

4.7.8 Order Manager Command Line Utility

Deleted. Not applicable for Release 7.23.

This page intentionally left blank.

4.7.9 OMS Configuration Command Line Interface

The OMS Configuration Command Line Interface (hereafter referred to as the OMS Configuration CI) provides DAAC operators with the ability to configure specific parameters for the OMS Server and Database that are not configurable via the OMS GUI. The ability to configure these parameters and settings in this utility is restricted to limited-capability operators.

Like most other ECS utilities, the OMS Configuration CI interacts directly with the OMS Database. However unlike many ECS command-line interfaces, the OMS Configuration CI uses an interactive menu system in addition to the ability to pass in command-line options.

Because of the nature and scope of the parameters and settings configurable with this utility, it should be rarely used. Most of the parameters and settings that would commonly and frequently be configured for the OMS as a system are done so via the OMS GUI.

Notes on Operator Capability Levels

In accordance with new Operator GUI security standards, the OMS GUI implements two levels of permissions such that only Full Capability operators have the ability to configure parameters and perform certain actions, while Limited Capability operators are limited to basic functionality as outlined in the OMS GUI section. The intention for the OMS Configuration CI is that it should be limited to operators of this “full capability” level. Certain parameters and capabilities were purposefully omitted from the OMS GUI to further restrict operator interaction due to the sensitivity of these parameters.

The Synergy V OMS Configuration CI provides Full Capability operators with the ability to:

- Configure Order Tracking Details for OMS
- Set which Production Module to use for a list of ESDT Collections.

4.7.9.1 Starting the OMS Configuration CI

Note: Although this utility is not protected in the same way as the OMS GUI, it was designed to be limited to Full Capability operators. Because it is a UNIX/Linux utility, it will employ standard UNIX/Linux security by protecting the execute permissions.

Script name: EcOmConfig.pl

Script path: /usr/ecs/<MODE>/CUSTOM/utilities/

Installation location: Installed onto the same machine as the OMS Server

Usage: EcOmConfig.pl [-ot <file> -odl <file>] [-help]

No mode is required.

Table 4.7.9-1 lists the options for the OMS configuration CI.

Table 4.7.9-1. Option Summary

Option	Description
-ot <file>	The flat file containing edited order tracking configuration for update in the OMS database (see formatting instructions below).
-odl <file>	The flat file containing the list of ESDT collections to be added or deleted for processing by a certain Production Module, depending on the selection made by the operator.
-help	Gives a brief overview of the input options that can be used with this utility.

Note: All the above options can be used concurrently.

4.7.9.2 OMS Configuration CI Operational Overview

Start the CI as specified above, passing in the options and any required flat files. The CI menu, as shown in Figure 4.7.9-1, will interactively prompt the operator to take certain actions with those files (it will never automatically process files). Select the type of configuration desired from the menu by typing the corresponding number at the prompt.

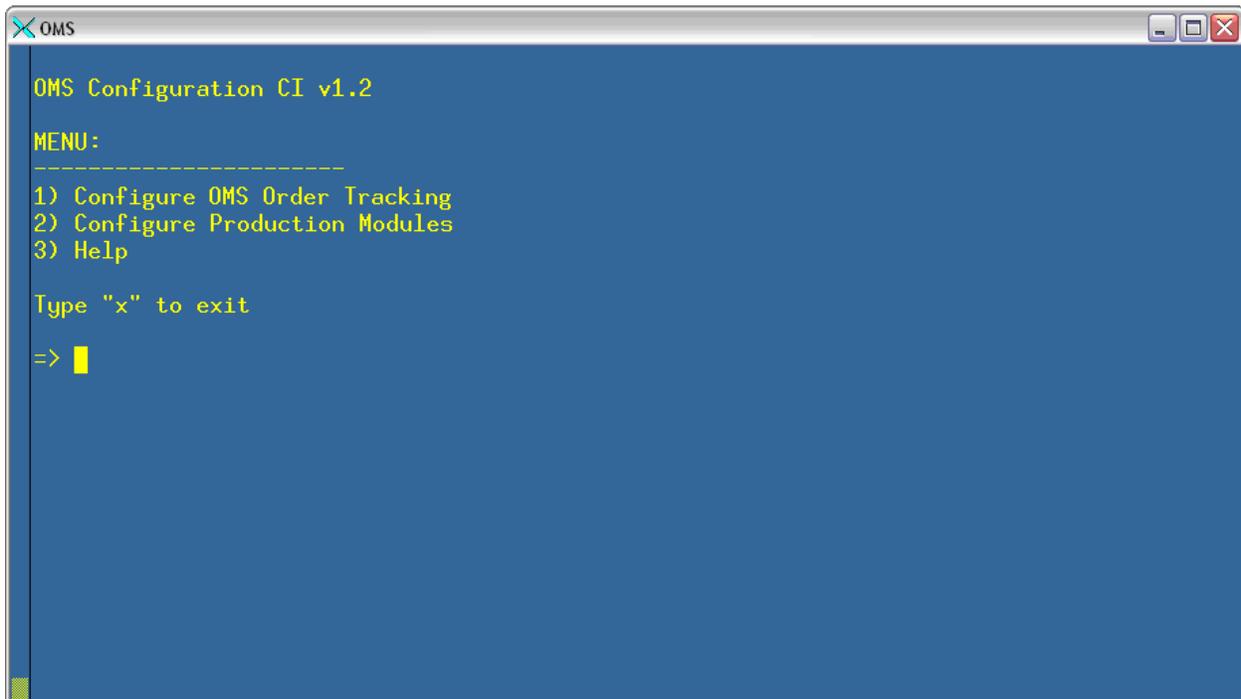
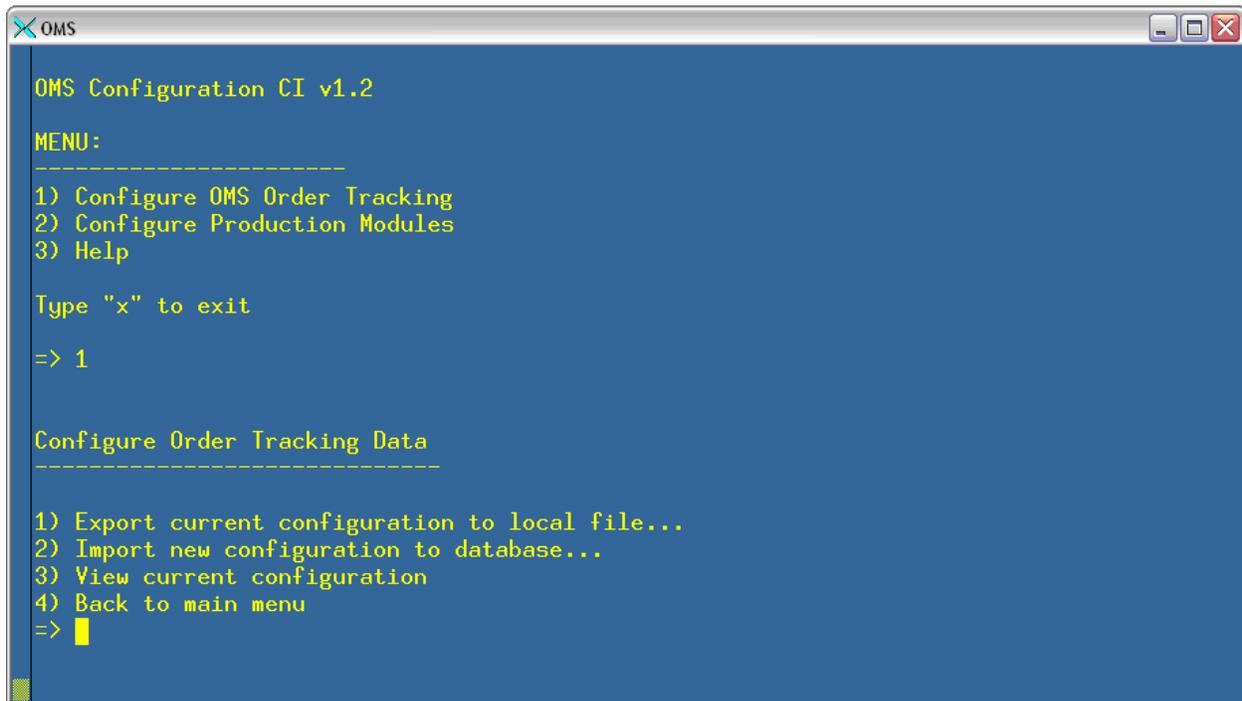


Figure 4.7.9-1. Main Menu

4.7.9.2.1 OMS Order Tracking Configuration

This feature allows the operator to configure how long order-tracking information is kept in the database. This can be configured by media type and order source.

When this item is selected from the main menu, the following submenu displayed in Figure 4.7.9-2 appears:



```
OMS Configuration CI v1.2
MENU:
-----
1) Configure OMS Order Tracking
2) Configure Production Modules
3) Help
Type "x" to exit
=> 1
Configure Order Tracking Data
-----
1) Export current configuration to local file...
2) Import new configuration to database...
3) View current configuration
4) Back to main menu
=> █
```

Figure 4.7.9-2. Configure Order Tracking Data Menu Screen (1 of 3)

The process here is to export the current configuration to a local file, edit that file, and import it back into the database.

To **export**, select item 1 (“Export current configuration to local file...”). The utility will create a unique file in the current directory as shown in Figure 4.7.9-3.

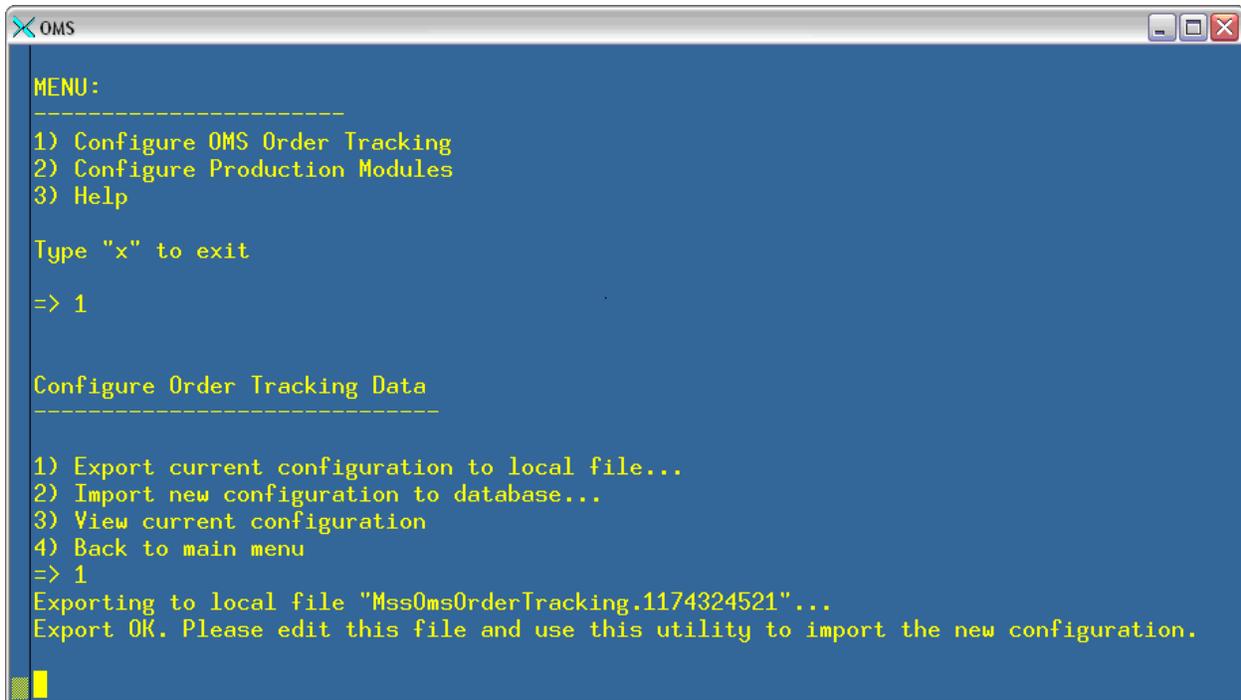


Figure 4.7.9-3. Configure Order Tracking Data Menu Screen (2 of 3)

The saved file contains the configuration for all media types and *all* order sources. The first item is the order source (D, S, E, or M), followed by the media type, followed by the retention time period in DAYS. See Table 4.7.9-2 below for the order source mappings.

Table 4.7.9-2. Order Source Mappings

Order Source Abbreviation	Order Source
D	Data Pool
S	Spatial Subscription Server
E	EWOC
M	DUE Machine-to-Machine Gateway

The file can be edited for any changes and then exported to the database. The file will be parsed out and the changes will be submitted to the database by “importing” the file as shown in Figure 4.7.9-4:

To **import** the file, run the utility again and pass in the edited file using the **-ot** option (see the beginning of this section). Select “Configure OMS Order Tracking” from the main menu and “Export new configuration to database...” from the subsequent menu.

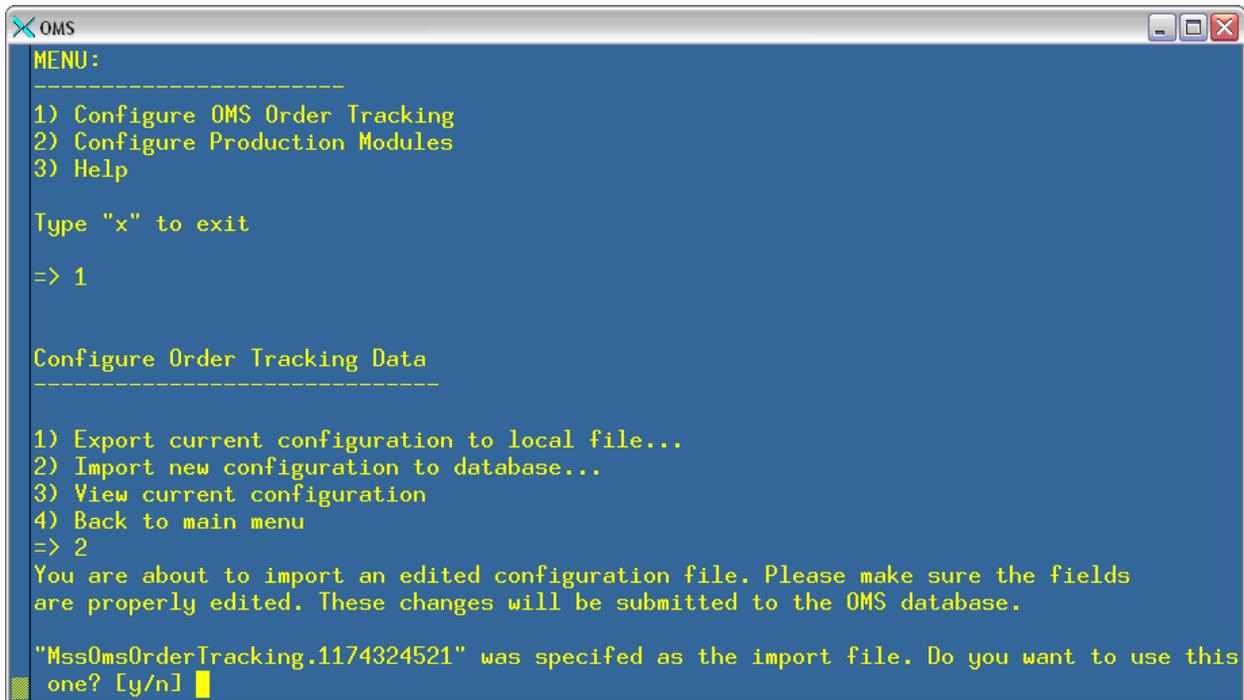


Figure 4.7.9-4. Configure Order Tracking Data Menu Screen (3 of 3)

The file will be checked for correct syntax and the changes will then be submitted.

Configure Production Modules

To configure Production Modules for use with certain ESDT Collections, start the Configuration CI with the `-odl <file>` parameter, where `<file>` is a text file of ESDT Collections, each on its own line. White space is ignored, so multiple lines can separate groups of collections. See example below:

```
MOD11_L2.001

MOD11_L2.002

GDAS_OZF.001

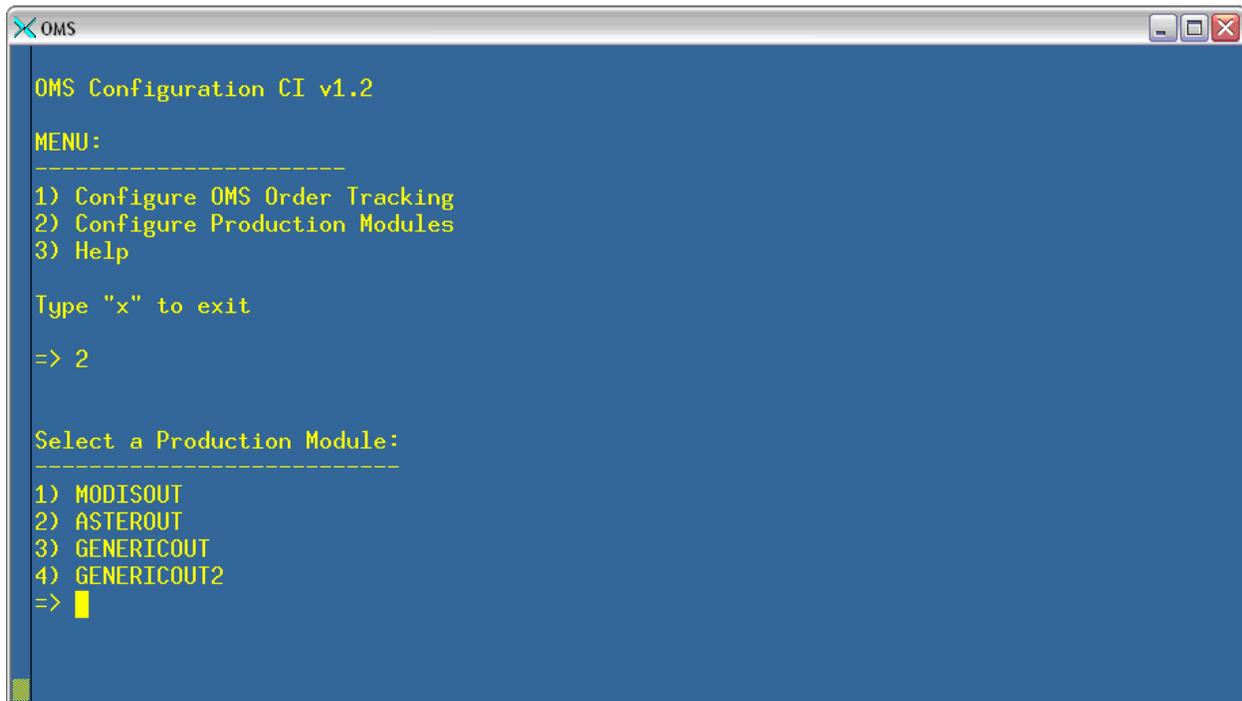
GDAS_OZF.002

.

.

.
```

Then, choose “Configure Production Modules” from the main menu, and the following submenu as shown in Figure 4.7.9-5 appears:



```
OMS Configuration CI v1.2
MENU:
-----
1) Configure OMS Order Tracking
2) Configure Production Modules
3) Help

Type "x" to exit

=> 2

Select a Production Module:
-----
1) MODISOUT
2) ASTEROUT
3) GENERICOUT
4) GENERICOUT2
=> █
```

Figure 4.7.9-5. Configure Production Module (1 of 4)

Select the Production Module from which you want to ADD or DELETE the list of ESDT Collections specified at script startup. The following submenu as shown in Figure 4.7.9-6 appears:

```
OMS
2) Configure Production Modules
3) Help

Type "x" to exit
=> 2

Select a Production Module:
-----
1) MODISOUT
2) ASTEROUT
3) GENERICOUT
4) GENERICOUT2
=> 1

Select an action:
-----
NOTE: File "odl.txt" will be used for the following actions:
1) ADD ESDTs in the file to "MODISOUT"
2) DELETE ESDTs in the file from "MODISOUT"
3) Cancel
=> █
```

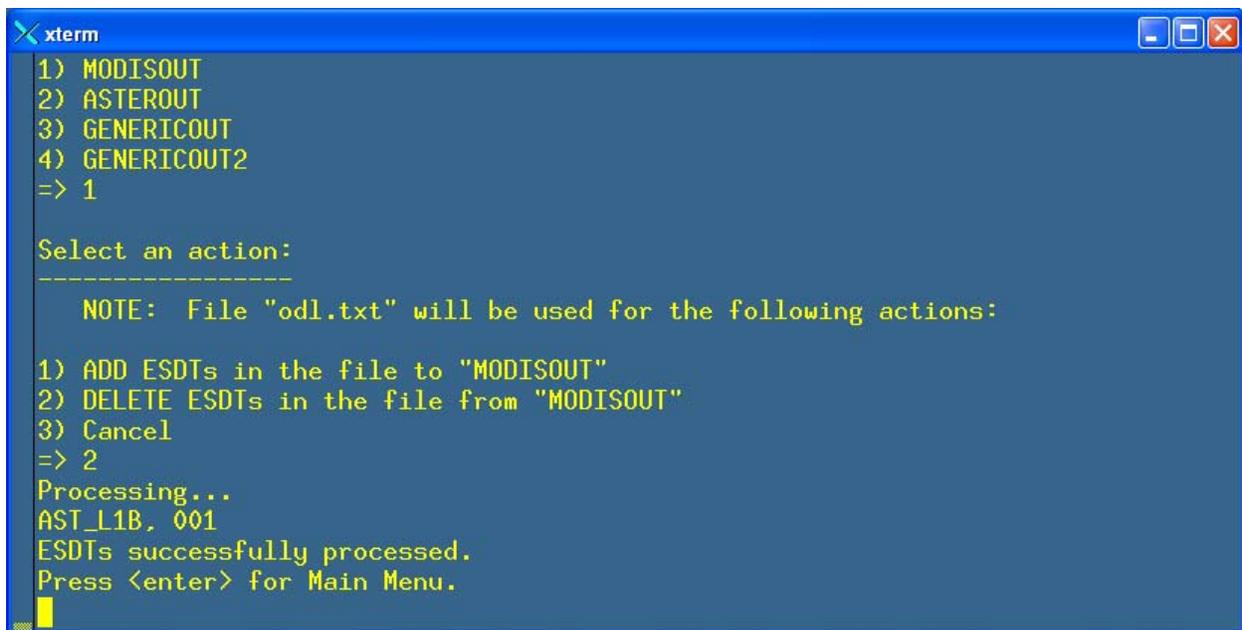
Figure 4.7.9-6. Configure Production Module (2 of 4)

Select to either ADD (see Figure 4.7.9-7) or DELETE (see Figure 4.7.9-8) the ESDT Collections found in the input file to the selected Production Module. One of the following screens will appear:

```
xterm
1) MODISOUT
2) ASTEROUT
3) GENERICOUT
4) GENERICOUT2
=> 1

Select an action:
-----
NOTE: File "odl.txt" will be used for the following actions:
1) ADD ESDTs in the file to "MODISOUT"
2) DELETE ESDTs in the file from "MODISOUT"
3) Cancel
=> 1
Processing...
AST_L1B. 001
ESDTs successfully processed.
Press <enter> for Main Menu.
█
```

Figure 4.7.9-7. Configure Production Module (3 of 4)



```
xterm
1) MODISOUT
2) ASTEROUT
3) GENERICOUT
4) GENERICOUT2
=> 1

Select an action:
-----
NOTE: File "odl.txt" will be used for the following actions:

1) ADD ESDTs in the file to "MODISOUT"
2) DELETE ESDTs in the file from "MODISOUT"
3) Cancel
=> 2
Processing...
AST_L1B.001
ESDTs successfully processed.
Press <enter> for Main Menu.
```

Figure 4.7.9-8. Configure Production Module (4 of 4)

OMS Configuration CI Help

Select “Help” from the main menu for a complete synopsis of the options and all available functions of the CI. The **-help** option also displays a brief overview of the input options. Figure 4.7.9-9 displays part of the help text. Below is part of the help text.

```
OMS
=> 3

OMS Configuration CI 1.2 HELP
-----

Type "q" at any time to quit help.

Usage:
EcOmConfig.pl [options]
-ot <file>      Order tracking export file
-odl <file>     ODL file list for configuration
                of production modules
-help          print out a summary of the command-line
                options

NOTE: MODE SWITCHING BETWEEN S3 AND S4 IS NO LONGER SUPPORTED AND
      ALL OPTIONS RELATED TO THIS HAVE BEEN REMOVED.

CONFIGURE MSS/OMS ORDER TRACKING
This allows an authorized operator to edit the retention time
period of order tracking information by source and by media type.
More--
```

Figure 4.7.9-9. Help Screen

4.7.9.3 Required Operating Environment

The following environment is required for the OMS Configuration CI to work properly.

The O/S requirements are Linux 2.x.

4.7.9.4 Interfaces and Data types

The OMS Configuration CI exchanges data between the Application (interaction w/ operator) and Sybase, using Perl DBI Modules.

4.7.9.5 Databases

The OMS Configuration CI accesses the OMS database.

4.7.9.6 Special Constraints

There are no special constraints to running the OMS Configuration CI.

4.7.9.7 Outputs

The Configuration CI is an interactive system, so messages and prompts are displayed to the operator on the screen. Error messages are displayed to the screen as well as printed to the log.

4.7.9.8 Events and Messages

The Configuration CI writes status and error messages to the EcOmConfig.log file in the directory /usr/ecs/<MODE>/CUSTOM/logs.

4.7.9.9 Reports

The Configuration CI does not generate reports.

4.7.10 OmPdCleanup GUI

Deleted. Not applicable for Release 7.23.

This page intentionally left blank.

4.7.11 Science Command Line Interface (OmSCLI) in OMS

The Science Command Line Interface (OmSCLI) allows the operator to acquire ECS products by sending orders directly to the Order Manager Server using an operator-provided file of granule identifiers and a parameter file of distribution options. The operator can request distribution of the ordered products by FtpPush, FtpPull, or secure copy (scp). The OmSCLI will not generate Metadata Control Files (MCFs) since that functionality is now performed by the ESDT Maintenance GUI.

The OmSCLI is installed on the same host as the Order Manager Server. It includes an acquire wrapper script, EcCoEnvPerl, and a C++ -based executable which interfaces with the OrderManager client. It has its own configuration file containing database environment parameters. It is invoked with arguments that are described in the following section.

4.7.11.1 Quick Start Using the OmSCLI

To invoke the OmSCLI, enter the following on the command line:

acquire <mode> -p <parameterfile> -f <file> -t <tag> [-decrypt]

Table 4.7.11-1 describes the OmSCLI command line parameters.

Table 4.7.11-1. Command Line Parameters of the SCLI Tool

Parameter Name	Description
<mode>	Required. The mode in which the tool runs (i.e. OPS, TS1).
-p <parameterfile>	Required. <parameterfile> is the full path name of a file containing all parameters needed to control distribution of the acquired granules. The parameters are listed one per line in the parameter file, in PARAMETERNAME = VALUE format.
-f <file>	Required. <file> is the full path name of a file that contains identifiers for up to MAXURSPERACQUIRE granules to be acquired. Granule identifiers are listed one per line in the file, and may be in UR, GeoID format, or LocalGranuleID format. The file may contain a mixture of URs, GeoIDs and LocalGranuleIDs.
-t <tag>	Required. <tag> is a unique alphanumeric request identification, used to track the distribution request internally.
-decrypt	Optional; Used only for FTTPush distribution requests. If present, indicates that the FTTPush and SCP password passed in is encrypted and needs to be decrypted by the OmSCLI. This option does not work. It will be fixed in NCR 8048998

4.7.11.2 OmSCLI Command Line Parameter Formats

-p <parameterfile>

The parameter file must contain the following distribution parameters and their values. Parameters are listed one per line in the parameter file, in PARAMETERNAME = VALUE format, and may be listed in any order.

DDISTMEDIATYPE = <FtpPush>|<FtpPull>|<scp> (required)

DDISTMEDIAFMT = FILEFORMAT (required)

FTPUSER = <FTP user id> (required for FtpPush, SCP distributions)

FTPPASSWORD = <FTP Password> (required for FtpPush, SCP distributions)

FTPHOST = <FTP host name> (required for FtpPush, SCP distributions)

FTPPUSHDEST = <full path name of push directory on FTPHOST>
(required for FtpPush, SCP distributions)

ECSUSERPROFILE = <user id associated with the request> (required)

PRIORITY = HIGH | VHIGH | NORMAL | LOW | XPRESS. (required)

DDISTNOTIFYTYPE = MAIL (required)

NOTIFY=<email address> (required)

USERSTRING=<text which describes the request; this string will be displayed on the Operator Intervention Detail page on the OMS GUI for FtpPush and FtpPull orders>.

