4.4 Data Management Subsystem Overview

The Data Management Subsystem (DMS) provides catalog interoperability between the Version 0 (V0) Information Management System (IMS) and the ECS. The DMS provides this service by supplying gateway processes. There is a dedicated gateway process for each external catalog interoperability direction.

The DMS provides protocol translation via the dedicated gateways. The DMS provides content translation via a data dictionary of data collection information. This data dictionary contains collection level metadata in addition to attribute and keyword mappings between the ECS and the external catalog systems.

The V0 EOS Data Gateway (V0 Web Client) marshals the EMD interface to the V0-IMS. Documentation for the V0 Web Client can be found at the following Universal Resource Locator:

http://delenn.gsfc.nasa.gov/~imswww/pub/manuals/imsdesign.html

Data Management Subsystem Context

Figure 4.4-1 is the Data Management Subsystem context diagrams. The diagrams show the events sent to the Data Management Subsystem and the events the Data Management Subsystem sends to other SDPS and CSMS subsystems. Table 4.4-1 provides descriptions of the interface events shown in the Data Management Subsystem context diagrams.

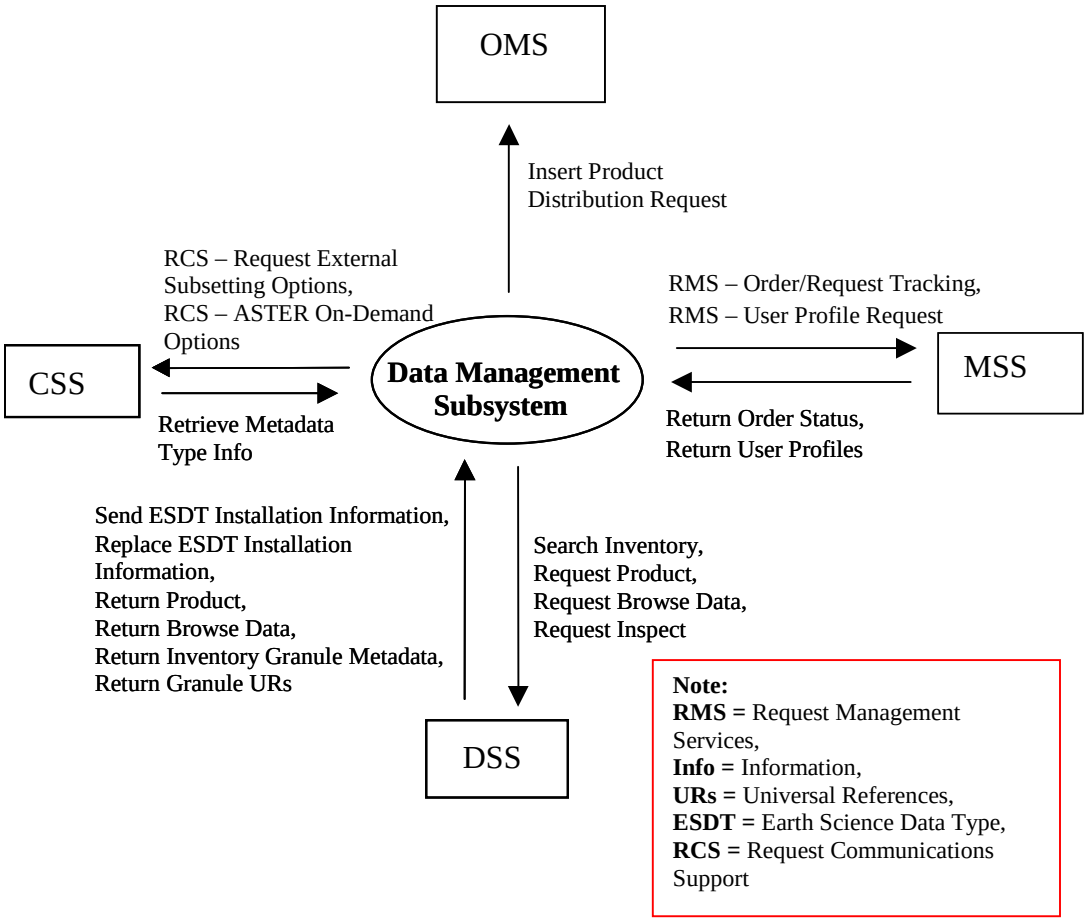


Figure 4.4-1. Data Management Subsystem Context Diagram

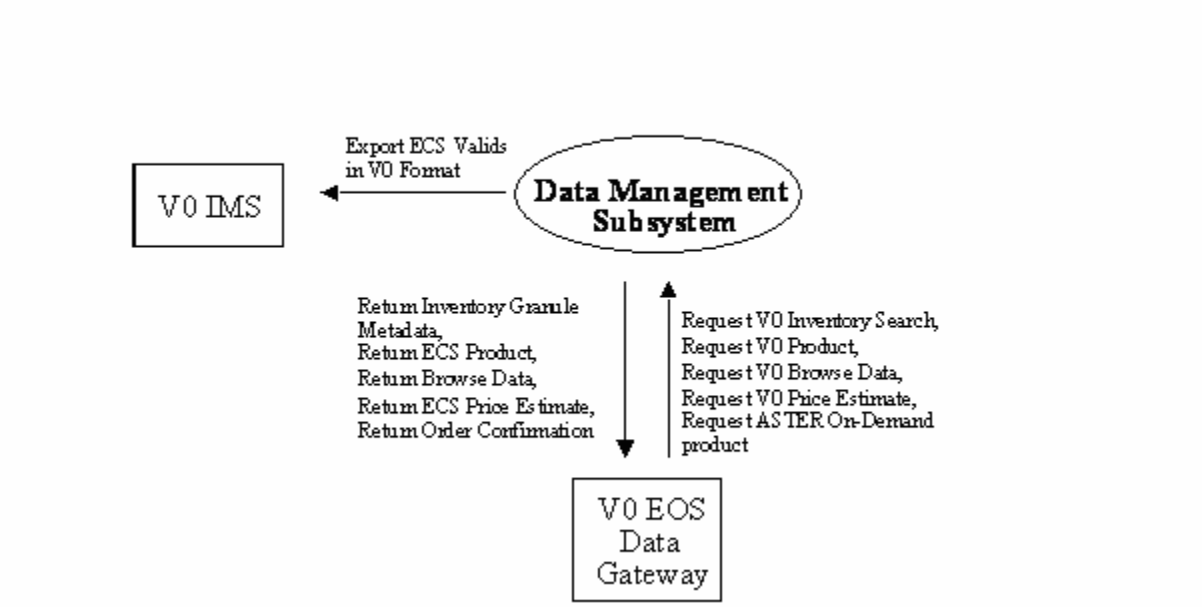


Figure 4.4-1. Data Management Subsystem Context Diagram (cont.)

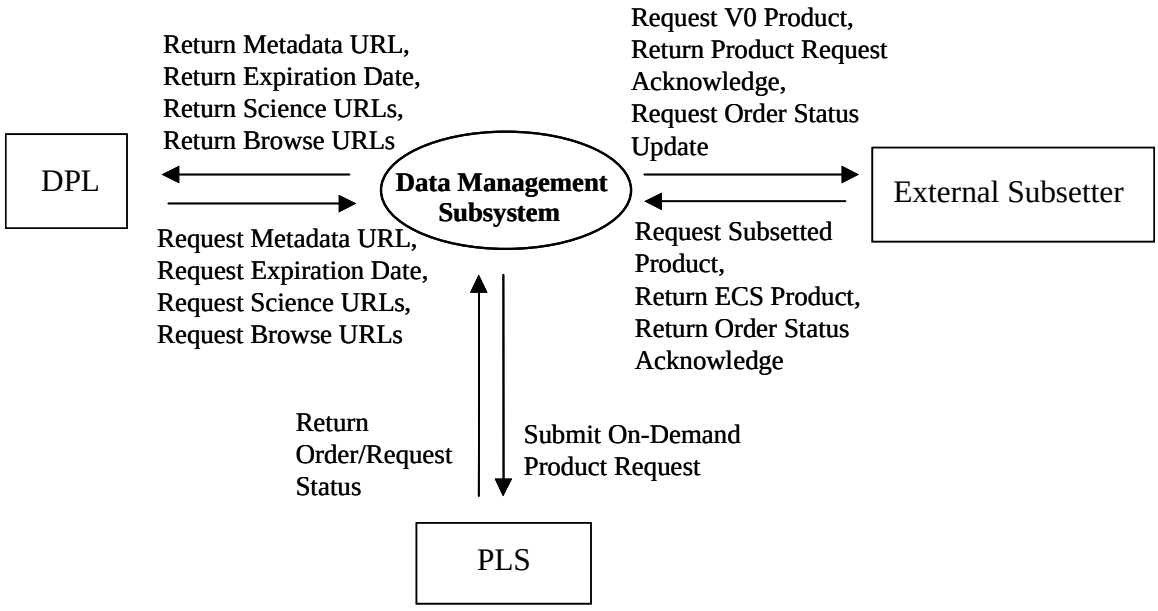


Figure 4.4-1. Data Management Subsystem Context Diagram (cont.)

Table 4.4-1. Data Management Subsystem Interface Events (1 of 3)

Event	Interface Event Description
Insert Product Distribution Request	The Data Management Subsystem (DMS) inserts product distribution requests in the Order Manager Data Base Management System within the **Order Manager Subsystem (OMS)**.
Request Management Services	The **MSS** provides a basic management library of services to the subsystems, implemented as client or server applications, using the CSS Process Framework. The basic management library of services include: • **System startup and shutdown -** Please refer to the release-related, current version of the Mission Operations Procedures for the ECS Project document (611) and the current ECS Project Training Material document (625), identified in Section 2.2.1 of this document. The MSS also interfaces with other subsystems to perform the following: • **Order/Request Tracking** - The DMS interfaces with the **MSS** Order/Request tracking service to create and track a user product order. • **User Profile Request** - The **MSS** provides requesting subsystems with User Profile parameters such as e-mail address and shipping address to support their processing activities.
Return Order Status	The **MSS** provides order ids and order status information for products requested by users.
Return User Profiles	The **MSS** returns user profile information requested by users or ECS processes.
Search Inventory	The DMS submits inventory search requests to the Science Data Server within the **DSS**.
Request Product	The DMS submits the product requests to the **DSS** to acquire data granules.
Request Browse Data	The DMS submits the browse requests to the **DSS** to obtain browse data to determine the type of product to order.
Request Inspect	The DMS sends a request for an inspection of granule metadata to the **DSS** in support of a price estimate request.
Send ESDT Installation Information	The **DSS** inserts new collection level information into the DMS Data Dictionary database as new Earth Science Data Types (ESDTs) are added to the SDPS.
Replace ESDT Installation Information	The **DSS** sends updated ESDT information to the DMS Data Dictionary whenever an ESDT is updated. This data consists of updated Inventory and Collection level metadata.
Return Product	The DMS receives products from the **DSS** based upon a product request.
Return Browse Data	The DMS receives browse data associated with a particular granule from the **DSS**.
Return Inventory Granule Metadata	The DMS receives the granule metadata identifying the scene within the granule based on an inventory search request from the **DSS**.
Return Granule URs	The DMS receives Earth Science Data Type (ESDT) Universal References (URs) for the requested granules from the **DSS**.
Retrieve Metadata Type Info	The **CSS** retrieves type information for qualifying metadata specified in a SIPS search request from the DMS.

Table 4.4-1. Data Management Subsystem Interface Events (2 of 3)

Event	Interface Event Description
Request Communications Support	The **CSS** provides a library of services available to each SDPS and CSMS subsystems. The subsystem services required to perform specific assignments are requested from the CSS. These services include: • CCS Middleware Support • Database Connection Services • Name/Address Services • Password Services • Server Request Framework (SRF) • Universal Reference (UR) • Error/Event Logging • Fault Handling Services • Mode Information • Query Registry - Retrieving the requested configuration attribute-value pairs from the Configuration Registry • Request Distribution Media Options from the Configuration Registry • Request External Subsetting Options from the Configuration Registry • Request ASTER On-Demand Options from the Configuration Registry
Return Status of User Billing Information	The **DORRAN Billing and Accounting System** returns the status of the user billing information to the DMS to allow or deny the user access to ECS data and services.
Return Inventory Granule Metadata	The DMS returns the inventory granule metadata identifying the scene within the granule based on an inventory search request to the user via the **V0 EOS Data Gateway**.
Request V0 Inventory Search	The DMS receives inventory search requests from the **V0 EOS Data Gateway** (**EDG**) on behalf of an external ECS user.
Request V0 Product	The DMS receives product requests from the **V0 EOS Data Gateway** and **External Subsetter** on behalf of an external ECS user.
Request V0 Browse Data	The DMS receives browse data requests from the **V0 EOS Data Gateway** on behalf of an external ECS user.
Request V0 Price Estimate	The user sends a price estimate request for an ECS product to the DMS via the **V0 EOS Data Gateway**.
Request Aster On-Demand Product	The DMS receives product requests for ASTER On-Demand products from the **V0 EOS Data Gateway**.

Table 4.4-1. Data Management Subsystem Interface Events (3 of 3)

Event	Interface Event Description
Return ECS Product	The DMS returns ECS data to the user via the **V0 EOS Data Gateway** based upon a product request.
Return Browse Data	The DMS returns Browse data associated with a particular granule to the user via the **V0 EOS Data Gateway**.
Return ECS Price Estimate	The DMS returns a price estimate for a price estimate request to the user via the **V0 EOS Data Gateway**.
Return Order Confirmation	The **V0 EOS Data Gateway** receives order confirmation of an ASTER On-Demand request from the DMS.
Export ECS Valids in V0 Format	The DMS sends the valids with the ECS core and PSA Attributes to the **V0 IMS**.
Return Product Request Acknowledge	The DMS receives confirmation of a subsetted request from an **External Subsetter**.
Request Order Status Update	The DMS receives order status update requests provided by **External Subsetters** and submits the update to MSS
Request Subsetted Product	The DMS receives product requests for subsetted products and sends requests for the products to be provided by an **External Subsetter**.
Return Order Status Acknowledge	The DMS returns confirmation of an order status update request to the **External Subsetter**.
Submit On-Demand Product Request	The **PLS** receives product requests from the DMS for ASTER On-Demand products.
Return Order/Request Status	The **PLS** returns the product request status to the DMS.
Request Metadata URL	The DMS retrieves Metadata URLs for granules from the **Data Pool Sybase ASE**.
Request Expiration Date	The DMS retrieves Expiration Dates for granules from the **Data Pool Sybase ASE**.
Request Science URLs	The DMS retrieves Science URL(s) for granules from the **Data Pool Sybase ASE**.
Request Browse URLs	The DMS retrieves Browse URL(s) for granules from the **Data Pool Sybase ASE**.
Return Metadata URLs	The **Data Pool Sybase ASE** returns Metadata URLs to the DMS.
Return Expiration date	The **Data Pool Sybase ASE** returns Expiration Dates to the DMS.
Return Science URLs	The **Data Pool Sybase ASE** returns zero or more Science URL(s) to the DMS.
Return Browse URLs	The **Data Pool Sybase ASE** returns zero or more Browse URL(s) to the DMS.

Data Management Subsystem Structure

The DMS is two CSCIs and one Hardware Configuration Item (HWCI):

- The Data Dictionary (DDICT) is a software configuration item. DDICT manages the definitions of data collections including the metadata, data domains (valid values), and data location. The Data Dictionary information is stored persistently in a Relational Database Management System (DBMS).
- The Version 0 Gateway (V0 GTWAY) is a software configuration item. The V0 GTWAY CSCI provides access to data and services between the SDSRV CSCI and the V0 IMS. V0 GTWAY services include inventory searches; requests for browse data, product requests, and price estimate requests.
- The DMS hardware comprises one hardware configuration item Data Management Hardware (DMGHW) CI and one hardware configuration item. DMGHW provides the servers and workstations needed for all data management functions. The DMGHW provides processing and storage for the DDICT and V0 GTWAY CSCIs.

Use of COTS in the Data Management Subsystem

Note: The following RogueWave Libraries are currently delivered with custom code as static libraries. A separate installation of dynamic libraries is no longer required.

- RogueWave's Tools.h++

  The Tools.h++ class libraries are used by the DMS to provide basic functions and objects such as strings and collections.

- RogueWave's DBTools.h++

  The DBTools.h++ C++ class libraries are used to interact with the Sybase database Structured Query Language (SQL) server. The use of DBTools buffers the DMS processes from the relational database used.

Other COTS products are also utilized to support the Data Management Subsystem.

- The ICS Builder Xcessory GUI

  The Builder Xcessory GUI builder tool modifies the displays of the Data Dictionary Maintenance Tool (Mtool). The builder tool also generates the C++ code to produce the Mtool displays at run time. There is no operational component of Builder Xcessory needed at run-time.

- Sybase Adaptive Server Enterprise (ASE)

  Sybase's ASE provides access for the Data Dictionary to insert, update, and delete Data Dictionary database information. The Sybase ASE must be running during operations for the Data Dictionary Server to execute, search, and update requests on the Data Dictionary database.

- CCS Middleware Client

  CCS Middleware Client provides DMS with communications between other subsystems. CCS Middleware can reside on one or both sides of the interface. An instance must be installed on the platform where DMS resides. Although the CCS Middleware Client is part of CSS, this COTS software item must be installed for DMS to run in the SDPS operational and test environment.

Error Handling and processing

EcUtStatus is a class used throughout the ECS custom code for general error reporting. It is almost always used as a return value for functions and allows detailed error codes to be passed back up function stacks.

The DDICT and V0 GTWAY CSCIs use two main mechanisms for error handling.

1. Return EcUtStatus values

   Functions can return an EcUtStatus object, indicating success, failure or other detailed status, which corresponds to EcUtStatus:OK, EcUtStatus:FAILED and EcUtStatus:DETAILED.

   For V0ToEcsGateway, the EcUtStatus values are reflected on the EDG to indicate the search or order status.

2. Exceptions

   Some functions (e.g., a main function) cannot return EcUtStatus values to indicate success or failure. These functions can catch some exceptions after a try block.

All error messages are sent to the ALOG file or debug log file.

For writing messages to the Applications Log (ALOG), the following functions are used:

EcLgLogError sends a message to the ALOG at severity level 1. For example, EcLgLogError("Error" , 0,"Failed to get Order ID from MSS. TransactionID=%s.",

myTransactionId.data());

EcLgLogInformational sends a message to the ALOG at severity level 3. For example, EcLgLogInformational ("DmGwManagedServer", 0, "Server Shutdown");

For writing messages to the debug log, the following macros are used:

PF_STATUS writes a message at a "log level" of 1 to the debug log. For example, PF_STATUS cerr << "Unable to create inputNonAggregate " << endl;

PF_VERBOSE writes a message at a "log level" of 2 to the debug log. For example, PF_VERBOSE cerr << "~~~ Connect to Science Data Server FAILED."

PF_DEBUG writes a message at a "log level" of 3 to the debug log. For example, PF_DEBUG cerr << "~~~ Failed to get Order ID from MSS." << endl;

4.4.1 Data Dictionary Software Description

4.4.1.1 Data Dictionary Functional Overview

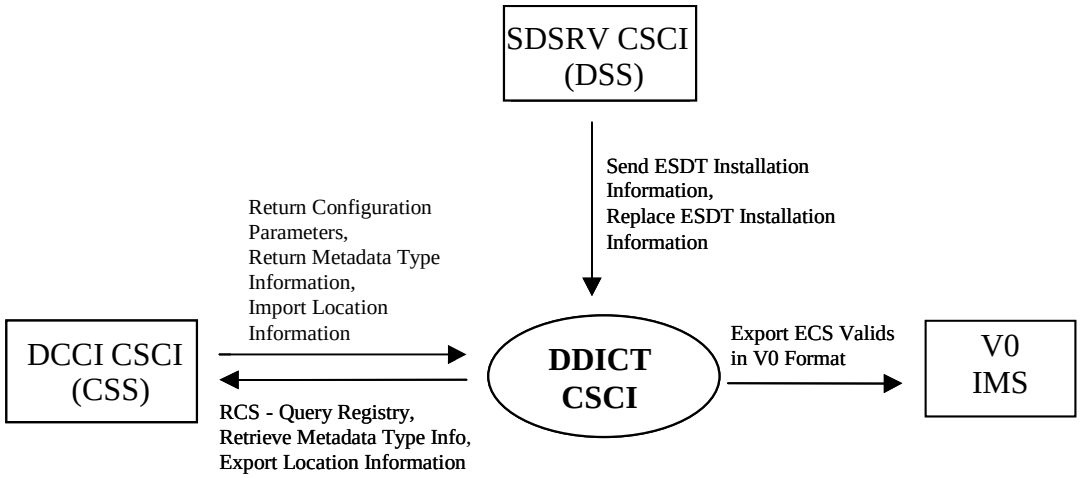
The Data Dictionary (DDICT) CSCI provides access to the Data Dictionary database containing information about science data collections, data attributes, data operations, and the domain(s) of the attributes. The DDICT CSCI describes the data objects accessible through Data Servers and the Gateways. The DDICT CSCI provides information support for users to retrieve definitions of the available items and provides infrastructure support to the other CSCIs within the DMS.

The Information contained within the Data Dictionary database includes all collections known within the SDPS. Clients (other SDPS or CSMS CSCIs, CSCs, or processes) of the Data Dictionary obtain data collection information by sending queries to the Data Dictionary. Mappings between the SDPS and the V0 IMS attributes and keywords are also maintained within the Data Dictionary. These mappings are used to translate requests between the V0 IMS and the ECS.

The location of a data collection within a data server at a particular site is also stored within the data dictionary. This information allows users to perform queries through user software such as the V0 EOS Data Gateway from any geographical location to forward inventory search, browse, and acquire requests to the appropriate Data Server or Gateway located at the site where the data is physically stored.

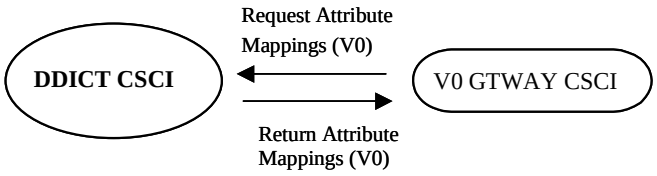
4.4.1.2 Data Dictionary Context

Figure 4.4-2 is the DDICT CSCI context diagrams. The diagrams show the events sent to other CSCIs or CSCs and the events the DDICT CSCI receives from other CSCIs and CSCs. Table 4.4-2 provides descriptions of the interface events shown in the DDICT CSCI context diagrams.



Note:
ESDT = Earth Science Data Type,
GDS = Ground Data System,
IMS = Information Management System,
Info = Information,
RCS = Request Communications Support

Figure 4.4-2. Data Dictionary CSCI Context Diagram



Note:
V0 = Version 0,
DAAC = Distributed Active Archive Center

Figure 4.4-2. Data Dictionary CSCI Context Diagram (cont.)

Table 4.4-2. Data Dictionary CSCI Interface Events

Event	Interface Event Description
Request Management Services	**System startup and shutdown –** Please refer to the release-related, current version of the Mission Operations Procedures for the ECS Project document (611) and the current ECS Project Training Material document (625), identified in Section 2.2.1 of this document.
Send ESDT Installation Information	The **SDSRV CSCI** inserts new collection level information into the DMS Data Dictionary database via the DDICT CSCI, as new Earth Science Data Types (ESDTs) are added to the SDPS.
Replace ESDT Installation Information	The **SDSRV CSCI** sends updated ESDT information to the DDICT CSCI whenever an ESDT is updated. This data consists of Inventory and Collection level metadata. The updated information replaces the ESDT information in the DDICT CSCI.
Request Communications Support	The **DCCI CSCI** provides a library of services available to each SDPS and CSMS CSCI. The CSCI services required to perform specific assignments are requested from the DCCI CSCI. These services include: • CCS Middleware Support • Database Connection Services • Name/Address Services • Password Services • Server Request Framework (SRF) • Universal Reference (UR) • Error/Event Logging • Fault Handling Services • Mode Information • Query Registry – Retrieving the requested configuration attribute-value pairs from the Configuration Registry
Retrieve Metadata Type Info	The **DCCI CSCI** retrieves metadata type information from the DDICT CSCI pertaining to search or search & order requests from the SIPS.
Export Location Information	The DDICT CSCI stores physical and logical information in the **CCS NameServer**.
Return Configuration Parameters	The DDICT CSCI receives the configuration parameters and associated values from the Registry Server within the **DCCI CSCI**.
Return Metadata Type Info	The **DCCI CSCI** receives metadata type information from the **DDICT CSCI**.
Import Location Information	The DDICT CSCI requests server location information from the **CCS NameServer**.
Request Attribute Mappings (V0)	A user, via the EOS Data Gateway within the V0 IMS, requests data collection attribute and keyword mappings (via the **V0 GTWAY CSCI**) from the DDICT CSCI to translate requests from the V0 IMS to the ECS and back again.
Return Attribute Mappings (V0)	The DDICT CSCI returns the data collection attribute and keyword mappings requested by the user via the **V0 GTWAY CSCI**.

4.4.1.3 Data Dictionary Architecture

Figure 4.4-3 is the DDICT CSCI architecture diagram. The diagram shows the events sent to the DDICT CSCI processes and the events the DDICT CSCI processes send to other processes.

The DDICT CSCI is two SDPS processes, the Data Dictionary Server (EcDmDictServer) and the Data Dictionary Maintenance Tool (EcDmMaintenanceTool) and a COTS process (the Sybase ASE). The Data Dictionary Server, Maintenance Tool, and Sybase ASE processes reside inside a DAAC and run on the DMGHW. The Data Dictionary uses one data store per DAAC, the EcDmDictServer Database, as shown in Figure 4.4-3.

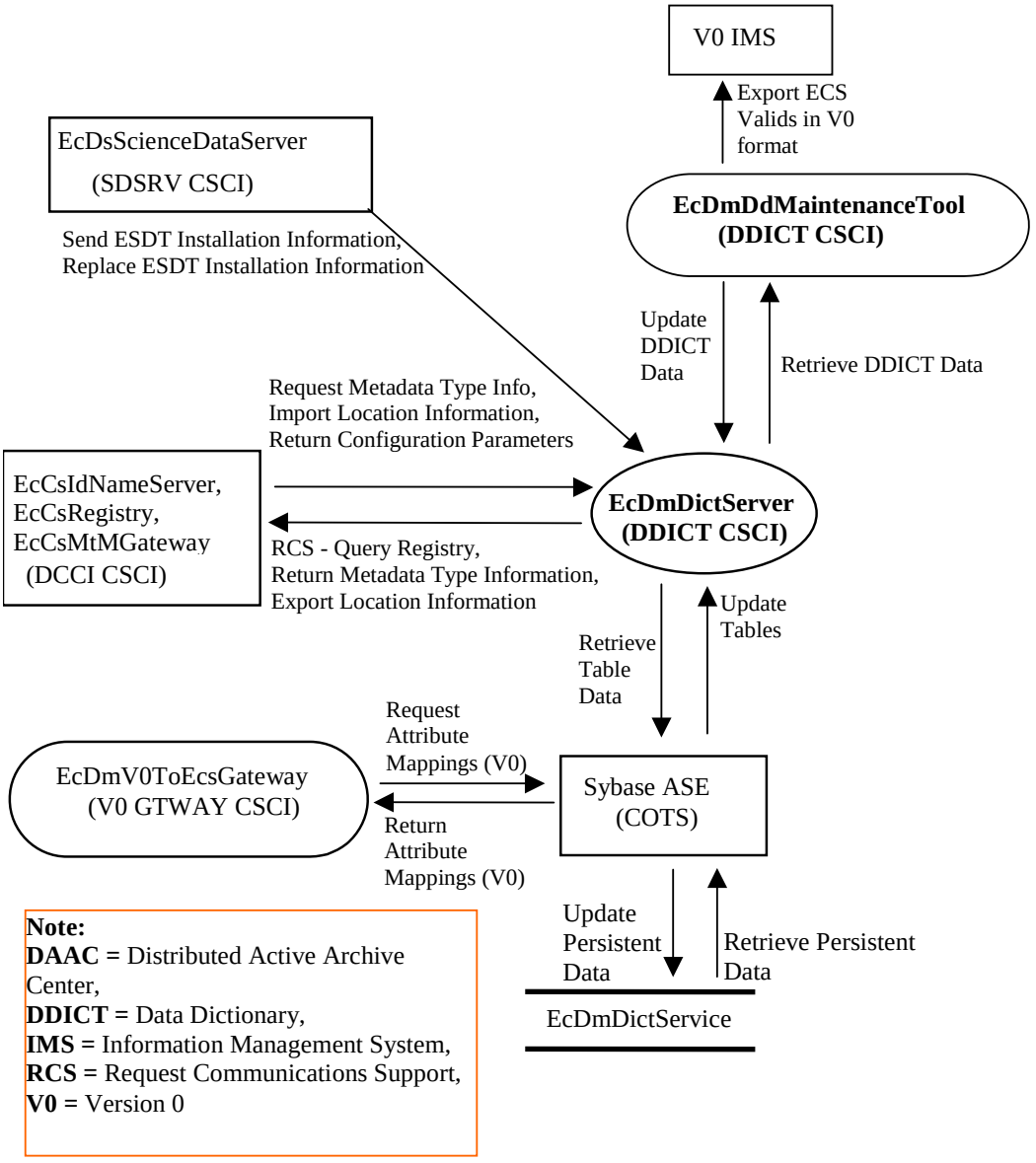


Figure 4.4-3. Data Dictionary CSCI Architecture Diagram

4.4.1.4 Data Dictionary Process Descriptions

Table 4.4-3 provides descriptions of the processes shown in the Data Dictionary CSCI architecture diagram.

Table 4.4-3. Data Dictionary CSCI Processes

Process	Type	Hardware CI	COTS/ Developed	Functionality
EcDmDictServer	Server	INTHW	Developed	The Data Dictionary Service is the primary server interface to collection and collection related information for the DMS and other subsystems. It allows DDICT client processes the capability to perform data searches, insertions, updates, or deletions to the collection information held in the DDICT database. The Data Dictionary offers two basic interfaces DDICT Data Search: The Data Dictionary Server allows a user to specify search requests on the Data Dictionary database using a GlParameter List. DDICT Data Insert and Delete: Provides a client process with the capability to insert and delete data within the Data Dictionary database. The Data Dictionary Service supports: Single requests at a time Synchronous request processing Asynchronous request processing
EcDmDdMaintenanceTool	GUI	INTHW	Developed	Provides a graphical user interface (GUI) to insert, update, or delete schema information held in the DDICT database, allowing DAAC operations staff to maintain the data stored in the Data Dictionary database. The Data Dictionary Maintenance Tool also provides the following capabilities: Import and Export of Valids: The tool allows DAAC operations staff to import and export data collection attribute valids to and from ECS, and V0 IMS for catalog interoperability. Data Collection Attribute and Keyword Mapping: Allows DAAC operations staff to map data collection attributes and keyword valids from ECS to V0 IMS. The V0 GTWAY CSCI process (EcDmV0ToEcsGateway) that translates requests from ECS to V0 IMS uses this information.
Sybase ASE	Server	ACMHW	COTS	The Sybase ASE acts as a SQL server for the Data Dictionary, and is only run at the DAACs by DAAC operations staff. Refer to Sybase documentation for details.

EMD Baseline Information System (EBIS) Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the HWCI, and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

4.4.1.5 Data Dictionary Process Interface Descriptions

Table 4.4-4 provides descriptions of the interface events shown in the Data Dictionary CSCI architecture diagram.

Table 4.4-4. Data Dictionary CSCI Process Interface Events (1 of 5)

Event	Event Frequency	Interface	Initiated By	Event Description
Request Management Services	At system startup or shutdown and for restarts	*Process:* EcDmDictServer	DAAC unique startup scripts	**System startup and shutdown -** Please refer to the release-related, current version of the Mission Operations Procedures for the ECS Project document (611) and the current ECS Project Training Material document (625), identified in Section 2.2.1 of this document.
Export ECS Valids in V0 Format	Based on V0 valids processing cycle ~ once per week	V0 IMS Host	DAAC Operations Staff *Process:* EcDmDdMaintenanceTool *Classes:* DmLmDbiUtils, DmDdMtMainWindow	The DAAC Operations Staff pushes the valids file, using the FTP service, manually to its destination location agreed upon earlier on a **V0 IMS** Host.
Retrieve DDICT data	One per Maintenance Tool search	*Process*: EcDmDdMaintenance Tool *Classes:* DmDdMtDBExtract, DmDdMtMainWindow	*Process:* EcDmDictServer *Library*: EcDmDdClient *Class*: DmDdClRequest	The EcDmDictServer returns data collection information including collection lists, and collection attributes and keyword valids to the **EcDmDdMaintenanceTool**.
Update DDICT data	One per table information update	*Process:* EcDmDictServer *Library*: EcDmDdClient *Class*: DmDdClSchemaRequest	Operations Staff *Process*: EcDmDdMaintenanceTool *Classes:* DmLmDbiUtils, DmDdMtMainWindow	The Operations Staff, using the **EcDmDdMaintenanceTool**, updates table information within the DDICT database (via the EcDmDictServer) including mapping collection attributes to keywords and mapping collections to information managers and adding, modifying, and deleting SDPS Core Attributes.

Table 4.4-4. Data Dictionary CSCI Process Interface Events (2 of 5)

Event	Event Frequency	Interface	Initiated By	Event Description
Update Tables	One per database update	*Process:* Sybase ASE (COTS SW) RWDBTools.h++ classes	*Process:* EcDmDictServer *Library*: DmDdReqProc *Classes*: DmDdMapper, DmDdProcMsg	The EcDmDictServer receives updated data from the Data Dictionary database once the **Sybase ASE** inserts and deletes collections and collection metadata, attributes and keywords, and attribute and keyword mappings.
Retrieve Persistent Data	One per set number of queries	*Process:* Sybase ASE (COTS)	*Data Base Table:* EcDmDictService	The **Sybase ASE** retrieves data persistently stored on disk(s) based on search queries from the EcDmDictServer.
Update Persistent Data	One per set number of queries	*Data Base Table:* EcDmDictService	*Process:* Sybase ASE (COTS)	The **Sybase ASE** updates data persistently stored on disk(s) based on queries from the EcDmDictServer.
Request Attribute Mappings (V0)	One per request from V0 Gateway	*Process:* EcDmDictServer *COTS SW Library:* RWDBTools.h++ *Library:* DmLmDbi *Class:* DmLmIntQuery	*Process:* EcDmV0ToEcsGateway *Library:* Persistent *Class*: DmGwAttributeMap	The **EcDmV0ToEcsGateway** requests data collection attribute and keyword mappings from the **EcDmDictServer** Data Dictionary database, via the Sybase ASE, to translate requests from the V0 IMS to the SDPS and back again.
Return Attribute Mappings (V0)	One per request	*Process:* EcDmV0ToEcsGateway *Library:* Persistent *Class*: DmGwAttributeMap	*Process:* EcDmDictServer *COTS SW Library:* RWDBTools.h++ *Library:* DmLmDbi *Class:* DmLmIntQuery	The **EcDmDictServer** Data Dictionary database, via the Sybase ASE, returns the data collection attribute and keyword mappings requested by the user to the **EcDmV0ToEcsGateway**.
Retrieve Table Data	One per retrieve from the database	*Process:* Sybase ASE (COTS SW) RWDBTools.h++ classes	*Process:* EcDmDictServer *Library*: DmDdServerJit *Class*: DmDdSearchRequest	The EcDmDictServer requests data (such as collections and collection metadata, attributes and keywords, and attribute and keyword mappings) from the Data Dictionary database within the **Sybase ASE**.

Table 4.4-4. Data Dictionary CSCI Process Interface Events (3 of 5)

Event	Event Frequency	Interface	Initiated By	Event Description
Request Communications Support	Request service(s) as required	***Process:*** EcCsIdNameServer ***Libraries:*** EcPf, Middleware, FoNs, FoIp, oodce *Classes:* EcPfManagedServer, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy ***Library (Common):*** EcUr *Class:* EcUrServerUR *Library:* event *Class:* EcLgErrorMsg ***Process:*** EcCsRegistry *Library:* EcCsRegistry *Class:* EcRgRegistryServer_C	*Process:* EcDmDictServer *Libraries:* EcDmDdClient, DmDdReqProc, DmDdServerJit *Classes:* DmDdClSchemaRequest, DmDdClRequest, DmDdMapper, DmDdProcMsg, DmDdSearchRequest	The **DCCI CSCI** provides a library of services available to each SDPS and CSMS process. The process services required to perform specific assignments are requested from the DCCI CSCI CSCs. These services include: • CCS Middleware Support • Database Connection Services • Name/Address Services • Password Services • Server Request Framework (SRF) • Universal Reference (UR) • Error/Event Logging • Fault Handling Services • Mode Information • Query Registry - Retrieving the requested configuration attribute-value pairs from the Configuration Registry

Table 4.4-4. Data Dictionary CSCI Process Interface Events (4 of 5)

Event	Event Frequency	Interface	Initiated By	Event Description
Return Metadata Type Info	Per search or order request	*Process:* EcCsMtMGateway *Class:* EcCsMtMAttributeDict	*Processes:* EcCsMtMGateway, EcDmDictServer *Libraries:* DmAsGwCommon, Common, DmGwV0Util, EcDmDdClient, DmDdMsg *Class:* DmDdClRequestServer	The EcCsMtMGateway receives metadata type information from the **EcDmDictServer**.
Export Location Information	Once at system startup and after each failure recovery	*Process:* EcCsIdNameServer *Libraries:* EcPf, Middleware, FoNs, FoIp, oodce *Classes:* EcPfManagedServer, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy	*Process:* EcDmDictServer *Library:* EcDmDdClient *Class:* DmDdClRequest	The EcDmDictServer stores physical and logical location information in the **EcCsIdNameServer**.
Request Metadata Type Info	One or more client request	*Processes:* EcCsMtMGateway, EcDmDictServer *Libraries:* DmAsGwCommon, Common, DmGwV0Util, EcDmDdClient, DmDdMsg *Class:* DmDdClRequestServer	*Process:* EcCsMtMGateway *Class:* EcCsMtMAttributeDict	The EcCsMtMGateway requests metadata type information from the **EcDmDictServer** based upon the qualifying metadata contained in the requests sent by the SIPS via CCS Middleware calls.

Table 4.4-4. Data Dictionary CSCI Process Interface Events (5 of 5)

Event	Event Frequency	Interface	Initiated By	Event Description
Import Location Information	As required for processing	*Process:* EcDmDictServer *Libraries:* EcDmDdClient, DmDdReqProc, DmDdServerJit *Classes:* DmDdClSchemaRequest, DmDdClRequest, DmDdMapper, DmDdProcMsg, DmDdSearchRequest	*Process:* EcCsIdNameServer *Libraries:* EcPf, Middleware, FoNs, FoIp, oodce *Classes:* EcPfManagedServer, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy	The EcDmDictServer requests server location information from the **EcCsIdNameServer**.
Return Configuration Parameters	One set per request	*Process:* EcDmDictServer *Library:* DmDdServerLib *Class:* DmDdManagedServer	*Process:* EcCsRegistry *Library:* EcCsRegistry *Class:* EcRgRegistryServer_C	The **EcCsRegistry** returns the attribute-value pairs (configuration parameters) to the EcDmDictServer upon request.
Send ESDT Installation Information	One per new ESDT added to ECS	*Process:* EcDmDictServer *Library*: EcDmDdClient *Class*: DmDdClSchemaRequest	*Process:* EcDsScienceDataServer *Library*: DsDe1 *Class*: DsDeDataDictController	The **EcDsScienceDataServer** inserts new collection information into the DMS Data Dictionary database when new Earth Science Data Types (ESDTs) are added to the SDPS.
Replace ESDT Installation Information	One per ESDT update	*Process:* EcDmDictServer *Library:* EcDmDdClient *Class:* DmDdClSchemaRequest	*Process:* EcDsScienceDataServer *Class:* DsDeDataDictController	The **EcDsScienceDataServer** sends updated ESDT information to the EcDmDictServer whenever an ESDT is updated. This data consists of updated Inventory and Collection level metadata.

4.4.1.6 Data Dictionary CSCI Data Stores

Table 4.4-5 provides descriptions of the data stores shown in the Data Dictionary CSCI architecture diagram.

