

4.8.13 Order Manager Command Line Utility

The Order Manager Command Line utility provides a mechanism by which the ECS Operations Staff can submit order requests into the Order Manager System (OMS) database directly without knowing whether the Order Manager Server is up or down. The order request submitted by the Command Line utility is in ODL format, which conforms to the Product Request ODL protocol in the ICD between the EOSDIS Core System (ECS) and the Version 0 System for Interoperability, with a few extensions.

4.8.13.1 Quick Start Using the Order Manager Command Line Utility

To execute the Order Manager Command line utility, use the command line interface command below.

4.8.13.1.1 Invoking Order Manager from the Command Line Interface

To execute the Order Manager from the command line interface, use the Command Line utility syntax provided below:

```
EcSrOmCliDriverStart <MODE > <rootname of ODL files> <numRequest> [<submissionInterval>  
<dbRetries> <dbRetryInterval>]
```

The mode parameter is required to indicate the mode (i.e., OPS, TS1, or TS2) in which the utility is being run. The command line parameters supported are described in Table 4.8.13-1.

Table 4.8.13-1. Order Manager Command Line Parameters

Parameter Name	Required	Description
rootname of ODL files	Yes	Specifies the full pathname of root name of ODL files. For example, if there are two requests to be submitted concurrently, there must be two ODL files with the same root name, say odl.rqst, but different suffixes "0" and "1" for each file (i.e., there must be two files named odl.rqst.0 and odl.rqst.1 on the disk). The root name of ODL files in this case is <fullpath>/odl.rqst. The program automatically appends those suffixes for you, starting from 0.
numRequest	Yes	Specifies the number of requests the Command Line utility submits concurrently.
submissionInterval	No	Specifies how many seconds apart the requests are submitted. The default value 0 means all the requests are submitted with no submission interval (i.e., at the same time).
dbRetries	No	Specifies how many db retries the utility tries when the OMS database is inaccessible. The default value is 2 (times).
dbRetryInterval	No	Specifies how many seconds apart between retries when the OMS database is inaccessible. The default value is 10 (seconds).

4.8.13.1.2 Order Manager Command Line Utility Configuration File

The Command Line Utility has an associated configuration file with values stored in a basic PARAMETER = VALUE format. Table 4.8.13-2 describes its contents:

Table 4.8.13-2. Order Manager Configuration File Parameters

Parameter Name	Value Description
Name	EcOmSrCliDriver
ProgramID	1300005
ApplicationID	1300000
Site	DAAC Name
SubSystem	OMS
MajorVersion	1
MinorVersion	0
AppLogSize	The maximum ALOG size
AppLogLevel	ALOG level
DebugLevel	Debug log level
Release	B
PrincipalName	EcOmSrCliDriver
SDSRV_SYBASE_SERVER	Name of OMS Sybase SQL Server
SYBINTERFACES	Location of Sybase open client library interface file
DSSSrUNIXEnv	SYBASE DSQUERY
DBMAXRESULTS	Maximum database return rows
DBNAME	OMS database name
DBPASSWDSEED	1300005 (the seed used to get Command Line utility database login password)
DBUSERNAME	OmSrCliDriver (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 SQS Server
SDSRV_DB_MAX_JOINS	Maximum number of database join operations
DSSSrEnv_DB	DBUSERNAME DBPASSWDSEED DBNAME DBMAXRESULTS SYBINTERFACES SDSRV_SYBASE_SERVER SDSRV_DB_MAX_JOINS
DSSSrEnv	\$DSSSrEnv_DB

4.8.13.1.3 ODL Template File

There are brackets ([]), and braces ({}) around some of the lines and groups. The brackets mean it is optional and subject to change its contents. The braces mean it is not optional but subject to change its contents. The lines or group of lines with no brackets or braces around them mean: “do not change them.”

To use the template file:

Step 1. Copy the template file to a new file.

Step 2. Customize those lines and groups with the brackets or braces in the new file.

Step 3. Remove the brackets and braces around the lines and groups from the new file.

4.8.13.1.4 ODL Template File for “FtpPull” Media Type

```
GROUP = PRODUCT_REQUEST
MESSAGE_ID = "B1027711830"
[REQUEST_ID = "37475:27364"]
```

The above line is optional. If it is there, the value part must be in the format of “order id:request id” which you retrieve from the MSS database, in this case <order id>=37475 and <request id>=27364. If it is not there, command line utility creates an order id and request id for this request.

```
DATA_CENTER_ID = "ECS-TEST"
[ECS_AUTHENTICATOR = "labuser"]
```

The above line is optional. If it is there, the value ought to be a valid ECS user in the ECS User Registration Database. If it is not there, this request is regarded as an “ECSGuest” user.

```
GROUP = USER_AFFILIATION
CATEGORY = "USA"
TYPE = "GOVERNMENT"
END_GROUP = USER_AFFILIATION
{
GROUP = CONTACT_ADDRESS
TITLE = ""
FIRST_NAME = "Yu"
MIDDLE_INITIAL = ""
LAST_NAME = "Zhongfei"
ORGANIZATION = ""
ADDRESS = ("abcd")
CITY = "Landover"
STATE = "MD"
ZIP = ""
COUNTRY = "UNITED STATES"
PHONE = "301-925-1042"
FAX = ""
EMAIL = "zyu@eos.hitc.com"
END_GROUP = CONTACT_ADDRESS
}
```

The above group is not optional, but the contents of each line could be customized.

```
{
GROUP = SHIPPING_ADDRESS
TITLE = ""
FIRST_NAME = "Yu"
MIDDLE_INITIAL = ""
LAST_NAME = "Zhongfei"
ORGANIZATION = ""
ADDRESS = ("abcd")
CITY = "Landover"
STATE = "MD"
ZIP = ""
COUNTRY = "UNITED STATES"
PHONE = "301-925-1042"
FAX = ""
EMAIL = "zyu@eos.hitc.com"
```

```
END_GROUP = SHIPPING_ADDRESS
}
```

The above group is not optional, but the contents of each line could be customized.

```
{
GROUP = BILLING_ADDRESS
TITLE = ""
FIRST_NAME = "Yu"
MIDDLE_INITIAL = ""
LAST_NAME = "Zhongfei"
ORGANIZATION = ""
ADDRESS = ("abcd")
CITY = "Landover"
STATE = "MD"
ZIP = ""
COUNTRY = "UNITED STATES"
PHONE = "301-925-1042"
FAX = ""
EMAIL = "zyu@eos.hitc.com"
END_GROUP = BILLING_ADDRESS
}
```

The above group is not optional, but the contents of each line could be customized.

```
GROUP = LINE_ITEM
{DATASET_ID = "LANDSAT-7 LEVEL-0R FLOATING SCENES V002"}
```

This line could be changed to the ESDT long name matching with the granule given in the next line.

```
{PACKAGE_ID = "SC:L70R.002:23420"}
```

This line could be customized in the format of "granule type:ESDT shortname:ESDT version id:db id."

```
PROCESSING_OPTIONS = "Native Granule"
```

```
{MEDIA_TYPE = "FtpPull"}
```

This line could be customized to any media type such as CDROM, DVD, DLT, or 8MM.

```
{MEDIA_FORMAT = "FILEFORMAT"}
```

This line could be changed to match the media type given in the above line.

```
EST_COST = 777.88
```

```
[
GROUP = SUBSET_SPEC
GROUP = SPECIALIZED_CRITERIA
CRITERIA_NAME = "Band Subsetting"
CRITERIA_TYPE = "STRING"
CRITERIA_VALUE = ("QA_BAND2_PRESENT", "QA_BAND3_PRESENT",
"QA_BAND4_PRESENT", "QA_BAND5_PRESENT", "QA_BAND6_PRESENT_F1",
"QA_BAND6_PRESENT_F2", "QA_BAND7_PRESENT", "QA_BAND8_PRESENT")
END_GROUP = SPECIALIZED_CRITERIA
GROUP = SPECIALIZED_CRITERIA
CRITERIA_NAME = "Spatial Subsetting"
CRITERIA_TYPE = "GEO"
CRITERIA_VALUE = "BY_POLYGON_LOC"
GROUP = POLYGON_LOC
TANGENT_LATITUDE = 81.8895
TANGENT_LONGITUDE = 158.423
MAP_PROJECTION_TYPE = "ORTHOGRAPHIC"
LATITUDE = (83.2017, 81.4847, 80.4686, 81.8274)
LONGITUDE = (-175.078, -176.234, 155.986, 151.309)
WG_ZOOM = 2
END_GROUP = POLYGON_LOC
END_GROUP = SPECIALIZED_CRITERIA
GROUP = SPECIALIZED_CRITERIA
```

```

    CRITERIA_NAME = "Scan Line Size"
    CRITERIA_TYPE = "INTEGER"
    CRITERIA_VALUE = 1104
    END_GROUP = SPECIALIZED_CRITERIA
    END_GROUP = SUBSET_SPEC
]

```

This group is optional, indicates the subset option goes along with this granule.

```

GROUP = PATH_ROW_LOC
    PATH = (119)
    ROW = (233)
END_GROUP = PATH_ROW_LOC
GROUP = POLYGON_LOC
    LATITUDE = (70.31, 69.6, 64.78, 65.36)
    LONGITUDE = (-80.91, -85.44, 136.97, 133.18)
    CENTROID_LAT = 81.94
    CENTROID_LON = -170.59
    POLE_INCLUDED = "X"
END_GROUP = POLYGON_LOC
END_GROUP = LINE_ITEM

```

The LINE_ITEM group could be repeated if there are more granules to be ordered in one request.

```

GROUP = MONITOR
    SESSION_ID = "cheyenne.hitc.com:24496:20020726:153027"
    TX_CLIENT = ("1027711832", "939137")
END_GROUP = MONITOR
GROUP = VERSION
    SENDER_VERSION = "imswww-3_4b_6"
    PROTOCOL_VERSION = 3.2
    IMS_STAFF = "1"
END_GROUP = VERSION
[PRIORITY = "HIGH"]