Example of parameter file:

1. FTP Pull:

```
ECSUSERPROFILE = ECSGuest  
PRIORITY = NORMAL  
DDISTMEDIATYPE = FtpPull  
DDISTMEDIAFMT = FILEFORMAT  
USERSTRING = Test_For_Pull  
DDISTNOTIFYTYPE = MAIL  
NOTIFY = testforpull@eos.hitc.com
```

2. Ftp Push

```
ECSUSERPROFILE = labuser  
FTPUSER = labuser  
FTPPASSWORD = mypasswd  
FTPHOST = f4eil01  
FTPPUSHDEST = /home/labuser/PushArea/  
PRIORITY = HIGH  
DDISTMEDIATYPE = FtpPush  
DDISTMEDIAFMT = FILEFORMAT  
USERSTRING = Test_For_Push  
DDISTNOTIFYTYPE = MAIL  
NOTIFY = testforpush@eos.hitc.com
```

3. SCP

```
PRIORITY=VHIGH  
DDISTMEDIATYPE=scp
```

DDISTNOTIFYTYPE=MAIL
DDISTMEDIAFMT=FILEFORMAT
ECSUSERPROFILE=labuser
FTPUSER=labuser
FTPHOST=f4spl01
USERSTRING=Test_For_Scp
FTPPUSHDEST=/home/labuser/ScpPushDir
FTPPASSWORD=testpasswd
NOTIFY=testforscp@raytheon.com

-t <tag>

The user populates the OmSCLI tag parameter with a unique alphanumeric request identification.

-f <file>

The <file> contains a list of up to MAXURSPERACQUIRE granule identifiers, one for each granule to be acquired.

Granule identifiers are listed one per line in the file, in UR, GeoID or LocalGranuleId format. The file may contain a mixture of URs, GeoIDs and LocalGranuleIds.

Example granule identifiers:

UR:10:DsShESDTUR:UR:15:DsShSciServerUR:10:[:DSSDSRV]:18:SC:MOD14.086:62196
MOD14.A2006159.0030.086.2006159125821.hdf
SC:MOD14.086:62197

For each LocalGranuleId listed in the file, the OmSCLI will invoke a search for that LocalGranuleId in the AIM inventory database (via an EcOmDb stored procedure) and will convert the LocalGranuleId to GeoID format. If more than one granule is found in the AIM inventory database for a given LocalGranuleId, all granules found will be included in the request.

For each UR listed in the file, OmSCLI will extract the GeoId from that UR.

4.7.11.3 SCLI Command Line Utility Configuration File

The Command Line Utility has an associated configuration file with values stored in a basic PARAMETER = VALUE format. The configuration file is called EcOmSCLI.CFG, and is stored in the /usr/ecs/<mode>/CUSTOM/cfg directory for the mode. Table 4.7.11-2 describes its contents:

Table 4.7.11-2. OmSCLI Configuration File Parameters

Parameter Name	Value Description
Site	DAAC Name
SubSystem	OMS
Name	EcOmSCLI
ApplicationID	1300000
ProgramID	1300008
AppLogSize	The maximum ALOG size
AppLogLevel	ALOG level
DebugLevel	Debug log level
MajorVersion	1
MinorVersion	0
SCLISeniorTransactionID	Unique ID for the request to the OrderManager
SCLI_MODE	The mode in which the SCLI is run
MAXURSPERACQUIRE	Maximum allowed URs per order
SYBASE_SERVER	Name of Sybase SQL Server
SYBASE	Location of the Sybase Open Client
SYBINTERFACES	Location of Sybase open client library interface file
DSSSrUNIXEnv	SYBASE DSQUERY
DBMAXRESULTS	Maximum database return rows
DBNAME	OMS database name
DBPASSWDSEED	1300008 (the seed used to get Command Line utility database login password)
DBUSERNAME	EcOmSCLI (the database login name of Command Line utility)
MAX_DB_CONNECTIONS	The maximum database connections Command Line utility uses to connect to the OMS Database
DSQUERY	Name of Sybase SQL Server
DB_MAX_JOINS	Maximum number of database join operations
DSSSrEnv_DB	DBUSERNAME DBPASSWDSEED DBNAME DBMAXRESULTS SYBINTERFACES SYBASE_SERVER DB_MAX_JOINS MAXURSPERACQUIRE SCLI_MODE
DSSSrEnv	\$(DSSSrEnv_DB) \$(DSSSrUNIXEnv)
num_retries	Number of retries when DB can't connect to the DB
sleep_sec	Number of seconds between retries

4.7.11.4 Required Operating Environment

This command line utility runs on the Linux 2.x platforms.

4.7.11.5 Interfaces and Data

Table 4.7.11-3 lists the supporting products this tool depends upon to function properly.

Table 4.7.11-3. Interface Protocols

Product Dependency	Protocols Used	Comments
OMS Database	SQL	Via SQL server machine
Sybase Open Client library	EcDbInterface	
Perl	Perl scripts	

4.7.11.6 Databases

Table 4.7.11-4 lists the databases stored procedures and tables used by the command line utility.

Table 4.7.11-4. SCLI Interaction with OrderManager Database

Database	Stored Procedure	Tables
EcOmDB_<MODE>	OmGetGranulesByLGID	EcInDb_<MODE>..DsMdGranules EcInDb_<MODE>..DsMdCollections

Because the OmSCLI passes its database environment to the Order Manager client, the user EcOmSCLI must be registered in the EcOmDb_<MODE>, mss_acct_db_<MODE>, EcInDb_<MODE>, and DataPool_<MODE> databases. In addition, the name EcOmSCLI must be added as a login.

4.7.11.7 Special Constraints

The acquire wrapper script and EcCoEnvPerl must be located in the /usr/ecs/<mode>/CUSTOM/utilities directory of the mode.

The OmSCLI and Perl scripts must be installed on the same host as the OrderManagerServer.

4.7.11.8 Outputs

The OmSCLI writes processing status messages to a single application log (SCLI.log) in the logs directory of the mode with the tag identifier and final submittal status

If OmSCLI fails, the OmSCLI writes to temporary log files with error information.

Example of sample output from OmSCLI:

```
f4oml01{cmshared}(yhuang_7047814_relb)58: acquire DEV04 -p parameterfile.acquire.push -f mixData -t yl1238
Successfully open file parameterfile.acquire.push
Successfully open file mixData
INFO: LocalGranuleId
MODPTQKM.A2005310.h00v09.004.2005312161035.hdf.0220.1193165254.72679.RGEN.hdf
11/09/09 10:31:58: Thread ID : 32374 : Creating database connection manager OMSPool
11/09/09 10:31:58: Thread ID : 32374 : Initializing database connection pool
11/09/09 10:31:58: Thread ID : 32374 : Connection pool initialized successfully
11/09/09 10:31:58: Thread ID : 32374 : OmSrDbInterface::OmSrDbInterface
Connecting to OMS database with the following parameters :
Server name : f4dbl03_srvr
Database name : EcOmDb_DEV04
```


11/09/09 10:31:58.224531 32374: insert #GRANS values (90426, 'MODPTQKM.086', NULL, 'SC')
11/09/09 10:31:58: Thread ID : 32374 : EcDbInterface::Execute SQL=insert #GRANS values (90426, 'MODPTQKM.086', NULL, 'SC')
11/09/09 10:31:58.225442 32374: exec OmPreValidate "0800031139"
11/09/09 10:31:58: Thread ID : 32374 : EcDbInterface::Execute SQL=exec OmPreValidate "0800031139"
11/09/09 10:31:58.275541 32374: The granules are all valid
11/09/09 10:31:58.275632 32374: ENTRY: OmSrDbInterface::BeginTransaction
11/09/09 10:31:58: Thread ID : 32374 : EcDbInterface::Execute SQL=begin transaction
11/09/09 10:31:58.275818 32374: exec OmInsBulkGranules_nondp '0800031139'
11/09/09 10:31:58: Thread ID : 32374 : EcDbInterface::Execute SQL=exec OmInsBulkGranules_nondp '0800031139'
11/09/09 10:31:58.297726 32374: exec OmCompleteReqSubmission '0800031139'
11/09/09 10:31:58: Thread ID : 32374 : EcDbInterface::Execute SQL=exec OmCompleteReqSubmission '0800031139'
11/09/09 10:31:58.300033 32374: ENTRY: OmSrDbInterface::EndTransaction
11/09/09 10:31:58: Thread ID : 32374 : EcDbInterface::Execute SQL=commit transaction
Order Submitted to OMS Successfully.
11/09/09 10:31:58.303448 32374: 0xf94fa50OmSrRequest(0800031139)::dtor: 0800025127
OrderId is: 0800025127
RequestId is: 0800031139

2009/11/09 10:31:58.311: 26758: Successfully run acquire with exit 0

4.7.11.9 Event and Error Messages

The SCLI.log contains the final success or failure status of submitting the request to OMS. During processing, information messages and error messages (IO error messages, database connection, and database processing messages) are written to the screen and the temporary logs.

4.7.11.10 Reports

None

This page intentionally left blank.

4.7.12 Overview of the ESDT Maintenance GUI

The ESDT Maintenance GUI is a Web-based interface that allows operators to manage earth science data types (ESDTs) in the Archive Information Management (AIM) system. Using this GUI, an operator can add new ESDTs into the system, as well as view, update, and delete existing ESDTs. The operator can also generate metadata configuration files (MCFs) and ESDT-specific schemas.

Since the ESDT Maintenance GUI is a Web-based interface, it can be accessed from virtually anywhere on the internal network. No custom software installation is required – all that is needed is a Web browser (see **Section 4.6.1.28** Browser Requirements).

This document shows and explains in detail all of the available features and functionality of the ESDT Maintenance GUI, from the first login to complex operator actions.

4.7.12.1 Login Page

This page first appears when the application is loaded. The operator will be required to enter a pre-assigned password, as shown in Figure 4.7.12-1. Once the operator is logged in, the Install ESDT Page will be displayed and the application will be enabled.

Access to the ESDT Maintenance GUI is restricted to a single database username. This username is configured in the ESDT Maintenance GUI configuration file. The operator will log in by providing the password for this user.

The ESDT Maintenance GUI will only allow for one authenticated session at a time. This is to prevent situations where multiple operators may perform conflicting actions. The session time-out value is configured in the web application settings and is configurable via ECS Assist.

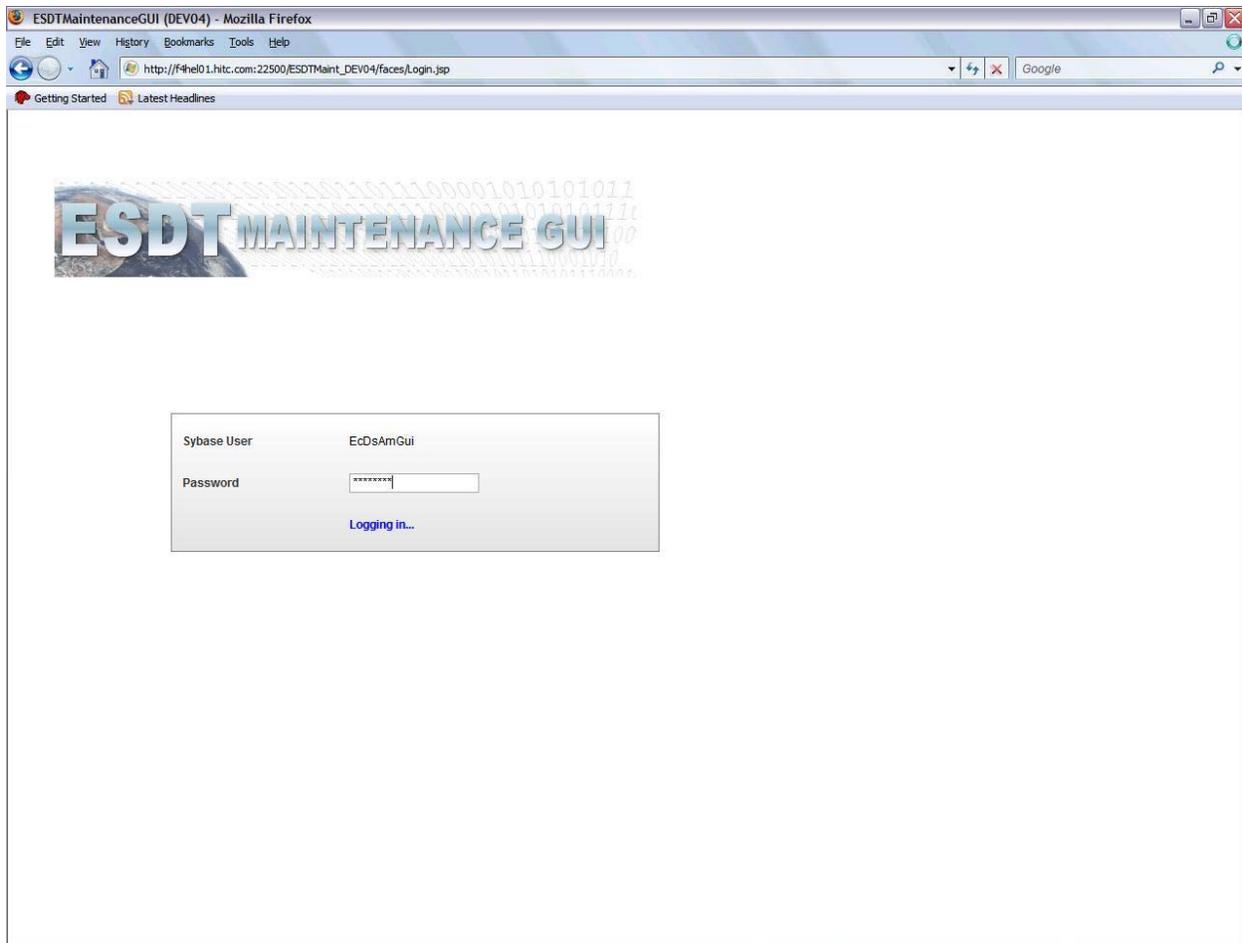


Figure 4.7.12-1. Login Page

4.7.12.2 List ESDT Page

This page first appears when the operator is logged in. The page lists the currently installed ESDTs, as shown in Figure 4.7.12-2.

From this page, the operator can perform the following actions:

- Search for an ESDT by using the browser's built-in search function
- Apply a filter to certain ESDTs
- View descriptor information for a specific ESDT
- Delete one or more ESDTs
- Generate MCFs for one or more ESDTs
- Generate ESDT-specific schemas for one or more ESDTs
- Navigate to the ESDT installation/update page

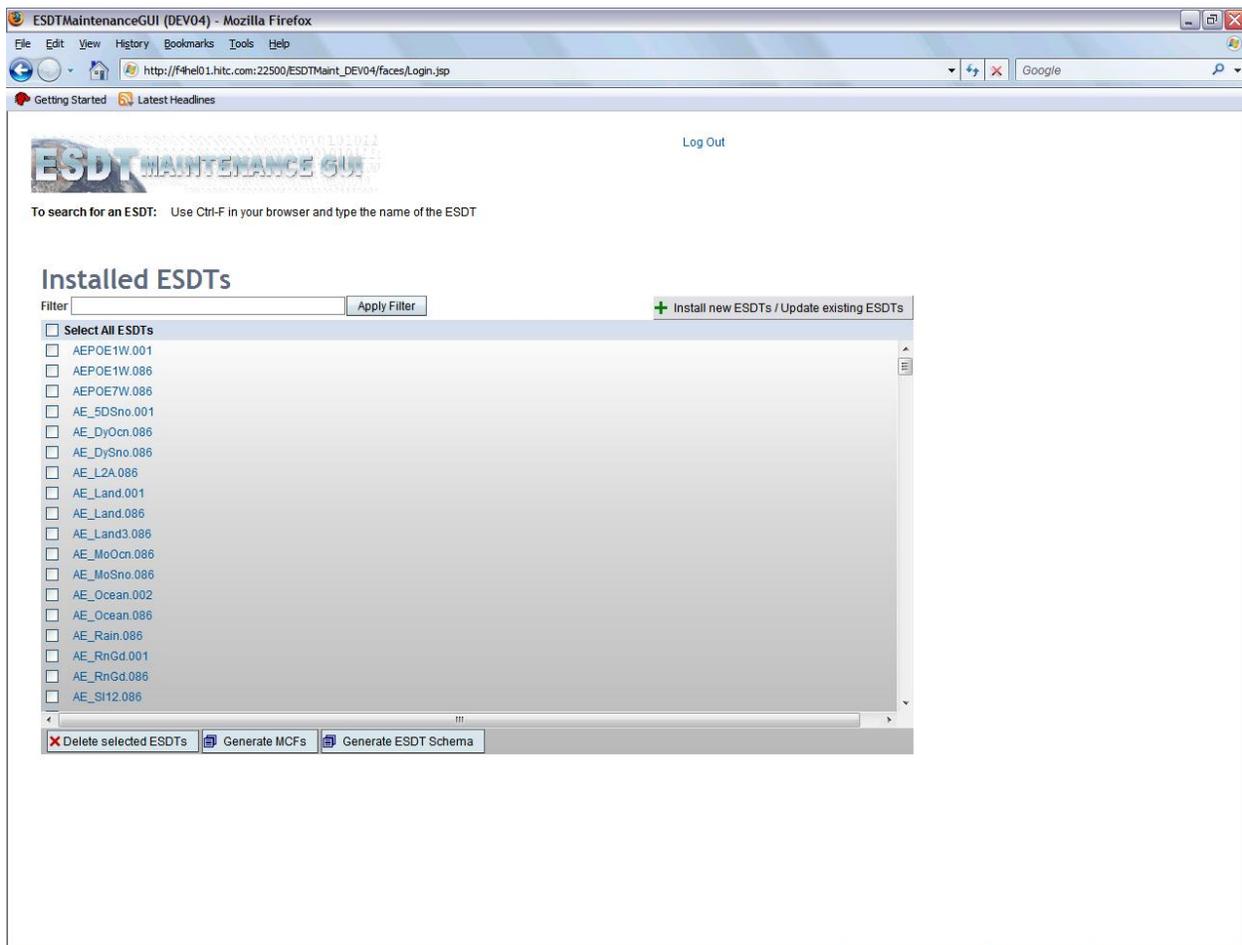


Figure 4.7.12-2. List ESDT Page

4.7.12.2.1 Search for an ESDT by using the browser's built-in search

To search for an ESDT, use Ctrl-F in your browser, type in the name of the ESDT, and click on "Find".

4.7.12.2.2 Filtering the Install ESDT Page

The List ESDT Page includes a filter that can be applied to the list of installed ESDTs. This is useful for selecting particular types of ESDTs for bulk action (i.e., deletion, MCF or schema generation). This is a simple text search and will search based upon the ESDT short name. As shown in Figure 4.7.12-3, "GLA" would return any ESDT with the "GLA" anywhere in the short name. The search is also case-sensitive.

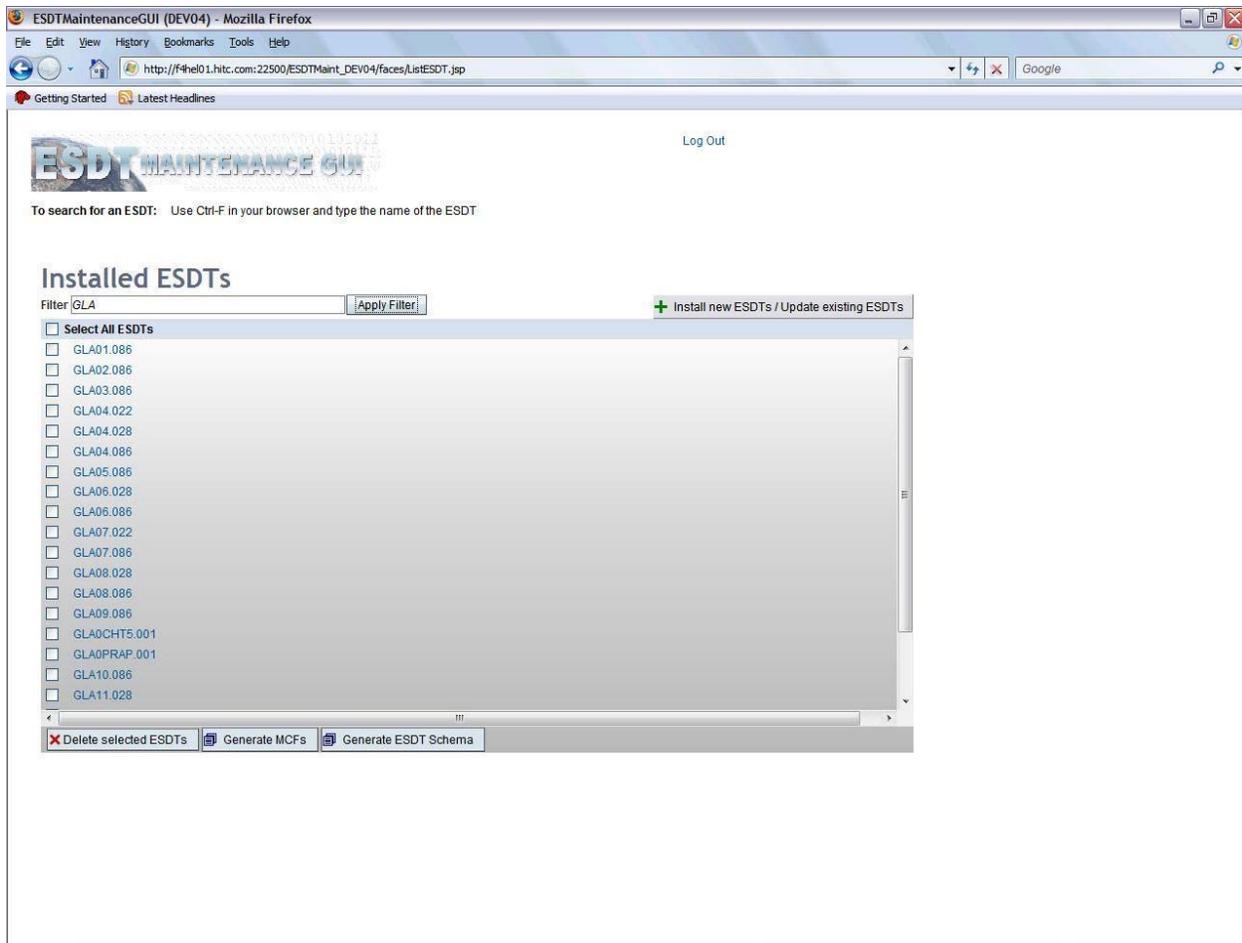


Figure 4.7.12-3. Filtering the List ESDT Page

4.7.12.2.3 View Descriptor Information for a Specific ESDT

From the List ESDT Page, click an ESDT link as shown in Figure 4.7.12-4 to view the ODL and XML descriptor information for a specific ESDT, and this will direct you to the ESDT Detail Page (see section 4.7.12.3 for more details).

4.7.12.2.4 Delete Selected ESDT Action

From the List ESDT Page, the operator can select a list of ESDTs and delete them by clicking on the “Delete selected ESDTs” button as shown in Figure 4.7.12-5.

Installed ESDTs

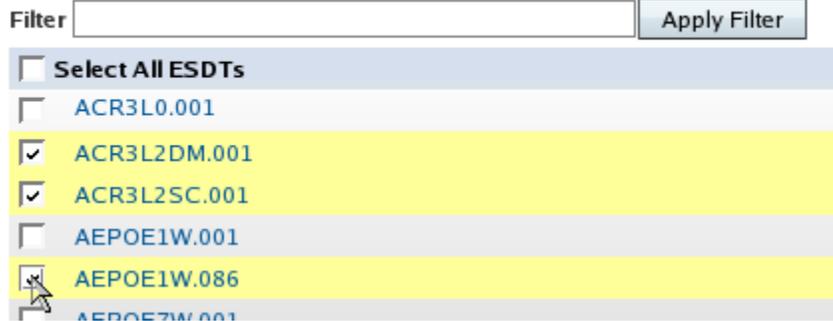


Figure 4.7.12-4. Selecting a list of ESDTs

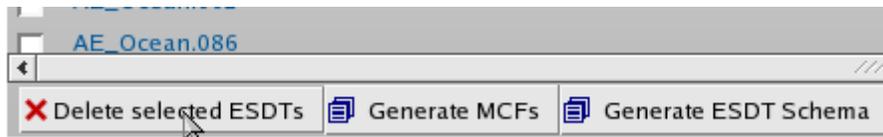


Figure 4.7.12-5. Delete selected ESDTs button

The operator will then be prompted for confirmation to perform the delete action on the selected ESDTs. Confirm the action to continue the delete process.

Once the delete action finishes, the result will be displayed at the top of the page as shown in Figure 4.7.12-6. Depending on the result, one to three possible result tables will be displayed as listed in Table 4.7.12-1.

Table 4.7.12-1. ESDT Delete Action Results

Table	Description
Failed ESDTs	The color of this table is red. This table displays any ESDTs that have failed processing previously and are in an intermediate state (installing, deleting, or updating). This table provides a “Clean Up” button allowing the operator to rollback/remove failed ESDTs. If the ESDTs are in the installing or deleting state, the cleanup action will remove the ESDTs from the database and remove all files (desc, schema, and mcf) from the physical directory. If the ESDTs are in the updating state, the cleanup action will rollback the ESDT to its previous installed state.
ESDTs with Errors	The color of this table is red. This table displays any ESDTs that have processing errors from the most recent action submitted.
Successful ESDTs	The color of this table is green. This table displays a list of ESDTs that have completed successfully from the most recent action submitted.

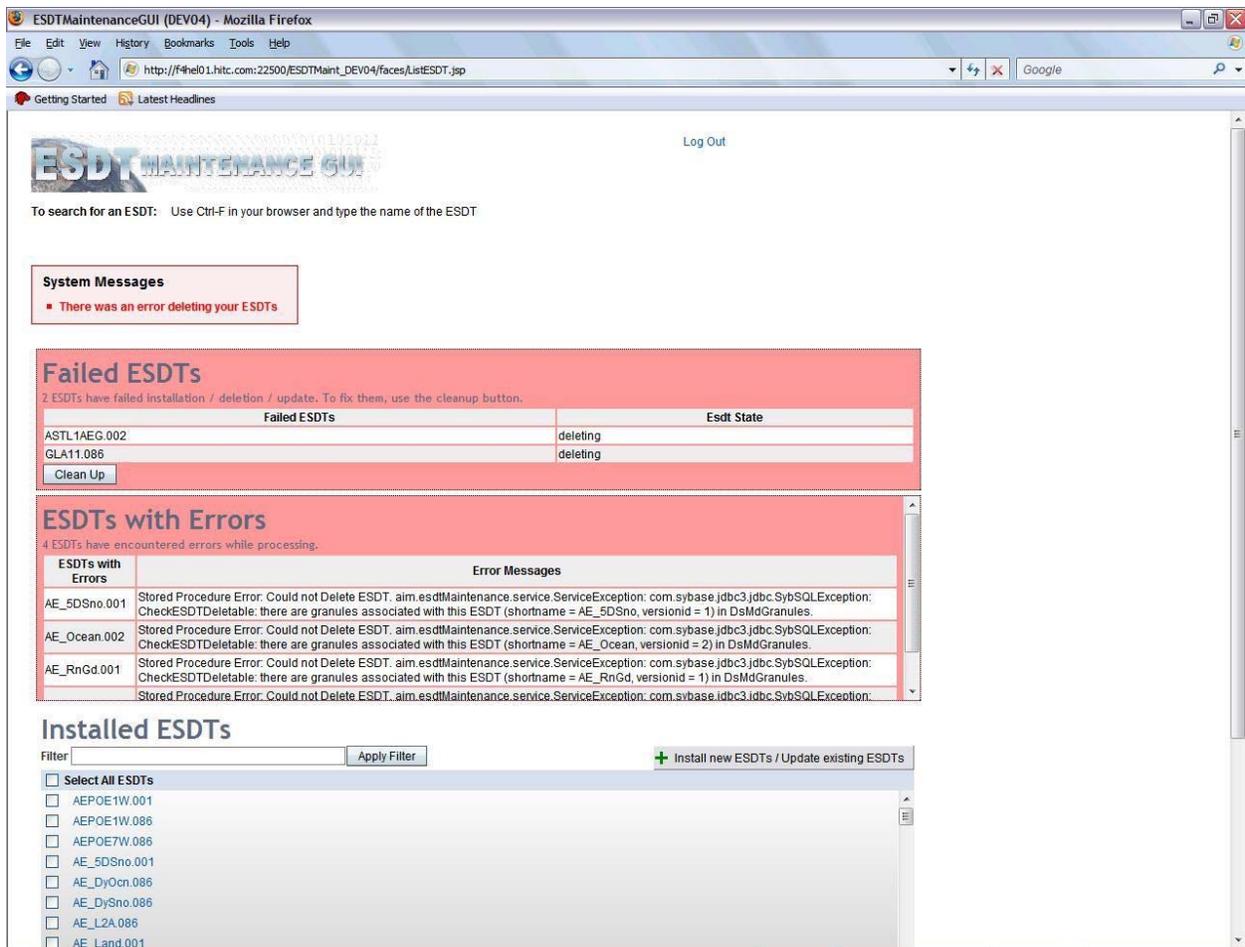


Figure 4.7.12-6. Delete ESDT Result Page

4.7.12.2.5 Generate MCFs for selected ESDTs

From the List ESDT Page, the operator can select a list of ESDTs and generate MCFs for them by clicking on the “Generate MCFs” button as shown in Figure 4.7.12-7.

