulname :EcSbTriggerEventRequestTriggerTrigger0
Msg: Firing subscriptions for event #8 Priority: 0 Time : 03/30/00 11:38:16
PID : 25042:MsgLink :0 meaningfulname :DsDbInterface::Connect()
Msg: Connectted to server: Priority: 2 Time : 03/30/00 11:38:16

- In the example note that at 11:38 Subscription Server received Event 805 and recorded a log entry **"Msg: Firing subscriptions for event #805"** but that this did not include any event for SubsMgr for subId 82. Note, in particular:

  PID : 25042:MsgLink :0 meaningfulname :EcMpMsgQueueOutInvokeInvoke
  Msg: DCE Exception: Object not found (dce / rpc) Priority: 2 Time : 03/30/00 11:38:09

- In the example, it is clear that a file was inserted at 11:38, but that the Subscription Server never sent event notification to the PLS Subscription Manager.

10 If a Subscription Server problem has been identified, notify the Operations Controller/System Administrator of the problem.

11 If **no** Subscription Server problem has been identified, return to the procedure that specified Handling Subscription Server Problems [e.g., **Checking to Determine Whether the DPR Is Waiting in the AutoSys Queue** (preceding section of this lesson)].

Responding to a DPR That Was Released But Failed Due to an AutoSys ID Failure

An "AutoSys ID" failure occurs when the Job Management Server cannot associate the AutoSys ID with the DPR that was activated. When the Job Management Server is started, it reads various tables in the PDPS database that provide the linkage between processing resources and the AutoSys instance. If data is missing from the tables or was added after the Job Management Server was started, an "AutoSys ID" failure can occur when any jobs are activated by the Planning Workbench.

The procedure for responding to a DPR that was released but failed due to an AutoSys ID failure starts with the assumption that the Production Monitor has logged in to the system.

Responding to a DPR That Was Released But Failed Due to an AutoSys ID Failure

---

1 Access a terminal window logged in to the Queuing Server host.

- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 In the terminal window logged in to the Queuing Server type **cd /usr/ecs/*MODE*/CUSTOM/logs** then press **Return/Enter**.

- Change directory to the directory containing the log files (e.g., EcDpPrJobMgmt.ALOG, EcDpPrDeletion.ALOG, *DPR#*.ALOG, *DPR#*.err, EcPlSubMgrDebug.log, EcDpPrJobMgmtDebug.log).

3 Type **pg EcDpPrJobMgmt.ALOG** then press **Return/Enter**.

- The first page of the log file is displayed.
- Although this procedure has been written for the **pg** command, any UNIX editor or visualizing command (e.g., **vi**, **view**, **more**) can be used to review the log file.

4 Review the Job Management ALOG file for an **“unable to find autosys id”** message.

- For example:

  PID : 7668:MsgLink :0 meaningfulname
  :DpPrAutosysMapList::GetAutosysIDByDpr
  Msg: unable to find autosys id for dpr: ACT#syn1#004130123TS1 Priority: 2 Time : 03/09/001:33:51
  PID : 7668:MsgLink :9 meaningfulname :CantFindAutoSysId
  Msg: Unable to find autosys id Priority: 2 Time : 03/09/00 11:33:51
  PID : 7668:MsgLink :10 meaningfulname
  :DpPrSchedulerDObjSmainCreateFailed
  Msg: RqFailed=CreateDpr DprID=ACT#syn1#004130123TS1 Priority: 2 Time : 03/09/00 11:33:51

5 If there is an **“unable to find autosys id”** message in the Job Management ALOG file, log in to the appropriate PDPS database as described in Steps 1 through 5 of the procedure for **Using ISQL to Check Database Tables** (previous section of this lesson).

- For example:

  x0sps02:/usr/ecs/TS1/CUSTOM/[4] > isql -U pdps_role –S x0sps02_srvr
  Password:
  1> use pdps_TS1
  2> go

6 If there is an **“unable to find autosys id”** message in the Job Management ALOG file, verify that the **PlResource** table in the PDPS database has at least one entry for a processing string and at least one entry for an AutoSys Instance.

- For example:

```
1> select * from PlResource
2> go
 resourceId  resourceName
        resourceState resourceType    activityTypeId onLineState
 ----------- ------------------------------------------------------------
        ------------- --------------- -------------- -----------
           1 x0spg01_disk
                     1 DEVICE                   1          1
           2 x0spg01_vc
                     1 MACHINE                  1          1
           3 x0spg01
                     0 REALCOMP                 1          1
           4 x0spg01_string
```

```
        0 VIRTUAL              1        1
5 FMR
        0 AUTOSYS              1        1
6 x0aqg02_disk
        0 DEVICE               1        1
7 x0aqg02_vc
        0 MACHINE              1        1
8 x0aqg02
        0 REALCOMP             1        1
9 x0aqg02_string
        0 VIRTUAL              1        1
```

(9 rows affected)

- In the example resourceId 4 is a string (x0spg01_string) and resourceId 5 is an AutoSys Instance (FMR).

7 If the **PlResource** table in the PDPS database either has no entry for a processing string or no entry for an AutoSys Instance, make a request to the Resource Planner to create the necessary entry(ies).

8 Verify that the **PlRscString** table in the PDPS database has at least one entry and that **autosysIdKey** matches the entry in the **PlResource** table.

- For example:

```
1> select * from PlRscString
2> go
 stringId    stringName
        autosysIdKey
 ----------- ------------------------------------------------------------
        ------------
          4 x0spg01_string
                  5
          9 x0aqg02_string
                  5

 (2 rows affected)
```

- In the example the **PlRscString** table in the PDPS database has at least one entry and the **autosysIdKey** for each matches the entry (i.e., 5) in the **PlResource** table.

9 If the **PlRscString** table in the PDPS database either has no entry or if the **autosysIdKey** does not match the entry in the **PlResource** table, make a request to the Resource Planner to make the necessary adjustments.

10 Verify that the **DpPrAutosysMapList** table in the PDPS database has at least one entry and that **resourceString** and **autosysIdKey** match the entries in the **PlRscString** table.

- For example:

```
1> select * from DpPrAutosysMapList
2> go
 resourceString      autosysId          autosysIdKey
 ------------------- ------------------- ------------
 x0aqg02_string       FMR                      5
 x0spg01_string       FMR                      5

(2 rows affected)
```

- In the example the **DpPrAutosysMapList** table in the PDPS database has at least one entry and the **resourceString** and **autosysIdKey** entries match the entries in the **PlRscString** table.

11 To exit from isql type **quit** at the **1>** prompt then press **Return/Enter**.

12 If the **DpPrAutosysMapList** table in the PDPS database either has no entry or if either the **resourceString** or **autosysIdKey** does not match the corresponding entry in the **PlRscString** table, make a request to the Resource Planner to make the necessary adjustments.

13 If Resource Planning has been done since the Job Management Server was brought up, make a request to the Operations Controller/System Administrator to bounce the server.

- The Job Management Server reads resource information at start-up; any changes since it was brought up will not have taken effect.

Responding to a DPR That Was Released But Failed Due to Invalid DPR

If a job that was activated does not get started in AutoSys, it may be that Job Management released the DPR but the job failed to start because the DPR was invalid. The procedure for responding to a DPR that was released but failed due to an invalid DPR starts with the assumption that the Production Monitor has logged in to the system.

Responding to a DPR That Was Released But Failed Due to Invalid DPR

1 Access a terminal window logged in to the Queuing Server host.

- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.

- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 In the terminal window logged in to the Queuing Server type **cd /usr/ecs/*MODE*/CUSTOM/logs** then press **Return/Enter**.

- Change directory to the directory containing the log files (e.g., EcDpPrJobMgmt.ALOG, EcDpPrDeletion.ALOG, *DPR#*.ALOG, *DPR#*.err, EcPlSubMgrDebug.log, EcDpPrJobMgmtDebug.log).

3 Type **pg EcDpPrJobMgmt.ALOG** then press **Return/Enter**.

- The first page of the log file is displayed.
- Although this procedure has been written for the **pg** command, any UNIX editor or visualizing command (e.g., **vi**, **view**, **more**) can be used to review the log file.

4 Review the Job Management ALOG file for an **"invalid DPR object"** message.

- For example:

  PID : 13169:MsgLink :0 meaningfulname :EnteringCreateDprJob
  Msg: Entering CreateDprJob... Priority: 0 Time : 01/22/00 19:01:24
  PID : 13169:MsgLink :10 meaningfulname :InvalidDprObject
  Msg: invalid DPR object Priority: 2 Time : 01/22/00 19:01:24
  PID : 13169:MsgLink :0 meaningfulname :CantFindAutoSysRecord
  Msg: Unable to locate autosys record Priority: 2 Time : 01/22/00 19:01:24
  PID : 13169:MsgLink :12 meaningfulname :CreateDprJob:ModAutoSysJobCounterProblem
  Msg: ModAutoSysJobCounter problem Priority: 2 Time : 01/22/00 19:01:24
  PID : 13169:MsgLink :13 meaningfulname :DpPrSchedulerDObjSmainCreateFailed
  Msg: RqFailed=CreateDpr DprID=ETS#OnDema01093011DEV04 Priority: 2 Time : 01/22/00 19:01:24

- An invalid DPR object is usually caused by missing **Performance** or **Resource** information for the PGE.

5 Perform the procedure for **Using ISQL to Check Database Tables** (previous section of this lesson) or use a database browser to check for an entry for the job (by searching on the **pgeId** column) in the **PlPgePerformance** table in the applicable PDPS database.

- For example:

  x0sps02:/usr/ecs/TS1/CUSTOM/[6] > isql -U pdps_role –S x0sps02_srvr
  Password:
  1> use pdps_TS1

```
2> go
1> select * from PlPgePerformance where pgeId like "ACT#syn4#001"
2> go
 pgeId                   cpuTime     pgeElapsedTime dprElapsedTime
        maxMemory            faults      swaps       blockInputOperation
        blockOutputOperation runCpuTime  runMaxMemory         runPgeElapsed
        runDprElapsed runFaults   runSwaps    runBlockInOperation
        runBlockOutOperation sharedMemory          runSharedMemory
 ------------------------ ----------- -------------- --------------
       -------------------- ----------- ----------- -------------------
       -------------------- ----------- -------------------- -------------
      ------------- ----------- ----------- -------------------
       -------------------- -------------------- --------------------
 ACT#syn4#001                      55          1800          1800
               10.000000        100         100              100
                    100          55           10.000000        1800
              1800         100         100              100
                    100            0.000000            0.000000

 (1 row affected)
1>  select * from PlPgePerformance where pgeId like "ACT#syn5#001"
2> go
 pgeId                   cpuTime     pgeElapsedTime dprElapsedTime
        maxMemory            faults      swaps       blockInputOperation
        blockOutputOperation runCpuTime  runMaxMemory         runPgeElapsed
        runDprElapsed runFaults   runSwaps    runBlockInOperation
        runBlockOutOperation sharedMemory          runSharedMemory
 ------------------------ ----------- -------------- --------------
       -------------------- ----------- ----------- -------------------
       -------------------- ----------- -------------------- -------------
       ------------- ----------- ----------- -------------------
       -------------------- -------------------- --------------------
 ACT#syn5#001                       0             0             0
                0.000000          0           0                0
                      0           0           0.000000           0
                0           0           0                0
                      0           0.000000            0.000000

(1 row affected)
```

6 Observe the entries in the **PlPgePerformance** table to determine whether the non-schedulable PGE(s) has (have) non-zero values in the various columns of the table.

- In the preceding example the entries for pgeId ACT#syn4#001 are mostly non-zero values whereas all of the entries for pgeID ACT#syn5#001 are zero values.
- There is performance data in the table for pgeID ACT#syn4#001 but none for pgeID ACT#syn5#001.

7 Perform the procedure for **Using ISQL to Check Database Tables** (previous section of this lesson) or use a database browser to check for an entry for the job (by searching on the **sswId** column) in the **PlResourceRequirement** table in the applicable PDPS database.

NOTE: The sswId (science software ID) is the first part of the pgeId. For example:

pgeID = ACT#syn4#001
sswId = ACT#syn4

- For example:

```
1> select * from PlResourceRequirement where sswId like "ACT#syn4"
2> go
 sswId
        string
        numOfCPUs
        computer
        diskSpace          topLevelShellName
        exeTarFileDiskSpace  mcfName
        ramSize            exeUntarFileDiskSpace
        exeTarUR
        pgeId                toolkitArchitecture  pgeCommands
 ---------------------
       ------------------------------------------------------------
       -----------
       ------------------------------------------------------------
       -------------------- ------------------------------
       -------------------- ------------------------------
       -------------------- ---------------------
       ---------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
---------------------------------------------------------
       ------------------------- -------------------- -----------
 ACT#syn4
        x0spg01_string
```

```
            1
       NULL
              10.000000 synpge_sgi6n32
               1.787376 NULL
               0.000000            1.805250

UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]:18:LM:P
GEEXE.001:3839
       ACT#syn4#01           sgi32            1110

(1 row affected)

1> select * from PlResourceRequirement where sswId like "ACT#syn5"
2> go
 sswId
       string
       numOfCPUs
       computer
       diskSpace          topLevelShellName
       exeTarFileDiskSpace  mcfName
       ramSize            exeUntarFileDiskSpace
       exeTarUR
       pgeId                toolkitArchitecture  pgeCommands
 ---------------------
      ------------------------------------------------------------
      -----------
      ------------------------------------------------------------
      -------------------- ------------------------------
      -------------------- ------------------------------
      -------------------- ---------------------
      -------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------
--------------------------------------------------------
      ------------------------- -------------------- -----------
 ACT#syn5
       NULL
            0
       NULL
               0.000000 NULL
               0.000000 NULL
               0.000000            0.000000
       NULL
```

ACT#syn5#01

(1 row affected)

8 Observe the entries in the **PlResourceRequirement** table to determine whether the non-schedulable PGE(s) has (have) non-zero values in the various columns of the table.

- In the preceding example the entries for sswId ACT#syn4 are mostly non-zero values whereas all of the entries for sswID ACT#syn5 are either zero or NULL values.
- There is resource data in the table for sswID ACT#syn4 but none for sswID ACT#syn5.

9 If entries for the non-schedulable PGE(s) in either the **PlPgePerformance** table or **PlResourceRequirement** table are all zero (0) or NULL (e.g., ACT#syn5 in the preceding examples), request the SSI&T team to run the SSIT Operational Metadata GUI and enter correct performance values.

10 If entries for the non-schedulable PGE(s) in either the **PlPgePerformance** table or **PlResourceRequirement** table are all zero (0) or NULL, request the Production Planner to delete and re-create the applicable DPRs (after the SSI&T team has run the SSIT Operational Metadata GUI and entered correct performance values).

- Activation should succeed on the next attempt after the corrections have been made.

Responding to a DPR That Was Released But Failed to Be Received by Job Management Server

If a DPR was released but failed to be received by the Job Management Server, the Planning Workbench would think it had successfully activated the DPR(s) but the Job Management Server would not have received the proper notification. Consequently, Job Management would not release the affected job(s) into AutoSys.

The procedure for responding to a DPR that was released but failed to be received by Job Management Server starts with the assumption that the Production Monitor has logged in to the system.

Responding to a DPR That Was Released But Failed to Be Received by Job Management Server

---

1 Access a terminal window logged in to the Queuing Server host.

- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.

- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 In the terminal window logged in to the Queuing Server type **cd /usr/ecs/*MODE*/CUSTOM/logs** then press **Return/Enter**.

- Change directory to the directory containing the log files (e.g., EcDpPrJobMgmt.ALOG, EcDpPrDeletion.ALOG, *DPR#*.ALOG, *DPR#*.err, EcPlSubMgrDebug.log, EcDpPrJobMgmtDebug.log).

3 Type **pg EcDpPrJobMgmtDebug.log** then press **Return/Enter**.

- The first page of the log file is displayed.
- Although this procedure has been written for the **pg** command, any UNIX editor or visualizing command (e.g., **vi**, **view**, **more**) can be used to review the log file.

4 Review the EcDpPrJobMgmtDebug.log file for problems with communication.

5 Notify the Operations Controller/System Administrator of suspected communication problems.

Single DPS Job Has Failed or Is Hanging

A single DPS job that has failed or is hanging represents one of the following conditions:

- Entire job box is hanging.
- Allocation function is hanging.
- Allocation function has failed.
- Staging function is hanging.
- Staging function has failed.
- Preprocessing job has failed.
- Execution job is hanging.
- Execution job has failed.
- Postprocessing job has failed.
- Insertion function has failed.

Perform the appropriate procedure(s) (as necessary) related to responding to a single DPS job that has failed or is hanging:

- **Handling a Box Job that is Hanging in AutoSys**.