```

This line is optional. The default is LOW with the possible values being LOW, NORMAL, HIGH, VHIGH and XPRESS.

```
[USERSTRING = "TESTFOR"]
```

This line is optional. But if it is there, the length must be less than 80 characters.

```
[NOTIFY = "zyu@eos.hitc.com"]
```

This line is optional. But if it is there, the length must be less than 255 characters.

```
[DDISTNOTIFYTYPE = "MAIL"]
```

This line is optional.

```

END_GROUP = PRODUCT_REQUEST
END

```

4.8.13.1.5 ODL Template File for "FtpPush" Media Type

```

GROUP = PRODUCT_REQUEST
    MESSAGE_ID = "B1027711830"
    [REQUEST_ID = "37475:27364"]
    DATA_CENTER_ID = "ECS-TEST"
    [ECS_AUTHENTICATOR = "labuser"]
GROUP = USER_AFFILIATION
    CATEGORY = "USA"
    TYPE = "GOVERNMENT"
END_GROUP = USER_AFFILIATION
{
GROUP = CONTACT_ADDRESS
    TITLE = ""
    FIRST_NAME = "Yu"
    MIDDLE_INITIAL = ""

```

```

    LAST_NAME = "Zhongfei"
    ORGANIZATION = ""
    ADDRESS = ("abcd")
    CITY = "Landover"
    STATE = "MD"
    ZIP = ""
    COUNTRY = "UNITED STATES"
    PHONE = "301-925-1042"
    FAX = ""
    EMAIL = "zyu@eos.hitc.com"
  END_GROUP = CONTACT_ADDRESS
}
{
  GROUP = SHIPPING_ADDRESS
  TITLE = ""
  FIRST_NAME = "Yu"
  MIDDLE_INITIAL = ""
  LAST_NAME = "Zhongfei"
  ORGANIZATION = ""
  ADDRESS = ("abcd")
  CITY = "Landover"
  STATE = "MD"
  ZIP = ""
  COUNTRY = "UNITED STATES"
  PHONE = "301-925-1042"
  FAX = ""
  EMAIL = "zyu@eos.hitc.com"
  END_GROUP = SHIPPING_ADDRESS
}
{
  GROUP = BILLING_ADDRESS
  TITLE = ""
  FIRST_NAME = "Yu"
  MIDDLE_INITIAL = ""
  LAST_NAME = "Zhongfei"
  ORGANIZATION = ""
  ADDRESS = ("abcd")
  CITY = "Landover"
  STATE = "MD"
  ZIP = ""
  COUNTRY = "UNITED STATES"
  PHONE = "301-925-1042"
  FAX = ""
  EMAIL = "zyu@eos.hitc.com"
  END_GROUP = BILLING_ADDRESS
}
  GROUP = LINE_ITEM
  {DATASET_ID = "JPL-GENERATED ASTER LEVEL 1B DATA - THERMAL IR CHANNELS ONLY
V001"}
  {PACKAGE_ID = "SC:AST_L1BT.001:7644"}
  PROCESSING_OPTIONS = "Native Granule"
  {MEDIA_TYPE = "FtpPush"}
  {MEDIA_FORMAT = "FILEFORMAT"}
  EST_COST = 777.88

```

```

GROUP = ORDER_SPEC
  GROUP = SPECIALIZED_CRITERIA
    CRITERIA_NAME = "FTPHOST"
    CRITERIA_TYPE = "STRING"
    {CRITERIA_VALUE = "origin"}
  END_GROUP = SPECIALIZED_CRITERIA
  GROUP = SPECIALIZED_CRITERIA
    CRITERIA_NAME = "FTPPASSWORD"
    CRITERIA_TYPE = "STRING"
    {CRITERIA_VALUE = "Sept6A02"}
  END_GROUP = SPECIALIZED_CRITERIA
  GROUP = SPECIALIZED_CRITERIA
    CRITERIA_NAME = "FTPPUSHDEST"
    CRITERIA_TYPE = "STRING"
    {CRITERIA_VALUE = "/devdata1/DEV01/PushArea"}
  END_GROUP = SPECIALIZED_CRITERIA
  GROUP = SPECIALIZED_CRITERIA
    CRITERIA_NAME = "FTPUSER"
    CRITERIA_TYPE = "STRING"
    {CRITERIA_VALUE = "labuser"}
  END_GROUP = SPECIALIZED_CRITERIA
  GROUP = SPECIALIZED_CRITERIA
    CRITERIA_NAME = "USERSTRING"
    CRITERIA_TYPE = "STRING"
    {CRITERIA_VALUE = "ABCD"}
  END_GROUP = SPECIALIZED_CRITERIA
END_GROUP = ORDER_SPEC

```

The **ORDER_SPEC** group is designed for specifying all the FtpPush parameters, it must be there for the FtpPush Media Type.

```

GROUP = RANGE_LOC
  NORTH_LATITUDE = 10.12
  WEST_LONGITUDE = -130.12
  SOUTH_LATITUDE = -10.12
  EAST_LONGITUDE = 63.1
END_GROUP = RANGE_LOC
END_GROUP = LINE_ITEM
GROUP = MONITOR
  SESSION_ID = "cheyenne.hitc.com:24496:20020726:153027"
  TX_CLIENT = ("1027711832", "939137")
END_GROUP = MONITOR
GROUP = VERSION
  SENDER_VERSION = "imswww-3_4b_6"
  PROTOCOL_VERSION = 3.2
  IMS_STAFF = "1"
END_GROUP = VERSION
[PRIORITY = "HIGH"]
[USERSTRING = "TESTFOR"]
[NOTIFY = "zyu@eos.hitc.com"]
[DDISTNOTIFYTYPE = "MAIL"]
END_GROUP = PRODUCT_REQUEST
END

```

4.8.13.1.6 ODL Template File for "CDROM" Media Type

```

GROUP = PRODUCT_REQUEST
  MESSAGE_ID = "B1027711830"

```

```

[REQUEST_ID = "37475:27364"]
DATA_CENTER_ID = "ECS-TEST"
[ECS_AUTHENTICATOR = "labuser"]
GROUP = USER_AFFILIATION
  CATEGORY = "USA"
  TYPE = "GOVERNMENT"
END_GROUP = USER_AFFILIATION
{
GROUP = CONTACT_ADDRESS
  TITLE = ""
  FIRST_NAME = "Yu"
  MIDDLE_INITIAL = ""
  LAST_NAME = "Zhongfei"
  ORGANIZATION = ""
  ADDRESS = ("abcd")
  CITY = "Landover"
  STATE = "MD"
  ZIP = ""
  COUNTRY = "UNITED STATES"
  PHONE = "301-925-1042"
  FAX = ""
  EMAIL = "zyu@eos.hitc.com"
END_GROUP = CONTACT_ADDRESS
}
{
GROUP = SHIPPING_ADDRESS
  TITLE = ""
  FIRST_NAME = "Yu"
  MIDDLE_INITIAL = ""
  LAST_NAME = "Zhongfei"
  ORGANIZATION = ""
  ADDRESS = ("abcd")
  CITY = "Landover"
  STATE = "MD"
  ZIP = ""
  COUNTRY = "UNITED STATES"
  PHONE = "301-925-1042"
  FAX = ""
  EMAIL = "zyu@eos.hitc.com"
END_GROUP = SHIPPING_ADDRESS
}
{
GROUP = BILLING_ADDRESS
  TITLE = ""
  FIRST_NAME = "Yu"
  MIDDLE_INITIAL = ""
  LAST_NAME = "Zhongfei"
  ORGANIZATION = ""
  ADDRESS = ("abcd")
  CITY = "Landover"
  STATE = "MD"
  ZIP = ""
  COUNTRY = "UNITED STATES"
  PHONE = "301-925-1042"
}

```

```

    FAX = ""
    EMAIL = "zyu@eos.hitc.com"
    END_GROUP = BILLING_ADDRESS
}
GROUP = LINE_ITEM
{DATASET_ID = "JPL-GENERATED ASTER LEVEL 1B DATA - THERMAL IR CHANNELS ONLY
V001"}
{PACKAGE_ID = "SC:AST_L1BT.001:7644"}
PROCESSING_OPTIONS = "Native Granule"
{MEDIA_TYPE = "CDROM"}
The above line specifies the CDROM media type, and the next line specifies the matching media format.
{MEDIA_FORMAT = "RockRidge"}
EST_COST = 777.88
END_GROUP = LINE_ITEM
GROUP = MONITOR
SESSION_ID = "cheyenne.hitc.com:24496:20020726:153027"
TX_CLIENT = ("1027711832", "939137")
END_GROUP = MONITOR
GROUP = VERSION
SENDER_VERSION = "imswww-3_4b_6"
PROTOCOL_VERSION = 3.2
IMS_STAFF = "1"
END_GROUP = VERSION
[PRIORITY = "HIGH"]
[USERSTRING = "TESTFOR"]
[NOTIFY = "zyu@eos.hitc.com"]
[DDISTNOTIFYTYPE = "MAIL"]
END_GROUP = PRODUCT_REQUEST
END

```

4.8.13.1.7 ODL Template File for DataPool-Only Granules

```

GROUP = PRODUCT_REQUEST
MESSAGE_ID = "B1027711830"
[REQUEST_ID = "37475:27364"]
DATA_CENTER_ID = "ECS-TEST"
[ECS_AUTHENTICATOR = "labuser"]
[METADATA_FLAG = "Y"]