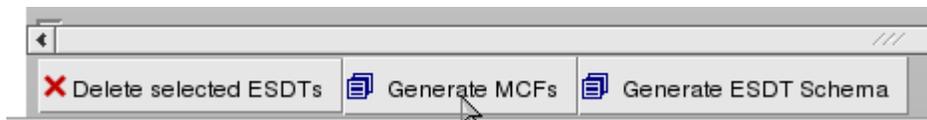


Figure 4.7.12-7. Generate MCF for Selected ESDT Button

The directory the MCFs will be generated to can be found in the EcDsAmESDTMaint.properties file. The result page for the generate MCF action is similar to the delete ESDT action as shown in Figure 4.7.12-8.

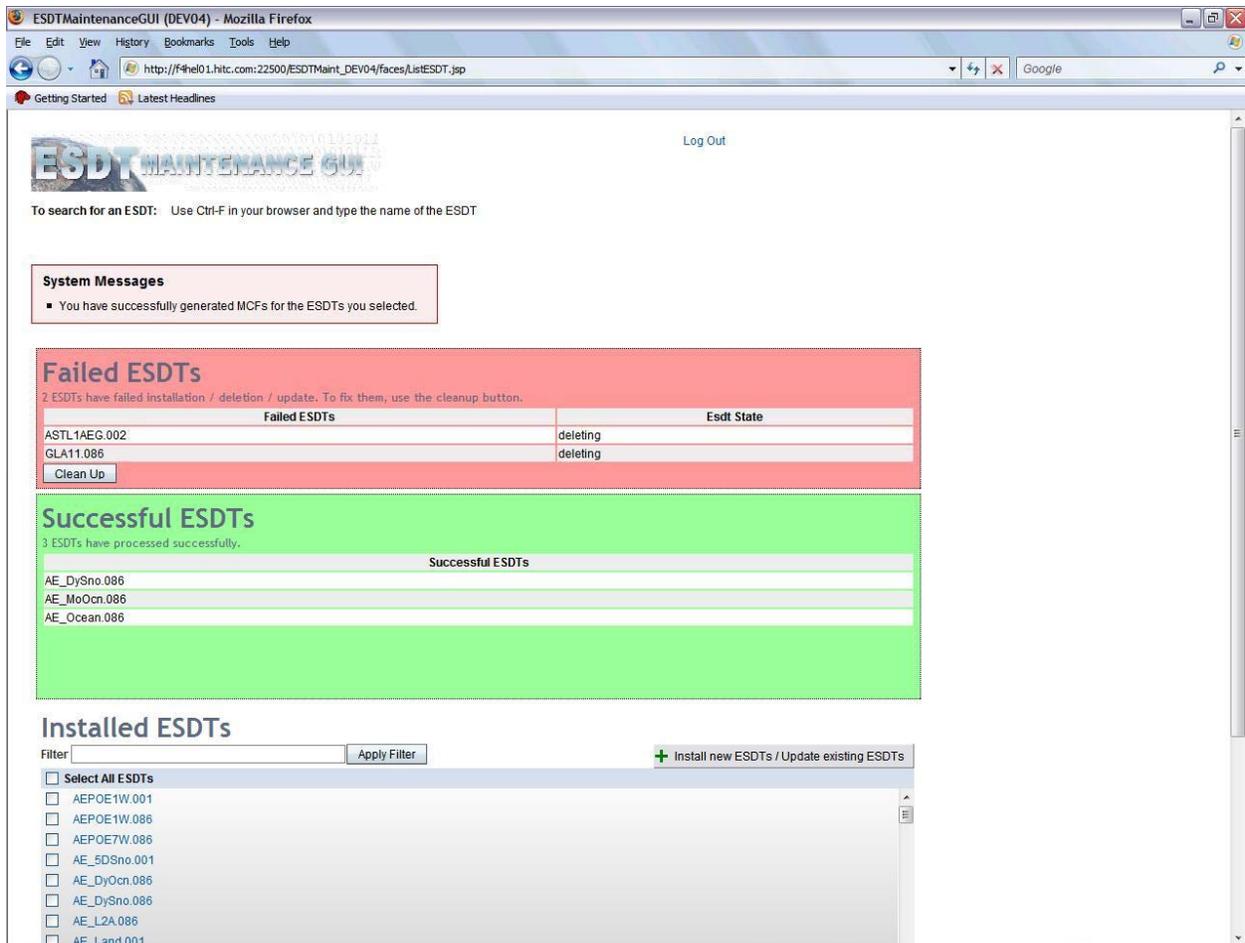


Figure 4.7.12-8. Generate MCF Result Page

4.7.12.2.6 Generate Schema for selected ESDTs

From the List ESDT Page, the operator can select a list of ESDTs and generate schema for them by clicking on the “Generate ESDT Schema” button as shown in Figure 4.7.12-9.

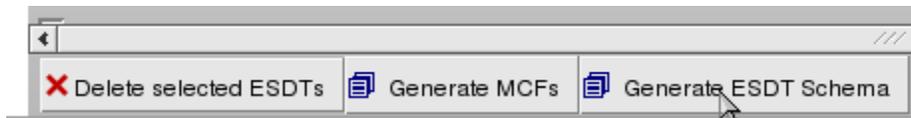


Figure 4.7.12-9. Generate Schema for Selected ESDT Button

The directory the schemas will be generated to can be found in the EcDsAmESDTMaint.properties file. The result page for the generate schema action is similar to the delete ESDT action as shown in Figure 4.7.12-10.

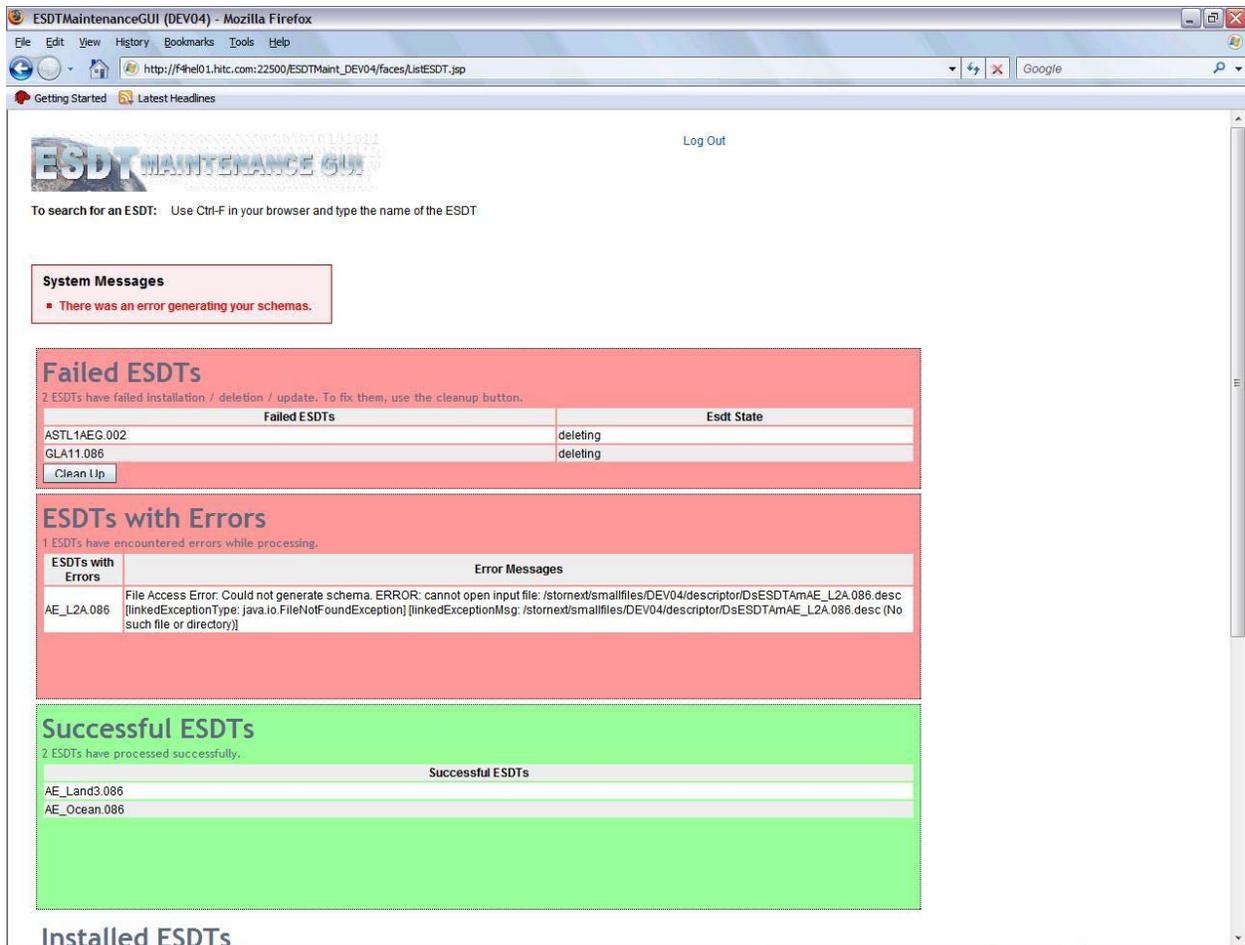


Figure 4.7.12-10. Generate Schema Result Page

4.7.12.3 ESDT Detail Page

From the List ESDT Page, the operator can click on the link for a particular ESDT to get more information about the installed ESDT. Once the link is clicked, the operator will be directed to the ESDT Detail Page as shown in Figures 4.7.12-11 and 4.7.12-12. This page allows the operator to view the ODL descriptor file or to view the XML representation of the descriptor file. To view the ODL file, click on the “ODL” tab and to view the XML file, click on the “XML” tab.

ESDT MAINTENANCE GUI Log Out

[<< back to ESDT List](#)

Descriptor Information

ESDT Name: AE_DySno.086

FileName: /stormext/sm/affiles/DEV04/descriptor/DsESDTAmAE_DySno.086.desc

XML ODL

```

<!--*****-->
<!--*          2000/01/28          ECSed25419          */-->
<!--* Initial PM-1 and ADEOS-II AMSR ESDT delivery          */-->
<!--*****-->
<METADATA>
<COLLECTIONMETADATA GROUPTYPE="MASTERGROUP">
  <DLName DATA_LOCATION="MCF" MANDATORY="TRUE" NUM_VAL="1">libDesDTsYBASIC.0019
  <GranuleTimeDuration DATA_LOCATION="MCF" MANDATORY="FALSE" NUM_VAL="1">90000</
  <SpatialSearchType DATA_LOCATION="MCF" MANDATORY="TRUE" NUM_VAL="1">Rectangle</
  <CollectionDescriptionClass>
    <ShortName DATA_LOCATION="MCF" MANDATORY="TRUE" NUM_VAL="1">AE_DySno</ShortN
    <LongName DATA_LOCATION="MCF" MANDATORY="TRUE" NUM_VAL="1">AMSR-E Aqua Daily
    <CollectionDescription DATA_LOCATION="MCF" MANDATORY="TRUE" NUM_VAL="1">AMSF
    <VersionID DATA_LOCATION="MCF" MANDATORY="TRUE" NUM_VAL="1">86</VersionID>
  </CollectionDescriptionClass>
  <ECSCollection>
    <RevisionDate DATA_LOCATION="MCF" MANDATORY="TRUE" NUM_VAL="1">2006-04-19</
    <ProcessingCenter DATA_LOCATION="MCF" MANDATORY="TRUE" NUM_VAL="1">AMSR-E SI
    <ArchiveCenter DATA_LOCATION="MCF" MANDATORY="TRUE" NUM_VAL="1">NSIDC</Arch
    <VersionDescription DATA_LOCATION="MCF" MANDATORY="TRUE" NUM_VAL="1">Createc
    <DatasetDisclaimerPointer DATA_LOCATION="MCF" MANDATORY="FALSE" NUM_VAL="1"
    <ECSCollectionGuide>
      <ECSCollectionGuidePointer DATA_LOCATION="MCF" MANDATORY="FALSE" NUM_VAL=
      <ECSCollectionGuidePointerComment DATA_LOCATION="MCF" MANDATORY="FALSE"
    </ECSCollectionGuide>
    <MiscellaneousInformation>
      <MiscellaneousInformationPointer DATA_LOCATION="MCF" MANDATORY="FALSE" NU
      <MiscellaneousInformationPointerComment DATA_LOCATION="MCF" MANDATORY="F
    </MiscellaneousInformation>
  </ECSCollection>
  <SingleTypeCollection>
    <CollectionState DATA_LOCATION="MCF" MANDATORY="TRUE" NUM_VAL="1">In Work</
    <MaintenanceandUpdateFrequency DATA_LOCATION="MCF" MANDATORY="TRUE" NUM_VAL=

```

Figure 4.7.12-11. ESDT Detail Page (view XML)

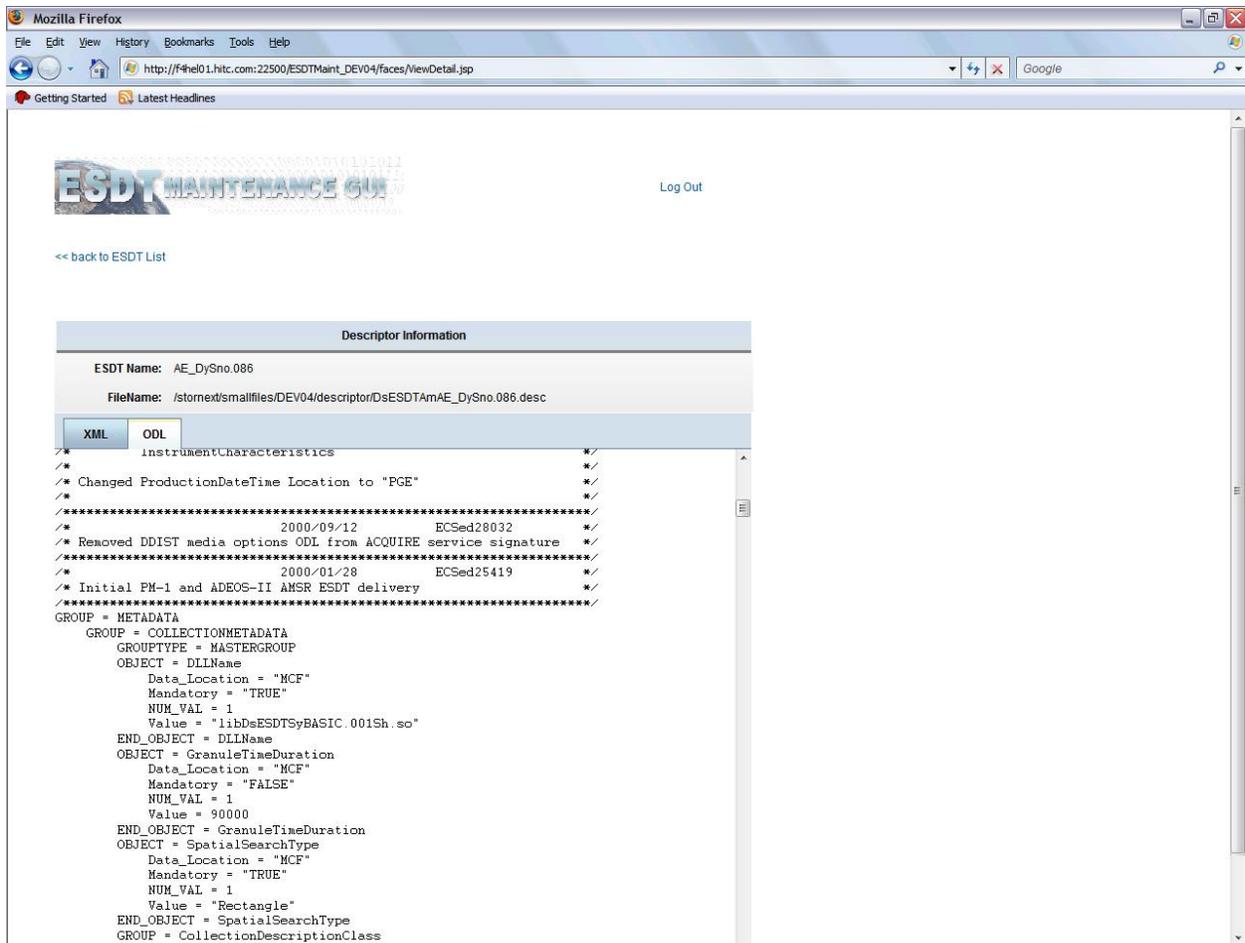


Figure 4.7.12-12. ESDT Detail Page (view ODL)

4.7.12.4 Install ESDT Page

This Install ESDT Page, as shown in Figure 4.7.12-13, appears when the operator clicks on the “Install new ESDTs / Update existing ESDTs” button from the List ESDT Page. From this page, the operator can install new or update existing ESDTs as well as cleanup failed ESDTs.

This page lists all of the ESDTs available for install, update, or cleanup. This list is populated from a pre-configured directory. In the far right column of each ESDT, the state of the ESDT can be viewed. The state indicates if the ESDT is already installed, failed, or not yet installed. Using this information, the operator can choose the appropriate ESDT to install, update, or cleanup.

The page provides shortcuts in selecting ESDTs. There is a button to select all the ESDTs listed, a button to unselect all the ESDTs selected, a button to select all the uninstalled ESDTs, a button to select all the failed ESDTs, and a button to select all the installed ESDTs.

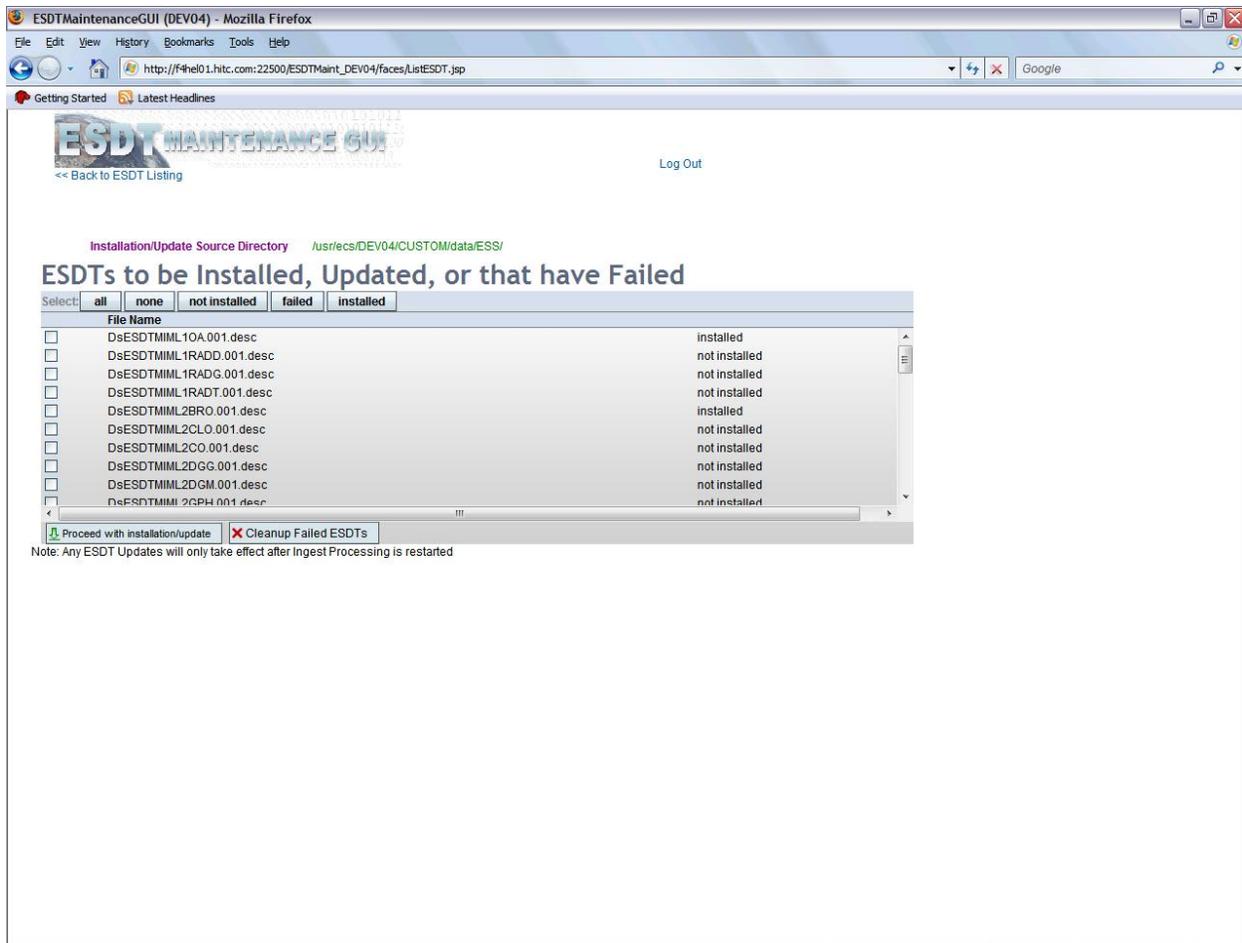


Figure 4.7.12-13. Install ESDT Page

4.7.12.4.1 Install/Update ESDTs

An operator performs installation or update on ESDTs by first selecting one or more ESDTs from the list as shown in Figure 4.7.12-14. The ESDT selected must be in the installed or not installed state.

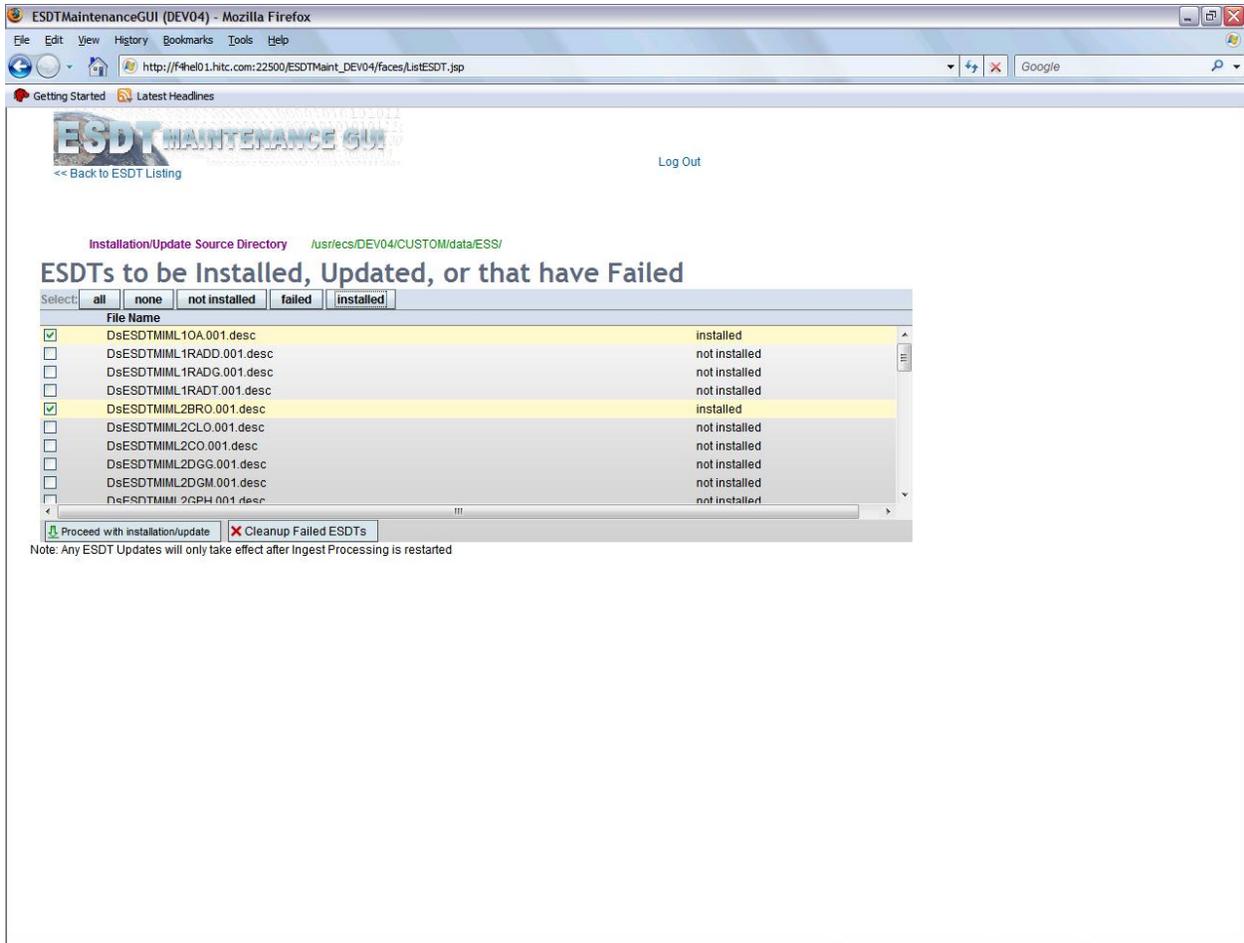


Figure 4.7.12-14. Install/Update ESDTs

Click on the “Proceed with installation/update” button to perform the installation or update action on the selected ESDTs as shown in Figure 4.7.12-15.

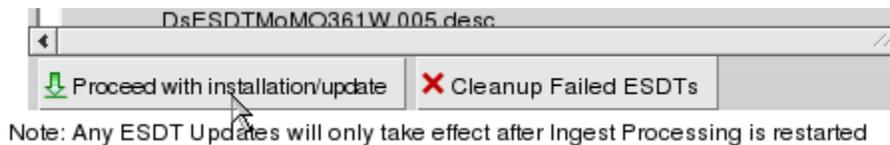


Figure 4.7.12-15. Install/Update ESDT button

An example of the result of the installation and update action is displayed in Figure 4.7.12-16.

If the installation or update completes successfully for all ESDTs, the installation files will be removed from this list, and a message will be displayed at the top of the screen indicating the success.

If the installation or update did not succeed for one or more ESDTs, a general error message will be displayed at the top of the screen. The “ESDTs with Errors” table at the top will display the detailed error information next to the failed ESDTs.

If an error is encountered during installation or update for any reason (i.e. validation error), the installation for that particular ESDT will fail, but other ESDTs will continue processing. As ESDTs are successfully installed or updated, the descriptor files are removed from the pre-configured (installation source) directory. Any files that remain in the list are those that failed installation/update or those that were not selected for processing.

The operator can choose to fix the problems and try the installation/update by selecting any of the remaining files from the list and click on the “Proceed with Installation/Update” button.