Table 4.4-5. Data Dictionary CSCI Data Stores

Data Store	Type	Functionality
EcDmDictService	Database	The Data Dictionary database, EcDmDictService is a Sybase relational database that persistently stores the collection and collection related information on a physical disk medium. Data stores in the Data Dictionary database include: • Collection Types: A list of all the data types within the ECS. • Collection Metadata: Various types of collection metadata including instrument, platform, sensor, topic, keyword, temporal and spatial data. • Collection Attributes and Keywords: Attributes and keywords associated with collections originating within and outside the ECS. • Collection Attribute and Keyword Mappings: Associations between the V0 IMS attributes and valid keywords and the ECS attributes and keywords are maintained. • Collection to Information Manager Mappings: Associations between the information manager and the ECS collections stored within them are maintained.

4.4.2 V0 Gateway Software Description

4.4.2.1 V0 Gateway Functional Overview

The Version 0 Gateway (V0 GTWAY) CSCI provides interoperability with the V0 Information Management System for inventory searches, browse requests, product orders, and price estimate requests.

The V0 GTWAY CSCI is one process, the V0 to ECS Gateway server. Queries are passed between the V0 IMS and the V0 Gateway processes using the Object Description Language (ODL) format. The structure of the V0 ODL messages is documented in "Interface Control Definition for the EOS Data Gateway (EDG) Messages and Development Data Dictionary" (423-42-06).

Since the V0 IMS uses different attributes to describe data collections within its data archive, the V0 GTWAY CSCI translates those attributes as defined in the SDPS. To perform the translation, the V0 Gateway uses the data collection attribute and valid keyword mapping information contained within the Data Dictionary database to translate the V0 attributes into equivalent SDPS attributes.

4.4.2.2 V0 Gateway Context

Figure 4.4-4 is the V0 GTWAY CSCI context diagrams. The diagrams show the events sent to the V0 GTWAY CSCI and the events the V0 GTWAY CSCI sends to other CSCIs. Table 4.4-6 provides descriptions of the interface events shown in the V0 GTWAY CSCI context diagrams.

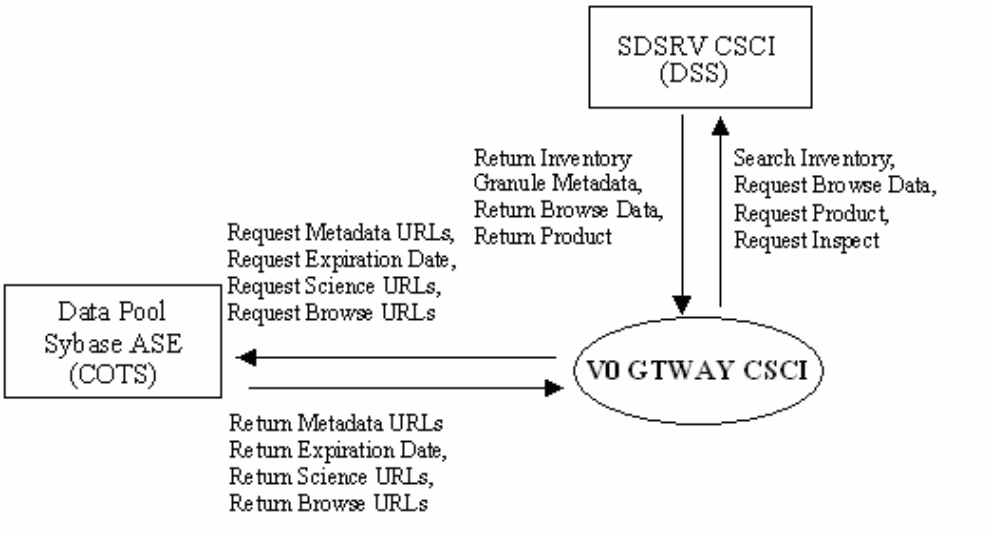


Figure 4.4-4. V0 GTWAY CSCI Context Diagram

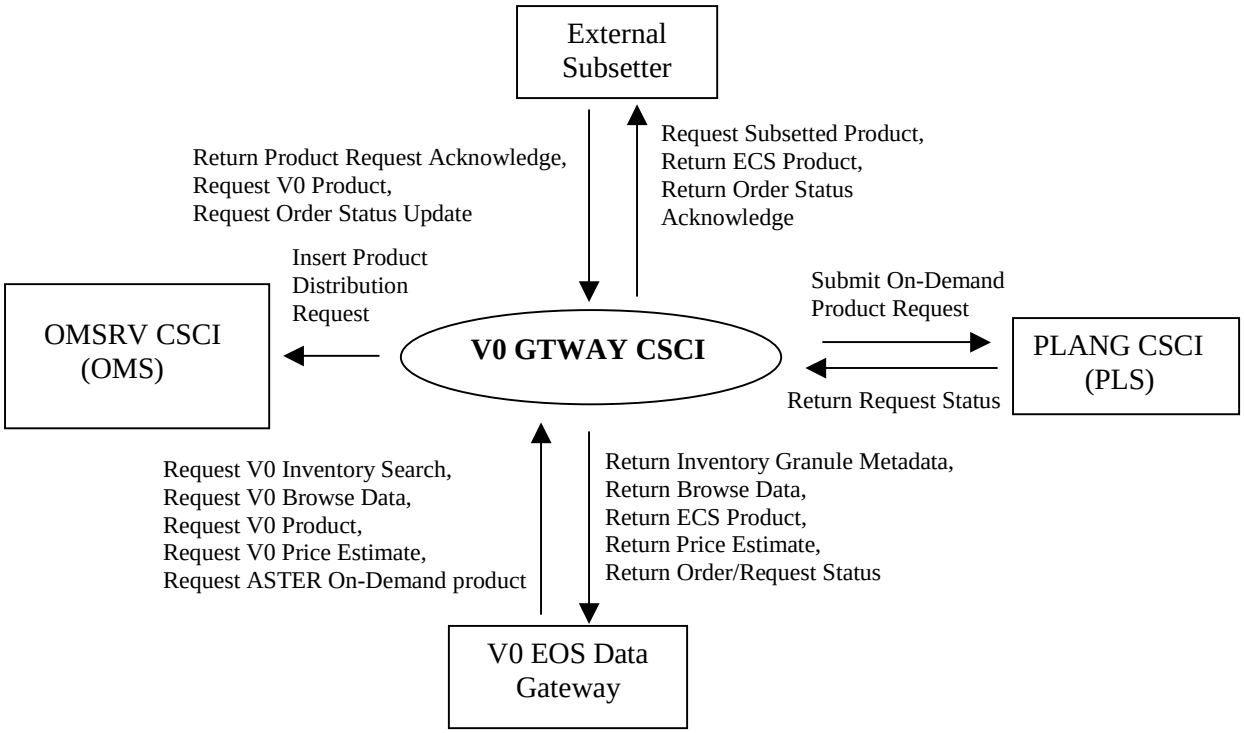


Figure 4.4-4. V0 GTWAY CSCI Context Diagram (cont.)

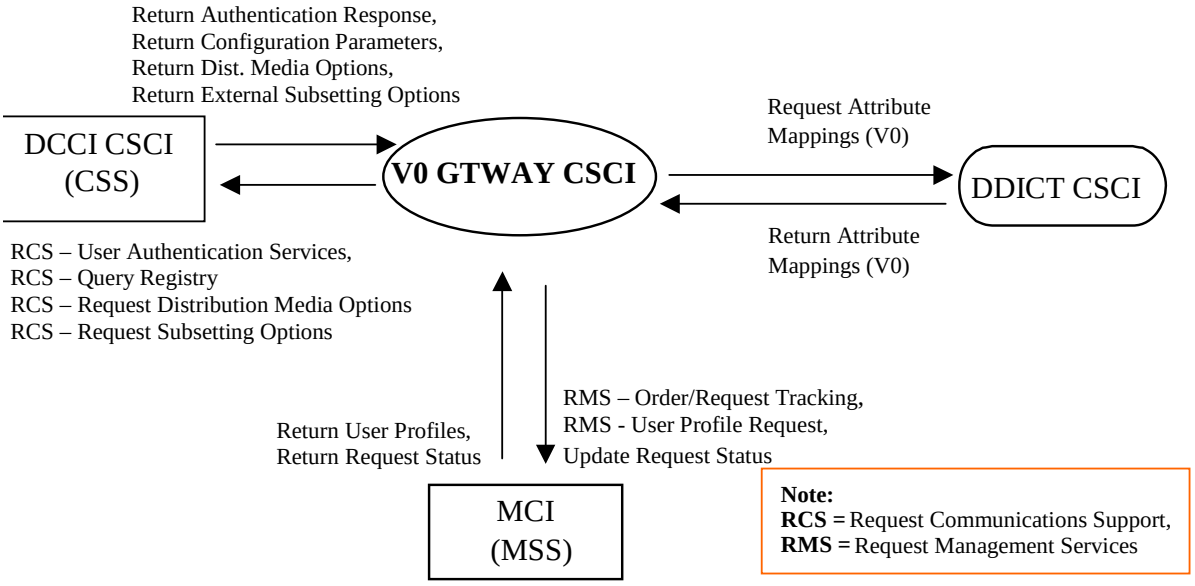


Figure 4.4-4. V0 GTWAY CSCI Context Diagram (cont.)

Table 4.4-6. V0 GTWAY CSCI Interface Events (1 of 4)

Event	Interface Event Description
Search Inventory	The V0 GTWAY CSCI submits inventory search requests to the **SDSRV CSCI** within the DSS on behalf of a user.
Request Browse Data	The V0 GTWAY CSCI receives browse requests from the V0 IMS via the EOS Data Gateway and submits the browse requests to the **SDSRV CSCI** within the DSS on behalf of a user.
Request Product	The V0 GTWAY CSCI submits product requests to the **SDSRV CSCI** within the DSS on behalf of a user.
Request Inspect	The V0 GTWAY CSCI sends a request for an inspection of granule metadata to the **SDSRV CSCI** in support of a price estimate request.
Return Status of User Billing Information	The **EDC DORRAN (Billing and Accounting System)** returns the status of the user billing information to the V0 GTWAY CSCI to allow or deny the user access to ECS data and services.
Return Metadata URLs	The **Data Pool Sybase ASE** returns Metadata URLs to the V0 GTWAY CSCI.
Return Expiration Date	The **Data Pool Sybase ASE** returns an Expiration Date to the V0 GTWAY CSCI.
Return Science URLs	The **Data Pool Sybase ASE** returns zero or more Science URL(s) to the V0 GTWAY CSCI.
Return Browse URLs	The **Data Pool Sybase ASE** returns zero or more Browse URL(s) to the V0 GTWAY CSCI.
Request Metadata URLs	The V0 GTWAY CSCI attempts to retrieve Metadata URLs from the **Data Pool Sybase ASE**.
Request Expiration Date	The V0 GTWAY CSCI attempts to retrieve Expiration Dates from the **Data Pool Sybase ASE**.
Request Science URLs	The V0 GTWAY CSCI attempts to retrieve Science URLs from the **Data Pool Sybase ASE**.
Request Browse URLs	The V0 GTWAY CSCI attempts to retrieve Browse URLs from the **Data Pool Sybase ASE**.
Return Inventory Granule Metadata	The V0 GTWAY CSCI receives the inventory granule metadata identifying the scene within the granule based on an inventory search request from the **SDSRV CSCI**.
Return Granule URs	The V0 GTWAY CSCI receives Earth Science Data Type (ESDT) Universal References (URs) for the granules from the **SDSRV CSCI**.
Return Browse Data	The V0 GTWAY CSCI receives browse data associated with a particular granule from the **SDSRV CSCI**.
Return Product	The V0 GTWAY CSCI receives granules from the **SDSRV CSCI** based upon a product request.
Request Subsetted Product	The V0 GTWAY CSCI receives product requests for subsetted products and requests the product be provided from an **External Subsetter**.
Return ECS Product	The V0 GTWAY CSCI returns granules (including Landsat 7 data archived within the ECS) to the **V0 EOS Data Gateway (EDG)**.
Return Order Status Acknowledge	The V0 GTWAY CSCI returns status of the order status update to an **External Subsetter**.
Submit On-Demand Product Request	The **PLANG CSCI** receives product requests from the V0 GTWAY CSCI for ASTER On-Demand products.

Table 4.4-6. V0 GTWAY CSCI Interface Events (2 of 4)

Event	Interface Event Description
Return Request Status	The **PLANG CSCI** returns the status of the product request to the V0 GTWAY CSCI. The **MCI** returns the status of the update request to the V0 GTWAY CSCI.
Return Inventory Granule Metadata	The V0 GTWAY CSCI returns the inventory granule metadata identifying the scene within the granule to the **V0 EOS Data Gateway (EDG)**.
Return Browse Data	The V0 GTWAY CSCI returns the browse data associated with a particular granule to the **V0 EOS Data Gateway**.
Return ECS Product	The V0 GTWAY CSCI returns granules (including Landsat 7 data archived within the ECS) to the **V0 EOS Data Gateway**.
Return Price Estimate	The V0 GTWAY CSCI returns a price estimate for a price estimate request to the user via the **V0 EOS Data Gateway**.
Return Order/ Request Status	The **V0 EOS Data Gateway** receives order and request status for ASTER On-Demand products from the V0 GTWAY CSCI.
Request V0 Inventory Search	The V0 GTWAY CSCI receives inventory search requests from the **V0 EOS Data Gateway** on behalf of an external ECS user.
Request V0 Browse Data	The V0 GTWAY CSCI receives browse data requests from the **V0 EOS Data Gateway** on behalf of an external ECS user.
Request V0 Product	The V0 GTWAY CSCI receives product requests from the **V0 EOS Data Gateway** or an **External Subsetter** on behalf of an external ECS user.
Request V0 Price Estimate	The V0 GTWAY CSCI receives price estimate requests from the **V0 EOS Data Gateway** on behalf of an external ECS user.
Request ASTER On-Demand product	The V0 GTWAY CSCI receives product requests for ASTER On-Demand products from the **V0 EOS Data Gateway**.
Insert Product Distribution Request	The V0 Gateway (V0 GTWAY) inserts product distribution requests into the Order Manager Data Base Management System (DBMS) within the **Order Manager Subsystem (OMS)**.
Return Product Request Acknowledge	The V0 GTWAY CSCI receives confirmation of a subsetted request from an **External Subsetter**.
Request Order Status Update	The V0 GTWAY CSCI receives order status update requests provided by an **External Subsetter**.
Request Attribute Mappings (V0)	The V0 GTWAY CSCI sends requests for data collection attribute and keyword mappings, on behalf of a user, to the **DDICT CSCI** Data Dictionary database via the Sybase ASE to translate requests from the V0 IMS to the ECS protocol and back again.
Return Attribute Mappings (V0)	The **DDICT CSCI** Data Dictionary database, via the Sybase ASE, returns the data collection attribute and keyword mappings to the V0 GTWY CSCI on behalf of a user.

Table 4.4-6. V0 GTWAY CSCI Interface Events (3 of 4)

Event	Interface Event Description
Request Management Services	The **MCI** provides a basic management library of services to the CSCIs, implemented as client or server applications, using the DCCI CSCI Process Framework. The basic management library of services includes: • **System startup and shutdown** - Please refer to the release-related, current version of the Mission Operations Procedures for the ECS Project document (611) and the current ECS Project Training Material document (625), identified in Section 2.2.1 of this document. The MCI also interfaces with other CSCIs to perform the following: • **Order/Request Tracking** - The V0 GTWAY CSCI interfaces with the **MCI** Order/Request tracking service (EcMsAcOrderSrvr) to create a user product order. • **User Profile Request** - The **MCI** provides requesting CSCIs with User Profile parameters such as e-mail address and shipping address to support their processing activities.
Update Request Status	The V0 GTWAY CSCI receives order status update requests provided by an **External Subsetter** and submits the update to the MSS.
Return User Profiles	The V0 GTWAY CSCI receives user profile information from the **MCI** to authenticate a user.
Request Communications Support	The **DCCI CSCI** provides a library of services available to each SDPS and CSMS CSCI. The CSCI services required to perform specific assignments are requested from the DCCI CSCI. These services include: • CCS Middleware Support • Database Connection Services • File Transfer Services • Network & Distributed File Services • Bulk Data Transfer Services • Name/Address Services • Password Services • Server Request Framework (SRF) • Universal Reference (UR) • Error/Event Logging • Fault Handling Services • Mode Information • Query Registry - Retrieving the requested configuration attribute-value pairs from the Configuration Registry • Request Distribution Media Options from the Configuration Registry • Request External Subsetting Options from the Configuration Registry • Request ASTER On-Demand Options from the Configuration Registry
Return Authentication Response	The V0 GTWAY CSCI receives the response to authenticate the user from the **DCCI CSCI**.

Table 4.4-6. V0 GTWAY CSCI Interface Events (4 of 4)

Event	Interface Event Description
Return Configuration Parameters	The V0 Gateway CSCI receives the configuration parameters and associated values from the Registry Server within the **DCCI CSCI**.
Return Dist. Media Options	The V0 GTWAY CSCI receives the requested distribution media options from the **DCCI CSCI**.
Return External Subsetting Options	The V0 GTWAY CSCI receives the requested external subsetting options from the **DCCI CSCI**.

4.4.2.3 V0 Gateway Architecture

Figure 4.4-5 is the V0 GTWAY CSCI architecture diagram. The diagram shows the events sent to the V0 GTWAY CSCI processes and the events the V0 GTWAY CSCI processes send to other processes.

The V0 GTWAY CSCI is one process, the EcDmV0ToEcsGateway, as shown in the V0 GTWAY CSCI architecture diagram.

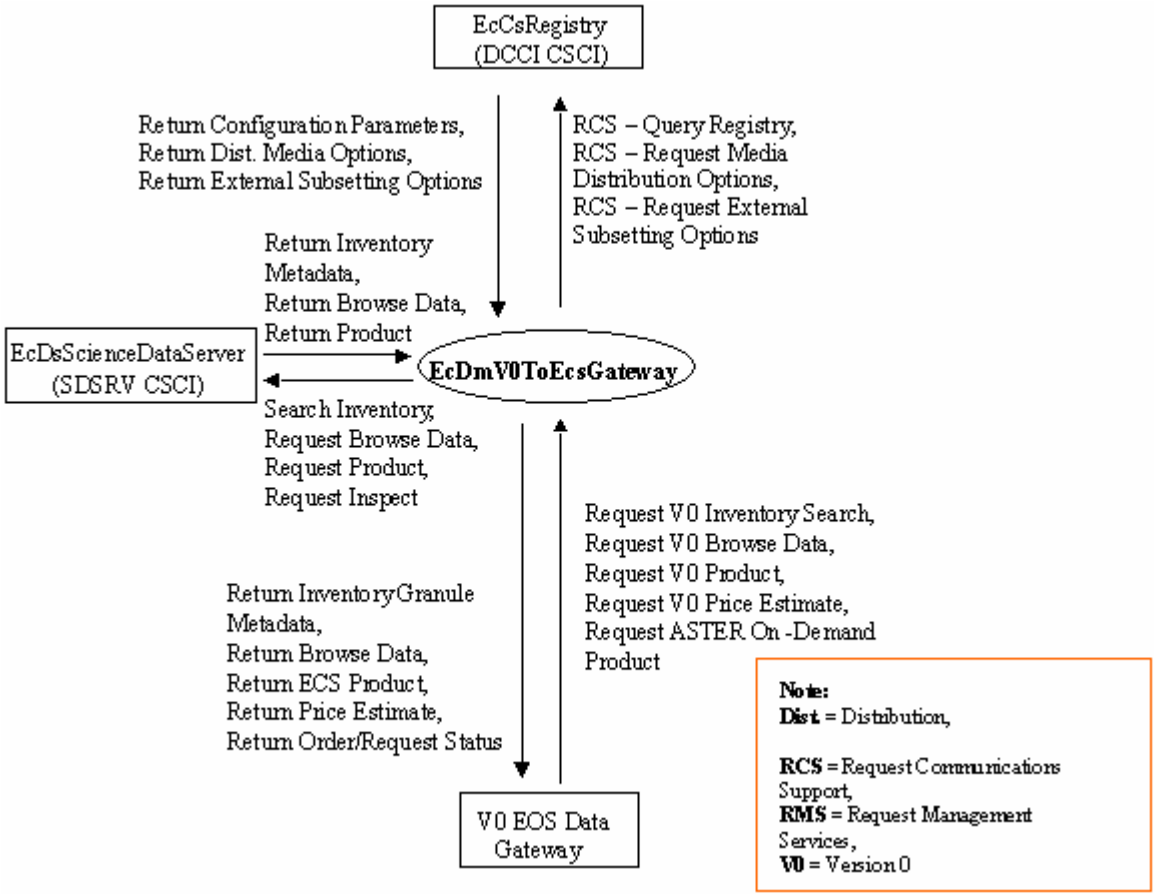


Figure 4.4-5. V0 GTWAY CSCI Architecture Diagram

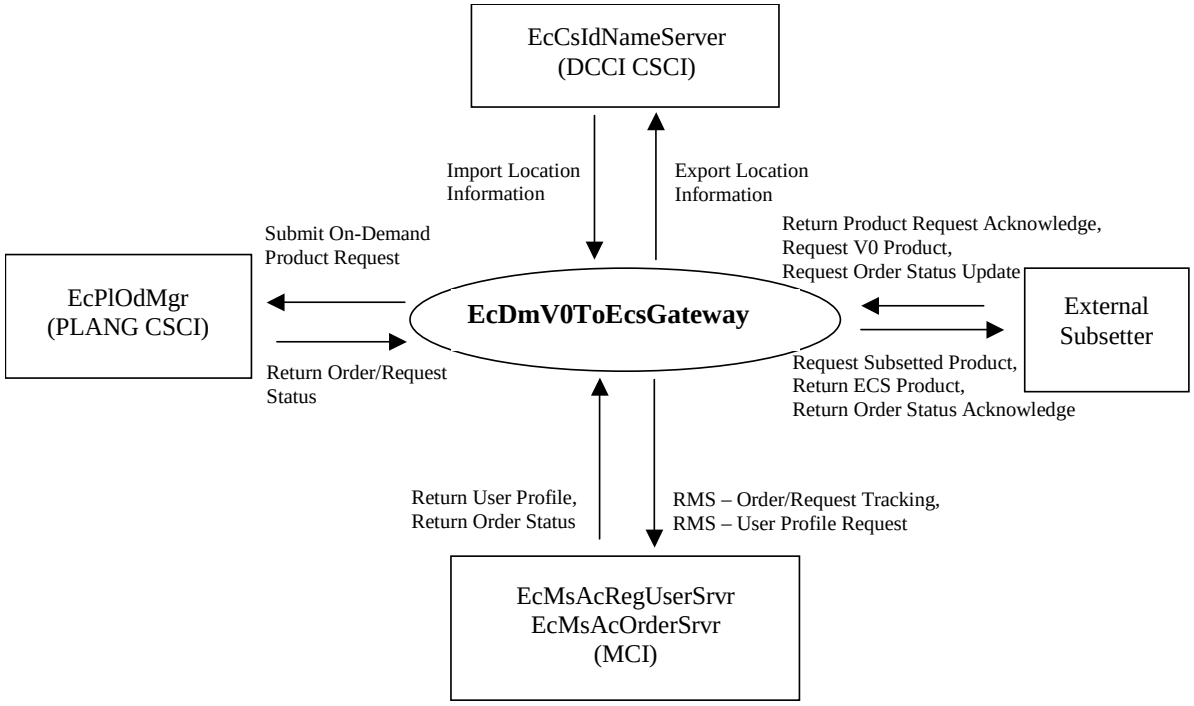


Figure 4.4-5. V0 GTWAY CSCI Architecture Diagram (cont.)

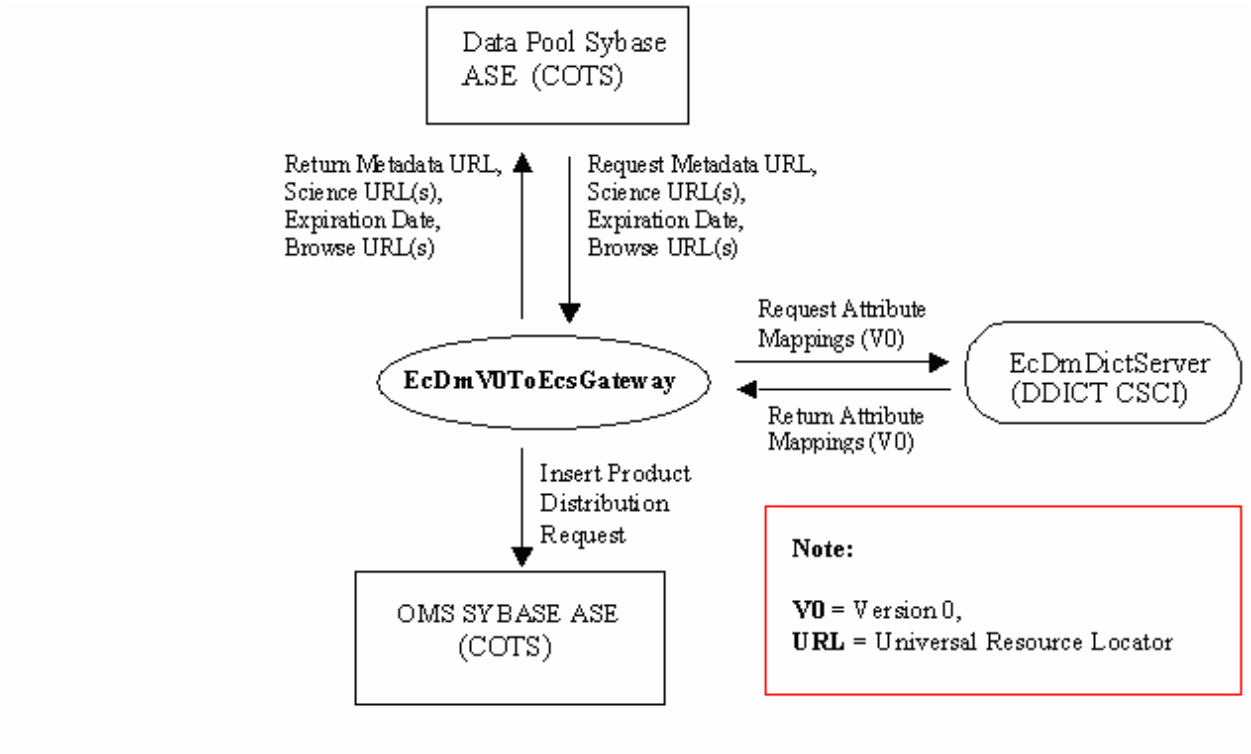


Figure 4.4-5. V0 GTWAY CSCI Architecture Diagram (cont.)

4.4.2.4 V0 Gateway Process Descriptions

Table 4.4-7 provides descriptions of the processes shown in the V0 GTWAY CSCI architecture diagrams.

Table 4.4-7. V0 GTWAY CSCI Processes

Process	Type	Hardware CI	COTS/ Developed	Functionality
EcDmV0ToEcsGateway	Server	INTHW	Developed	The V0 to ECS Gateway server allows users of the V0 IMS to query on data and services defined within the SDPS. Major Interfaces: • Inventory Search: Allows a user to perform searches for data granules within the SDPS archive. URLs to metadata, browse and science granules stored within the Data Pool are also returned along with the expiration date of those URLs. • Browse: Allows users to browse data granules previously found during a search • Product request: Provides the capability for users to submit data acquire requests for obtaining data granules from the SDPS archive • Order Status Update Request: Provides the capability for External Subsetters to provide status updates of orders being fulfilled by the Subsetter but submitted to ECS • Price Estimate requests: Allows users to request a price estimate for a given set of granules with spatial and/or band sub-setting constraints • Inspect Requests: Requests DSS to inspect granule metadata in support of a price estimate request. • ASTER On-Demand: Allows a user to do a product request on ASTER On-Demand products Server Supports: • Synchronous request processing • Asynchronous request processing • Multiple concurrent requests

4.4.2.5 V0 Gateway Process Interface Descriptions

Table 4.4-8 provides descriptions of the interface events shown in the V0 GTWAY CSCI architecture diagrams.

Table 4.4-8. V0 GTWAY CSCI Process Interface Events (1 of 9)

Event	Event Frequency	Interface	Initiated By	Event Description
Request Communications Support	Per process request	*Process:* EcCsIdNameServer *Libraries:* EcPf, Middleware, FoNs, FoIp, oodce *Classes:* EcPfManagedServer, EcPfClient, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy *Library (Common):* EcUr *Class:* EcUrServerUR *Library:* event *Class:* EcLgErrorMsg *Process:* EcCsRegistry *Library:* EcCsRegistry *Class:* EcRgRegistryServer_C	Process: EcDmV0ToEcsGateway *Library:* Common *Class:* DmGwManagedServer	The **DCCI CSCI** provides a library of services available to each SDPS and CSMS process. The process services required to perform specific assignments are requested from the DCCI CSCI. These services include: • CCS Middleware Support • Database Connection Services • File Transfer Services • Network & Distributed File Services • Bulk Data Transfer Services • Name/Address Services • Password Services • Server Request Framework (SRF) • Universal Reference (UR) • Error/Event Logging • Fault Handling Services • Mode Information • Query Registry - Retrieving the requested configuration attribute-value pairs from the Configuration Registry • Request Distribution Media Options from the Configuration Registry • Request External Subsetting Options from the Configuration Registry • Request ASTER On-Demand Options from the Configuration Registry

Table 4.4-8. V0 GTWAY CSCI Process Interface Events (2 of 9)

Event	Event Frequency	Interface	Initiated By	Event Description
Request V0 Inventory Search	One per inventory search request	*Process:* EcDmV0ToEcsGateway *Library:* RequestProcessing *Class:* DmGwInventoryRequest	*Process:* V0 EOS Data Gateway	The EcDmV0ToEcsGateway receives inventory search requests from the **V0 EOS Data Gateway** on behalf of an external ECS user.
Request V0 Browse Data	One per browse request	*Process:* EcDmV0ToEcsGateway *Library:* RequestProcessing *Class*: DmGwBrowseRequest	*Process:* V0 EOS Data Gateway	The EcDmV0ToEcsGateway receives browse data requests from the **V0 EOS Data Gateway** on behalf of an external ECS user.
Request V0 Product	One per product request	*Process:* EcDmV0ToEcsGateway *Library:* RequestProcessing *Class:* DmGwAcquireRequest	***Process:*** V0 EOS Data Gateway ***Process:*** External Subsetter *Library:* Ik *Class:* IK_RxODL	The EcDmV0ToEcsGateway receives product requests from the **V0 EOS Data Gateway** or **External Subsetter** on behalf of an external ECS user.

Table 4.4-8. V0 GTWAY CSCI Process Interface Events (3 of 9)

Event	Event Frequency	Interface	Initiated By	Event Description
Request V0 Price Estimate	One per product	*Process:* EcDmV0ToEcsGateway *Library:* RequestProcessing *Class:* DmGwPriceEstimateRequest	*Process:* V0 EOS Data Gateway	The EcDmV0ToEcsGateway receives price estimate requests from the **V0 EOS Data Gateway** on behalf of an external ECS user.
Submit On-Demand Product Request	One per product request	*Process:* EcDmV0ToEcsGateway *Library:* RequestProcessing *Class:* DmGwAcquireRequest	***Process:*** V0 EOS Data Gateway ***Process:*** EcPlOdMgr *Library:* PlOdMgrClient *Class:* PlOdMsgProxy	The EcDmV0ToEcsGateway receives product requests from the **V0 EOS Data Gateway** or **EcPlOdMgr** for ASTER On-Demand products.
Return Inventory Granule Metadata	One per requested granule	*Process:* V0 EOS Data Gateway *Process:* EcDsScienceDataServer *Library:* DsCl *Classes:* DsClESDTReferenceCollector, DsClRequest	*Process:* EcDmV0ToEcsGateway *Library:* RequestProcessing *Class:* DmGwInventoryRequest	The EcDmV0ToEcsGateway forwards the inventory granule metadata identifying the scene within the granule based on an inventory search request from the **V0 EOS Data Gateway** or the **EcDsScienceDataServer**.
Return Browse Data	One per request	*Process:* V0 EOS Data Gateway *Process:* EcDsScienceDataServer *Library:* DsCl *Classes:* DsClESDTReferenceCollector, DsClRequest	*Process:* EcDmV0ToEcsGateway *Library:* RequestProcessing *Class:* DmGwV0BrowseRequest	The EcDmV0ToEcsGateway receives browse data associated with a particular granule from the **V0 EOS Data Gateway** or the **EcDsScienceDataServer**.

Table 4.4-8. V0 GTWAY CSCI Process Interface Events (4 of 9)

Event	Event Frequency	Interface	Initiated By	Event Description
Return ECS Product	One per product request	*Process:* V0 EOS Data Gateway *Process:* External Subsetter *Library:* Ik *Class:* IK_RxODL	*Process:* EcDmV0ToEcsGateway *Library:* RequestProcessing *Class:* DmGwProductRequest	The EcDmV0ToEcsGateway receives products based upon V0 user product requests from the **V0 EOS Data Gateway** or an **External Subsetter**.
Return Price Estimate	One estimate per order	*Process:* V0 EOS Data Gateway	*Process:* EcDmV0ToEcsGateway *Library:* RequestProcessing *Class:* DmGwPriceEstimateRequest	The EcDmV0ToEcsGateway calculates the price estimate and sends the estimate to the user via the **V0 EOS Data Gateway**.
Return Order/ Request Status	One per product request	*Process:* EcDmV0ToEcsGateway *Library:* RequestProcessing *Class:* DmGwAcquireRequest	*Process:* V0 EOS Data Gateway	The EcDmV0ToEcsGateway returns an order status and request status acknowledgment back to the **V0 EOS Data Gateway**.
Search Inventory	One per inventory search request	*Process:* EcDsScienceDataServer *Library:* DsCl *Classes:* DsClESDTReferenceCollector, DsClRequest	*Process:* EcDmV0ToEcsGateway *Library:* RequestProcessing *Class:* DmGwInventoryRequest	The EcDmV0ToEcsGateway submits inventory search requests to the **EcDsScienceDataServer** on behalf of a user.
Request Browse Data	One per browse request	*Process:* EcDsScienceDataServer *Library:* DsCl *Classes:* DsClESDTReferenceCollector, DsClRequest	*Process:* EcDmV0ToEcsGateway *Library:* RequestProcessing *Class*: DmGwBrowseRequest	The EcDmV0ToEcsGateway submits requests for browse data to the **EcDsScienceDataServer** on behalf of a user.

Table 4.4-8. V0 GTWAY CSCI Process Interface Events (5 of 9)

Event	Event Frequency	Interface	Initiated By	Event Description
Request Product	One per product request	*Process:* EcDsScienceDataServer *Library:* DsCl *Classes*: DsClESDTReferenceCollector, DsClRequest	*Process:* EcDmV0ToEcsGateway *Library:* RequestProcessing *Class:* DmGwAcquireRequest	The EcDmV0ToEcsGateway submits product requests to the **EcDsScienceDataServer** on behalf of a user.
Request Inspect	One per price estimate request	*Process:* EcDsScienceDataServer *Library:* DsCl *Classes:* DsClESDTReferenceCollector, DsClRequest	*Process:* EcDmV0ToEcsGateway *Library:* RequestProcessing *Class:* DmGwPriceEstimateRequest	The EcDmV0ToEcsGateway sends a request for an inspection of granule metadata to the **EcDsScienceDataServer** in support of a price estimate request.
Return Product	One per product request	*Process:* EcDmV0ToEcsGateway *Library:* RequestProcessing *Class:* DmGwAcquireRequest	*Process:* EcDsScienceDataServer *Library:* DsCl *Classes*: DsClESDTReferenceCollector, DsClRequest	The **EcDsScienceDataServer** returns products based upon a product request from a user.
Return Configuration Parameters	One set per request	*Process:* EcDmV0ToEcsGateway *Library:* Common *Class:* DmGwConfigItems	*Process:* EcCsRegistry *Library:* EcCsRegistry *Class:* EcRgRegistryServer_C	The **EcCsRegistry** returns the attribute-value pairs (configuration parameters) to the EcDmV0ToEcsGateway upon request.
Return Dist. Media Options	One set per request	*Process:* EcCsRegistry *Library:* EcCsRegistry *Class:* EcRgRegistryServer_C	*Process:* EcDmV0ToEcsGateway *Library:* Common *Class:* DmGwConfigItems	The **EcCsRegistry** returns the requested distribution media options to the EcDmV0ToEcsGateway.

Table 4.4-8. V0 GTWAY CSCI Process Interface Events (6 of 9)

Event	Event Frequency	Interface	Initiated By	Event Description
Return External Subsetting Options	One set per request	*Process:* EcCsRegistry *Library:* EcCsRegistry *Class:* EcRgRegistryServer_C	*Process:* EcDmV0ToEcsGateway *Library:* Common *Class:* DmGwConfigItems	The **EcCsRegistry** returns the requested external subsetting options to the EcDmV0ToEcsGateway.
Export Location Information	Once at system startup and after each failure recovery	*Process:* EcCsIdNameServer *Libraries:* EcPf, Middleware, FoNs, FoIp, oodce *Classes:* EcPfManagedServer, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy	*Process:* EcDmV0ToEcsGateway *Library:* Common *Class:* DmGwManagedServer	The EcDmV0ToEcsGateway stores physical and logical server location information in the **EcCsIdNameServer**.
Return Product Request Acknowledge	One per product request	*Process:* EcDmV0ToEcsGateway *Library:* RequestProcessing *Class:* DmGwAcquireRequest	*Process:* External Subsetter *Library:* Ik *Class:* IK_RxODL	The EcDmV0ToEcsGateway receives a product result from an **External Subsetter** upon submission of a request for a subsetted product.
Request Order Status Update	One to three per subsetted product request	*Process:* EcDmV0ToEcsGateway *Library:* RequestProcessing *Class:* DmGwOrderStatusUpdate Request	*Process:* External Subsetter *Library:* Ik *Class:* IK_RxODL	The EcDmV0ToEcsGateway receives order status update requests from an **External Subsetter**. 1. Being Processed 2. Completed Processing or Terminated 3. Shipped
Request Subsetted Product	One per product request	*Process:* EcDmV0ToEcsGateway *Library:* RequestProcessing *Class:* DmGwAcquireRequest	*Process:* External Subsetter *Library:* Ik *Class:* IK_RxODL	The EcDmV0ToEcsGateway receives product requests from the **External Subsetter** for subsetted products on behalf of an external ECS user.

Table 4.4-8. V0 GTWAY CSCI Process Interface Events (7 of 9)

Event	Event Frequency	Interface	Initiated By	Event Description
Return Order Status Acknowledgment	One per Order Status Update request	*Process:* EcDmV0ToEcsGateway *Library:* RequestProcessing *Class:* DmGwOrderStatusUpdate Request	*Process:* External Subsetter *Library:* Ik *Class:* IK_RxODL	The EcDmV0ToEcsGateway returns an order status acknowledgment upon receipt of an order status update request from an **External Subsetter**.
Return User Profile	One per profile request	*Process:* EcDmV0ToEcsGateway *Class:* DmGwRequestReceiver	*Process:* EcMsAcRegUserSrvr *Library:* MsAcClnt *Class:* EcAcProfileMgr	The **EcMsAcRegUserSrvr** returns a user profile based upon a profile request from a user.
Return Order Status	One per order request	*Process:* EcDmV0ToEcsGateway *Class:* DmGwRequestReceiver	*Process:* EcMsAcOrderSrvr *Library:* MsAcClnt *Class:* EcAcOrderCMgr	The **EcMsAcOrderSrvr** returns an order id and status to the EcDmV0ToEcsGateway to send back to the user.
Return Order/Request Status	One per product request	*Process:* EcDmV0ToEcsGateway *Library:* Request Processing *Class:* DmGwAcquireRequest	*Process:* EcPlOdMgr *Library:* PlOdMgrClient *Class:* PlOdMsgProxy	The EcDmV0ToEcsGateway receives order/request status for ASTER On-Demand product request from the EcPlOdMgr.
Submit On-Demand Product Request	One per product request	*Process:* EcDmV0ToEcsGateway *Library:* Request Processing *Class:* DmGwAcquireRequest	*Process:* V0 EOS Data Gateway	The EcDmV0ToEcsGateway sends product requests to the EcPlOdMgr for ASTER On-Demand products

Table 4.4-8. V0 GTWAY CSCI Process Interface Events (8 of 9)

Event	Event Frequency	Interface	Initiated By	Event Description
Import Location Information	As required for processing	*Process:* EcDmV0ToEcsGateway *Class:* DmGwRequestReceiver	*Process:* EcCsIdNameServer *Libraries:* EcPf, Middleware, FoNs, FoIp, oodce *Classes:* EcPfManagedServer, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy	The EcDmV0ToEcsGateway requests server location information from the **EcCsIdNameServer**.
Request Metadata URL, Science URLs, Expiration Date, Browse URLs	One per granule	Data Pool Sybase ASE (COTS) *Library:* RWDb *Class:* RWDBManager	*Process:* EcDmV0ToEcsGateway *Library:* Common *Class:* DmGwDatapoolProxy	The EcDmV0ToEcsGateway retrieves metadata URL, science URLs, browse URLs and expiration dates from the **Sybase ASE**.
Request Attribute Mappings (V0)	One set per request	*Process:* EcDmDictServer, *COTS SW Library:* RWDBTools.h++ *Library:* DmLmDbi *Class:* DmLmIntQuery	*Process:* EcDmV0ToEcsGateway *Library:* Persistent *Class:* DmGwAttributeMap	The EcDmV0ToEcsGateway sends requests to the **EcDmDictServer Data Dictionary database via the Sybase ASE** for data collection and keyword mappings to translate requests from the V0 IMS to the SDPS and back again.
Return Attribute Mappings (V0)	One set per request	*Process:* EcDmV0ToEcsGateway *Library:* Persistent *Class:* DmGwAttributeMap	*Process:* EcDmDictServer *COTS SW* *Library:* RWDBTools.h++ *Library:* DmLmDbi *Class:* DmLmIntQuery	The EcDmV0ToEcsGateway receives data collection and keyword mappings to translate requests from the V0 IMS to the SDPS and back again from the **EcDmDictServer Data Dictionary database via the Sybase ASE**.