```

The above line specifies the whether the Meta Data file associated with the granule in LINE_ITEM group is ordered or not.

```

GROUP = USER_AFFILIATION
CATEGORY = "USA"
TYPE = "GOVERNMENT"
END_GROUP = USER_AFFILIATION
{
GROUP = CONTACT_ADDRESS
TITLE = ""
FIRST_NAME = "Yu"
MIDDLE_INITIAL = ""
LAST_NAME = "Zhongfei"
ORGANIZATION = ""
ADDRESS = ("abcd")
CITY = "Landover"
STATE = "MD"
ZIP = ""
COUNTRY = "UNITED STATES"
}

```

```

        PHONE = "301-925-1042"
        FAX = ""
        EMAIL = "zyu@eos.hitc.com"
    END_GROUP = CONTACT_ADDRESS
}
{
    GROUP = SHIPPING_ADDRESS
        TITLE = ""
        FIRST_NAME = "Yu"
        MIDDLE_INITIAL = ""
        LAST_NAME = "Zhongfei"
        ORGANIZATION = ""
        ADDRESS = ("abcd")
        CITY = "Landover"
        STATE = "MD"
        ZIP = ""
        COUNTRY = "UNITED STATES"
        PHONE = "301-925-1042"
        FAX = ""
        EMAIL = "zyu@eos.hitc.com"
    END_GROUP = SHIPPING_ADDRESS
}
{
    GROUP = BILLING_ADDRESS
        TITLE = ""
        FIRST_NAME = "Yu"
        MIDDLE_INITIAL = ""
        LAST_NAME = "Zhongfei"
        ORGANIZATION = ""
        ADDRESS = ("abcd")
        CITY = "Landover"
        STATE = "MD"
        ZIP = ""
        COUNTRY = "UNITED STATES"
        PHONE = "301-925-1042"
        FAX = ""
        EMAIL = "zyu@eos.hitc.com"
    END_GROUP = BILLING_ADDRESS
}
    GROUP = LINE_ITEM

```

```

        {DATASET_ID = "Non-ECS granule"}
        {PACKAGE_ID = "WB:AST_L1BT.001:7644"}

```

The above line specifies the DataPool-only granule, the granule type must be "WB"

```

        PROCESSING_OPTIONS = "Native Granule"
        {MEDIA_TYPE = "8MM"}
        {MEDIA_FORMAT = "TARFORMAT"}
        EST_COST = 777.88
    END_GROUP = LINE_ITEM
    GROUP = LINE_ITEM
        {DATASET_ID = "JPL-GENERATED ASTER LEVEL 1B DATA - THERMAL IR
CHANNELS ONLY V001"}
        {PACKAGE_ID = "DP:AST_L1BT.001:7645"}
        PROCESSING_OPTIONS = "Native Granule"
        {MEDIA_TYPE = "8MM"}
        {MEDIA_FORMAT = "TARFORMAT"}

```

```

EST_COST = 777.88
END_GROUP = LINE_ITEM
GROUP = MONITOR
SESSION_ID = "cheyenne.hitc.com:24496:20020726:153027"
TX_CLIENT = ("1027711832", "939137")
END_GROUP = MONITOR
GROUP = VERSION
SENDER_VERSION = "imswww-3_4b_6"
PROTOCOL_VERSION = 3.2
IMS_STAFF = "1"
END_GROUP = VERSION
[PRIORITY = "HIGH"]
[USERSTRING = "TESTFOR"]
[NOTIFY = "zyu@eos.hitc.com"]
[DDISTNOTIFYTYPE = "MAIL"]
END_GROUP = PRODUCT_REQUEST
END

```

4.8.13.1.8 Examples

Example 1

EcSrOmCliDriverStart <MODE> /usr/ecs/<MODE>/CUSTOM/data/OMS/template/odl.rqst 1

This means one file called **odl.rqst.0** in directory /usr/ecs/<MODE>/CUSTOM/data/OMS/template/

Example 2

EcSrOmCliDriverStart <MODE> /usr/ecs/<MODE>/CUSTOM/data/OMS/template/odl.rqst 10

This means ten files must be named, **odl.rqst.0, odl.rqst.1, odl.rqst.2, odl, rqst.3, odl.rqst.4, odl.rqst.5, odl.rqst.6, odl.rqst.7, odl.rqst.8, odl.rqst.9** in directory /usr/ecs/<MODE>/CUSTOM/data/OMS/template/

Example 3

EcSrOmCliDriverStart <MODE> /usr/ecs/<MODE>/CUSTOM/data/OMS/template/odl.rqst 3 5

There are 3 requests to be submitted with 5 seconds submission interval.

Example 4

EcSrOmCliDriverStart <MODE> /usr/ecs/<MODE>/CUSTOM/data/OMS/template/odl.rqst 3 5
10 20

There are 3 requests to be submitted with 5 seconds submission interval and 5 db retries and 20 seconds db retry interval if database is inaccessible.

4.8.13.2 Order Manager Command Line Utility Main Screen

There is no main screen for this utility. This is a command line interface only.

4.8.13.3 Required Operating Environment

The Command Line Utility runs on the Linux 2.x platforms.

4.8.13.4 Databases

Table 4.8.13-3 lists the databases, stored procedures and tables used by the Command Line utility.

Table 4.8.13-3. Order Manager Data Bases

Database	Stored Procedure	Table(s)
EcOmDB_<MODE>	OmCreateRequest	OmRequest
		OmRequestOptions
	OmInsertGranule	OmGranule
	OmInsertSubSetInfo	OmSubSettingInfo
	OmInsertAction	OmActionQueue
mss_acct_db_<MODE>		EcAcRequest
		EcAcAddress

4.8.13.5 Special Constraints

Table 4.8.13-4 lists the COTS product dependencies for the Command Line Utility.

Table 4.8.13-4. Order Manager COTS Products Dependencies

Product Dependency	Protocols Used	Comments
OMS Database	SQL	Via SQL server machine
Sybase Open Client library	Sybase client/server communication	Requires proper baseline version of Open Client library

4.8.13.6 Outputs

The Command Line Utility does not produce any reports but sends messages to the operator via a log file.

4.8.13.7 Event and Error Messages

The Command Line Utility writes information useful to the operator to a log file. The file is stored in the /usr/ecs/<MODE>/CUSTOM/logs directory and is named

EcOmSrCliDriverDebug.log and EcOmSrCliDriver.ALOG. The utility renames the debug log and ALOG files to the name with current time stamp suffixes if they already exist and create new debug log and ALOG files.

There are two types of messages written to the utility's log file: errors and informative messages. Error messages include information about program internal/external faults, unplanned disconnects with the Sybase Server, general database errors, unable to open ODL files due permission or nonexistence, and configuration file problems. Informative messages include when the utility starts and stops and progress messages. All messages are date and time stamped.

If there are syntax errors in command-line invocation, a usage message is printed to the screen.

4.8.13.8 Reports

The Command Line Utility does not produce any reports.

This page intentionally left blank.

4.8.14 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.

4.8.14.1 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 4.8.14-1 lists the options for the OMS configuration CI.

Table 4.8.14-1. 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.

4.8.14.2 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 4.8.14-1, 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.

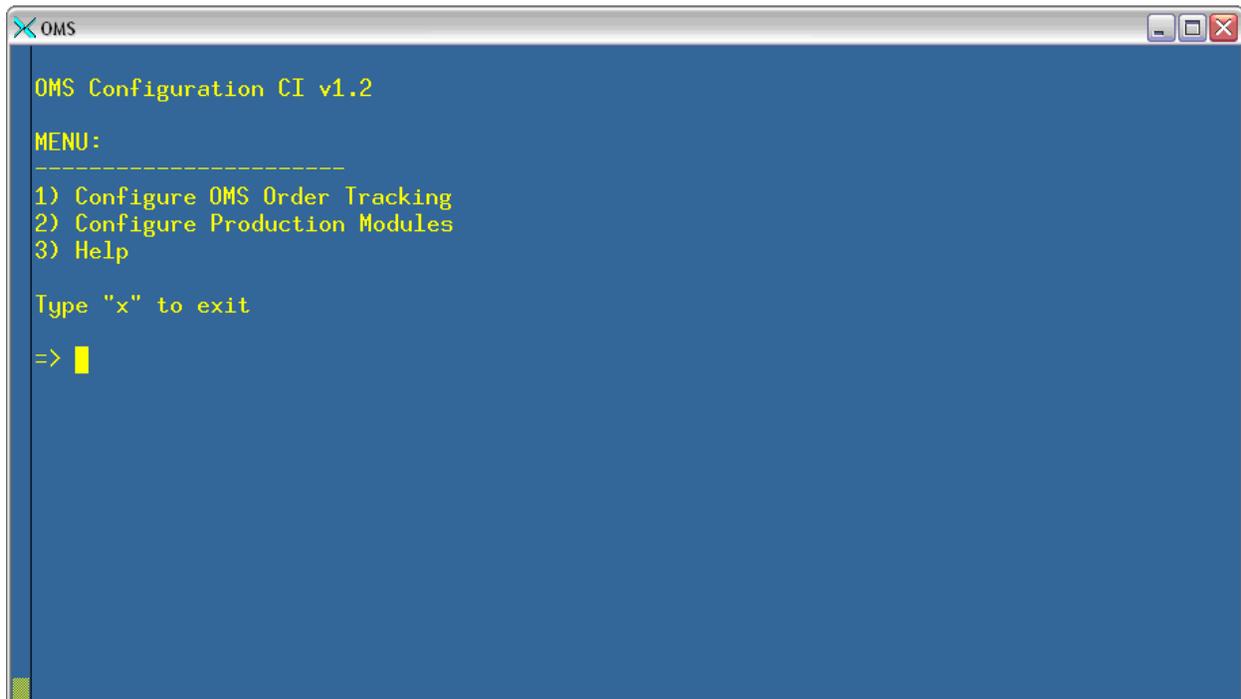
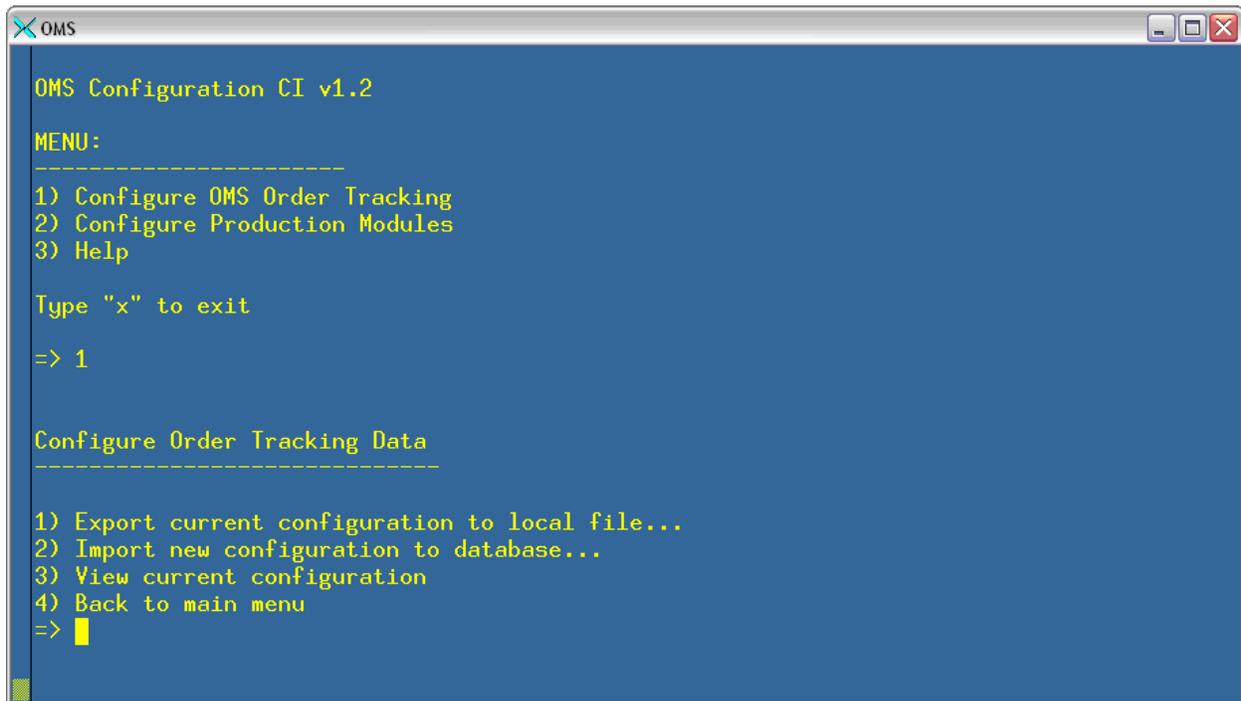