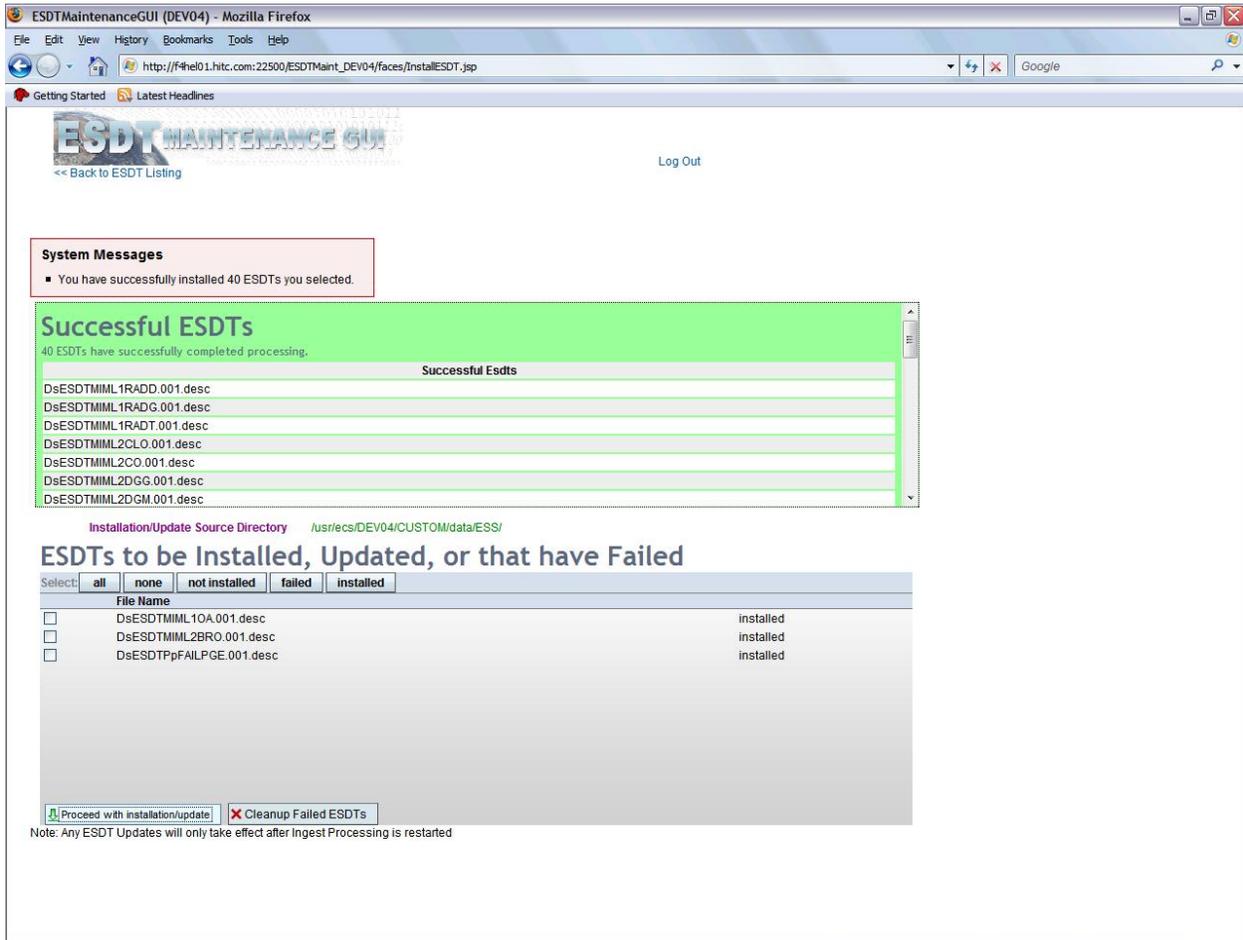


Figure 4.7.12-16. Install/Update ESDTs Result Page

4.7.12.4.2 Cleanup Failed ESDTs

An operator performs cleanup of failed ESDTs by first selecting one or more ESDTs from the list that are in the “failed” state. Then click on the “Cleanup Failed ESDTs” button to perform the cleanup action on the selected ESDTs as shown in Figure 4.7.12-17.

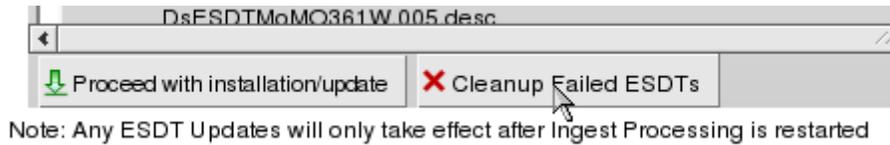


Figure 4.7.12-17. Cleanup Failed ESDTs button

An example of the result of the cleanup failed ESDTs action displayed in Figure 4.7.12-18.

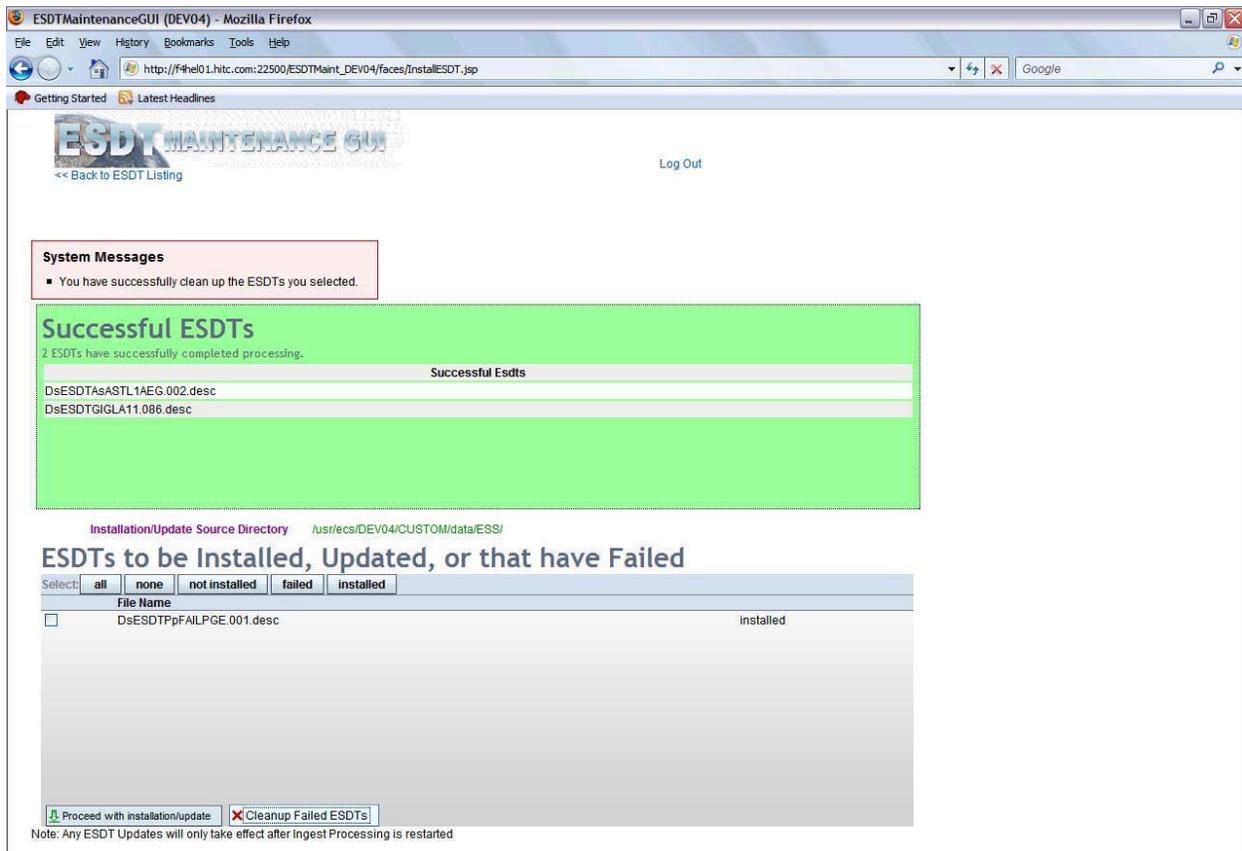


Figure 4.7.12-18. Cleanup Failed ESDTs Result Page

4.7.12.5 Browser Requirements

The specific browser requirements are stated elsewhere in this document. The recommended browsers are the only ones that should be used, as other browsers may not handle rendering and JavaScript correctly (for example, IE handles some JavaScript differently than Firefox).

JavaScript must also be enabled to run the application. In most cases, the cache size is automatically set and should be sufficient. Java is not required and need not be enabled in the browser to run the ESDT Maintenance GUI.

This page intentionally left blank.

4.7.13 AIM Granule Deletion Utilities

The Granule Deletion Utilities are a set of command line utilities, including EcDsBulkSearch, EcDsBulkDelete, EcDsBulkUndelete and EcDsDeletionCleanup, which provides the EMD Operations Staff with the ability to search granules for deletion, mark granules for deletion, restore (undelete) granules marked for deletion, and physically remove (clean up) granules which have been marked for deletion.

Granule deletion involves two phases:

- Phase I: Marking granules for deletion. Granules may be marked in one of two ways: 1) by using the `-physical` option of the EcDsBulkDelete utility, which sets the deleteEffectiveDate in DsMdGranules in the AIM inventory database; or 2) by using the `-dfa` option of the EcDsBulkDelete utility, which sets the DeleteFromArchive flag in DsMdGranules in the AIM inventory database. Once granules have been marked for deletion in either of these two ways, they are no longer accessible and are eligible for physical deletion in Phase II.
- Phase II: Physical deletion of marked/flagged granules. If a granule has been marked for deletion with the `-physical` option of the EcDsBulkDelete utility, physical deletion will remove: a) all entries for the granule from the inventory database; b) the xml metadata file for the granule from the xml archive; and c) the science file(s) for the granule from the science file archive.

If a granule has been marked for deletion with the `-dfa` option of the EcDsBulkDelete utility, physical deletion will remove the science file(s) for the granule from the science file archive, but will NOT remove the inventory database entries for the granule, nor will it remove the xml metadata file for the granule.

To delete granules from the archive, Operations Staff must perform the following sequence of activities: Generate a GeoID file listing all granules to be deleted, mark granules for deletion (Phase I), and physically delete granules (Phase II). Operations staff may also choose to restore (Undelete) granules marked for deletion, instead of proceeding with physical deletion of these granules. Once granules have been physically deleted (Phase II), restoration of the granules is not possible.

By default, when a science granule is marked for deletion with the `-physical` option, the EcDsBulkDelete utility will also mark all Browse, PH and QA granules associated with the science granule for deletion, unless the associated granules are also referenced by other non-deleted science granules. The operator may chose to override this default behavior and suppress deletion of associated Browse, PH, and QA granules altogether.

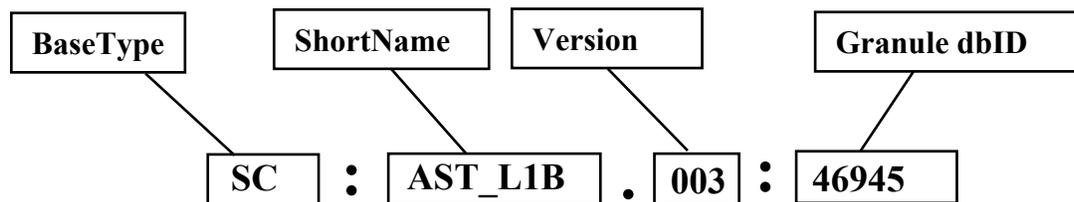
The sections below describe in detail the use of the EcDsBulkSearch, EcDsBulkDelete, EcDsBulkUndelete, and EcDsDeletionCleanup utilities.

Conventions of the command line used in the following sections

Convention	Description	Example
[]	Optional command parameter	[-localgranulefile <path/filename>]
	Only one of the parameters can be specified	[-physical -dfa] means either [-physical] or [-dfa]
<>	Require user to specify a value for a command line parameter	</home/labuser/myGEOid_file>

4.7.13.1 Generate a GeoID file

A GEO ID, formally called an Internal Granule Identifier, is a formatted string consisting of four segments. The following figure shows its structure:



BaseType is symbolized by two capital letters. Valid basetypes are:

- SC** – Science granule
- BR** – Browse granule
- PH** – Processing history granule
- QA** – Quality Assurance granule
- DP** – DAP granule
- LM** – Limited Granule

ShortName represents the data type of the granule such as “AST_L1B”.

Version ID segment represents the version id for the granule. It is always a 3 digit string, such as “003”.

Granule dbID is a unique inventory database ID for the granule.

The GeoID file consists of a list of granules represented by their GeoIDs, one GEO ID per line, as represented in the figure below:

GeoID File structure

```
SC:AST_L1B.003:46940
SC:AST_L1B.003:46943
SC:AST_L1B.003:46945
BR:Browse.001:46942
BR:Browse.001:46944
```

A prerequisite to marking granules for deletion is having a file of GeoIDs, which is used as input to the EcDsBulkDelete Utility. Although it is possible to manually create this file, an easier way is to use the EcDsBulkSearch Utility to generate a file containing a list of GeoIDs based on criteria specified when running the EcDsBulkSearch Utility.

4.7.13.1.1 Using the Bulk Search Utility

The generic format for invoking the Bulk Search Utility (EcDsBulkSearch) is the following:

```
►EcDsBulkSearch.pl [-begindate <DATETIME>]
                    [-enddate <DATETIME>]
                    [-acquirebegin <mm/dd/yyyy> [<hh:mm:ss>] ]
                    [-acquireend <mm/dd/yyyy> [<hh:mm:ss>] ]
                    [-insertbegin <DATETIME>]
                    [-insertend <DATETIME>]
                    [-localgranulefile <path/filename>]
                    [-physical | -dfa]
                    [-name <ShortName>]
                    [-version <VersionID>]
                    [-limit <n>]
                    -user <db_user>
                    [-password <passwd>]
```

-geoidfile <path/filename>
 -mode <MODE>
 -server <SYBASE_SERVER>
 -database <DB_NAME>

Table 4.7.13-1 provides a description of the command line parameters for the EcDsBulkSearch utility.

Table 4.7.13-1. Command Line Parameters of the EcDsBulkSearch.pl (1 of 2)

Parameter Name	Mandatory	Description
name	No	ESDT Short Name of the granules to delete.
version	No	ESDT Version ID of the granules to delete.
begindate	No	<mm/dd/yyyy> [<hh:mm:ss>] Search only for granules with a BeginningDateTime greater than or equal to the specified date and time.
enddate	No	<mm/dd/yyyy> [<hh:mm:ss>] Search only for granules with an EndingDateTime less than or equal to the specified date and time.
acquirebegin	No	<mm/dd/yyyy> [<hh:mm:ss>] Search only for granules with a BeginningDateTime greater than or equal to the specified date and time. This option is the same as '-begindate', except that it can be combined with 'acquireend' and used in a 'BETWEEN' clause.
acquireend	No	<mm/dd/yyyy> [<hh:mm:ss>] Search only for granules with a BeginningDateTime less than or equal to the specified date and time. This option is usually used in conjunction with 'acquirebegin'.
insertbegin	No	<mm/dd/yyyy> [<hh:mm:ss>] Search only for granules with an insertTime greater than or equal to the specified date and time
insertend	No	<mm/dd/yyyy> [<hh:mm:ss>] Search only for granules with an insertTime less than or equal to the specified data and time
localgranulefile	No	The name of a file containing Local Granule IDs to be converted into Geoids
geoidfile	Yes	Full path name of output file containing geoids.
physical	No	Search only for granules marked for deletion using the -physical option of the EcDsBulkDelete utility.
dfa	No	Search only for granules marked for deletion using the -dfa option of the EcDsBulkDelete utility.
mode	Yes	The ECS mode in which the utility is to operate; this parameter can be omitted if the environment variable MODE is set.

Table 4.7.13-1. Command Line Parameters of the EcDsBulkSearch.pl (2 of 2)

Parameter Name	Mandatory	Description
limit	No	Search will return up to <n> granules specified by limit (see below for sort order)
user	Yes	The user ID for database login
server	Yes	The name of the Sybase server. This parameter may be omitted if the environment variable SYB_SQL_SERVER is set.
database	Yes	The name of the database. This parameter may be omitted if the environment variable SYB_DB_NAME is set.
password	No	The inventory database login password. The utility will prompt the user to enter the password if it is not specified on the command line (for security reasons, it is not recommended to specify the password on the command line)

The output of the Bulk Search utility is the GeoID File, a plain text file which contains a list of internal granule identifiers (also called GEOIDs). The user must specify the path and geoidfile name. The user running the EcDsBulkSearch.pl utility script must have write privileges for the specified geoid file directory.

The Bulk Search utility also outputs a search report file under the same directory as the GEOID file, having the same filename as the geoidfile but terminated with a **.rpt** extension.

The GeoIDs in the geoidfile and the report file are ordered or sorted in a way that depends on the options specified on the command line:

If “-limit” is specified:

- 1) The granules will be ordered by **BeginningDateTime** if **-begindate** or **-acquirebegin** are specified;
- 2) The granules will be ordered by **insertTime** if **-insertbegin** is specified;
- 3) The granules will be ordered by dbID for all other situations.

No “-limit” option specified:

The granules will be always ordered by dbID.

The following sections describe twelve typical search scenarios. In order to simplify the command line, we assume that the user set and exported the following environment variables before running the search utility:

```
setenv MODE <MODE>
setenv SYB_SQL_SERVER xxdbl01_svr
setenv SYB_DB_NAME EcInDb_<MODE>
```

No **-password** is specified in any of the following commands. The utility will prompt the user to enter the password during runtime.

NOTE: The following search scenarios are those commonly used; they do not exhaust all possible combinations.

4.7.13.1.2 Search Granules by ShortName, VersionID and Inclusive Temporal Range

One common scenario is to search for a set of granules by specifying a ShortName, a VersionID, and an inclusive temporal range [**acquirebegin** , **acquireend**] such that the BeginningDateTime of the desired granules is greater than or equal to the date time specified by **acquirebegin**, and less than or equal to the date time specified by **acquireend** (**acquirebegin** ≤ BeginningDateTime ≤ **acquireend**). The following command generates a Geoid file containing granules which meet the above search criteria, sorted in ascending order of dbID:

```
►EcDsBulkSearch.pl -geoidfile <path/geoidfile_name>
                    -name <ShortName>
                    -version <VersionID>
                    -acquirebegin <mm/dd/yyyy> [<hh:mm:ss>]
                    -acquireend <mm/dd/yyyy> [<hh:mm:ss>]
                    -user <db_userid>
```

4.7.13.1.3 Search a Limited Number of Granules

The search in 4.7.13.1.2 may return a large number of granules. A limited number of granules can be output by specifying option **-limit <n>** where **n** is an integer. The Bulk Search Utility sorts the granules according to the rules in 4.7.13.1.1 and returns the first n granules. The following search is the same as 4.7.13.1.2 except for the **-limit** option. The **-limit** option may also be used with other search scenarios:

```
►EcDsBulkSearch.pl -geoidfile <path/geoidfile_name>
                    -name <ShortName>
                    -version <VersionID>
                    -acquirebegin <mm/dd/yyyy> [<hh:mm:ss>]
                    -acquireend <mm/dd/yyyy> [<hh:mm:ss>]
                    -user <db_userid>
                    -limit <n>
```

4.7.13.1.4 Search a Set of Granules for Which the Acquisition Date is between a Specified BeginningDateTime and EndingDateTime

With the EcDsBulkSearch utility, Operations staff can search for a set of granules by specifying a ShortName, a VersionID, and an inclusive date time range [**begindate**, **enddate**] such that the BeginningDateTime of the desired granules is greater than or equal to **begindate**, and the EndingDateTime of the desired granules is less than or equal to **enddate** (BeginningDateTime \geq **begindate** AND EndingDateTime \leq **enddate**). The following command generates a Geo ID file containing granules which meet the above search criteria, sorted in ascending order of dbID:

```
►EcDsBulkSearch.pl -geoidfile <path/geoidfile_name>
                    -name <ShortName>
                    -version <VersionID>
                    -begindate <mm/dd/yyyy> [<hh:mm:ss>]
                    -enddate <mm/dd/yyyy> [<hh:mm:ss>]
                    -user <db_userid>
```

4.7.13.1.5 Search a Set of Granules for Which the BeginningDateTime is Greater Than or Equal to a Specified Date Time

With the EcDsBulkSearch utility, Operations staff can search for a set of granules by specifying a ShortName, a VersionID, and an open-ended time range [**begindate**, ∞] such that the BeginningDateTime of the desired granules is greater than or equal to **begindate** (BeginningDateTime \geq **begindate**). The following command generates a Geo ID file containing granules which meet the above search criteria, sorted in ascending order of dbID:

```
►EcDsBulkSearch.pl -geoidfile <path/geoidfile_name>
                    -name <ShortName>
                    -version <VersionID>
                    -begindate <mm/dd/yyyy> [<hh:mm:ss>]
                    -user <db_userid>
```

4.7.13.1.6 Search a Set of Granules for Which the BeginningDateTime is Less Than or Equal to a Specified Date Time

With the EcDsBulkSearch utility, Operations staff can search for a set of granules by specifying a ShortName, a VersionID, and a maximum value (**acquireend**) such that the BeginningDateTime of the desired granules is less than or equal to the specified **acquireend**. (BeginningDateTime \leq **acquireend**). The following command generates a GeoID file containing granules which meet the above search criteria, sorted in ascending order of dbID:

```
►EcDsBulkSearch.pl -geoidfile <path/geoidfile_name>
```

```
-name <ShortName>
-version <VersionID>
-acquireend <mm/dd/yyyy> [<hh:mm:ss>]
-user <db_userid>
```

4.7.13.1.7 Search a Set of Granules for Which the EndingDateTime is Less Than or Equal to a Specified Date Time

With the EcDsBulkSearch utility, Operations Staff search for a set of granules by specifying a ShortName, a VersionID, and a maximum value (**enddate**) such that the EndingDateTime of the desired granules is less than or equal to the specified **enddate** (EndingDateTime ≤ **enddate**). The following command generates a GeoID file containing granules which meet the above search criteria, sorted in ascending order of dbID:

```
►EcDsBulkSearch.pl -geoidfile <path/geoidfile_name>
-name <ShortName>
-version <VersionID>
-enddate <mm/dd/yyyy> [<hh:mm:ss>]
-user <db_userid>
```

4.7.13.1.8 Search a Set of Granules for Which the insertTime is Within the Specified Date Time Range

With the EcDsBulkSearch utility, Operations staff can search for a set of granules by specifying a ShortName, a VersionID, and inclusive date time range [**insertbegin**, **insertend**] such that the insertTime of the desired granules is bounded by specified values of **insertbegin** and **insertend** (**insertbegin** ≤ insertTime ≤ **insertend**). The following command generates a GeoID file containing granules which meet the above search criteria, sorted in ascending order of dbID:

```
►EcDsBulkSearch.pl -geoidfile <path/geoidfile_name>
-name <ShortName>
-version <VersionID>
-insertbegin <mm/dd/yyyy> [<hh:mm:ss>]
-insertend <mm/dd/yyyy> [<hh:mm:ss>]
-user <db_userid>
```

4.7.13.1.9 Search a Set of Granules for which the insertTime is Greater than or Equal to the Specified Date Time

With the EcDsBulkSearch utility, Operations staff can search for a set of granules by specifying a ShortName, a VersionID, and a minimum date time (**insertbegin**) such that the insertTime of the desired granules is greater than or equal to the date time specified by **insertbegin** ($\text{insertTime} \geq \text{insertbegin}$). The following command generates a GeoID file containing granules which meet the above search criteria, ordered by granule dbID, sorted in ascending order of dbID:

```
►EcDsBulkSearch.pl -geoidfile <path/geoidfile_name>
                    -name <ShortName>
                    -version <VersionID>
                    -insertbegin <mm/dd/yyyy> [<hh:mm:ss>]
                    -user <db_userid>
```

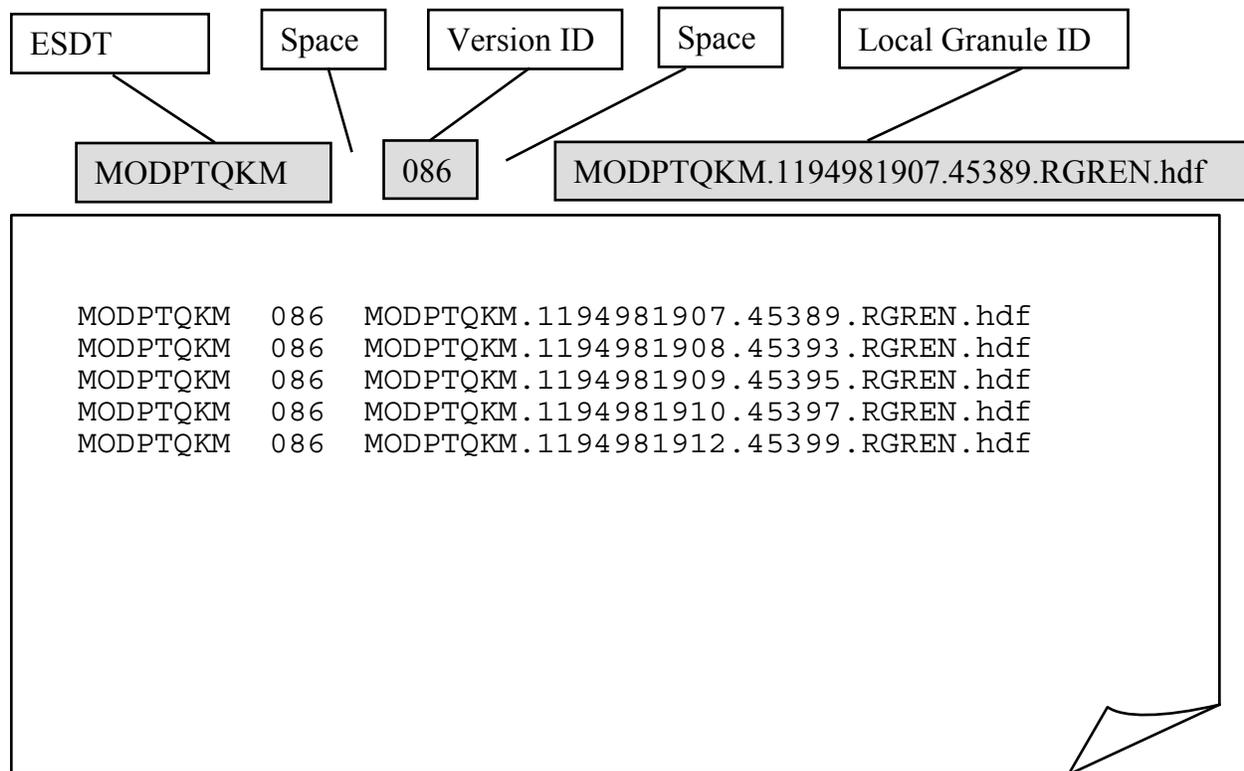
4.7.13.1.10 Search a Set of Granules for Which the insertTime is Less than or Equal to the Specified Date Time

With the EcDsBulkSearch utility, Operations staff can search for a set of granules by specifying a ShortName, a VersionID, and a maximum date time (**insertend**) such that the insertTime of the desired granules is less than or equal to the date time specified by **insertend** ($\text{insertTime} \leq \text{insertend}$). The following command generates a GeoID file containing granules which meet the above search criteria, sorted in ascending order of dbID:

```
►EcDsBulkSearch.pl -geoidfile <path/geoidfile_name>
                    -name <ShortName>
                    -version <VersionID>
                    -insertend <mm/dd/yyyy> [<hh:mm:ss>]
                    -user <db_userid>
```

4.7.13.1.11 Convert Local Granule ID(s) to GeoID(s)

There may be a situation in which Operations staff already know the shortname, version and science filename (Local Granule ID) for list of granules to be deleted. Operations staff can convert this list of local granule IDs into a set of GeoIDs, which can be used by the EcDsBulkDelete utility to mark the granules for deletion. One way to achieve such a conversion is through the use of the EcDsBulkSearch utility and a **localgranulefile**. A localgranulefile is simply a text file containing a list of granules, formatted as “ShortName VersionID LocalGranuleId”, one such granule per line. The following figure shows the structure and format of a **localgranulefile** :



The following command transforms the localgranulefile into an equivalent GeoID file. The resulting GeoIDs match the local granule IDs specified in the localgranulefile:

```

►EcDsBulkSearch.pl -geoidfile <path/geoidfile_name>
-localgranulefile <path/local_granule_file_name>
-user <db_userid>

```

4.7.13.1.12 Search for Granules Marked for Deletion with **-physical** option

The EcDsBulkSearch utility also allows Operations staff to search for granules that have already been marked for deletion. In this example, the EcDsBulkSearch utility returns a GeoID file of granules that have been marked for deletion using the **-physical** option of the EcDsBulkDelete utility (see 4.7.13.2.1), i.e., granules where the deleteEffectiveDate is set in the DsMdGranules table in the inventory database:

```

►EcDsBulkSearch.pl -physical
-user <db_userid>
-geoidfile <path/geoidfile_name>

```

4.7.13.1.13 Search for Granules Marked for Deletion with **-dfa** option

In this example, the EcDsBulkSearch utility returns a GeoID file of granules that have been marked for deletion using the **-dfa** option of the EcDsBulkDelete utility (see 4.7.13.2.2), i.e., granules where the DeleteFromArchive flag is set in the DsMdGranules table in the inventory database:

► **EcDsBulkSearch.pl -dfa**

-user <db_userid>

-geoidfile <path/geoidfile_name>

4.7.13.2 Granule Deletion, Phase I: Marking Granules for Deletion

In the first phase of the granule deletion process, the EcDsBulkDelete.pl utility marks the specified granules for deletion. The marked granules can no longer be accessed but their associated files and metadata have not yet been physically removed.