Table 4.4-8. V0 GTWAY CSCI Process Interface Events (9 of 9)

Event	Event Frequency	Interface	Initiated By	Event Description
Insert Product Distribution Request	One per service request	*Process:* Sybase Server (COTS)	*Process:* EcDmV0ToEcsGateway *Library:* OmClientlib *Classes:* OmSrDbInterface, OmSrV0InputIf	The EcDmV0ToEcsGateway inserts product distribution requests into the Order Manager DBMS.
Return Metadata URL, Science URL(s), Expiration Date, Browse URL(s)	One per granule	*Process:* EcDmV0ToEcsGateway *Library:* Common *Class:* DmGwDatapoolProxy	Data Pool Sybase ASE (COTS) *Library:* RWDb *Class:* RWDBManager	The **Sybase ASE** returns the Metadata URL, Science URL(s), Browse URL(s) and Expiration Dates to the EcDmV0ToEcsGateway.
Request Management Services (RMS)	At system startup or shutdown and for restarts	*Process:* EcDmV0ToEcsGateway	DAAC unique startup scripts	**System startup and shutdown -** Please refer to the release-related, current version of the Mission Operations Procedures for the ECS Project document (611) and the current ECS Project Training Material document (625), identified in Section 2.2.1 of this document.
RMS (Cont.)	One per product order	*Process:* EcMsAcOrderSrvr *Library:* MsAcClnt *Class:* EcAcOrderCMgr	*Process:* EcDmV0ToEcsGateway *Class:* DmGwRequestReceiver	**DMS Order/Request Tracking Update** – The V0 GTWAY interfaces with the Order/Request tracking service, **EcMsAcOrderSrvr**, to create a user product order.
RMS (Cont.)	One per profile request	*Process:* EcMsAcRegUserSrvr *Library:* MsAcClnt *Class:* EcAcProfileMgr	*Process:* EcDmV0ToEcsGateway *Class:* DmGwRequestReceiver	**User Profile Request** - The **EcMsAcRegUserSrvr** provides requesting processes with User Profile information such as e-mail address and shipping address to support their processing activities.

4.4.2.6 V0 Gateway Data Stores

Table 4.4-9 provides descriptions of the data stores shown in the V0 GTWAY CSCI architecture diagram. The V0 to ECS and ECS to V0 Gateway processes access the Data Dictionary data store.

Table 4.4-9. V0 GTWAY CSCI Data Store

Data Store	Type	Functionality
EcDmDictService	Database	The Data Dictionary database, EcDmDictService, is a Sybase relational database that persistently stores the collection and collection related information on a physical disk medium. The DDICT database is replicated wholly to each DAAC. The data stores in the Data Dictionary database used by the V0 GTWAY CSCI are: • Collection Types: A list of all the data types within the SDPS • Collection Attributes and Keywords: Attributes and keywords associated with collections originating within and outside the SDPS are used by the V0 GTWAY CSCI to translate requests between the V0 IMS and the SDPS.

4.4.3 Data Management Subsystem Hardware

The primary components of the Data Management Subsystem include two hardware CIs, Data Management Hardware CI (DMGHW) and Interface Hardware CI (INTHW), co-owned by the Interoperability Subsystem, as described below. Custom code and applications are loaded on the internal disks of all hosts to prevent dependencies on specific hosts or peripherals. The general-purpose workstations are standalone hosts without fail-over capability. In the event of a host failure, any of the available workstations could be used to support end user DAAC maintenance.

4.4.3.1 Data Management Hardware CI (DMGHW) Description

The DMGHW CI includes general-purpose low-end SUN workstations. In the EBIS Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the HWCI, and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping. These workstations are used as end user workstations in maintenance of each of the respective DAAC sites. The Server is used to support Sybase database replication and backup.

4.4.3.2 Interface Hardware CI (INTHW) Description, as used by the Data Management Subsystem

The INTHW CI includes an Interface Server. Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the Interface HWCI and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping. The Interface Server supports the Client Subsystem and a portion of the Communications Subsystem. DMS software runs on this host: DDICT and V0 GTWAY. The Data Dictionary Server (EcDmDictServer) allows authorized users to perform data searches, inserts, updates and deletions to data within the Data Dictionary Database. The V0 GTWAY consists of multiple processes to allow access to data and services between the ECS Data Server and the V0 IMS System.

4.5 Order Manager Subsystem Overview

The Order Manager subsystem (OMS) manages all orders arriving via the Data Management subsystem's V0 Gateway (V0 GTWAY), (i.e., submitted by EDG and ECHO users), data orders submitted by the Spatial Subscription Server (NSBRV), orders from Machine to Machine Gateway (MTMGW), orders from SCLI (i.e. submitted by S4PM) and orders from the Data Pool Web GUI (including HEG orders). There are plans to interface all EMD components that handle user orders with the new Order Management Service in the future, though not during Synergy V.

The capability includes a server (the Order Manager Server) to which the data distribution orders are submitted. If the media type or ESDTs of a request are configured as Synergy III processing mode, or the Order Manager Server is configured for Synergy III mode, the server distributes the order to the appropriate EMD service (i.e., Product Distribution System or Science Data Server), depending on whether the request is for hard media or electronic distribution. Otherwise, the server stages the order to Data Pool storage (and creates links from staged files to the FtpPull directory in the Data Pool storage if the distribution type is FtpPull), distributes the order to the appropriate service (i.e., Production Modules or OMS Ftp Driver) depending on whether distribution type is hard media or FtpPush, then sends a Distribution Notice to the end user when the order is considered shipped (i.e., for media order, request status is updated as "Shipped" in the MSS Database; for FtpPull order, as soon as the order is staged and file links are made in the Data Pool storage; for FtpPush order, the Order Manager Server finishes pushing all the data for the order to its destination). HEG orders require further processing by the HEG Server. From the original HEG order and its associated processing instructions, the Order Manager creates HEG requests. The HEG requests are submitted to the HEG Server through the HEG API. HEG server processes the HEG requests and returns the final output to the Order Manager Server which then distributes the final output to the end user. The Order Manager Subsystem also includes a database that stores all order information persistently as soon as an order is received by EMD and before its receipt is acknowledged. This allows operators to resubmit an order if it encounters errors downstream, and allows the Order Management Service to perform some up front checks on the order and alert the operators if their intervention is needed.

Order Manager Subsystem Context

Figure 4.5-1 is the Order Manager Subsystem context diagram. The diagram shows the events sent to the Order Manager Subsystem and the events the Order Manager Subsystem sends to other subsystems. Table 4.5-1 provides descriptions of the interface events shown in the Order Manager Subsystem context diagram.

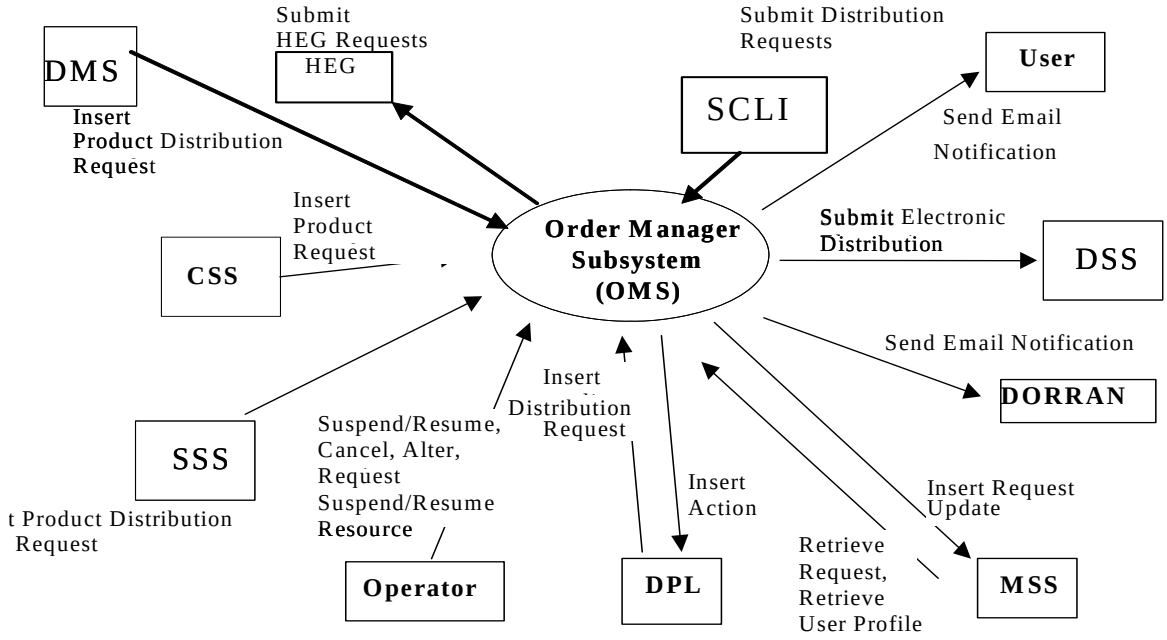


Figure 4.5-1. Order Manager Subsystem Context Diagram

Table 4.5-1. Order Manager Subsystem Interface Events (1 of 2)

Event	Interface Event Description
Submit Media Distribution Requests	The OMS submits all physical media requests classified as Synergy III processing mode using the V0 ODL protocol.
Submit Electronic Distribution Requests	The OMS submits all product distribution requests whose processing mode is Synergy III, and which specify electronic media types (e.g., FTPPUSH, FTPPULL) to the **Data Server Subsystem** (DSS).
Insert Product Distribution Requests	The OMS receives Product Distribution Requests from the **Data Management Subsystem (DMS)**, the **Spatial Subscription Server** (SSS) and the **CSS** (Machine to Machine Gateway CSC).
Insert Media Distribution Requests	The OMS receives Media Distribution Requests from the **DPL Subsystem** (Web GUI).
Insert Actions	The OMS submits DPL insert actions from the **DPL Subsystem**.
Suspend/Resume, Cancel, Alter and Retrieve Requests	The **Operator** suspends, resumes, cancels, alters and retrieves requests from the OMS (OMS DB).
Suspend/Resume Resources	The Operator suspends, resumes dispatching to all or selected resources.
Send Email Notification	The OMS sends email notification to **users** and **DORRAN** when a request is altered, canceled or shipped or when a request is intervened by operators, or when an alert or intervention is generated. The OMS sends email notification to users from the **DPL** when OMS receives requests.

Table 4.5-1. Order Manager Subsystem Interface Events (2 of 2)

Event	Interface Event Description
Insert Request	The OMS inserts requests into the MSS (MSS DB).
Update Request	The OMS updates request information in the MSS (MSS DB).
Retrieve Request	The OMS retrieves request information from the MSS (MSS DB).
Retrieve User Profile	The OMS retrieves user profile information from the MSS (MSS DB).
Submit Distribution Request	The OMS receives Distribution Requests from the SCLI.
Submit Heg Request	The OMS submits HEG requests to the HEG Server.

Order Manager Subsystem Structure

- The Order Manager Subsystem consists of three CSCIs: the OMSRV, the OM GUI (described in the 609 document), and the Production Module. The Order Manager Server (EcOmOrderManager) is a software configuration item. The Order Manager Server receives product distribution requests and submits them to the appropriate EMD component based upon the media type specified for the request. If the media type or ESDTs of the request are configured as Synergy III processing mode, then request is dispatched to the SDSRV. Otherwise, the server stages the request to the DPL storage and distributes them to the OMS or pushes them directly to the end-user. For HEG Orders, the Order Manager creates HEG requests based on processing instructions and submits the HEG requests to the HEG Server. The output of the HEG requests are later distributed to the end user. Order Manager Subsystem information is stored persistently in a relational Database Management System (DBMS). The Order Manager GUI (OMGUI) is used to monitor and control the operations of the Order Manager Server. In addition, the OMGUI is used to respond to Operator Intervention Requests generated by the Order Manager Server. Production module is responsible for creating the physical medial associated to hard media requests.

Use of COTS in the Order Manager Subsystem

- RogueWave's Tools.h++

  The Tools.h++ class libraries are used by the OMS to provide basic functions and objects such as strings and collections. These libraries must be installed with the OMS software for any of the OMS processes to run.

- RogueWave's DBTools.h++

  The DBTools.h++ C++ class libraries are used to interact with the Sybase database SQL server. The use of DBTools buffers the OMS processes from the relational database used. These libraries must be installed with the OMS software for the Order Manager Server to run and allow client processes to perform queries of Order Manager database information.

- Sybase Open Client / CT_LIB

  The Sybase Open Client provides access between OMS custom code and the Sybase SQL Server DBMS.

- Sybase Server

  The Sybase SQL server provides access for OMS to insert, update and delete Product Distribution Requests, OMS configurations, and Operator Interventions. The Sybase SQL Server must be running during operations for the OMS to process Product Distribution Requests.

4.5.1 Order Manager Subsystem Software Description

4.5.1.1 Order Manager Server CSCI Functional Overview

The Order Manager Server (OMSRV) CSCI executes as a process and interacts with the following CSCIs: Order Manager Database, the Science Data Server (SDSRV), and the Data Pool System (DPL). The V0 Gateway, the Spatial Subscription Server (SSS), Machine to Machine Gateway (MTMGW), HEG Server, and the Data Pool Web GUI submit product distribution requests to the OMS. These requests are all validated. If the request is valid and the ESDT or media type of the request is configured in such a way that it must go through the Synergy III processing mode or the request is pre-staged, then the request is submitted directly to the SDSRV. Otherwise, the server stages the request to DPL storage. Hard media requests staged in DPL storage area are distributed through the production module while electronic media requests are directly pushed to the end user. Note that Order Manager Server dispatches HEG requests to the HEG Server for processing before being distributed to the end user. For invalid request, an Operator Intervention is generated. DAAC OPS personnel can use the Order Manager GUI to correct and resubmit the request. In response to an intervention, the Operator can also generate an email message, which is sent to the user by the Order Manager Server. The Order Manager Server also generates an alert and sends an email to a pre-configured email address when it detects internal or external resource failure. In the meanwhile, it halts dispatching of the requests that are utilizing those failed resources.

4.5.1.2 Order Manager Server CSCI Context

Figure 4.5-2 is the Order Manager Server CSCI context diagrams. The diagrams show the events sent to the Order Manager Server CSCI and the events the Order Manager Server CSCI sends to other CSCIs. Table 4.5-2 provides descriptions of the interface events shown in the Order Manager Server CSCI context diagrams.

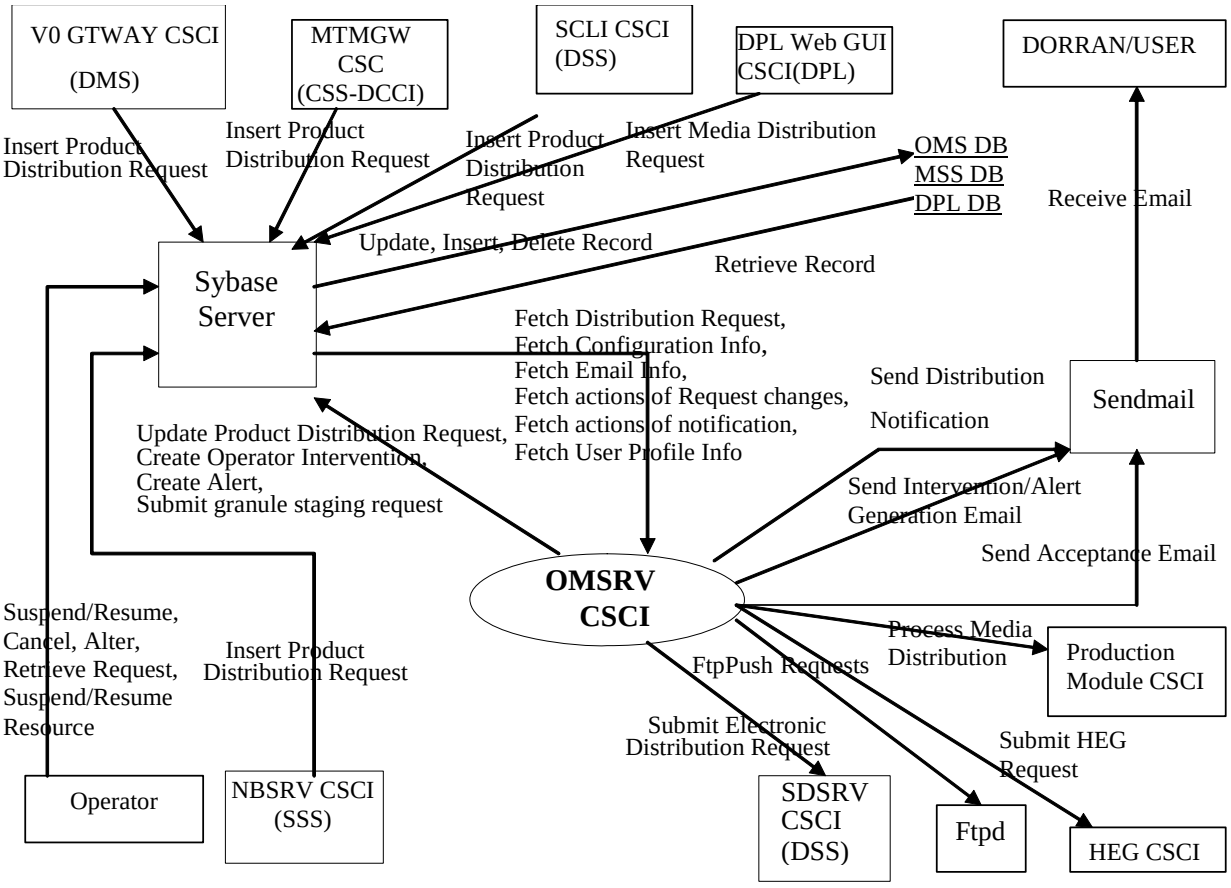


Figure 4.5-2. Order Manager Server CSCI Context Diagram

Table 4.5-2. Order Manager Server CSCI Interface Events (1 of 2)

Event	Interface Event Description
Submit Media Distribution Request	The OMSRV CSCI submits Synergy III requests for product distribution on physical media.
Submit Electronic Distribution Request	The OMSRV CSCI submits electronic product distribution requests to the **Science Data Server (SDSRV) CSCI** if the media types or ESDTs of the request are configured in such a way that they must go through Synergy III processing mode.
Send Distribution Notification	The OMSRV CSCI sends distribution notifications to the **end-users**.
Send Intervention/Alert Generation Email	The OMSRV CSCI sends intervention/alert generation email to the **end-users**.
Send Acceptance Email	The OMSRV CSCI sends a request acceptance email to the **DPL Web GUI** users upon receiving the request in the OMS DB.
Receive Email	The **DORRAN** billing and accounting system and **User** receive a status email sent by the OMSRV CSCI when request is interventioned, shipped or failed

Table 4.5-2. Order Manager Server CSCI Interface Events (2 of 2)

Event	Interface Event Description
Insert Product Distribution Request	The V0 GTWAY CSCI, NBSRV CSCI, SCLI CSCI and Machine–to-Machine Gateway (MTMGW) CSC insert product distribution requests into the **Sybase Server** (**OMS DB**) to be queued for processing by the OMSRV CSCI.
Insert Media Distribution Request	The DPL Web GUI inserts media distribution request into the **Sybase Server** (**OMS DB**).
Update Product Distribution Request	The OMSRV CSCI updates product distribution requests in the **Sybase Server** (**OMS DB** as well as the **MSS DB**).
Create Operator Intervention	The OMSRV CSCI creates new Operator Intervention for request failures in the **Sybase Server (OMS database)**.
Create Alert	The OMSRV CSCI creates an alert for resource failures and stores the alert in the **Sybase Server (OMS database)**.
Submit granule staging request	The OMSRV CSCI submits a request to stage a granule to the **Sybase Server** (DPL storage in the OMS DB), which in turn calls DPL stored procedures to insert an action into the DPL DB.
Fetch Distribution Request	The OMSRV CSCI retrieves information associated with a product distribution request from the **Sybase Server** (OM Database).
Fetch Configuration Info	The OMSRV CSCI retrieves the OMSRV Configuration information from the **Sybase Server**.
Fetch Email Info	The OMSRV CSCI retrieves information related to an operator intervention required to generate an email notification from the **Sybase Server**.
Fetch actions of Request changes	The OMSRV CSCI retrieves actions regarding request changes, such as, request priority change, cancel request, suspend request, and update request ftppush parameters from the **Sybase Server**.
Fetch actions of notification	The OMSRV CSCI retrieves actions regarding granule staged and DPL file system modified notification from the **Sybase Server**.
Fetch User Profile Info	The OMSRV CSCI retrieves user profile information from the Sybase Server (OMS DB), which in turn calls an MSS stored procedure to retrieve user profile information from the MSS DB.
FtpPush Request	The OMSRV CSCI Ftp Pushes a request to the **end-user**.
Process Media Distribution	The OMSRV CSCI submits physical media request with Synergy IV processing mode to the Production Module.
Submit HEG Request	The OMSRV CSCI submits HEG requests to the HEG Services for processing.
Suspend/Resume, Cancel, Alter and Retrieve Request	The Operator suspends, resumes, cancels, alters and retrieves requests from the **Sybase Server** (**OMS DB**).
Suspend/Resume Resources	The OMSRV CSCI suspends or resumes dispatching to all or selected resources in the **Sybase Server**.
Update, Insert, Delete Record	The **Sybase Server** performs update, insert, and delete database operations to the OMS DB, MSS DB and DPL DB.
Retrieve Record	The **Sybase Server** performs retrieval database operations to/from the OMS DB, MSS DB and DPL DB.

4.5.1.3 Order Manager Server CSCI Architecture

Figure 4.5-3 is the Order Manager Server (OMSRV) CSCI architecture diagram. The diagram shows the events sent to the OMSRV CSCI processes and the events the OMSRV CSCI processes send to other processes.

The OM Server CSCI consists of one process. This process is the EcOmOrderManager process.

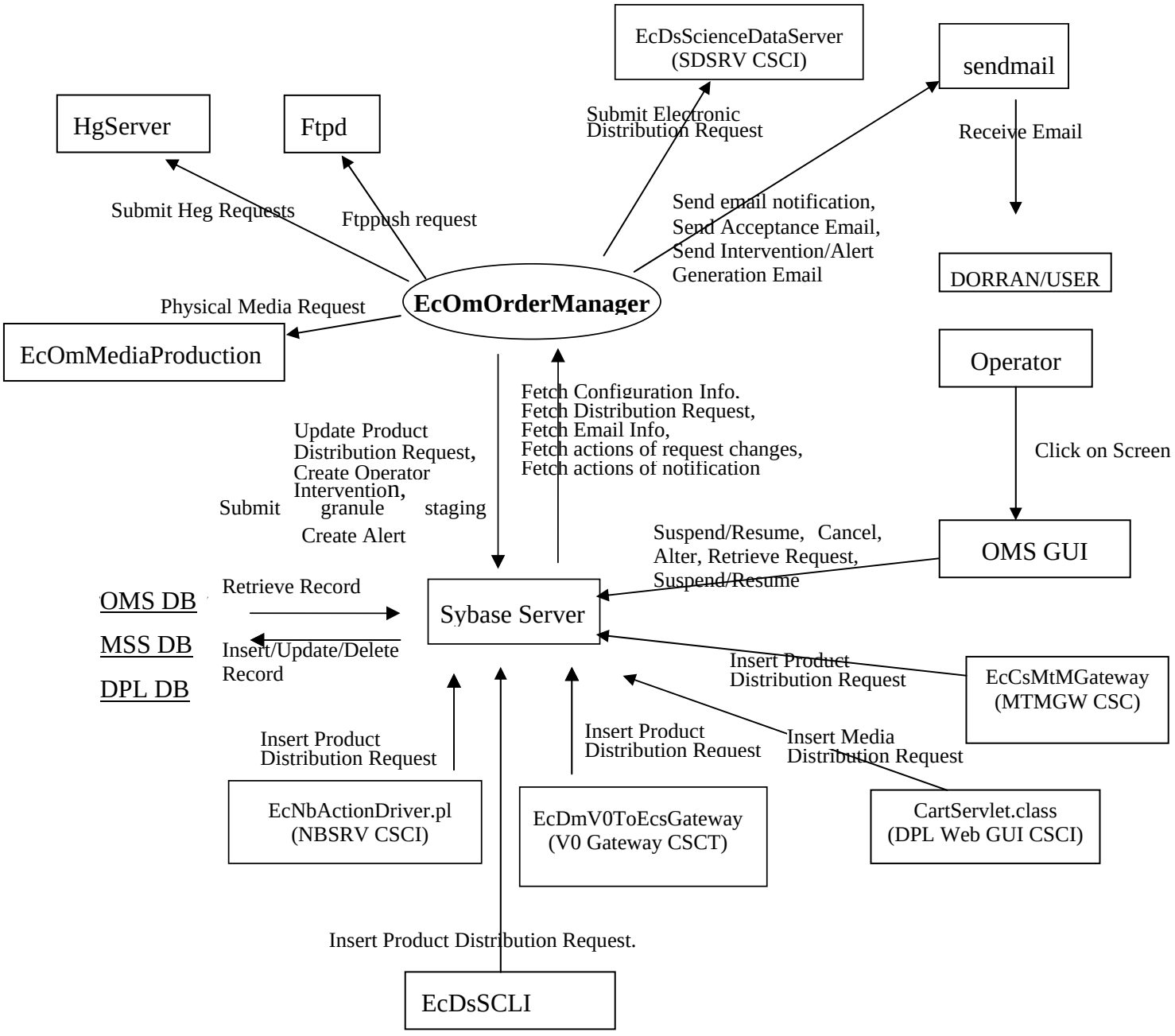


Figure 4.5-3. Order Manager Server CSCI Architecture Diagram

4.5.1.4 Order Manager Server CSCI Process Description

Table 4.5-3 provides descriptions of the processes shown in the OMSRV CSCI architecture diagram.

Table 4.5-3. OMSRV CSCI Process

Process	Type	Hardware CI	COTS/ Developed	Functionality
EcOmOrderManager	Server	ACMHW	Developed	In Synergy III mode of operation, the Order Manager Server retrieves product distribution requests from the Order Manager DBMS and dispatches them to the SDSRV. Otherwise, the Order Manager Server stages the request data into the DPL storage. Hard media request data staged in DPL storage are distributed through the Production Module while Ftp media requests are directly pushed to the end user. Note that HEG requests are first dispatched to the HEG Server for processing. The processed output is then distributed to the end user. Order Manager Server sends a Distribution Notification to the end-user on completing an order.

EMD Baseline Information System (EBIS) Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the HWCI, and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

4.5.1.5 Order Manager Server CSCI Interface Description

Table 4.5-4 provides descriptions of the interface events shown in the Order Manager Server (OMSRV) CSCI architecture diagram.

Table 4.5-4. Order Manager Server CSCI Process Interface Events (1 of 5)

Event	Event Frequency	Interface	Initiated By	Event Description
Physical Media Requests	One or more per service requests	*Process:* EcOmPdMediaProduction	*Process:* EcOmOrderManager *Class:* OmSrPrepareMediaAction OmSrCreateMediaAction	If the request processing mode is Synergy IV and the media type is hard media, the Order Manager Server uses the production module to create the physical media.

Table 4.5-4. Order Manager Server CSCI Process Interface Events (2 of 5)

Event	Event Frequency	Interface	Initiated By	Event Description
Submit Electronic Distribution Request	One or more per service request	*Process:* EcDsScienceData Server	*Process:* EcOmOrderManager *Executable:* EcOmSrCLI *Library:* DsClientSideLibs *Class:* OmSdsrvIf	The EcOmOrderManager submits requests for Electronic Product Distribution Requests to the EcDsScienceDataServer if the request processing mode is Synergy III.
Insert Product Distribution Request	One per service request	*Process:* Sybase Server (COTS)	*Process:* EcDmV0ToEcsGateway EcCsMtMGateway *Script:* EcNbActionDriver.pl	The EcDmV0ToEcsGateway, EcCsMtMGateway and EcNbActionDriver.pl insert product distribution request into OMS DB.
Submit Heg Request	One or more per Service Request	*Process:* *HgServer*	*Process:* EcOmOrderManager *Class:* OmSrHegProcessingActio n	The EcOmOrderManager submits HEG requests to the Heg Server.
Insert Media Distribution Request	One per service request	*Process:* Sybase Server (COTS)	*Process:* CartServlet.class	The CartServlet.class (DPL Web GUI CSCI) inserts media distribution request. Media distribution request could be HEG request.
Send acceptance email	One per email notification request	*Process:* Sendmail (COTS)	*Process:* EcOmOrderManager *Class:*	The EcOmOrderManager sends email notification to the DPL Web GUI end-users upon receipt of the request.
Send Intervention/ Alert Generation email	One per email notification request	*Process:* Sendmail (COTS)	*Process:* EcOmOrderManager *Class:*	The EcOmOrderManager sends intervention/ alert generation email to a configured user email account.
Receive Email	One per email notification	*End User:* DORRAN Configured User	*Process:* Sendmail (COTS)	The DORRANbilling and accounting system and User receive email sent by the EcOmOrderManager.
Send email notification	One per email notification request	*Process:* Sendmail (COTS)	*Process:* EcOmOrderManager *Class:* OmSrEmailRequest	The EcOmOrderManager sends email notifications to the end-users.

Table 4.5-4. Order Manager Server CSCI Process Interface Events (3 of 5)

Event	Event Frequency	Interface	Initiated By	Event Description
Fetch configuration info	One per startup/ One per configurable interval	*Process:* Sybase Server (COTS)	*Process:* EcOmOrderManager *Library:* Sybase Ct-library *Class:* OmSrDbInterface	The EcOmOrderManager retrieves configuration information from the Sybase Server (OMS database).
Fetch Distribution Request	One per configurable interval	*Process:* Sybase Server (COTS)	*Process:* EcOmOrderManager *Library:* Sybase Ct-library *Classes:* OmSrDbInterface, OmSrDistributionRequest	The EcOmOrderManager retrieves information associated with a product distribution request from the database.
Fetch actions of request changes	One per configurable interval	*Process:* Sybase Server (COTS)	*Process:* EcOmOrderManager *Library:* Sybase Ct-library *Classes:* OmSrDbInterface, OmSrDistributionRequest	The EcOmOrderManager retrieves actions regarding request changes, such as, request priority change, cancel request, suspend request, and update request Ftp Push parameters.
Fetch actions of notification	One per configurable interval	*Process:* Sybase Server (COTS)	*Process:* EcOmOrderManager *Library:* Sybase Ct-library *Classes:* OmSrDbInterface, OmSrDistributionRequest	The EcOmOrderManager retrieves actions regarding granule staged, and DPL file system modified notification.
Fetch email info	One per configurable interval	*Process:* Sybase Server (COTS)	*Process:* EcOmOrderManager *Library:* Sybase Ct-library *Classes:* OmSrDbInterface, OmSrDistributionRequest, OmSrEmailRequest	The EcOmOrderManager retrieves information related to an operator intervention required to generate an email notification from the Sybase Server (OM DB).

Table 4.5-4. Order Manager Server CSCI Process Interface Events (4 of 5)

Event	Event Frequency	Interface	Initiated By	Event Description
Update Product Distribution Request	One per request	*Process:* Sybase Server (COTS)	*Process:* EcOmOrderManager *Library:* Sybase Ct-library *Classes:* OmSrDbInterface, OmSrDistributionRequest	The EcOmOrderManager updates existing product distribution requests in the Sybase Server (OMS DB and MSS DB).
Create Operator Intervention	One per request	*Process:* Sybase Server (COTS)	*Process:* EcOmOrderManager *Library:* Sybase Ct-library *Class:* OmSrDbInterface	The EcOmOrderManager creates a new Operator Intervention request in the Sybase Server (OM DB).
Create Alert	One per resource	*Process:* Sybase Server (COTS)	*Process:* EcOmOrderManager *Library:* Sybase Ct-library *Class:* OmSrDbInterface	The EcOmOrderManager creates an alert related to a resource failure such as Ftp Push Destination, Archive, and DPL File System failure to store in the Sybase Server.
Insert Product Distribution Request	One per service request	*Process:* Sybase Server (COTS)	*Process:* EcDmV0ToEcsGateway EcCsMtMGateway EcDsSCLI *Script:* EcNbActionDriver.pl *Library:* OmClientlib *Classes:* OmSrDbInterface, OmSrV0InputIf	The EcDmV0ToEcsGateway and the EcNbActionDriver, EcCsMtMGateway, the EcOmSCLI, DPL Web GUI insert product distribution requests into the Sybase Server (Order Manager DB).
Insert/Update/ Delete/Retrieve Record	One per request	*Database:* OMS DB, MSS DB, DPL DB (COTS)	*Process:* Sybase Server (COTS)	The Sybase server performs database operations (inserts, updates, deletions and retrievals) to the OMS DB, DPL DB and MSS DB.
Ftppush request	One per file	*Process:* Ftpd (COTS)	*Process:* EcOmOrderManager	The EcOmOrderManager Ftp Pushes request data to the end-user.
Click on Screen	One per click	*Scripts:* OMS GUI scripts.	Operator	The Operator clicks on the screen to select the action.

Table 4.5-4. Order Manager Server CSCI Process Interface Events (5 of 5)

Event	Event Frequency	Interface	Initiated By	Event Description
Suspend/ Resume, Cancel, Alter and Retrieve Requests	One per click	*Process:* Sybase Server (COTS)	*Script:* OMS GUI script	The OMS GUI scripts send suspend/resume, cancel, alter and retrieve request commands to the Sybase Server.
Suspend/ Resume Resource	One per click	*Process:* Sybase Server (COTS)	*Script:* OMS GUI script	The OMS GUI scripts send suspend/resume resource commands to the Sybase Server.

4.5.1.6 Data Stores

There are data stores associated with the Order Manager Server. They are the OMS DB, DPL DB and MSS DB. Table 4.5-5 provides a description of these data stores.

Table 4.5-5. CSCI Data Stores

Data Store	Type	Description
OMS DB	Sybase	OMS Database is designed to store the persistent information of user request, processing mode configuration, media configuration, staging policy configuration, Ftp Push policy configuration, request aging configuration, and request cleanup configuration.
DPL DB	Sybase	The Data Pool (DPL) database implements the large majority of the persistent data requirements for the DPL subsystem which supports Large online cache of important EMD data at each DAAC and avoids tape access to EMD archive.
MSS DB (Order Tracking DB)	Database	The Order Tracking DB contains product orders and user requests with the associated current processing status.

4.5.1.7 Production Module CSCI Functional Overview

The Production Module is the interface between Order Manager Server and the various tape and disc hardware. Order Manager Server places an input file for the Production Module across NFS mount onto the platform to which the physical media devices are attached. The Production Module references this file and stages the indicated data under a single directory so that it can be processed to tape or disc. The Production Module processes one volume of a request at a time. For tapes the Production Module writes to media through a system call to pax. For discs, it places a control file and the ISO image created from the staged data in a directory that can be seen by the Rimage CD/DVD production software residing on a PC. The Production Module returns overall job status to Order Manager Server through QuickServer. If there are errors for particular granules, the Production Module places a list of the problem granules in a file, which can be referenced by Order Manager across the same NFS mount in the event of failure.

Verification is handled by a Perl script, OmPdQcMain.pl. The script interacts with the tape and disc devices to do a listing of the media and compares the filenames and file sizes listed with those created in a summary file during production. Volume and Granule level verification status is returned to the Order Manager in the same manner as in production.

4.5.1.8 Production Module CSCI Context

Figure 4.5-4 is the Production Module CSCI Context Diagram. The diagram indicates the interaction the Production Module has with the Order Manager CSCI and the Data Pool File System.

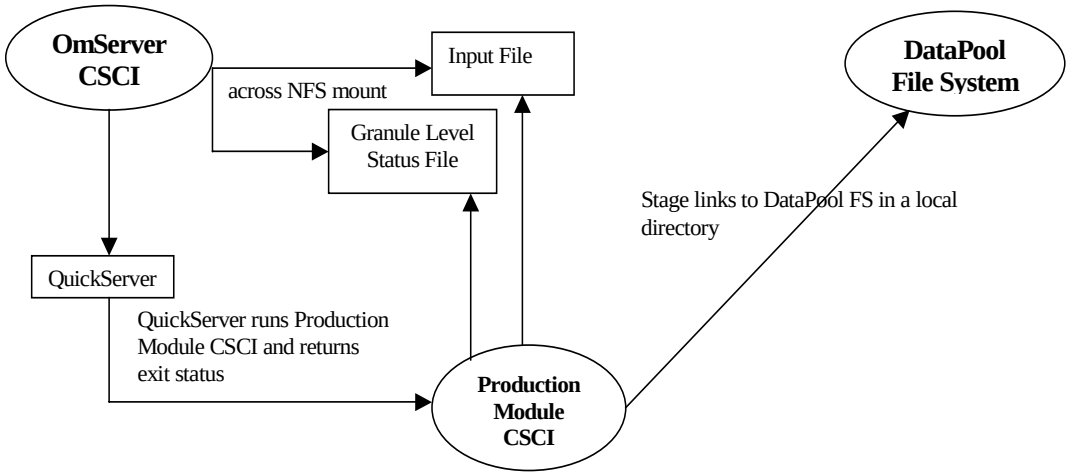


Figure 4.5-4. Production Module CSCI Context Diagram

Table 4.5-6 describes the Production Module CSCI Interface Events.

Table 4.5-6. Production Module CSCI Interface Events (1 of 2)

Event	Interface Event Description
Tape Media Preparation	The Production Module CSCI stages soft links to the indicated Data Pool files under a single directory. It creates a summary file and the file used for the tape label. It returns status to the Order Manager CSCI through QuickServer.
Tape Media Creation	The Production Module CSCI tars the data to tape, prints the tape label and returns status to the Order Manager CSCI through QuickServer.