- **Handling a Hanging Allocation Function**.
- **Running Execution Management Outside of AutoSys**.
- **Handling a Failed Allocation Function**.
- **Force-Starting a Job**.
- **Responding to a Restart of a Job That Fails Although All Known Problems Have Been Corrected**.
- **Handling a Hanging Staging Function**.
  - Perform the **Handling a Hanging Allocation Function** procedure.
- **Handling a Failed Staging Function**.
- **Cleaning Up the DPS File Tables**.
- **Handling a Failed Preprocessing Job**.
- **Handling a Hanging Execution Job**.
  - Perform the **Checking AutoSys Status** procedure.
- **Handling a Failed Execution Job**.
  - Perform the **Checking AutoSys Status** procedure.
- **Responding to Execution Job and/or Postprocessing Job That Have (Has) Failed**.
- **Responding to Execution Job That Has Failed and the DPR Has Gone into "Failed-PGE" Processing**.
- **Handling a Failed Postprocessing Job**.
- **Handling Failure of Both Execution and Postprocessing Jobs**.
- **Handling a Failed Insertion Function**.
- **Handling a Failed Deallocate Function**.

Handling a Box Job that is Hanging in AutoSys

This condition is determined by noting that the entire Job Box on **JobScape** (including all three job steps) is the same color, and that color is either the one indicated for "Inactive" jobs or the one for "On Hold" jobs.

- The AutoSys Event server or one of the AutoSys clients could be down.
  - Refer to **Checking AutoSys Status** (previous section of this lesson).

- A "glitch" could have caused the job to go into AutoSys in an "inactive" state.
  - Refer to **Running Execution Management Outside of AutoSys** (subsequent section of this lesson).

Handling a Hanging Allocation Function

A hanging allocation function may be indicated when the Preprocessing job, which had turned green on JobScape or TimeScape to indicate that it was running, never turns either red (failed) or blue (success). Any of the following conditions may cause the allocation function to hang:

- The Science Data Server (SDSRV) may be waiting for a request to Data Distribution (DDIST) to distribute the PGE tar file, but the file cannot be distributed because Storage Management (STMGT) is down.
- The Science Data Server (SDSRV) may be waiting for a request to Data Distribution (DDIST) to distribute the PGE tar file, but the file cannot be distributed because Storage Management cannot ftp the file to the data directory on the science processor disk.
- The request may be waiting for the archive to stage the file. If there are several other requests in progress, the PGE "acquire" request may have to wait until one or more of the other requests completes.

The procedure for handling a hanging allocation function starts with the assumption that the Production Monitor has logged in to the Queuing Server (e.g., e0sps04 or l0sps03) and started the **AutoSys GUI Control Panel**.

Handling a Hanging Allocation Function

---

1 Access a terminal window logged in to the Queuing Server host.

- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 Type **tail -f /usr/ecs/*MODE*/CUSTOM/logs/*jobname*.err** then press **Return/Enter**.

- ***jobname*** refers to the name of the AutoSys job for which the .err log file (e.g., ACT#syn1#004130123OPS.err) is to be reviewed.
- If there is no activity or if the job is in a retry loop, the job is hanging.

3 If there is no activity in the .err log, request the Distribution Technician or Operations Controller (as appropriate) to determine the status of the distribution request for the PGE tar file.

- The requester should be EcDpPrEM.
- If the status is "Suspended with Errors," Storage Management servers may have to be bounced, then the Distribution Technician can resume the request.
- If the status is "Staging," the request may be waiting for the archive to stage the file.
  - If there are several other requests in progress, the PGE acquire may have to wait until one or more of them completes.
  - If the request is in the "Staging" state, it may eventually complete.

4 If distribution of the PGE tar file does not resume, type **cd /usr/ecs/*MODE*/CUSTOM/pdps/*processor*/data/DpPrRm/*processor*_disk** then press **Return/Enter**.

- ***processor*** refers to the Science Processor host (e.g., e0spg11 or l0spg11).
- The ***processor*_disk** directory (e.g., e0spg11_disk) or one of its subdirectories is the target directory where the data server puts the inputs needed for processing.

5 If distribution of the PGE tar file does not resume, type **ls -al** then press **Return/Enter**.

- A listing of the files and subdirectories on the science processor disk (for the mode) is displayed.
- The target directory for the PGE tar file is a subdirectory identified by the sswID (science software identification) of the PGE.
  - For example, if the job in AutoSys is **ACT#syn1#004130123OPS** on the science processor disk there should be an **ACT#syn1** subdirectory.

6 If the target directory does not exist, notify the Operations Controller/System Administrator of the problem.

7 If the target directory does exist, try to ftp a file to the target directory.

- For example:

x0sps02:/usr/ecs/OPS/CUSTOM/logs[109] > ftp x0spg01
Connected to x0spg01.xdc.ecs.nasa.gov.
220-NOTICE: ,
220-*
220-
220-THIS U.S. GOVERNMENT COMPUTING SYSTEM IS FOR AUTHORIZED USERS

220-ONLY. ANYONE USING IT IS SUBJECT TO MONITORING AND RECORDING
220-OF ALL KEYSTROKES WITHOUT FURTHER NOTICE. THIS RECORD MAY BE
220-PROVIDED AS EVIDENCE TO LAW ENFORCEMENT OFFICIALS.
220-
220-*
220-
220-
220-
220 x0spg01.xdc.ecs.nasa.gov FTP server ready.
Name (x0spg01:cmshared):
331 Password required for cmshared.
Password:
230 User cmshared logged in.
ftp> cd /usr/ecs/OPS/CUSTOM/pdps/x0spg01/data/DpPrRm/x0spg01_disk/ACT#syn1
250 CWD command successful.
ftp> put ACT#syn1#004130123OPS.ALOG
200 PORT command successful.
150 Opening ASCII mode data connection for 'ACT#syn1#004130123OPS.ALOG'.
226 Transfer complete.
local: ACT#syn1#004130123OPS.ALOG remote: ACT#syn1#004130123OPS.ALOG
13055 bytes sent in 0.034 seconds (3.7e+02 Kbytes/s)
ftp> quit
221 Goodbye.
x0sps02:/usr/ecs/OPS/CUSTOM/logs[110] > cd /usr/ecs/OPS/CUSTOM/pdps/x0spg01/data/DpPrRm/x0spg01_disk/ACT#syn1
x0sps02:/usr/ecs/OPS/CUSTOM/pdps/x0spg01/data/DpPrRm/x0spg01_disk/ACT#syn1[111] > ls -al
total 5760
drwxrwxr-x   2 cmshared cmshared   65536 Apr 17 10:45 .
drwxrwxr-x 23 cmshared cmshared   65536 Apr 14 13:17 ..
-rw-r--r--   1 cmshared cmshared   12898 Apr 17 10:45 ACT#syn1#004130123OPS.ALOG

- In the example the log file **ACT#syn1#004130123OPS.ALOG** was successfully transferred by ftp from **x0sps02 /usr/ecs/OPS/CUSTOM/logs** to **x0spg01 /usr/ecs/OPS/CUSTOM/pdps/x0spg01/data/DpPrRm/x0spg01_disk/ACT#syn1** as verified by changing directory to the **x0spg01_disk/ACT#syn1** subdirectory and performing a long listing of the directory contents.

8 If the ftp fails, notify the Operations Controller/System Administrator to have the ftp problem fixed.

9 If the Allocation job is in a retry loop, ensure that it is possible to connect to the necessary hosts and servers (listed in Table 6).

- For detailed instructions refer to the section on **Checking Connections to Hosts/Servers** (subsequent section of this lesson).
- Note that the first retry is designed to fail, because the software is retrieving server-side information to refresh the client-side at this point. However, multiple subsequent retries indicate a "retry loop."

10 If no problem has been identified and the job is still hanging, run the Execution Manager in the debugger as described in the procedure for **Running Execution Management Outside of AutoSys** (subsequent section of this lesson).

- Execution Manager (EcDpPrEM) is the DPS program that runs during allocation.

Running Execution Management Outside of AutoSys

To debug problems or to run unit tests, it is sometimes necessary to run Execution Manager (EcDpPrEM) outside of AutoSys.

The procedure for running Execution Management outside of AutoSys starts with the assumption that the Production Monitor has logged in to the Queuing Server (e.g., e0sps04 or l0sps03) and started the **AutoSys GUI Control Panel** and **JobScape**.

Running Execution Management Outside of AutoSys

---

1 Click on the name of the job displayed on **JobScape**.

- The job name is displayed in the **Current Job Name** field in the Control Region of **JobScape**.

2 Click on the **Job Console** button on **JobScape**.

- The **Job Activity Console** GUI (also known as the **Ops Console** GUI) is displayed with information concerning the current job.

3 Access a terminal window logged in to the Queuing Server host.

- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.

- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

4 In a terminal window logged in to the Queuing Server type **cd /usr/ecs/*MODE*/CUSTOM/bin/DPS** then press **Return/Enter**.

5 Type **sh** then press **Return/Enter**.

- A Bourne shell is started.

6 Type **. auto.profile** then press **Return/Enter**.

7 If running in the debugger is desired, type **debugger EcDpPrEM &** then press **Return/Enter**.

- The Execution Manager is brought up in the debugger.

8 Type the command listed in the **Command** field of the **Job Activity Console** GUI (**Ops Console** GUI) then press **Return/Enter**.

- EcDpPrEM starts running.
- The command listed in the **Command** field of the **Job Activity Console** GUI (**Ops Console** GUI) is the command that AutoSys was going to use to run EM.
- For example:

  EcDpPrEM ConfigFile /usr/ecs/OPS/CUSTOM/cfg/EcDpPrEM.CFG ecs_mode OPS –alloc PGE07#1.0#01080596155400

Handling a Failed Allocation Function

If allocation fails, the Preprocessing job turns red on JobScape or TimeScape.

The procedure for handling a failed allocation function starts with the assumption that the Production Monitor has logged in to the Queuing Server (e.g., e0sps04 or l0sps03) and started the **AutoSys GUI Control Panel** and **JobScape**.

Handling a Failed Allocation Function

---

1 Access a terminal window logged in to the Queuing Server host.

- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 Type **cd /usr/ecs/*MODE*/CUSTOM/logs** then press **Return/Enter**.

- Change directory to the directory containing the data processing log files (e.g., EcDpPrJobMgmt.ALOG, EcDpPrDeletion.ALOG).

3 If there is an ALOG file for the job, type **pg *filename*** then press **Return/Enter**.

- ***filename*** refers to the data processing log file to be reviewed (e.g., *DPR#*.ALOG).
- The first page of the log file is displayed.
- Although this procedure has been written for the **pg** command, any UNIX editor or visualizing command (e.g., **vi**, **view**, **more**) can be used to review the log file.

4 If there is an ALOG file for the job, review the log file for the following types of error messages.

- **Error: unable to update Machine in Autosys**
- **Unable to determine type of UR**
- An **"Error: unable to update Machine in Autosys"** message means that DPS is unable to access the AutoSys database. The auto.profile in /usr/ecs/*MODE*/CUSTOM/bin/DPS has the wrong settings for AUTOSYS and AUTOUSER parameters.
  - Although they may differ slightly from DAAC to DAAC, the expected values are generally as follows:

    AUTOSYS = /usr/ecs/*MODE*/COTS/autotreeb/autosys
    AUTOUSER = /usr/ecs/*MODE*/COTS/autotreeb/autouser
- A message of **"Unable to determine type of UR"** means that the PGE tar file has not been inserted.

5 If an "**Error: unable to update Machine in Autosys**" message was present in the log, notify the Operations Controller/System Administrator to have the auto.profile file corrected.

- Either AutoSys Mkcfg has to be run again or the auto.profile file has to be changed manually.

6 If an **"Unable to determine type of UR"** message was present in the log, perform the procedure for **Using ISQL to Check Database Tables** (previous section of this lesson) or use a database browser to check the **PlResourceRequirement** table in the PDPS database for a non-null entry for the field **exeTarUR**.

- For example:

```
x0sps02:/usr/ecs/OPS/CUSTOM[127] > isql -U pdps_role –S x0sps02_srvr
Password:
1> use pdps
2> go
1> select sswId,exeTarUR from PlResourceRequirement where sswId like
"ACT#syn1"
2> go
 sswId
        exeTarUR
 -----------------
        ------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------
------------------------------------------------------
 ACT#syn1

UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[DAC:DSSDSRV]:18:LM:P
GEEXE.001:5521

 (1 row affected)

1> select sswId,exeTarUR from PlResourceRequirement where sswId like
"AM1Eph#2.0"
2> go
 sswId
        exeTarUR
 -----------------
        ------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------
------------------------------------------------------
 AM1Eph#2.0
        NULL

(1 row affected)
```

- In the first query of the example (where sswId like "ACT#syn1"), there is a value listed for the exeTarUR for ACT#syn1.

- In the second query of the example (where sswId like "AM1Eph#2.0"), there is a null value for the exeTarUR for AM1Eph#2.0.

7 If the value for **exeTarUR** in the **PlResourceRequirement** table in the PDPS database is null, make a request to the SSI&T team to have the EXE Tar File inserted.

- When the EXE Tar File has been inserted, it should be possible to restart the job and have it complete successfully. (Refer to **Force-Starting a Job**.)

8 If there is **no** ALOG file for the job, click on the name of the job displayed on **JobScape**.

- The job name is displayed in the **Current Job Name** field in the Control Region of **JobScape**.

9 Click on the **Job Console** button on **JobScape**.

- The **Job Activity Console** GUI (**Ops Console** GUI) is displayed with information concerning the current job.

10 Review the entry in the **Exit Code** field on the **Job Activity Console** GUI.

- A value of 122 means that owner of the job does not have "write" permission to the log files directory.

11 Determine the ownership of the job using the procedure for **Determining the Ownership of an AutoSys Job** (previous section of this lesson).

12 In the terminal window type **ls –al** then press **Return/Enter**.

- A long listing of the logs directory is displayed.
- For example:

```
x0sps02:/usr/ecs/OPS/CUSTOM/logs[137] > ls -al
total 178600
drwxrwxr-x   2 cmops    cmops       3584 Apr 17 12:55 .
drwxrwxr-x  18 cmops    cmops       1024 Oct 11  1999 ..
-rwxrwxrwx   1 cmshared cmshared   12898 Mar 30 11:38
ACT#syn1#004130123OPS.ALOG
-rw-rw-r--   1 cmshared cmshared  105397 Mar 30 11:38
ACT#syn1#004130123OPS.err
-rwxrwxrwx   1 cmshared cmshared   12565 Mar 31 13:24
ACT#syn1#014020000OPS.ALOG
-rw-rw-r--   1 cmshared cmshared   98501 Mar 31 13:24
ACT#syn1#014020000OPS.err
…
```

13 Compare the “write” permission for logs in the logs directory with the owner of the job.

- In the preceding example the user cmshared (and others in the “cmshared” group) has “write” permission for the log files listed.
- If cmshared is the “owner” of the jobs listed in the directory, there should be no problem.

14 If there is a discrepancy between the “write” permission for logs in the logs directory and the owner of the job, report the problem to the Operations Controller/System Administrator for resolution.

15 If there is **no** discrepancy between the “write” permission for logs in the logs directory and the owner of the job, type **pg /var/adm/messages** then press **Return/Enter**.

- The first page of the “messages” file is displayed.
- For example:

  x0sps02:/usr/ecs/OPS/CUSTOM/logs[139] > pg /var/adm/messages
  Apr 4 10:13:39 x0sps02 unix: NFS server x0mss04 not responding still trying
  Apr 4 10:13:39 x0sps02 unix: NFS server x0mss04 not responding still trying
  Apr 4 10:14:39 x0sps02 unix: NFS server x0mss04 ok
  Apr 4 10:14:39 x0sps02 unix: NFS server x0mss04 ok
  Apr 4 10:16:37 x0sps02 reboot: rebooted by root
  Apr 4 10:16:37 x0sps02 syslogd: going down on signal 15
  Apr 4 10:20:04 x0sps02 unix: cpu0: SUNW,UltraSPARC (upaid 6 impl 0x10 ver 0x40 clock 168 MHz)
  Apr 4 10:20:04 x0sps02 unix: cpu1: SUNW,UltraSPARC (upaid 7 impl 0x10 ver 0x40 clock 168 MHz)

- Although this procedure has been written for the **pg** command, any UNIX editor or visualizing command (e.g., **vi**, **view**, **more**) can be used to review the log file.

16 Review the entries in the message log for a message indicating that the security file EcSeRandomDataFile could not be found.

17 If there is a message indicating that that the security file EcSeRandomDataFile could not be found, notify the Operations Controller/System Administrator to have the file created.

- The allocation function should run successfully when the security file has been created.

18 If no problem has been identified, run the Execution Manager in the debugger as described in the procedure for **Running Execution Management Outside of AutoSys** (subsequent section of this lesson).

- Execution Manager (EcDpPrEM) is the DPS program that runs during Allocation.

Force-Starting a Job

To solve or recover from many problems it is necessary to restart the job by force-starting it.

Guidelines for Force-Starting Jobs:

- Force-start command jobs (e.g., preprocessing, postprocessing) only; do not attempt to force-start a box job.
  - The software does not support box job force-starts. (Although it may work fine in some cases, it can cause the PDPS database to get out of sync and prevent the DPR (and possibly other DPRs) from running successfully.)
  - If a box job were force-started, the allocation function would run again. Allocation might choose a different science processor than was chosen the previous time the job ran. Using a different science processor could cause failure of the job.
  - After each job (and often within each job) the state of the DPR is tracked in various tables in the database. Box job force-starts lack the code needed to check the state of the box and perform the cleanup activities necessary for starting over.
- Ensure that the GUI has refreshed and the job to be force-started is not already running before trying to force-start a job. (If a job is already running, it should not be force-started.)
  - It should not be possible to force-start jobs that are already running.
  - If you need to restart a job that is still running, you need to kill it via the AutoSys menu (the same one that has the Force Start choice) and then Force Start it.
- If any command job other than execution fails, force-start the job that failed only. Do not force-start any preceding or succeeding jobs in the box.
- If execution fails, it is not safe to restart it unless the postprocessing job had been put on hold and the failure was detected before postprocessing started running.
- If execution fails and the failure was not detected before postprocessing started running, the DPR must run to completion as a failed PGE and the DPR must be deleted and recreated.