The generic command line format for the EcDsBulkDelete utility is the following:

► **EcDsBulkDelete.pl -physical | -dfa**

[-delref]

[-noassoc]

-user <db_user>

[-password <passwd>]

-geoidfile <path/filename>

-mode <MODE>

-server <SYBASE_SERVER>

-database <DB_NAME>

[-log <log_file_name>]

Table 4.7.13-2 provides a description of the parameters used in executing the EcDsBulkDelete.pl script.

Table 4.7.13-2. Command Line Parameters for EcDsBulkDelete.pl

Parameter Name	Mandatory	Description
geoidfile	Yes	Name of input file containing geoids of the granules to be marked for deletion.
physical	Yes if not dfa	Specifying this parameter will mark granules in the geoidfile for deletion of science files as well as deletion of inventory database entries and xml metadata.
dfa	Yes if not physical	Specifying this parameter (can not combine with physical) will mark the granules in the geoidfile for deletion of science files only. (Inventory database entries and xml metadata files associated with the granules will not be deleted).
delref	No	Optional. When given, indicates that non SC/LM granules should be deleted even if they are associated with undeleted SC/LM granules. Only used with –physical option; ignored with –dfa option. Note: This option has no effect on deleting SC/LM granules. They are always deleted regardless of whether they are referenced or not.
noassoc	No	Optional. When given, indicates that associated granules (Browse, PH, QA) will not be marked for deletion. Only used with physical option; ignored with –dfa option.
mode	Yes	The ECS mode in which the utility is to operate. This parameter may be omitted if the environment variable MODE is set.
user	Yes	The user ID for database login
server	Yes	The name of the Sybase server. This parameter may be omitted if the environment variable SYB_SQL_SERVER is set
database	Yes	The name of the database. This parameter may be omitted if the environment variable SYB_DB_NAME is set
password	No	The inventory database login password. The utility will prompt the user to enter the password if it is not specified on the command line (for security reasons, it is not recommended to specify the password on the command line)
log	No	The name of the file to which utility messages will be logged. If this is not provided, the log file name will default to EcDsBulkDelete.log. The log file will be stored in the /usr/ecs/<MODE>/CUSTOM/logs directory.

In order to simplify the command line, the user may set the following environment variables before running the Bulk Delete utility:

```

setenv MODE <MODE>
setenv SYB_SQL_SERVER xxdbl01_svr
setenv SYB_DB_NAME EcInDb_<MODE>

```

The EcDsBulkDelete utility may be used either to mark granules for physical deletion of both data files and metadata (via the `-physical` command line option), or to mark granules for physical deletion of the data files only (via the `-dfa` command line option). Examples of these use cases are shown below.

4.7.13.2.1 Mark Granules for physical deletion of science files and metadata

The EcDsBulkDelete utility may be used to mark granules for physical deletion of all science file(s) associated with the granules, as well as all metadata associated with the granules, including the xml metadata file in the xml archive and all entries for the granule in the inventory database.

Executing the following command will mark all granules in the specified geoid file for physical deletion of the granule's science files as well as its metadata:

► **EcDsBulkDelete.pl -physical**

```
-geoidfile <path/geoidfile_name>  
-user <db_userid>  
[-log <log_file_name.log>]
```

In this example, since the `-noassoc` parameter is not used, all associated granules, such as Browse, QA and PH granules, are marked for deletion as well.

4.7.13.2.2 Mark Granules for deletion of science files only (Delete From Archive)

Executing the following command will mark all granules in the specified geoid file for “DFA” deletion, i.e., deletion of the science files only (the xml metadata file in the xml archive and all entries for the granule in the inventory database will remain):

► **EcDsBulkDelete.pl -dfa**

```
-geoidfile <path/geoidfile_name>  
-user <db_userid>  
[-log <log_file_name.log>]
```

Marking granules for “DFA” deletion will not impact any associated granules, such as Browse, QA and PH granules. Although the utility doesn't prevent the user from specifying `-delref` and `-noassoc` with the `-dfa` option, these two optional parameters are ignored when used with the `-dfa` option.

4.7.13.3 Undeleting Granules

Granules marked for deletion by the EcDsBulkDelete utility can be restored (undeleted) by the EcDsBulkUndelete utility. Granules which have been undeleted are once again accessible and are no longer eligible for physical deletion by the EcDsDeletionCleanup utility.

The EcDsBulkUndelete utility takes as input a geoid file, in which all granules intended to be undeleted are listed. Either the -physical or -dfa option must be specified. The -noassoc option may be used with the -physical option to indicate that associated granules (Browse, PH, QA) marked for deletion should not be restored/undeleted.

The generic command line format for the EcDsBulkUndelete utility is the following:

```

►EcDsBulkUndelete.pl -physical | -dfa
    [-noassoc]
    -user <db_user>
    [-password <passwd>]
    -geoidfile <path/filename>
    -mode <MODE>
    -server <SYBASE_SERVER>
    -database <DB_NAME>
    [-log <log_file_name>]
  
```

Table 4.7.13-3 provides a description of the parameters used in executing the EcDsBulkUndelete.pl script.

Table 4.7.13-3. Command Line Parameters for EcDsBulkUndelete.pl (1 of 2)

Parameter Name	Mandatory	Description
geoidfile	Yes	Name of input file containing geoids of the granules to be restored/undeleted..
physical	Yes	Specifying this parameter will restore (“undelete”) granules specified in the geoid file which have been previously marked for deletion using the -physical option of the EcDsBulkDelete utility..
Dfa	Yes	Specifying this parameter (can not combine with physical) will restore (“undelete”) granules specified in the geoid file which have been previously marked for deletion using the -dfa option of the EcDsBulkDelete utility..
noassoc	No	Optional. When given, indicates that associated granules (Browse, PH, QA) will not be restored (“undeleted”). Only valid with the -physical option; ignored with -dfa option.
Mode	Yes	The ECS mode in which the utility is to operate; this parameter may be omitted if the environment variable MODE is set.
User	Yes	The user ID for database login
server	Yes	The name of Sybase server. This parameter may be omitted if the environment variable SYB_SQL_SERVER is set.

Table 4.7.13-3. Command Line Parameters for EcDsBulkUndelete.pl (2 of 2)

Parameter Name	Mandatory	Description
database	Yes	The name of the database. This parameter may be omitted if the environment variable SYB_DB_NAME is set.
password	No	The inventory database login password. The utility will prompt the user to enter the password if it is not specified on the command line (for security reasons, it is not recommended to specify the password on the command line)
Log	No	The name of the file to which utility messages will be logged. If this is not provided, the log file name will default to EcDsBulkUndelete.log. The log file will be stored in the /usr/ecs/<MODE>/CUSTOM/logs directory.

In order to simplify the command line, the user can set the following environment variables before running the EcDsBulkUndelete utility:

```
setenv MODE <MODE>
setenv SYB_SQL_SERVER xxdbi01_svr
setenv SYB_DB_NAME EcInDb_<MODE>
```

The EcDsBulkUndelete utility may be run with either the `-physical` or `-dfa` option. Examples of each of these options are given in the next two sections below.

4.7.13.3.1 Running the EcDsBulkUndelete utility with the `-physical` option

Prior to running the EcDsBulkUndelete utility, a geoid file which lists granules marked for deletion with the `-physical` option should be prepared manually or by running EcDsBulkSearch with the `-physical` option.

Executing the following command will “undelete” granules listed in the specified geoid file (i.e., will reset the deleteEffectiveDate for these granules in the DsMdGranules table in the inventory database):

```
►EcDsBulkUndelete.pl -physical
    [ -noassoc ]
    -geoidfile <path/geoidfile_name>
    -user <db_userid>
    [-log <log_file_name.log>]
```

4.7.13.3.2 Running the EcDsBulkUndelete utility with the `-dfa` option

Prior to running the EcDsBulkUndelete utility, a geoid file which lists granules marked for deletion with the `-dfa` option should be prepared manually or by running EcDsBulkSearch with the `-dfa` option.

Executing the following command will “undelete” granules listed in the specified geoid file (i.e., will reset the DeleteFromArchive flag for these granules in the DsMdGranules table in the inventory database):

```
►EcDsBulkUndelete.pl -dfa
                        -geoidfile <path/geoidfile_name>
                        -user <db_userid>
                        [-log <log_file_name.log>]
```

4.7.13.4 Granule Deletion, Phase II: Physical Deletion or Cleanup

In phase II of the granule deletion process, the user runs the EcDsDeletionCleanup utility against granules that have been marked for deletion in Phase I. The EcDsDeletionCleanup utility permanently removes science files and optionally metadata for granules which are marked for deletion.

For granules marked for deletion with either the `-physical` or `-dfa` option of the EcDsBulkDelete utility, the EcDsDeletionCleanup utility will remove all science files for the granule from the archive. If the granules were marked for deletion with the `-physical` option, and the `-noassoc` option was not specified, all files for associated Browse, PH, and QA granules will also be removed from the archive.

If granules were marked for deletion using the `-physical` option of the EcDsBulkDelete utility, the EcDsDeletionCleanup utility will also clean up all metadata for the marked granules. This includes removing all inventory database entries for the marked granules, and removing corresponding metadata XML files from the XML archive, as well.

When the EcDsDeletionCleanup utility is executed, it will check if there is any unfinished work from previous run(s). If so, the utility will prompt the user with a selection menu. The operator may choose to rerun the unfinished run(s) only, which will be resumed from the interrupted point and continue physical cleanup for granules identified for deletion in the previous interrupted run(s). Alternately, the operator may choose to complete unfinished run(s) and start a new run, which will complete cleanup from the previous interrupted run(s), but also clean up granules identified for deletion by the new run.

Unlike the Bulk Search, Bulk Delete and Undelete utilities, EcDsDeletionCleanup requires user interactions during execution.

The generic format for the command line of the EcDsDeletionCleanup utility is the following:

►EcDsDeletionCleanup.pl -user <db_user>
-mode <MODE>
-server <SYBASE_SERVER>
-database <DB_NAME>
[-batch <number>]
[-grbatch <number>]
[-phbatch <number>]
[-log <log_file_name>]
[-xmlbatch <number>]
[-databatch <number>]
[-logbatch <number>]

Table 4.7.13-4 provides a description of the parameters used in executing the EcDsDeletionCleanup.pl script.

Table 4.7.13-4. Command Line Parameters for EcDsDeletionCleanup.pl (1 of 3)

Parameter Name	Mandatory	Description
Mode	Yes	The ECS mode in which the utility is to operate. This parameter may be omitted if the environment variable MODE is set.
User	Yes	The user ID for database login
server	Yes	The name of Sybase server. This parameter may be omitted if the environment variable SYB_SQL_SERVER is set.
database	Yes	The name of the database. This parameter may be omitted if the environment variable SYB_DB_NAME is set.
Log	No	The name of the file to which utility messages will be logged. If this is not provided, the utility will prompt the user at runtime to either supply a log file name, or to accept the default log file name, EcDsDeletionCleanup.log. The log file will be stored in the /usr/ecs/<MODE>/CUSTOM/logs directory.

Table 4.7.13-4. Command Line Parameters for EcDsDeletionCleanup.pl (2 of 3)

Parameter Name	Mandatory	Description
Batch	No	<p>This is a tuning parameter. It represents the batch size for populating the DsStPendingDelete table. This parameter may be omitted if the environment variable BATCH_SIZE_GRANULE is set.</p> <p>If the environment variable BATCH_SIZE_GRANULE is set, and -batch <number> is also specified, the value from command line parameter -batch will be used.</p> <p>If neither the environment variable BATCH_SIZE_GRANULE is set nor -batch is specified, the user will be prompted at runtime to enter a value (a value of 10000 is suggested by the prompt text).</p>
grbatch	No	<p>This is a tuning parameter. It represents the batch size used for physical granule file cleanup. This parameter may be omitted if the environment variable BATCH_SIZE_GRANULE is set. (Note: the environment variable BATCH_SIZE_GRANULE applies to both -batch and -grbatch).</p> <p>If the environment variable BATCH_SIZE_GRANULE is set, and -grbatch <number> is also specified, the value from command line parameter -grbatch will be used.</p> <p>If neither the environment variable BATCH_SIZE_GRANULE is set nor -grbatch is specified, the user will be prompted at runtime to enter a value (a value of 10000 is suggested by the prompt text).</p>
phbatch	No	<p>This is a tuning parameter. It represents the phbatch size for PH granule deletion. Because PH granule deletion could be time consuming, setting a high batch size for PH granule deletion could lock the database for a long period of time; therefore, a low value of phbatch is recommended. This parameter may be omitted if the environment variable BATCH_SIZE_PH is set.</p> <p>If the environment variable BATCH_SIZE_PH is set and -phbatch <number> is also specified, the value from command line parameter -phbatch will be used.</p> <p>If neither the environment variable BATCH_SIZE_PH is set nor -phbatch is specified, the user will be prompted at runtime to enter a value (a value of 5 is suggested by the prompt text).</p>
xmlbatch	No	<p>This is a tuning parameter. It represents the batch size for processing the deletion of XML files from the XML Archive. The utility will iterate through the deletion of XML files retrieving and deleting "xmlbatch" files per iteration. This controls memory growth of the utility when processing a large number of granules. If this parameter is not specified, the utility will prompt for a value (suggesting a size of 1000).</p>

Table 4.7.13-4. Command Line Parameters for EcDsDeletionCleanup.pl (3 of 3)

Parameter Name	Mandatory	Description
databatch	No	This is a tuning parameter. It represents the batch size for processing the deletion of data files from the “offline/backup” Archive (note: this archive is typically on tape). This utility doesn’t remove files from the “Online Archive.” The utility will iterate through the deletion of data files retrieving and deleting “databatch” files per iteration. This controls memory growth of the utility when processing a large number of granules. If this parameter is not specified, the utility will prompt for a value (suggesting a size of 10000).
logbatch	No	The “logbatch” parameter controls the frequency of progress messages in the log. If this parameter is not specified, the utility will prompt for a value (suggesting a size of 100).

The following sections describe two typical physical cleanup scenarios. In order to simplify the command line, we assume that the user has set the following environment variables before running the EcDsDeletionCleanup utility:

```
setenv MODE <MODE>
setenv SYB_SQL_SERVER xxdbi01_srvr
setenv SYB_DB_NAME EcInDb_<MODE>
```

The utility will prompt the user to enter the database password during runtime.

4.7.13.4.1 Run a New Physical Cleanup

Command for starting a new run:

```
➤EcDsDeletionCleanup.pl -user <db_user>
    [-batch <number>]
    [-grbatch <number>]
    [-phbatch <number>]
    [-log <log_file_name>]
```

If no log file name is specified on the command line, the utility will prompt the user to enter one. The user also may select to use the default log file name, which will be EcDsDeletionCleanup.log. The log file will be created under the directory /usr/ecs/<MODE>/CUSTOM/logs/.

The physical cleanup utility EcDsDeletionCleanup will prompt for the database password, and then prompt for any required parameter(s) which have not been set via corresponding environment variables and were not specified on the command line.

The utility will first check if there are any granule(s) that have been marked for deletion (with either the `-physical` or `-dfa` option); if not, the utility will terminate.

If there are granules marked for deletion, the following menu will be displayed for user selection:

==== Menu for Lag Time ====

1. Select granules for a specific day (lag<n> or date <mm/dd/yyyy> format)

2. Select all granules older than a specific day (lag<n> or date <mm/dd/yyyy> format)

3. Quit

Select 1, 2 or 3: _

The user needs to enter 1, 2 or 3 for:

- 1 – Only cleanup granules whose deletion date falls into a single day specified by the lag time;
- 2 – Cleanup all granules whose effective deletion date is older than the date specified by the lag time;
- 3 – Nothing to cleanup, exit.

If the user chooses menu selection 1 or 2, the user will next be prompted to enter either a lag time in units of days OR a date (in `<mm/dd/yyyy>` format, such as `04/18/2003`). An entry of zero is equivalent to today's date. (See below for a more detailed description of how lag time is used.) Once a lag time or a date is entered, the user will be requested to confirm the entry. If the user answers "N", the utility will prompt the user to re-enter the lag time or date.

A lag time is used to exclude or include a set of granules marked for deletion from the current cleanup run.

For menu selection 1, if an integer `<n>` is entered for lag time, only granules which were marked for deletion at any time on the date `<n>` days ago (with respect to the current system date) will be eligible for clean up in the current run. For example, if the utility is run on `03/04/2008` and a lag time of 3 is specified, only granules which were marked for deletion at any time on `03/01/2008` will be eligible for clean up.

If a date such as `05/11/2008` is entered, only granules which were marked for deletion at any time on `05/11/2008` will be eligible for clean up in the current run.

For menu selection 2, if an integer `<n>` is entered for lag time, only granules which were marked for deletion more than `<n>` days ago (with respect to the current system date) will be eligible for clean up in the current run. For example, if the utility is run on `03/04/2008` and a lag time of 3 is specified, only granules which were marked for deletion on or before midnight on `03/01/2008` will be eligible for clean up in the current run. If a date such as `05/11/2008` is entered, only granules which were marked for deletion on or before midnight on `05/11/2008` will be eligible for clean up in the current run.

After a lag time is confirmed, the utility will display another menu for user selection:

==== *Menu for Data Type* ====

1. Specify datatype(s) and version for deletion by an input file

The file format: one ESDT.Version <AST_L1BT.001 or AST_L1B.001> per line*

2. Select all datatypes for deletion

3. Quit

Select 1, 2 or 3: _

The user needs to enter 1, 2 or 3 for:

- 1 – Cleanup granules marked for deletion which have an ESDT shortname and versionid in the input file. The input file lists one ESDT per line, in <shortname.versionid> format. A wildcard * may be used as part of the ESDT shortname;
- 2 – Cleanup all granules marked for deletion, regardless of their ESDT.version;
- 3 – Nothing to cleanup, exit.

Selecting 1 or 2 will start physical cleanup. The utility will present a list of ESDTs and granule counts to be deleted. The operator is prompted to continue or not. If “y” is selected the operator is then prompted to determine whether the utility should run interactively or not. If the operator selects to run interactively, the utility will delete a set of granules from the archive based upon the value of “databatch.” When the utility completes the “databatch” number of granule deletions, it presents an updated list of the remaining granules to be deleted and prompts the operator to continue again. This process is repeated until all files are deleted. If the operator specifies “n” to the prompt for running interactively, the utility will process all selected granules without any prompting. This is useful when the operator wants to run the utility unattended and is confident about what is going to be deleted. In all cases the utility progress and error information will be written in the log file.

4.7.13.4.2 Rerun unfinished Physical Cleanup

The EcDsDeletionCleanup utility always checks if there were any unprocessed granule(s) left over from a previous unfinished run(s). If so, when the EcDsDeletionCleanup utility is invoked normally (see 4.7.13.4.1), leftover information will be displayed and logged, and a menu will be displayed for the user to select how to run the cleanup:

Previous run was not completed, you can choose to:

1. Rerun unfinished run only

2. Start a new run which includes unfinished run(s)

3. Quit

Select 1, 2 or 3:

Select 1 to complete the unfinished run(s) only. Cleanup will resume from the interrupted point in the previous run(s). (For example,, start to cleanup leftover XML files which had not been cleaned up in previous run(s).)

Select 2 to complete the unfinished run(s) and start a new run. The new run will clean up granules which have been marked for deletion since the unfinished previous run(s) and which meet the lag time and data type criteria for the new run.

4.7.14 DataPool Checksum Verification Utility

The DataPool Checksum Verification utility (DPCV) provides a mechanism by which the ECS Operations Staff can perform checksum verification for files in the Data Pool. It can be scheduled and run as a background process to proactively verify the integrity of files in the Data Pool. For example, the utility could be set up as a background process that would verify the checksum of a file every “Nth” month by specifying a checksum verification option based on time elapsed since the last time checksum was verified. The utility could also be run on-demand by the DAAC operator to verify checksum values for a particular set of files.

- The utility is capable of performing checksum verification by sampling files based on ESDT and insert date range, or elapsed time since the last time checksum was verified, or a given granule list.
- According to the sampling options specified, the utility scans the appropriate files and verify their checksum values.
- Upon successful checksum verification, the utility will update the time when checksum was verified for each file in the DataPool database.
- Upon detection of checksum verification failure after a configurable number of retry attempts, the utility will log detailed information about the failure which will include granule ID, ESDT, insert time, complete file path and file name, along with the checksum information -- including checksum type, checksum values (computed value vs. the corresponding value stored in database), the last time the file was checksummed, checksum origin (who performed the last checksum). This information will also be provided in a report produced by the utility at the end of a run.
- The verification report will also include statistical summary information including total number of files checked, number of files that failed checksum, percentage of files that failed checksum, categorized by ESDT.
- This utility is designed such that the checksum verification can be throttled so it does not impact on-going daily operations.
- The primary use case of this utility is to perform checksum verification for DataPool as a background job. It performs all the checksum operations on the local host and does not distribute the workload to other hosts. The utility will be installed on all the EMD service hosts and can be started on multiple hosts if load balancing is desired.
- Since multiple instances of the utility can be started on the same host, operator should be aware of the number of instances that has been started to avoid overloading the system. All DPCV runs are logged in DIDpcvHistory table in the DataPool database.

4.7.14.1 Using the DataPool Checksum Verification Utility

The DataPool Checksum Verification utility should be started by the user cmshared (or similar). The DataPool Checksum Verification utility is started by entering the following command:

```
EcDIDPCVStart <mode> <command line parameters>
```

There are nine command line parameters that may be used. Table 4.7.14-1 provides a description of those parameters.

Table 4.7.14-1. Command Line Parameter

Parameter Name	Required	Description
verifyOnly	No	Optional parameter to specify whether to only verify existing checksum. When the option is present in the command line, DPCV will only verify checksum if it is present in the database; When the option is not present, DPCV will calculate a checksum for files that do not have checksum in database.
esdts	No	Optional parameter to specify ESDTs needs to be verified. Its value could be keyword ALL (meaning all ESDTs) or a specific list of ESDTs separated by " ". It can not be combined with the file option.
insertBeginTime	No	Optional parameter to specify lower limit of insertTime used to qualify granules to be verified. It can not be combined with the file option.
insertEndTime	No	Optional parameter to specify upper limit of insertTime used to qualify granules to be verified. It can not be combined with the file option.
daysSinceLastChecksum	No	Parameter to specify the cut off value of number of days since the file is last checksummed. Files that are checksummed within the cut off value of days will not be checksummed again.
file	No	Parameter to specify a list of DataPool granule ids to be verified. It can not be combined with the esdts, insertBeginTime or insertEndTime option.
percentage	No	Parameter to specify the percentage of files in the qualifying range that is verified.
fg	No	Parameter to specify the DPCV process to run as a foreground process. If present, it has to be the first parameter in the parameter list. By default, DPCV will run as a background process. This is reserved for cron job run.
noprompt	No	Parameter to specify the log file name not to be prompted back on the standard out. This is reserved for cron job run.

When running EcDIDPCVStart without any parameter on the command line, it will print out its usage. See below:

Usage: EcDIDpcvStart [-fg] <MODE> [-verifyOnly] [-esdts (keyword ALL or list of ShortName.VersionId e.g. ALL or "AE_Land.086|PH.001|QA.001")] [-insertBeginTime (MM/DD/YYYY HH:MM:SS)] [-insertEndTime (MM/DD/YYYY HH:MM:SS)] [-daysSinceLastChecksum (number of days)] [-file (text file containing DataPool GranuleIds)] [-percentage (integer from 0-100)] [-noprompt]

4.7.14.1.1 DataPool Checksum Verification Utility Command Line Examples

1. For all granules ingested within a period of time run:

```
EcDlDpcvStart OPS -verifyOnly -esdts ALL -insertBeginTime "11/27/2008 00:00:00" -insertEndTime "12/25/2008 23:59:59"
```

The DataPool Checksum Verification utility will perform checksum verification for all granule files ingested between Thanksgiving and Christmas that have existing checksum information.

```
EcDlDpcvStart OPS -verifyOnly -esdts ALL -insertBeginTime "11/27/2008 00:00:00" -insertEndTime "12/25/2008 23:59:59" -percentage 50
```

The DataPool Checksum Verification utility will perform checksum verification for 50% of the granule files ingested between Thanksgiving and Christmas that have existing checksum information.

```
EcDlDpcvStart OPS -verifyOnly -esdts ALL -insertBeginTime "11/27/2008 00:00:00" -insertEndTime "12/25/2008 23:59:59" -daysSinceLastChecksum 30
```

The DataPool Checksum Verification utility will perform checksum verification for all the granule files ingested between Thanksgiving and Christmas that have existing checksum information and haven't been verified for at last 30 days.

2. For granules belong to a list of specified ESDTs ingested within a period of time run:

```
EcDlDpcvStart OPS -esdts "AST_L1A.003|MOD29P1D.005" -insertBeginTime "11/27/2008 00:00:00" -insertEndTime "12/25/2008 23:59:59"
```

The DataPool Checksum Verification utility will perform checksum verification for all granule files that are of ESDT AST_L1A.003 or MOD29P1D.005 ingested between Thanksgiving and Christmas. If there is no existing checksum information, DPCV will calculate one based on the default checksum type and insert it into the database.

3. For a "file" run:

```
EcDlDpcvStart OPS -file dplgranuleids.txt
```

The DataPool Checksum Verification utility will perform checksum verification for all granule files that are listed in the dplgranuleids.txt.

4. For a cron run:

```
EcDlDpcvStart -fg OPS -noprompt -verifyOnly -esdts ALL -daysSinceLastChecksum 30
```

Put the above in the crontab to set up the cron job to verify checksum for files that have not been verified for at least 30 days.

4.7.14.2 DataPool Checksum Verification Utility Configuration File

The DataPool Checksum Verification utility uses a configuration file: EcDIDpcv.properties, located in /usr/ecs/<mode>/CUSTOM/cfg directory. The configuration parameters are stored in a PARAMETER = VALUE format with each parameter/value pair as a separate line entry in the file. Table 4.7.14-2 describes the configuration parameters.

Table 4.7.14-2. Configuration Parameters

Parameter Name	Description
PGM_ID	Program ID used for connecting to the Data Pool database.
SYBASE_USER	The user name for the Sybase connection.
SYBASE_HOST	The name of the host Sybase SQL server is on.
SYBASE_PORT	The port number of Sybase server
SYBASE_DBNAME	The name of the Data Pool database you intend to connect to
SYBASE_JDBC_DRIVER_CLASS	The Sybase jdbc driver class.
DB_RETRIES	The number of times the utility attempts to connect to the database before exiting. The recommended default is 5.
DB_SLEEPSECONDS	The number of seconds the utility waits ('sleep') between connection attempts. The recommended default is 10.
SQL_TIMEOUT_SECONDS	The number of seconds to timeout a db operation.
DPCV_EXPIRATION_TIME	The number of hours the utility uses to mark an un-finished process as expired.
DPCV_HISTORY_RETENTION_TIME	The number of days the utility uses to cleanup old DPCV run record in database.
SECONDS_BETWEEN_CHECKSUMS	The number of seconds between checksum operations
NUM_CHECKSUM_RETRIES	The number of retries on checksum failures
HOST_NAME	The host name where DPCV is running
VALIDATION_OUTPUT_DIR	The directory where the validation output files will be saved.

4.7.14.3 DataPool Checksum Verification Utility Main Screen

The DataPool Checksum Verification utility does not have a main screen. It has a command line interface only.

4.7.14.4 Required Operating Environment

The DataPool Checksum Verification utility will run on a Linux platform.