Table 4.5-6. Production Module CSCI Interface Events (2 of 2)

Event	Interface Event Description
Disc Media Preparation	The Production Module CSCI stages soft links to the indicated Data Pool files under a single directory. It calls the COTS product, mkisofs, to create an ISO image file in a directory that can be seen by the Rimage CD/DVD Production PC. It creates the files for the jewel case insert, the summary file and returns status to the Order Manager CSCI through QuickServer.
Disc Media Creation	The Production Module CSCI creates the Rimage Control file in a directory that can be seen by the Rimage interface software, Data Publisher. The Production Module polls the Data Publisher log until it detects completion of the disc burning by Rimage and prints the Jewel Case Insert.
Granule Level Error	The Production Module CSCI writes the granule numbers and error code to a file which can be seen by Order Manager and returns a failure status code
Tape Verification	The Production Module CSCI QC script does a listing of the tape in the specified drive. It compares the filenames and sizes in a listing with the summary file created during the production process. Status is returned to Order Manager Server though QuickServer.
Disc Verification on PC	A Production Module CSCI script runs continually on a PC. It polls a mapped Unix drive for the appearance of a signal file. Using QuickServer, Order Manager starts the Production Module CSCI QC script, which places the signal file. The PC script does a listing of the CD/DVD drive on the PC and places it into a file on the Unix drive. The QC script then parses this file and the summary file and compares the filenames and sizes. Status is returned to Order Manager Server though QuickServer.
Disc Verification on UNIX	The Production Module CSCI QC script does a listing of the /CDROM directory and compares the filenames and sizes with those in the summary file. Status is returned to Order Manager Server though QuickServer.
Order Cleanup	The Production Module CSCI cleanup script is executed by Order Manager through QuickServer and deletes temporary files not needed for later event tracing.
Archive/Cleanup	The Production Module CSCI Archive/Cleanup GUI creates a cleanup.sh file and writes an entry in the crontab to execute this file. Remaining artifacts and logs can be assigned for archiving or removal

4.5.1.9 Production Module CSCI Architecture

Figure 4.5-5 is the Production Module CSCI architecture diagram. The diagram shows the interaction with the media hardware as directed by the Order Manager CSCI remotely through QuickServer.

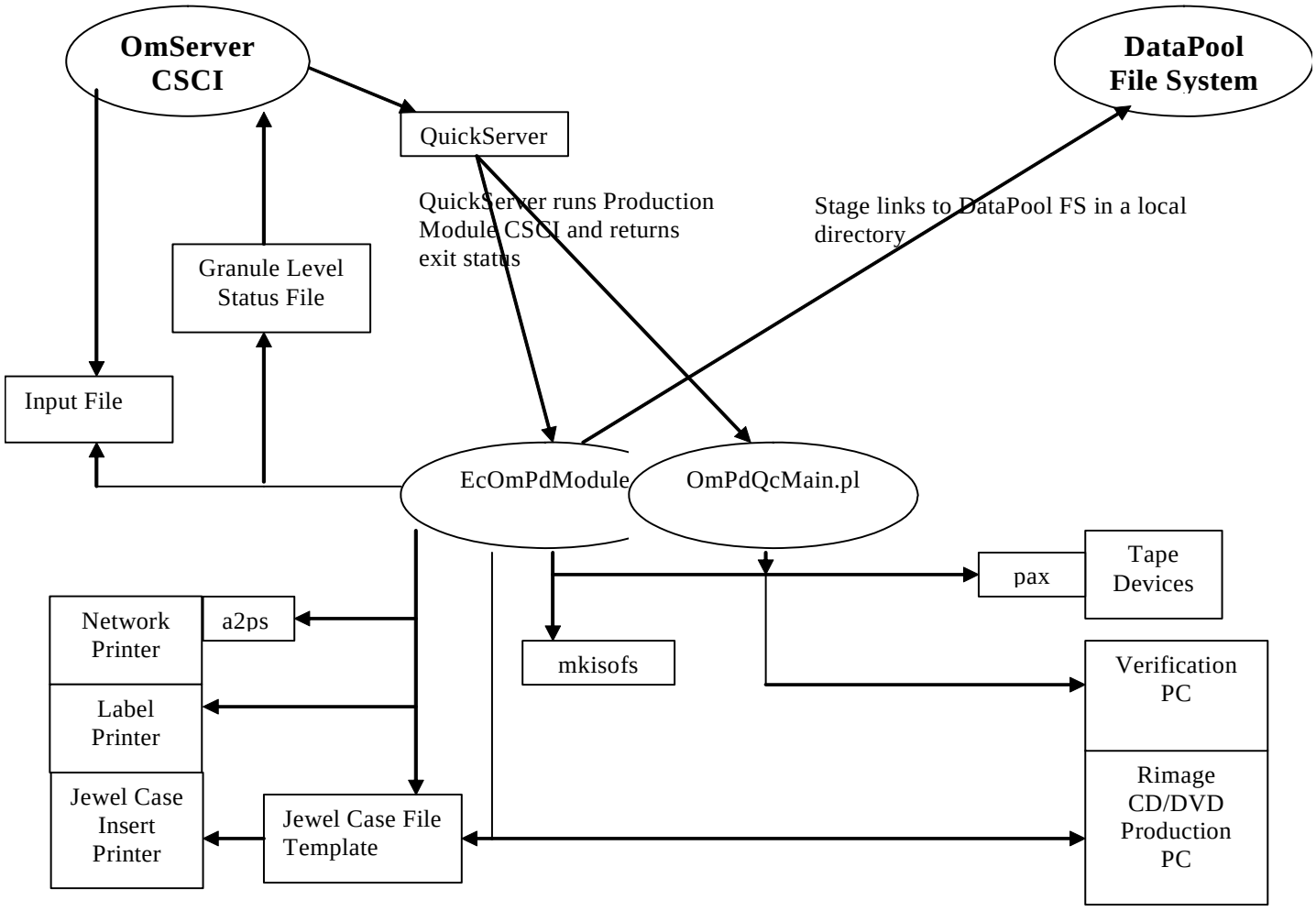


Figure 4.5-5. Production Module CSCI Architecture Diagram

4.5.1.10 Production Module CSCI Process Description

Table 4.5-7 provides descriptions of the processes shown in the Production Module Architecture Diagram.

Table 4.5-7. Production Module CSCI Process

Process	Type	Hardware CI	COTS/ Developed	Functionality
EcOmPdModule	driver	DIPHW	Developed	Receives instructions for physical media processing from Order Manager through a .PPF file. Also it prints the tape label and the jewel case insert.
OmPdQcMain.pl	Perl script	DIPHW	Developed	This script interfaces with the tape devices, the Unix /CDROM directory and a script running on the QC PC to verify filenames and sizes written to physical media.
OmPdNtQC.pl	Perl script	DIPHW	Developed	Runs on the QC PC and polls the Unix drive for the arrival of a signal file. It then does a listing of the PC's CD/DVD drive and puts the result in a file on the unix box where it can be seen by OmPdQcMain.pl.
OmPdCleanup.pl	Perl script	DIPHW	Developed	A script run by Order Manager through QuickServer which deletes leftover request files not needed for logging or later reference.
OmPdCleanupGUI	tcl GUI	DIPHW	Developed	Directs periodic removal or archive of remaining request and log files through an entry in the crontab.
EcOmPdPrintJCIFile.pl	Perl script	DIPHW	Developed	Inserts granule and request data into the Jewel Case Insert template and sends the new file to the Jewel Case printer.
mkisofs	process	DIPHW	COTS	Creates an ISO image file for burning to a CD/DVD that can be read by PC or UNIX operating systems.
a2ps	printer driver	DIPHW	COTS	Facilitates the formatted printing of the Packing List and the QC reports.
Jewel Case Insert template	file	DIPHW	COTS	Postscript format file containing images and place holders for granule and request data.

EBIS Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the HWCI, and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

4.5.1.11 Production Module CSCI Interface Description

Table 4.5-8 provides descriptions of the interface events shown in the Order Manager Server CSCI architecture diagram.

Table 4.5-8. Production Module CSCI Interface Events

Event	Event Frequency	Interface	Intiated By	Event Description
Media Preparation	Once per request volume	QuickServer	Order Manager	Order Manager submits a RequestId_Vol.PPF file across NFS mount containing all non-configuration data needed for physical media processing.
Media Creation	Once per request volume	QuickServer	Order Manager	Order Manager submits a RequestId_Vol.PPF file across NFS mount containing all non-configuration data needed for physical media processing.
Verification	Once per request volume	QuickServer	Order Manager	Order Manager submits a command containing the requestId and the volume number.
Cleanup	Once per order	QuickServer	Order Manager	Order Manager submits a command containing the requestId.

4.5.1.12 Data Stores

The Production Module CSCI receives all needed data from a configuration file and an input file from Order Manager. It does not interface with a relational database.

4.5.1.13 Production Module Hardware

The Production Module Hardware is not shared with any other subsystem.

4.6 Planning Subsystem Overview

The Planning Subsystem (PLS) manages the data production activities at EMD sites in support of the operations staff by providing the following capabilities:

- Identifies the data processing tasks (via data processing requests) performed by a site.
- Generates the data production plans for scheduling the identified processing tasks according to different production rules, which define how a particular Product Generation Executable (PGE) is to be run.
- Coordinates data production with the DSS and the DPS to achieve an automated production system.

Planning Subsystem Context Diagram

Figure 4.6-1 is the context diagrams for the PLS. The diagrams show the events sent to other SDPS and CSMS subsystems and the events the PLS receives from other SDPS and CSMS subsystems. Table 4.6-1 provides descriptions of the interface events shown in the Planning Subsystem Context Diagrams.

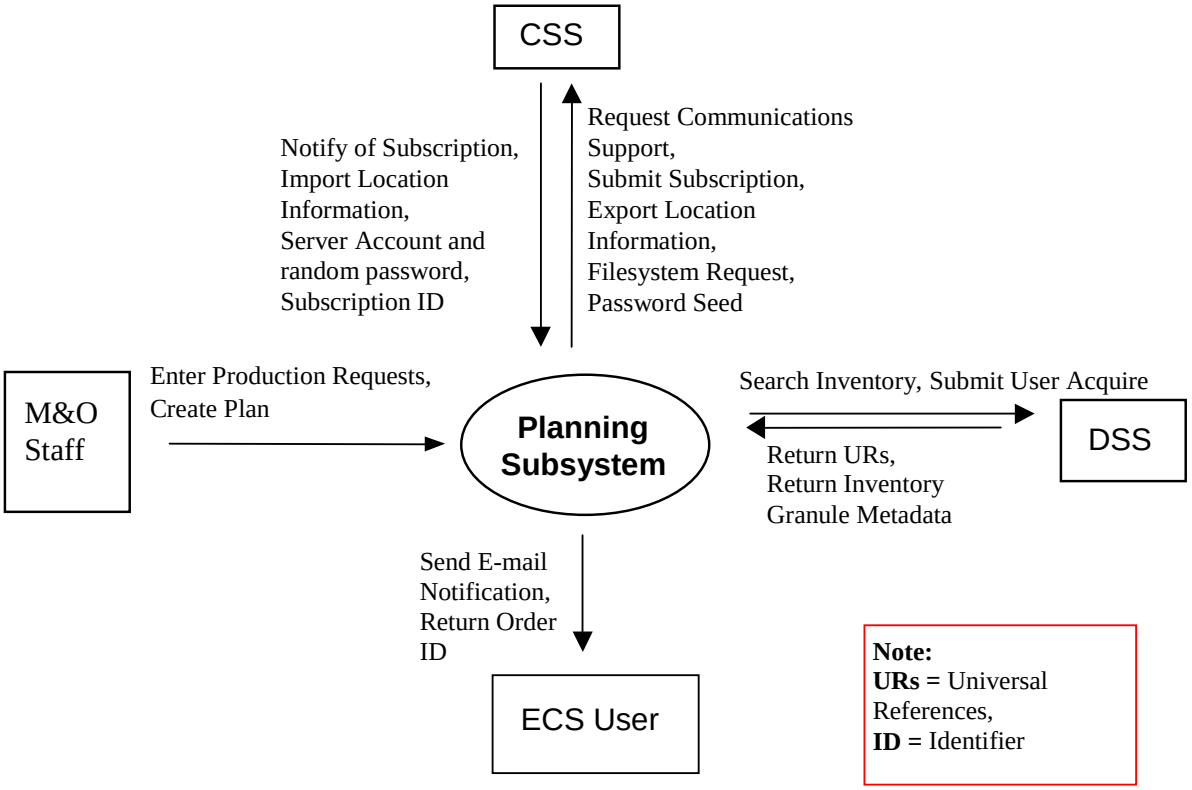


Figure 4.6-1. Planning Subsystem Context Diagram

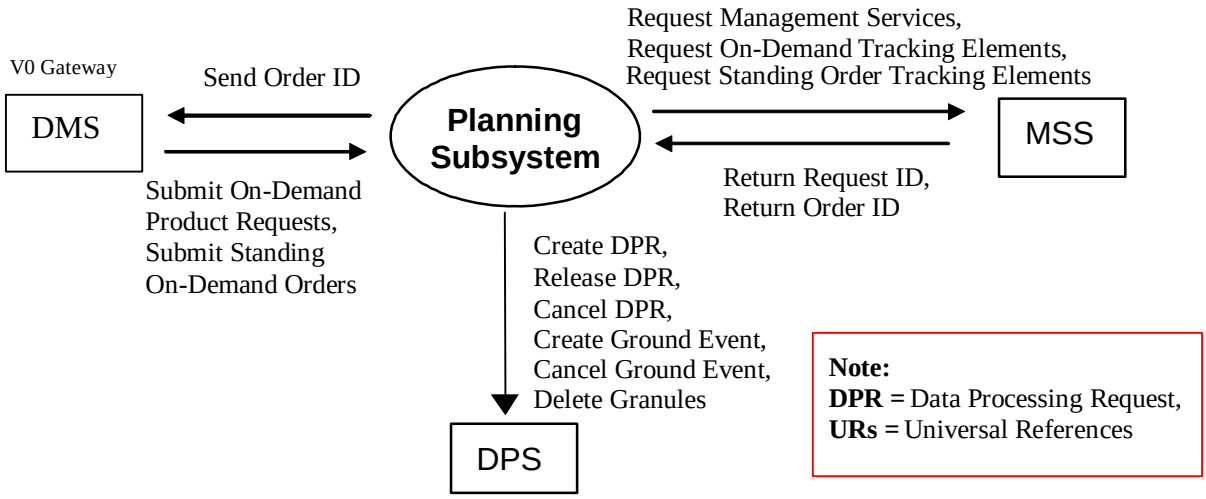


Figure 4.6-1. Planning Subsystem Context Diagram (cont.)

Table 4.6-1. Planning Subsystem Interface Events (1 of 3)

Event	Interface Event Description
Request Communications Support	The **CSS** provides a library of services available to each SDPS and CSMS subsystem. The subsystem services required to perform specific assignments are requested from the CSS. These services include: • CCS Middleware Support • Database Connection Services • Network & Distributed File Services • Name/Address Services • Password Services • Server Request Framework (SRF) • Universal Reference (UR) • Error/Event Logging • Message Passing • Fault Handling Services • Mode Information • Query Registry - Retrieving the requested configuration attribute-value pairs from the Configuration Registry
Submit Subscription	The PLS creates a subscription, sent to the **CSS**, using the advertisement for subscribing to an insert event for an ESDT. In response, PLS receives a corresponding subscription identifier.

Table 4.6-1. Planning Subsystem Interface Events (2 of 3)

Event	Interface Event Description
Export Location Information	The PLS stores physical and logical server location information in the **CSS**.
Filesystem Request	The PLS requests EMD files and directories via an established mount point from the **CSS**. The CSS makes the storage device(s) and its data accessible for use by clients.
Password Seed	The PLS requests an account and provides a password to the **CSS**.
Search Inventory	The PLS sends inventory search or inspect requests to the **DSS** to search the SDPS inventory/archives (granules). In response, the PLS receives URs for the respective granules satisfying the search.
Submit User Acquire	The PLS submits an acquire command to the **DSS** on behalf of the user. The user gets a response via the DSS upon data distribution.
Return URs	The PLS receives Earth Science Data Type (ESDT) Universal References (URs) for the granules from the **DSS**.
Return Inventory Granule Metadata	The PLS receive the inventory granule metadata identifying the scene within the granule based on an inventory search request sent to the **DSS**.
Send E-mail Notification	The PLS sends e-mail notification to the **user** when a standing order granule arrives and when a standing order expires.
Return Order ID	The PLS sends an order id to the **user** to track his on-demand product or order requests.
Enter Production Requests	The **operator** request production by selecting a PGE type and the time duration for the PGE to process the input data in the PLS.
Create Plan	The **operator** issue commands to control the creation and activation of a data production plan to the PLS.
Notify of Subscription	A message passing callback in the PLS subscription manager is called, by the **CSS**, with the UR of the granule inserted into the Data Server as one of the calling parameters.
Import Location Information	The PLS requests server location information from the **CSS**.
Server Account and random password	The PLS receives an account and random password from the **CSS** after providing a password seed to establish an account.
Subscription ID	The PLS receives a subscription identifier from the **CSS** after submitting a subscription.
Request Management Services	**System startup and shutdown -** Please refer to the release-related, current version of the Mission Operations Procedures for the EMD Project document (611) and the current EMD Project Training Material document (625), identified in Section 2.2.1 of this document.
Request On-Demand Tracking Elements	The PLS sends product requests to the **MSS** for on-demand orders.
Request Standing Order Tracking Elements	The PLS sends requests to the **MSS** for standing orders of ASTER high level on-demand products.

Table 4.6-1. Planning Subsystem Interface Events (3 of 3)

Event	Interface Event Description
Return Request ID	The PLS receives the request identifier from the **MSS** for on-demand tracking elements requests.
Return Order ID	The PLS receives the order identifier from the **MSS** for standing order tracking elements requests.
Create DPR	The PLS sends, to the **DPS**, the Data Processing Request Identification (dprId) and whether the DPR is waiting for external input data.
Release DPR	The PLS sends the dprId to the **DPS** for DPR release.
Cancel DPR	The PLS sends a request to cancel the dprId to the **DPS** for the deletion of a DPR.
Create Ground Event	The PLS sends the ground event id, resource id, and start time to the **DPS** to create a ground event to perform maintenance activities on data processing resources.
Cancel Ground Event	The PLS sends the ground event id, resource id, and start time to the **DPS** to cancel a ground event.
Delete Granules	The PLS sends requests to the **DPS** to delete granules associated with cancelled DPRs.
Submit Standing On-Demand Orders	The PLS receives standing on-demand orders from the **DMS**.
Send Order ID	The PLS returns the order identifier to the **DMS**.

The following paragraphs describe the relationships between the PLS and other SDPS subsystems.

DPS Interface

The PLS uses a database link with the DPS Processing CSCI to describe the Product Generation Executable (PGEs) needed to fulfill the production goals. A Data Processing Request (DPR) describes a PGE run to the DPS. A DPR describes the specific input granules, output filenames, and run-time parameters for a PGE, as well as dependencies and predicted run-times. The DPS provides status and processing completion information to the PLS.

DSS Interface

The PLS queries the DSS inventory for data required for processing. If the data exists, the DSS responds to the PLS with granule information (identification, metadata, and location).

CSS Interface

The CSS Subscription server provides a notification on the arrival of EMD data. The PLS exchanges mode management information with and receives event notifications from the CSS.

MSS Interface

The PLS sends fault management, accounting, security, and performance data to the MSS for logging. The PLS receives Order tracking information for On-Demand Products and standing orders.

DMS Interface

The PLS submits non-standard L1B orders through the DMS (EMD to GDS Gateway). The PLS receives requests for On-Demand Products from DMS (from EDG to V0 GWTY to PLS).

Planning Subsystem Structure

The PLS is comprised of one CSCI, Production Planning (PLANG CSCI) and one hardware CI, Production Planning (PLNHW).

The Planning and Data Processing Subsystems (PDPS) database resides in the PLNHW and serves both planning and scheduling activities.

Use of COTS in the Planning Subsystem

- Hughes- Delphi Scheduling Class Libraries.

  The Delphi Scheduling Class Libraries are used to schedule the Resource Planning Workbench and the Production Planning Workbench. Delphi uses C++ classes to provide user-oriented, integrated, and modular planning and scheduling software utilities.

- RogueWave's Tools.h++

  The Tools.h++ class libraries provide libraries of object strings and collections. These libraries are delivered statically linked with the custom code delivery.

- RogueWave's DBTools.h++

  The DBTools.h++ C++ class libraries interact with the Sybase ASE database Structured Query Language (SQL) server and buffer the processes from the relational database used. These libraries are delivered statically linked with the custom code delivery.

- ICS' Builder Xcessory

  The Builder Xcessory GUI builder tool modifies displays. The Builder Xcessory generates the C++ code to produce the Maintenance Tool (Mtool) display at run time. There is no operational component of Builder Xcessory needed at run-time.

- Sybase Adaptive Server Enterprise (ASE)

  The Sybase ASE provides the capabilities to read, insert, update and delete PDPS database content. The Sybase ASE must be operational during the PLS operations.

- CCS Middleware Client

  CCS Middleware Client provides PLS with communications between other subsystems. CCS Middleware can reside on one or both sides of the interface. An instance must be

installed on the platform where PLS resides. Although the CCS Middleware Client is part of CSS, this COTS product must be installed for PLS to run in the SDPS operational and test environment.

4.6.1 Production Planning (PLANG) Software Description

4.6.1.1 Production Planning Functional Overview

The PLANG CSCI manages the data production activities at each site by providing the operator with the following capabilities:

- Defining the data processing tasks (via data processing requests) to perform at the site.
- Generating data production plans for scheduling processing and reprocessing tasks.
- Coordinating data production with the DSS and the DPS to automate the production system.

If the order is submitted from the EDG, the OrderId and RequestId will be passed in as additional parameters to the ODPRM. Currently, Standing Orders cannot be submitted from the EDG.

If the order is a DEM, the ODPRM simply creates the associated dummy DPRs and granules and returns a MSS order ID. Once the request is completed the operator inserts the DEM into the SDSRV and invokes the command line tool again to update the user

If the order is for a higher-level product, the ODPRM creates the PRs and DPRs necessary to fulfill the order. If all inputs are available, the data processing request(s) are submitted to the Data Processing Subsystem and the product is produced. As each DPR changes state, the MSS Order Tracking GUI is updated to reflect the new state and the overall order status is updated accordingly.

If the order is a standing order associated with a DAR, ODPRM stores the product and the DAR information associated with this order in the PDPS database. As granules associated with the DAR arrive, ODPRM automatically generates on demand PRs and DPRs, which follow the same production process as ordinary ASTER high level on demand requests.

4.6.1.2 Production Planning Context

Figure 4.6-2 is the PLANG CSCI context diagrams. The diagrams show the events sent to the PLANG CSCI and the events the PLANG CSCI sends to other CSCIs and the production operators. Table 4.6-2 provides descriptions of the interface events shown in the PLANG CSCI context diagrams.

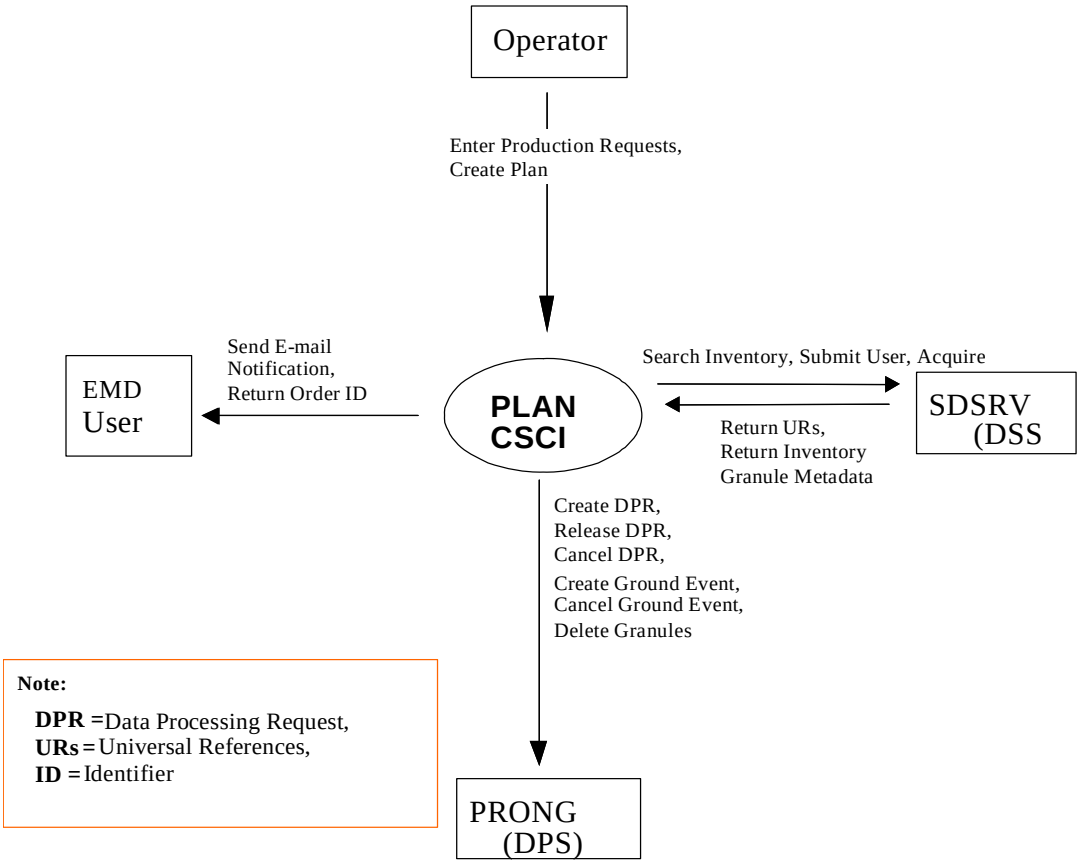


Figure 4.6-2. PLANG CSCI Context Diagram

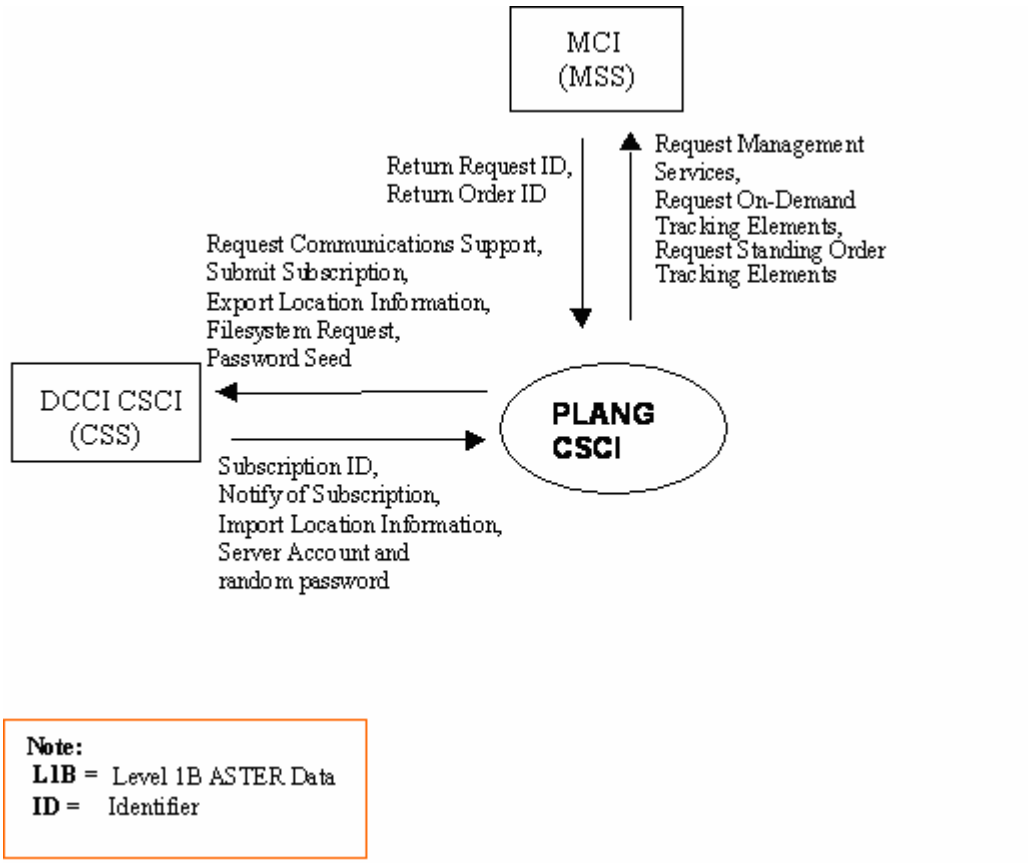


Figure 4.6-2. PLANG CSCI Context Diagram (cont.)

Table 4.6-2. PLANG CSCI Interface Events (1 of 3)

Event	Interface Event Description
Enter Production Requests	The **operator** request production by selecting a PGE type and the time duration for the PGE to process the input data in the PLANG CSCI.
Create Plan	The **operator** issues commands to control the creation and activation of a data production plan to the PLANG CSCI.
Search Inventory	The PLANG CSCI sends inventory search or inspect requests to the **SDSRV CSCI** to search the EMD inventory/archives (granules). In response, the PLANG CSCI receives granule URs satisfying the search.
Submit User Acquire	The PLANG CSCI submits an acquire command to the **SDSRV CSCI** on behalf of the user. The user gets a response via the DDIST CSCI upon data distribution.
Return URs	The PLANG CSCI receives Earth Science Data Type (ESDT) Universal References (URs) for the granules from the **SDSRV CSCI**.

Table 4.6-2. PLANG CSCI Interface Events (2 of 3)

Event	Interface Event Description
Return Inventory Granule Metadata	The PLANG CSCI receives the inventory granule metadata identifying the scene within the granule based on an inventory search request sent to the **SDSRV CSCI**.
Create DPR	The PLANG CSCI sends the Data Processing Request Identification (dprId) and whether the DPR is waiting for external input data to the **PRONG CSCI**.
Release DPR	The PLANG CSCI sends the dprId to the **PRONG CSCI**.
Cancel DPR	The PLANG CSCI sends a request to cancel the dprId to the **PRONG CSCI** for the deletion of a DPR.
Create Ground Event	The PLANG CSCI sends the ground event id, resource id, and start time to the **PRONG CSCI** to create a ground event to perform maintenance activities on data processing resources.
Cancel Ground Event	The PLANG CSCI sends the ground event id, resource id, and start time to the **PRONG CSCI** to delete a ground event.
Delete Granules	The PLANG CSCI sends requests to the **PRONG CSCI** to delete granules associated with cancelled DPRs.
Send E-mail Notification	The PLANG CSCI sends e-mail notification to the **user** when a standing order granule arrives and when a standing order expires.
Return Order ID	The PLANG CSCI sends an order id to the **user** to track his on-demand product or order requests.
Request Management Services	**System startup and shutdown -** Please refer to the release-related, current version of the Mission Operations Procedures for the EMD Project document (611) and the current EMD Project Training Material document (625), identified in Section 2.2.1 of this document.
Request On-Demand Tracking Elements	The PLANG CSCI sends product requests to the **MCI** for orders.
Request Standing Order Tracking Elements	The PLANG CSCI sends requests to the **MCI** for standing orders of ASTER high-level products.
Subscription ID	The PLANG CSCI receives a subscription identifier from the **DCCI CSCI** after submitting a subscription.
Notify of Subscription	A message passing callback in the PLANG CSCI subscription manager is called, by the **DCCI CSCI**, with the granule UR inserted into the SDSRV inventory as a calling parameter.
Import Location Information	The PLANG CSCI requests server location information from the **DCCI CSCI**.
Server Account and random password	The PLANG CSCI receives an account and random password from the **DCCI CSCI** after providing a password seed to establish an account.

Table 4.6-2. PLANG CSCI Interface Events (3 of 3)

Event	Interface Event Description
Request Communications Support	The **DCCI CSCI** provides a library of services available to each SDPS and CSMS CSCI. The CSCI services required to perform specific assignments are requested from the DCCI CSCI. These services include: • CCS Middleware Support • Database Connection Services • Network & Distributed File Services • Name/Address Services • Password Services • Server Request Framework (SRF) • Universal Reference (UR) • Error/Event Logging • Message Passing • Fault Handling Services • Mode Information • Query Registry - Retrieving the requested configuration attribute-value pairs from the Configuration Registry
Submit Subscription	The PLANG CSCI submits a subscription for an ESDT insert event to the **DCCI CSCI**. In response, the PLANG CSCI receives a subscription identifier.
Export Location Information	The PLANG CSCI stores physical and logical server location information in the **DCCI CSCI**.
Filesystem Request	The PLANG CSCI requests EMD files and directories via an established mount point from the **DCCI CSCI**. The DCCI CSCI makes the storage device(s) and its data accessible for use by clients.
Password Seed	The PLANG CSCI requests an account and provides a password to the **DCCI CSCI**.

PLANG CSCI interfaces include:

PDPS Database Interface (Common database pseudo-interface with DPS)

The DPS Algorithm Integration and Test Tools (AITTL) CSCI stores PGE data, which is retrieved by the PLS. This PGE data includes the PGE executable, the input data type(s) it requires, the output data type(s) it generates, and the resource requirements (e.g., hardware platform, memory, and disk storage). The PGE data is used by the PLS to schedule data processing requests with the DPS.

The PLS manages the database space by deleting DPRs from the PDPS database and some of its associated granules not used by other DPRs.

Operator Interface

The production operators enter Production Requests into the PLS via the Planning User Interface. Production Requests provide the information necessary for data to be produced by the DPS. Production Requests are used to process new data (Routine Production Requests) or for reprocessing data (Reprocessing Production Requests). The PLS uses the PGE profile information from the Production Requests to generate the DPRs needed to fulfill the request for

data. The Planning User Interface also issues commands to initiate plan creation, plan activation and plan cancellations, and provide reports and status of plan progress. The production operators perform resource planning for the entire DAAC through the Planning User Interface with awareness of the impact of ground events on data processing resources.

4.6.1.3 Production Planning Architecture

Figure 4.6-3a is the PLANG CSCI architecture diagrams without the On-Demand Manager included. The diagrams show the events sent to the PLANG CSCI processes and the events sent by the PLANG CSCI processes to other processes. Figure 4.6-3b is the PLANG CSCI architecture diagrams with the On-Demand Manager featured. The diagrams show the events sent to the PLANG CSCI processes and the events sent by the PLANG CSCI processes to other processes.

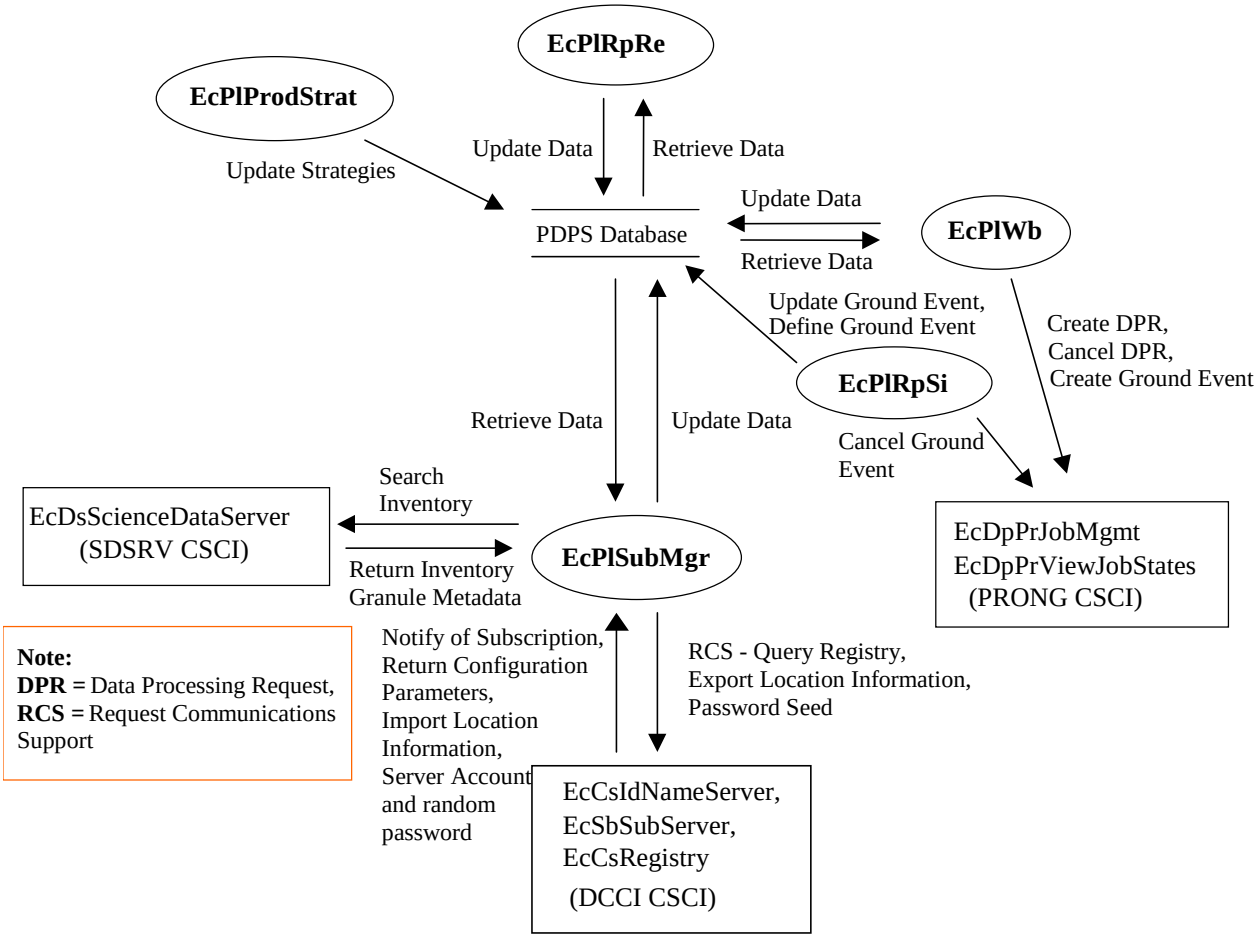


Figure 4.6-3a. PLANG CSCI Architecture Diagram

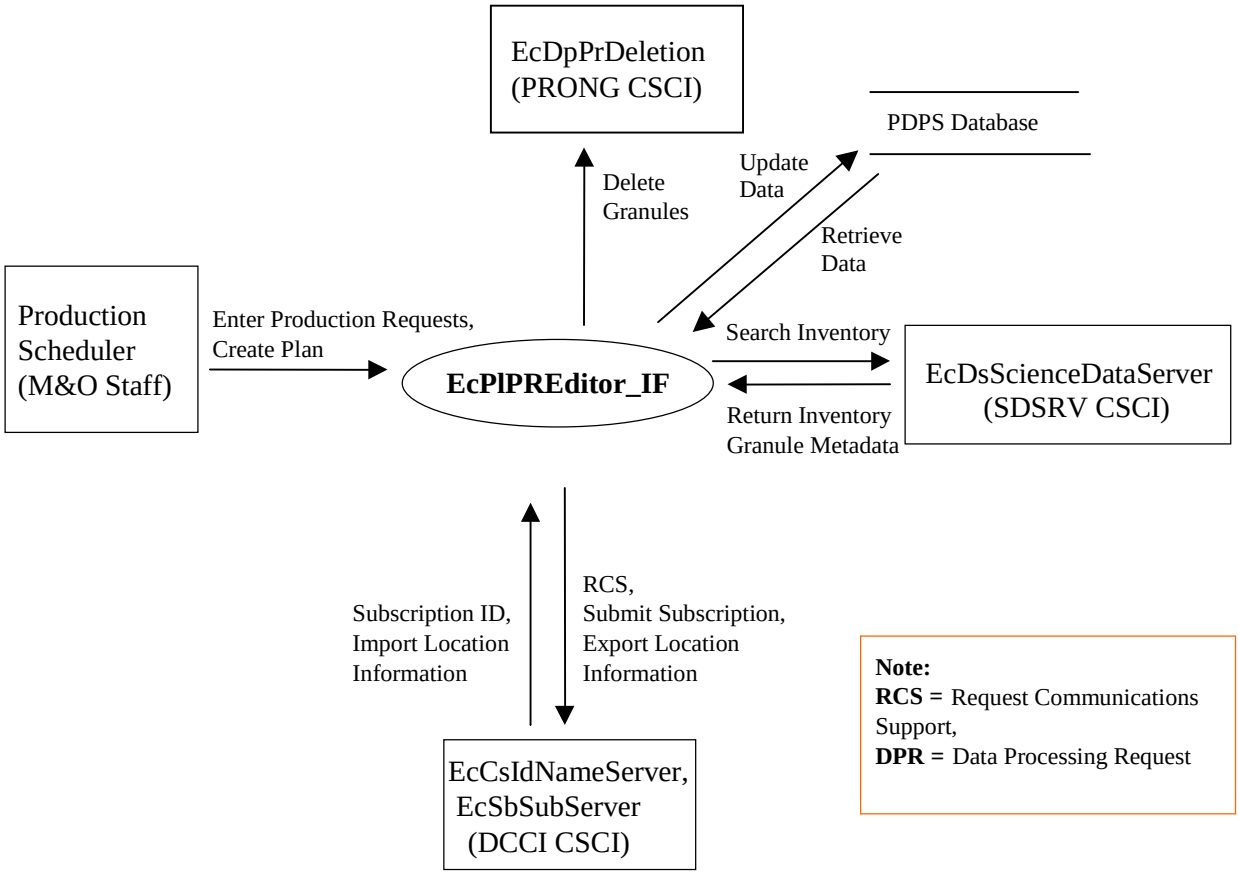


Figure 4.6-3a. PLANG CSCI Architecture Diagram (cont.)