Figure 4.8.14-1. Main Menu

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 4.8.14-2 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 4.8.14-2. 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 4.8.14-3.

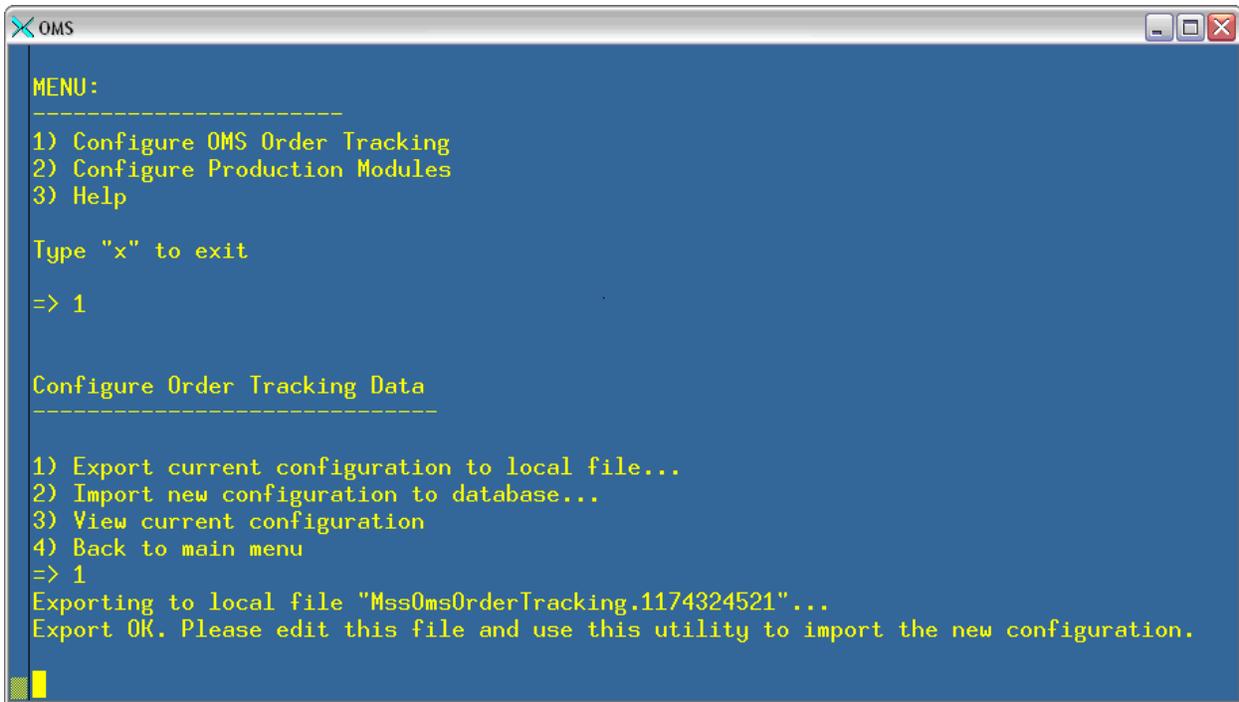


Figure 4.8.14-3. Configure Order Tracking Data Menu Screen (2 of 3)

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 4.8.14-2 below for the order source mappings.

Table 4.8.14-2. 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 4.8.14-4:

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.

```
OMS
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

=> 2
You are about to import an edited configuration file. Please make sure the fields
are properly edited. These changes will be submitted to the OMS database.

"Mss0msOrderTracking.1174324521" was specified as the import file. Do you want to use this
one? [y/n]
```

Figure 4.8.14-4. 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 `-odi <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

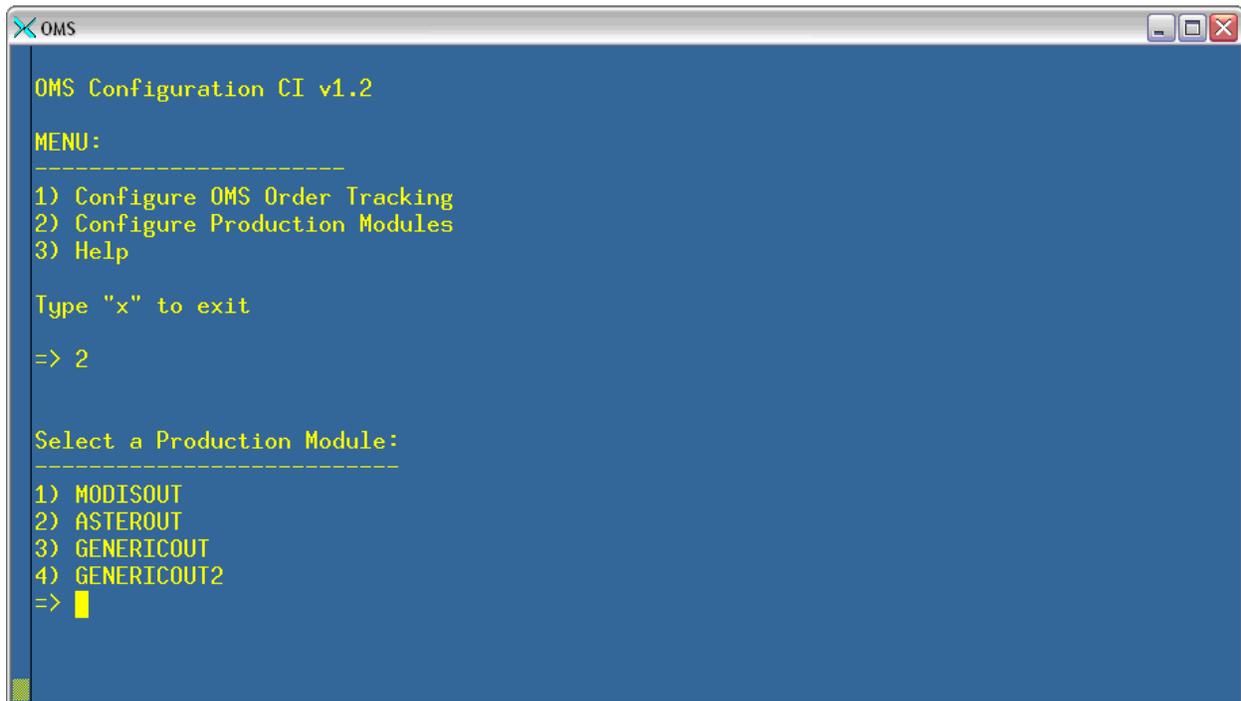
GDAS_OZF.002

.

.