4.7.14.5 Databases

Table 4.7.14-3 lists the supporting products that this tool depends upon in order to function properly.

Table 4.7.14-3. Product Dependencies

Product Dependency	Protocols Used	Comments
DataPool Database	SQL	Via SQL server machines
Inventory Database	SQL	Via SQL server machines

4.7.14.6 Special Constraints

The DataPool Checksum Verification utility runs only if the Datapool database and Inventory database server is running and if the databases are available. It also assumes the stored procedures are present, EcDIDpcv is a registered user in both databases and have proper permission to execute the stored procedures.

4.7.14.7 Outputs

DPCV generates a log file for each run (See Section 4.7.14.10 for details). Besides the log file generated for each run, DPCV will also produce a phantom report and a checksum mismatch report if necessary under directory that is configured as VALIDATION_OUTPUT_DIR in the configuration file. The phantom report lists the DPL granule ids of the phantom granules found. The checksum mismatch lists the DPL granule ids of the granules that have checksum mismatch failures. The naming convention for the phantom report is: Phantom_dplids_RepairByRestoreOlaFromTape.<pid>.<timestamp>. The naming convention for the checksum mismatch report is: ChecksumMismatch_dplids_RepairByRestoreOlaFromTape.<pid>.<timestamp>.

The generated reports can be used as input to the EcDIRestoreOlaFromTape utility to restore the granules that are identified. The syntax to invoke the EcDIRestoreOlaFromTape utility is: EcDIRestoreOlaFromTapeStart <MODE> -file <absolute_DPCV_report_file_name> -contents dplids. Please refer to the EcDIRestoreOlaFromTape utility 609 document for details.

If DPCV runs successfully and no errors or mismatches are identified, it will exit with an exit code of 0; If DPCV run encountered an internal error, the exit code will be set to 1; If DPCV run completes successfully and identified some files with mismatch errors, the exit code will be set to 2; If DPCV run completes successfully and identified no file mismatch errors but some phantom files, the exit code will be set to 3.

4.7.14.8 Event and Error Messages

Usage errors will be displayed to the terminal screen. Processing error messages are written to the log file.

4.7.14.9 Reports

See outputs above.

4.7.14.10 Logs

Since multiple instances of DPCV can run at the same time, the utility produces a log file called EcDIDpcv.log.<pid> for each DPCV run. The log file name will be displayed on the terminal after the run is started. The log file will reside in the /usr/ecs/<mode>/CUSTOM/logs directory. This way, each DPCV run will have its own log file which makes it easy to trace and debug.

Besides the log file, a record is logged in the DIDpcvHistory table in the database for each DPCV run. It has the pid of the DPCV process, the parameters of the run, the hostname on which the DPCV is running, the start and end time of the run, the status and the statistics of the current progress.

4.7.14.11 Recovery

The DataPool Checksum Verification utility does not provide recovery for previous abnormally terminated runs, but starting the DPCV with identical parameters as the previous run will in fact perform the recovery. The recovery will be most efficient if the utility is run with the daysSinceLastChecksum parameter specified.

4.7.14.12 Sybase Error Handling

If a Sybase error occurs, the actual Sybase error string will most likely be logged in the log. Possible errors include that the database server is unavailable, that the connection to the database was dropped, or that there was an error executing a stored procedure. In the event of a Sybase-sourced error, the utility will not be able to process any granules.

In the event that a connection to the Data Pool database cannot be established, the utility may repeatedly attempt to connect to the database, depending on how the configuration file was set. If, for example, NUM_RETRIES was set to 3 and RETRY_INTERVAL was set to 10, the utility will try to connect to the database 3 times, and will wait 10 seconds between each attempt – a total of 30 seconds if all attempts are unsuccessful.

4.7.15 Inventory Validation Tool

The Inventory Validation Tool provides the EEB Operations Staff with a command-line interface to identify the discrepancies between the AIM and DPL databases.

4.7.15.1 Using the Inventory Validation Tool

The Inventory Validation Tool is started by entering the following command from the /usr/ecs/<mode>/CUSTOM/utilities directory:

> **EcDIInventoryValidationTool.pl <command line parameters>**

There are various command line parameters that are used in combination with each other. Table 4.7.15-1 provides a description of these parameters.

Table 4.7.15-1. Command Line Parameters of the Inventory Validation Tool

Parameter Name	Description
<mode>	Mandatory. Specifies the mode of operation. This must be the first parameter passed, and it must be a valid, existing Data Pool mode with a format of OPS or TS[1-4] or DEV0[1-9].
-outputDir	Optional. Specifies the relative path under the base directory defined under parameter VALIDATION_OUTPUT_DIR in the configuration file EcDIInventoryValidationTool.CFG. Note: the base directory has to exist; The relative directory (only one level down) will be created if it doesn't exist. This is where all the output files reside. If the relative path is not provided, the output files will go to the base directory.
-suppressLDeleted	Optional. When identifying granules that are missing in the DPL database, don't include the ones that have been logically deleted (deleteEffectiveDate is not null) in the AIM database.
-suppressDFAed	Optional. When identifying granules that are missing in the DPL database, don't include the ones that have been DFAed (DeleteFromArchive = "Y") in the AIM database.

There is no required ordered sequence of the parameters except for the <mode> which must be the first parameter. A command line input error results in a 'usage' display. The reason why the input was incorrect is also displayed.

4.7.15.2 Inventory Validation Tool Commands

Below is an example for invoking this tool:

```
1. EcDIInventoryValidationTool.pl DEV04 -outputDir inventory  
-suppressLDeleted
```

This checks for discrepancies between the DPL and AIM databases. Output files will be written to an inventory subdirectory under the VALIDATION_OUTPUT_DIR directory. The output result will not include any granules that are logically deleted in AIM and missing in DPL.

4.7.15.3 Required Operating Environment

The Online Archive Validation Tool will run on the same box as EcDICleanupFilesOnDisk.pl.

4.7.15.4 Interfaces and Data Types

Table 4.7.15-2 lists the supporting products that this tool depends upon in order to function properly.

Table 4.7.15-2. Interface Protocols

Product Dependency	Protocols Used	Comments
Data Pool database	SQL	Via SQL server machines

4.7.15.5 Configuration File Format – EcDIInventoryValidationTool.CFG

The configuration file contains vital details about how to connect to the Sybase server. Without this file, the tool can not run. Sample configuration file is listed below.

Parameter Name	Description
SYB_USER	The user name for the Sybase connection.
SYB_SQL_SERVER	The name of the Sybase SQL server.
PGM_ID	Program ID used for connecting to the Sybase database.
NUM_RETRIES	The number of times that the utility attempts to connect to the database before exiting.
SLEEP_SEC	The number of seconds the utility waits between database connection attempts.
ROWCOUNT	Used to restrict the number of rows returned when running the Sybase query to retrieve discrepancies regarding granules which exist in the AIM database but not in the DPL database. This parameter may be used to prevent an out-of-memory error where there are large numbers of discrepancies between the AIM and DPL databases – e.g., before migration is complete. When set to 0, the rowcount is not restricted.
SKIP_MISSINGDPL	Set to Y or N. Allows the user to skip the checking that identifies granules which exist in AIM but not in DPL.
VALIDATION_OUTPUT_DIR	The base directory where output files from the utility are written. The recommended value is /workingdata/emd/<mode>/lvu

4.7.15.6 Special Constraints

The Inventory Validation tool runs only if the Data Pool and AIM databases are available.

4.7.15.7 Outputs

Output files are created under the base directory defined in the configuration file under VALIDATION_OUTPUT_DIR if the -outputdir parameter is not provided on the command line. Otherwise, the output files will be created in the base directory/<outputdir>.

There are 5 output files generated by the Inventory Validation utility.

The names are:

- InventoryDiscrp_dplids_RepairByDPLCleanup.<pid>.<yyyymmddhhmmss>: granules that are in the Data Pool database but not in the AIM database.
- InventoryDiscrp_ecsids_RepairByPublish.<pid>.<yyyymmddhhmmss>: granules that are in the hidden Data Pool but are in public collections and are eligible to be public, as well as granules which are in the AIM database but not in the Data Pool database.
- InventoryDiscrp_dplids_RepairByUnpublish.<pid>.<yyyymmddhhmmss>: granules that are in the public Data Pool but should be in the hidden Data Pool.
- InventoryCksDiscrepancy_ecsids.<pid>.<yyyymmddhhmmss>: granules whose checksum value in the AIM database is different than the checksum value in the DPL database.
- InventoryCksTypeDiscrepancy_ecsids.<pid>.<yyyymmddhhmmss>: granules whose checksum type in the AIM database is different than the checksum type in the DPL database.

Note for replacement granules: IVT was modified to take granule replacement/collision into account when identifying granules in hidden datapool that need to be published. Now the candidate granule can only make it to the Inventorydiscrp_ecsids-RepairByPublish file if there doesn't exist any granule in the public datapool **with which the granule would collide (replacementOn = N) or for which the currently public granule is a more recent replacement (replacementON = Y)**. This is intended to **prevent predictable publishing failures**.

However, publishing failures **can still occur** when there are several granule versions in the datapool of which none is public. **This can occur, for example**, if the public **version** somehow got deleted, or if previously replacementOn was set to "N" and **recently changed** to "Y". In these cases, **all** versions of the hidden granules are **considered** eligible to be published. Since they're replacements to each other some might fail (depending on the sequence of the publishing **operations**). Once the latest version is published, the remaining hidden versions will no longer **be considered for publishing and subsequent runs will not include them** in their output file.

Note for multifile granules: The utility cannot reliably match up a file in Data Pool and a file in AIM that belong to the same multifile granule. So, when reporting checksum/checksumtype discrepancies for multifile granules, the utility may log the following messages:

The following record(s) have mismatch checksum values

- dpl info (dplId,ecsId,filename,checksumValue,checksumType) are:
 - <list of Data Pool files for the granule for which there is no corresponding AIM file with the same checksum type and same checksum value>

- aim info (ecsId, filename, checksumValue, checksumType) are:
 - <list of AIM files for the granule for which there is no corresponding Data Pool file with the same checksum type and same checksum value>

DAAC personnel must investigate the situation and decide which pairs of files are matching pairs.

4.7.15.8 Event and Error Messages

Errors will be displayed to the screen as well as logged in the log file.

4.7.15.9 Logs

The tool logs messages in the /usr/ecs/<mode>/CUSTOM/logs/EcDIInventoryValidationTool.log file.

4.7.15.10 Recovery

If the Inventory Validation Tool is interrupted by a fault, when the utility is restarted, it will just rerun everything and produce a new set of output files.

4.7.16 EcDILinkCheck.ksh

Deleted. This section is a duplicate of 4.8.18. Please refer to section 4.8.18 for details.

This page intentionally left blank.

4.7.17 Publish Utility

The DPL publish Utility is a command line tool that publishes specified granules from a file, command line or collection. It is primarily designed to publish granules that already exist in the Data Pool, but it can also be used to insert granules into the Data Pool from AIM. Note that the Publish Utility does not perform the insert and/or publication actions directly; instead, it submits requests to the Data Pool Action Driver to perform the work on its behalf.

4.7.17.1 Using the Data Pool Publish Utility

The Data Pool Publish Utility is started via the following script, from the /usr/ecs/<mode>/CUSTOM/utilities directory:

```
EcDIPublishUtilityStart <MODE> -ecs [-file <file_name_path> | -collection
<ShortName.VersionId> | -g <ecsId1>[ ][,<ecsId2> ] [-theme <themeName>] [-batchlabel
<batchLabel>] [-maxnumconactions <num>] [-register] [-publish] or $0 <MODE> -nonecs -file
<file_name_path> [-theme <themeName>] [-batchlabel <batchLabel>] [-maxnumconactions
<num>]
```

Table 4.7.17-1 Provides a description of these parameters.

Table 4.7.17-1. Data Pool Publish Utility Command Line Parameters (1 of 2)

Parameter Name	Description
-ecs	Specifies all the granules to be published are ECS granules
-file <file_name_path>	Tells the publish utility to read the list of ECS ids of granules to be published from a file. <i>input_file</i> specifies the full path of the file. Or if nonecs option is specified, it contains a list of xml files which specify the nonecs data to be published.
-g <id1>,<id2>...	Specifies the ECS ids of the granules to publish on the command line. Any number of granules may be provided (within any limits the shell places on command line length).
-collection <Shortname.Versionid>	Tells the Publish Utility to publish all granules belonging to a given collection.
-theme <themeName>	Specifies the theme name associated with the granules to be published.
-batchlabel <batchLabel>	Specifies the batch label associated with the granules to be published.
-nonecs	Specifies all the granules to be published are NonECS granules.
-maxnumconactions	Indicates the number of concurrent actions that may be submitted to the Data Pool Action Driver. This option can be use to limit the impact on existing operations. If not provided, it defaults to 5,000, which effectively assumes that it has exclusive use of the Action Driver.

Table 4.7.17-1. Data Pool Publish Utility Command Line Parameters (2 of 2)

Parameter Name	Description
-register	Indicates that the Publish Utility should make sure that the given granules exist in the Data Pool. Any granule that does not exist in the Data Pool will be inserted (registered). No granules will be published (placed into the public Data Pool). Granules may be inserted into the Data Pool even if they are logically deleted, or marked as hidden (i.e. DeleteFromArchive = 'H'). Granules will not be inserted if it is marked as deleted from archive (DeleteFromArchive = 'Y').
-publish	Indicates that the Publish Utility publish the given granules in the Data Pool. Only granules that already exist in the Data Pool will be published. Any granule that does not exist in the Data Pool will not be inserted. Granules that belong to a collection that is marked as not public (GranPublicFlag='N'), or are logically deleted or hidden, will not be published. Note also, that older versions of a granule will not replace a newer version.

Note that if neither `-register`, not `-publish` is provided, the default behavior is to register, then publish.

4.7.17.2 Data Pool Publish Utility usage examples

1. *EcDIPublishStart OPS -ecs -file /home/cmshared/granuleIds.txt*

Insert and publish granules for the granule ids contained in the specified file. The file contains one ECS granule id per line.

2. *EcDIPublishStart OPS -ecs -g 12345, 23456 -publish*

Publish the two hidden granules whose ECS ids are given.

3. *EcDIPublishStart OPS -ecs -collection MYD29P1D.004 -maxnumconactions 10*

Make sure all granules belonging to collection MYD29P1D version 4 are public in the Data Pool, limiting the number of concurrent Action Driver requests to 20. This is a low impact way to make sure a complete collection is public, but could take days to run to completion.

4. *EcDIPublishStart OPS -ecs -g 12345 -theme "test"*

Publish 1 ECS granule and establish the theme "test" to the granule.

5. *EcDIPublishStart OPS -nonecs -file /home/cmshared/nonecs_xml.txt*

Publish nonecs granules which are specified in a list of xml files in "nonecs_xml.txt".

4.7.17.3 Required Operating Environment

The Publish Utility will run on a LINUX platform. It shall be installed on the DPL platform as part of the New Data Pool Insert Utility installation.

4.7.17.4 Interfaces and Data Types

Table 4.7.17-2 lists the supporting products that this tool depends upon in order to function properly.

Table 4.7.17-2. Interface Protocols

Product Dependency	Protocols Used	Comments
Data Pool and AIM databases	SQL	Via SQL server machines
Sybase JDBC driver	JDBC	Requires proper install of jConnect (the Sybase JDBC driver).
StoreNext client	Proprietary	Exposes the DPL file system on the DPL platform.

4.7.17.5 Input File Format

One granuleId per line for ECS data, one xml file name per line for NonECS data.

4.7.17.6 Configuration File

No special configuration file is needed to run the utility. It uses the same configuration file as the Data Pool Insert Utility (DPIU) and the New Data Pool Insert Utility (NDPIU), namely EcDIInsertUtility.properties.

4.7.17.7 Special Constraints

The mode specific database needs to be up and running and the installation platform need to have access to the Data Pool Storage Area Network.

4.7.17.8 Outputs

The output of pertinent events is recorded in the /usr/ecs/<mode>logs/EcDIPublish.log.PID log file.

4.7.17.9 Event and Error Messages

Usage errors will be displayed to the screen. Processing error messages are written to the log files.

4.7.17.10 Reports

None

4.7.17.11 Logs

The utility produces log files in the standard log file location. The log file name is EcDIPublish.log.PID. The verbosity of the log file is controlled by the DEBUG_MESSAGES entry in the EcDIInsertUtility configuration file.

4.7.17.12 Recovery

No recovery mechanism is required for this utility. In the event of an interrupted run, the run may be invoked again with the same command-line parameters. Any granules already processed will be detected and not processed again.

4.7.17.13 Database Error Handling

If a database error occurs, the specific error details will be logged. Some database errors are retried internally (i.e. deadlocks), others will cause processing of the current granule to fail and the utility to start work on the next granule in the list.

4.7.18 Unpublish Utility

The DPL Unpublish Utility is a command line tool that unpublishes specified granules from the Data Pool. Granules may be specified in a file, or by command line.

The Unpublish utility was developed for the on-line archive capability. It will:

- unpublish the specified science granules and associated QA/PH granule links if there are any
- unpublish QA/PH granules and remove links associated to the corresponding science granules if there are any
- remove associated browse granule if permitted

The Unpublish utility can also be used to unpublish granules which are marked for deletion in the AIM database (deleteEffectiveDate is set, or DFA flag is set to “Y” or “H”), for example, as would occur after a run of the Granule Deletion Utility.

4.7.18.1 Using the Data Pool Unpublish Utility

The Data Pool Unpublish Utility is started via the following script, from the /usr/ecs/<mode>/CUSTOM/utilities directory:

```
EcDIUnpublishStart.pl -mode <mode> [-file <input_file>] [-granules <id1>,<id2>...] [-aim -offset <#days>]
```

EcDIUnpublishStart.pl -help for instructions.

Table 4.7.18-1 provides a description of these parameters.

Table 4.7.18-1. Data Pool Unpublish Utility Command Line Parameters

Parameter Name	Description
-file <input_file>	The file which contains a list of DPL granule ids for unpublish. Input_file specifies the full path and file name of the file.
-granules <id1>, <id2> ...	DPL granule ids for unpublish.
-aim	Tells the unpublish utility to unpublish granules deleted from the AIM database. If this option is used, the -offset option should also be provided.
-offset <#days>	Specifies the past number of days for which to find deleted AIM granules. This option is only valid in conjunction with the -aim option.
-help	Display instructions to run the utility.

An incorrect command line will result in a ‘usage’ syntax display. The log file for the utility is /usr/ecs/<mode>/CUSTOM/logs/EcDIUnpublish.log.PID.

4.7.18.2 Data Pool Unpublish Utility usage examples

1. *EcDIUnpublishStart.pl -mode OPS -file /home/cmshared/granuleIds.txt*

Unpublish public granules for the granuleIds contained in the specified file. The file contains one Data Pool granuleId per line.

2. *EcDIUnpublishStart.pl -mode OPS -granules 12345, 23456*

Unpublish public granules for the granuleIds specified in the command line, separated by “,”.

3. *EcDIUnpublishStart.pl -mode OPS -aim -offset 12*

Unpublish granules deleted from the AIM database since the current time – 12 days.

4.7.18.3 Required Operating Environment

The Unpublish Utility will run on a LINUX platform. It shall be installed on the DPL platform as part of the New Data Pool Insert Utility installation.

4.7.18.4 Interfaces and Data Types

Table 4.7.18-2 lists the supporting products that this tool depends upon in order to function properly.

Table 4.7.18-2. Interface Protocols

Product Dependency	Protocols Used	Comments
Data Pool database	SQL	Java JDBC invocation of Stored Procedures.
StoreNext client	Proprietary	Exposes the DPL file system on the DPL platform.

4.7.18.5 Input File Format

One granuleId per line.

4.7.18.6 Configuration File

No special configuration file is needed to run the utility. It uses the same configuration file as the Data Pool Insert Utility (DPIU) and the New Data Pool Insert Utility (NDPIU), namely EcDIInsertUtility.properties.

4.7.18.7 Special Constraints

The mode specific database needs to be up and running and the installation platform need to have access to the Data Pool Storage Area Network.

4.7.18.8 Outputs

The output of pertinent events is recorded in the /usr/ecs/<mode>logs/EcDIUnpublish.log.PID log file.

4.7.18.9 Event and Error Messages

Usage errors will be displayed to the screen. Processing error messages are written to the log files.

4.7.18.10 Reports

None

4.7.18.11 Logs

The utility produces log files in the standard log file location. The log file name is EcDIUnpublish.log.PID. The verbosity of the log file is controlled by the DEBUG_MESSAGES entry in the EcDIInsertUtility configuration file.

4.7.18.12 Recovery

No recovery mechanism is required for this utility. In the event of an interrupted run, the run may be invoked again with the same command-line parameters. Any granules already processed will be detected and not processed again.

4.7.18.13 Database Error Handling

If a database error occurs, the specific error details will be logged. Some database errors are retried internally (i.e. deadlocks), others will cause processing of the current granule to fail and the utility to start work on the next granule in the list.

This page intentionally left blank.

4.7.19 Archive Checksum Validation Utility

The Archive Checksum Validation utility (ACVU) provides a mechanism by which the ECS Operations Staff can perform checksum verification of files in the AIM archive. The utility allows the operator to specify which files to verify, by sampling files based on media ID (a single media ID or a list of media IDs), volume group (a single volume group or a list of volume groups), or granule ID (a single granule ID, a list of granule IDs, or an input file containing granule IDs). The operator may also restrict verification to files which have not had their checksum verified within an operator-specified time period.

According to the sampling criteria specified, the utility will identify the files to be verified, organize the result by location on tape, verify their checksum values, and update the last checksum verification time and status in the AIM Inventory database. The utility will need to verify that an LTO tape is in the near-line archive (i.e. not off-line) and alert the operator if the tape is off-line.

Upon detection of checksum verification failure after a configurable number of retry attempts (NUM_CHECKSUM_RETRIES in configuration file), the utility will log detailed information about the failure including media ID, volume group, granule ID, ESDT, insert time, complete file path and file name, along with the checksum information -- including checksum type, checksum values (computed value vs the corresponding value stored in database), the last time the file was checksummed, and checksum origin (who performed the last checksum). The checksum status of the file will be updated in the AIM Inventory database to mark it as a case of checksum verification failure.

The log will also include statistical summary information including total number of files checked, number of files that failed checksum, percentage of files that failed checksum, categorized by ESDT. This utility is designed such that the checksum verification can be throttled (by adjusting the number of concurrent tapes and number of concurrent tape reads) so it does not impact on-going daily operations.

4.7.19.1 Using the Archive Checksum Validation Utility

The Archive Checksum Validation utility should be started by the user cmshared (or similar). The Archive Checksum Validation utility is started by entering the following command:

```
EcDsAmAcvu.pl <mode> <command line parameters>
```

There are eight command line parameters that may be used. Table 4.7.19-1 provides a description of those parameters.

```
EcDsAmAcvu.pl <MODE> [-calculate]
                    [-days <NUMBER OF DAYS>]
                    [-percent <PERCENT 1-100>]
                    [-norecovery]
                    (-volumegroup <VOLUME GROUPS> |
                    -mediaid <MEDIAIDS> |
```

```

-granuleid <GRANULEIDS> |
-file <FILENAME>)
[-outputDir <DIRECTORY>]

```

Table 4.7.19-1. Command Line Parameter

Parameter Name	Required	Description
calculate	No	Optional parameter to specify whether to calculate and store checksums for files found currently without checksums.
days	No	Optional parameter to specify days since last checked.
percent	No	Optional parameter to specify percentage of files to check.
norecovery	No	Optional parameter to specify not to recover from previous run.
volumeGroup	Yes, if mediaid, granuleid, or file parameters are not present	Parameter to specify volume groups whose files will have their checksum verified. This is a comma separated list of one or more volume groups (no spaces). Volume groups should be specified by full path name.
mediaid	Yes, if volumeGroup, granuleid, or file parameters are not present	Parameter to specify mediaids whose files will have their checksum verified. This is a comma separated list of one or more mediaids (no spaces).
granuleid	Yes, if volumeGroup, mediaid, or file parameters are not present	Parameter to specify granules whose files will have their checksum verified. This is a comma separated list of one or more granule ids (no spaces).
file	Yes, if volumeGroup, mediaid, or granuleid parameters are not present	Parameter to specify the name of an input file containing granuleids of granules whose files will have their checksum verified. Granuleids should be listed in the input file separated by newlines.
outputDir	No	Parameter to specify directory for error files under /workingdata/emd/<MODE>/Acvu

4.7.19.1.1 Archive Checksum Validation Utility Command Line Examples

1. For a "volumeGroup" run:

```
EcDsAmAcvu.pl OPS -volumeGroup /stornext/snfs1/OPS/MODIS
```

The Archive Checksum Validation Utility will perform checksumming for all files in specified volumeGroup (/stornext/snfs1/OPS/MODIS).

```
EcDsAmAcvu.pl OPS -volumegroup
```

```
  /stornext/snfs1/OPS/MODIS,/stornext/snfs1/OPS/ASTER -percent 50
```

The Archive Checksum Validation Utility will perform checksumming for 50% of the files in the specified volume groups.

2. For a "media id" run:

```
EcDsAmAcvu.pl OPS -mediaid VG7029
```

The Archive Checksum Validation Utility will perform checksumming for all files on the specified tape.

```
EcDsAmAcvu.pl OPS -mediaid VG7029,TG8024 -days 10
```

The Archive Checksum Validation Utility will perform checksumming for the files on the specified tapes which have not been verified in the past 10 days.

3. For a "granuleid" run:

```
EcDsAmAcvu.pl OPS -granuleid 22083,22085,22087
```

The Archive Checksum Validation Utility will perform checksumming for the files related to granules 22083, 22085, and 22087 in OPS mode.

```
EcDsAmAcvu.pl OPS -granuleid 22083,22085,22087 -calculate
```

The Archive Checksum Validation Utility will perform checksumming for the files related to granules 22083, 22085, and 22087 in OPS mode and if the files do not have a checksum, one will be calculated for it.

4. For a "file" run:

```
EcDsAmAcvu.pl OPS -file granuleids.txt
```

The Archive Checksum Validation Utility will perform checksumming for the files related to granules specified in granuleids.txt.

```
EcDsAmAcvu.pl OPS -file granuleids.txt -norecovery
```

The Archive Checksum Validation Utility will ignore recovery for any previous run and perform checksumming for the files related to granules specified in granuleids.txt.

4.7.19.2 Archive Checksum Validation Utility Configuration File

The Archive Checksum Validation utility uses a configuration file, EcDsAmAcvu.CFG, located in /usr/ecs/<mode>/CUSTOM/cfg directory. The configuration parameters are stored in a PARAMETER = VALUE format with each parameter/value pair as a separate line entry in the file. Table 4.7.19-2 describes the configuration parameters.

Table 4.7.19-2. Configuration Parameters

Parameter Name	Value Description
SYB_USER	Sybase login name for the user of the Inventory database.
SYB_SQL_SERVER	Name of Sybase SQL Server hosting Inventory database.
SYB_DBNAME	Name of Inventory database.
PGM_ID	Program identifier used as seed to generate database password.
NUM_RETRIES	Number of times database operation will be attempted.
RETRY_INTERVAL	Number of seconds between retries.
SNSM_HOST	The Stornext host
SNSM_PORT	The Stornext port
SNSM_TEMP_DIR	The directory to place file listings for tapes. This directory should be cross mounted between the Stornext host and the oml host. The suggested directory is /workingdata/emd/<MODE>/Acvu/TempDir The directory should be readable by cmshared with write permissions for the Stornext user(smuser). To achieve this we suggest having the directory owned by smuser, a groupid of cmshared, and 775 permissions. This directory should be cleaned up manually.
MAX_BLOCKINFO_PROCESSES	Number of processes to get block info from media concurrently
MAX_TAPE_READS	Number of read requests per tape at once
MAX_CONCUR_TAPES	Number of tapes that can be read from at once
NUM_CHECKSUM_RETRIES	Number of times a checksum will be attempted.
VALIDATION_OUTPUT_DIR	The default directory to place error output files. The directory should be readable/writeable by cmshared. The suggested directory is /workingdata/emd/<MODE>/Acvu

4.7.19.3 Archive Checksum Validation Utility Main Screen

The Archive Checksum Validation Utility does not have a main screen. It has a command line interface only.

4.7.19.4 Required Operating Environment

The Archive Checksum Validation Utility will run on a Linux platform.

4.7.19.5 Databases

Table 4.7.19-3 lists the supporting products that this tool depends upon in order to function properly.