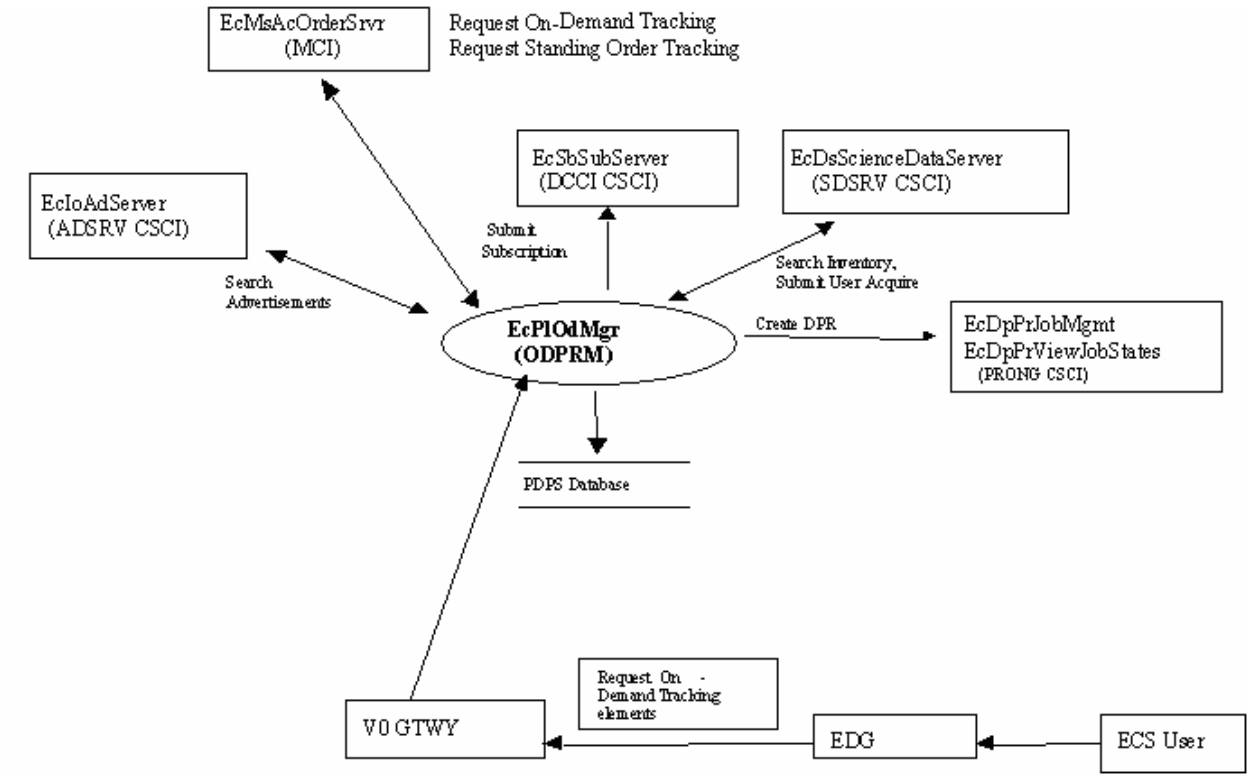


Figure 4.6-3b. PLANG CSCI Architecture Diagram

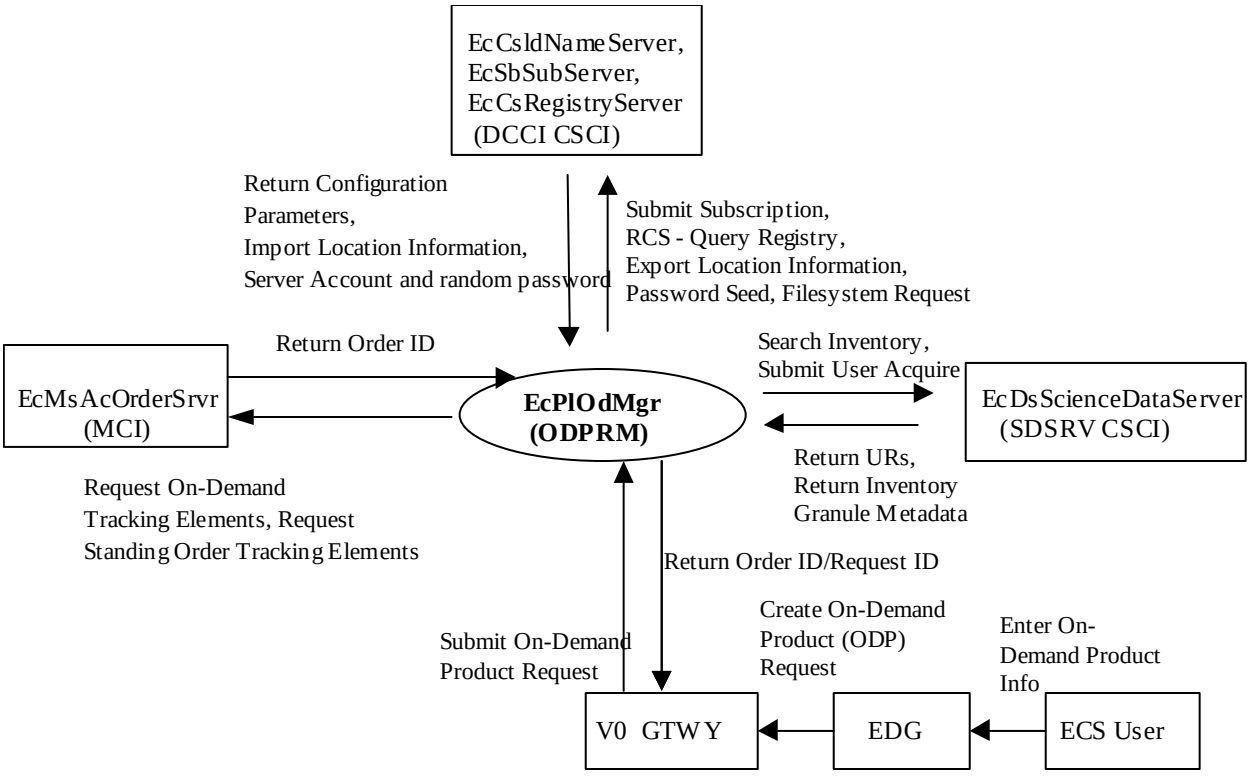


Figure 4.6-3b. PLANG CSCI Architecture Diagram (cont.)

4.6.1.4 Production Planning Process Descriptions

Table 4.6-3 provides descriptions of the Production Planning processes shown in the PLANG CSCI architecture diagrams.

Table 4.6-3. PLANG CSCI Processes (1 of 2)

Process	Type	Hardware CI	COTS / Developed	Functionality
EcPlRpRe	GUI	PLNHW	Developed code using Delphi Class Libraries.	The Resource Planning Workbench prepares a schedule for the resources at each respective site, and forecasts the start and completion times of the ground events and the impact on the resources used within the schedule. The workbench allows the production operator to: • Edit the resources currently available at a site • Associate the resources with production strings (logical groupings of resources used by AutoSys and Data Processing) when allocating resources for a particular PGE.
EcPlWb	GUI	PLNHW	Developed code using Delphi Class Libraries.	The Production Planning Workbench prepares a schedule for the production at a site, and forecasts the start and completion times of the activities within the schedule. Specifically, the Workbench allows: • Candidate Plan Creation—from the production requests prepared by the Production Request Editor • Plan Activation—activating a candidate plan • Update of the Active Plan—feedback from the DPS activities are incorporated into the active plan • Cancellation/Modification of the Active Plan Activating a plan entails rolling a portion of a selected plan into the AutoSys COTS via the DPS. The “schedule” is managed within the DPS. The forecast times generated by the planner are used to set up operator alerts to gross departures from the predicted schedule. Ground Events are sent to the DPS via the EcPlWb.
EcPlPREditor_IF	GUI	PLNHW	Developed	The Planning User Interface (Production Request Editor) allows the production operator to submit production requests to describing the data products to generate. The production request uses the PGE descriptions (profiles) entered during Algorithm Integration and Test (AI&T) to define the Data Processing Requests. The request adds, modifies, and deletes Production Requests, and reviews and modifies the resulting Data Processing Requests. The user specifies rules for producing the individual DPRs for the reprocessing requests. The production request editor is a distinct application and separate from the workbench because defining a production request is unrelated to the planning of a production request.
EcPlRpSi	GUI	PLNHW	Developed code using Delphi Class Libraries	The Planning Resource GUI allows the operator to: • Define or cancel ground events (maintenance, etc.) on the allocated resources

Table 4.6-3. PLANG CSCI Processes (2 of 2)

Process	Type	Hardware CI	COTS / Developed	Functionality
EcPlSubMgr	Server	SPRHW	Developed	The Subscription Manager receives subscription notifications from the EcDsScienceDataServer via the EcSbSubServer. Subscription notification notifies planning of the arrival of required input data. The Subscription Notification is handled through the Infrastructure message passing service and contains URs pointing to the data objects stored in the EcDsScienceDataServer. The Subscription Manager updates the PDPS database when data is available. When all input data for a DPR is available, the job defined for that DPR is released within the DPS.
EcPlProdStrat	GUI	PLNHW	Developed	The Production Strategies GUI is used to create a set of planning priorities to be applied to each DPR in a plan. This strategy takes user, PGE type, PGE instance, and Production Request priorities into account. This strategy is then saved to the PDPS database.
EcPlOdMgr	Server	SPRHW	Developed	The On-Demand manager (ODPRM) receives requests for data from the scientist via the EDG. The scientist can request a DEM, higher-level product or a Standing Order through the ODFRM. Through the EDG, a scientist can request a DEM or a higher-level product. Once the ODPRM has generated the Production Request (PR) necessary to fulfill the request, in the case of the higher-level product, it submits the PR to DPS for processing. Once DPS is finished, the ODPRM distributes the data to the scientist who requested it.

EMD Baseline Information System (EBIS) Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the HWCI, and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

4.6.1.5 Production Planning Process Interface Descriptions

Table 4.6-4 provides descriptions of the interface events shown in the PLANG CSCI architecture diagrams without the On-Demand Manager included. Table 4.6-4 provides descriptions of the interface events shown in the PLANG CSCI architecture diagrams with the On-Demand Manager featured (starting at table 12 of 19).

Table 4.6-4. PLANG CSCI Process Interface Events (1 of 19)

Event	Event Frequency	Interface	Initiated By	Event Description
Retrieve Data	One per retrieve request	PDPS Database	*Process:* EcPlRpRe *Library:* PlRpRe *Classes:* PlRpReAutosysWin, PlRpReComputerWin, PlRpReDiskWin, PlRpReHardwareWin, PlRpReRealComputerWin, PlRpReStringWin, PlRpReWin *Process:* EcPlWb *Library:* plwb *Class:* PlWbScheduler *Process:* EcPlSubMgr Library: PlCore1 Classes: PlDataGranule, PlProductionRequest	The EcPlRpRe, EcPlWb and EcPlSubMgr processes send requests to the **PDPS database** to retrieve information to allow the PGE to be scheduled and executed.

Table 4.6-4. PLANG CSCI Process Interface Events (2 of 19)

Event	Event Frequency	Interface	Initiated By	Event Description
Update Data	One update per request	PDPS Database	*Process:* EcPlWb *Library:* plwb *Class:* PlWbScheduler *Process:* EcPlSubMgr Library: PlCore1 Classes: PlDataGranule, PlProductionRequest *Process:* EcPlRpRe *Library:* PlRpRe *Classes:* PlRpReAutosysWin, PlRpReComputerWin, PlRpReDiskWin, PlRpReHardwareWin, PlRpReRealComputerWin, PlRpReStringWin, PlRpReWin	The EcPlWb, EcPlSubMgr and EcPlRpRe processes send requests to the **PDPS database** to update granule information (location, size, etc.), processing status and check pointing.
Update Ground Event	Once per defined ground event	PDPS Database	*Process:* EcPlRpSi *Library:* PlRpSi *Classes:* PlRpSiScheduler, PlRpSiWinAbs	The EcPlRpSi also updates ground event information in the **PDPS database**.

Table 4.6-4. PLANG CSCI Process Interface Events (3 of 19)

Event	Event Frequency	Interface	Initiated By	Event Description
Define Ground Event	One per request	PDPS Database	*Process:* EcPlRpSi *Library:* PlRpSi *Classes:* PlRpSiScheduler, PlRpSiWinAbs	The EcPlRpSi provides the information to identify the ground events to be done and sends the information to the **PDPS database**.
Create DPR	One per list of predecessor DPRs	*Process:* EcDpPrJobMgmt *Library:* DpPrJM *Class:* DpPrScheduler	*Process:* EcPlWb *Class:* PlWbScheduler	The EcPlWb process sends the dprId and whether the DPR is waiting for external data to the **EcDpPrJobMgmt** process to create a job in the data production process.
Cancel DPR	One per dprId send	*Process:* EcDpPrJobMgmt *Library:* DpPrJM *Class:* DpPrScheduler	*Process:* EcPlWb *Library:* PlCore1 *Class:* PlDpr	The EcPlWb process sends a request to cancel the dprId to the **EcDpPrJobMgmt** process for the deletion of a DPR.
Create Ground Event	One per defined ground event	*Process:* EcDpPrJobMgmt *Library:* DpPrJM *Class:* DpPrScheduler	*Process:* EcPlWb *Library:* plwb *Class:* PlWbScheduler	The EcPlWb process sends the ground event id, resource id, and start time to the **EcDpPrJobMgmt** process to perform maintenance activities on data processing resources.
Cancel Ground Event	One per defined ground event	*Process:* EcDpPrJobMgmt *Library:* DpPrJM *Class:* DpPrScheduler	*Process:* EcPlRpSi *Library:* PlRpSi *Classes:* PlRpSiScheduler, PlRpSiWinAbs	The EcPlRpSi sends a request to cancel a ground event to the **EcDpPrJobMgmt** process for the deletion of a ground event.

Table 4.6-4. PLANG CSCI Process Interface Events (4 of 19)

Event	Event Frequency	Interface	Initiated By	Event Description
Request Communications Support	One per process request.	*Process:* EcCsIdNameServer *Libraries:* EcPf, Middleware, FoNs, FoIp, oodce *Classes:* EcPfManagedServer, EcPfClient, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy *Library (Common):* EcUr *Class:* EcUrServerUR *Process:* EcSbSubServer *Library:* EcSbSr *Class*: EcSbSubscription *Library:* EcDcMsgPsng1 *Library:* event *Class:* EcLgErrorMsg *Library:* EcSeCmi Class: EcSeCmi *Process:* EcCsRegistry *Library:* EcCsRegistry *Class:* EcRgRegistryServer_C	*Process:* EcPlSubMgr *Library:* PlCore1 *Classes:* Most PLS Classes	The **DCCI CSCI** provides a library of services available to each SDPS and CSMS CSCI. The CSCI services required to perform specific assignments are requested from the DCCI CSCI. These services include: • CCS Middleware Support • Database Connection Services • Network & Distributed File Services • Name/Address Services • Password Services • Server Request Framework (SRF) • Universal Reference (UR) • Error/Event Logging • Message Passing • Fault Handling Services • Mode Information • Query Registry - Retrieving the requested configuration attribute-value pairs from the Configuration Registry

Table 4.6-4. PLANG CSCI Process Interface Events (5 of 19)

Event	Event Frequency	Interface	Initiated By	Event Description
Export Location Information	Once at system startup and after each failure recovery	*Process:* EcCsIdNameServer *Libraries:* EcPf, Middleware, FoNs, FoIp, oodce *Classes:* EcPfManagedServer, EcPfClient, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy	*Process:* EcPlSubMgr *Library:* PlCore1 PlCore1IF PlCore1Sub PlCore2 PlSSIT PlUtil *Classes:* PlSubscriptionManagerProcess.C, PlSubscriptionManager.C	The EcPlSubMgr stores physical and logical location information in the **EcCsIdNameServer**.
Password Seed	One per password seed	*Process:* Cryptographic Management Interface *Library:* EcSeCmi *Class:* EcSeCmi	*Processes:* EcPlSubMgr, EcPlRpRe, EcPlRpRm, EcPlRpSi, EcPlRpT1, EcPlWb	The server provides a unique number as a seed for generating a password to the **EcPlSubMgr,** EcPlRpRe, EcPlRpRm, EcPlRpSi, EcPlRpT1 **and EcPlWb** processes.
Notify of Subscription	One per message passing callback	*Process:* EcPlSubMgr *Class:* PlSubMsgCb	*Process:* EcSbSubServer *Library:* EcSbSr *Class:* EcSbSubscription	The **EcSbSubServer** calls a message passing callback in the Subscription Service, with the granule UR inserted into the data server as a calling parameter, to send notification of a subscription event to the EcPlSubMgr.
Return Configuration Parameters	One set per request	*Process:* EcPlSubMgr *Library:* PlCore1 PlCore1IF PlCore1Sub PlCore2 PlSSIT PlUtil *Classes:* PlSubscriptionManagerProcess.c, PlSubscriptionManager.C	*Process:* EcCsRegistry *Library:* EcCsRegistry Class: EcRgRegistryServer_C	The **EcCsRegistry** returns the attribute-value pairs (configuration parameters) to the EcPlSubMgr upon request.

Table 4.6-4. PLANG CSCI Process Interface Events (6 of 19)

Event	Event Frequency	Interface	Initiated By	Event Description
Import Location Information	As required for processing	*Process:* EcPlSubMgr *Library:* PlCore1 PlCore1IF PlCore1Sub PlCore2 PlSSIT PlUtil *Classes:* PISubscriptionManagerProcess.c, PISubscriptionManager.C	*Process:* EcCsIdNameServer *Libraries:* EcPf, Middleware, FoNs, FoIp, oodce *Classes:* EcPfManagedServer, EcPfClient, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy	The EcPlSubMgr requests server location information from the **EcCsIdNameServer**.
Server Account and random password	One per account and password	*Processes:* EcPlSubMgr, EcPlRpRe, EcPlRpRm, EcPlRpSi, EcPlRpT1, EcPlWb *Library:* PlCore1 PlCore1IF PlCore1Sub PlCore2 PlSSIT PlUtil *Classes:* PISubscriptionManagerProcess.c, PISubscriptionManager.C	*Process:* Cryptographic Management Interface *Library:* EcSeCmi *Class:* EcSeCmi	The Cryptographic Management Interface generates a random password for the account based on the seed provided by the EcPlSubMgr, and EcPlRpRe, EcPlRpRm, EcPlRpSi, and EcPlRpT1 processes.

Table 4.6-4. PLANG CSCI Process Interface Events (7 of 19)

Event	Event Frequency	Interface	Initiated By	Event Description
Search Inventory	One per query	*Process:* EcDsScienceDataServer *Library:* DsCl *Class:* DsClQuery	*Process:* EcPlSubMgr *Library:* DpPrDssIF *Class:* DpPrDSSInterface	The EcPlSubMgr process creates two types of queries. One type only has the ESDT short name and data start and stop times and the other type also includes spatial coordinates. The EcPlSubMgr process creates an ESDT Reference from an UR after receiving a subscription notification or receiving an ESDT reference from a query. The EcPlSubMgr process queries when predicted data is not available. The **EcDsScienceDataServer** returns metadata information about the granule being inspected.
Return Inventory Granule Metadata	All metadata per inventory search	*Process:* EcPlSubMgr *Library:* DpPrDssIF *Class:* DpPrDSSInterface	*Process:* EcDsScienceDataServer *Library:* DsCl *Class:* DsClQuery	The EcPlSubMgr receives the inventory granule metadata identifying the scene within the granule based on an inventory search request sent to the **EcDsScienceDataServer**.
Update Strategies	One per strategy created.	Active strategies screen	Production operator *Process:* EcPlProdStrat *Library:* PlGUI *Class:* PlProdStratActive	The **operator** creates strategies via the EcPlProdStrat process when certain jobs need to be prioritized over others. The strategy is saved by name and later can be read by the EcPlWb to prioritize the DPRs in a plan.

Table 4.6-4. PLANG CSCI Process Interface Events (8 of 19)

Event	Event Frequency	Interface	Initiated By	Event Description
Delete Granules	One set of granules per cancelled DPR	*Process:* EcDpPrDeletion *Class:* DpDeletion	*Process:* EcPlPREditor_IF *Library:* PlCore1 *Classes:* PlWbScheduler, PlDpr	The EcPlPREditor_IF sends requests to the **EcDpPrDeletion** process to delete granules associated with a DPR, which has been cancelled.
Update Data	One per update/retrieve request	PDPS Database	*Process:* EcPlPREditor_IF *Library:* PlCore1 *Class:* PlDataType	Requests are sent for updates of granule information (location, size, etc.), processing status, and check pointing from the EcPlPREditor_IF to the **PDPS Database**.
Retrieve Data	One per retrieve request	PDPS Database	*Process:* EcPlPREditor_IF *Library:* PlCore1 *Class:* PlDataType	The EcPlPREditor_IF processes send requests to the **PDPS database** to update/retrieve data defining a PGE. Also, the requests contain information to allow the PGE to be scheduled and executed.
Search Inventory	One per query	*Process:* EcDsScienceDataServer *Library:* DsCl *Class:* DsClQuery	*Process:* EcPlPREditor_IF *Library:* DpPrDssIF *Class:* DpPrDSSInterface	The EcPlPREditor_IF process creates two types of queries. One type only has the ESDT short name and data start and stop times and the other type also includes spatial coordinates. The EcPlPREditor_IF process queries when the predicted data is available. The EcPlPREditor_IF process creates an ESDT Reference from an UR after receiving an ESDT Reference from a query. The **EcDsScienceDataServer** returns ESDT References for granules to satisfy the query.
Return Inventory Granule Metadata	All metadata per inventory search	*Process:* EcPlPREditor_IF *Library:* DpPrDssIF *Class:* DpPrDSSInterface	*Process:* EcDsScienceDataServer *Library:* DsCl *Class:* DsClQuery	The EcPlPREditor_IF receives the inventory granule metadata identifying the scene within the granule based on an inventory search request sent to the **EcDsScienceDataServer**.

Table 4.6-4. PLANG CSCI Process Interface Events (9 of 19)

Event	Event Frequency	Interface	Initiated By	Event Description
Request Communications Support	One per process request.	***Process:*** EcCsIdNameServer ***Libraries:*** EcPf, Middleware, FoNs, FoIp, oodce *Classes:* EcPfManagedServer, EcPfClient, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy ***Library (Common):*** EcUr *Class:* EcUrServerUR ***Process:*** EcSbSubServer *Library:* EcSbSr *Class*: EcSbSubscription ***Library:*** EcDcMsgPsng1 ***Library:*** event *Class:* EcLgErrorMsg ***Library:*** EcSeCmi Class: EcSeCmi ***Process:*** EcCsRegistry *Library:* EcCsRegistry *Class:* EcRgRegistryServer_C	*Process:* EcPlPREditor_IF *Library:* PREGUI PlCore1 PlCore1IF PlCore1Sub PlCore2 PlSSIT PlUtil *Classes:* main_c.cxx, PIPRapp.cxx, PIProdReqEdit.cxx	The **DCCI CSCI** provides a library of services available to each SDPS and CSMS CSCI. The CSCI services required to perform specific assignments are requested from the DCCI CSCI. These services include: • CCS Middleware Support • Database Connection Services • Network & Distributed File Services • Name/Address Services • Password Services • Server Request Framework (SRF) • Universal Reference (UR) • Error/Event Logging • Message Passing • Fault Handling Services • Mode Information • Query Registry - Retrieving the requested configuration attribute-value pairs from the Configuration Registry

Table 4.6-4. PLANG CSCI Process Interface Events (10 of 19)

Event	Event Frequency	Interface	Initiated By	Event Description
Submit Subscription	One per subscription created	*Process:* EcSbSubServer *Library:* EcSbSr *Class:* EcSbSubscription	*Process:* EcPlPREditor *Library*: PlCore1 *Class*: PlDataType	The EcPlPREditor_IF process creates subscriptions using the advertisement for subscribing to an ESDT insert event and enters the subscriptions via the **EcSbSubServer**.
Export Location Information	Once at system startup and after each failure recovery	*Process:* EcCsIdNameServer *Libraries:* EcPf, Middleware, FoNs, FoIp, oodce *Classes:* EcPfManagedServer, EcPfClient, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy	*Process:* EcPlPREditor_IF *Library:* PREGUI PlCore1 PlCore1IF PlCore1Sub PlCore2 PlSSIT PiUtil *Classes:* main_c.cxx, PIPRapp.cxx, PIProdReqEdit.cxx	The EcPlPREditor_IF stores physical and logical location information in the **EcCsIdNameServer**.
Subscription ID	One per subscription request	*Process:* EcPlPREditor *Library*: PlCore1 *Class*: PlDataType	*Process:* EcSbSubServer *Library:* EcSbSr *Class:* EcSbSubscription	The EcPlPREditor_IF receives a subscription identifier from the **EcSbSubServer** after submitting a subscription.

Table 4.6-4. PLANG CSCI Process Interface Events (11 of 19)

Event	Event Frequency	Interface	Initiated By	Event Description
Import Location Information	As required for processing	*Process:* EcPlPREditor_IF *Library:* PREGUI PlCore1 PlCore1IF PlCore1Sub PlCore2 PlSSIT PiUtil *Classes:* main_c.cxx, PIPRapp.cxx, PIProdReqEdit.cxx	*Process:* EcCsIdNameServer *Libraries:* EcPf, Middleware, FoNs, FoIp, oodce *Classes:* EcPfManagedServer, EcPfClient, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy	The EcPlPREditor_IF requests server location information from the **EcCsIdNameServer**.
Enter Production Requests	One per production request	*Process:* EcPlPREditor *Library:* DpPrDssIF *Class:* DpPrDSSInterface	production operator	The production operator request production by selecting a PGE type and the time duration for the PGE to process the input data.
Create Plan	One per created and activated plan	*Process:* EcPlWb *Library:* plwb *Class:* PlWbScheduler	production operator	The production operator creates and activates a data production plan.
Request Management Services	At system startup or shutdown and for restarts	*Processes:* EcPlRpRe, EcPlSubMgr	DAAC unique startup scripts	**System startup and shutdown -** Please refer to the release-related, current version of the Mission Operations Procedures for the EMD Project document (611) and the current EMD Project Training Material document (625), identified in Section 2.2.1 of this document.

Table 4.6-4. PLANG CSCI Process Interface Events (12 of 19)

Event	Event Frequency	Interface	Initiated By	Event Description
Cancel DPR	One per list of predecessor DPRs	*Process:* EcDpPrJobMgmt *Library:* DpPrJM *Class:* DpPrScheduler	*Process:* EcPlOdMgr *Class:* PlPrActivator	The EcPlOdMgr process sends a request to cancel the dprId to the **EcDpPrJobMgmt** process for the deletion of a DPR.
Retrieve Data	One per update/retrieve request	*Process:* EcPlOdMgr *Libraries:* PlCore1, PlOdMgrClient *Class:* PlDataType	PDPS Database	The EcPlOdMgr process receives from the **PDPS database** data defining a PGE and information to allow the PGE to be scheduled and executed.
Enter On-Demand Product Info	One per user request	*Process:* Netscape Communicator Web Browser (COTS)	User	The user fills in the User Information on the Login screen and presses the submit button.
Create On-Demand Product (ODP) Requests	One per user request	*Process:* ODFRM EDG	Process: Netscape Communicator Web Browser (COTS)	The ODFRM receives the On-Demand product information and validates the information.
Pass Validated ODP Request	One per user request	*Process:* Sun One Web Server (COTS)	*Process:* ODFRM	The **Sun One Web Server** spawns the process EcClOdRequest with the User Login information (name and password).
Send Validated ODP Request	One per user request	CGI Interface, *Process:* EcClOdProductRequest *Library:* ClOdCommon *Class:* ClOdProductRequest	Sun One Web Server (COTS) *Class:* PlOrderFactory	The EcClOdProductRequest process accesses the MSS database and sends the user back the Authentication.

Table 4.6-4. PLANG CSCI Process Interface Events (13 of 19)

Event	Event Frequency	Interface	Initiated By	Event Description
Submit On-Demand Product Request	One per user request	*Process:* EcPlOdMgr *Library:* PlOdMgrClient *Class:* PlOdMsgProxy	CGI Interface, *Process:* EcClOdProductRequest *Library:* ClOdCommon *Class:* ClOdProductRequest	The **EcClOdProductRequest** process causes an order to be created in the MSS database and sends the request to the EcPlOdMgr, which sends the user back the Order ID.
Submit Standing On-Demand Orders	One per user request	*Process:* EcPlOdMgr *Library:* PlOdMgrClient *Class:* PlOdMsgProxy	CGI Interface, *Process:* EcClOdProductRequest *Library:* ClOdCommon *Class:* ClOdProductRequest	The **EcClOdProductRequest** process causes an order to be created in the MSS database and sends the request to the EcPlOdMgr, which sends the user back the Order ID.
Send Order ID	One per on-demand product request or standing order	CGI Interface, *Process:* EcClOdProductRequest *Library:* ClOdCommon *Class:* ClOdProductRequest	*Process:* EcPlOdMgr *Library:* PlOdMgrClient *Class:* PlOdMsgProxy	The EcPlOdMgr returns an order id to the CGI Interface to provide to the user to track the order.
Update Data	One per update/retrieve request	PDPS Database	*Process:* EcPlOdMgr *Libraries:* PlCore1, PlOdMgrClient *Class:* PlDataType	The EcPlOdMgr process sends requests to the **PDPS database** to update data defining a PGE. Requests are also sent for updates of granule information (location, size, etc.), processing status, and check pointing.
Submit Subscription	One per subscription created	*Process:* EcSbSubServer *Library:* EcSbSr *Class:* EcSbSubscription	*Processes:* EcPlOdMgr *Library*: PlCore1 *Class*: PlDataType	The EcPlOdMgr processes create subscriptions for an ESDT insert event and enter the subscriptions via the **EcSbSubServer**.

Table 4.6-4. PLANG CSCI Process Interface Events (14 of 19)

Event	Event Frequency	Interface	Initiated By	Event Description
Request Communications Support	One per process request.	***Process:*** EcCsIdNameServer ***Libraries:*** EcPf, Middleware, FoNs, FoIp, oodce *Classes:* EcPfManagedServer, EcPfClient, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy ***Library (Common):*** EcUr *Class:* EcUrServerUR ***Process:*** EcSbSubServer *Library:* EcSbSr *Class*: EcSbSubscription ***Library:*** EcDcMsgPsng1 ***Library:*** event *Class:* EcLgErrormsg ***Library:*** EcSeCmi Class: EcSeCmi ***Process:*** EcCsRegistry *Library:* EcCsRegistry *Class:* EcRgRegistryServer_C	*Process:* EcPlOdMgr *Library:* PlOdMgrClient PlOdMgrClientStub PlCore1 PlCore1IF PlCore1Sub PlCore2 PlUtil PlSSIT *Classes:* PlOdMgr.c, PlOdMsg.c, PlOdMgrClient.c	The **DCCI CSCI** provides a library of services available to each SDPS and CSMS CSCI. The CSCI services required to perform specific assignments are requested from the DCCI CSCI. These services include: • CCS Middleware Support • Database Connection Services • Network & Distributed File Services • Name/Address Services • Password Services • Server Request Framework (SRF) • Universal Reference (UR) • Error/Event Logging • Message Passing • Fault Handling Services • Mode Information • Query Registry - Retrieving the requested configuration attribute-value pairs from the Configuration Registry

Table 4.6-4. PLANG CSCI Process Interface Events (15 of 19)

Event	Event Frequency	Interface	Initiated By	Event Description
Export Location Information	Once at system startup and after each failure recovery	*Process:* EcCsIdNameServer *Libraries:* EcPf, Middleware, FoNs, FoIp, oodce *Classes:* EcPfManagedServer, EcPfClient, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy	*Process:* EcPlOdMgr *Library:* PlOdMgrClient PlOdMgrClientStub PlCore1 PlCore1IF PlCore1Sub PlCore2 PlUtil PlSSIT *Classes:* PlOdMgr.c, PlOdMsg.c, PlOdMgrClient.c	The EcPlOdMgr stores physical and logical location information in the **EcCsIdNameServer**.
Password Seed	One per password seed	*Process:* Cryptographic Management Interface *Library:* EcSeCmi *Class:* EcSeCmi	*Process:* EcPlOdMgr *Libraries:* PlOdMgrClient, PlOdMgrClientStub, PlCore1, PlCore1IF, PlCore1Sub, PlCore2, PlUtil, PlSSIT *Classes:* PlOdMgr.c, PlOdMsg.c, PlOdMgrClient.c	The server provides a unique number as a seed to the Cryptographic Management Interface (within DCCI CSCI) for generating a password to the EcPlOdMgr processes.

Table 4.6-4. PLANG CSCI Process Interface Events (16 of 19)

Event	Event Frequency	Interface	Initiated By	Event Description
Filesystem Request	Once at system startup and after each failure recovery	*Process:* NFS Clients (COTS)	*Process:* EcPlOdMgr *Libraries:* PlOdMgrClient, PlOdMgrClientStub, PlCore1, PlCore1IF, PlCore1Sub, PlCore2, PlUtil, PlSSIT *Classes:* PlOdMgr.c, PlOdMsg.c, PlOdMgrClient.c	The NFS clients (via the EcPlOdMgr) request EMD files or directories via an established mount point. The NFS Server makes the storage device(s) and its data accessible for use by the clients.
Search Inventory	One per query	*Process:* EcDsScienceDataServer *Library:* DsCl *Class:* DsClQuery	*Process:* EcPlOdMgr *Library:* DpPrDssIF *Class:* DpPrDSSInterface	The EcPlOdMgr process creates two types of queries. One type only has the ESDT short name and data start and stop times and the other type also includes spatial coordinates. The EcPlOdMgr process creates an ESDT Reference from an UR after receiving a subscription notification or receiving an ESDT reference from a query. The EcPlOdMgr process queries when predicted data is not available. The **EcDsScienceDataServer** returns metadata information about the granule being inspected.

Table 4.6-4. PLANG CSCI Process Interface Events (17 of 19)

Event	Event Frequency	Interface	Initiated By	Event Description
Submit User Acquire	One per request	*Process:* EcDsScienceDataServer *Library:* DsCl *Classes:* DsClRequest, DsClCommand, DsClESDTReferenceCollector	*Process:* EcPlOdMgr *Library:* PlCore2 *Classes:* PlPrActivator, DpPrDSSInterface	The EcPlOdMgr submits an acquire command to the **EcDsScienceDataServer** on behalf of the user. The user gets a response via the EcDsDistributionServer upon data distribution.
Return URs	Per inventory query	*Process:* EcPlOdMgr *Library:* DpPrDssIF *Class:* DpPrDSSInterface	*Process:* EcDsScienceDataServer *Library:* DsCl *Class:* DsClQuery	The EcPlOdMgr receives Earth Science Data Type (ESDT) Universal References (URs) for the granules from the **EcDsScienceDataServer**.
Return Inventory Granule Metadata	All metadata per inventory search	*Process:* EcPlOdMgr *Library:* DpPrDssIF *Class:* DpPrDSSInterface	*Process:* EcDsScienceDataServer *Library:* DsCl *Class:* DsClQuery	The EcPlSubMgr receives the inventory granule metadata identifying the scene within the granule based on an inventory search request sent to the **EcDsScienceDataServer**.
Request On-Demand Tracking Elements	Per order	*Process:* EcMsAcOrderSrvr *Libraries:* MsAcClnt, MsAcComm	*Process:* EcPlOdMgr *Library:* PlCore2	The EcPlOdMgr requests on-demand tracking elements (i.e., Order ID and Request ID) from the **EcMsAcOrderSrvr**.
Request Standing Order Tracking Elements	Per order	*Process:* EcMsAcOrderSrvr *Libraries:* MsAcClnt, MsAcComm	*Process:* EcPlOdMgr *Library:* PlCore2	The EcPlOdMgr requests standing order tracking elements (i.e., Order ID and Request ID) from the **EcMsAcOrderSrvr**.
Return Order ID	Per order	*Process:* EcPlOdMgr *Library:* PlCore2	*Process:* EcMsAcOrderSrvr *Libraries:* MsAcClnt, MsAcComm	The **EcMsAcOrderSrvr** sends an order id to the EcPlOdMgr to allow the user to track on-demand product or order requests.

Table 4.6-4. PLANG CSCI Process Interface Events (18 of 19)

Event	Event Frequency	Interface	Initiated By	Event Description
Return Configuration Parameters	One set per request	*Process:* EcPlOdMgr *Libraries:* PlOdMgrClient, PlOdMgrClientStub, PlCore1, PlCore1IF, PlCore1Sub, PlCore2, PlUtil, PlSSIT *Classes:* PlOdMgr.c, PlOdMsg.c, PlOdMgrClient.c	*Process:* EcCsRegistry *Library:* EcCsRegistry *Class:* EcRgRegistryServer_C	The **EcCsRegistry** returns the attribute-value pairs (configuration parameters) to the EcPlOdMgr upon request.
Import Location Information	As required for processing	*Process:* EcPlOdMgr *Libraries:* PlOdMgrClient, PlOdMgrClientStub, PlCore1, PlCore1IF, PlCore1Sub, PlCore2, PlUtil, PlSSIT *Classes:* PlOdMgr.c, PlOdMsg.c, PlOdMgrClient.c	*Process:* EcCsIdNameServer *Libraries:* EcPf, Middleware, FoNs, FoIp, oodce *Classes:* EcPfManagedServer, EcPfClient, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy	The EcPlOdMgr requests server location information from the **EcCsIdNameServer**.

Table 4.6-4. PLANG CSCI Process Interface Events (19 of 19)

Event	Event Frequency	Interface	Initiated By	Event Description
Server Account and random password	One per account and password	*Process:* EcPlOdMgr *Libraries:* PlOdMgrClient, PlOdMgrClientStub, PlCore1, PlCore1IF, PlCore1Sub, PlCore2, PlUtil, PlSSIT *Classes:* PlOdMgr.c, PlOdMsg.c, PlOdMgrClient.c	*Process:* Cryptographic Management Interface *Library:* EcSeCmi *Class:* EcSeCmi	The Cryptographic Management Interface generates a random password for the account based on the seed provided by the EcPlOdMgr process.
Request Management Services	At system startup or shutdown and for restarts	*Process:* EcPlOdMgr	DAAC unique startup scripts	**System startup and shutdown -** Please refer to the release-related, current version of the Mission Operations Procedures for the EMD Project document (611) and the current EMD Project Training Material document (625), identified in Section 2.2.1 of this document.

4.6.1.6 Production Planning Data Stores

Table 4.6-5 provides descriptions of the production planning data stores shown in the PLANG CSCI architecture diagram.

Table 4.6-5. PLANG CSCI Data Stores

Data Store	Type	Functionality
PDPS Database	Database	The PDPS database is replicated within each site for fault handling purposes. This PDPS database holds all the persistent data (and facilitates the sharing of this data) including, but not limited to: • Resource information entered with the Resource Planning utilities • PGE and data type information entered at SSIT • Production Request, Data Processing Request and Data Granule Information entered using the Production Request Editor • Plan information entered using the Production Planning Workbench • Task recovery information • Production strategies entered using the Production Strategy GUI The PDPS database also provides security, fault tolerance, and verifies requests for concurrent access to data.

4.6.2 Planning Subsystem Hardware Components

4.6.2.1 Planning Hardware CI (PLNHW) Description

The PLNHW hardware (PLNHW) consists of a server with the Sybase database management system (DBMS) and the workstations to support the Operations staff by providing the Planning Workbench.

Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the Planning HWCI and document 920-TDX-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

4.6.2.2 Planning Workstation Description

The Planning workstation contains and runs the Planning Workbench software. The Planning Workbench is the Production Planning function and the Resource Planning function. One or more workstations are used to run Production Planning and/or Resource Planning at each site based on the site size.