.
```

Then, choose “Configure Production Modules” from the main menu, and the following submenu as shown in Figure 4.8.14-5 appears:



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

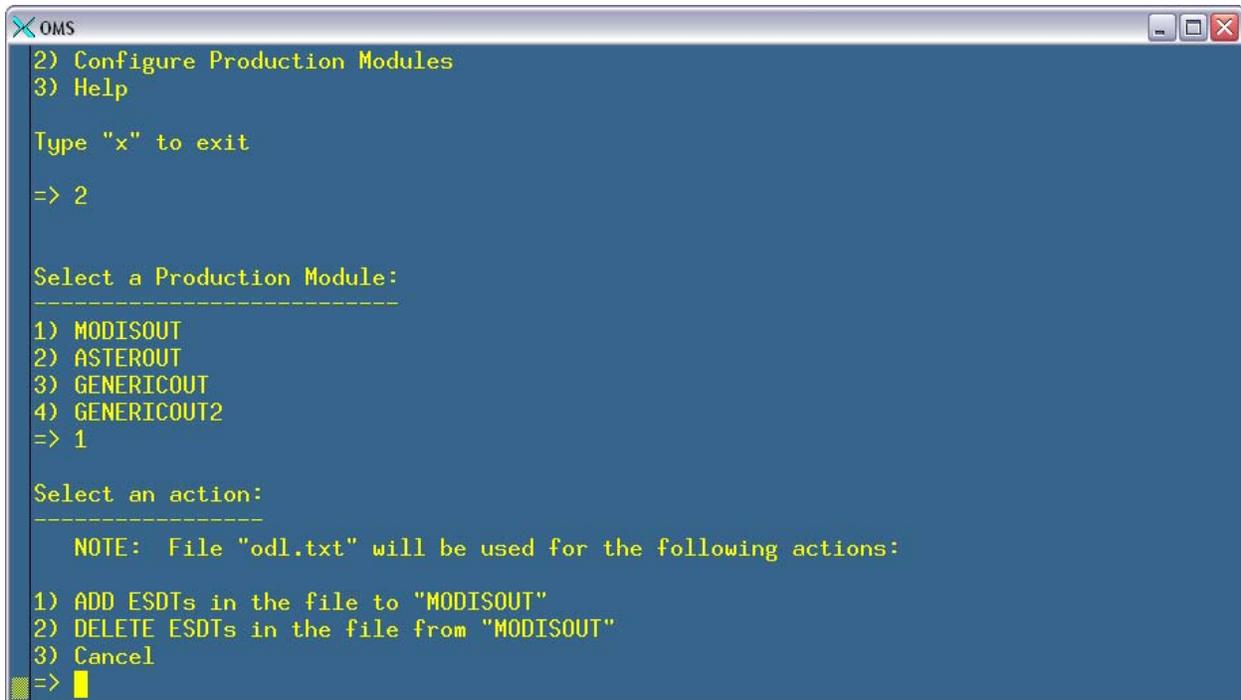
Type "x" to exit

=> 2

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

Figure 4.8.14-5. 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 4.8.14-6 appears:



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

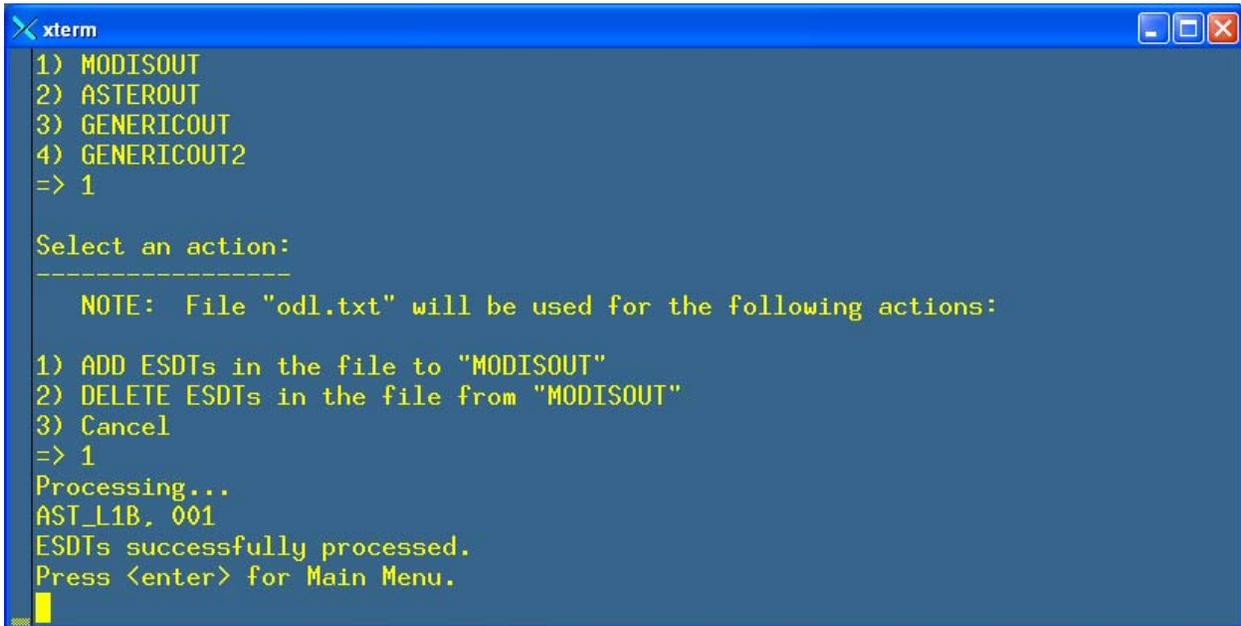
Type "x" to exit
=> 2

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 4.8.14-6. Configure Production Module (2 of 4)

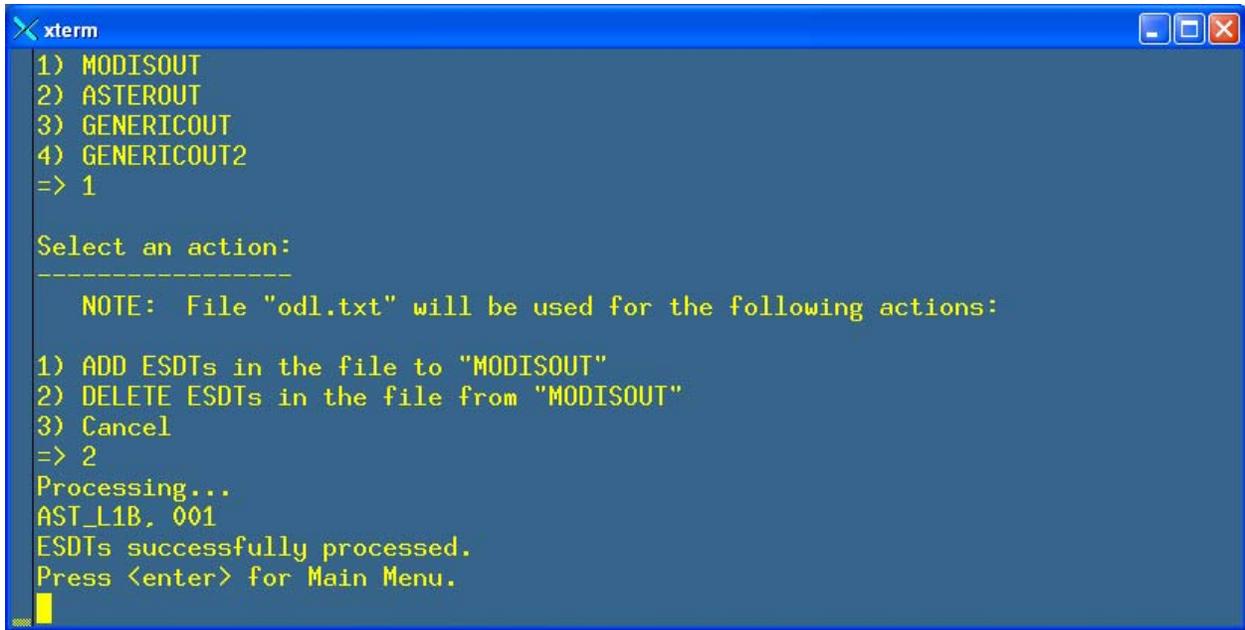
Select to either ADD (Figure 4.8.14-7) or DELETE (Figure 4.8.14-8) the ESDT Collections found in the input file to the selected Production Module. One of the following screens will appear:



```
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 4.8.14-7. 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 4.8.14-8. 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 4.8.14-9 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 4.8.14-9. Help Screen

4.8.14.3 Required Operating Environment

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

The O/S requirements are Linux 2.x.

4.8.14.4 Interfaces and Data types

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

4.8.14.5 Databases

The OMS Configuration CI accesses the OMS database.

4.8.14.6 Special Constraints

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

4.8.14.7 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.

4.8.14.8 Events and Messages

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

4.8.14.9 Reports

The Configuration CI does not generate reports.

4.8.15 Bulk URL Utility

No longer applicable. The Bulk URL functionality is now part of the Bulk Metadata Generation Tool (BMGT) and is covered in section 4.8.9.

This page intentionally left blank.

4.8.16 OmPdCleanup GUI

This Tool Command Language (Tcl) GUI cleans up or archives files that are no longer needed for reference, reuse, or troubleshooting. These include:

- EcOmPdModule debug logs
- order summary files
- tape label files
- QC reports and logs.

These files can be removed or archived to different directories at intervals using the OmPdCleanup GUI. Other artifacts from Physical Media production are cleaned up regularly after each request is shipped or canceled by the Order Manager Server in a call to OmPdCleanup.pl. OmPdCleanup.pl, which runs on the physical media box, cleans up the staging directory, the ISO image file, and the files used to build the jewel case insert file.

4.8.16.1 OmPdCleanup GUI Location

The OmPdCleanup GUI is located under /usr/ecs/\$MODE/CUSTOM/utilities

The location of the archive directory is configurable in /usr/ecs/\$MODE/CUSTOM/cfg/EcOmPdModule.CFG

Table 4.8.16-1 describes the common ECS operator functions performed through the OmPd Cleanup Manager.