Table 4.7.19-3. Product Dependencies

Product Dependency	Protocols Used	Comments
Inventory Database	SQL	Via SQL server machines

4.7.19.6 Special Constraints

The Archive Checksum Validation runs only if the Inventory database server is running and if the database is available. It also assumes the stored procedures are present and its temporary database table has been created.

4.7.19.7 Outputs

Output of update events and errors will be always appended to a single log file. The Acvu will also produce a failed file (AIMChecksumMismatch_ecsids_RepairByRestoreTapeFromOla.<pid>.<date>). It will be placed in /workingdata/emd/<MODE>/Acvu. This directory may be further extended using the –outputDir command line option.

4.7.19.8 Event and Error Messages

Usage errors will be displayed to the terminal screen. Processing error messages are written to the log files.

4.7.19.9 Reports

None

4.7.19.10 Logs

The utility produces a log file called EcDsAmAcvu.log in the /usr/ecs/<mode>/CUSTOM/logs directory. If this log file already exists, the new information will automatically be appended. If there is no existing log file by this name, a new log file with this name will automatically be created.

Since the log file may grow to a considerable size after constant use, it is recommended that it be saved off into a separate file from time to time for maintainability.

4.7.19.11 Recovery

The Archive Checksum Validation Utility provides a capability to recover from interruptions caused by situations such as system faults or database errors leaving all or some of the files not checksummed. The utility will detect such failure upon the next run and continue processing the directories and files that were left unprocessed in the previous run. The operator can ignore recovery by using the –norecovery option. Recovery will only be needed if the utility was interrupted after it started checksumming files.

4.7.19.12 Sybase Error Handling

If a Sybase error occurs, the actual Sybase error string will most likely be displayed on the screen and in the log. Possible errors include that the database server is unavailable, that the connection

to the database was dropped, or that there was an error executing a stored procedure. In the event of a Sybase-sourced error, the utility will immediately stop running.

In the event that a connection to the Data Pool database cannot be established, the utility may repeatedly attempt to connect to the database, depending on how the configuration file was set. If, for example, NUM_RETRIES was set to 3 and RETRY_INTERVAL was set to 10, the utility will try to connect to the database 3 times, and will wait 10 seconds between each attempt – a total of 30 seconds if all attempts are unsuccessful.

4.7.20 XML Check Utility

The XML Check utility provides a mechanism by which the ECS Operations Staff can periodically check for corruption in the XML Archive.

In order to detect corruption, the utility verifies the contents of the files are well formed using xmllint.

4.7.20.1 Using the XML Check Utility

The XML Check utility should be started by the user cmshared (or similar). The XML Check utility is started by entering the following command:

```
EcDsAmXcu.pl <mode> <command line parameters>
```

There are seven command line parameters that may be used. Table 4.7.20-1 provides a description of those parameters.

Table 4.7.20-1. Command Line Parameter

Parameter Name	Required	Description
days	No	Optional parameter to specify days since last checked.
percent	No	Optional parameter to specify percentage of files to check
ESDT	Yes, if granuleid or file parameters are not present	Parameter to specify which ESDTs to check. This is a comma separated list (no spaces). Can also specify "ALL" to include all ESDTs.
startdate	No	Optional parameter used with –ESDT option. Specifies starting insert date to use for ESDTs.
enddate	No	Optional parameter used with –ESDT option. Specifies ending insert date to use for ESDTs.
granuleid	Yes, if ESDT or file parameters are not present	Parameter to specify which granules to check. This is a comma separated list (no spaces).
file	Yes, if ESDT or granuleid parameters are not present	Parameter to specify which granules to check. Granule ids should be listing in a file separated by newlines.
outputDir	No	Parameter to specify directory for error files under /workingdata/emd/<MODE>/Xcu

4.7.20.1.1 XML Check Utility Command Line Examples

1. For an "ESDT" run:

```
EcDsAmXcu.pl OPS -ESDT ALL
```

The XML Check Utility will perform checking for all xml files in OPS mode

```
EcDsAmXcu.pl OPS -ESDT AST_L1A.003,MOD29.005 -startdate Jan 20 2008 -  
enddate Dec 1 2008
```

The XML Check Utility will performed checking for all AST_L1A.003 and MOD29.005 xml files whose granules have been inserted between Jan 20 2008 and December 1 2008.

```
EcDsAmXcu.pl OPS -ESDT AST_L1B.003 -percent 50 -days 10
```

The XML Check Utility will perform checking for 50% of AST_L1B.003 granules which have not been checked in the last 10 days.

2. For a "granuleid" run:

```
EcDsAmXcu.pl OPS -granuleid 22083,22085,22087
```

The XML Check Utility will perform checking for the xml files related to granules 22083, 22085, and 22087in OPS mode

3. For a "file" run:

```
EcDsAmXcu.pl OPS -file granuleids.txt
```

The XML Check Utility will perform checking for the xml files related to granules specified in granuleids.txt

4.7.20.2 XML Check Configuration File

The XML Check utility uses a configuration file, EcDsAmXcu.CFG, located in /usr/ecs/<mode>/CUSTOM/cfg directory. The configuration parameters are stored in a PARAMETER = VALUE format with each parameter/value pair as a separate line entry in the file. Table 4.7.20-2 describes the configuration parameters.

Table 4.7.20-2. Configuration Parameters (1 of 2)

Parameter Name	Value Description
SYB_USER	Sybase login name for the user of the Inventory database.
SYB_SQL_SERVER	Name of Sybase SQL Server hosting Inventory database.
SYB_DBNAME	Name of Inventory database.
PGM_ID	Program identifier used as seed to generate database password.
NUM_RETRIES	Number of times database operation will be attempted.

Table 4.7.20-2. Configuration Parameters (2 of 2)

Parameter Name	Value Description
RETRY_INTERVAL	Number of seconds between retries.
MAX_CONCUR_CHECKS	Number of concurrent calls to xmllint that will be allowed.
VALIDATION_OUTPUT_DIR	The default directory to place error output files. The directory should be readable/writeable by cmshared. The suggested directory is /workingdata/emd/<MODE>/Xcu

4.7.20.3 XML Check Utility Main Screen

The XML Check Utility does not have a main screen. It has a command line interface only.

4.7.20.4 Required Operating Environment

The XML Check Utility will run on a Linux platform.

4.7.20.5 Databases

Table 4.7.20-3 lists the supporting products that this tool depends upon in order to function properly.

Table 4.7.20-3. Product Dependencies

Product Dependency	Protocols Used	Comments
Inventory database	SQL	Via SQL server machines

4.7.20.6 Special Constraints

The XML Check utility runs only if the Inventory database server is running and if the database is available. It also assumes the stored procedures are present.

4.7.20.7 Outputs

Output of update events and errors will be always appended to a single log file. The Xcu will also produce a phantom file(AIMPhantomXMLs_<ecsids>.<pid>.<date>) and a failed file(AIMFailedXMLCheck_<ecsids>.<pid>.<date>). They will be placed in /workingdata/emd/<MODE>/Xcu. This directory may be further extended using the -outputDir command line option.

4.7.20.8 Event and Error Messages

Usage errors will be displayed to the terminal screen. Processing error messages are written to the log files.

4.7.20.9 Reports

None

4.7.20.10 Logs

The utility produces a log file called EcDsAmXcu.log in the /usr/ecs/<mode>/CUSTOM/logs directory. If this log file already exists, the new information will automatically be appended. If there is no existing log file by this name, a new log file with this name will automatically be created.

Since the log file may grow to a considerable size after constant use, it is recommended that it be saved off into a separate file from time to time for maintainability.

4.7.20.11 Recovery

The XML Check Utility can recover from interruptions caused by situations such as the system faults or database errors leaving all or some of the xml files unchecked. To recover, the operator would need to specify the `-days` parameter and enter the number of days since the last time the utility was run. This will ensure xml files which have already been checked are not rechecked.

4.7.20.12 Sybase Error Handling

If a Sybase error occurs, the actual Sybase error string will most likely be displayed on the screen and in the log. Possible errors include that the database server is unavailable, that the connection to the database was dropped, or that there was an error executing a stored procedure. In the event of a Sybase-sourced error, the utility will immediately stop running.

In the event that a connection to the Inventory database cannot be established, the utility may repeatedly attempt to connect to the database, depending on how the configuration file was set. If, for example, `NUM_RETRIES` was set to 3 and `RETRY_INTERVAL` was set to 10, the utility will try to connect to the database 3 times, and will wait 10 seconds between each attempt – a total of 30 seconds if all attempts are unsuccessful.

4.7.21 RestoreOlaFromTape

The *RestoreOlaFromTape* utility will repair individual granules or files that are lost or damaged in the on-line archive provided that the inventory entries of the corresponding granules are completely intact. This is because *RestoreOlaFromTape* does not have the capability to repair Data Pool inventory database entries. In all other cases, granules must be restored using the Publish utility (e.g., if file entries or browse cross references are missing, or Data Warehouse entries for public granules were damaged or lost). The publish utility has the capability to reconstruct the Data Pool inventory entries for a granule.

The *RestoreOlaFromTape* utility shall:

- Restore defective granules from their tape archive location.
- Verify the checksum of the tape copy.
- Rename the files according to Data Pool rules.
- Restore granule metadata files from the XML file archive.
- Restore browse, QA and PH symbolic links for the science granule that are restored.
- Restore browse granules or files from the browse file archive, which is a disk archive. If the corrupted or lost browse files belong to a public browse granule, the corresponding browse images will be extracted from the original browse file.

In addition, the *RestoreOlaFromTape* utility shall:

- Optimize the restore of the files from the tape archive by organizing them by tape. Within a collection of files from the same tape, files will be organized in ascending block number order. This organization will optimize the tape read operations.
- Allow configurable parallelization of the tape restore operations by providing a configuration parameter that specifies the number of tape drives to be used for the restore operation. Please note that for a given tape, no concurrent/parallel access will be provided.
- Manage the capacity demand of bulk repairs to avoid serious degradation of operational workloads (e.g., limits on concurrent tape mounts, tape reads, on-line archive writes, checksumming operations).

Input is provided via an input file.

4.7.21.1 Running the RestoreOlaFromTape Utility

The following command line syntax must be used to start the RestoreOlaFromTape Utility:

```
> EcDIRestoreOlaFromTapeStart <mode> -file <file name and path with contents specified by –contents parameter> -contents <dplids | ecsids | dplfiles> [-restoremisbr] [-restorelinks [only]] [-recovery [no]] [-email <usertoreceivestatusemail>]
```

Table 4.7.21-1 provides a description of the above command line parameters.

Table 4.7.21-1. RestoreOlaFromTape Utility Parameters

Parameter Name	Description
<i>-file <file name></i>	Name and path of the input file to be used by the utility
<i>-contents <contents type></i>	The type of contents present in the file. Any of the following options are allowed: <ul style="list-style-type: none"> ⇒ dplids: the input file contains the DPL granule IDs or browse IDs of the on-line archive granules that must be repaired ⇒ ecsids: the input file contains the ECS granule IDs (dbIDs) or browse IDs of the on-line archive granules that must be repaired ⇒ dplfiles: the input file contains the DPL filenames of the files that must be repaired. Browse files in JPG or HDF format are also accepted
<i>[-restoremisbr]</i>	Indicates if the utility should restore MISBR browse granule in the DPL. If the parameter is not set, the MISBR browse granule will not be restored. NOTE: This parameter will cause the utility to MISBR browse granule only when the configuration parameter MISR_SPECIAL_PROCESSING is set to "Y".
<i>[-restorelinks [only]]</i>	Indicates if the utility should restore browse/QA/PH symbolic linkage file for the given science granule. When '-restorelinks' is not provided in command line, only science granule metadata and data files are restored. When '-restorelinks' is specified in command line, both science granule files and browse/QA/PH symbolic links are restored. When '-restorelinks only' is specified in command line, only browse/QA/PH symbolic links are restored. Note: A list of science granule DPL Ids or ECS Ids should be used to restore browse/QA/PH symbolic links.
<i>[-recovery [no]]</i>	Indicates if the utility should not recover from the last unsuccessful run. By default, the utility will disregard the current input file and read and complete the latest unsuccessful run (request) from the database. NOTE: if NO recovery is desired, the last unsuccessful run will be set to "Aborted" in the database.
<i>[-email recipient_email_address]</i>	Indicates the Email address of the user to receive the termination status of the utility. Multiple email addresses may be entered, separated by semicolons. If errors occurred, detail about the errors or how to retrieve the details will be present in the Email message.

4.7.21.2 Sample invocations of the RestoreOlaFromTape Utility

Below are some examples for invoking this utility:

1. **EcDIRestoreOlaFromTapeStart OPS -file </home/john/dplids.txt> -contents dplids – recovery no –email cmshared@ecs.nasa.gov**

Restores to the on-line archive from tape the DPL granules with the DPL IDs present in the dplids.txt flat file. The utility will NOT recover from an unsuccessful previous run and will set the restore from tape request to “Aborted” in the DPL database for the unsuccessful previous run. An Email with the request status will be sent to the cmshared@ecs.nasa.gov once the utility finishes the current request.

2. **EcDIRestoreOlaFromTapeStart OPS -file </home/john/ecsids.txt> -contents ecsids – recovery no**

Restores to the on-line archive from tape the DPL granules with the ECS IDs present in the ecsids.txt flat file. The utility will NOT recover from an unsuccessful previous run and will set the restore from tape request to “Aborted” in the DPL database for the unsuccessful previous run.

3. **EcDIRestoreOlaFromTapeStart OPS -file </home/john/dplfiles.txt> -contents dplfiles –recovery no**

Restores to the on-line archive from tape the DPL files with the full path and filenames specified in the dplfiles.txt flat file. The utility will NOT recover from an unsuccessful previous run and will set the restore from tape request to “Aborted” in the DPL database for the unsuccessful previous run.

4. **EcDIRestoreOlaFromTapeStart OPS -file </home/john/dplids.txt> -contents dplids**

Reruns the previous unsuccessful restore from tape request based on the information saved in the DPL database tables used by the utility. The current input file is NOT used. In order to restore the granules specified in the input file, the utility must be restarted after the recovery run completes.

4.7.21.3 RestoreOlaFromTape Utility Main Screen

The RestoreOlaFromTape Utility does not have a main screen. It has a command line interface only.

4.7.21.4 Required Operating Environment

The RestoreOlaFromTape Utility runs on Linux platforms. It will be deployed on the Data Pool machine.

4.7.21.5 Databases

Table 4.7.21-2 lists the supporting products this tool depends upon to function properly.

Table 4.7.21-2. Interface Protocols

Product Dependency	Protocols Used	Comments
Data Pool and AIM databases	SQL	Via SQL server machines
Sybase JDBC driver	JDBC	Requires proper install of jConnect (the Sybase JDBC driver).

If a Sybase error occurs, you are most likely to see the actual Sybase error string displayed on the screen and in the log. Some errors can be that the database server is unavailable, the connection to the database was dropped, or there was an error executing a stored procedure. In the event of a Sybase-sourced error, the utility immediately stops running.

In the event that a connection to the Data Pool database or AIM database cannot be established, the utility will exit immediately.

4.7.21.6 Configuration File Format – RestoreOlaFromTape.properties

The configuration file contains vital details about how the utility will operate. The utility will exit immediately if a configuration file is not available. The file is a plain text ASCII file and has the following format as shown in Table 4.7.21-3:

Table 4.7.21-3. Individual Configuration Parameters (1 of 2)

Parameter Name	Description
PGM_ID	Sybase connectivity, the ID (10000030) is used to decrypt the DB password based on ECS standards
SYBASE_HOST	Sybase connectivity, the host name for the SYBASE data server
SYBASE_PORT	Sybase connectivity, the port number for the SYBASE data server on the specified host
SYBASE_USER	Sybase connectivity, the user name (EcDIRestoreOlaFromTape) used to login to the SYBASE data server. DPL and AIM databases
SYBASE_DPL_DBNAME	Sybase connectivity, the database name for the DPL database
SYBASE_AIM_DBNAME	Sybase connectivity
SYBASE_DPL_POOL_SIZE	Sybase connectivity, the database connection pool size for the DPL
SYBASE_AIM_POOL_SIZE	Sybase connectivity, the database connection pool size for the AIM
SYBASE_JDBC_DRIVER_CLASS	Sybase connectivity
DB_RETRIES	Number of retries of a RETRYABLE DB operation (e.g. deadlock)
DB_SLEEPSECONDS	Number of sleep seconds between retries
SQL_TIMEOUT_SECONDS	Time in seconds that an SQL query will execute before timing out.

Table 4.7.21-3. Individual Configuration Parameters (2 of 2)

Parameter Name	Description
DB_BATCH_SIZE	The batch size for the database retrieve operations, its default value is 50
DEBUG_MESSAGES	(Y/N) indicates if detailed debugging information will be written to the log file.
CHECKSUM_SERVICE_HOSTS	The service hosts to be used for checksumming. The service hosts are configured in the format of <host_name_1>:<port_num>:<num_of_slots_1>, <host_name_2>:<port_num>:<num_of_slots_2>, ...
CHECKSUM_TIMEOUT	Number of seconds before timeout a checksum operation
COPY_SERVICE_HOSTS	The service hosts to be used for copy operation. The service hosts are configured in the format of <host_name_1>:<port_num>:<num_of_slots_1>, <host_name_2>:<port_num>:<num_of_slots_2>, ...
COPY_TIMEOUT	Number of seconds before timeout a copy operation
SNSM_QS_HOST	StorNext Metadata Server Quick Server host
SNSM_QS_PORT	StorNext Metadata Server Quick Server port
CONNECT_QS_RETRIES	Number of retries for Quick Server call failures
CONNECT_QS_RETRY_SECONDS	Number of sleep seconds between the retries of a Quick Server call
COPY_BLOCK_SIZE_KBYTES	copy block size used by EcAdCopy
COPY_RETRIES	number of retries for EcAdCopy on read/write failures
REQUEST_RETENTION_DAYS	The request retention time in days
EMAIL_SMTP_HOST	The Email SMTP server host
EMAIL_FROM_ADDRESS	Outbound email from address to operator
DEDICATED_TAPE_DRIVES	Number of tape drives (tapes) that can be concurrently used for restores.
CONCURRENT_RESTORES	Number of restores that can be issued concurrently for a given drive containing a restore tape. The restores will not happen concurrently per say but they will be enqueued by the tape management COTS and will be executed concurrently. The parameter optimizes tape reads by preventing the tape from being stopped during the restore. Recommended values can be anywhere between 5 and 10.
DTD_VERSION	DTD Version of xml files for DAP, PH, QA granules
DATA_CENTER_ID	DATA_CENTER_ID of xml files for DAP, PH, QA granules
CONCURRENT_GET_FILETAPEINFO	Number of threads that can be issued concurrently when retrieving and updating file tape information
MISR_SPECIAL_PROCESSING	controls if MISR Browse special processing module is ON (Y) or OFF (N)

4.7.21.7 Special Constraints

The RestoreOlaFromTape Utility runs only if the Data Pool and AIM database servers are running and if at least one checksum service host is available.

4.7.21.8 Outputs

Output of events and errors is always appended to a single log file.

4.7.21.9 Event and Error Messages

Events and error messages are written to the log file /usr/ecs/<mode>/CUSTOM/logs/EcDIRestoreOlaFromTape.log. If this log file already exists, the new information is automatically appended. If there is no existing log file by this name, a new log file with this name is automatically created.

Since the log file may grow to a considerable size after constant use, it is recommended that it be saved off into a separate file from time to time for maintainability.

4.7.21.10 Reports

None

4.7.22 RestoreTapeFromOla

The *RestoreTapeFromOla* utility will repair individual files that are lost or corrupted on tape based on the primary file instance that is present in the on-line archive. The files being restored must be inventoried both in the AIM and DPL databases because the utility does not create new AIM or DPL database entries. The utility shall:

- Allow DAAC staff to replace individual granules in the tape archive from their on-line copy (after verification that the on-line copy is still intact). Files will be renamed appropriately to conform to the tape archive naming conventions.
- Manage the capacity demand of bulk repairs to avoid serious degradation of operational workloads (e.g., limits on concurrent tape mounts, on-line archive reads, tape writes, and checksumming operations).

Notes:

- Since the on-line Browse archive is not part of the Data Pool, this repair function will not cover Browse archive repairs. They can be repaired using StorNext utilities like today.
- The *RestoreTapeFromOla* utility will not cover XML metadata files. The XML file archive restore function is performed using other procedures.

Input is provided via an input file.

4.7.22.1 Running the RestoreTapeFromOla Utility

The following command line syntax must be used to start the RestoreOlaFromTape Utility:

```
> EcDIRestoreTapeFromOlaStart <mode> -file <file name and path of input file whose contents type is specified by the -contents parameter> -contents mediaids | tapefiles | dplids | ecsids [-removeonlyfile] [-recovery [no]] [-email <usertoreceivestatusemail>]
```

Table 4.7.22-1 provides a description of the above command line parameters.

Table 4.7.22-1. RestoreTapeFromOla Utility Parameters (1 of 2)

Parameter Name	Description
-file <file name>	Name and path of the input file to be used by the utility
-contents <contents type>	The type of contents present in the file. Any of the following options are allowed: <ul style="list-style-type: none"> ⇒ mediaids: the input file contains the media IDs (tape labels) of the tapes that were lost / damaged. ⇒ tapefiles: the input file contains the complete file names and paths of the tape files that must be repaired. ⇒ dplids: the input file contains the DPL granule IDs of the tape granules that must be repaired ⇒ ecsids: the input file contains the ECS granule IDs (dbIDs) of the tape granules that must be repaired

Table 4.7.22-1. RestoreTapeFromOla Utility Parameters (2 of 2)

Parameter Name	Description
[-removereadonlyfile]	<p>Indicates that the utility should remove the original tape file from archive if the file cannot be restored to its original location. The utility always restores the file to the currently opened volume groups. Details below:</p> <p>If the option is not present, the utility will not try to remove the original tape file from archive. If the file cannot be restored to its original location, it will be restored in the currently opened volume group with the new file name, and the original file will remain on tape at the original location, without any corresponding AIM inventory record. The utility will not even try to remove the original file, regardless of the permissions on it.</p> <p>If the option is present, the utility will try to remove the original file. The file restored has the same name as original file. The utility will prompt the user to verify that the permissions to the RO volume group have been changed to RW if necessary:</p> <p>Have you changed the RO permissions to RW in the RO volume group affected by the restore (Y/N)?</p> <p>On Y the utility will proceed and:</p> <p>If the permissions to the affected files are RW, it will remove the original files that are affected.</p> <p>If the permissions to the affected files are RO, it will FAIL the restore of the granules involved. It is the responsibility of DAAC operations to inspect the log, identify the failed granules and rerun the utility after setting the correct RW permissions to the closed Volume Group. The reason for the failure is that if we would in fact restore the granule, the original file will remain on tape at the original location, and other application will find the bad copy.</p> <p>On N the utility will exit.</p>
[-recovery [no]]	<p>Indicates that the utility should not recover from the last unsuccessful run. By default, the utility will disregard the current input file and read and complete the latest unsuccessful run (request) from the database.</p> <p>NOTE: if NO recovery is desired, the last unsuccessful run will be set to "Aborted" in the database.</p>
[-email recipient_email_address]	<p>Indicates the Email address of the user to receive the termination status of the utility. Multiple email addresses may be specified, separated by semicolons. If errors occurred, detail about the errors or how to retrieve the details will be present in the Email message.</p>

4.7.22.2 Sample invocations of the RestoreOlaFromTape Utility

Below are some examples for invoking this utility:

- 1. EcDIRestoreTapeFromOlaStart OPS -file </home/john/mediads.txt> -contents mediads -recovery no -email cmsshared@ecs.nasa.gov**

Restores all files on the tape(s) specified in the mediaids.txt input file from their on-line archive copy. The utility will NOT recover from an unsuccessful previous run and will set the previous restore on-line archive to tape request to “Aborted” in the DPL database. An Email with the request status will be sent to the cmshared@ecs.nasa.gov once the utility finishes the current request.

2. EcDIRestoreTapeFromOlaStart OPS -file </home/john/tapefiles.txt> -contents tapefiles –recovery no

Restores the tapes files specified in the tapefiles.txt input file from their on-line archive copy. The utility will NOT recover from an unsuccessful previous run and will set the previous restore on-line archive to tape request to “Aborted” in the DPL database.

3. EcDIRestoreTapeFromOlaStart OPS -file </home/john/dplids.txt> -contents dplids –recovery no

Restores the granules with the DPL IDs specified in the dplids.txt input file from their on-line archive copy. The utility will NOT recover from an unsuccessful previous run and will set the previous restore on-line archive to tape request to “Aborted” in the DPL database.

4. EcDIRestoreTapeFromOlaStart OPS -file </home/john/ecslids.txt> -contents ecsids

Restores the granules with the ECS IDs specified in the ecsids.txt input file from their on-line archive copy. If there was an unsuccessful previous run, the utility will recover from that run based on the information saved in the DPL database tables used by the utility, and the current input file will not be used. The current runs must be restarted after the recovery run is completed.

4.7.22.3 RestoreTapeFromOla Utility Main Screen

The RestoreTapeFromOla Utility does not have a main screen. It has a command line interface only.

4.7.22.4 Required Operating Environment

The RestoreTapeFromOla Utility runs on Linux platforms. It will be deployed on the Data Pool machine.

4.7.22.5 Databases

Table 4.7.22-2 lists the supporting products this tool depends upon to function properly.

Table 4.7.22-2. Interface Protocols

Product Dependency	Protocols Used	Comments
Data Pool and AIM databases	SQL	Via SQL server machines
Sybase JDBC driver	JDBC	Requires proper install of jConnect (the Sybase JDBC driver).

If a Sybase error occurs, you are most likely to see the actual Sybase error string displayed on the screen and in the log. Some errors can be that the database server is unavailable, the connection to the database was dropped, or there was an error executing a stored procedure. In the event of a Sybase-sourced error, the utility immediately stops running.

In the event that a connection to the Data Pool database or AIM database cannot be established, the utility will exit immediately.

4.7.22.6 Configuration File Format – RestoreTapeFromOla.properties

The configuration file contains vital details about how the utility will operate. The utility will exit immediately if a configuration file is not available. The file is a plain text ASCII file and has the following format as shown in Table 4.7.22-3.

Table 4.7.22-3. Individual Configuration Parameters (1 of 2)

Parameter Name	Description
PGM_ID	Sybase connectivity, the ID (10000031) is used to decrypt the DB password based on ECS standards
SYBASE_HOST	Sybase connectivity, the host name for the SYBASE data server
SYBASE_PORT	Sybase connectivity, the port number for the SYBASE data server on the specified host
SYBASE_USER	Sybase connectivity, the user name (EcDIRestoreTapeFromOla) used to login to the SYBASE data server. DPL and AIM databases
SYBASE_DPL_DBNAME	Sybase connectivity, the database name for the DPL database
SYBASE_AIM_DBNAME	Sybase connectivity AIM database name
SYBASE_DPL_POOL_SIZE	Sybase connectivity, the database connection pool size for the DPL
SYBASE_AIM_POOL_SIZE	Sybase connectivity, the database connection pool size for the AIM
SYBASE_JDBC_DRIVER_CLASS	Sybase connectivity, JDBC driver class
DB_RETRIES	Number of retries of a RETRYABLE DB operation (e.g. deadlock)
DB_SLEEPSECONDS	Number of sleep seconds between retries
SQL_TIMEOUT_SECONDS	Time in seconds that an SQL query will execute before timing out.