Document 920-TDx-001 (HW Design Diagram) provides descriptions of the Planning HWCI and document 920-TDX-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

4.7 Data Processing Subsystem Overview

The Data Processing Subsystem (DPS) provides the Data Processing capabilities at each EMD site. The DPS capabilities include:

- A queued processing environment to support data product generation. The DPS executes DPRs on available processing resources, as an associated processing job containing all the information needed to accomplish the processing. DPRs are submitted by the PLS and triggered by the arrival of data or triggered internally by the PLS (i.e., reprocessing). PGEs resulting from the integration and test of delivered science algorithms and encapsulated into the SDPS with the Science Data Processing (SDP) Toolkit are used by DPRs to process data. User-specified methods are also used for processing specific data types.
- The Operational interfaces required to monitor the execution of the science software (PGEs).
- Support for science algorithm execution via the SDP Toolkit. The SDP Toolkit is a set of tools to provide a common interface for encapsulating each science algorithm into the SDPS environment. (See the SDP Toolkit Users Guide for the EMD Project (333-EMD-001)) and PGS Toolkit Requirements Specification for the ECS Project (193-801-SD4-001, a.k.a. GSFC 423-16-02) for guidance on the roles and responsibilities of the SDP Toolkit to support the execution of science software.
- Support for the preliminary format processing of data sets (L0 data products) required by the science algorithms.
- Providing an Algorithm Integration and Test (AI&T) environment to integrate new science algorithms, new versions of existing science algorithms, and user methods into the SDPS environment. The system acquires the algorithm or method via an ingest process reflecting local site policies for acceptance of software for integration into the environment. (See Section 4.7.2 “Algorithm Integration and Test Tools (AITTL) CSCI Description).
- The DAAC Quality Assurance (QA) procedures and conditions to verify each data product by the scientific personnel at each DAAC. All data products, both those generated by and input to a submitted job, are available for examination by DAAC scientific personnel to verify data content to be in accordance with quality standards set by the DAAC.

Data Processing Subsystem Context

Figure 4.7-1 is the Data Processing Subsystem context diagrams. The diagrams show the events sent to the DPS and the events the DPS sends to other SDPS and CSMS subsystems and the Operations staff.

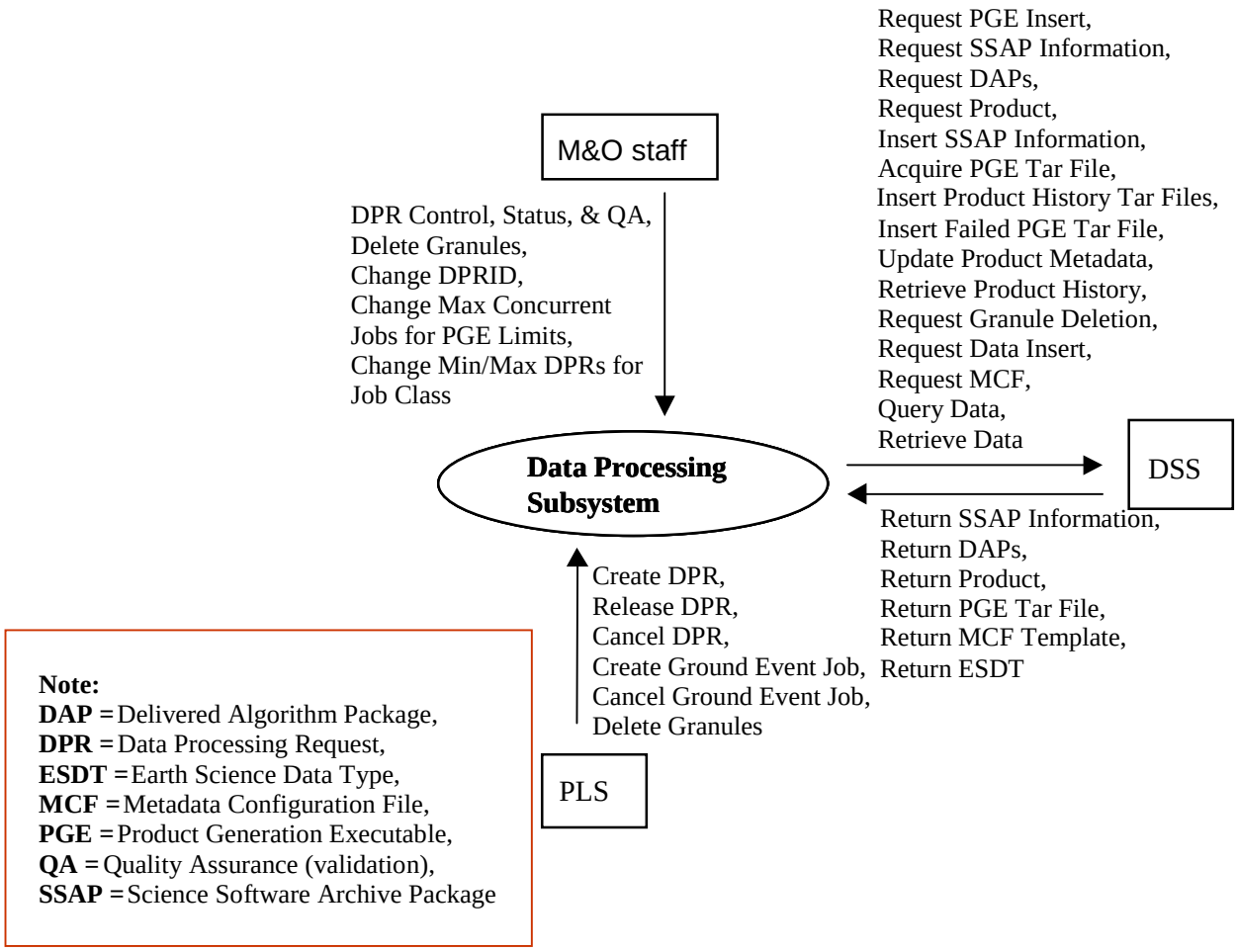


Figure 4.7-1. Data Processing Subsystem Context Diagram

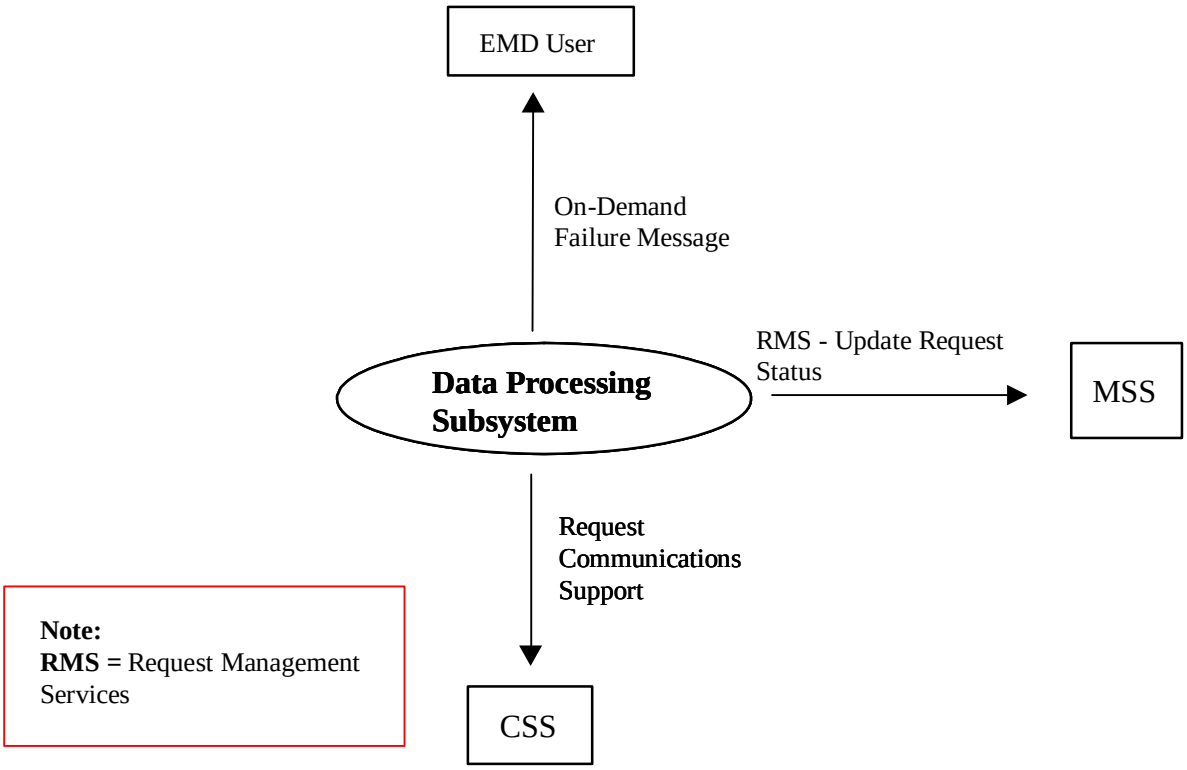


Figure 4.7-1. Data Processing Subsystem Context Diagram (cont.)

Table 4.7-1 provides descriptions of the interface events shown in the Data Processing Subsystem context diagrams.

Table 4.7-1. Data Processing Subsystem Interface Events (1 of 4)

Event	Interface Event Description
DPR Control, Status, & QA	The **M&O staff** provides Data processing control and supports Data Processing Request (DPR) status and Quality Assurance activities.
Delete Granules	The **M&O staff** sends requests to the DPS to delete granules associated with cancelled DPRs.
Change DPR ID	The **M&O staff** can change the DPR ID of an existing DPR.
Change Max Concurrent Jobs for PGE Limits	The **M&O staff** can add to or update values in the DpPrPgeLimits database table.
Change Min/Max DPRs for Job Class	The **M&O staff** can add to or update values in the DpPrClassSchedulingLimits database table.

Table 4.7-1. Data Processing Subsystem Interface Events (2 of 4)

Event	Interface Event Description
Request PGE Insert	The DPS sends requests to the **DSS** to insert data that defines a Product Generation Executable (PGE) and allows it to be scheduled and executed.
Request SSAP Information	The DPS sends requests to the **DSS** for Science Software Archive Package (SSAP) information, including names of existing SSAPs and the information associated with a specific SSAP. In response, the DSS sends lists of SSAPs and related information.
Request DAPs	The DPS requests DAPs based on Universal References (URs) from the **DSS**. The DAPs are placed on a local DPS disk.
Request Product	The DPS sends requests, to the **DSS**, for particular data granules to be pushed, via the File Transfer Protocol (FTP) service, onto the DPS science processor as input for data processing or for Science System Integration and Test (SSIT) work.
Insert SSAP Information	The Operations staff sends requests to the **DSS** to insert SSAP information, via the DPS SSAP Graphical User Interface (GUI), including SSAP name, SSAP version number, PGE name, PGE version number, and SSAP Acceptance Date.
Acquire PGE tar file	The DPS acquires a tar file for any PGE not currently local to the science processor from the **DSS**. The executable is extracted from the tar file and used during PGE execution.
Insert Product History Tar Files	The DPS sends a request to the **DSS** to insert the PGE Production History Tar File resulting outputs for permanent archive after the PGE has successfully completed executing.
Insert Failed PGE Tar File	After an unsuccessful execution of a PGE, the DPS obtains the Tar file containing the PGE log files, core dump (if any), Process Control File (PCF) and other files, and requests the files be inserted into the **DSS** for permanent archive.
Update Product metadata	The Operations Staff uses the Quality Assurance (QA) Monitor GUI in the DPS to send requests to update product metadata in the DSS.
Retrieve Product History	The Operations Staff uses the QA Monitor GUI to submit requests to the **DSS** to transfer the Production History tar file from the Science Data archives to the user's host machine.
Request Granule Deletion	The DPS sends delete requests, to the **DSS**, for particular granules (interim data) in the archive and the associated metadata to be deleted from the SDSRV inventory.
Request Data Insert	The DPS sends requests to the **DSS** to insert a particular file or files into the archive, and catalog the associated metadata in the SDSRV inventory. These files can be processing output, static files received with PGEs, PGE tar files, Algorithm Packages (APS), SSAPs or Delivered Archive Packages (DAPs), failed PGE tar files, or production history files.
Request MCF	The INS and DPS request the Metadata Configuration File (MCF) template, from the **DSS**, prior to a data insert request.
Query Data	The DPS submits requests of this type to the **DSS**. It searches the archive for granules that match the user-supplied selection criteria: data type and begin/end date. Results are displayed to the user.
Retrieve Data	The DPS sends retrieval requests, to the **DSS**, for a particular data granuleId. The product is transferred (pushed), via the FTP service, onto the DPS science processor and used as input for PGE processing or for SSIT work.

Table 4.7-1. Data Processing Subsystem Interface Events (3 of 4)

Event	Interface Event Description
Return SSAP Information	The DPS receives lists of SSAPs and related information from the **DSS**.
Return DAPs	The DAPs are placed on a local DPS disk by the **DSS**.
Return Product	The data granules requested by the DPS are sent from the **DSS**.
Return PGE Tar File	After an unsuccessful execution of a PGE, the DPS obtains the Tar file containing the PGE log files, core dump (if any), Process Control File (PCF) and other files, and requests the files be inserted into the **DSS** for permanent archive.
Return MCF Template	The **DSS** provides the MCF template, to the DPS, to populate as part of the GetMCF service call.
Return ESDT	The **DSS** returns the requested ESDT to the DPS.
Create DPR	The DPS uses the dprId from the **PLS** to insert a job box for a DPR into AutoSys if all the required input data is available. If the input data is not available, information used to construct the job in AutoSys is queued by the Job Management CSC until a release DPR is received.
Release DPR	The DPS uses the dprId from the **PLS** to release jobs currently waiting for external data in the Job Management queue into AutoSys.
Cancel DPR	The DPS uses the dprId from the **PLS** to delete jobs in AutoSys or from the queue.
Create Ground Event Job	The DPS uses the ground event Id from the **PLS** to create a ground event job in AutoSys.
Cancel Ground Event Job	The DPS uses the ground event Id from the **PLS** to delete a ground event job in AutoSys.
Delete Granules	The **PLS** sends requests to the DPS to delete granules associated with cancelled DPRs.
On-Demand Failure Message	The DPS informs the **EMD user** (the submitter) of an On-Demand Processing Request when such a request has an unrecoverable failure.
Request Management Services	The **MSS** provides a basic management library of services to the subsystems, implemented as client or server applications, using the CSS Process Framework. The basic management library of services includes: • **System startup and shutdown** - Please refer to the release-related, current version of the Mission Operations Procedures for the EMD Project document (611) and the current EMD Project Training Material document (625), identified in Section 2.2.1 of this document. • **Update Request Status** - The DPS informs the **MSS** to update the status of an On-Demand Processing Request, when such request changes status (i.e., from Running to Completed or from Running to Failure).

Table 4.7-1. Data Processing Subsystem Interface Events (4 of 4)

Event	Interface Event Description
Request Communications Support	The **CSS** provides a library of services available to each SDPS and CSMS subsystem. The subsystem services required to perform specific assignments are requested from the CSS. These services include: • CCS Middleware Support • Database Connection Services • Network & Distributed File Services • Name/Address Services • Password Services • Server Request Framework (SRF) • Universal Reference (UR) • Error/Event Logging • Fault Handling Services • Mode Information • Query Registry - Retrieving the requested configuration attribute-value pairs from the Configuration Registry

The DPS has an internal interface to the COTS software product AutoSys. The DPS creates, starts, and deletes job boxes in AutoSys via this interface.

The PLS determines the processing activities required to generate data products specified by the Operations staff in a Production Request. Each processing activity is called a Data Processing Request (DPR). The PLS creates, releases or cancels DPRs in AutoSys via the DPS.

The DSS accesses the data archives via authorized user requests. The DPS requests the required input data for a Product Generation Executable (PGE) and Metadata Configuration Files (MCFs) from the DSS. The DPS also inserts PGE generated products, provides product production histories, and provides failed PGE information for debugging purposes. The DPS uses the DSS as a permanent repository for PGE tar files, Algorithm Packages (APs), Science Software Archive Packages (SSAPs) and Delivered Algorithm Packages (DAPs).

Data Processing Subsystem Structure

The DPS is comprised of three CSCIs:

- The Processing (PRONG) CSCI manages and monitors the Science Data Processing (SDP) environment to execute Science Software and algorithms (called PGEs) and generates data products.
- The Algorithm Integration and Test Tools (AITTL) CSCI is a set of tools for test and integration of new science software, new versions of existing science software, and user methods in the SDP operational environment. AITTL combines custom developed code with COTS software starting from a central application called the SSIT Manager.
- The SDP Toolkit (SDPTK) CSCI provides a set of software libraries to integrate Science Software into the SDPS environment. By promoting the POSIX standard, these libraries allow the SDP environment to support the generation of data products in a heterogeneous computer hardware environment. (See SDP Toolkit Design Specification (455-TP-001-001) for the SDPTK architecture).

Use of COTS in the Data Processing Subsystem

- **Computer Associate's AutoSys** is a job scheduling software application to automate operations in a distributed UNIX environment. AutoSys performs automated job control functions for scheduling, monitoring, and reporting on the jobs residing on any Unix machine attached to an EMD network on the Science Data Processing hardware. AutoSys provides job-scheduling support with an Operator Console for monitoring and human intervention in the job stream. The Operator console allows the M&O staff to restart failed jobs and to view the status of events related to the job's execution. The Operator console includes an alarm manager, set in the job definition, to assist the Operations staff when responding to fault situations.

- **Computer Associate's AutoXpert** is a GUI providing different methods of viewing a job schedule progress. Noting color changes on the JOBSCAPE GUI can monitor the progression of DPR execution. Failed jobs can be detected and restarted if the job has failed due to the unavailability of an external resource. The HostScape GUI can be used to view the status of the science processors.

- **Sybase Adaptive Server Enterprise (ASE)**

  The Sybase ASE provides the capabilities to insert, update and delete PDPS database content. The Sybase ASE must be operational during the DPS operations.

- **CCS Middleware Client**

  CCS Middleware Client provides DSS with communications between other subsystems. CCS Middleware can reside on one or both sides of the interface. An instance must be installed on the platform where DSS resides. Although the CCS Middleware Client is part of CSS, this COTS product must be installed for DSS to run in the SDPS operational and test environment.

The DPS provides the hardware resources for science software execution, queuing, dispatching, and managing in a distributed environment of computing platforms. The DPS hardware consists of three CIs:

- Science Processor - The Science Processor HWCI (SPRHW) contains processing resources (central processing units, memory, disk storage, and input/output subsystems) necessary to perform first-time processing, reprocessing, and Algorithm Integration and Test (AI&T). Also, SPRHW provides the hardware resources (a Queuing Server) to support management of the science processes.

- Algorithm Quality Assurance - The Algorithm Quality Assurance HWCI (AQAHW) supports the DAAC Operations staff in performing the planned science and non-science product data quality validation procedures.

- Algorithm Integration and Test - The AI&T HWCI (AITHW) resources provide the operating system and support for the integration and test of science software at each DAAC. AITHW is the workstations and hardware tools required for software integration

and test. AITHW does not, in this case, provide the computer capacity required for science software test (SPRHW provides the test capacity).

Error Handling and processing

EcUtStatus is a class used throughout the EMD custom code for general error reporting.

The Data Processing Subsystem (DPS) uses the EcUtStatus class for general error handling. It is almost always used as a return value for functions and allows detailed error codes to be passed back up function stacks.

The PRONG and AITTL CSCIs use two main mechanisms for error handling.

1. Return Values

   Functions can return an EcUtStatus object, which can be used to indicate a general success/failure status. Also, more detailed information on the exact reason for the failure can be provided. This is the most widely used mechanism within the PRONG and AITTL and in general these errors get propagated back up to the top-level functions with ALOG error messages being generated along the way.

2. Exceptions

   Some functions (for example, class constructors) cannot return values to indicate success or failure. These functions can throw exceptions. These errors are usually caught by other functions at a low level and converted into EcUtStatus return values (as described in 1).

Currently, the PRONG and AITTL CSCIs client interfaces only support returning error messages back to client programs, along with a generic success/failed status.

In addition, the PRONG and AITTL use the following method of tracking errors; stored procedure tracing. Some PRONG and AITTL stored procedures write error and informational messages to a table in the Planning and Data Processing Subsystems (PDPS) database named DpPrTrace. Table 4.7-2 shows an example. The stored procedure ProcInsertTrace is used to do this. A trigger on this table truncates the messages after 10,000 messages have been written to the table. There are no other specialized error libraries and classes used by the PRONG and AITTL.

Table 4.7-2. Example Row from the DPS DpPrTrace Table

Column	Example
Time	Apr 2 2002 14:01:27.960000
Procedure Name	ProcGetReadyDPRs
Called From	(NULL)
Dpr Id	MODPGE02#s28021500OPS
Message	99 pgeId slots available for pgeId MODPGE02#syn#001

For writing messages to the Applications Log (ALOG), the following functions are used:

EcLgLogError sends a message to the ALOG at severity level 1. For example, EcLgLogError(methodName, returnStatus.GetLogMessageLink(),

"Error: unable to copy tar file from '%s%s' to '%s' ",

dataserverpath.data(), pgeTarName.data(), destinationPath.data());

EcLgLogWarning sends a message to the ALOG at severity level 2. For example, EcLgLogWarning(methodName, lstatus.GetLogMessageLink(), "error terminating DPR job") ;

EcLgLogInformational sends a message to the ALOG at severity level 3. For example, EcLgLogInformational(methodName,

status.GetLogMessageLink(),"not using DpPrAutoSysProfile");

For writing messages to the debug log, the following macros are used:

PF_STATUS writes a message at a "log level" of 1 to the debug log. For example,

PF_STATUS { cerr << methodName << "Can't retrieve DpPrDeleteFailedPGEJobs";

cerr << " param. -- setting to FALSE" << endl;}

PF_VERBOSE writes a message at a "log level" of 2 to the debug log. For example,

PF_VERBOSE {cerr << methodName <<"urName = " << urName << endl;}

PF_DEBUG writes a message at a "log level" of 3 to the debug log. For example,

PF_DEBUG {cerr << methodName << "can not get PF config file pointer!" << endl;}

TOOLKIT Error/Status Reporting (SMF Tools)

To detect and report error and status conditions in a consistent manner, standardized status messages and status codes must first be established. The method used to institute these message/code pairs is by way of the ‘smfcompile’ utility. But first, users need to create Status Message Files (SMFs) to contain their custom status messages and corresponding status identifiers. These identifiers take the form of user defined mnemonics that visually convey the essence of the status message. The user makes direct use of these mnemonics in their software when testing for status conditions and when interfacing with the SMF Toolkit functions. Once an SMF is completed, the smfcompile utility is run to bind the status messages and mnemonics with

integral status codes. This process facilitates the run-time access of all status messages and provides for the referencing of status mnemonics within the user's code.

The status codes generated by the 'smfcompile' utility are guaranteed to be unique to ensure that there are no ambiguous status conditions, in the event that code fro different Science Computing Facilities (SCFs) is merged into a single executable and/or PGE. This uniqueness is possible because "seed" values, which are different for every SMF, are used in the generation of the status codes. Typically, many SMF files are created in the course of software development; therefore many seed numbers are required. However, it is important to note that valid seed numbers can only be obtained from the Toolkit development team. Any attempt to produce SMFs from "home-grown" seed values can result in the SMFs being unusable at integration & test time.

The SDP Toolkit routines actually contain their own collection of status codes and associated status messages for describing the state of each Toolkit function. Users of the Toolkit functions should examine the return values of each tool before performing any other action. To inform a calling unit (user's software) about the exit state of a called Toolkit routine, each Toolkit function sets a status message and assigns a status code to the return value. On the basis of its interpretation of this return value, the calling unit may elect to perform some error handling. As part of this procedure, the user should either propagate the existing status code up through their calling hierarchy, or set a status code and message to represent the outcome of any local error handling attempt.

Upon detection of an error state, users are advised to report on the existing error prior to performing an error handling procedure. The content of these reports might include the following:

- A user-defined message string to convey the nature of the status condition
- A user-defined action string to indicate the next operation to be performed in response to the status condition
- A system defined string that uniquely identifies the environment in which the status condition occurred.

However, this is merely a suggestion; the users are free to define the content of the status reports to satisfy their own requirements. The method for reporting this information involves the generation of a report from the information just described and the subsequent transmission of that report to the appropriate destination(s).

The tools provided here allow for the propagation of status information within a PGE executable to facilitate a user's error handling process. They also provide the means to communicate status and error information to various monitoring authorities and event logs. Additionally, there is a tool that enables the user to specify, a priori, the action to be taken in the wake of a fatal arithmetic event. This mechanism allows users to take their own corrective measures to control an event that is terminal by default. Note that all other event conditions fall under the purview of system processing and are thereby controlled by the governing SDPS software.

Several new features have been incorporated into these tools for Toolkit 5 in order to improve their efficiency. One of those features allows for the buffering of individual status messages up to some user defined run-time limit. This should greatly reduce the amount of I/O required to access these messages. As a process proceeds to completion, new status messages are buffered as older, less used status messages become un-buffered. The goal here is to only access status messages from their run-time file when they are being referenced for the first time. The actual observed improvement depends on the degree to which a process' status messages are localized (i.e., A particular status message should ideally only be referenced within a short body of code.) and the buffer size, which is set by the user. Another feature reduces the number of replicated status messages that can appear in the status log file. This is accomplished by "compressing" duplicate messages into a count of such messages. This feature should significantly reduce the size of the status log file and contribute to its general readability.

Since each function has only one return value, every effort has been made to preserve the most important warning or error value on returning. Given that subordinate functions often have several possible returns, and different users have different priorities, it is always advisable to check the message log as well as examining the return. When totally inconsistent behavior is found in a return from a subordinate function, the returned value is PGS_E_TOOLKIT. Example: a Toolkit function passes an internally generated vector, whose length is certain to be nonzero, to a subordinate function. The lower-level function then returns a warning or error return stating that the vector is of zero length; while the higher-level function returns PGS_E_TOOLKIT. Another example: if a valid spacecraft tag is passed in, but rejected as invalid down the processing line, the error PGS_E_TOOLKIT is returned by the higher-level function. Thus return value PGS_E_TOOLKIT indicates a flaw in the software, the violation of an array boundary, hardware, compiler, or system error, corrupted data, or some similarly serious condition that invalidates the processing.

Logging Control

PCF entry:
10114|Logging Control; 0=disable logging, 1=enable logging|1

This can be used to disable logging altogether. If logging is disabled NO message will output to any log files (although a small header is still written to the log files indicating logging for this PGE has been disabled). The Default State is for logging to be enabled.

Trace Control

PCF entry:
10115|Trace Control; 0=no trace, 1=error trace, 2=full trace|0

This can be used to specify the trace level for message logging. Tracing is a feature made possible by the addition of two SMF tools: PGS_SMF_Begin and PGS_SMF_End. Users may include these tools at the beginning and ending of their functions (respectively) to signal to the SMF system when each user defined function is entered and exited. Three levels of tracing are possible:

No Tracing

This is the Default State. No information concerning the entering or exiting of functions is recorded to the log files. No information concerning the path of a function call is recorded to the log files.

Example Log Entry:
func4():PGSTD_W_PRED_LEAPS:27652
predicted value of TAI-UTC used (actual value unavailable)

Error Tracing

If error tracing is enabled, information concerning the path of a function call is recorded to the log files any time a status message is logged to a log file. This is useful in determining where in a chain of function calls an error occurred. No information concerning the entering or exiting of functions is recorded in this state.

Example Log Entry:
```
main():
   func1():
      func2():
         func3():
            func4():PGSTD_W_PRED_LEAPS:27652
            predicted value of TAI-UTC used (actual value unavailable)
```

Full Tracing

If full tracing is enabled, a message is written to the log files each time a function is entered and exited (only those user functions with the PGS_SMF_Begin/End calls, see above). Indenting is also done to show the path of each function call.

Example Log Entry:

```
PGS_SMF_Begin: main()

  PGS_SMF_Begin: func1()

    PGS_SMF_Begin: func2()

      PGS_SMF_Begin: func3()

        PGS_SMF_Begin: func4()

          func4():PGSTD_W_PRED_LEAPS:27652
          predicted value of TAI-UTC used (actual value unavailable)

        PGS_SMF_End: func4()

PGS_SMF_End: func3()

PGS_SMF_End: func2()
```

PGS_SMF_End: func1()

PGS_SMF_End: main()

Process ID Logging

PCF entry:
10116|Process ID logging; 0=don't log PID, 1=log PID|0

This can be used to enable the tagging of log file entries with the process ID of the process from which the entry came. This is useful for PGEs that run concurrent processes, which are all writing to a single log file simultaneously. If process ID logging is enabled, each log entry is tagged with the process ID of the process making the entry. This can facilitate in post-processing a log file.

Example Log Entry:
func4():PGSTD_W_PRED_LEAPS:27652 (PID=2710)
predicted value of TAI-UTC used (actual value unavailable)

Status Level Control

PCF entry:
10117|Disabled status level list (e.g., W S F)|<status level list>

This can be used to disable the logging of status codes of specific severity levels. A list of levels to be disabled should be substituted for <status level list> (e.g.: N M U). No message of a status level indicated in the list is recorded to any log. The Default State is to enable logging for all status levels.

Status Message Seed Control

PCF entry:
10118|Disabled seed list|<status code seed list>

This can be used to disable the logging of status codes generated from specific seed values. A list of seed values, the status codes derived from which should be disabled, should be substituted for <status code seed list> (e.g.: 3 5). No message derived from a seed value indicated in the list is recorded to any log. The Default State is to enable logging for all seed values.

Individual Status Code Control

PCF entry:
10119|Disabled status code list|<status code list>

This can be used to disable the logging of specific status codes. A list of status code mnemonics and/or numeric status codes should be substituted for <status code list> (e.g.: PGSTD_M_ASCII_TIME_FMT_B 678954). Note that using mnemonics can disable only Toolkit status codes. To disable a user generated status code a numeric status code must be

used. No messages whose status codes or mnemonics included in the list are recorded to any log file. The Default State is to enable logging for all status codes.

Generating Run-time E-Mail Messages

A PGE may be configured to automatically generate and send e-mail messages during run-time when specific user defined status codes are logged. This is done by assigning an e-mail action to a given user defined status code.

An e-mail action is an SMF code with the special status level of "C" and a mnemonic that begins with the characters "PGSEMAIL" (the rest of the mnemonic may contain any other valid mnemonic characters), for example:

PGS_*C_PGSEMAIL*_SEND_EMAIL
ASTER_*C_PGSEMAIL*_ALERT
MODIS_*C_PGSEMAIL*_ERROR

An e-mail message is generated anytime a user defined status code with an associated e-mail action is logged via the SMF logging routines. The contents (body) of these messages are the text (message) associated with the user defined status code. The subject of these messages is the mnemonic associated with the user defined status code. The list of recipients is defined in the e-mail action definition.

Example:
In a user defined status message file the following status code mnemonic label and e-mail action mnemonic label have been defined (the e-mail action is associated with the status code via the "::" syntax):

```
MODIS_E_PGE_INIT_FAILED   The PGE failed to initialize.
                          ::MODIS_C_PGSEMAIL_NOTIFY
MODIS_C_PGSEMAIL_NOTIFY   john@modis.org, sue@modis.org
```

The following lines appear in a C source code file:

```
    returnStatus = initializePGE();
    if (returnStatus == MODIS_E_PGE_INIT_FAILED)
    {PGS_SMF_SetStaticMsg(returnStatus, "main()");
        exit(1);}
```

At run-time, if the returned status code from the function initializePGE() has the value defined by MODIS_E_PGE_INIT_FAILED, this status is logged via the SMF function PGS_SMF_SetStaticMsg(), and because this status code has an e-mail action associated with it, an e-mail message is generated.

The e-mail message is sent to: sue@modis.org and john@modis.org
the subject field of the e-mail message is : MODIS_E_PGE_INIT_FAILED
The text of the e-mail message is : The PGE failed to initialize.

Note:
This functionality will be disabled at the DAACs.

4.7.1 Processing Software Description

4.7.1.1 Processing Functional Overview

The Processing (PRONG) CSCI initiates, monitors, and manages the execution of science software algorithms (referred to as PGEs). The PRONG CSCI is informed of the required execution of a PGE through a DPR received from the PLS. When all necessary input data becomes available, PRONG initiates the execution of the PGE. (N.B.: Some or all input data can reside in a Data Server not at the DAAC site.)

The PRONG CSCI has the following capabilities:

- Manages execution of science software algorithms
- Manages SDP computer hardware resources
- Manages the data flow required to execute a science software algorithm
- Manages the data flow generated by the execution of a science software algorithm
- Monitors processing status, and allows manual intervention, when necessary, in the SDP operations environment, including processing queue control
- Supports validation of product data quality
- Provides status and user updates (in the event of a failure) for On-Demand Processing Requests

4.7.1.2 Processing Context

Figure 4.7-2 is the PRONG CSCI context diagrams. The diagrams show the events sent to the PRONG CSCI and the events the PRONG CSCI sends to other CSCIs and the Maintenance and Operations (M&O) staff.

Table 4.7-3 provides descriptions of the interface events shown in the PRONG CSCI context diagrams.

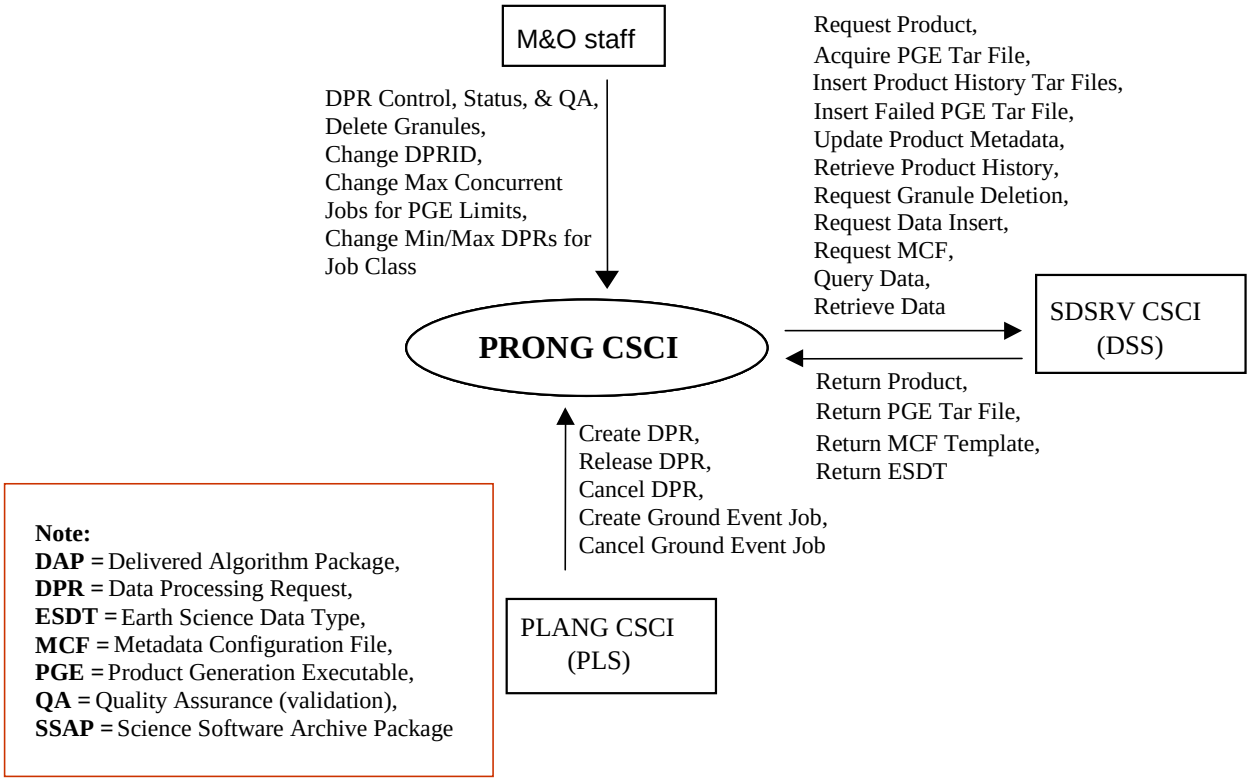


Figure 4.7-2. PRONG CSCI Context Diagram

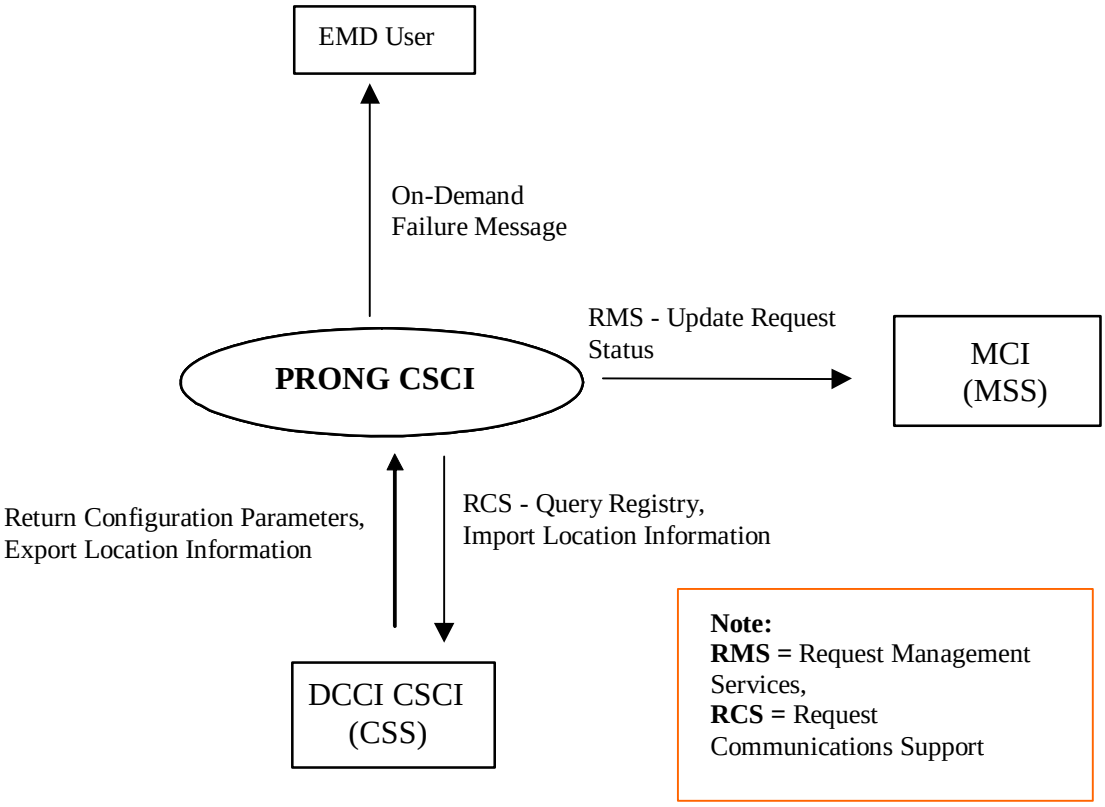


Figure 4.7-2. PRONG CSCI Context Diagram (cont.)

Table 4.7-3. PRONG CSCI Interface Events (1 of 3)

Event	Interface Event Description
Request Product	The PRONG CSCI sends requests, to the **SDSRV CSCI**, for particular data granules to be pushed, via the FTP service, onto the DPS science processor as input for data processing or for SSIT work.
Acquire PGE tar file	The PRONG CSCI acquires a tar file for any PGE not currently local to the science processor from the **SDSRV CSCI**. The executable is extracted from the tar file and used during PGE execution.
Insert Product History Tar Files	The PRONG CSCI sends a request to the **SDSRV CSCI** to insert the PGE Production History Tar File resulting outputs for permanent archive after the PGE has successfully completed executing.
Insert Failed PGE Tar File	After an unsuccessful execution of a PGE, the PRONG CSCI obtains the Tar file containing the PGE log files, core dump (if any), PCF and other files, and requests the files be inserted into the **SDSRV CSCI** for permanent archive.
Update Product metadata	The Operations Staff uses the QA Monitor GUI in the PRONG CSCI to send requests to update product metadata in the **SDSRV CSCI**.
Retrieve Product History	The Operations Staff uses the QA Monitor GUI to submit requests to the **SDSRV CSCI** to transfer the Production History tar file from the Science Data archives to the user's host machine.
Request Granule Deletion	The PRONG CSCI sends delete requests to the **SDSRV CSCI** for particular granules (interim data) in the archive and associated metadata to be deleted from the SDSRV inventory.
Request Data Insert	The PRONG CSCI sends requests to the **SDSRV CSCI** to insert a particular file or files into the archive, and catalog the associated metadata in the SDSRV inventory. These files can be processing output, static files received with PGEs, PGE tar files, APs, SSAPs or DAPs, failed PGE tar files, or production history files.
Request MCF	The PRONG CSCI requests a Metadata Configuration File (MCF) template for each output data type for specific PGEs from the **SDSRV CSCI** and the MCF template is populated with metadata from the output granule.
Query Data	The PRONG CSCI submits requests of this type to the **SDSRV CSCI**. It searches the archive for granules that match the user-supplied selection criteria: data type and begin/end date. Results are displayed to the user.
Retrieve Data	The PRONG CSCI sends retrieval requests, to the **SDSRV CSCI**, for a particular data granuleId. The product is transferred (pushed), via the File Transfer Protocol (FTP) service, onto the DPS science processor and used as input for Product Generation Executive (PGE) processing or for Science Software Integration and Test (SSIT) work.
Return Product	The data granules requested by the PRONG CSCI are sent from the **SDSRV CSCI**.
Return PGE Tar File	After an unsuccessful execution of a PGE, the PRONG CSCI obtains the Tar file containing the PGE log files, core dump (if any), PCF and other files, and requests the files be inserted into the **SDSRV CSCI** for permanent archive.