Table 4.8.16-1. OmPd Cleanup Manager Common Operator Functions (1 of 2)

Operating Function	GUI Controls	Description
Remove or Archive DebugLog files	Radio button & edit field	Delete or archive the individual debug logs left by EcOmPdModule. The location of these files is configurable in /usr/ecs/\$MODE/CUSTOM/cfg/EcOmPdModule.CFG but are usually found in /usr/ecs/MODE/CUSTOM/logs on the mdl01 box. The names of these files are Request_Volume.log
Remove or Archive the Summary files	Radio button & edit field	Delete or archive the summary files. The location of these files is in <PMD_Root_DIR>/summary
Remove or Archive the Label Directory files	Radio button & edit field	Delete or archive the Jewel Case Insert files, tape label files, and any other files left under <PMD_Root_DIR>/label from an incomplete request cleanup
Remove or Archive the QC Reports and the QC logs	Radio button & edit field	Delete or archive the QC log, the QC Summary file, and any of the individual QC Reports. These are found under: <PMD_Root_DIR>/QC

Table 4.8.16-1. OmPd Cleanup Manager Common Operator Functions (2 of 2)

Operating Function	GUI Controls	Description
Remove staged files	Checkbox & edit field	This removes any files under <PMD_Root_DIR>/PM. This should only be left over staging directories and the OmPdCleanup.pl log
Setting the cleanup time.	Radio button & edit field	Sets the daily cleanup time for the crontab.

4.8.16.2 OmPdCleanup GUI Startup

1. Make sure that a crontab for cmshared exists on the mdl01 box – the application will give an error if the crontab file does not exist.
2. If running for the first time remove ~/.pdscleanup if it exists.
3. setenv \$MODE
4. Make sure that your DISPLAY variable is set.
5. From the command line run: /usr/ecs/\$MODE/CUSTOM/utilities/OmPdCleanupGUI
pdscleanup

When the OmPdCleanup GUI starts up it reads ~/.pdscleanup. If this file exists and belongs to an earlier version the GUI may complain and fail to come up. If this is the case, delete the existing ~/.pdscleanup.

Make sure there is an existing crontab file. It will fail on exit if the crontab file does not already exist. If this is the case just run crontab -e and enter some commented text. Then rerun OmPdCleanup GUI.



Figure 4.8.16-1. OmPdCleanup GUI Dialog Box

Running the GUI creates 3 files:

- ~/.pdscleanup
 - Startup options
- /usr/ecs/\$MODE/CUSTOM/utilities/cleanup.sh
 - Contains the ksh commands to move or delete files over a certain age
- crontab

4.8.16.3 Trouble Shooting

Set environment variables DISPLAY and MODE

When it starts up it reads ~/.pdscleanup. If this file exists and belongs to an earlier version the GUI may complain and fail to come up. If this is the case, delete the existing ~/.pdscleanup.

It will fail on exit if the crontab file does not already exist. If this is the case just run crontab -e and enter some commented text. Then rerun OmPdCleanup GUI.

4.8.16.4 OmPdCleanup GUI Main Screen

The OmPdCleanup GUI dialog box (Figure 4.8.16-1) contains options to delete or archive debug logs, summary files, label files and QC artifacts. It also allows for the deletion of stale staging directories resulting from incomplete runs. Lastly the dialog box has fields to direct the timing of the cron.

4.8.16.5 Required Operating Environment

The following environment is required for the OmPdCleanup GUI to work properly:

- The O/S requirement is the Linux platform.

4.8.16.6 Databases

The OmPdCleanup GUI does not access any databases.

4.8.16.7 Special Constraints

There are no special constraints to running the OmPdCleanup GUI.

4.8.16.8 Outputs

There are no outputs from the OmPdCleanup GUI except for the error messages.

4.8.16.9 Event and Error Messages

The OmPdCleanup GUI writes no status and error messages.

4.8.16.10 Reports

The OmPdCleanup GUI does not generate reports.

4.8.17 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, a Perl module containing database connection functionality, 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.

4.8.17.1 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 4.8.17-1 describes the OmSCLI command line parameters.

Table 4.8.17-1. 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 GeoID format or LocalGranuleID format. The file may contain a mix of 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 FTPPush distribution requests. If present, indicates that the FTPpassword passed in is encrypted and needs to be decrypted by the OmSCLI.

4.8.17.2 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.

Electronic Media Selection:

DDISTMEDIATYPE = <FtpPush>|<FtpPull>|<scp> (required)
DDISTMEDIAFMT = FILEFORMAT (required)\

FTP Push:

FTPUSER = <FTP user id> (required for FtpPush distributions)
FTPPASSWORD = <FTP Password> (required for FtpPush distributions; may be encrypted)
FTPHOST = <fully qualified FTP host name> (required for FtpPush distributions)
FTPPUSHDEST = <full path name of push directory on FTPHOST> (required for FtpPush)

User Profile:

ECSUSERPROFILE = <user id associated with the request; this id is not validated> (required)

Request Priority:

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

E-Mail notification:

DDISTNOTIFYTYPE = MAIL (required)
NOTIFY=<email address> (required)

Request identification:

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>.

-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 either GeoID or LocalGranuleId format. The file may contain a mix of GeoIDs and LocalGranuleIds. The following are example granule identifiers:

"SC:MOD02HKM.002:2020633145" (a geoID)

"MYD29P1N.A2007266.h10v08.005.2007267221028.hdf" (a Local Granule Id)

Note: 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.

4.8.17.3 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 4.8.17-2 describes its contents:

Table 4.8.17-2. 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
SDSRV_SYBASE_SERVER	Name of OMS 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
SDSRV_DB_MAX_JOINS	Maximum number of database join operations
DSSSrEnv_DB	DBUSERNAME DBPASSWDSEED DBNAME DBMAXRESULTS SYBINTERFACES SDSRV_SYBASE_SERVER SDSRV_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

4.8.17.4 Required Operating Environment

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

4.8.17.5 Interfaces and Data

Table 4.8.17-3 lists the supporting products this tool depends upon to function properly.

Table 4.8.17-3. Interface Protocols

Product Dependency	Protocols Used	Comments
OMS Database	SQL	Via SQL server machine
Sybase Open Client library	Perl scripts are using the Database Interface (DBI)	
Perl	Perl scripts	

4.8.17.6 Databases

Table 4.8.17-4 lists the databases stored procedures and tables used by the command line utility.

Table 4.8.17-4. 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.

4.8.17.7 Special Constraints

The acquire wrapper script and the EcOmDCommon.pm Perl module 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.

4.8.17.8 Outputs

The OmSCLI writes processing status messages to a single application log (SCLI.log) in the logs directory of the mode. While processing, the OmSCLI writes to temporary data files and temporary log files so that a process ID can be appended to them. This enables concurrent

processing with the OmSCLI. At the end of an OmSCLI request, all temporary files are deleted, and the SCLI.log in the logs directory is appended with the final submittal status.

4.8.17.9 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.

4.8.17.10 Reports

None.

This page intentionally left blank.

4.8.18 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.

4.8.18.1 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 4.8.18-1. 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.

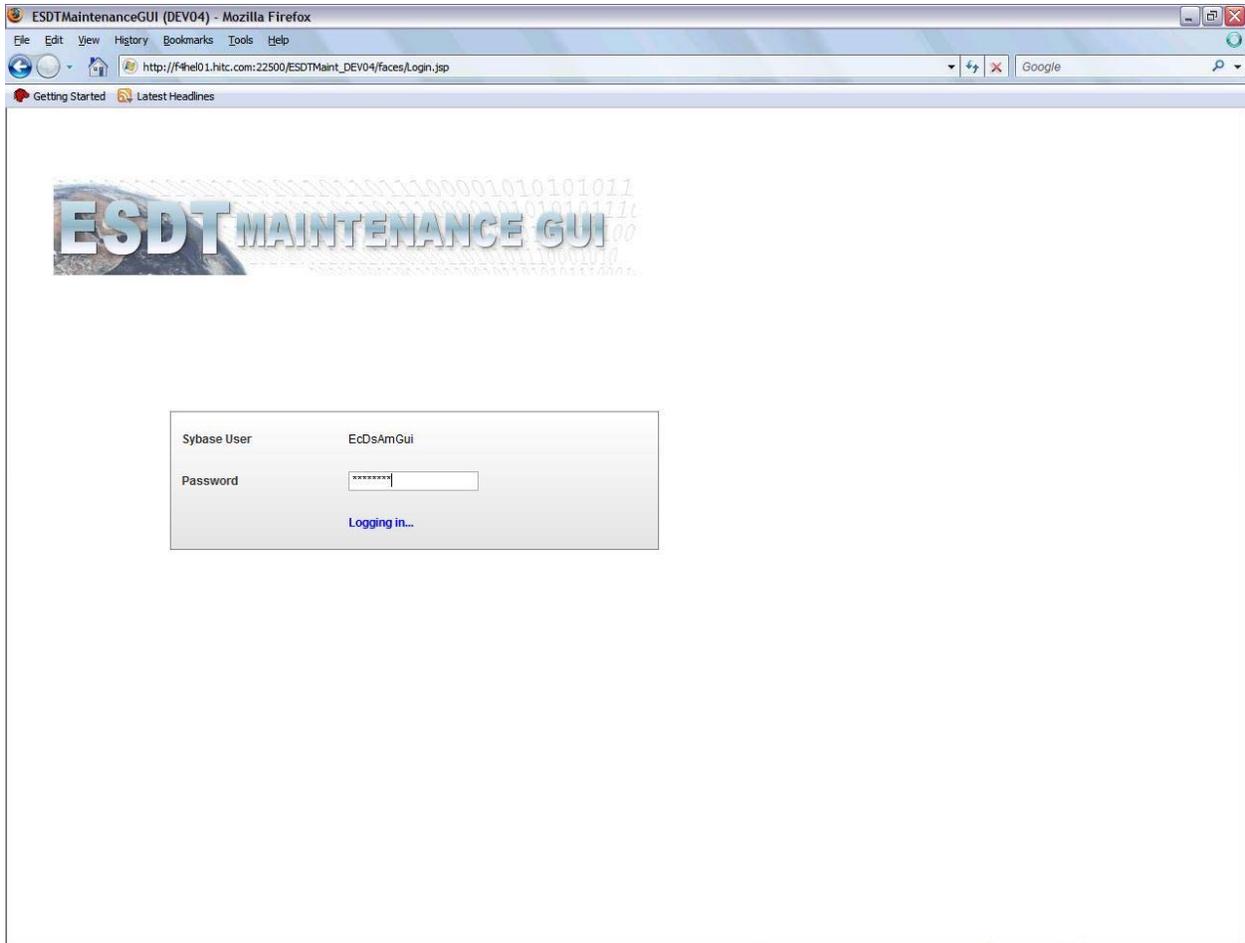


Figure 4.8.18-1. Login Page

4.8.18.2 List ESDT Page

This page first appears when the operator is logged in. The page lists the currently installed ESDTs, as shown in Figure 4.8.18-2.