Table 4.7.22-3. Individual Configuration Parameters (2 of 2)

Parameter Name	Description
DB_BATCH_SIZE	The batch size for the database retrieve operations, its default value is 50
DEBUG_MESSAGES	(Y/N) indicates if detailed debugging information will be written to the log file.
CHECKSUM_SERVICE_HOSTS	The service hosts to be used for checksumming. The service hosts are configured in the format of <host_name_1>:<port_num>:<num_of_slots_1>, <host_name_2>:<port_num>:<num_of_slots_2>, ...
CHECKSUM_TIMEOUT	Number of seconds before timeout a checksum operation
COPY_SERVICE_HOSTS	The service hosts to be used for copy operation. The service hosts are configured in the format of <host_name_1>:<port_num>:<num_of_slots_1>, <host_name_2>:<port_num>:<num_of_slots_2>, ...
COPY_TIMEOUT	Number of seconds before timeout a copy operation
SNSM_QS_HOST	StorNext Metadata Server Quick Server host
SNSM_QS_PORT	StorNext Metadata Server Quick Server port
SNSM_QS_OUTPUT_DIR	The directory where StorNext Metadata Server Quick Server puts the output files. The directory should be visible from both the host where the StorNext Metadata Server Quick Server runs and from the host where the RestoreTapeFromOla utility runs. The directory should not be shared with other applications.
CONNECT_QS_RETRIES	Number of retries for Quick Server call failures
CONNECT_QS_RETRY_SECONDS	Number of sleep seconds between the retries of a Quick Server call
COPY_BLOCK_SIZE_KBYTES	copy block size used by the copy utility
COPY_RETRIES	number of retries for the copy utility on read/write failures
REQUEST_RETENTION_DAYS	The request retention time in days
EMAIL_SMTP_HOST	The Email SMTP server host
EMAIL_FROM_ADDRESS	Outbound email from address to operator
CONCURRENT_TAPE_ARCHIVE_CACHE_WRITES	Number of concurrent writes to the tape archive cache. This is a throttling mechanism that controls how many files can be concurrently copied from the on-line archive to tape.

4.7.22.7 Special Constraints

The RestoreTapeFromOla Utility runs only if the Data Pool and AIM database servers are running and if at least one checksum service host is available.

4.7.22.8 Outputs

Output of events and errors is always appended to a single log file.

4.7.22.9 Event and Error Messages

Events and error messages are written to the log file /usr/ecs/<mode>/CUSTOM/logs/EcDIRestoreTapeFromOla.log. If this log file already exists, the new information is automatically appended. If there is no existing log file by this name, a new log file with this name is automatically created.

Since the log file may grow to a considerable size after constant use, it is recommended that it be saved off into a separate file from time to time for maintainability.

4.7.22.10 Reports

None

4.7.23 Migration Stat Viewer Tool

Deleted. Not applicable for Release 7.23.

This page intentionally left blank.

4.7.24 Migration Checksum Error Support Tool

Deleted. Not applicable in Release 7.23.

This page intentionally left blank.

4.7.25 Migration Migrate Error Support Tool

Deleted. Not applicable in Release 7.23.

This page intentionally left blank.

4.7.26 Migration Stage Error Support Tool

Deleted. Not applicable in Release 7.23.

This page intentionally left blank.

4.7.27 Migration Insert Error Support Tool

Deleted. Not applicable for Release 7.23.

This page intentionally left blank.

4.7.28 EMS Dataset Extract Utility

The Earth Science Data and Information System (ESDIS) Metrics System (EMS) Dataset Extract utility provides DAAC Operations Staff an operational support tool that automatically extracts data and information from DAAC databases and transmits the data to the EMS metric reporting tool.

The EMS Dataset Extract utility extracts data from DAAC operational database tables and outputs the data into ASCII text flat files. The utility is designed to run as a CRON on a daily basis. The flat files prepared for EMS are formatted so that one line in the file represents one record of information. The output files have field information delimited by “|&|”. The flat files are transferred via ‘SCP’ to the centralized EMS location from which EMS metric reports can be generated.

The EMS Dataset Extract utility is run with a set of optional and required DAAC defined command line parameters. The utility can also be run manually from the Linux command prompt with the optional and required parameters specified. The utility will behave differently depending on the combination of parameters entered. Daily checks by EMS operations personnel will ensure that the data exported by the EMS Dataset Extract utility was received at the central EMS location. Updated flat files are to be sent to EMS whenever data processing failures are encountered or data corruption is detected.

The utility is designed to extract data in periods of 24-hours or one day. If the data transfer fails for a few days, the utility is designed to perform data recovery automatically for the period of time missed as soon as communication is restored.

4.7.28.1 Running the EMS Dataset Extract Utility

The EMS Dataset Extract utility is run from a CRON (see Section 4.7.28.2) or started by entering the following command from the /usr/ecs/<mode>/CUSTOM/utilities directory Linux command line:

```
>EcDbEMSdataExtractor.pl-m <mode> -s <start date> -e <end date> -x <extract type> -v -o -i
```

Table 4.7.28-1 shows the parameters for the EMS Dataset Extract utility.

Table 4.7.28-1. Command Line Parameters of the EMS Dataset Extract Utility (1 of 2)

Parameter Name	Description
-(m)ode	Mandatory. Specifies the mode in which the extraction is to occur. It must be a valid, existing mode with a format of OPS or TS[1-4] or DEV0[1-9].
-(s)tartdate	Optional. The startDate time for ExtractType processing with a format of “mm dd, yyyy” or “mm/dd/yyyy”.
-(e)nddate	Optional. The endDate time for ExtractType processing with a format of “mm dd, yyyy” or “mm/dd/yyyy”.
-e(x)tracttype	Optional. Identifies the type of data being extracted. The following values are valid extracttypes: Meta, searchExp, Ing, Arch, DistFTP, DistHTTP, DistMedia.

Table 4.7.28-1. Command Line Parameters of the EMS Dataset Extract Utility (2 of 2)

Parameter Name	Description
-(o)verride	Optional. Identifies whether a period of time longer than the default 24-hour period will be used for the date range for the extracttype. If the override command line parameter is specified, entries are required for the startdate, enddate, and extracttype.
-(v)erbose	Optional. Prints messages to screen as well as log
-(i)ntial	Optional. Will enter 'default' into the ExecutionMode field in EcEMSExtractRecord table for the indicated dataset.

The -mode parameter is mandatory. For each command line parameter, a dash “-“ followed by the letter in parenthesis indicated in the above table can be used instead of the full parameter name.

Table 4.7.28-2 describes datasets that are extracted and exported using the extraction utility:

Table 4.7.28-2. Datasets of the EMS Dataset Extract Utility

Dataset Name	Description
Meta	Product attribute metadata
searchExp	Product attribute search
Ing	Data Ingest
Arch	Data Archive
DistFTP (DataPool, FtpPush, FtpPull) DistHTTP (DataPool)	Physical media distribution orders
DistMedia (Cdrom, Dlt, Dvd)	Electronic media distribution orders

NOTE: Initially the EMS system needed to know about the DAAC users and Product attributes. This data was extracted and sent to EMS. Metadata and Product Attribute Flat file information maintained in the AIM database was sent. User information was also sent. Also, prior to running the EMS Dataset Extract utility in the default execution mode a baseline for the default execution was established. The baseline was the date from which the default execution should start processing. A record of this baseline is recorded in the MSS database EcEMSExtractRecord table. Each time the EMS Dataset Extract utility is executed in default mode this table is checked to determine the last time a dataset was processed and to determine the date range to use for the current run of the dataset.

4.7.28.2 EMS Dataset Extract Utility Examples

Below are examples for invoking this tool:

1. **EcDbEMSdataExtractor.pl -m <mode>**

Running the EMS Dataset Extract utility with only the `-m` option is the default way to run the utility, and this should be the only parameter used when running the utility as a CRON. To set up the CRON, access the Linux server as the Sybase user. Set up the CRON by running the CRONTAB `-e` command. The command will be something like: `51 16 * * 2 (export LD_LIBRARY_PATH=/tools/sybOCv12.5.1/lib:/home/cmops/lib:/bin/csh -c "cd /usr/ecs/TS1/CUSTOM/utilities; EcDbEMSdataExtractor.pl -m TS1")`. This command may be different based on the configuration for the server.

Since the “start date” and “end date” parameters are not provided, the EMS Dataset Extract utility will access the `EcEMSExtractRecord` table for each dataset and retrieve the most current record for the dataset that has been marked “Default” in the `ExecutionMode` field. The beginning “start date” and “end date” for the Dataset run will be calculated based on the retrieved value for the last run of the Dataset.

2. **EcDbEMSdataExtractor.pl -m <mode> -s “start date” -e “end date” -x DistFTP**

Running the EMS Dataset Extract with these options will create output files for DistFTP data or the specified dataset for each day greater than or equal to the start date and less than the end date. A record for each day of the run will be inserted into the `EcEMSExtractRecord` table. The date range specified by the start date and end date must be at least 24 hours. The dates should be entered without hour or minutes specified. A record of the run will also be logged in the log file. If the `-x` parameter is omitted, then output files for all Datasets will be created.

3. **EcDbEMSdataExtractor.pl -m <mode> -s “start date” -e “end date”**

Running the EMS Dataset Extract with these options will create output files for all datasets for each day greater than or equal to the start date and less than the end date. A record for each day of the run will be inserted into the `EcEMSExtractRecord` table. The date range specified by the start date and end date must be at least 24 hours. The dates should be entered without hour or minutes specified. A record of the run will also be logged in the log file.

4. **EcDbEMSdataExtractor.pl -m <mode> -s "start date" -e "end date(start date + one day)" -i**

The preceding command should be run from the Linux prompt to initialize the datasets for default execution: If the `-x` option is used then only the specified Dataset will be initialized. For the Dataset execution “Default” will be placed in the `ExecutionMode` field for the record of the run. Subsequent runs of the EMS Dataset Extract utility without the date range specified will access the `EcEMSExtractRecord` table for the dataset and retrieve the most current record that has been marked “Default” in the `ExecutionMode` field. The beginning “start date” and “end date” for the Dataset run will be calculated based on the retrieved value for the last run of the Dataset.

4.7.28.3 Required Operating Environment

The EMS Dataset Extract utility runs on the Linux platform.

4.7.28.4 Interfaces and Data Types

Table 4.7.28-3 lists the supporting products that this tool depends upon in order to function properly.

Table 4.7.28-3. Interface Protocols

Product Dependency	Protocols Used	Comments
Sybase	SQL	Via SQL server machine.
Perl module	Perl	Module to connect to the database and print out the nicely formatted help page.

4.7.28.5 Configuration File Format – EcDbEMSdataExtractor.CFG properties

The EcDbEMSdataExtractor.pl utility requires a configuration file. This configuration file, “EcDbEMSdataExtractor.CFG”, is located in the /usr/ecs/<mode>/CUSTOM/cfg directory on the x4spl01 server. All edits of the EcDbEMSdataExtractor.CFG” file will be implemented using a Linux editor, such as “vi”. The configuration file contains vital details about how to connect to the Sybase server and EMS host machine. Without this file, the tool cannot run. Table 4.7.28-4 describes the configuration parameters:

Table 4.7.28-4. Configuration Parameters (1 of 2)

Parameter Name	Recommended Value	Description
SERVER	<x4dbl01_srvr>	Enter sybase server name e.g. x4dbl01_srvr.
PROVIDER	<DAAC NAME>	Enter provider name e.g. DAAC identifier.
EMSEXTRACTDIR	/usr/ecs/<mode>/CUSTOM/data/DSS	Enter EMS extraction directory location. This is the directory path specifying where data is extracted to when bcp'd out of database e.g. /usr/ecs/<mode>/CUSTOM/data/DSS.
EMSUSER	cmshared OR allmode	Enter user name to gain access to host represented by IPADDRESS - provided by EMS team.
PGMID	7000900	Static value. Same for all DAACs and Modes.

Table 4.7.28-4. Configuration Parameters (2 of 2)

Parameter Name	Recommended Value	Description
DBUSER	EcDbEMSdataExtractor	Static value. Same for all DAACs and Modes.
IPADDRESS	The following is an example <123.456.789.1>	Enter IP Address or host name e.g. ws1.ems.eosdis.nasa.gov - provided by EMS team. The IP Address identifying EMS host to SCP the data files produced by the utility.
STORNEXT	<Descriptor Directory Path>	Location of ESDT descriptor files.
LAG	<-1>	The default LAG time set in configuration file is -1, meaning if the EMS extract script is run today for default configuration, data will be provided up to the day before yesterday. Setting the value to 0 will provide data up to yesterday. Setting the value to 1 will provide data up to current time
DESTINATIONDIR	<blank>	The default DESTINATIONDIR is blank, meaning that the data extracted by the EMS extract script will be sent to the home directory on the server specified by IPADDRESS. This allows for sending data to a subdirectory. The full path for the subdirectory should be specified.

4.7.28.6 Flat Files Naming Convention

The name of the flat file consists of three parts: timestamp, root file name, and extension.

1. Timestamp

Timestamp designates the year, month and day the content of the data file was created. If a revised file is being sent (see below) the timestamp represents the date on which the original file was created with the .rev<1-n> file extension used to identify the file as a revision.

2. Root File Name

Root File Name consists of the Provider, File Type, and Data Source components of a file name. It must be unique for each provider.

3. Extension

Extension designates the type of file and the revision status by appending a number 1-n to the end of the file name.

The name of the data files is in the following format for all the Data Providers:

<YYYYMMDD>_<Provider>_<FileType>_<DataSource>.flt.rev<1-n>

Where: YYYY designates the 4 digit year for the time the Data Ingest Flat File was created

MM designates the 2 digit month, 01 through 12

DD_ designates the 2 digit day, 01 through 31, followed by an underscore
Provider_ designates the provider of the data, mutually agreed upon acronym defined in the Operations Agreements (OA), followed by an underscore.
FileType_ designates the type of flat file sent, followed by an underscore where type is one of the following values:

“Ing”	Data Ingest Flat Files
“Arch”	Data Archive Flat Files
“searchExp”	Product Attribute Search Files
“Meta”	Product Attribute Metadata Flat Files
“DistMedia”	Media distribution log Flat File
“DistFTP”	FTP distribution log Flat File
“DistHTTP”	HTTP distribution log Flat File

DataSource designates the database table
.flt indicates the file is a flat file
.rev<1-n> indicates the file has been resent because of errors; the number is incremented for each update (e.g. rev1, rev2, rev3... revN)

4.7.28.7 Flat Files Updates

Updated flat files are to be sent to the EMS whenever data processing failures are encountered or data corruption is detected. The naming convention for the updated data flat file must follow the format described above with the appended “.rev<1-n>”. The EMS extract utility run manually will facilitate this update process.

4.7.28.8 Flat File Format

A flat file contains the output data from DAAC operational database tables. Each line of the flat file represents one record information and each field of a record is ASCII text delimited by “|&|”. The extracted flat files are located in the directory that is specified by the “EMSEXTRACTDIR” (Table 4.7.28-4) in configuration file.

1. Data Ingest Flat File

Table 4.7.28-5 describes the data ingest flat file layout information:

Table 4.7.28-5. Data Ingest Flat File Layout (1 of 2)

Field Name	Description	Table Name	Column Name	Column Datatype	Nulls
ECSGranuleID	AIM UID for a granule	InHistoricGranule	ECSGranuleID	numeric (16)	No
Data Type	This holds primary ESDT short-name of an ECS data type that is handled by a particular data server. (i.e.,AM-1 L0, SAGEIII L0, Radat ALT L0, Landsat7 L0R, SeaWinds,Ancillary, etc.)	InHistoricGranule	Data Type	vchar (32)	No
DataGranuleVolume	Total data volume to be ingested for a data granule in an ingest request. The total data volume for the data granule is determined by summing the data volumes for the files comprising the data granule.	InHistoricGranule	DataGranuleVolume	float(8)	Yes
DataGranuleState	This is the state of a data granule.	InHistoricGranule	DataGranuleState	varchar(30)	Yes
ExternalDataProvider	This is the name of the External data provider.	InHistoricGranule	ExternalDataProvider	varchar(20)	No
ProcessingStartDateTime	This is the processing start date and time for ingest of a data granule.	InHistoricRequest	ProcessingStartDateTime	varchar(18)	Yes
ProcessingEndDateTime	This is the processing end date and time for ingest of a data granule.	InHistoricRequest	ProcessingEndDateTime	varchar(18)	Yes

Table 4.7.28-5. Data Ingest Flat File Layout (2 of 2)

Field Name	Description	Table Name	Column Name	Column Datatype	Nulls
TimeToArchive	Time (in seconds) from submit of archive request to Data Server to receipt of completion status (success or fail).	InHistoricRequest	TimeToArchive	int	Yes
TimeToPreprocess	Time (in seconds) from start of preprocessing of granule to time of completion (success or fail) of preprocessing.	InHistoricRequest	TimeToPreprocess	int	Yes
TimeToXfer	Time (seconds) from start of transfer for 1st file in granule to time of receipt of status (success or fail) for last file in granule.	InHistoricRequest	TimeToXfer	int	Yes

2. Data Archive Flat File

Table 4.7.28-6 describes the data archive flat file layout information:

Table 4.7.28-6. Data Archive Flat File Layout (1 of 2)

Field Name	Description	Table Name	Column Name	Column Datatype	Nulls
dbID	The unique ID which identifies granule.	EMSArchData	dbID	numeric (16)	Yes
ShortName	Short name associated with the collection or granule.	EMSArchData	ShortName	varchar(8)	Yes
sizeDataGranule	Size of granule in Bytes.	EMSArchData	sizeDataGranule	float	Yes
totalFiles	Total number of files.	EMSArchData	totalFiles	int	Yes
insertTime	The time of original insertion.	EMSArchData	insertTime	varchar(18)	Yes
BeginningDateTime	The attribute within AIM Inventory that allows both the SingleDateTime (TimeofDay) and RangeDateTime(RangeBeginningDate) to be efficiently indexed and searched.	EMSArchData	BeginningDateTime	varchar(18)	Yes

Table 4.7.28-6. Data Archive Flat File Layout (2 of 2)

Field Name	Description	Table Name	Column Name	Column Datatype	Nulls
EndingDateTime	The attribute within AIM Inventory that allows both the SingleDateTime (TimeOfDay) and RangeDateTime (RangeEndingDate) to be efficiently indexed and searched.	EMSArchData	EndingDateTime	varchar(18)	Yes
ProductionDateTime	The date and time a specific granule was produced by a PGE.	EMSArchData	ProductionDateTime	varchar(18)	Yes
LocalGranuleID	Data provider-supplied identifier for a granule that ECS ingests and is required to capture.	EMSArchData	LocalGranuleID	varchar(80)	Yes
VersionID	Version identifier of the data collection.	EMSArchData	VersionID	tinyint	Yes
DeleteFromArchive	Granules deleted from the archives. 'Y' =Scheduled for deletion, 'N' = Not scheduled for deletion, 'H' = Hidden, 'G' = Never delete.	EMSArchData	DeleteFromArchive	char(1)	Yes
deleteEffectiveDate	Date on which the entry may be deleted.	EMSArchData	deleteEffectiveDate	varchar(18)	Yes
lastUpdate	The time of the last update.	EMSArchData	lastUpdate	varchar(18)	Yes

3. Product Attribute Search Flat Files

Table 4.7.28-7 describes the product attribute search flat file layout information:

Table 4.7.28-7. Product Attribute Search Flat File Layout

Field Name	Description	Table Name	Column Name	Column Datatype	Nulls
ShortName	Short name associated with the collection or granule.	DsMdCollections	ShortName	varchar(8)	No
subType	The internally created column used to hold the ShortName.	DsMdCollections	subType	varchar(30)	No
dataSource	The source that provides data.	EcEMSExtractRecord	DataSource	varchar(50)	Yes

4. Product Attribute Metadata Flat Files

Table 4.7.28-8 describes the product attribute metadata flat file layout information:

Table 4.7.28-8. Product Attribute Metadata Flat File Layout (1 of 2)

Field Name	Description	Table Name	Column Name	Column Datatype	Nulls
ShortName	Short name associated with the collection or granule.	DsMdCollections EMSShortNameTemp	ShortName	varchar(8)	No
LongName	The long name associated with the collection includes dataset name/product name. This is the reference name used in describing the scientific contents of the data collection.	EMSShortNameTemp	LongName	varchar(80)	No
ProcessingLevelID	This attribute reflects the classification of the science data processing level, which defines in general terms the characteristics of the output of the processing performed.	DsMdCollections	ProcessingLevelID	char(6)	Yes
TopicKeywords	Keyword that describes the ShortName science area.	EMSShortNameTemp	TopicKeywords	varchar(500)	Yes
ProcessingCenter	Center where collection was or is being processed. i.e. name of DAAC or SCF.	DsMdCollections	ProcessingCenter	varchar(20)	Yes
ArchiveCenter	Center where collection is archived.	DsMdCollections	ArchiveCenter	varchar(20)	No
Missions	Related missions, Aqua, Aura, etc.	EMSShortNameTemp	Missions	varchar(500)	Yes

Table 4.7.28-8. Product Attribute Metadata Flat File Layout (2 of 2)

Field Name	Description	Table Name	Column Name	Column Datatype	Nulls
Instruments	An integrated collection of hardware containing one or more sensors and associated controls designed to produce data on an environment. For a multiinstrument product from one mission, list all instruments separated by a comma (.). If the product is a combined product from multi-missions involving multiple instruments, a group of the instruments from each mission should be separated by a semi-colon (;).	EMSShortNameTemp	Instruments	varchar(500)	Yes
eosFlag	Constant, set to 'E'				No
productFlag	Constant, set to '1'				No

5. Electronic Media Distribution Flat Files

Table 4.7.28-9 describes the electronic/physical media distribution flat file layout information:

Table 4.7.28-9. Electronic Media Distribution Flat File Layout (1 of 2)

Field Name	Description	Table Name	Column Name	Column Datatype	Nulls
requestId	Identifier for a request	EcAcRequest	requestId	varchar(10)	No
orderId	Identifier for an order	EcAcRequest	orderId	varchar(10)	No
userId	Identification of user submitting a request for distribution; Ftp User corresponds to ftpAddress field in EcAcRequest.	EcAcOrder	userId	varchar(14)	No
orderSource	Origination of this order (MTMGW, SSS, DPLGUI, VOGW, etc).	EcAcOrder	orderSource	varchar(21)	Yes
orderType	The type of an order.	EcAcOrder	orderType	varchar(2)	Yes

Table 4.7.28-9. Electronic Media Distribution Flat File Layout (2 of 2)

Field Name	Description	Table Name	Column Name	Column Datatype	Nulls
ShortName	This name will identify the short name associated with the collection or granule.	OmRequestGranule	EsdtType	char(12)	Yes
VersionID	Version identifier of the data collection.	OmRequestGranule	EsdtType	char(12)	Yes
finishDateTime	Date/Time this request was marked done in EcAcRequest.finishDateTime.	EcAcRequest	finishDateTime	datetime	Yes
tranDuration	Transfer time for request.	EcAcRequest	receiveDateTime finishDateTime	Datetime Datetime	Yes Yes
ECS_GranuleId	Unique identifier for granule from AIM. Internal GranId may be used.	OmRequestGranule OmFile	ECS_GranuleId GranId	numeric(16) numeric(16)	Yes Yes
StatusDesc	Description of the state of the granule.	EcAcRequest OmStatus	StatusDesc requestStatus	varchar(25) char(30)	No Yes
eMailAddr	Email Address associated with this request.	EcAcRequest	eMailAddr	varchar(255)	Yes
Billable	Contains billing related Information.	OmRequestGranule	BillingInfo	varchar(255)	Yes
FileType	S or M for Science File or MetaData File.	OmFile OmRequestGranule	FileType GranType	char(1) char(2)	Yes Yes
FileSize	Size of file	OmFile	FileSize	float(8)	Yes
fileNamePath	Location of file in datapool. Or the ESDT name if file name is empty.	OmRequestGranule OmFile	DirectoryPath FileName EsdtType	varchar(255) varchar(255) char(12)	Yes Yes Yes
Domain	Contains Ftp Host specified in MSS. Or Email address.	EcAcRequest	eMailAddr destinationNode	varchar(255) varchar(100)	Yes Yes
shipAddrCity	City associated with shipping address.	EcAcRequest	shipAddrCity	varchar(35)	Yes
shipAddrState	State associated with shipping address.	EcAcRequest	shipAddrState	varchar(20)	Yes
shipAddrZip	Zip Code associated with shipping address.	EcAcRequest	shipAddrZip	varchar(15)	Yes
shipAddrCountry	Country associated with shipping address.	EcAcRequest	shipAddrCountry	varchar(30)	Yes
IntendedUsage	Intended usage of the request.	OmRequestGranule	IntendedUsage	varchar(100)	Yes

Both FTP, HTTP, and Media methods generate the same flat files. The layouts generated from MSS/OMS database are the same as the layouts in Table 4.7.28-9.

Table 4.7.28-10 describes the physical media distribution flat file layout information generated from DataPool database for DistFTP and DistHTTP:

Table 4.7.28-10. Media Distribution Flat File Layout

Field Name	Description	Table Name	Column Name	Column Datatype	Nulls
dbID	The unique ID which identifies the granule.	DIGranuleAccess	dbId	ID	No
age	The difference between the time at which the file was accessed through FTP or Web and the time at which the file was inserted into Data Pool.	DIGranuleAccess	age	int	Yes
fileSize	The size of the browse file stored on the Data Pool disk. The size of the file in Data Pool.	DIGranuleAccess	fileSize	numeric(16,0)	No
fileType	The type of file.	DIGranuleAccess	fileType	varchar(10)	Yes
accessTime	The time at which the file was accessed through FTP or Web.	DIGranuleAccess	accessTime	datetime	No
ecsId	The ID that identifies the ECS browse granule. It matches the browse id in AIM database. The unique ID which identifies the granule.	DIGranuleAccess	ecsId	ID	No
transferTime	Total transfer time in seconds.	DIGranuleAccess	transferTime	int	Yes
ipAddress	IP Address of the user.	DIGranuleAccess	ipAddress	char(15)	No
fileName	Associates to file name. If the file name is null, use ShortName as fileName.	DIGranuleAccess DsMdGranules	filename ShortName	varchar(255) char(8)	No No
ShortName	This name will identify the short name associated with the collection or granule.	DsMdGranules	ShortName	char(8)	No
VersionId	Version identifier of the data collection.	DsMdGranules	VersionID	tinyint	No

4.7.28.9 Special Constraints

The EMS Dataset Extract utility runs only if the Sybase server is operational. EMS code must be installed in the mode. The EMS configuration file must be configured. SCP must be configured to run in the user environment from which the extract utility will be executed. EMS utility initial set-up should have been executed in the mode.

4.7.28.10 Outputs

Outputs will be printed to standard out if the `-v` flag is included with on the command line of the EMS Dataset Extract utility. Messages are also output to the `EcDbEMSdataExtractor.log` file (see Section 4.7.28.12). The DAAC Operations Staff should review the messages printed to the log file.

4.7.28.11 Event and Error Messages

Error messages will be displayed on standard out if the `-v` flag is included with the executed EMS command. Error messages will be logged in the `EcDbEMSdataExtractor.log` file (see Section 4.7.28.12). EMS Dataset Extract utility events are recorded in the MSS database `EcEMSextractRecord` table. Field descriptions for the `EcEMSextractRecord` table are described in Table 4.7.28-11.

Table 4.7.28-11. EcEMSextractRecord Table

Name	Datatype	Null	Description
ExtractId	numeric(8,0)	No	Monotonic Key.
ExtractType	varchar(255)	Yes	Dataset type, ie, Arch, DistFTP, DistHTTP, DistMedia, Ing, Meta, or searchExp.
RunStartTime	datetime	Yes	The time the dataset began processing.
RunCompletionTime	datetime	Yes	The time the dataset completed processing.
StartDate	datetime	Yes	The Start Date of the dataset run.
EndDate	datetime	Yes	The End Date of the dataset run.
ExtractFileName	varchar(500)	Yes	The name of the Extract File, including the directory path.
FTPcompletionTime	datetime	Yes	The date the dataset was SCP'd to IP address indicated in the configuration file.
ExecutionMode	varchar(8)	Yes	Execution Mode of the dataset run; either Default or override.
MediaType	varchar(20)	Yes	MediaType of dataset run; either NULL, DLT, DVD, Scp, CDROM, FtpPull, or FtpPush.
DataSource	varchar(50)	Yes	Mode and Media type combined – used in constructing the ExtractFileName.
Provider	varchar(50)	Yes	DAAC identifier.

4.7.28.12 Logs

The tool logs messages in the `/usr/ecs/<mode>/CUSTOM/logs/EcDbEMSdataExtractor.log` file.

4.7.28.13 Recovery

The EMS Dataset Extract utility supports automatic recovery from an interrupted run. If the utility has not been run for a period of time, then the utility can start running from the time it was previously run and files will be generated for the missing days. Also, if a dataset file was extracted to the extract directory, but not SCP'd to EMS, a subsequent run of the utility will SCP this file and mark the file as SCP'd in the EcEMSExtractRecord table by updating the FTPcompletionTime for the file record. Also, if a dataset file has been removed from the extract directory, but not SCP'd, a subsequent run of the utility will mark the record as SCP'd, in the EcEMSExtractRecord table by updating the FTPcompletionTime with the date "Jan 1, 1900" and a note documenting this will be written to the log.

This page intentionally left blank.