The procedure for force-starting a job starts with the assumption that the Production Monitor has launched the **AutoSys GUI Control Panel** and either **JobScape** or **TimeScape** is being displayed.

Force-Starting a Job

---

1 Click and hold on the applicable job symbol in **JobScape** or **TimeScape** with the **right** mouse button.

- Pop-up menu appears.

2 Select (highlight) **Force Start Job** from the pop-up menu (release the right mouse button).

- The job symbol in **JobScape** or **TimeScape** should turn green ("starting") within a short period of time.

3 If the job symbol in **JobScape** or **TimeScape** does **not** turn green ("starting") within a short period of time, return to Step 1.

Responding to a Restart of a Job That Fails Although All Known Problems Have Been Corrected

If a job fails to restart although all known problems have been corrected, the retry information in the DpPrRpcID database table may not be synchronized between servers.

The procedure for responding to a restart of a job that fails although all known problems have been corrected starts with the assumption that the Production Monitor has logged in to the system.

Responding to a Restart of a Job That Fails Although All Known Problems Have Been Corrected

---

1 Examine the **readableTag** column in the **DpPrRpcID** table of the applicable PDPS database to identify the out-of-sync entry.

- For example:

```
x0sps02:/usr/ecs/OPS/CUSTOM[140] > isql -U pdps_role –S x0sps02_srvr
Password:
1> use pdps
2> go
1> select * from DpPrRpcID
2> go
 readableTag
        object
 ------------------------------------------------------------------
```


-

(0 rows affected)

2 Remove the out-of-sync entry.

3 Restart the job using the procedure for **Force-Starting a Job** (previous section of this lesson).

Handling a Hanging Staging Function

The problems that cause a staging function to hang are generally the same as those that cause an allocation function to hang. Likely causes include the following problems:

- The Science Data Server (SDSRV) may be waiting for a request to Data Distribution (DDIST) to distribute the PGE tar file, but the file cannot be distributed because Storage Management (STMGT) is down.
- The Science Data Server (SDSRV) may be waiting for a request to Data Distribution (DDIST) to distribute the PGE tar file, but the file cannot be distributed because Storage Management cannot ftp the file to the data directory on the science processor disk.
- The request may be waiting for the archive to stage the file(s). If there are several other requests in progress, the "acquire" request may have to wait until one or more of the other requests completes.

Refer to the procedure for **Handling a Hanging Allocation Function**.

Handling a Failed Staging Function

If staging fails, the Preprocessing job turns red on JobScape or TimeScape.

The procedure for handling a failed staging function starts with the assumption that the Production Monitor has logged in to the Queuing Server (e.g., e0sps04 or l0sps03) and started the **AutoSys GUI Control Panel** and **JobScape**.

Handling a Failed Staging Function

1 Access a terminal window logged in to the Queuing Server host.

- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 Type **cd /usr/ecs/*MODE*/CUSTOM/logs** then press **Return/Enter**.

- Change directory to the directory containing the data processing log files (e.g., EcDpPrJobMgmt.ALOG, EcDpPrDeletion.ALOG).

3 If there is an ALOG file for the job, type **pg *filename*** then press **Return/Enter**.

- ***filename*** refers to the data processing log file to be reviewed (e.g., *DPR#*.ALOG).
- The first page of the log file is displayed.
- Although this procedure has been written for the **pg** command, any UNIX editor or visualizing command (e.g., **vi**, **more**) can be used to review the log file.

4 If there is an ALOG file for the job, review the log file for the following types of error messages.

- **ESDT Acquire Failed for UR....**
  - A message of **"ESDT Acquire Failed for UR...."** means that SDSRV had trouble processing one of the acquire requests.
- **found no meta data entry for UR....**
  - A message of **"found no meta data entry for UR...."** means that for some reason the tables that DPS uses to keep track of files on the local disks are no longer synchronized (have gotten out of sync).
- **GetESDTReferenceFailed**
  - A message of **"GetESDTReferenceFailed"** means that the ESDT Reference could not be created for the UR displayed in the message containing "Inside AcquireOneGranuleToSDSRV".
  - It is likely that the granule that is attempting to stage was **not** inserted into the Data Server or has been deleted.
- Database deadlock error messages.
  - A **deadlock problem** accessing the PDPS database is indicated by the following type of message:

    SybaseErrorCode1 =1205;SybaseErrorMesage1 ="x0sps02_srvr"
    SybaseErrorCode2 =13;SybaseErrorMesage2 ="40001"
    Priority : 0 Time : 10/19/99 01:53:48
    PID : 19909:MsgLink :0 meaningfulname :EcPoErrorA1
    Msg: EcPoError::HandleRWError RogueWaveDBToolsError#
    RogueWaveDBToolsErrorCode =21;RogueWaveDBToolsErrorMesage ="[NOREADER] This object cannot support readers"
    Priority : 0 Time : 10/19/99 01:53:48

PID : 19909:MsgLink :0 meaningfulname :DpPrDbIF::SelectAndReadColumns Msg: SelectAndReadColumns failed due to [NOREADER] This object cannot support readers Priority : 2 Time : 10/19/99 01:53:48

– While most deadlock problems are retried, deadlocks on the reading of tables (though rare) currently cannot be retried.

– The error in the example could indicate that a “read” deadlock occurred.

5 If an **"ESDT Acquire Failed for UR...."** message was present in the log, restart the job using the procedure for **Force-Starting a Job** (previous section of this lesson).

- The job should restart and run successfully.

6 If a **"found no meta data entry for UR...."** message was present in the log, restart the job using the procedure for **Force-Starting a Job** (previous section of this lesson).

7 If a **"found no meta data entry for UR...."** message was present in the log and restarting the job was not successful, clean up the DPS file tables using the procedure for **Cleaning Up the DPS File Tables** (subsequent section of this lesson).

8 If a **"found no meta data entry for UR...."** message was present in the log and the DPS file tables have been cleaned up, restart the job using the procedure for **Force-Starting a Job** (previous section of this lesson).

- The job should restart and run successfully.

9 If a **"GetESDTReferenceFailed"** message was present in the log, notify the Production Planner to take the following actions:

- Request to have the granule re-inserted.
- Delete the affected DPR(s).
- Re-create the affected DPR(s).

10 If a message indicating a “read” deadlock was present in the log (as shown in the preceding example), restart the job using the procedure for **Force-Starting a Job** (previous section of this lesson).

- The job should restart and run successfully.

11 If no problem has been identified, run the Execution Manager in the debugger as described in the procedure for **Running Execution Management Outside of AutoSys** (preceding section of this lesson).

- Execution Manager (EcDpPrEM) is the DPS program that runs during staging.

Cleaning Up the DPS File Tables

To solve or recover from some problems, it is necessary to clean up the DPS file tables in the applicable PDPS database. The DPS file tables are as follows:

- **DpPrFile** - a list of staged files and metadata files.
- **DpPrGranuleLocation** - the location of the staged files.
- **DpPrDiskAllocation** - how much disk space the files require.

The offending entries have to be deleted from the tables. The "offending entries" are found using a universalReference (for DpPrFile), a granuleId (for DpPrGranuleLocation), or a fileName (for DpPrDiskAllocation).

The procedure for cleaning up the DPS file tables starts with the assumption that the Production Monitor has logged in to the system.

Cleaning Up the DPS File Tables

---

1 Log in to the appropriate PDPS database as described in Steps 1 through 5 of the procedure for **Using ISQL to Check Database Tables** (previous section of this lesson).

- For example:

  x0sps02:/usr/ecs/OPS/CUSTOM[4] > isql -U pdps_role –S x0sps02_srvr
  Password:
  1> use pdps
  2> go

2 Observe the entries in the **granuleId** column of the **PlDprData** table to determine what granules affect the job (e.g., ACT#syn1#014020000OPS).

- For example:

```
1> select dprId,granuleId from PlDprData where dprId like
"ACT#syn1#014020000OPS"
2> go
 dprId
        granuleId
 -----------------------------
       -------------------------------------------------------------------------------------------------
---------------------------------
 ACT#syn1#014020000OPS
        AST_09T#00102141998020000000
 ACT#syn1#014020000OPS
```

AST_ANC#001L1004
ACT#syn1#014020000OPS
AST_L1B#00102141998020000000

(3 rows affected)

3 Use the **granuleId** entries from the **PlDprData** table to locate the correct entries in the **DpPrGranuleLocation** table.

- For example:

1> select * from DpPrGranuleLocation where granuleId like "AST_L1B#00102141998020000000"
2> go

granuleId
machineId
stageState dprId

--
--
----------- -----------------------------

AST_L1B#00102141998020000000
x0spg01
2 ACT#syn1#014020000OPS

(1 row affected)

4 Delete from the **DpPrGranuleLocation** table the entries that match the **granuleId** entries from the **PlDprData** table.

- For example:

1> delete * from DpPrGranuleLocation where granuleId like "AST_L1B#00102141998020000000"
2> go

5 Use the **granuleId** entries from the **PlDprData** table to verify that the applicable entries have been deleted from the **DpPrGranuleLocation** table.

- Refer to the example in Step 3.

6 Use the **granuleId** entries from the **PlDprData** table to locate the corresponding entries in the **PlDataGranuleShort** table.

- For example:

1> select granuleId,dataTypeId,universalReference from PlDataGranuleShort where granuleId like "AST_L1B#00102141998020000000"
2> go

granuleId
dataTypeId
universalReference

--

AST_L1B#00102141998020000000
AST_L1B#001

UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]:19:SC:AST_L1B.001:5528

(1 row affected)

7 Use the **universalReference** entries from the **PlDataGranuleShort** table to identify corresponding **fileName** entries in the **DpPrFile** table.

- For example:

1> select fileName,universalReference from DpPrFile where universalReference like "%AST_L1B.001:5528"
2> go

fileName
universalReference

--

AST_L1B#00102141998020000000.hdf

UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]:19:SC:AST_L1B.001:5528

AST_L1B#001021419980200000000000.met

UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[XDC:DSSDSRV]:19:SC:AST_L1B.001:5528

(2 rows affected)

- The use of a wild card (as shown by the use of the percent sign in the example) is recommended because isql will not provide a reliable search on a **universalReference**.
- There should be two **fileName** entries in the **DpPrFile** table (i.e., one **.hdf** and one **.met** file) for each **universalReference**.

8 Delete from the **DpPrFile** table the entries that match the **universalReference** entries from the **PlDataGranuleShort** table.

- For example:

  1> delete * from DpPrFile where fileName like "AST_L1B#00102141998020000000%"
  2> go

- In the example the .hdf and .met **fileName** entries are of different lengths; consequently, the shorter of the two (the .hdf **fileName**) has been used with the wild card to specify which files are to be deleted.

9 Use the **universalReference** entries from the **PlDataGranuleShort** table to verify that the applicable entries have been deleted from the **DpPrFile** table.

- Refer to the example in Step 7.

10 Use the **fileName** entries from the **DpPrFile** table to locate the correct entries in the **DpPrDiskAllocation** table.

- For example:

  1> select * from DpPrDiskAllocation where fileName like "AST_L1B#00102141998020000000%"
  2> go
  diskAllocationId computerId diskPartitionId diskAllocationType
  path
  diskAllocationSize diskAllocationUser

```
diskAllocationActual
fileName
---------------- ----------- --------------- ------------------
------------------------------------------------------------
------------------- -----------------------------
-------------------
------------------------------------------------------------
 2001957         3             1                1
/x0spg01_disk/
        50.085064 ACT#syn1#014020000OPS
        50.085064
AST_L1B#00102141998020000000.hdf

(1 row affected)
```

11 Delete from the **DpPrDiskAllocation** table the entries that match the **fileName** entries from the **DpPrFile** table.

- For example:

  1> delete * from DpPrDiskAllocation where fileName like "AST_L1B#00102141998020000000%"
  2> go

12 Use the **fileName** entries from the **DpPrFile** table to verify that the applicable entries have been deleted from the **DpPrDiskAllocation** table.

- Refer to the example in Step 10.

Handling a Failed Preprocessing Job

If preprocessing fails, the Preprocessing job turns red on JobScape or TimeScape.

The procedure for handling a failed Preprocessing job starts with the assumption that the Production Monitor has logged in to the Queuing Server (e.g., e0sps04 or l0sps03) and started the **AutoSys GUI Control Panel** and **JobScape**.

Handling a Failed Preprocessing Job

---

1 Access a terminal window logged in to the Queuing Server host.

- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.

- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 Type **cd /usr/ecs/*MODE*/CUSTOM/logs** then press **Return/Enter**.

- Change directory to the directory containing the data processing log files (e.g., EcDpPrJobMgmt.ALOG, EcDpPrDeletion.ALOG).

3 If there is an ALOG file for the job, type **pg *filename*.ALOG** then press **Return/Enter**.

- ***filename*** refers to the data processing log file to be reviewed (e.g., *DPR#*.ALOG).
- The first page of the log file is displayed.
- Although this procedure has been written for the **pg** command, any UNIX editor or visualizing command (e.g., **vi**, **view**, **more**) can be used to review the log file.

4 If there is an ALOG file for the job, review the log file for the following type of error message:

- **NOFREECPUS**
  - A message of **"NOFREECPUS"** means that all of the Science Processor CPUs are busy and the Preprocessing job went through its maximum number of retries to find an available CPU, possibly for one of the following reasons:
    - PGEs are taking longer to run than expected. DPS plans for execution times specified during SSIT, and if those times are exceeded by a large margin (by an executing PGE) it is possible that a PGE that is "ready to run" will be CPU-starved.
    - Somebody has scheduled a PGE that takes up more CPUs than will ever be available. If a PGE is defined (at SSIT) to require five CPUs and there are only three on any given machine, the job will never succeed.

5 If a **"NOFREECPUS"** message was present in the log, restart the job using the procedure for **Force-Starting a Job** (previous section of this lesson).

6 If the preceding problem was not mentioned in the log file, type **pg *filename*.err** then press **Return/Enter**.

- ***filename*** refers to the data processing log file to be reviewed (e.g., *DPR#*.err).
- The first page of the log file is displayed.
- Although this procedure has been written for the **pg** command, any UNIX editor or visualizing command (e.g., **vi**, **view**, **more**) can be used to review the log file.

7 If using either the **pg** or **more** command, at the **:** prompt type **/-prep** then press **Return/Enter**.

- The start of the Preprocessing job log is displayed.
- The following type of entry indicates the start of the Preprocessing activities:

  Command used => 'EcDpPrEM ConfigFile /usr/ecs/OPS/CUSTOM/cfg/EcDpPrEM.CFG ecs_mode OPS -preproc AM1Eph#30012312200OPS' START_TIME: 03/30/00 10:50:32

  – The **–preproc** indicates "Preprocessing."

8 Review the Preprocessing portion of the .err log file for the **"rm: Unable to remove directory *long_directory_pathname*: File exists"** type of error message.

- The **"rm: Unable to remove directory *long_directory_pathname*: File exists"** type of error message means that there is a permission problem and the Execution Manager job could **not** delete the files within the directory.
- The ***long_directory_pathname*** will be similar to the following example:

  /usr/ecs/OPS/CUSTOM/pdps/x0spg01/data/DpPrRm/x0spg01_disk/ AM1Eph#30001/AM1Eph#30012312200OPS_x0spg01/

9 If the **"rm: Unable to remove directory *long_directory_pathname*: File exists"** type of message was present in the Preprocessing portion of the .err file, in the terminal window type **cd *long_directory_pathname*** then press **Return/Enter**.