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

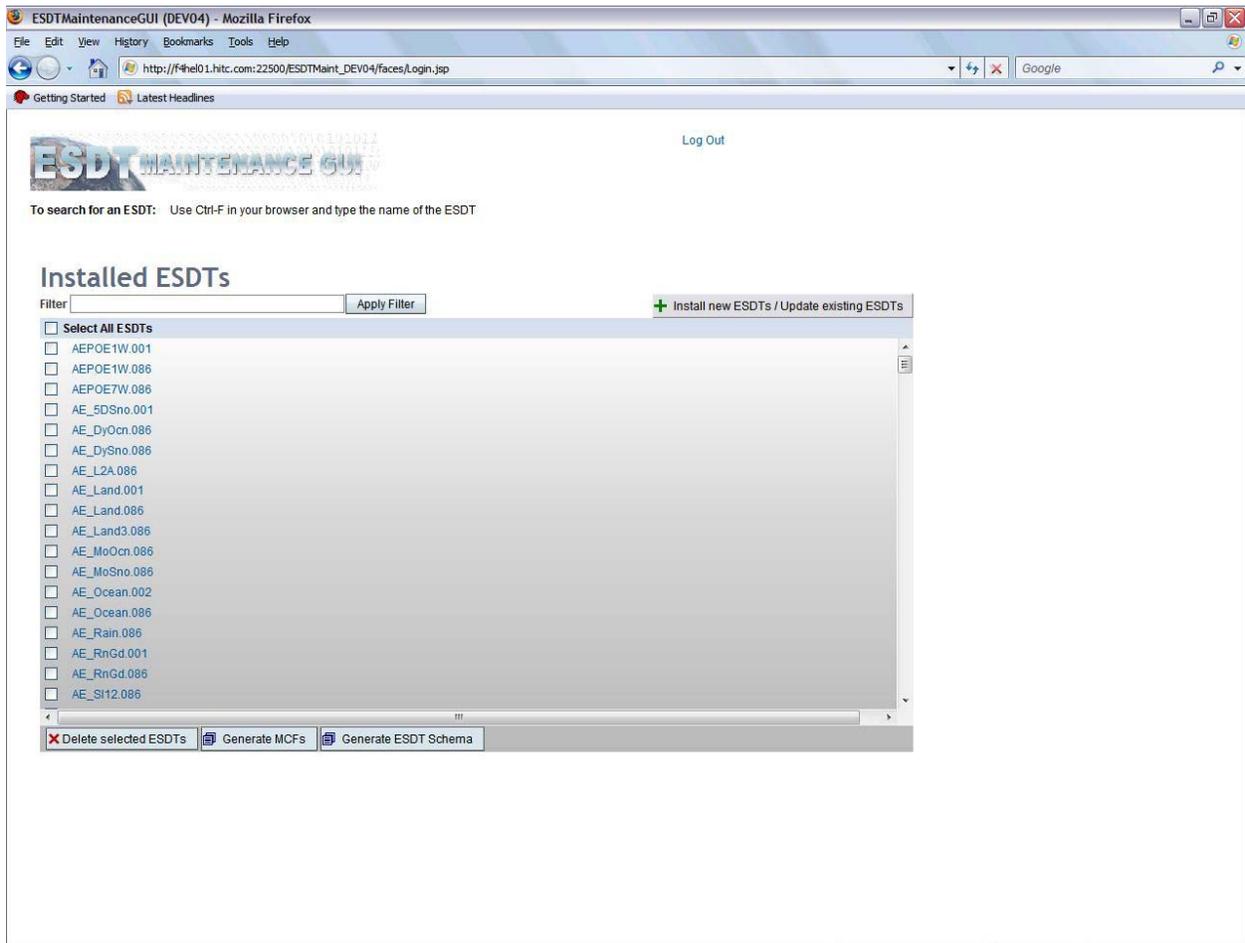


Figure 4.8.18-2. List ESDT Page

4.8.18.2.1 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”.

4.8.18.2.2 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 4.8.18-3, “GLA” would return any ESDT with the “GLA” anywhere in the short name. The search is also case-sensitive.

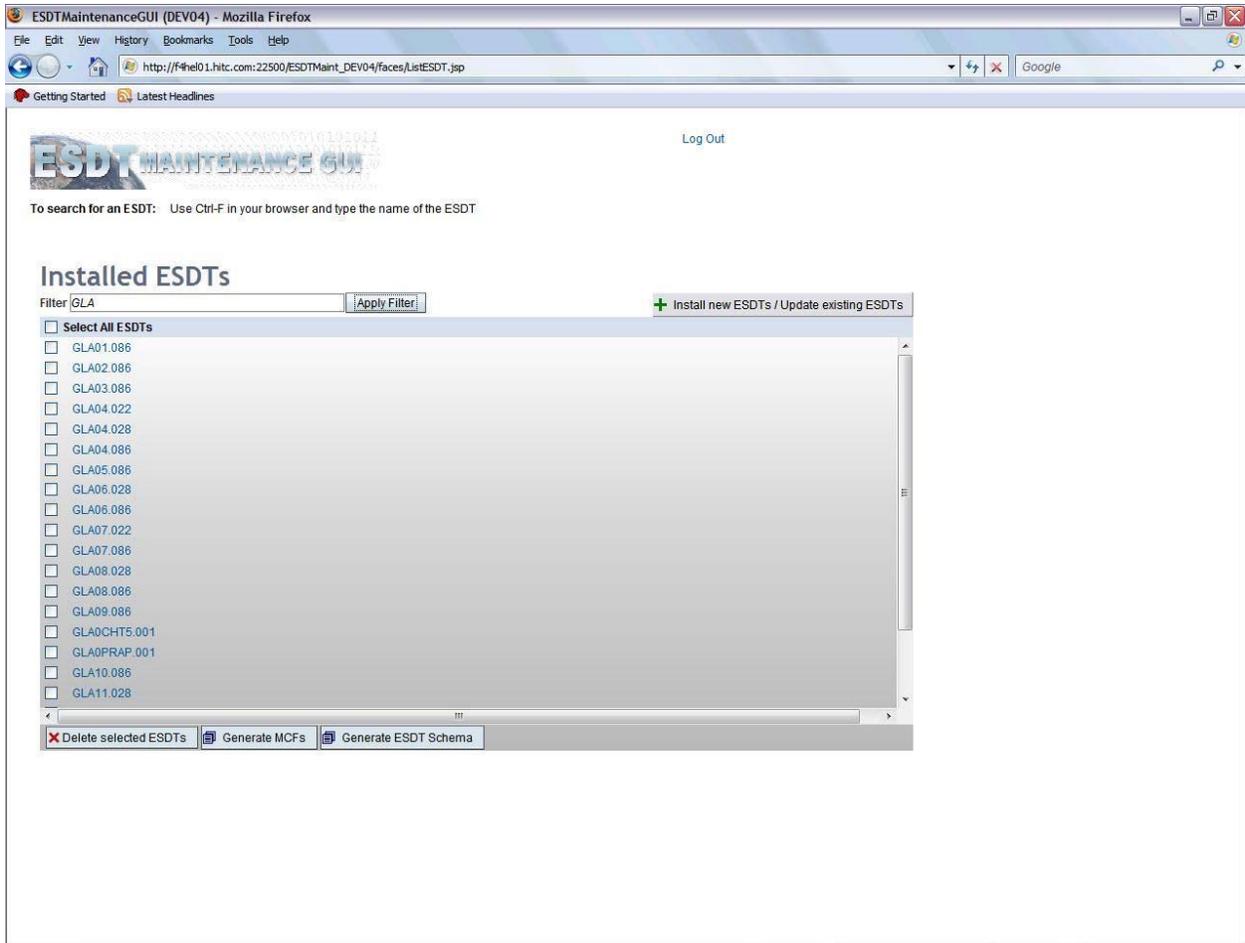


Figure 4.8.18-3. Filtering the List ESDT Page

4.8.18.2.3 View Descriptor Information for a Specific ESDT

From the List ESDT Page, click an ESDT link as shown in Figure 4.8.18-4 to view the ODL and XML descriptor information for a specific ESDT, and this will direct you to the ESDT Detail Page (see section 4.8.18.3 for more details).

4.8.18.2.4 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 4.8.18-5.