Table 4.7-3. PRONG CSCI Interface Events (2 of 3)

Event	Interface Event Description
Return MCF Template	The PRONG CSCI receive the MCF template to populate, from the **SDSRV CSCI**, as part of the GetMCF service call.
Return ESDT	The SDSRV CSCI returns the requested ESDT to the PRONG CSCI.
Create DPR	The PRONG CSCI uses the dprId from the **PLANG CSCI** to insert a job box for a DPR into AutoSys if all the input data required for the DPR is available. If the input data is not available, information used to construct the job in AutoSys is queued by the Job Management CSC until a release DPR is received.
Release DPR	The PRONG CSCI uses the dprId from the **PLANG CSCI** to release jobs currently waiting for external data in the Job Management queue into AutoSys.
Cancel DPR	The PRONG CSCI uses the dprId from the **PLANG CSCI** to delete jobs in AutoSys or from the queue.
Create Ground Event Job	The PRONG CSCI uses the ground event Id from the **PLANG CSCI** to create a ground event job in AutoSys to perform maintenance activities on data processing resources.
Cancel Ground Event Job	The PRONG CSCI uses the ground event Id from the **PLANG CSCI** to delete a ground event job in AutoSys.
DPR Control, Status, & Q/A	The **M&O staff** controls Data Processing Request (DPR) activity with the capability to cancel, suspend, resume, and modify a DPR. The M&O staff supports status collecting, PRONG hardware resource monitoring and Quality Assurance validating processes.
Delete Granules	The **M&O staff** sends requests to the PRONG CSCI to delete granules associated with cancelled DPRs.
Change DPR ID	The **M&O staff** can change the DPR ID of an existing DPR.
Change Max Concurrent Jobs for PGE Limits	The **M&O staff** can add to or update values in the DpPrPgeLimits database table.
Change Min/Max DPRs for Job Class	The **M&O staff** can add to or update values in the DpPrClassSchedulingLimits database table.
On-Demand Failure Message	The PRONG CSCI informs the **EMD user** (the initiator of the request) if an On-Demand Processing Request fails, if such failure is unrecoverable.
Request Management Services	The **MCI** provides a basic management library of services to the CSCIs, implemented as client or server applications, using the DCCI CSCI Process Framework. The basic management library of services includes: • **System startup and shutdown -** Please refer to the release-related, current version of the Mission Operations Procedures for the EMD Project document (611) and the current EMD Project Training Material document (625), identified in Section 2.2.1 of this document. • **Update Request Status** - The PRONG CSCI informs the **MCI** to update the status of an On-Demand Processing Request, when such request changes status (i.e., from Running to Completed or from Running to Failure).

Table 4.7-3. PRONG CSCI Interface Events (3 of 3)

Event	Interface Event Description
Request Communications Support	The **DCCI CSCI** provides a library of services available to each SDPS and CSMS CSCI. The CSCI services required to perform specific assignments are requested from the DCCI CSCI. These services include: • CCS Middleware Support • Database Connection Services • Network & Distributed File Services • Bulk Data Transfer Services • Name/Address Services • Password Services • Server Request Framework (SRF) • Universal Reference (UR) • Error/Event Logging • Fault Handling Services • Mode Information • Query Registry - Retrieving the requested configuration attribute-value pairs from the Configuration Registry
Import Location Information	The PRONG CSCI requests server location information from the **DCCI CSCI** (CCS Name Server).
Return Configuration Parameters	The **DCCI CSCI** returns the requested configuration parameters to the PRONG CSCI.
Export Location Information	The CCS Middleware CSC stores physical and logical location information received from PRONG CSCI in the **DCCI CSCI** (CCS Name Server).

4.7.1.3 Processing Architecture

Figure 4.7-3 is the PRONG CSCI architecture diagrams. The diagrams show the events sent to the PRONG CSCI processes and the events the PRONG CSCI processes send to other processes. The PRONG CSCI consists of COTS software and EMD developed processes.

The PRONG CSCI has interfaces with:

- Planning Subsystem – The PLS creates a production plan executed by the PRONG CSCI through the use of DPRs. Each DPR represents one processing job performed by a DPS computer resource. The PRONG CSCI provides DPR status information to the PLS to assist in production management activities.

- Data Server Subsystem – The PRONG CSCI supports SDPS data generation by requesting and receiving data (Data Staging) from a Data Server maintaining raw data and generated products. Also, the PRONG CSCI transfers data (Data de-staging) to a Data Server to archive generated data products.

- SDP Toolkit – The PRONG CSCI provides the location of input data and the location for the generated output data products. While a PGE is executing, the PRONG CSCI monitors the execution and provides current status to the M&O staff. Status includes current processing event history (e.g., data staging, and execution). Process monitoring

includes checking resource usage by the PGE. At PGE execution completion, the PRONG CSCI initiates the transfer of the generated data product to the respective Data Server.

- System Management Subsystem – The PRONG CSCI relies on the MSS services for resource management and thus provides system management information including fault, accounting, configuration, security, performance, and accountability to the MSS. It also uses the Request Status Tracking capability to update the status of On-Demand Requests made by users.
- Operations – Supports PGE execution management and monitoring and the generation of SDPS Data Products via a Human Machine Interface (HMI). The HMI supports status information collection for a DPR, controlling DPR executions, and monitoring the status of the DPS hardware resources. The HMI also supports manual quality assurance activities performed at the DAAC.

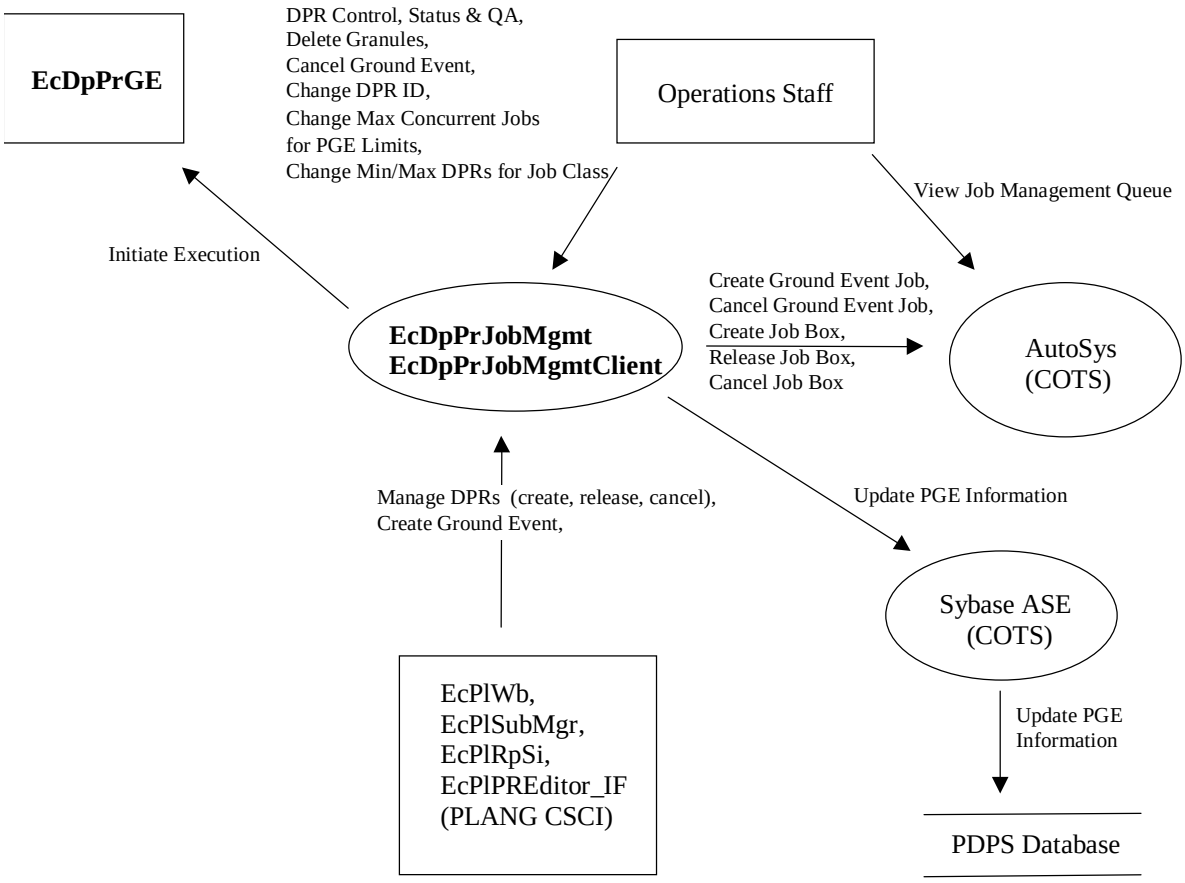


Figure 4.7-3. PRONG CSCI Architecture Diagram

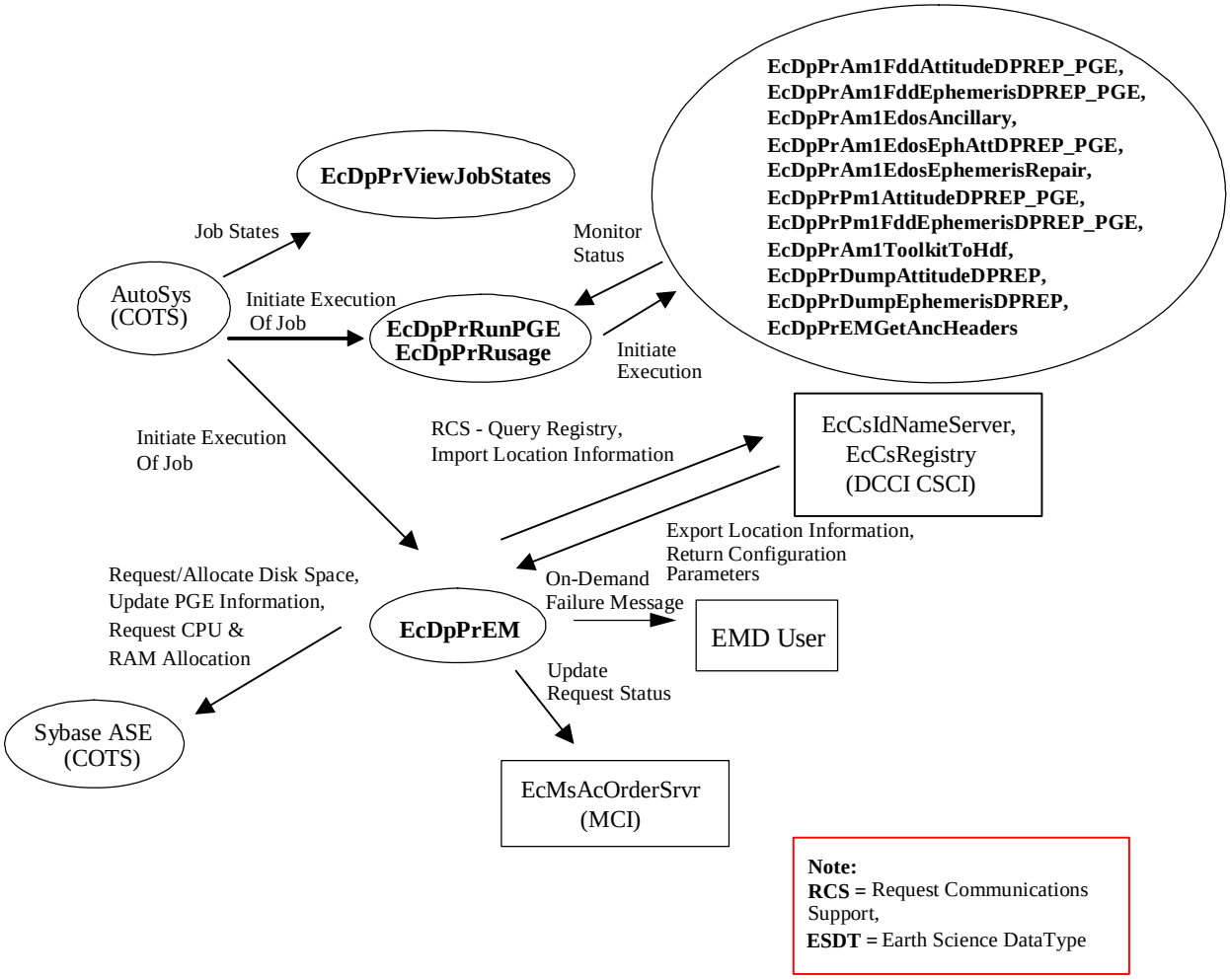


Figure 4.7-3. PRONG CSCI Architecture Diagram (cont.)

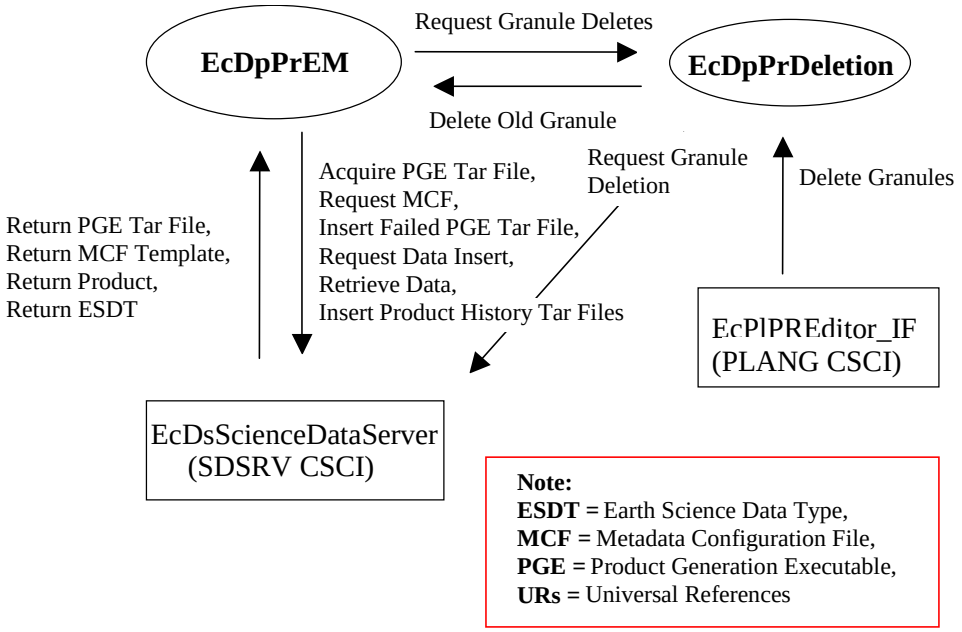


Figure 4.7-3. PRONG CSCI Architecture Diagram (cont.)

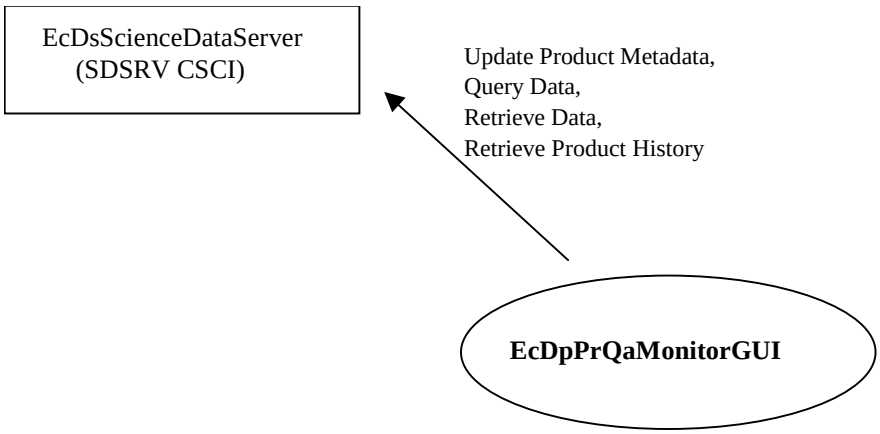


Figure 4.7-3. PRONG CSCI Architecture Diagram (cont.)

4.7.1.4 Processing Process Descriptions

Table 4.7-4 provides descriptions of the processes shown in the PRONG CSCI architecture diagrams. **Note:** Resource Management and the data manager are an integral part of PRONG. Unlike the Execution Manager, they are libraries of routines used by the Execution Manager

instead of being executed like it. The Resource Manager checks the resource requirements (i.e., disk space, memory, and Central Processing Unit (CPU)) of a PGE during processing, determines the availability of those resources, and dedicates their usage until PGE execution finishes. The Data Management library (DpPrDM) manages the flow of science data to and from science processing resources. Data Management manages data retention on science processing resources to support PGE executions.

DPREP Background Information

A Flight Dynamics Division (FDD) data set consists of a couple of headers followed by a number of records. Each FDD record has fifty ephemeris points, spaced in time by a spacing, which is mentioned in the first FDD header. Each ephemeris point consists of time, three position coordinates, three velocity coordinates and a quality flag. The quality flag is set by DPREP. The rest comes in the FDD data set. Each FDD data set will have two hours of data for Terra (AM-1) and 24 hours worth of data for Aqua (PM-1) and Aura.

DPREP must read in all the FDD records and look for node crossings. A node crossing occurs when the satellite crosses the earth's equatorial plane. If the crossing is from south to north, it is called an ascending node crossing. If the crossing is from north to south it is called a descending node crossing.

The raw data comes in the form of position and velocity coordinates. The position transition from one point to another has to be checked for an equator crossing and the type and longitude of the crossing computed. Orbits are defined as being from one ascending node to another and are sequentially numbered from the first orbit. In addition, it is mandated for all orbits touched, orbit metadata (which is defined as the orbit number, the ascending and descending node crossing times and the descending node crossing longitude) has to be computed and appended at the end of the processed ephemeris data.

In addition, QA analysis must be performed. There are mainly two kinds of QA: data gaps and range violations. Data gaps correspond to missing data in the file or between data sets. A range violation is position or velocity magnitudes, which are less than a critical minimum or more than a critical maximum. This QA is written out in the header of the output HDF and native format data sets and also in the metadata files.

For a description of DPREP functionality, refer to the following documents:

Terra Spacecraft Ephemeris and Attitude Data Processing (Document Number 500-EMD-001)

EOSDIS Spacecraft Ephemeris and Attitude Data Specification: Contents and Structure (available on the World Wide Web at http://newsroom.gsfc.nasa.gov/sdptoolkit/appendl.html)

Release 7 SDP Toolkit User's Guide for the EMD Project, Appendix L (document number 333-EMD-001)

The list of available documents will grow as more platforms are brought on-line for which DPREP must process ephemeris and attitude data.

Table 4.7-4. PRONG CSCI Processes (1 of 4)

Process	Type	Hardware CI	COTS/ Developed	Functionality
EcDpPrEM	Other	SPRHW	Developed	The Execution Management process initiates the execution of PGEs (via the COTS product AutoSys). EcDpPrEM supports the preparation activities prior to the execution of PGEs and subsequent activities to the execution of PGEs. EcDpPrEM also provides status on On-Demand Processing Requests and send out 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.
EcDpPrRunPGE EcDpPrRusage	Other	SPRHW	Developed	The PGE Execution Manager process controls and monitors PGE executions including Process Control File creation and output product storage growth. EcDpPrRusage measures the actual resources used by the PGE and reports to AutoSys unexpected resource usage.
EcDpPrDeletion	Server	SPRHW	Developed	This CCS Middleware notifies the EcDsScienceDataServer to remove interim granules via the execution management process (EcDpPrEM) when they are no longer needed. The interim products are removed after the last PGE in the chain has used them or a pre set time has expired after the last use of the interim product. It also is used by the PLS to delete granules associated with a cancelled DPR.

Table 4.7-4. PRONG CSCI Processes (2 of 4)

Process	Type	Hardware CI	COTS/ Developed	Functionality
AutoSys	GUI	SPRHW	COTS	AutoSys is a job scheduling software application used to automate operations in a distributed UNIX environment. AutoSys executes jobs to automate support for PGE execution. AutoSys creates job boxes consisting of a series of related jobs, and manages job dependencies. AutoSys provides graphical depictions of completed jobs and jobs being processed. It includes the Operator Console GUI to allow human intervention into monitoring and altering the AutoSys job stream. The daily job schedule is submitted to the Job Management server at the start of the processing day. Jobs, which have data available, are released into AutoSys. To support job executions, AutoSys requires additional help for: Allocation of sufficient resources (e.g., disk space) to support executions. The EcDpPrEM provides the capabilities to manage disk space and monitor resources. Managing remote host data acquisition, data retention on the DPS processing host, and data distribution from the DPS processing host Initialization and PGE executions.

Table 4.7-4. PRONG CSCI Processes (3 of 4)

Process	Type	Hardware CI	COTS/ Developed	Functionality
EcDpPrJobMgmt EcDpPrJobMgmtClient EcDpPrViewJobStates	Server	SPRHW	Developed	The Job Management process uses the AutoSys COTS product to create and initiate execution of PRONG administrative jobs for managing SPRHW assets and for PGE execution. Three Unix processes bundled together into an AutoSys job box perform this work. Job Management is responsible for efficient AutoSys management so the maximum number of jobs possible can be continuously run using the product. This involves controlling the flow of jobs through AutoSys by only allowing jobs ready to run into the product and by removing jobs as they complete. Job Management also creates and starts execution of Ground Event jobs in AutoSys. The Job Management Client process is used by programs that need access to the Job Management Server services to modify jobs in AutoSys to change the priority of the jobs. The various events this process provides are: CreateDPR: A data processing request identified is then translated into seven standard process steps (one, the PGE execution, the remaining performing support activities). If the science data is available, the job box containing the seven individual jobs is released into AutoSys. ReleaseDPR: A previously created data processing request waiting for the availability of science data is released into AutoSys to begin execution. Change DPR Id: Rename a DPR Change Max Concurrent Jobs for PGE Limits Table: Modify DpPrPgeLimits database table. Change Min/Max DPRs for Job Class: Modify DpPrClassSchedulingLimits database table. CancelDPR: This provides the capability to cancel/terminate a data processing request. CreateGEvntJob: Create a Ground Event Job in AutoSys. CancelGEvntJob: Cancel a Ground Event Job. ViewJobManagement DPR Queue: The View Job States process allows the Operations Staff to view jobs in the queue to determine the completed jobs, the jobs executing, and the jobs awaiting execution.

Table 4.7-4. PRONG CSCI Processes (4 of 4)

Process	Type	Hardware CI	COTS/ Developed	Functionality
EcDpPrQaMonitorGUI	GUI	PLNHW	Developed	This process provides the capability to transfer science data from the archives, browse data images, and examine and update science metadata. It is an automated tool for performing data analysis in support of DAAC Quality Assurance activities.
EcDpPrAm1EdosAncillary, EcDpPrAm1EdosEphAttDPREP_PGE, EcDpPrAm1EdosEphemerisRepair, EcDpPrAm1FddAttitudeDPREP_PGE, EcDpPrAm1FddEphemerisDPREP_PGE, EcDpPrAm1ToolkitToHdf, EcDpPrDumpAttitudeDPREP, EcDpPrDumpEphemerisDPREP, EcDpPrPm1AttitudeDPREP_PGE, EcDpPrPm1FddEphemerisDPREP_PGE, EcDpPrEMGetAncHeaders, EcDpPrAuraEphemerisDPREP_PGE, EcDpPrAuraAttitudeDPREP_PGE	Other	SPRHW	Developed	Data Preprocessing manages attitude and ephemeris data preprocessing for inputs to PGEs. Data preprocessing is the preliminary processing or application of an operation on a data set that does not alter or modify scientific content of the data set. Preprocessing includes data set format changes by reordering the lower level byte structure, data set reorganization (ordering data items within and between physical files), and preparing additional metadata based on lower level metadata. PM1 and Aura Orbit Processing include reformatting the FDD definitive ephemeris data sets into the toolkit native format and the HDF format, respectively. DPREP generates the orbit metadata records for the data sets. AM1 DPREP processes L0 ancillary data to produce ephemeris and attitude files formatted for use by the toolkit and in HDF. PM1 and Aura uses “carry-out” attitude information and FDS generated ephemeris to produce attitude files formatted for use by the Toolkit and in HDF format.
Sybase ASE	Server	ACMHW	COTS	The Sybase ASE acts as a SQL server for the PDPS database.
EcDpPrGE	Other	SPRHW	Developed	The EcDpPrJobMgmt server initiates the EcDpPrGE 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 (cpu, ram, etc.) off-line and the computer resource is not available for PGEs.

EMD Baseline Information System (EBIS) Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the HWCI, and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

4.7.1.5 Processing Process Interface Descriptions

Table 4.7-5, provides descriptions of the interface events shown in the PRONG CSCI architecture diagrams.

Table 4.7-5. PRONG CSCI Process Interface Events (1 of 13)

Event	Event Frequency	Interface	Initiated By	Event Description
Request Management Services	At system startup or shutdown and for restarts	*Processes:* EcDpPrJobMgmt, EcDpPrJobMgmtClient	DAAC unique startup scripts	**System startup and shutdown -** Please refer to the release-related, current version of the Mission Operations Procedures for the EMD Project document (611) and the current EMD Project Training Material document (625), identified in Section 2.2.1 of this document.
DPR Control, Status & QA	Per data processing request	*Process:* EcDpPrJobMgmt *Library:* DpPrJM *Class:* DpPrScheduler	Operations Staff/ Operations Staff terminal	The **Operations staff** controls Data Processing Request (DPR) activity with the capability to cancel, suspend, resume, and modify a DPR. The M&O staff supports status collecting, PRONG hardware resource monitoring and Quality Assurance validating processes.
Delete Granules	Per operations request	*Processes:* EcDpPrJobMgmt EcDpPrJobMgmtClient	M&O staff	The **M&O staff** sends requests to the EcDpPrJobMgmt/EcDpPrJobMgmtClient to delete granules associated with cancelled DPRs.
Cancel Ground Event	Per operations request	*Processes:* EcDpPrJobMgmt EcDpPrJobMgmtClient	M&O staff	The **M&O staff** sends a request to the EcDpPrJobMgmt process to cancel ground events in AutoSys.
Change DPR ID	One per defined DPR	*Processes:* EcDpPrJobMgmt EcDpPrJobMgmtClient	M&O staff	The **M&O staff** can change the DPR ID of an existing DPR.
Change Max Concurrent Jobs for PGE Limits Table	Per Operations Staff Request	*Process:* EcDpPrJobMgmt EcDpPrJobMgmtClient	M&O staff	The **M&O staff** can add to or update values in the DpPrPgeLimits database table.
Change Min/Max DPRs for Job Class	Per operations staff request	*Process:* EcDpPrJobMgmt EcDpPrJobMgmtClient	M&O staff	The **M&O staff** can add to or update values in the DpPrClassSchedulingLimits database table.

Table 4.7-5. PRONG CSCI Process Interface Events (2 of 13)

Event	Event Frequency	Interface	Initiated By	Event Description
View Job Management DPR Queue	Per Operations Staff request.	*Process:* EcDpPrViewJobStates *Class:* DpPrListJobs	Operations Staff/ Operations Staff terminal	The **Operations staff** can view the job state via the EcDpPrViewJobStates process, as an aid in scheduling jobs.
Create Ground Event Job	One per defined ground event	*Process:* AutoSys (COTS)	*Process:* EcDpPrJobMgmt *Library:* DpPrJM *Class:* DpPrScheduler	The EcDpPrJobMgmt process sends a request to **AutoSys** to create a ground event job to perform maintenance activities on data processing resources.
Cancel Ground Event Job	One per defined ground event	*Process:* AutoSys (COTS)	*Process:* EcDpPrJobMgmt *Library:* DpPrJM *Class:* DpPrScheduler	The EcDpPrJobMgmt process sends a request to **AutoSys** to cancel a ground event job for the deletion of a ground event.
Create Job Box	One per DPR	*Process:* AutoSys (COTS)	*Process:* EcDpPrJobMgmt *Library:* DpPrJM *Class:* DpPrScheduler	The EcDpPrJobMgmt sends a request to **AutoSys** to create a job box when all DPR input data is available using Job Interface Language (JIL). If all the input data is not available, the EcDpPrJobMgmt stores the DPR in a priority-based queue.
Release Job Box	One per execution of a DPR in the job mgmt queue	*Process:* AutoSys (COTS)	*Process:* EcDpPrJobMgmt *Library:* DpPrJM *Class:* DpPrScheduler	The EcDpPrJobMgmt sends a request to release (or delete) jobs box in the **AutoSys** queue for a DPR awaiting execution.
Cancel Job Box	One per job box deletion	*Process:* AutoSys (COTS)	*Process:* EcDpPrJobMgmt *Library:* DpPrJM *Class:* DpPrScheduler	The EcDpPrJobMgmt deletes a job box in **AutoSys** and performs any cleanup when an operator requests a DPR to be canceled.

Table 4.7-5. PRONG CSCI Process Interface Events (3 of 13)

Event	Event Frequency	Interface	Initiated By	Event Description
Update PGE Information	One per update/retrieve data request	Sybase ASE Database (COTS)	*Processes:* EcDpPrJobMgmt, EcDpPrJobMgmtClient *Library:* DpPrRM	The EcDpPrJobMgmt and EcDpPrJobMgmtClient processes, using the DpPrRM library, request update and retrieval of data in the **Sybase ASE** database that defines a PGE and allows the PGE to be scheduled and executed by AutoSys. These processes request updates for granule information (location, size, etc.), processing status, and check-pointing information stored in the database.
Manage DPRs (create, release, cancel)	One per control request	*Process:* EcDpPrJobMgmt *Library:* DpPrJM *Class:* DpPrScheduler	*Processes:* EcPlWb, EcPlSubMgr, EcPlRpSi *Library:* PlCore1 *Classes:* PlWbScheduler, PlDpr	The **EcPlWb** process sends requests to the EcDpPrJobMgmt to create and cancel DPR jobs in AutoSys. The EcPlPREditor_IF process sends requests to delete DPRs in AutoSys to the EcDpPrJobMgmt process. The EcPlSubMgr process sends requests to the EcDpPrJobMgmt process to release DPR jobs in AutoSys.
Create Ground Event	One per ground resource	*Process:* EcDpPrJobMgmt *Library:* DpPrJM *Class:* DpPrScheduler	*Process:* EcPlWb *Class:* PlWbScheduler	The **EcPlWb** sends requests to the EcDpPrJobMgmt process to create ground events for maintenance activities on data processing resources.
Cancel Ground Event	One per ground resource	*Process:* EcDpPrJobMgmt *Library:* DpPrJM *Class:* DpPrScheduler	*Process:* EcPlRpSi	The **EcPlRpSi** process sends a request to the EcDpPrJobMgmt process to cancel ground events in AutoSys.
Initiate execution (control)	One per PGE job execution	*Process:* EcDpPrGE	*Process:* EcDpPrJobMgmt *Library:* DpPrJM *Class:* DpPrScheduler	The EcDpPrJobMgmt process initiates the **EcDpPrGE** process when ground events occur.

Table 4.7-5. PRONG CSCI Process Interface Events (4 of 13)

Event	Event Frequency	Interface	Initiated By	Event Description
Request Management Services	At system startup or shutdown and for restarts	*Process:* EcDpPrDeletion	DAAC unique startup scripts	**System startup and shutdown -** Please refer to the release-related, current version of the Mission Operations Procedures for the EMD Project document (611) and the current EMD Project Training Material document (625), identified in Section 2.2.1 of this document.
Job States	Per AutoSys Status update.	*Process:* EcDpPrViewJobStates *Class:* DpPrListJobs	*Process:* AutoSys (COTS)	The **AutoSys** provides the job state (completed, executing, or in a queue to be executed) to the EcDpPrViewJobStates process.
Initiate execution of job (control)	One per PGE job execution	***Process:*** EcDpPrEM *Library:* DpPrEM *Class:* DpPrExecutionManager ***Process:*** EcDpPrRunPGE	Process: AutoSys (COTS)	**AutoSys** initiates the execution of the EcDpPrEM and the EcDpPrRunPGE processes to control the preparation (data staging), execution, and archiving of higher level products, which are produced, and cleanup of each PGE run.

Table 4.7-5. PRONG CSCI Process Interface Events (5 of 13)

Event	Event Frequency	Interface	Initiated By	Event Description
Monitor status (status)	One per PGE job execution	*Processes:* EcDpPrAm1EdosAncillary, EcDpPrAm1EdosEphAttDPREP_PGE, EcDpPrAm1EdosEphemerisRepair, EcDpPrAm1FddAttitudeDPREP_PGE, EcDpPrAm1FddEphemerisDPREP_PGE, EcDpPrAm1ToolkitToHdf, EcDpPrDumpAttitudeDPREP, EcDpPrDumpEphemerisDPREP, EcDpPrPm1FddEphemerisDPREP_PGE, EcDpPrEMGetAncHeaders, EcDpPrPm1AttitudeDPREP_PGE, EcDpPrAuraEphemerisDPREP_PGE, EcDpPrAuraAttitudeDPREP_PGE, Other PGE executables created by science teams	*Process:* EcDpPrRunPGE	The EcDpPrRunPGE process, apart from initiating the PGE process, also monitors the PGE's computer resources. If the PGE's computer resources exceed its expected usage an alarm is sent to the AutoSys. The EcDpPrAm1EdosAncillary, EcDpPrAm1EdosEphAttDPREP_PGE, EcDpPrAm1EdosEphemerisRepair, EcDpPrAm1FddAttitudeDPREP_PGE, EcDpPrAm1FddEphemerisDPREP_PGE, EcDpPrAm1ToolkitToHdf, EcDpPrDumpAttitudeDPREP, EcDpPrDumpEphemerisDPREP, EcDpPrPm1FddEphemerisDPREP_PGE, EcDpPrEMGetAncHeaders and EcDpPrPm1AttitudeDPREP_PGE, EcDpPrAuraEphemerisDPREP_PGE, EcDpPrAuraAttitudeDPREP_PGE and other PGE processes send status to the EcDpPrRunPGE process.

Table 4.7-5. PRONG CSCI Process Interface Events (6 of 13)

Event	Event Frequency	Interface	Initiated By	Event Description
Initiate execution (control)	One per PGE job execution	*Processes:* EcDpPrAm1EdosAncillary, EcDpPrAm1EdosEphAttDPREP_PGE, EcDpPrAm1EdosEphemerisRepair, EcDpPrAm1FddAttitudeDPREP_PGE, EcDpPrAm1FddEphemerisDPREP_PGE, EcDpPrAm1ToolkitToHdf, EcDpPrDumpAttitudeDPREP, EcDpPrDumpEphemerisDPREP, EcDpPrPm1FddEphemerisDPREP_PGE, EcDpPrEMGetAncHeaders, EcDpPrPm1AttitudeDPREP_PGE, EcDpPrAuraEphemerisDPREP_PGE, EcDpPrAuraAttitudeDPREP_PGE, Other PGE executables created by science teams	*Process:* EcDpPrRunPGE	The EcDpPrRunPGE provides a buffer between AutoSys and the PGE. This serves as a wrapper to the PGE process, initiates the PGE execution and captures the PGE's exit status. The EcDpPrRunPGE initiates the **EcDpPrAm1EdosAncillary**, **EcDpPrAm1EdosEphAttDPREP_PGE**, **EcDpPrAm1EdosEphemerisRepair**, **EcDpPrAm1FddAttitudeDPREP_PGE**, **EcDpPrAm1FddEphemerisDPREP_PGE**, **EcDpPrAm1ToolkitToHdf**, **EcDpPrDumpAttitudeDPREP**, **EcDpPrDumpEphemerisDPREP**, **EcDpPrPm1FddEphemerisDPREP_PGE**, **EcDpPrEMGetAncHeaders**, **EcDpPrPm1AttitudeDPREP_PGE**, **EcDpPrAuraEphemerisDPREP_PGE,** and **EcDpPrAuraAttitude** processes.

Table 4.7-5. PRONG CSCI Process Interface Events (7 of 13)

Event	Event Frequency	Interface	Initiated By	Event Description
Request Communications Support	One service per request.	*Process:* EcCsIdNameServer *Libraries:* EcPf, Middleware, FoNs, FoIp, oodce *Classes:* EcPfManagedServer, EcPfClient, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy *Library (Common):* EcUr *Class:* EcUrServerUR *Library:* event *Class:* EcLgErrorMsg *Process:* EcCsRegistry *Library:* EcCsRegistry *Class:* EcRgRegistryServer_C	*Process:* EcDpPrEM *Library:* DpPrDM *Class:* DpPrDataManager	The **DCCI CSCI** provides a library of services available to each SDPS and CSMS CSCI. The CSCI services required to perform specific assignments are requested from the DCCI CSCI. These services include: • CCS Middleware Support • Database Connection Services • Network & Distributed File Services • Bulk Data Transfer Services • Name/Address Services • Password Services • Server Request Framework (SRF) • Universal Reference (UR) • Error/Event Logging • Fault Handling Services • Mode Information • Query Registry - Retrieving the requested configuration attribute-value pairs from the Configuration Registry

Table 4.7-5. PRONG CSCI Process Interface Events (8 of 13)

Event	Event Frequency	Interface	Initiated By	Event Description
Import Location Information	As required for processing	*Process:* EcDpPrEM *Library:* DpPrDM *Class:* DpPrDataManager	*Process:* EcCsIdNameServer *Library:* EcPf, Middleware, FoNs, FoIp, oodce *Classes:* EcPfManagedServer, EcPfClient, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy	The EcDpPrEM requests server location information from the **EcCsIdNameServer**.
Export Location Information	Once at system start up and after each failure recovery	*Process:* EcCsIdNameServer *Library:* EcPf, Middleware, FoNs, FoIp, oodce *Classes:* EcPfManagedServer, EcPfClient, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy	*Process:* EcDpPrEM *Library:* DpPrDM *Class:* DpPrDataManager	The EcDpPrEM places physical and logical location information in the **EcCsIdNameServer**.
Return Configuration Parameters	One per configuration registry query	*Process:* EcDpPrEM *Library:* DpPrDM *Class:* DpPrDataManager	*Process:* EcCsRegistry *Library:* EcCsRegistry *Class:* EcRgRegistryServer_C	The **EcCsRegistry** returns the requested configuration parameters to the EcDpPrEM.

Table 4.7-5. PRONG CSCI Process Interface Events (9 of 13)

Event	Event Frequency	Interface	Initiated By	Event Description
On-Demand Failure Message	Only in the event of On-Demand Processing Request failure	User *Library:* CsEmMailRelA *Class:* CsEmMailRelA	*Process:* EcDpPrEM *Library:* DpPrEM *Class:* DpPrExecutionManager	The EcDpPrEM process sends an e-mail message to the **originator** (**a user**) of an On-Demand Processing Request if any DPR generated by that request fails.
Update Request Status	One for each PGE run to satisfy the requested On-Demand Product	*Process:* EcMsAcOrderSrvr *Libraries:* MsAcSrv, MsAcComm *Class:* EcAcOrderCMgr	*Process:* EcDpPrEM *Library:* DpPrEM *Class:* DpPrExecutionManager	The EcDpPrEM process updates the status of an On-Demand Processing Request, via the **EcMsAcOrderSrvr**, when a DPR for the order completes or fails processing.
Request/Allocate Disk Space	One per disk request	Sybase ASE (COTS)	*Process:* EcDpPrEM *Libraries:* DpPrRM, DpPrDM *Class:* DpPrResourceManager	The EcDpPrEM requests disk space via the **Sybase ASE** by using the DpPrRM and DpPrDM library software for each input granule that needs to be staged to the local processing disk and output granule needed by a PGE.
Update PGE Information	One per update/retrieve data request	Sybase Database (COTS)	*Process:* EcDpPrEM *Libraries:* DpPrRM, DpPrDM	The EcDpPrEM process, using the DpPrRM and DpPrDM libraries, requests update and retrieval of data in the **Sybase ASE** database that defines a PGE and allows the PGE to be scheduled and executed by AutoSys. These processes request updates for granule information (location, size, etc.), processing status, and check-pointing information stored in the database.
Request CPU and RAM Allocation	One per PGE job execution	Sybase ASE (COTS)	*Process:* EcDpPrEM *Libraries:* DpPrRM, DpPrDM *Class:* DpPrResourceManager	The EcDpPrEM process requests CPU and RAM allocations via the **Sybase ASE** by using the DpPrRM library software for each PGE based on values entered at SSIT.