10 In the terminal window type **ls –al** then press **Return/Enter**.

- A long listing of the directory is displayed, for example:

  x0sps02:/usr/ecs/OPS/CUSTOM/pdps/x0spg01/data/DpPrRm/x0spg01_disk/
  AM1Eph#30001/AM1Eph#30012312200OPS_x0spg01[82] > ls -al
  total 3712
  drwxr-xr-x 2 cmshared cmshared 65536 Apr 18 11:00 .
  drwxrwxr-x 3 cmshared cmshared 65536 Apr 18 10:59 ..
  -rw-rw-r-- 1 cmshared cmshared 5384 Apr 18 10:59 AM1ATTH0#001.MCF
  -rw-rw-r-- 1 cmshared cmshared 5385 Apr 18 10:59 AM1ATTN0#001.MCF
  -rw-rw-r-- 1 cmshared cmshared 7183 Apr 18 10:59 AM1EPHH0#001.MCF
  -rw-rw-r-- 1 cmshared cmshared 7183 Apr 18 10:59 AM1EPHN0#001.MCF
  -rw-r--r-- 1 cmshared cmshared 4437 Apr 18 11:00
  AM1Eph#30012312200OPS.Log
  -rw-rw-r-- 1 cmshared cmshared 31764 Apr 18 11:00
  AM1Eph#30012312200OPS.Pcf

-rw-rw-r-- 1 cmshared cmshared 382 Apr 18 11:00 AM1Eph#30012312200OPS.Profile
-rw-rw-r-- 1 cmshared cmshared 958 Apr 18 11:00 AM1Eph#30012312200OPS.TkReport
-rw-rw-r-- 1 cmshared cmshared 3299 Apr 18 11:00 AM1Eph#30012312200OPS.TkStatus
-rw-rw-r-- 1 cmshared cmshared 956 Apr 18 11:00 AM1Eph#30012312200OPS.TkUser
-rw-rw-r-- 1 cmshared cmshared 1195 Apr 18 11:00 AM1Eph#30012312200OPS_PGE.IN
-rw-rw-r-- 1 cmshared cmshared 7291 Apr 18 11:00 MCFWrite.temp
-rw-rw-r-- 1 cmshared cmshared 434111 Apr 18 11:00 pc19811823201201485810900110024
-rw-rw-r-- 1 cmshared cmshared 7291 Apr 18 11:00 pc19811823201201485810900110024.met
-rw-rw-r-- 1 cmshared cmshared 451584 Apr 18 11:00 pc19811823201201485810900110025

11 Determine who has write permission for files in the directory.

- In the preceding example the user **cmshared** and members of the cmshared group have write permission for the directory.

12 If possible (assuming write permission) type **mv *filename1* [... *filenamex*] *destination_directory*** then press **Return/Enter** to move the files to another directory.

13 If write permission is not available, notify the System Administrator of the need to remove the files from the directory.

14 If no problem has been identified, run the Execution Manager in the debugger as described in the procedure for **Running Execution Management Outside of AutoSys** (preceding section of this lesson).

- Execution Manager (EcDpPrEM) is the DPS program that runs during Preprocessing.

Handling a Hanging Execution Job

This condition is determined by noting that the Execution job has turned orange or oscillates between orange and green on JobScape or TimeScape. It is most likely that the AutoSys client is down. Perform the procedure for **Checking AutoSys Status** (preceding section of this lesson).

Handling a Failed Execution Job

This condition is indicated when the Execution (PGE) job only is red on JobScape or TimeScape. This is hard to do, because the AutoSys job definition for this job says to allow **any** exit code to indicate success. It is set up this way so the next job, the Postprocessing job, continues even when the Execution job fails. The Execution job goes to a "success" state even when the PGE Wrapper job, EcDpPrRunPGE, does not exist. However, the Execution job can fail if AutoSys cannot see the machine. Consequently, if the Execution job (only) fails, perform the procedure for **Checking AutoSys Status** (preceding section of this lesson).

Responding to Execution Job and/or Postprocessing Job That Have (Has) Failed

This condition is determined by noting that the Execution job has turned red in JobScape or TimeScape or the entire job box has turned red (failedPGE scenario).

Perform the appropriate procedure(s) (as necessary) related to responding to Execution and/or Postprocessing Jobs that have failed:

- **Responding to Execution Job That Has Failed and the DPR Has Gone into "Failed-PGE" Processing**.
- **Handling a Failed Postprocessing Job**.
- **Handling Failure of Both Execution and Postprocessing Jobs**.

Responding to Execution Job That Has Failed and the DPR Has Gone into "Failed-PGE" Processing

This condition is indicated when the entire job box has turned red in JobScape or TimeScape along with the Postprocessing job. A Failed PGE tar file has been created and archived.

A PGE may fail for many reasons. Some of the possible causes are documented here:

- The PGE has the wrong architecture.
- One of the expected inputs for the PGE is missing.
- The leap seconds file is incorrect.
- The file-watcher program detected that the PGE was writing files much larger than expected.
- There are problems accessing the Toolkit on the Science Processor.
- The PGE has not been staged.

The procedure for responding to an Execution job that has failed and the DPR has gone into "Failed-PGE" processing starts with the assumption that the Production Monitor has logged in to the Queuing Server (e.g., e0sps04 or l0sps03) and started the **AutoSys GUI Control Panel** and **JobScape**.

Responding to Execution Job That Has Failed and the DPR Has Gone into "Failed-PGE" Processing

---

1 Notify the SSI&T team to check the PGE architecture using the SSIT tools.

- If the PGE has the wrong architecture, it is probably because the PGE was improperly defined as **New32**, **Old32** or **64** from the SSIT Operational Metadata GUI.
  - The PGE core dumps because of this problem.
- After the SSI&T team has entered the correct architecture using the SSIT Operational Metadata GUI the Production Planner has to delete and recreate all DPRs created for that PGE.

2 Access a terminal window logged in to the Queuing Server host.

- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

3 Type **cd /*path*** then press **Return/Enter**.

- Change directory to the run-time directory for the job (e.g., /usr/ecs/OPS/CUSTOM/pdps/x0spg01/data/DpPrRm/x0spg01_disk/ACT#syn1/ ACT#syn1#004130123OPS_x0spg01/).

4 In the terminal window type **ls** then press **Return/Enter**.

- A listing of the directory is displayed, for example:

x0sps02:/usr/ecs/OPS/CUSTOM/pdps/x0spg01/data/DpPrRm/x0spg01_disk/
AM1Eph#30001/AM1Eph#30012312200OPS_x0spg01[82] > ls
AM1ATTH0#001.MCF
AM1ATTN0#001.MCF
AM1EPHH0#001.MCF
AM1EPHN0#001.MCF
AM1Eph#30012312200OPS.Log
AM1Eph#30012312200OPS.Pcf
AM1Eph#30012312200OPS.Profile
AM1Eph#30012312200OPS.TkReport
AM1Eph#30012312200OPS.TkStatus
AM1Eph#30012312200OPS.TkUser
AM1Eph#30012312200OPS_PGE.IN

MCFWrite.temp
pc19811823201201485810900110024
pc19811823201201485810900110024.met
pc19811823201201485810900110025

5 Review the contents of the runtime directory of the PGE/DPR to determine whether status logs (e.g., *DPR#*.TkStatus, *DPR#*.Tkuser, and *DPR#*.TkLog) were created for the PGE.

- DPS uses a Toolkit command to start the PGE. So if no status logs were created for the PGE, it is very likely that the Toolkit was not installed properly on the Science Processor.

6 If no status logs were created for the PGE, type **cd /*path*** then press **Return/Enter**.

- Change directory to the CUSTOM directory for the mode (e.g., /usr/ecs/OPS/CUSTOM).

7 If no status logs were created for the PGE, type **ls -al** then press **Return/Enter**.

- The following type of directory listing is displayed:

```
x0spg01:/usr/ecs/OPS/CUSTOM[45] > ls -al
total 392
drwxrwxrwx   20 cmops    cmops      4096 Apr  6 17:05 .
drwxr-xr-x    4 root     sys          37 Oct  7  1997 ..
-rw-rw-r--    1 cmops    cmops      3834 Mar 27 12:41 .applications
-rw-rw-r--    1 cmops    cmops      1603 Mar 27 12:42 .cache
-rw-rw-r--    1 cmops    cmops     16547 Mar 27 12:41 .cfgpatch
-rw-rw-r--    1 cmops    cmops      6160 Mar 27 12:41 .envvars
-rw-rw-r--    1 cmops    cmops     22841 Mar 27 12:41 .executables
-rw-rw-r--    1 cmops    cmops      4368 Mar 27 12:41 .hostmap
drwxrwxr-x    6 cmops    cmops        61 Oct  8  1999 .installed
-rw-rw-r--    1 cmops    cmops     12616 Mar 27 12:41 .installtypes
-rw-rw-r--    1 cmops    cmops      8657 Mar 27 12:41 .sitehostmap
-rw-rw-r--    1 cmops    cmops     72760 Mar 27 12:41 .sitemap
-rw-rw-r--    1 cmops    cmops      1845 Mar 27 12:41 .subsystems
drwxr-xr-x    6 cmops    cmops      4096 Dec 14 09:14 Aadata
drwxrwxr-x    9 cmops    cmops       122 Mar 27 12:35 HDF
drwxrwxr-x    4 cmops    cmops        45 Nov 18 15:01 HDFEOS
drwxrwxrwx    3 cmops    cmops        25 Oct 11  1999 TOOLKIT
drwxrwxr-x    3 cmops    cmops        21 Oct 11  1999 WWW
drwxr-xr-x   27 cmops    cmops      4096 Nov 18 15:02 backup
drwxrwxr-x    6 cmops    cmops       142 Nov 30 15:38 bin
drwxrwxr-x    2 cmops    cmops         9 Oct 11  1999 cfg
```

```
drwxr-xr-x    4 cmops    cmops      4096 Sep 27  1999 daac_toolkit_f77
drwxrwxr-x    7 cmops    cmops        69 May  7  1999 data
drwxrwxr-x    3 cmops    cmops        21 Oct 11  1999 dbms
drwxrwxr-x    6 cmops    cmops        57 Oct 11  1999 lib
drwxrwxr-x    2 cmops    cmops      4096 Apr 18 19:52 logs
drwxr-xr-x    4 root     sys          41 Jun 24  1998 pdps
drwxrwxr-x    2 cmops    cmops        36 Mar 27 12:41 security
drwxr-xr-x    3 root     sys          25 Dec 18  1997 ssit
lrwxr-xr-x    1 cmops    cmops        36 Mar 27 17:13 toolkit ->
/usr/ecs/OPS/CUSTOM/daac_toolkit_f77
drwxrwxr-x    2 cmops    cmops      4096 Mar 27 12:41 utilities
```

- There should be a "toolkit" subdirectory as shown in the example.

8 If the Toolkit was not properly installed, notify the Operations Controller/System Administrator to have the problem corrected.

9 If the Toolkit was properly installed and there are status logs in the runtime directory, type **cd** ***/path*** then press **Return/Enter**.

- Change directory to the run-time directory for the job (e.g., /usr/ecs/OPS/CUSTOM/pdps/x0spg01/data/DpPrRm/x0spg01_disk/ACT#syn1/ACT#syn1#004130123OPS_x0spg01/).

10 Type **pg** ***DPR#*.TkStatus** then press **Return/Enter**.

- ***DPR#*** refers to the name of the job (e.g., AM1Eph#30012312200OPS or ACT#syn1#004130123OPS).
- The first page of the log file is displayed.
- Although this procedure has been written for the **pg** command, any UNIX editor or visualizing command (e.g., **vi**, **view**, **more**) can be used to review the log file.

11 Review the *DPR#*.TkStatus log in the runtime directory of the PGE/DPR for messages concerning a missing input or inability to get number of files.

- If the PGE is missing an input, it is probably because the DPR was released into AutoSys although not all of its inputs were available at the Data Server.

12 If a **logicalId** is mentioned in the context of a missing input or inability to get number of files, review the ***DPR#*.Pcf** file in the runtime directory of the PGE/DPR to determine whether that **logicalId** is present in the file.

- If the PGE is missing an input, the **logicalId** will not be present in the ***DPR#*.Pcf** file.

13 Log in to the appropriate PDPS database as described in Steps 1 through 5 of the procedure for **Using ISQL to Check Database Tables** (previous section of this lesson).

- For example:

```
x0sps02:/usr/ecs/OPS/CUSTOM[4] > isql -U pdps_role –S x0sps02_srvr
Password:
1> use pdps
2> go
```

14 Search on the **dprId** for the **logicalId** and the corresponding value in the **accepted** column in the **PlDprData** table.

- For example:

```
1>  select dprId,granuleId,logicalId,accepted from PlDprData where dprId like
"ACT#syn1#014020000OPS"
2> go
 dprId
        granuleId
        logicalId   accepted
 -----------------------------
       -------------------------------------------------------------------------------------------------
---------------------------------
       ----------- -----------
 ACT#syn1#014020000OPS
        AST_09T#00102141998020000000
             2000           6
 ACT#syn1#014020000OPS
        AST_ANC#001L1008
             1200           5
 ACT#syn1#014020000OPS
        AST_L1B#00102141998020000000
             1100           5
 ACT#syn1#014020000OPS
        GDAS_0ZF#001O1006
             1101           5

 (4 rows affected)
```

15 If the **accepted** field has “0” as its value, notify the Production Planner to delete the DPR and re-create it.

– If the **accepted** field has “0” as its value, the DPR was released without all of its inputs and that is why the PGE failed.

16 Type **cd** ***/path*** then press **Return/Enter**.

- Change directory to the directory containing the log files for the DPR (e.g., /usr/ecs/OPS/CUSTOM/logs).

17 Type **ls** then press **Return/Enter**.

- A listing of files in the logs directory is displayed.

18 Type **pg** ***filename*** then press **Return/Enter**.

- The ***filename*** refers to the name of the log file (e.g., ACT#syn1#014020000OPS.err, ACT#syn1#014020000OPS.ALOG).
- The first page of the log file is displayed.
- Although this procedure has been written for the **pg** command, any UNIX editor or visualizing command (e.g., **vi**, **view**, **more**) can be used to review the log file.

19 Review the DPR log file(s) for message(s) concerning a missing **logicalId**.

- Search for error messages concerning a missing Logical Id.
  - The error message may refer to either “missing” or “Logical Id” in lower case or upper case depending on the type of log file.
  - “Logical Id” may be one word or two (e.g., Logical Id, logicalId) depending on the type of log file.
- If one of the expected inputs for the PGE is missing, it is possible that an expected input of the PGE is not defined in the PGE ODL file.

20 Access a terminal window logged in to the AIT Workstation host.

- Examples of AIT Workstation host names include **e0ais03 and l0ais01.**
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

21 Type **cd** ***/path*** then press **Return/Enter**.

- Change directory to the directory containing the ODL files (e.g., /usr/ecs/OPS/CUSTOM/data/DPS/ODL).

22 Type **ls** then press **Return/Enter**.

- A listing of files in the ODL directory is displayed.

23 Type **pg PGE_*PGE#*.odl** then press **Return/Enter**.

- ***PGE#*** refers to the name of the PGE (e.g., ACT#syn1#01).
- The first page of the ODL file is displayed.

- Although this procedure has been written for the **pg** command, any UNIX editor or visualizing command (e.g., **vi**, **view**, **more**) can be used to review the log file.

24 Review the PGE ODL file for the expected **logicalId** (listed as **LOGICAL_ID** in the ODL file).

- For example:

```
OBJECT = PCF_ENTRY
  CLASS = 11
  LOGICAL_ID = 1100
  PCF_FILE_TYPE = 1
  DATA_TYPE_NAME = "AST_L1B"
  DATA_TYPE_VERSION = "001"
  DATA_TYPE_REQUIREMENT = 1
  BEGIN_PERIOD_OFFSET = 0
  END_PERIOD_OFFSET = 0
  INPUT_TYPE = "Required"
  NUMBER_NEEDED = 1
  KEY_INPUT = "Y"
/**** Entry needed for all I/O (except for Temporary) ****/
/**** Only modify if multiple files and/or file types for this PCF entry ****/
  OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
  END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY
```

- The example shows the PGE ODL entry for **logicalId** 1100 (AST_L1B), which is input for ACT PGEs. There are additional PCF_ENTRY objects in the ODL file for the other files associated with the PGE.
- If the PGE is synthetic, it is possible that the ODL was filled out incorrectly because of special parameters that the synthetic PGE expects.

25 If the ODL file is incorrect, notify the SSI&T team to have the file corrected.

26 Log in to the applicable PDPS database if not still logged in.

- Refer to the example in Step 13.

27 Observe the entries in the **PlDataGranuleShort** table to determine whether all inputs to the DPR have "1" as their **availability** flag setting.

- For example:

  1> select granuleId,availability from PlDataGranuleShort where granuleId like "AST_L1B#00102141998020000000"
  2> go

  granuleId
  availability

  AST_L1B#00102141998020000000
  1

  (1 row affected)

- In the example the availability flag of the granule is set at "1," which indicates that the granule is available.
- The same sort of query would be accomplished for the other two inputs; i.e., granuleId AST_ANC#001L1008 and granuleId GDAS_0ZF#001O1006.

28 Observe the entries in the **PlDprData** table to determine whether all inputs to the DPR have "1" in their **"accepted"** field.

- As shown in Step 14, the **"accepted"** fields for the inputs for the example have the following values:
  - AST_ANC#001L1008 5
  - AST_L1B#00102141998020000000 5
  - GDAS_0ZF#001O1006 5
- Note that AST_09T#00102141998020000000 is an output, not an input.
- A miscommunication can cause the Subscription Manager to release a PGE despite the fact that it is missing one (or more) input(s).

29 In a terminal window logged in to the Queuing Server review the *DPR#*.TkStatus log (see Steps 10 and 11) in the runtime directory of the PGE/DPR for an error message indicating that the Toolkit had trouble processing some time associated with the PGE.

- An error message indicating that the Toolkit had trouble processing some time associated with the PGE may indicate that the leap seconds file is incorrect.

30 If it is suspected that that the leap seconds file is incorrect (e.g., the TkStatus log has an error message indicating that the Toolkit had trouble processing some time associated with the PGE), notify the SSI&T team to have the leap seconds file updated.

31 Review the *DPR#*.Log file for the PGE in the runtime directory of the PGE/DPR for a message from the file watcher indicating that the PGE was killed because of output file size.

- The file watcher runs in the background and verifies that the PGE does not exceed its output file sizes by a configurable amount.
- If a PGE creates a file that is "too large" the file watcher kills the PGE.

32 If file watcher killed the PGE because the output file exceeded its expected size, notify the SSI&T team of the problem.

33 If no file watcher-associated problem was found in the *DPR#*.Log file, type **cd /*path*** then press **Return/Enter**.

- Change directory to the directory where the PGE should have been staged (e.g., /usr/ecs/OPS/CUSTOM/pdps/x0spg01/data/DpPrRm/x0spg01_disk/ACT#syn1/).

34 Type **ls** then press **Return/Enter**.

- A listing of files in the directory is displayed.

35 Review the directory listing to determine the presence of the PGE executable.

- DPS must acquire (stage) the PGE from Science Data Server before it runs for the first time.
- In the subdirectory that matches the *PGE_Name#Ssw_Id* (i.e., ACT#syn1 in the example) the PGE should have been staged and untarred. If no PGE executable exists in the directory, that is the reason for the PGE failure.
- Possible reasons why DPS would have run the PGE execution job without a PGE:
  - The DpPrExecutable table has entries after the database has been cleaned.
  - There is a file permission problem.

36 Log in to the applicable PDPS database if not still logged in.

- Refer to the example in Step 13.

37 Observe the entries in the **DpPrExecutable** table to determine whether there is an entry for the failing PGE with a setting of “0” in the **execLayer** column.

- For example:

  1> select sswId,execLayer from DpPrExecutable where sswId like "ACT#syn1"

2> go
sswId execLayer
----------------- -----------
ACT#syn1 0

(1 row affected)

- If there is an entry for the failing PGE in the **DpPrExecutable** database table and it has an entry of "0" in the **execLayer** column, DPS thinks that it has already staged the PGE.

38 If there is an entry for the failing PGE in the **DpPrExecutable** database table and it has an entry of "0" in the **execLayer** column, delete the entry from the **DpPrExecutable** table.

- For example:

1> delete * from DpPrExecutable where sswId like "ACT#syn1"
2> go

39 If the **DpPrExecutable** database table contained an entry (for the failing PGE) that was deleted, make a request to the Production Planner to replan the DPR(s).

40 If the **DpPrExecutable** database table has an entry of some value other than "0" in the **execLayer** column, review the *DPR#*.TkStatus log (see Steps 10 and 11) in the runtime directory of the PGE/DPR to identify the file or type of file (i.e., metadata file or data file) to which the PGE is having trouble writing.

41 Type **cd /*path*** then press **Return/Enter**.

- Change directory to the directory containing the file to which the PGE is having trouble writing.
- The directory structure under which DPS manages its files is defined by the machines (science processors), what are called the DataRoots for those machines, and the disks defined by the system. The directory structure can be represented as follows:

{machine name}_DataRoot
{disk one} {disk two} . . . .
{PGENAME}#{software version}
{DPR Name}_{machine name}

- The **{machine name}_DataRoot** parameter is found in the Configuration Registry for Execution Management (EcDpPrEM). The **{machine name}** is the name(s) of the science processor(s). There is a data root for each science processor in the system. The **{machine name}_DataRoot** is considered the top-level directory, where a directory for each disk in the system is placed. So under this directory are directories named after the various disks on the system.

- In the disk directories input and output files are acquired and produced.
- Underneath the disk directories are directories for each PGE of the form **{PGENAME}#{Software Version}**. This is where the PGE tar files and toolkit files are staged.
- Beneath the PGE directories is a directory for each DPR (**{DPR Name}_{machine name}**), which is where the toolkit status files, PGE logs and temporary files are placed.

- The PGE creates the following files:
  - An output file for every granule that it produces.
    - Each output file has the form **{granuleId}.{file number}**, where the Granule Id matches the granuleId in the **PlDataGranuleShort** and **DpPrGranuleLocation** database tables. File number is added to differentiate different files within a granule.
  - A .met file for every granule that it produces.
    - Each .met file has the form **{granule Id}.met**.
  - Toolkit status logs.
    - The toolkit status logs have the form **{DPRID}.Tkstatus**, **{DPRID}.Tkuser** and **{DPRID}.TkLog**. They are placed in the runtime directory of the PGE/DPR.

42 Type **ls -al** then press **Return/Enter**.

- A long listing of files in the directory is displayed.
- For example:

x0sps02:/usr/ecs/OPS/CUSTOM/pdps/x0spg01/data/DpPrRm/x0spg01_disk[94] > ls -al | pg
total 14185088
drwxrwxr-x 26 cmshared cmshared 65536 Apr 18 19:52 .
drwxrwxr-x 3 cmops cmops 30 Apr 30 1999 ..
drwxrwxr-x 2 cmshared cmshared 65536 Apr 17 10:49 ACT#syn1
-rw-r--r-- 1 EcDpPrEm users 384 Apr 18 16:06 AM1ANC#001010120000000000000000
-rw-r--r-- 1 EcDpPrEm users 20988 Apr 18 16:06 AM1ANC#001010120000000000000000.met

```
-rw-r--r--   1 EcDpPrEm users     450048 Apr 18 16:06
AM1ANC#00101012000000000000001
-rw-r--r--   1 EcDpPrEm users        384 Apr 18 16:11
AM1ANC#00101012000020000000000
-rw-r--r--   1 EcDpPrEm users      20988 Apr 18 16:11
AM1ANC#00101012000020000000000.met
-rw-r--r--   1 EcDpPrEm users     449984 Apr 18 16:11
AM1ANC#00101012000020000000001
-rw-r--r--   1 EcDpPrEm users        384 Nov 23 13:32
AM1ANC#00107311997060000000000
-rw-r--r--   1 EcDpPrEm users      20988 Nov 23 13:32
AM1ANC#00107311997060000000000.met
-rw-r--r--   1 EcDpPrEm users     449984 Nov 23 13:32
AM1ANC#00107311997060000000001
-rw-r--r--   1 EcDpPrEm users        384 Nov 23 13:33
AM1ANC#00107311997080000000000
-rw-r--r--   1 EcDpPrEm users      20988 Nov 23 13:33
AM1ANC#00107311997080000000000.met
-rw-r--r--   1 EcDpPrEm users     450048 Nov 23 13:33
AM1ANC#00107311997080000000001
-rw-r--r--   1 EcDpPrEm users        384 Apr 18 16:05
AM1ANC#00112311999220000000000
pr 18 16:05 AM1ANC#00112311999220000000000.met
-rw-r--r--   1 EcDpPrEm users     449984 Apr 18 16:05
AM1ANC#00112311999220000000001
-rw-rw-r--   1 cmshared cmshared 7187328 Dec 15 16:35
AST_09T#00107211993231212003000
-rw-rw-r--   1 cmshared cmshared   13012 Dec 15 16:31
AST_09T#00107211993231212003000.met
-rw-rw-r--   1 cmshared cmshared 7187636 Dec 15 16:52
AST_09T#00107211993231212004000
-rw-rw-r--   1 cmshared cmshared   13012 Dec 15 16:48
AST_09T#00107211993231212004000.met
-rw-rw-r--   1 cmshared cmshared   13012 Dec 15 18:06
AST_09T#00107211993231212005000.met
```

43 Determine who has "write" permission for the file to which the PGE is having trouble writing.

- In the preceding example the user **EcDpPrEM** (the Execution Manager in DPS) has "write" permission for the AM1ANC files shown; the user **cmshared** and members of the cmshared group have “write” permission for the AST_09T files (output of the ACT PGE) shown.

- The PGE must be able to write to the directory where the data is kept (for its PGE outputs and .met files) and to the runtime directory where the log files and temporary files are kept.
- Both the DPS jobs and the PGE must be able to write to the directories.
  - Although it is unlikely, it is possible that the DPS jobs and the PGE have different owners.

44 If there is a discrepancy between the “write” permission for the file and the owner of the job, report the problem to the Operations Controller/System Administrator for resolution.

45 When the problem has been resolved, make a request to the Production Planner to replan the affected DPR(s).

Handling a Failed Postprocessing Job

If postprocessing fails, the Postprocessing job turns red on JobScape or TimeScape.

The procedure for handling a failed Postprocessing job starts with the assumption that the Production Monitor has logged in to the Queuing Server (e.g., e0sps04 or l0sps03) and started the **AutoSys GUI Control Panel** and **JobScape**.

Handling a Failed Postprocessing Job

---

1 Access a terminal window logged in to the Queuing Server host.

- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 Type **cd /usr/ecs/*MODE*/CUSTOM/logs** then press **Return/Enter**.

- Change directory to the directory containing the data processing log files (e.g., EcDpPrJobMgmt.ALOG, EcDpPrDeletion.ALOG).

3 If there is an ALOG file for the job, type **pg *filename*.ALOG** then press **Return/Enter**.

- ***filename*** refers to the data processing log file to be reviewed (e.g., *DPR#*.ALOG).
- The first page of the log file is displayed.

- Although this procedure has been written for the **pg** command, any UNIX editor or visualizing command (e.g., **vi**, **more**) can be used to review the log file.

4 If there is an ALOG file for the job, review the log file for an error message concerning DPS having trouble finding a log file.

- This means that the PGE probably did not run due to one of the following problems:
  - The toolkit links on the science processor are not correct.
  - The **auto.profile** configuration file has not been generated correctly.

5 Access a terminal window logged in to the applicable Science Processor.

- Examples of Science Processor host names include **e0spg11** and **l0spg11**.
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

6 Type **cd /usr/ecs/*MODE*/CUSTOM** then press **Return/Enter**.

7 Type **ls -al** then press **Return/Enter**.

- The following type of directory listing is displayed:

```
x0spg01:/usr/ecs/OPS/CUSTOM[45] > ls -al
total 392
drwxrwxrwx   20 cmops    cmops      4096 Apr  6 17:05 .
drwxr-xr-x    4 root     sys          37 Oct  7  1997 ..
-rw-rw-r--    1 cmops    cmops      3834 Mar 27 12:41 .applications
-rw-rw-r--    1 cmops    cmops      1603 Mar 27 12:42 .cache
-rw-rw-r--    1 cmops    cmops     16547 Mar 27 12:41 .cfgpatch
-rw-rw-r--    1 cmops    cmops      6160 Mar 27 12:41 .envvars
-rw-rw-r--    1 cmops    cmops     22841 Mar 27 12:41 .executables
-rw-rw-r--    1 cmops    cmops      4368 Mar 27 12:41 .hostmap
drwxrwxr-x    6 cmops    cmops        61 Oct  8  1999 .installed
-rw-rw-r--    1 cmops    cmops     12616 Mar 27 12:41 .installtypes
-rw-rw-r--    1 cmops    cmops      8657 Mar 27 12:41 .sitehostmap
-rw-rw-r--    1 cmops    cmops     72760 Mar 27 12:41 .sitemap
-rw-rw-r--    1 cmops    cmops      1845 Mar 27 12:41 .subsystems
drwxr-xr-x    6 cmops    cmops      4096 Dec 14 09:14 Aadata
drwxrwxr-x    9 cmops    cmops       122 Mar 27 12:35 HDF
drwxrwxr-x    4 cmops    cmops        45 Nov 18 15:01 HDFEOS
drwxrwxrwx    3 cmops    cmops        25 Oct 11  1999 TOOLKIT
drwxrwxr-x    3 cmops    cmops        21 Oct 11  1999 WWW
drwxr-xr-x   27 cmops    cmops      4096 Nov 18 15:02 backup
drwxrwxr-x    6 cmops    cmops       142 Nov 30 15:38 bin
```

```
drwxrwxr-x    2 cmops    cmops         9 Oct 11  1999 cfg
drwxr-xr-x    4 cmops    cmops      4096 Sep 27  1999 daac_toolkit_f77
drwxrwxr-x    7 cmops    cmops        69 May  7  1999 data
drwxrwxr-x    3 cmops    cmops        21 Oct 11  1999 dbms
drwxrwxr-x    6 cmops    cmops        57 Oct 11  1999 lib
drwxrwxr-x    2 cmops    cmops      4096 Apr 18 19:52 logs
drwxr-xr-x    4 root     sys          41 Jun 24  1998 pdps
drwxrwxr-x    2 cmops    cmops        36 Mar 27 12:41 security
drwxr-xr-x    3 root     sys          25 Dec 18  1997 ssit
lrwxr-xr-x    1 cmops    cmops        36 Mar 27 17:13 toolkit ->
/usr/ecs/OPS/CUSTOM/daac_toolkit_f77
drwxrwxr-x    2 cmops    cmops      4096 Mar 27 12:41 utilities
```

- There should be a “toolkit” subdirectory as shown in the example.

8 If there is **no** toolkit directory, notify the Operations Controller/System Administrator to have the directory created and linked correctly.

9 If there is a toolkit directory, notify the Operations Controller/System Administrator that the **auto.profile** file(s) may need to be corrected.

- There is an auto.profile file in the /usr/ecs/*MODE*/CUSTOM/bin/DPS directory on the Queuing Server and on each Science Processor.
- There may be a discrepancy between the auto.profile file and what is specified in the EcDpPrAutosysMkcfg or EcDpScAutosysMkcfg file (in the /usr/ecs/*MODE*/CUSTOM/utilities directory).
  - The EcDpPrAutosysMkcfg (Queuing Server) and EcDpScAutosysMkcfg (Science Processor) files are used in generating the auto.profile files.
- The AutoSys Mkcfg may have to be run again or the auto.profile file may have to be changed manually.

10 If no problem has been identified, run the Execution Manager in the debugger as described in the procedure for **Running Execution Management Outside of AutoSys** (preceding section of this lesson).

- Execution Manager (EcDpPrEM) is the DPS program that runs during Postprocessing.

Handling Failure of Both Execution and Postprocessing Jobs

This condition is indicated when both the Execution and Postprocessing Jobs are red in JobScape or TimeScape, but no other jobs are red. This indicates that the Postprocessing job has read the log file created by EcDpPrRunPGE in the runtime directory and has found an exit status not equal to zero (0). However, it failed to destage (insert) the failed PGE tar file.

The procedure for handling failure of both Execution and Postprocessing jobs starts with the assumption that the Production Monitor has logged in to the Queuing Server (e.g., e0sps04 or l0sps03) and started the **AutoSys GUI Control Panel** and **JobScape**.

Handling Failure of Both Execution and Postprocessing Jobs

---

1 Access a terminal window logged in to the Queuing Server host.

- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 Type **cd /usr/ecs/*MODE*/CUSTOM/logs** then press **Return/Enter**.

- Change directory to the directory containing the data processing log files (e.g., EcDpPrJobMgmt.ALOG, EcDpPrDeletion.ALOG).

3 Type **pg *filename*.err** then press **Return/Enter**.

- ***filename*** refers to the data processing log file to be reviewed (e.g., *DPR#*.err).
- The first page of the log file is displayed.
- Although this procedure has been written for the **pg** command, any UNIX editor or visualizing command (e.g., **vi**, **view**, **more**) can be used to review the log file.

4 Review the .err file for a “FAILPGE” string.

5 Review the .err file for the return value from the Science Data Server around the insertion of the failed PGE tar file.

6 If a problem with insertion of the FAILPGE tar file is suspected, continue with the procedure for **Handling a Failed Insertion Function** (subsequent section of this lesson).

Handling a Failed Insertion Function

If the insertion function fails, the Postprocessing job turns red on JobScape or TimeScape.

The procedure for handling a failed insertion function starts with the assumption that the Production Monitor has logged in to the Queuing Server (e.g., e0sps04 or l0sps03) and started the **AutoSys GUI Control Panel** and **JobScape**.

1 Access a terminal window logged in to the Queuing Server host.

- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 Type **cd /usr/ecs/*MODE*/CUSTOM/logs** then press **Return/Enter**.

- Change directory to the directory containing the data processing log files (e.g., EcDpPrJobMgmt.ALOG, EcDpPrDeletion.ALOG).

3 Type **pg *filename*.err** then press **Return/Enter**.

- ***filename*** refers to the data processing log file to be reviewed (e.g., *DPR#*.err).
- The first page of the log file is displayed.
- Although this procedure has been written for the **pg** command, any UNIX editor or visualizing command (e.g., **vi, view**, **more**) can be used to review the log file.

4 Review the .err file for the following types of error messages:

- **Failure inserting metadata into catalog**
  - Possible causes of a **"Failure inserting metadata into catalog"** message include the following items:
    - Problem with Storage Management.
    - Problem with a SDSRV temporary directory getting filled up.
    - Metadata file (and possibly the data file) cannot be located by Science Data Server because the mount point between the Science Processor and the Science Data Server machine may have been lost.
    - Filenames sent to the Science Data Server are invalid or null (e.g., if the **DpPrFile** table in the PDPS database has duplicate entries).
    - Duplicate file entries in the DPS file tables.
- **Error archiving files**
  - A message that indicates **"Error archiving files"** means that SDSRV is having trouble getting Storage Management to place the file(s) in the archive.

- **Error modifying file usage**
  - A message that indicates **"Error modifying file usage"** means that the **numberOfUsage** column in the **DpPrFile** table for a particular file is at zero (0) and the software is trying to decrement it, which it cannot do.