Table 4.7-5. PRONG CSCI Process Interface Events (10 of 13)

Event	Event Frequency	Interface	Initiated By	Event Description
Request Granule Deletes	Per granule delete request	*Process:* EcDpPrDeletion *Class:* DpDeletion	*Process:* EcDpPrEM *Libraries:* DpPrDssIF, DpPrDM *Class:* DpPrDSSInterface	The EcDpPrEM sends requests to the EcDpPrDeletion process to delete interim granules a PGE had used in processing or after a defined storage period has elapsed. The EcDpPrDeletion process submits the request to the **EcDsScienceDataServer**.
Delete Old Granule	One per granule	*Process:* EcDpPrEM *Library:* DpPrDM *Class:* DpPrGranuleLocator	*Process:* EcDpPrDeletion *Class:* DpDeletion	The **EcDpPrDeletion** process gets a request from the EcPlWb, via the EcDpPrEM, to delete a granule from one of the local science processing disks. The EcDpPrDeletion process uses the DpPrDM library software to delete the files from the disk and update the PDPS database.
Delete Granules	Per granule delete request	*Process:* EcDpPrDeletion *Class:* DpDeletion	*Process:* EcPlPREditor_IF *Library:* PlCore1 *Classes:* PlWbScheduler, PlDpr	The **EcPlPREditor_IF** sends requests to the EcDpPrDeletion process to delete granules associated with a DPR, which has been cancelled.
Request Granule Deletion	Per granule delete request	*Process:* EcDsScienceDataServer *Library:* DsCl *Classes:* DsClRequest, DsClCommand, DsClESDTReferenceCollector	*Process:* EcDpPrDeletion *Class:* DpDeletion	The EcDpPrDeletion process submits the request to delete interim granules a PGE had used in processing or after a defined storage period has elapsed to the **EcDsScienceDataServer**.
Acquire PGE Tar file	One per tar file acquire	*Process:* EcDsScienceDataServer *Library:* DsCl *Classes:* DsClRequest, DsClCommand	*Process:* EcDpPrEM *Library:* DpPrDssIF *Class:* DpPrDSSInterface	The EcDpPrEM acquires a tar file for any PGE not currently local to the science processor from the **EcDsScienceDataServer**. The tar file is removed from the tape archive and used during PGE execution.

Table 4.7-5. PRONG CSCI Process Interface Events (11 of 13)

Event	Event Frequency	Interface	Initiated By	Event Description
Request MCF	One per MCF request	*Process:* EcDsScienceDataServer *Library:* DsCl *Classes:* DsClDescriptor	*Process:* EcDpPrEM *Library:* DpPrDssIF *Class:* DpPrDSSInterface	The EcDpPrEM requests a MCF template for each output data type for specific PGEs from the **EcDsScienceDataServer** and the MCF template is populated with metadata from the output granule.
Insert Failed PGE Tar file	One per unsuccessful PGE execution	*Process:* EcDsScienceDataServer *Library:* DsCl *Classes :* DsClRequest, DsClCommand	*Process:* EcDpPrEM *Library:* DpPrDssIF *Class:* DpPrDSSInterface	After an unsuccessful execution of a PGE, the EcDpPrEM obtains the Tar file containing the PGE log files, core dump (if any), PCF and other files, and requests the files be inserted into the **EcDsScienceDataServer** for permanent archive.
Request Data Insert	One granule per request	*Process:* EcDsScienceDataServer *Library:* DsCl *Classes :* DsClRequest, DsClCommand, DsClESDTReferenceCollector	*Process:* EcDpPrEM *Libraries :* DpPrDssIF, DpPrDM *Class:* DpPrDSSInterface	After the PGE has successfully completed executing, the EcDpPrEM sends insert requests for the **EcDsScienceDataServer** to store the output granules into the SDSRV inventory/archives.
Retrieve Data	One granule per request	*Process:* EcDsScienceDataServer *Library:* DsCl *Classes :* DsClRequest, DsClCommand, DsClESDTReferenceCollector	*Process:* EcDpPrEM *Libraries :* DpPrDssIF, DpPrDM *Class:* DpPrDSSInterface	The EcDpPrEM acquires the input granules needed by a PGE not currently local on a science processor from the **EcDsScienceDataServer** by sending a UR for each granule.

Table 4.7-5. PRONG CSCI Process Interface Events (12 of 13)

Event	Event Frequency	Interface	Initiated By	Event Description
Insert Product History Tar Files	One per successful PGE execution	*Process:* EcDsScienceDataServer *Library:* DsCl *Classes :* DsClRequest, DsClCommand	*Process:* EcDpPrEM *Libraries:* DpPrDssIF, DpPrDM *Class:* DpPrDSSInterface	After the PGE has successfully completed executing and archiving the resulting outputs, the EcDpPrEM requests the PGE Production History Tar file be inserted into the **EcDsScienceDataServer** for permanent archive.
Return PGE Tar File	One per request	*Process:* EcDpPrEM *Library:* DpPrDssIF *Class:* DpPrDSSInterface	*Process:* EcDsScienceDataServer *Library:* DsCl *Classes :* DsClRequest, DsClCommand	After an unsuccessful execution of a PGE, the EcDpPrEM obtains the Tar file containing the PGE log files, core dump (if any), Process Control File (PCF) and other files, and requests the files be inserted into the **EcDsScienceDataServer** for permanent archive.
Return MCF Template	One per set of external data received by EMD	*Process:* EcDpPrEM *Library:* DpPrDssIF *Class:* DpPrDSSInterface	*Process:* EcDsScienceDataServer *Library:* DsCl *Class:* DsClDescriptor	The **EcDsScienceDataServer** provides the MCF template as part of the GetMCF service call to the EcDpPrEM process.
Return Product	One product per request	*Process:* EcDpPrEM *Library:* DpPrDssIF *Class:* DpPrDSSInterface	*Process:* EcDsScienceDataServer *Library:* DsCl *Class:* DsClDescriptor	The data granules requested by the EcDpPrEM are sent from the **EcDsScienceDataServer**.
Update Product Metadata	One per metadata product update	*Process:* EcDsScienceDataServer *Library:* DsCl *Classes:* DsClCommand, DsClRequest, DsClESDTReferenceCollector	*Process:* EcDpPrQaMonitorGUI *Library:* DpPrQaMonitor *Class:* DpPrQAGranuleQaFlags	The EcDpPrQaMonitorGUI provides the operator with capabilities to update product metadata in the **EcDsScienceDataServer**.

Table 4.7-5. PRONG CSCI Process Interface Events (13 of 13)

Event	Event Frequency	Interface	Initiated By	Event Description
Query Data	One per query	*Process:* EcDsScienceDataServer *Library:* DsCl *Class:* DsClESDTReferenceCollector	*Process:* EcDpPrQaMonitorGUI *Library:* DpPrQaMonitor *Class:* DpPrQaDataGranule	The EcDpPrQaMonitorGUI submits requests of this type to the **EcDsScienceDataServer**. It searches the archive for granules that match the user-supplied selection criteria: data type and begin/end date. Results are displayed to the user.
Retrieve Data	One per request	*Process:* EcDsScienceDataServer *Library:* DsCl *Class:* DsClAcquireCommand	*Process:* EcDpPrQaMonitorGUI *Library:* DpPrQaMonitor *Class:* DpPrQaMonitor	The EcDpPrQaMonitorGUI submits requests of this type to the **EcDsScienceDataServer**. It transfers a granule from the Science Data archive to the user's host machine.
Retrieve Product History	One per request	*Process:* EcDsScienceDataServer *Library:* DsCl *Class:* DsClAcquireCommand	*Process:* EcDpPrQaMonitorGUI *Library:* DpPrQaMonitor *Class:* DpPrQaMonitor	The EcDpPrQaMonitorGUI submits requests of this type to the **EcDsScienceDataServer**. It transfers the Production History tar file from the Science Data archive to the user's host machine.

4.7.1.6 Processing Data Stores

Table 4.7-6 provides descriptions of the data stores shown in the PRONG CSCI architecture diagram.

Table 4.7-6. PRONG CSCI Data Stores

Data Store	Type	Functionality
PDPS database	Database	The PDPS database is replicated within the same site and holds the persistent data for PDPS. The persistent data includes, but is not limited to, resource information entered with the Resource Planning utilities, PGE and data type information entered at SSIT, Production Requests, Data Processing Requests and Data Granule information entered using the Production Request Editor and plan information entered using the Production Planning Workbench.

4.7.2 Algorithm Integration and Test Tools Software Description

4.7.2.1 Functional Overview

The DAAC Integration and Test (I&T) team to use the Algorithm Integration and Test Tools (AITTL):

- Retrieve science software and submit it for configuration control
- Compile and link the delivered source files
- Execute test cases
- Provide error diagnosis using interactive debuggers, and data viewers
- Collect resource metrics of CPU time, memory, and disk space to build the PGE Profile and thus enable the PLANG and PRONG CSCIs to execute the science software
- Update the system databases after the science software completes acceptance testing

The AITTL tools are in the following categories:

- Compilers, linkers, debuggers, and other development and operating system tools
- Tools for viewing science software documentation
- Tools for checking compliance of science software to Earth Science Data and Information System (ESDIS)-specified coding standards
- Code analysis tools (e.g., Forte Developer, ProDev Workshop)
- Data viewing tools (e.g., EOSView)
- Tools for comparing HDF files
- Tools for comparing Binary files
- Tools for providing executable profiles (to get a PGE performance profile)
- Tools to register the science software with the Planning and Data Processing Subsystems
- Tools to add and update Science Software Archive Packages (SSAPs) in the Data Server
- Tools for writing reports and maintaining the I&T logs
- Tools for checking Process Control Files and for prohibited functions
- Tools to display product metadata

For information on science software integration and test procedures, see Science User's Guide and Operations Procedures Handbook for the ECS Project (205-CD-002-001) Part 4, and Science Software Integration and Test (JU9403V1). For information on the ESDIS science software coding standards and guidelines, see Data Production Software and Science Computing Facility (SCF) Standards and Guidelines (423-16-01).

Note: The directory structure for the AITTL software has the name SSIT and not AITTL. The use of the SSIT directory structure name is to denote the main purpose of the

Algorithm Test Tools as tools to support the Science Software Integration and Test activities as part of the SDPS data processing.

4.7.2.2 Algorithm Integration and Test Tools Context

Figure 4.7-4 is the Algorithm Integration and Test Tool (AITTL) CSCI context diagram. The diagram shows the events sent to the AITTL CSCI and the events the AITTL CSCI sends to other EMD subsystems. **Note: System startup and shutdown –** Please refer to the release-related, current version of the Mission Operations Procedures for the EMD Project document (611) and the current EMD Project Training Material document (625), identified in Section 2.2.1 of this document.

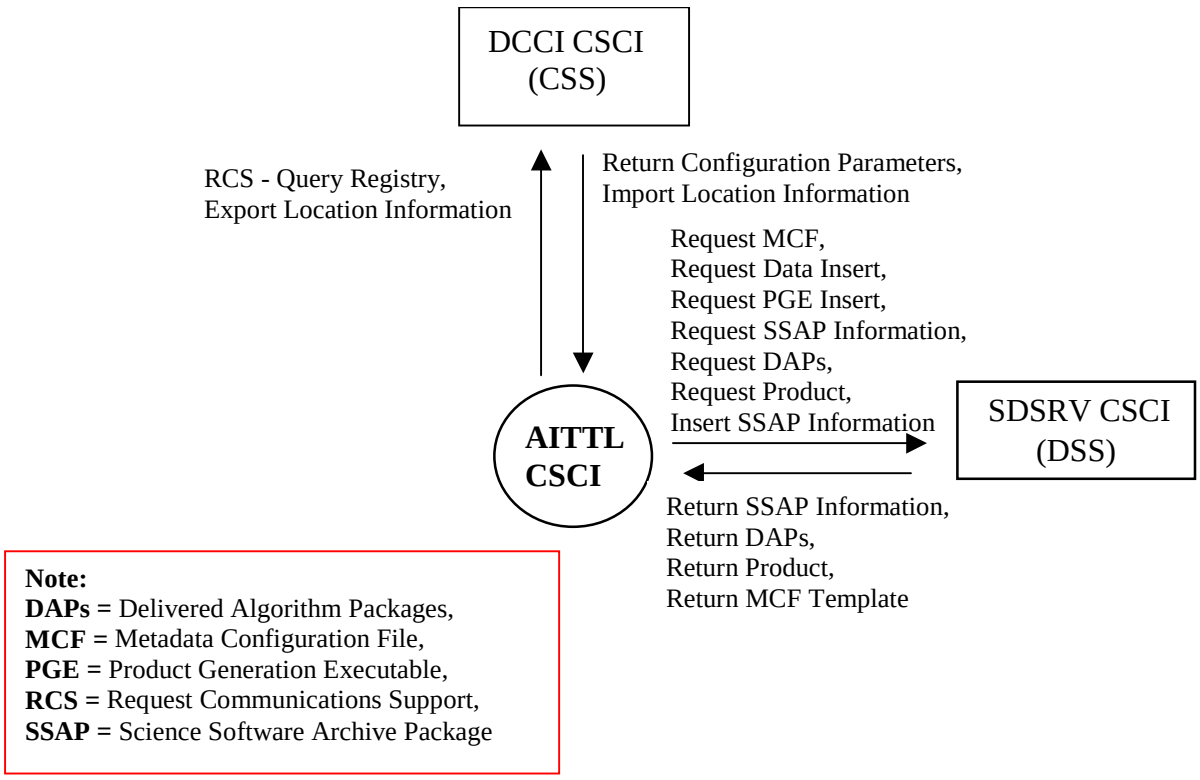


Figure 4.7-4. AITTL Context Diagram

Table 4.7-7 provides descriptions of the interface events shown in the AITTL Context Diagram.

Table 4.7-7. AITTL Interface Events (1 of 2)

Event	Interface Event Description
Return Configuration Parameters	The **DCCI CSCI** returns the requested configuration parameters to the AITTL CSCI.
Import Location Information	The AITTL CSCI requests server location information from the **DCCI CSCI** (CCS Name Server).
Request MCF	The AITTL CSCI sends requests to the **SDSRV CSCI** for the MCF template for use during SSIT. The PRONG CSCI also requests the MCF template from the SDSRV CSCI prior to a data insert request.
Request Data Insert	The AITTL CSCI puts various types of data into the SDSRV inventory (**SDSRV CSCI**), from SSAP information to Static files and PGE executables. In response, the AITTL CSCI gets the results of the insert and the UR to perform an acquire request.
Request PGE Insert	The AITTL CSCI sends requests to the **SDSRV CSCI** to insert data that defines a PGE and allows it to be scheduled and executed.
Request SSAP Information	The AITTL CSCI sends requests to the **SDSRV CSCI** for SSAP information, including names of existing SSAPs and the information associated with a specific SSAP. In response, the SDSRV CSCI sends lists of SSAPs and related information.
Request DAPs	The AITTL CSCI requests DAPs based on Urs from the **SDSRV CSCI**. The requested DAPs are placed on a local AITTL disk.
Request Product	The AITTL CSCI sends requests, to the **SDSRV CSCI**, for particular data granules to be pushed, via the FTP service, onto the DPS science processor as input for data processing or for SSIT work.
Insert SSAP Information	The M&O Staff sends requests to the **SDSRV CSCI** to insert SSAP information, via the SSAP GUI, including SSAP name, SSAP version number, PGE name, PGE version number, and SSAP Acceptance Date.
Return SSAP Information	The **SDSRV CSCI** sends lists of SSAPs and related information to the AITTL CSCI.
Return DAPs	The **SDSRV CSCI** places the DAPs on a local AITTL CSCI disk.
Return Product	The data granules requested by the AITTL CSCI are sent from the **SDSRV CSCI**.
Return MCF Template	The AITTL CSCI receives the MCF template to populate, from the **SDSRV CSCI**, as part of the GetMCF service call.

Table 4.7-7. AITTL Interface Events (2 of 2)

Event	Interface Event Description
Request Communications Support	The **DCCI CSCI** provides a library of services available to each SDPS and CSMS CSCI. The CSCI services required to perform specific assignments are requested from the DCCI CSCI. These services include: • CCS Middleware Support • Database Connection Services • Network & Distributed File Services • Bulk Data Transfer Services • Name/Address Services • Password Services • Server Request Framework (SRF) • Universal Reference (UR) • Error/Event Logging • Fault Handling Services • Mode Information • Query Registry – Retrieving the requested configuration attribute-value pairs from the Configuration Registry
Export Location Information	The CCS Middleware CSC stores physical and logical location information received from the AITTL CSCI in the **DCCI CSCI** (CCS Name Server).

4.7.2.3 Algorithm Integration and Test Tools Architecture

Figure 4.7-5 is the AITTL CSCI architecture diagrams. The diagrams show the events that launch the AITTL CSCI processes and the events the AITTL CSCI processes send to processes in other CSCIs.

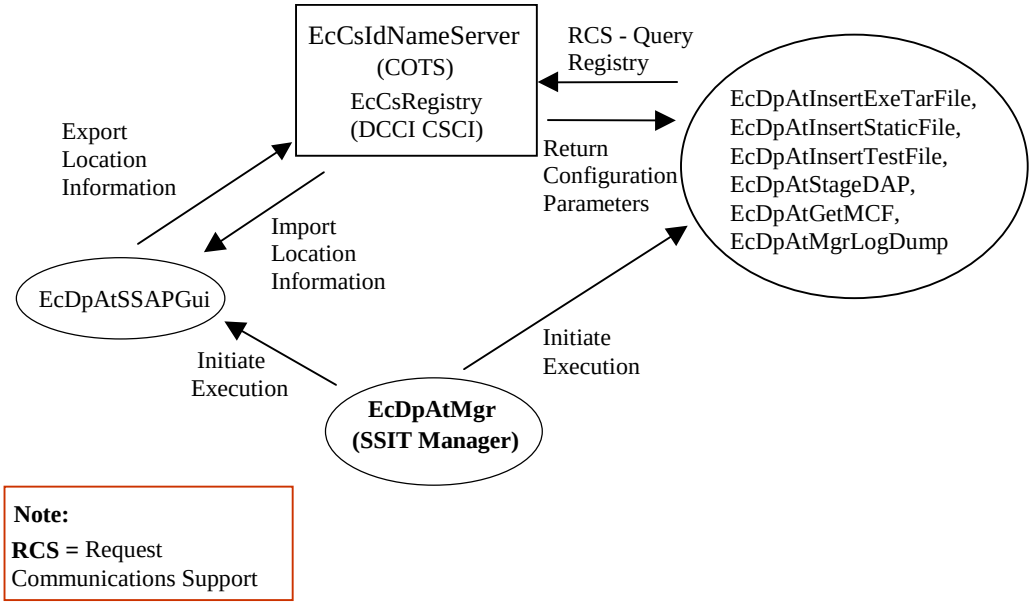


Figure 4.7-5. AITTL CSCI Architecture Diagram

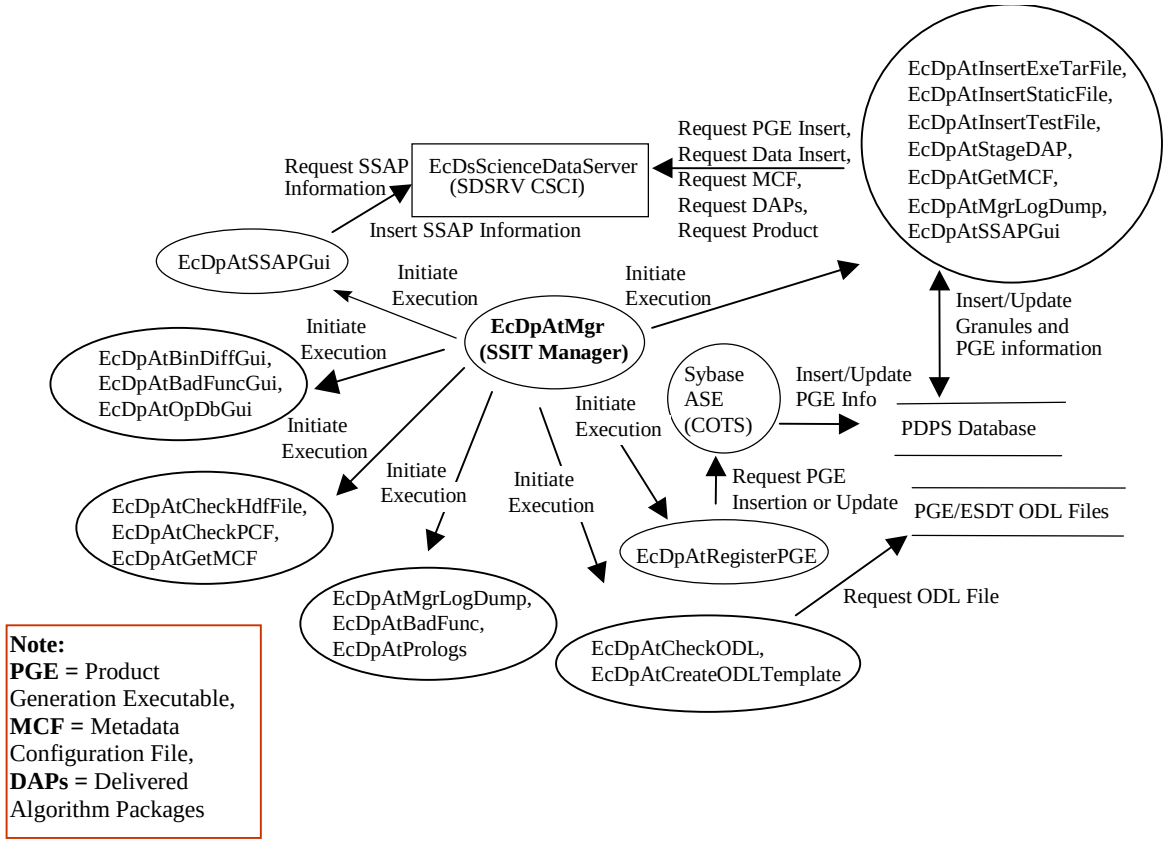


Figure 4.7-5. AITTL CSCI Architecture Diagram (cont.)

4.7.2.4 Algorithm and Test Tools Process Description

Table 4.7-8 provides descriptions of the processes shown in the AITTL CSCI architecture diagram.

Table 4.7-8. AITTL Processes (1 of 3)

Process	Type	Hardware CI	COTS / Developed	Functionality
EcDpAtSSAPGui	GUI	AITHW	Developed	This GUI allows the M&O staff to create, update and delete SSAPs and to acquire information about an SSAP for modification or testing (such as test plans).

Table 4.7-8. AITTL Processes (2 of 3)

Process	Type	Hardware CI	COTS / Developed	Functionality
EcDpAtMgr	GUI	AITHW	Developed	This application provides menus to launch other SSIT applications and provides a checklist to users for marking each SSIT function as completed.
EcDpAtRegisterPGE, EcDpAtCheckODL, EcDpAtCreateODLTemplate, EcDpAtOpDbGui	GUI and cmd line interface (I/F)	AITHW	Developed	This application group allows a PGE to be defined in the PDPS database. ODL is read and checked by the tools and translated into the fields defining a PGE in the PDPS database. If the ODL files are valid, each row already existing in the PDPS database is updated and non-existent rows are inserted. The SSIT personnel input performance information via a GUI.
EcDpAtBinDiffGui, hdiff	GUI and cmd line I/F	AITHW	Developed and COTS	This application group supports data file viewing and comparisons. The group includes EOSView, the COTS language IDL, and tools to compare binary and HDF files. The shell programs EcDpAtCheckHdfFile and EcDpAtMgrXdiff are used to assist with the viewing and comparisons.
EcDpAtBadFunc, EcDpAtBadFuncGui, EcDpAtPrologs, EcDpAtCheckPCF	GUI	AITHW	Developed and COTS	This application group checks the source code for PGEs and PGE PCFs for errors or prohibited functions. The Sparc Works COTS product is included for editing and debugging functions and a checker is provided for use during testing. Also provided is a checker to monitor the software for any prohibited calls.
EcDpAtInsertExeTarFile, EcDpAtInsertStaticFile, EcDpAtInsertTestFile, EcDpAtStageDAP, EcDpAtGetMCF	cmd line I/F	AITHW	Developed	This application group provides mechanisms to insert and acquire data items from the EcDsScienceDataServer in the SDSRV CSCI. Static Files and PGE executables are inserted into the SDSRV archives by these tools, and the respective PDPS database tables are updated with the results. The Delivery Archive Package (DAP) and MCFs are acquired via these tools for command line testing the PGE.

Table 4.7-8. AITTL Processes (3 of 3)

Process	Type	Hardware CI	COTS / Developed	Functionality
EcDpAtMgrLogDump	Cmd line I/F	AITHW	Developed	Command line interface to dump the SSIT checks list database to a file that can be sent to the printer.
Sybase ASE	Server	ACMHW	COTS	The Sybase ASE is the interface between AITTL processes and the PDPS database for PGE insertion and update of PGE information in the PDPS database to support Data Processing activities.

EBIS Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the HWCI, and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

4.7.2.5 Algorithm and Test Tools Process Interface Descriptions

Table 4.7-9 provides descriptions of the interface events shown in the AITTL CSCI architecture diagrams.

Table 4.7-9. AITTL Process Interface Events (1 of 7)

Event	Event Frequency	Interface	Initiated By	Event Description
Request Management Services	At system startup or shutdown and for restarts	*Process:* EcDpAtMgr	DAAC unique startup scripts	**System startup and shutdown** – Please refer to the release-related, current version of the Mission Operations Procedures for the EMD Project document (611) and the current EMD Project Training Material document (625), identified in Section 2.2.1 of this document.
Request Communications Support	One service per request	*Process:* EcCsIdNameServer *Libraries:* EcPf, Middleware, FoNs, FoIp, oodce *Classes:* EcPfManagedServer, EcPfClient, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy *Library (Common):* EcUr *Class:* EcUrServerUR *Library:* event *Class:* EcLgErrorMsg *Process:* EcCsRegistry *Library:* EcCsRegistry *Class:* EcRgRegistryServer_C	*Processes:* EcDpAtSSAPGui, EcDpAtInsertExeTarFile, EcDpAtInsertStaticFile, EcDpAtInsertTestFile, EcDpAtStageDAP, EcDpAtGetMCF *Library:* DpAtDsrv	The **DCCI CSCI** provides a library of services available to each SDPS and CSMS CSCI. The CSCI services required to perform specific assignments are requested from the DCCI CSCI. These services include: • CCS Middleware Support • Database Connection Services • Network & Distributed File Services • Bulk Data Transfer Services • Name/Address Services • Password Services • Server Request Framework (SRF) • Universal Reference (UR) • Error/Event Logging • Fault Handling Services • Mode Information • (Query Registry) Retrieving the requested configuration attribute-value pairs from the Configuration Registry

Table 4.7-9. AITTL Process Interface Events (2 of 7)

Event	Event Frequency	Interface	Initiated By	Event Description
Return Configuration Parameters	One set per request	*Processes:* EcDpAtInsertExeTarFile, EcDpAtInsertStaticFile, EcDpAtInsertTestFile, EcDpAtStageDAP, EcDpAtGetMCF, EcDpAtMgrLogDump	*Process:* EcCsRegistry *Library:* EcCsRegistry *Class:* EcRgRegistryServer_C	The **EcCsRegistry** returns the attribute-value pairs (configuration parameters) to the EcDmDictServer upon request.
Initiate Execution	One per tool initialization	UNIX system calls	*Process:* EcDpAtMgr *Library:* DpAtDsrv *Class:* DpAtMgrLogGuiMainWindow	The EcDpAtMgr initiates the tools (**EcDpAtBinDiffGui, EcDpAtBadFuncGui, EcDpAtOpDbGui, EcDpAtCheckHdfFile, EcDpAtCheckPCF, EcDpAtGetMCF, EcDpAtMgrLogDump, EcDpAtBadFunc, EcDpAtPrologs, EcDpAtCheckODL, EcDpAtCreateODLTemplate, EcDpAtRegisterPGE, EcDpAtInsertExeTarFile, EcDpAtInsertStaticFile, EcDpAtInsertTestFile, and EcDpAtStageDAP**) and the GUI interface (**EcDpAtSSAPGui**) from a menu.

Table 4.7-9. AITTL Process Interface Events (3 of 7)

Event	Event Frequency	Interface	Initiated By	Event Description
Export Location Information	Once at system startup and after each failure recovery	*Process:* EcCsIdNameServer *Library:* EcPf, Middleware, FoNs, FoIp, oodce *Classes:* EcPfManagedServer, EcPfClient, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy	*Process:* EcDpAtSSAPGui	The EcDpAtSSAPGui places physical and logical location information in the **EcCsIdNameServer**.
Import Location Information	As required for processing	*Process:* EcDpAtSSAPGui	*Process:* EcCsIdNameServer *Library:* EcPf, Middleware, FoNs, FoIp, oodce *Classes:* EcPfManagedServer, EcPfClient, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy	The EcDpAtSSAPGui requests server location information from the **EcCsIdNameServer**.

Table 4.7-9. AITTL Process Interface Events (4 of 7)

Event	Event Frequency	Interface	Initiated By	Event Description
Request PGE Insert	One per insert request	*Process:* EcDsScienceDataServer *Library:* DsCl *Classes :* DsClRequest, DsClCommand	*Processes:* EcDpAtInsertExeTarFile, EcDpAtInsertStaticFile, EcDpAtInsertTestFile *Library:* PlCore2 *Classes:* DpAtDsrv, PlResourceRequirement	The EcDpAtInsertExeTarFile, EcDpAtInsertStaticFile, and EcDpAtInsertTestFile send PGE insert requests to the **EcDsScienceDataServer** for data that defines a PGE and allows it to be scheduled and executed.
Request Data Insert	One per data put into the archive	*Process:* EcDsScienceDataServer *Library:* DsCl *Classes :* DsClRequest, DsClCommand	*Processes:* EcDpAtInsertTestFile, EcDpAtInsertExeTarFile, EcDpAtInsertStaticFile, EcDpAtSSAPGui *Library:* DpAtDsrv *Class:* DpAtDsrv	The EcDpAtInsertStaticFile, EcDpAtInsertTestFile, EcDpAtInsertExeTarFile, and EcDpAtSSAPGui processes send requests to the **EcDsScienceDataServer** to put various types of data into the archive, from SSAP information to Static files and PGE executables.
Request MCF	One per MCF request	*Process:* EcDsScienceDataServer *Library:* DsCl *Class:* DsClDescriptor	*Process:* EcDpAtGetMCF *Library:* DpAtDsrv *Class:* DpAtDsrv	The EcDpAtGetMCF process sends a request for a MCF template to the **EcDsScienceDataServer**. In response, the MCF template is returned and populated.
Request DAPs	One per DAPs request	*Process:* EcDsScienceDataServer *Library:* DsCl *Classes:* DsClESDTReferenceCollector, DsClRequest, DsClCommand	*Process:* EcDpAtStageDAP *Library:* DpAtDsrv *Class:* DpAtDsrv	The EcDpAtStageDAP requests DAPs from the SDSRV Archives (at the **EcDsScienceDataServer**) based on the UR. In response, the DAPs are returned and stored on the local AITTL disk.

Table 4.7-9. AITTL Process Interface Events (5 of 7)

Event	Event Frequency	Interface	Initiated By	Event Description
Request Product	One per user request	Process: EcDsScienceDataServer *Library:* DsDdSSh *Classes:* DsDdScheduler, DsDdRequestMgrReal	*Process:* EcDpAtStageDAP *Library:* DpAtDsrv *Class:* DpAtDsrv	The EcDpAtStageDAP sends requests to the **EcDsScienceDataServer** for particular data granules to be pushed, via the FTP service, onto the DPS science processor as input for data processing or for SSIT work.
Return MCF Template	One per set of external data received by EMD	*Process:* EcDpAtGetMCF *Library:* DpPrDssIF *Class:* DpPrDSSInterface	*Process:* EcDsScienceDataServer *Library:* DsCl *Class:* DsClDescriptor	The **EcDsScienceDataServer** provides the MCF template as part of the GetMCF service call to the EcDpAtGetMCF process.
Return DAPs	One DAP per request	*Process:* EcDpAtStageDAP *Library:* DpAtDsrv *Class:* DpAtDsrv	*Process:* EcDsScienceDataServer *Library:* DsCl *Classes:* DsClESDTReferenceCollector, DsClRequest, DsClCommand	The **EcDsScienceDataServer** returns DAPs from the SDPS archives to the EcDpAtStageDAP, which stores the DAPs on the local AITTL disk.
Return SSAP Information	One per SSAP information request	*Process:* EcDpAtSSAPGui *Libraries:* DpAtSSAP, DpAtDsrv *Classes:* DpAtSSAPManager, DpAtDsrv	*Process:* EcDsScienceDataServer *Library:* DsCl *Classes:* DsClESDTReferenceCollector, DsClRequest, DsClCommand	The **EcDsScienceDataServer** returns information about SSAPs, including names of existing SSAPs and the components associated with a specific SSAP to the EcDpAtSSAPGui.

Table 4.7-9. AITTL Process Interface Events (6 of 7)

Event	Event Frequency	Interface	Initiated By	Event Description
Return Product	One per product order request	*Processes:* EcDpAtInsertExeTarFile, EcDpAtInsertStaticFile, EcDpAtInsertTestFile *Library:* DpAtDsrv *Class:* DpAtDsrv	*Process:* EcDsScienceDataServer *Library:* DsCl *Classes:* DsClRequest, DsClCommand	The **EcDsScienceDataServer** returns data granules to the EcDpAtInsertExeTarFile, EcDpAtInsertStaticFile and EcDpAtInsertTestFile processes as input for data processing or for SSIT work.
Insert/ Update Granules and PGE information	One per insert/update of granule information	PDPS Database	*Processes:* EcDpAtInsertStaticFile, EcDpAtInsertExeTarFile *Library:* PlCore1 *Classes:* DpAtDsrv, PlDataGranule	Insert/update granule information in the **PDPS Database**: • Received from a static granule insert request • About a modified, existing PGE
Insert/Update PGE Info	One PGE per request	PDPS Database	Sybase ASE (COTS)	The **Sybase ASE** inserts or updates PGE information in the PDPS database.
Request PGE Insertion or Update	One per insert/update request	Sybase ASE (COTS)	*Process:* EcDpAtRegisterPGE *Library:* PlCore2 *Class:* PlResourceRequirement	The EcDpAtRegisterPGE process sends insert or update requests to the **Sybase ASE** to add or modify PGE information in the PDPS database to perform data processing tasks.
Request ODL File	One per ODL file request	PGE/ESDT ODL Files	*Processes:* EcDpAtCheckODL, EcDpAtRegisterPGE *Library:* DpAtMetadata *Classes:* DpAtDatabase, DpAtScienceMd	In response to a request for an **ODL File**, the EcDpAtCheckODL and EcDpAtRegisterPGE processes receive data in “parameter = value” format about a **PGE**, its inputs and outputs, and scheduling information.

Table 4.7-9. AITTL Process Interface Events (7 of 7)

Event	Event Frequency	Interface	Initiated By	Event Description
Initiate Execution	One per tool initialization	*Processes:* EcDpAtInsertExeTarFile, EcDpAtInsertStaticFile, EcDpAtInsertTestFile, EcDpAtStageDAP, EcDpAtGetMCF, EcDpAtMgrLogDump, EcDpAtRegisterPGE, EcDpAtCheckODL, EcDpAtCreateODLTemplate, EcDpAtBadFunc, EcDpAtPrologs, EcDpAtCheckHdfFile, EcDpAtCheckPCF, EcDpAtBinDiffGui, EcDpAtBadFuncGui, EcDpAtOpDbGui, EcDpAtSSAPGui	*Process:* EcDpAtMgr *Library:* DpAtDsrv *Class:* DpAtMgrLogGuiMainWindow	The EcDpAtMgr initiates the tools (**EcDpAtInsertExeTarFile, EcDpAtInsertStaticFile, EcDpAtInsertTestFile, EcDpAtStageDAP, EcDpAtGetMCF, EcDpAtMgrLogDump, EcDpAtRegisterPGE, EcDpAtCheckODL, EcDpAtCreateODLTemplate, EcDpAtBadFunc, EcDpAtProlog**s, **EcDpAtCheckHdfFile, and EcDpAtCheckPCF**) and the GUI interfaces (**EcDpAtBinDiffGui, EcDpAtBadFuncGui, EcDpAtOpDbGui, and EcDpAtSSAPGui**) from a menu using UNIX system calls.
Request SSAP Information	One per SSAP information request	*Process:* EcDsScienceDataServer *Library:* DsCl *Classes:* DsClRequest, DsClCommand, DsClESDTReferenceCollector	*Process:* EcDpAtSSAPGui *Libraries:* DpAtSSAP, DpAtDsrv *Classes:* DpAtSSAPManager, DpAtDsrv	The EcDpAtSSAPGui sends requests to the **EcDsScienceDataServer** for information about SSAPs, including names of existing SSAPs and the components associated with a specific SSAP.
Insert SSAP Information	One per SSAP	*Process:* EcDsScienceDataServer *Library:* DsCl *Classes :* DsClRequest, DsClCommand	*Process:* EcDpAtSSAPGui *Library:* DpAtDsrv *Class:* DpAtSSAPManager, DpAtDsrv	The EcDpAtSSAPGui sends requests to the **EcDsScienceDataServer** to insert new SSAP information or update any existing SSAP information.

4.7.2.6 Algorithm and Test Tools Data Stores

Table 4.7-10 provides descriptions of the data stores shown in the AITTL CSCI architecture diagram.

Table 4.7-10. AITTL Data Stores

Data Store	Type	Functionality
PGE/ESDT ODL files	Files	These files are written in *parameter = value* formats to define the inputs and outputs of a PGE and any relevant scheduling information (including Production Rules), and are created by the Instrument Teams and the SSIT personnel.
PDPS database	Database	The PDPS database is replicated at each site for fault handling and recording purposes. The PDPS database holds all the persistent data including: • Resource information entered with the Resource Planning utilities • PGE and data type information entered at SSIT • Production Request, Data Processing Request and Data Granule information entered using the Production Request Editor • Plan information entered using the Production Planning Workbench

4.7.3 Data Processing Hardware Components

4.7.3.1 Science Processor Hardware CI (SPRHW) Description

Science Processor hardware (SPRHW) consists of the Science Processor Hardware and the Queuing Server Hardware.

Document 920-TDx-001 provides descriptions of the Science Processing HWCI and document 920-TDX-002 provides site-specific hardware/software mapping.

4.7.3.2 Algorithm Quality Assurance Hardware CI Description

Algorithm Quality Assurance Hardware (AQAHW) is used to validate the quality of EMD products includes non-science QA, in-line QA, and SCF-based QA. Non-science QA is specified by the DAAC Operations staff and includes data integrity checks on the data products and the metadata. In-line QA is a form of science QA validating product content using science algorithms. The EMD provides support for SCF-based QA by providing archive and communications capacity for the SCFs to sample and validate the contents of the products.

4.7.3.3 Descriptions of the Algorithm Quality Assurance HWCI. Algorithm Integration and Test Hardware CI Description

The Algorithm Integration and Test Hardware (AITHW) Configuration Item is the hardware to support the system level software validation, integration, and test and the integration and test of science software at a DAAC.

Document 920-TDx-001 provides descriptions of the Algorithm Integration and Test HWCI and document 920-TDX-002 provides site-specific hardware/software mapping.