5 If a **"Failure inserting metadata into catalog"** message was present in the log and the problem occurred for an existing ESDT that has previously worked within the past day or two, notify the Operations Controller or Archive Manager that there may be a problem with Storage Management.

- The STMGT log files may contain information concerning changes/defects in the stored procedures.

6 If a **"Failure inserting metadata into catalog"** message was present in the log and the problem occurred for a new or recently installed ESDT, type **cd /*path*** then press **Return/Enter**.

- Change directory to the run-time directory for the job (e.g., /usr/ecs/OPS/CUSTOM/pdps/x0spg01/data/DpPrRm/x0spg01_disk/ACT#syn1/ ACT#syn1#004130123OPS_x0spg01/).

7 If a **"Failure inserting metadata into catalog"** message was present in the log, type **pg *DPR#*.MCF** then press **Return/Enter**.

- ***DPR#*** refers to the name of the job (e.g., AM1Eph#30012312200OPS or ACT#syn1#004130123OPS).
- The first page of the MCF file is displayed.
- Although this procedure has been written for the **pg** command, any UNIX editor or visualizing command (e.g., **vi**, **view**, **more**) can be used to review the log file.

8 Review the *DPR#*.MCF file in the runtime directory of the PGE/DPR to determine values of the mandatory parameters in the metadata file.

9 If a **"Failure inserting metadata into catalog"** message was present in the log, report values of the mandatory parameters in the metadata file to the Operations Controller or Science Data Specialist so they can be compared with "valids" from the SDSRV database.

- In the GlParameter list from Science Data Server there may be error messages that may indicate which metadata values the Data Server did not like.

10 If a **"Failure inserting metadata into catalog"** message was present in the log, type **cd /*path*** then press **Return/Enter**.

- Change directory to the pdps mount point (e.g., /usr/ecs/OPS/CUSTOM/pdps/x0spg01/data).

11 If the mount point is missing, notify the Operations Controller/System Administrator to have it restored.

12 If a **"Failure inserting metadata into catalog"** message was present in the log, access a terminal window logged in to the Sun internal server host.

- Examples of Sun Consolidation Internal Server host names include **e0acs11** and **l0acs03**.
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

13 If a **"Failure inserting metadata into catalog"** message was present in the log, type **cd /*path*** then press **Return/Enter**.

- Change directory to the pdps mount point (e.g., /usr/ecs/OPS/CUSTOM/pdps/x0spg01/data).
- The pdps mount point should be visible from both the Queuing Server and the Sun internal server hosts (and several other hosts as well).

14 If the mount point is missing, notify the Operations Controller/System Administrator to have it restored.

15 If a **"Failure inserting metadata into catalog"** message was present in the log, log in to the appropriate PDPS database as described in Steps 1 through 5 of the procedure for **Using ISQL to Check Database Tables** (previous section of this lesson).

- For example:

  x0sps02:/usr/ecs/OPS/CUSTOM[4] > isql -U pdps_role –S x0sps02_srvr
  Password:
  1> use pdps
  2> go

16 If a **"Failure inserting metadata into catalog"** message was present in the log, search in the **DpPrFile** table on the **fileName** corresponding to the ".err" log filename that failed to be inserted (refer to Step 4).

- For example:

  1> select fileName from DpPrFile where fileName like "AST_09T#0010214199802012%"
  2> go

  fileName
  --

  AST_09T#00102141998020120000000
  AST_09T#00102141998020120000000.met

(2 rows affected)

- Look for duplicate entries in the table.
- There should be two entries for each file in the **DpPrFile** table; i.e., one for the data file and one for the metadata file (as shown in the example). If there are three or more entries, the table has duplicate entries that are causing the problem.

17 If a **"Failure inserting metadata into catalog"** message was present in the log and if duplicate entries were found in the **DpPrFile** table, notify the Production Planner to delete the DPR whose Insertion job failed.

18 If a **"Failure inserting metadata into catalog"** message was present in the log and if duplicate entries were found in the **DpPrFile** table, clean up the DPS file tables using the procedure for **Cleaning Up the DPS File Tables** (previous section of this lesson).

19 If a **"Failure inserting metadata into catalog"** message was present in the log, when the DPS file tables have been cleaned up, notify the Production Planner to recreate the DPR whose Insertion job failed.

20 If an **"Error archiving files"** message was present in the log, notify the Operations Controller or Archive Manager that there may be a problem with Storage Management.

- An **"Error archiving files"** message means that SDSRV is having trouble getting Storage Management to place the file(s) in the archive.
- When the **"Error archiving files"** has been corrected, it should be possible to restart the job and have it complete successfully. (Refer to **Force-Starting a Job**.)

21 If an **"Error archiving files"** message was present in the log, access a terminal window logged in to the Sun Consolidation Internal Server host.

- Examples of Sun Consolidation Internal Server host names include **e0acs11** and **l0acs03**.
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

22 If an **"Error archiving files"** message was present in the log, type **cd /*path*** then press **Return/Enter**.

- Change directory to the pdps mount point (e.g., /usr/ecs/OPS/CUSTOM/pdps/x0spg01/data).

23 If the mount point is missing, notify the Operations Controller/System Administrator to have it restored.

24 If an **"Error modifying file usage"** message was present in the log, log in to the applicable PDPS database.

- Refer to the example in Step 15.

- A message that indicates **"Error modifying file usage"** means that the **numberOfUsage** column in **DpPrFile** table for a particular file is at zero (0) and the software is trying to decrement it. The column is an increment/decrement counter and is not normally decremented more times than it is incremented when under software control. However, if someone manually changes the database then the value may get out of sync.

25 If an **"Error modifying file usage"** message was present in the log, update the **numberOfUsage** column in **DpPrFile** table for the particular file so it is set at one (1).

- For example:

  1> update DpPrFile set numberOfUsage=1 where fileName like "AST_09T#0010214199802012%"
  2> go

- When the database has been corrected, it should be possible to restart the job and have it complete successfully. (Refer to **Force-Starting a Job**.)

26 If no problem has been identified, run the Execution Manager in the debugger as described in the procedure for **Running Execution Management Outside of AutoSys** (preceding section of this lesson).

- Execution Manager (EcDpPrEM) is the DPS program that runs during insertion.

Handling a Failed Deallocate Function

If the deallocate function fails, the Postprocessing job turns red on JobScape or TimeScape.

The procedure for handling a failed deallocate function starts with the assumption that the Production Monitor has logged in to the Queuing Server (e.g., e0sps04 or l0sps03) and started the **AutoSys GUI Control Panel** and **JobScape**.

Handling a Failed Deallocate Function

---

1 Ensure that it is possible to connect to the necessary hosts and servers (listed in Table 6).

- For detailed instructions refer to the section on **Checking Connections to Hosts/Servers** (subsequent section of this lesson).

2 If hosts/servers are all “up,” check the log files (e.g., *DPR#*.ALOG) in the /usr/ecs/*MODE*/CUSTOM/logs directory for error messages.

- For detailed instructions refer to the procedure for **Checking Log Files** (subsequent section of this lesson).

Handling a Failed On-Demand Processing Request

An On-Demand Processing request can fail for any of the following reasons:

- Failures in Submitting the On-Demand Request from EDG.
- Failures in OdMgr getting the On-Demand Request to Run in AutoSys.
  - The Job Management Server is down.
  - The DPR is waiting in the AutoSys queue (never got released).
  - The DPR failed in Job Management due to Invalid DPR.
  - The DPR failed in OdMgr due to a failure in getting the PGE ID.
  - Subscription Server Problems.
  - The DPR failed in Job Management due to a JIL failure.
  - The DPR failed in Job Management due to an AutoSys ID failure.
  - The DPR failed to be received by Job Management Server.
  - AutoSys is not functional.
  - AutoSys is full.
- Failures in AutoSys of the On-Demand PGE(s).
- Failures in OdMgr Distribution of the On-Demand product.

When the DPR for an On-Demand Processing Request is in AutoSys, the jobs can fail for any of the same reasons that any other type of job might fail and might be restarted in the same manner as any other type of job (as described in preceding sections).

Perform the appropriate procedure(s) (as necessary) related to handling a failed on-demand processing request:

- **Checking Connections to Hosts/Servers**.
- **Checking Log Files**.
- **Responding to Hanging of the Processing System**.
- **Checking AutoSys Status**.
- **Checking the AutoSys Log**.
- **Checking for Database Deadlocks**.
- **Checking for Resource Locks in the PDPS Database**.
- **Responding to Failure of Jobs to Start in AutoSys**.

- **Checking Job Management Server Status**.
- **Checking to Determine Whether the DPR Is Waiting in the AutoSys Queue**.
- **Using ISQL to Check Database Tables**.
- **Checking to Determine Whether AutoSys Is Full**.
- **Responding to a Condition Where a DPR Was Released But Failed Due to a JIL Failure**.
- **Handling Subscription Server Problems**.
- **Responding to a DPR That Was Released But Failed Due to an AutoSys ID Failure**.
- **Responding to a DPR That Was Released But Failed Due to Invalid DPR**.
- **Responding to a DPR That Was Released But Failed to Be Received by Job Management Server**.
- **Responding to a Single DPS Job That Has Failed or Is Hanging**.
- **Handling a Box Job that is Hanging in AutoSys**.
- **Handling a Hanging Allocation Function**.
- **Running Execution Management Outside of AutoSys**.
- **Handling a Failed Allocation Function**.
- **Force-Starting a Job**.
- **Responding to a Restart of a Job That Fails Although All Known Problems Have Been Corrected**.
- **Handling a Hanging Staging Function**.
  – Perform the **Handling a Hanging Allocation Function** procedure.
- **Handling a Failed Staging Function**.
- **Cleaning Up the DPS File Tables**.
- **Handling a Failed Preprocessing Job**.
- **Handling a Hanging Execution Job**.
  – Perform the **Checking AutoSys Status** procedure.
- **Handling a Failed Execution Job**.
  – Perform the **Checking AutoSys Status** procedure.

- **Responding to Execution Job and/or Postprocessing Job That Have(Has) Failed**.
- **Responding to Execution Job That Has Failed and the DPR Has Gone into "Failed-PGE" Processing**.
- **Handling a Failed Postprocessing Job**.
- **Handling Failure of Both Execution and Postprocessing Jobs**.
- **Handling a Failed Insertion Function**.
- **Handling a Failed Deallocate Function**.
- **Responding to a DPR that Failed in OdMgr because the PGE ID Could Not Be Found**.

Responding to a DPR that Failed in OdMgr because the PGE ID Could Not Be Found

If a DPR failed in OdMgr because the PGE ID could not be found, the options selected on the EDG screen do not match any of the Profiles for the PGE associated with the selected input. The parameters selected on EDG for the product are passed to OdMgr and are used to select the appropriate PGE Profile. The selection of a PGE Profile fails if the parameters passed by EDG to OdMgr do not exactly match those in one of the profiles.

The procedure for responding to a DPR that failed in OdMgr because the PGE ID could not be found starts with the assumption that the Production Monitor has logged in to the Queuing Server (e.g., e0sps04 or l0sps03).

Responding to a DPR that Failed in OdMgr because the PGE ID Could Not Be Found

---

1 Access a terminal window logged in to the Queuing Server host.

- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 Type **cd /usr/ecs/*MODE*/CUSTOM/logs** then press **Return/Enter**.

- Change directory to the directory containing the data processing log files (e.g., EcDpPrJobMgmt.ALOG, EcDpPrDeletion.ALOG).

3 Type **pg OdMgr.ALOG** then press **Return/Enter**.

- The first page of the log file is displayed.
- Although this procedure has been written for the **pg** command, any UNIX editor or visualizing command (e.g., **vi, view**, **more**) can be used to review the log file.

4 Review the ALOG file for a "**Getting PGEID failed from PlOrderFactory"** type of error message.

- For example:

  PID : 24890:MsgLink :0 meaningfulname :READFAILED
  Msg: Getting PGEID failed from PlOrderFactory Priority: 2 Time : 01/27/00 13:41:43
  PID : 24890:MsgLink :10 meaningfulname :PGELISTFAILED
  Msg: Getting PgeList from GlParameterList failed Priority: 2 Time : 01/27/00 13:41:54
  PID : 24890:MsgLink :11 meaningfulname :CANTCREATEORDER
  Msg: Unable to create a PlOrder Priority: 2 Time : 01/27/00 13:41:54

- The problem is most likely caused by not having the latest ODLs.

5 Notify the SSI&T team of the problem.

- The SSI&T team should recopy the On Demand ODLs and re-register the PGEs.

Checking Log Files

Log files can provide indications of the following types of problems:

- Communication problems.
- Database problems.
- Lack of disk space.

The procedure for checking log files starts with the assumption that the operator has logged in to the system.

Checking Log Files

---

1 Access a terminal window logged in to the Planning/Management Workstation, Queuing Server, or PDPS DBMS Server as applicable.

- In general Planning Subsystem applications are installed on the Planning/ Management Workstation (e.g., e0pls03 or l0pls02) and Data Processing Subsystem (PRONG) applications are installed on the Queuing Server (e.g., e0sps04 or l0sps03). Note the following exceptions:
  - QA Monitor is on the Planning/Management Workstation.

- Subscription Manager is on the Queuing Server (e.g., e0sps04 or l0sps03).

- Examples of Planning/Management Workstation host names include **e0pls03** and **l0pls02.**
- Examples of Queuing Server host names include **e0sps04** and **l0sps03**.
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 Type **cd /usr/ecs/*MODE*/CUSTOM/logs** then press **Return/Enter**.

- Change directory to the directory containing the log files (e.g., EcDpPrJobMgmt.ALOG, EcDpPrDeletion.ALOG, EcPlSubMgrDebug.log, or EcDpPrJobMgmtDebug.log).

3 Type **pg *filename*** then press **Return/Enter**.

- ***filename*** refers to the log file to be reviewed (e.g., EcDpPrJobMgmt.ALOG, EcDpPrDeletion.ALOG, *DPR#*.ALOG, *DPR#*.err, EcPlSubMgrDebug.log, EcDpPrJobMgmtDebug.log).
- The first page of the log file is displayed.
- Although this procedure has been written for the **pg** command, any UNIX editor or visualizing command (e.g., **vi**, **view**, **more**) can be used to review the log file.

4 Review the log file to identify problems that have occurred.

5 Respond to problems as follows:

- Production Processing-related problems.
  - Perform the appropriate procedure(s) from the list near the beginning of the **Production Processing Troubleshooting Procedures** section.
- Communication problems.
  - Notify the Operations Controller/System Administrator of suspected communication problems.
- Database problems.
  - Verify that relevant database servers are running.
  - Check for lack of (or corruption of) data in the database using either a database browser or isql commands.
  - Notify the Database Administrator of suspected database problems.

- Lack of disk space.
  - Remove unnecessary files.
  - Notify the Operations Controller/System Administrator of recurring disk space problems.

Launching the QA Monitor

Launching the QA Monitor

Access to the QA Monitor must be gained through the use of UNIX commands. The process starts with the assumption that the applicable servers are running and the Production Monitor has logged in to the system.

Launching the QA Monitor

1 Access a terminal window logged in to the Planning/Management Workstation host.

- Examples of Planning/Management Workstation host names include **e0pls03** and **l0pls02.**
- For detailed instructions refer to the procedure for **Logging in to System Hosts** (preceding section of this lesson).

2 Type **setenv ECS_HOME /usr/ecs/** then press the **Return/Enter** key.

- When logging in as a system user (e.g., cmshared), the ECS_HOME variable may be set automatically so it may not be necessary to perform this step.

3 Type **cd /usr/ecs/*MODE*/CUSTOM/utilities** then press **Return/Enter**.

- Change directory to the directory containing the QA Monitor start script (e.g., EcDpPrQaMonitorGUIStart).
- The ***MODE*** will most likely be one of the following operating modes:
  - OPS (for normal operation).
  - TS1 (for SSI&T).
  - TS2 (new version checkout).
- Note that the separate subdirectories under /usr/ecs apply to (describe) different operating modes.

4 Type **EcDpPrQaMonitorGUIStart** ***MODE*** then press **Return/Enter** to launch the **QA Monitor** GUI.

- The **QA Monitor** GUI (Figure 82) is displayed.

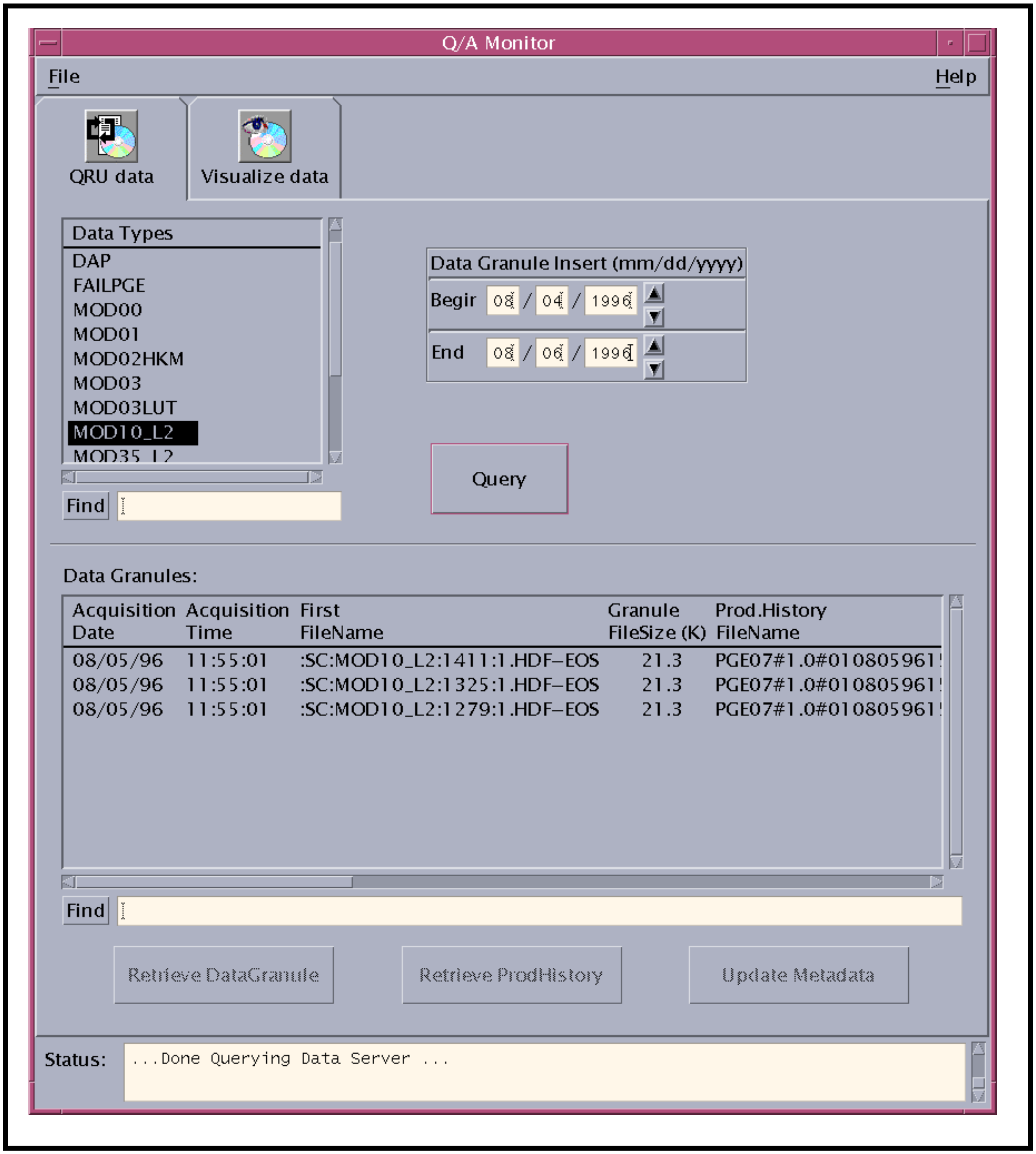

Figure 82. Q/A Monitor GUI - QRU Data Tab

Performing Science Product Quality Assurance

Science Product Quality Assurance

Science product quality assurance (QA) involves the use of the Q/A Monitor application. Science Computing Facility (SCF) personnel have the responsibility for performing QA of their products. The Production Monitor's role in QA is limited to updating the QA metadata. The procedure for updating QA metadata is performed in response to a request from SCF personnel to set the metadata flags on specified granule(s).

The procedure for updating QA metadata starts with the assumption that all applicable servers are currently running and the **Q/A Monitor** GUI **QRU data** tab (Figure 82) (QRU = Query, Retrieve and Update) is being displayed.

Updating Quality Assurance (QA) Metadata

1 In the **Data Types** field, click on the data type to be checked.

- It may be necessary to scroll through the **Data Types** list.
- The selected data type is highlighted.
- Only one data type can be selected at a time.
- The **Find** button provides a means of performing a keyword search of the **Data Types** window.
  - A keyword search is performed by typing relevant text in the text entry box to the right of the **Find** button then clicking on the **Find** button.
    - The first entry in the **Data Types** window that has the search text is highlighted.
    - Click on the **Find** button again (as necessary) to highlight additional occurrences of the search text.
  - The **Find** field is case-sensitive.

2 Click in the appropriate **Data Granule Insert** window field(s) and either type or use the up/down arrow buttons to enter the **Begin** date and **End** date in ***MM/DD/YYYY*** format.

- In the **Data Granule Insert** window it is necessary to specify the range of dates (between the **Begin** date and the **End** date) to formulate a query for searching for the desired granule(s) to be checked.

- Time is based upon day of insert into the data server. If no dates are entered, an error message is displayed.
- The up and down arrows next to the duration fields may be used for modifying entries in each field.
- The **Tab** key may be used to move from field to field.

3 Click on the **Query** button.

- Granules within the specified date range appear in the **Data Granules** field.

4 In the **Data Granules** field, click on the granule for which metadata is to be updated.

- It may be necessary to scroll through the list of granules.
- The selected granule is highlighted.
- The **Find** button provides a means of performing a keyword search of the **Data Granules** window.
  - A keyword search is performed by typing relevant text in the text entry box to the right of the **Find** button then clicking on the **Find** button.
    - The first entry in the **Data Granules** window that has the search text is highlighted.
    - Click on the **Find** button again (as necessary) to highlight additional occurrences of the search text.
  - The **Find** field is case-sensitive.

5 Click on the **Update Metadata** button.

- The **Granule Parameters** window (Figure 83) is displayed.
- The **Granule Parameters** window displays one line for each parameter for the selected granule.

6 In the **Granule Parameters** window click on a parameter for which the metadata is to be updated.

- The **Update Meta Data** window (Figure 84) is displayed.
- The **Find** button provides a means of performing a keyword search of the **Granule Parameters** window.
  - A keyword search is performed by typing relevant text in the text entry box to the right of the **Find** button then clicking on the **Find** button.
    - The first entry in the **Data Granule Parameters** window that has the search text is highlighted.

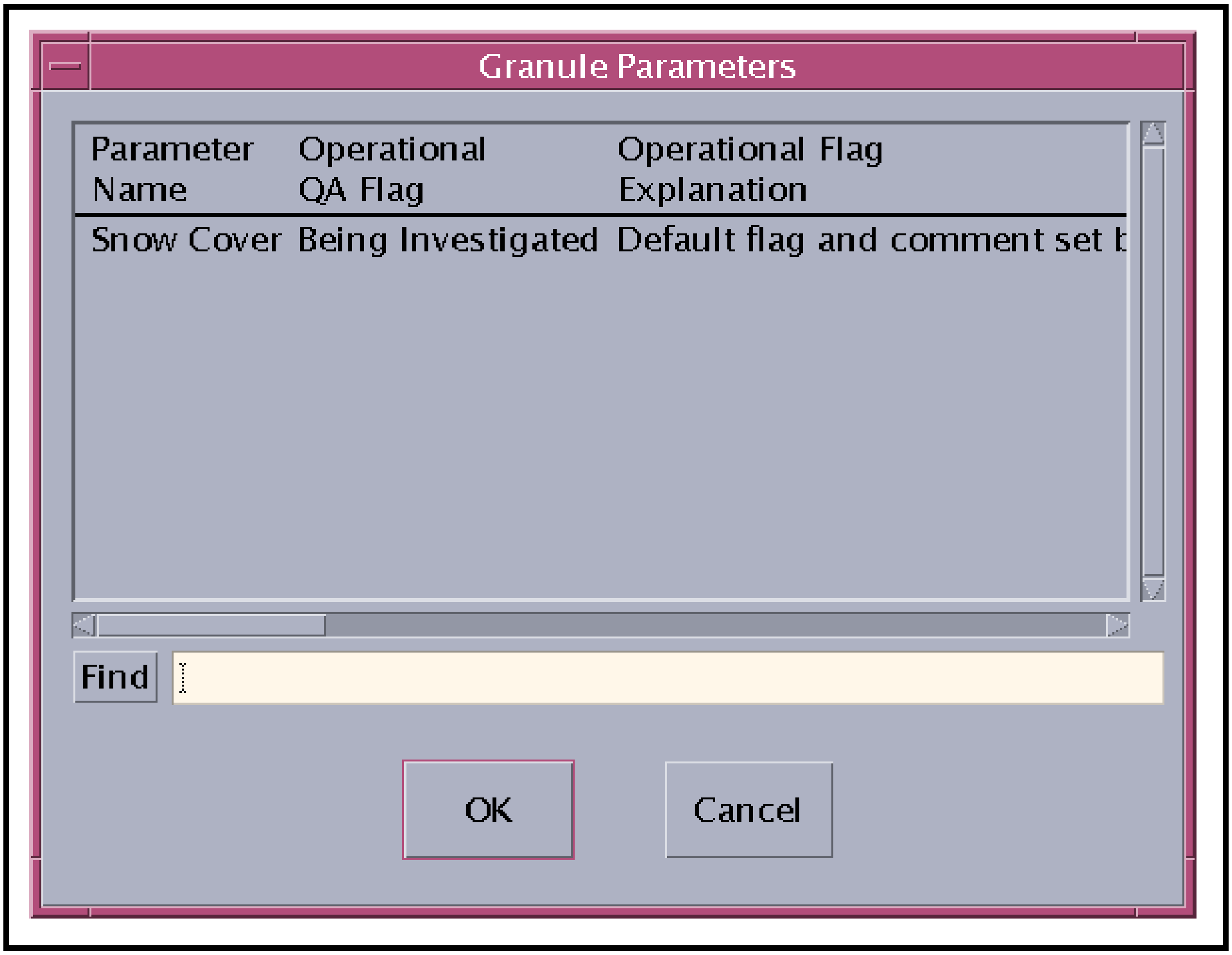

Figure 83. QA Monitor Granule Parameters Window

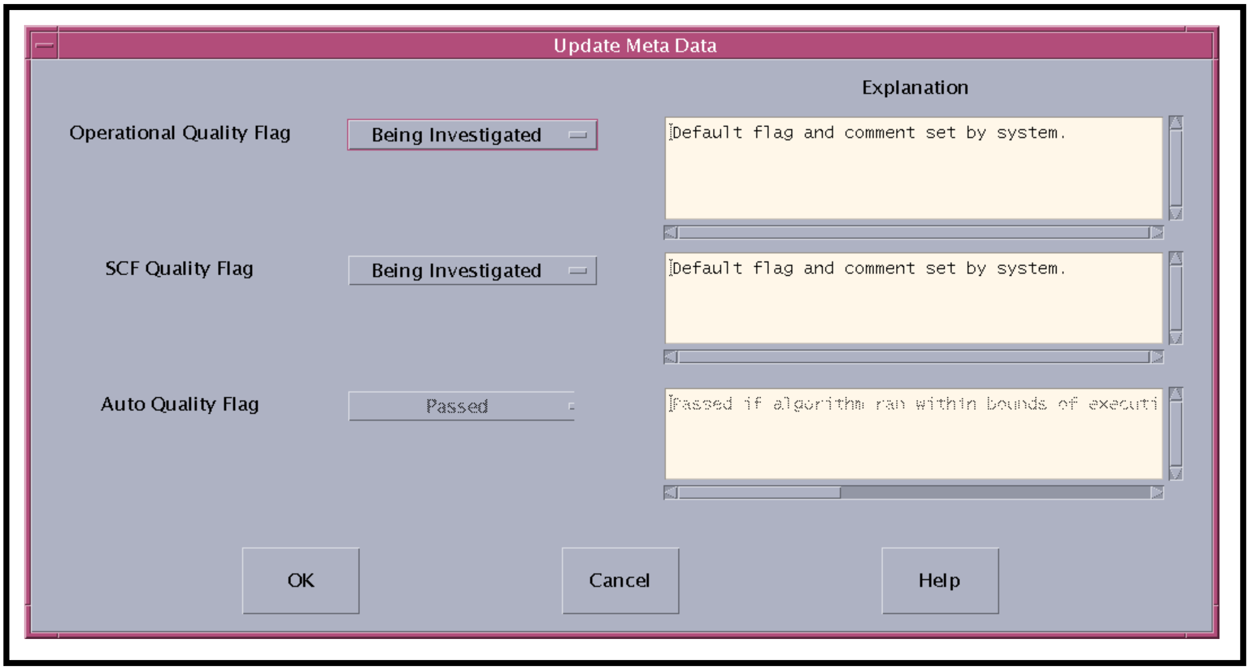

Figure 84. QA Monitor Update Meta Data Window

- Click on the **Find** button again (as necessary) to highlight additional occurrences of the search text.

7 Click and hold on the **Operational QA Flag** option button, move the mouse cursor to the desired selection (highlighting it), then release the mouse button.

- The selected metadata flag is displayed on the **Operator QA Flag** option button.
- The following options are available:
  - **Passed.**
  - **Failed.**
  - **Being Investigated.**
  - **Not Being Investigated.**

8 Type an explanation of the QA flag selection in the **Explanation** field.

9 If the SCF has specified that the SCF Quality Flag should be set to a particular value, click and hold on the **SCF Quality Flag** option button, move the mouse cursor to the SCF-specified selection (highlighting it), then release the mouse button.

- The selected metadata flag is displayed on the **SCF Quality Flag** option button.

- The same options are available as those on the **Operational Quality Flag** option button.

10 Type an explanation of the QA flag selection in the **Explanation** field.

NOTE: The **Auto Quality Flag** option button should not be accessible.

11 Click on the appropriate button from the following selections:

- **OK** - to accept the QA flag settings and dismiss the **Update Meta Data** window.
  - The **Granule Parameters** window (Figure 83) is displayed.
- **Cancel** - to cancel any changes to the QA flag settings and dismiss the **Update Meta Data** window.
  - The **Granule Parameters** window (Figure 83) is displayed.

NOTE: The **Help** buttons at the bottom of various **QA Monitor** windows are non-functional (do not work).

12 Observe the entries in the **Granule Parameters** window to verify that the QA flag settings have actually been applied to the granule.

- The QA flag values and explanations entered using the **Update Meta Data** window are displayed.
- Repeat Steps 6 through 11 as necessary to revise the QA metadata for the granule parameter.

13 Repeat Steps 6 through 12 to update the QA metadata for any additional granule parameters.

14 When the QA flags for all relevant parameters have been set with the desired values and verified, click on the **OK** button in the **Granule Parameters** window.

- The **Granule Parameters** window is dismissed.
- The directory for visualizing data retrieved from the archive is as follows: /usr/ecs/*MODE*/CUSTOM/data/DPS

This page intentionally left blank.

Regenerating Granules

Regenerating Granules in Response to Loss of Files from the Archive

The reason for regenerating granules is to produce replacements for previously generated granules that have been lost or corrupted due to failure in the archive. The overall process involves the following general operations:

- Retrieval of the Production History files (PH) for lost granules to determine parameters for the generation of replacement granules.
- Creating Production Requests for the generation of replacement granules.
- Creating and activating a Production Plan that includes the Production Requests for the generation of replacement granules.
- Preparing (if applicable) a "PDPS Residual Granules List," which identifies granules that either cannot or should not be regenerated at the DAAC.
  - Some granules do need not be reproduced; e.g., if there is a more recent version of the product available.

The regeneration process is initiated when the Production Planner receives a list of "Granules for PDPS Re-Generation." The list contains information about the granules to be regenerated and Universal References (URs) for the associated Production History tar files. The list is the product of a Science Data Server (SDSRV) procedure concerning SDSRV Retrieval of Granule Production History Metadata.

The procedure for regenerating granules in response to loss of files from the archive starts with the assumption that all applicable production planning servers and data server servers are running.

Regenerating Granules in Response to Loss of Files from the Archive

1 Determine which granules in the **Granules for PDPS Re-Generation** list should be regenerated (and which granules do not need to be reproduced).

2 Add granules that either cannot or do not need to be reproduced to the **PDPS Residual Granules** list.

3 Retrieve (using the **QA Monitor GUI**) the Production History tar file from the archive for each granule in the **Granules for PDPS Re-Generation** list that needs to be reproduced.

- Use the lost granule's datatype, and begin date and end date values that encompass its RangeBeginningDateTime and RangeEndingDateTime.
- **QA Monitor** interprets dates in UTC format.
- For each granule that meets the query conditions and is displayed on the **QA Monitor GUI**, the granule's UR, its Production History tar file's UR, and the name of the Production History tar file are shown. For only one of the granules will the URs (both the granule UR and the Production History UR) match the URs for this granule in the input list.
- The Production History tar file is acquired to a directory that is configurable.
  - The name of the configuration parameter is DpPrQA_DATA_DIR.
  - The default value for the parameter is $ECS_HOME/<*MODE*>/CUSTOM/data/DPS.
- If more than one granule in the input list maps to the same Production History tar file, the Production History tar file need not be retrieved multiple times.

4 Extract information needed to re-run the PGE from the Process Control File (PCF) in the PH.

- Information to be extracted from the PCF:
  - PGE Name.
  - PGE Version.
  - PGE Profile ID.
  - DPR Start time.
  - DPR Stop time.
  - PGE runtime parameters and their associated values.
- Identification of information in the PCF:
  - The PGE Name, PGE Version, and the PGE Profile appear in the System Runtime Parameters section of the PCF. They are concatenated (with a # sign to separate them) and appear in the place reserved for "Software ID".
  - DPR Start time appears in the User Defined Parameter Section of the PCF under the logical ID 10258.

– DPR Stop time also appears in the User Defined Parameter Section of the PCF under logical ID 10259.

– All other logical IDs in the User Defined Parameter Section of the PCF form the run time parameters and their associated values. Note the logical ID and its corresponding values.

5 If the Query failed or did not return any hit that matched, add the granule to the **PDPS Residual Granules** list.

6 If the PGE name (including version and profile) that is extracted from the PCF does not appear as an Existing/New PGE, add the granule to the **PDPS Residual Granules** list.

7 From the SSIT host, launch the **SSIT Manager** GUI and invoke the **PDPS Operational Metadata** GUI.

8 Invoke the **PDPS Operational Metadata** GUI.

9 If the PGE is not registered, register the PGE using the **PDPS Science Update Metadata Update** from the **SSIT Manager** GUI.

- The PGE must be registered before a PR can be entered.

10 If it is decided not to re-register the PGE, add the granule to the **PDPS Residual Granules** list.

11 Launch the **Production Request Editor** GUI.

12 Create a Production Request for the relevant PGE/version/profile ID.

- Use **Reprocessing** for the **Processing Type**.
- Use the DPR Start and Stop Time listed in the Production History for the **Begin** and **End** times.
- Compare the default PGE runtime parameters with the runtime parameters obtained from the Production History tar file.
  – Modify the runtime parameter values to match exactly what was used in the original run.
- If granules to be regenerated are produced by PGEs that are chained, the PRs must be entered in the proper order.
  – For instance, if granules A and B are to regenerated, and PGEs P1 & P2 produce them and if P1 & P2 are chained (P2 takes P1's outputs as its inputs), then the production request for P1 must be entered before entering one for P2.

13 Launch the **Planning Workbench**.

14 Create and activate a production plan that includes the newly created Production Request(s).

15 Send the **PDPS Residual Granules** list to the originator of the **Granules for PDPS Re-Generation** list.

Practical Exercise

Introduction

This exercise is designed to give the students practice in production planning and processing activities.

Equipment and Materials

One workstation per student.

Statement of the requirements for the exercise.

Release 7.11 Operations Tools Manual for the EMD Project, 609-EMD-001, one copy per student.

Release 7.11 Mission Operation Procedures for the EMD Project, 611-EMD-001, one copy per student.

Logging in to System Hosts

The exercise involves logging in to system hosts. The exercise begins with a student acting in the role of Production Planner or Production Monitor receiving the necessary information/ requirements for logging in to a system host. The student logs in to a system host as specified in the requirements.

Perform the following steps:

1. Access the command shell.
2. Set the DISPLAY environmental variable.
3. Log in to the specified host using secure shell and the specified user ID.

Launching the Production Request Editor

The exercise involves launching the Production Request Editor using UNIX commands. The exercise begins with a student acting in the role of Production Planner recognizing the need to launch the Production Request Editor. The student launches the Production Request Editor as specified in the requirements.

Perform the following steps:

1. Log in to the Planning/Management Workstation.
2. Set environmental variables if necessary.
3. Start the Production Request Editor GUI in the appropriate mode.

Creating a New Production Request Using the Production Request Editor GUI

The exercise involves the preparation of a new production request. The exercise begins with a student acting in the role of Production Planner receiving the necessary information/requirements for creating a new production request. The student prepares a new production request that is consistent with the requirements.

Perform the following steps:

1. Access the Production Request Editor.
2. Select the PR Edit tab on the Production Request Editor GUI.
3. Prepare a new production request that is consistent with the written or stated requirements.
4. Save the new production request.

Creating New Production Requests Using the Production Request Generator (Command-Line Interface)

The exercise involves the preparation of new production requests using the PR Generator (command-line interface). The exercise begins with a student acting in the role of Production Planner receiving the necessary information/requirements for creating new production requests. The student prepares new production requests that are consistent with the requirements.

Perform the following steps:

1. Log in to the appropriate host.
2. Prepare an input file specifying the PgeId and GEOId values to be used in creating the production requests.
3. Start the PR Generator to create new production requests.
4. Check the PR Generator debug log to determine the results of running the PR Generator.

Editing/Modifying a Production Request

The exercise requires the editing of a production request. The exercise begins with a student acting in the role of Production Planner receiving the necessary information/requirements for editing an existing production request. The student modifies the production request consistent with the requirements.

Perform the following steps:

1. Access the Production Request Editor.
2. Select the PR Edit tab on the Production Request Editor GUI.
3. Select the Production Request to be modified.
4. Make production request modifications consistent with the written or stated requirements.
5. Save the modified production request.

Deleting a Production Request

The exercise involves deleting a production request. The exercise begins with a student acting in the role of Production Planner receiving the necessary information/requirements for deleting an existing production request. The student deletes the production request as specified in the requirements.

Perform the following steps:

1. Select the PR List tab on the Production Request Editor GUI.
2. Select the production request to be deleted from those listed.
3. Delete the production request.

Reviewing Data Processing Requests

The exercise involves reviewing data processing requests. The exercise begins with a student acting in the role of Production Planner being directed to review specific data processing requests to determine specified characteristics. The student reviews the data processing requests consistent with the requirements.

Perform the following steps:

1. Select the DPR List tab on the Production Request Editor GUI.
2. Select a Production Request from the list on the option button.
3. Select a DPR from the list displayed.
4. Open the DPR.
5. Respond without error to questions concerning the characteristics of the DPR.

Deleting a Data Processing Request

The exercise involves deleting a data processing request. The exercise begins with a student acting in the role of Production Planner being directed to delete a specific data processing request. The student deletes the data processing request as specified in the requirements

Perform the following steps:

1. Select the DPR List tab on the Production Request Editor GUI.
2. Select the appropriate Production Request from the list on the option button.
3. Select the DPR to be deleted from the list displayed.
4. Delete the DPR.

Launching the Production Strategies GUI

The exercise involves launching the Production Strategies GUI using UNIX commands. The exercise begins with a student acting in the role of Production Planner recognizing the need to launch the Production Strategies GUI. The student launches the Production Strategies GUI as specified in the requirements.

Perform the following steps:

1. Log in to the Planning/Management Workstation.
2. Set environmental variables if necessary.
3. Start the Production Strategies GUI in the appropriate mode.

Launching Planning Workbench-Related GUIs

The exercise involves launching planning workbench-related GUIs using UNIX commands. The exercise begins with a student acting in the role of Production Planner recognizing the need to launch planning workbench-related GUIs. The student launches planning workbench-related GUIs as specified in the requirements.

Perform the following steps:

1. Log in to the Planning/Management Workstation.
2. Set environmental variables if necessary.
3. Start the Planning Workbench GUI and the Planning Master Timeline in the appropriate mode.

Defining a Production Strategy

The exercise involves the preparation of a production strategy. The exercise begins with a student acting in the role of Production Planner receiving the necessary information/requirements for creating a production strategy. The student prepares a production strategy that is consistent with the requirements.

Perform the following steps:

1. Select priorities for the values for PR Type, User Type, and PGE Type.
2. Type weights for the preceding three DPR attributes (as needed).
3. Type a weight in the Production Request Editor field.
4. Click on the Normalize button.
5. Type delta priority for Late Start Delta (if needed).
6. Save the Production Strategy.

Creating a New Production Plan

The exercise involves the preparation of a new production plan. The exercise begins with a student acting in the role of Production Planner receiving the necessary information/requirements for creating a new production plan. The student prepares a new production plan that is consistent with the requirements.

Perform the following steps:

1. Access the Planning Workbench.
2. Prepare a new production plan that is consistent with the written or stated requirements.
3. Save the new production plan.
4. Activate the plan (if specified in the requirements).

Reviewing a Plan Timeline

The exercise involves reviewing a production plan timeline. The exercise begins with a student acting in the role of Production Planner receiving the necessary information/requirements for reviewing a production plan timeline. The student reviews the specified production plan timeline and responds to questions concerning timeline characteristics.

Perform the following steps:

1. Access the specified Production Planning Master Timeline.
2. Adjust the timeline display as necessary to view the specified production requests.

3. Review the specified Production Planning Master Timeline.

4. Respond without error to questions concerning the Production Planning Master Timeline.

Cleaning the PDPS Database and Science Processing Disks

The exercise involves cleaning the PDPS database and science processing disks. The exercise begins with a student acting in the role of Production Planner receiving the necessary information/requirements for cleaning the PDPS database and science processing disks. The student takes action to clean the PDPS database in accordance with the requirements.

Perform the following steps:

1. Log in to the Planning/Management Workstation using secure shell.

2. Set the ECS_HOME environmental variable if necessary.

3. Start the EcPlDbClean script using the appropriate arguments.

4. Log in to the Queuing Server host using secure shell.

5. Set the ECS_HOME environmental variable if necessary.

6. Start the appropriate script using suitable arguments.

Troubleshooting Production Planning Problems

The exercise involves troubleshooting production planning problems. The exercise begins with a student acting in the role of Production Planner receiving the necessary trouble symptom information and requirements for troubleshooting the problem(s). The student reviews the specified trouble symptoms, takes action to correct the problem(s), and responds to questions concerning the possible cause(s).

Perform the following steps:

1. Review the trouble symptoms.

2. Check the status of relevant hosts/servers as necessary.

3. Check log files as necessary.

4. Take action to correct the problem(s).

5. Respond without error to questions concerning the possible cause(s).

Launching the AutoSys GUI Control Panel

The exercise involves launching the AutoSys GUI Control Panel using UNIX commands. The exercise begins with a student acting in the role of Production Monitor recognizing the need to launch the AutoSys GUI Control Panel. The student launches the AutoSys GUI Control Panel as specified in the requirements.

Perform the following steps:

1. Log-in to the Queuing Server host.
2. Set environmental variables if necessary.
3. Source the appropriate file.
4. Start the GUI for the appropriate instance of AutoSys.

Configuring AutoSys Runtime Options

The exercise involves the configuration of AutoSys runtime options. The exercise begins with a student acting in the role of Production Monitor receiving the necessary information/ requirements for configuring AutoSys runtime options. The student configures AutoSys consistent with the requirements.

Perform the following steps:

1. Launch the AutoSys GUI Control Panel.
2. Access the AutoSys functions specified in the written or stated requirements.
3. Select the AutoSys runtime options specified in the written or stated requirements.
4. Apply the AutoSys runtime options specified in the written or stated requirements.

Monitoring/Controlling Job Processing

The exercise involves monitoring/controlling job processing. The exercise begins with a student acting in the role of Production Monitor receiving the necessary information/requirements for monitoring/controlling job processing. The student monitors/controls job processing as specified in the requirements and responds without error to questions concerning information displayed on the AutoSys displays.

Perform the following steps:

1. Access JobScape.
2. Access TimeScape.
3. Access the Job Activity Console GUI.
4. Configure AutoSys runtime options for JobScape and/or TimeScape.
5. If necessary, select jobs to be displayed on JobScape and/or TimeScape.
6. If necessary, generate a list of jobs to be displayed on the Job Activity Console GUI.
7. Respond without error to questions concerning information displayed on JobScape, TimeScape, and/or the Job Activity Console GUI.
8. Control job processing as appropriate.

Responding to Alarms (Including Selecting Alarms to Be Displayed)

The exercise involves reviewing AutoSys alarms and selecting alarms to be displayed. The exercise begins with a student acting in the role of Production Monitor receiving the necessary information/ requirements for selecting and reviewing AutoSys alarms. The student reviews and selects AutoSys alarms as specified in the requirements and responds without error to questions concerning alarms.

Perform the following steps:

1. Access the Alarm Manager through the Ops Console.
2. Respond without error to questions concerning alarms.
3. Exit from the AutoSys Alarm Manager.

Specifying Job Selection Criteria and Reviewing Job Activities

The exercise involves specifying job selection criteria and reviewing job activities using AutoSys. The exercise begins with a student acting in the role of Production Monitor receiving the necessary information/requirements for specifying job selection criteria and reviewing job activities using AutoSys. The student specifies job selection criteria and reviews job activities using AutoSys as specified in the requirements and responds without error to questions concerning job activities.

Perform the following steps:

1. Access the AutoSys Job Activity Console.
2. Specify job selection criteria as specified in the written or stated requirements.
3. Respond without error to questions concerning job activities.
4. Exit from the AutoSys Job Activity Console.

Determining the Ownership of an AutoSys Job

The exercise involves determining the ownership of an AutoSys job using the AutoSys Job Definition GUI. The exercise begins with a student acting in the role of Production Monitor receiving the necessary information/requirements for determining the ownership of an AutoSys job. The student determines the ownership of an AutoSys job as specified in the requirements and responds without error to questions concerning ownership of the job.

Perform the following steps:

1. Access the pop-up menu from the relevant job on either JobScape or TimeScape.
2. Access the AutoSys Job Definition GUI from the pop-up menu.
3. Respond without error to questions concerning ownership of the job.
4. Exit from the Job Definition GUI.

Modifying Job Status (Including Sending an Event to a Job) Using AutoSys

The exercise involves modifying job status using AutoSys. The exercise begins with a student acting in the role of Production Monitor receiving the necessary information/requirements for modifying job status using AutoSys. The student modifies job status using AutoSys as specified in the requirements.

Perform the following steps:

1. Access the AutoSys Job Activity Console (Ops Console).
2. Specify job selection criteria for the AutoSys Job Activity Console.
3. Select the job with the status to be modified.
4. Click on the button corresponding to the desired action to be taken with respect to the selected job (if there is a corresponding button in the Actions region of the Job Activity Console).
5. If there is no button corresponding to the desired action, access the Send Event GUI by clicking on the Send Event button in the Actions Region of the Job Activity Console.
6. Make appropriate entries/selections on the Send Event GUI.
7. Send the event to the job from the Send Event GUI.

Reviewing Activity Reports and Job Dependency Reports

The exercise involves reviewing activity reports and job dependency reports. The exercise begins with a student acting in the role of Production Monitor receiving the necessary information/ requirements for reviewing activity reports and job dependency reports. The student reviews an activity report and a job dependency report as specified in the requirements and responds without error to questions concerning the activity report and job dependency report.

Perform the following steps:

1. Access the command shell.
2. Type the appropriate command for an activity report as specified in the requirements.
3. Respond without error to questions concerning the activity report.
4. Type the appropriate command for a job dependency report as specified in the requirements.
5. Respond without error to questions concerning the job dependency report.

Defining and Running Monitors/Browsers

The exercise involves defining and running monitors/browsers. The exercise begins with a student acting in the role of Production Monitor receiving the necessary information/requirements for defining and running monitors/browsers. The student defines and runs monitor/browser as specified in the requirements and responds without error to questions concerning data displayed on the monitor/browser.

Perform the following steps:

1. Access the Monitor/Browser GUI.
2. Define monitor/browser as specified in the requirements.
3. Save the monitor/browser.
4. Run the monitor/browser.
5. Respond without error to questions concerning data displayed on the monitor/browser.

Troubleshooting Processing Problems

The exercise involves troubleshooting production processing problems. The exercise begins with a student acting in the role of Production Monitor receiving the necessary trouble symptom information and requirements for troubleshooting the problem(s). The student reviews the specified trouble symptoms, takes action to correct the problem(s), and responds to questions concerning the possible cause(s).

Perform the following steps:

1. Review the trouble symptoms.
2. Check the status of relevant hosts/servers as necessary.
3. Check log files as necessary.
4. Take action to correct the problem(s).
5. Respond without error to questions concerning the possible cause(s).

Launching the QA Monitor GUI

The exercise involves launching the QA Monitor GUI using UNIX commands. The exercise begins with a student acting in the role of Production Monitor recognizing the need to launch the QA Monitor. The student launches the QA Monitor as specified in the requirements.

Perform the following steps:

1. Log in to the Planning/Management Workstation.
2. Set the necessary environmental variables.
3. Start the QA Monitor GUI in the appropriate mode.

Updating Quality Assurance (QA) Metadata

The exercise involves updating the QA metadata of a science product granule at the request of Science Computing Facility (SCF) personnel. The exercise begins with a student acting in the role of Production Monitor receiving the necessary information/requirements for performing a science product QA metadata update. The student performs science product QA metadata update as specified in the requirements.

Perform the following steps:

1. Set up and query the database using the QA Monitor GUI.
2. Select the granule with QA metadata to be updated.
3. Set the operational and SCF quality flags to the appropriate value (as specified in the requirements).
4. Verify that the flags have actually been set in the database.

Regenerating Granules in Response to Loss of Files from the Archive

The exercise involves regenerating granules in response to a loss of files from the archive. The exercise begins with a student acting in the role of Production Planner receiving the necessary information/requirements for regenerating granules. The student determines which granules to regenerate, creates the necessary Production Request(s), and creates and activates a Production Plan as specified in the requirements.

Perform the following steps:

1. Retrieve the Production History files (PH) for lost granules.
2. Create Production Requests for the generation of replacement granules.
3. Create and activate a Production Plan that includes the Production Requests for the generation of replacement granules.
4. Prepare (if applicable) a “PDPS Residual Granules List.”

This page intentionally left blank.

Slide Presentation

Slide Presentation Description

The following slide presentation represents the slides used by the instructor during the conduct of this lesson.

This page intentionally left blank.