Installed ESDTs

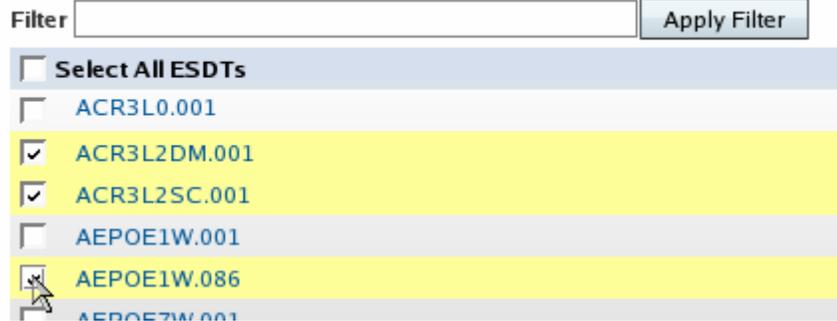


Figure 4.8.18-4. Selecting a list of ESDTs

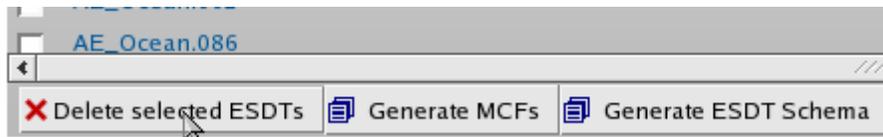


Figure 4.8.18-5. Delete selected ESDTs button

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 4.8.18-6. Depending on the result, one to three possible result tables will be displayed as listed in Table 4.8.18-1.

Table 4.8.18-1. 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.

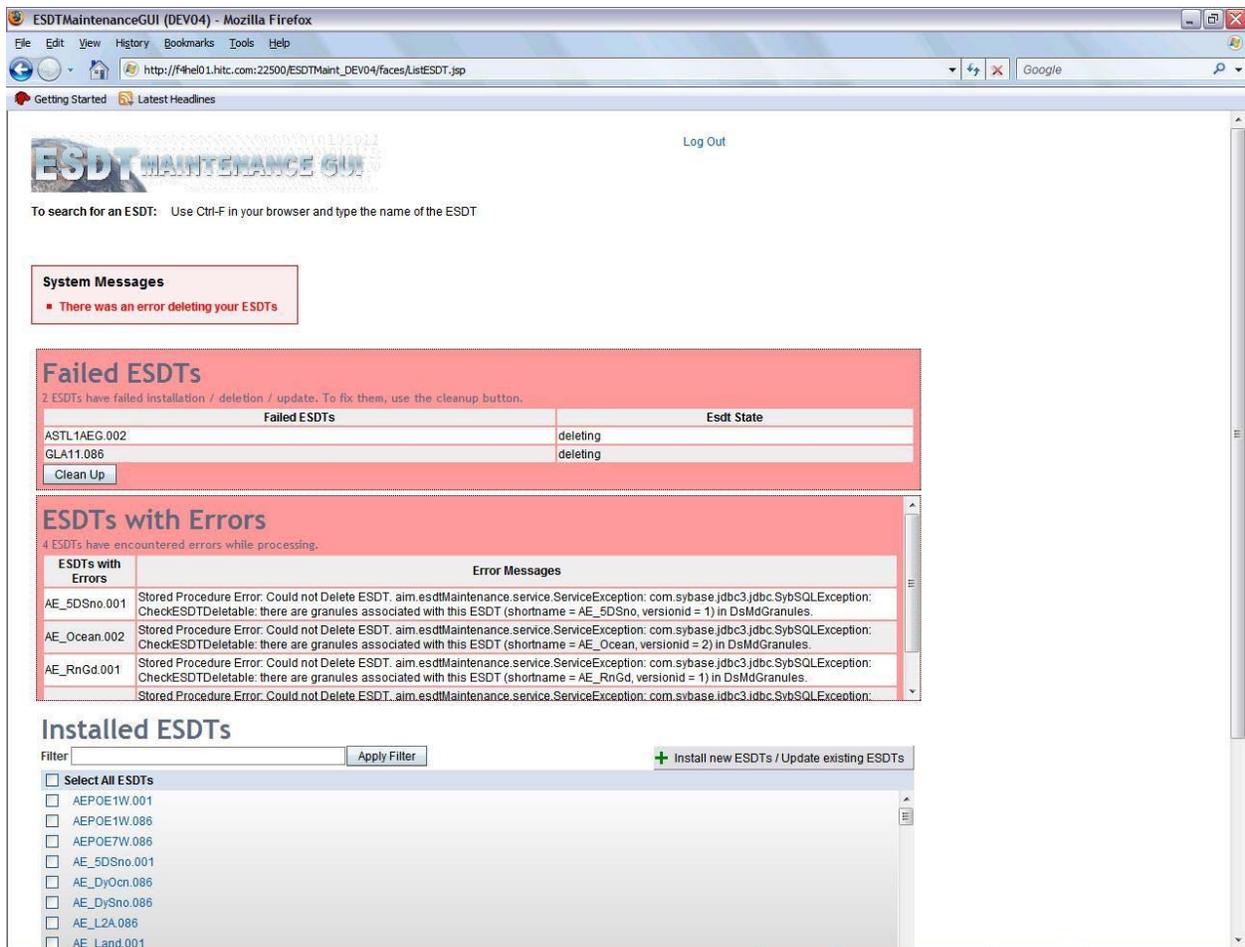


Figure 4.8.18-6. Delete ESDT Result Page

4.8.18.2.5 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 4.8.18-7.

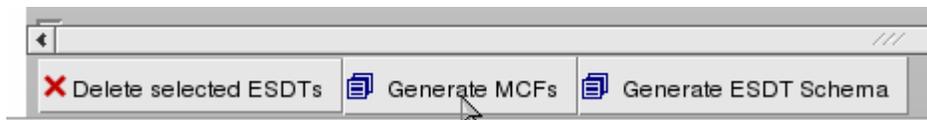


Figure 4.8.18-7. Generate MCF for Selected ESDT Button

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 4.8.18-8.

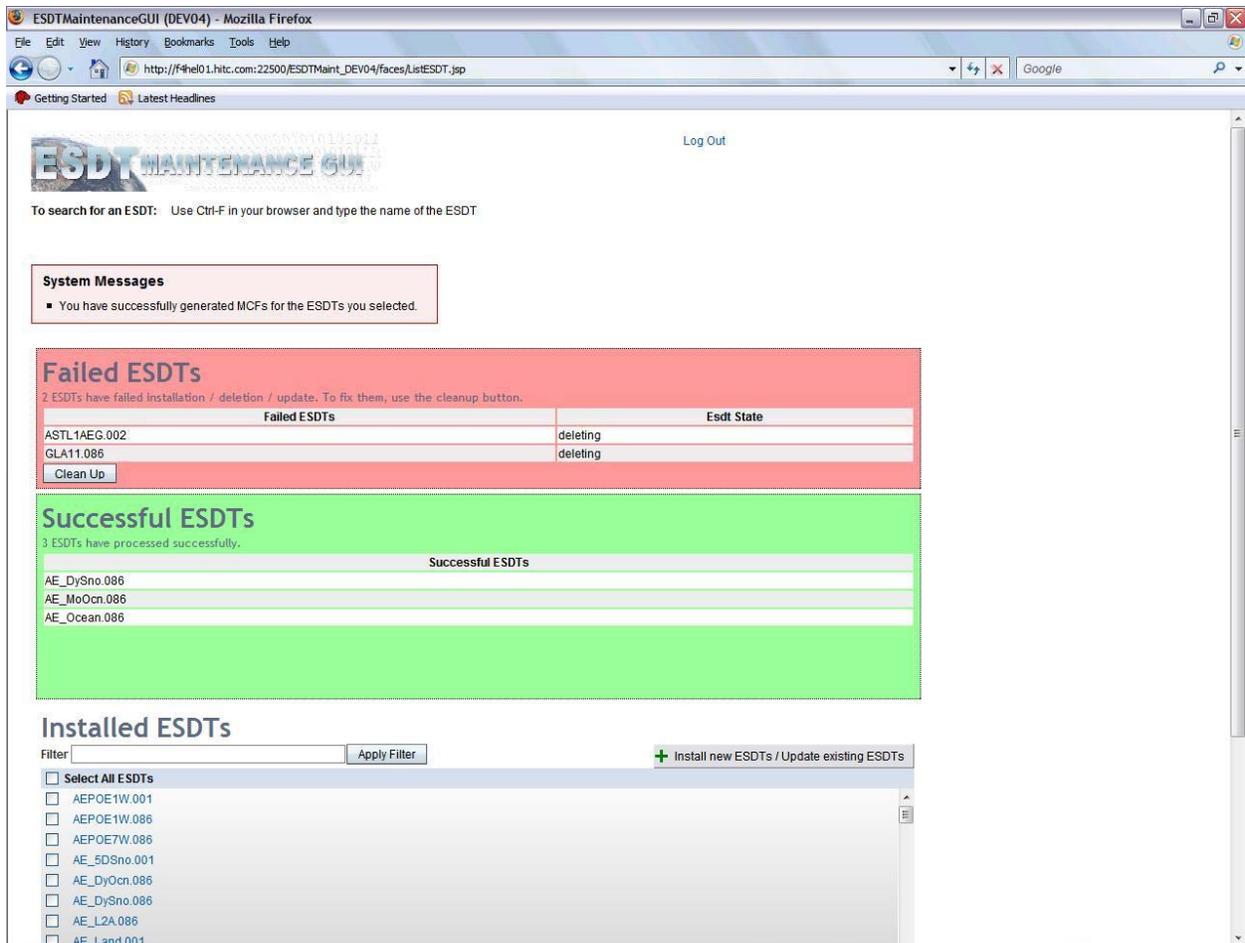


Figure 4.8.18-8. Generate MCF Result Page

4.8.18.2.6 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 4.8.18-9.

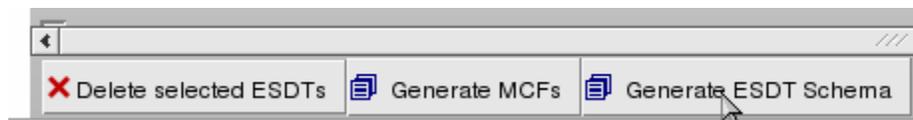


Figure 4.8.18-9. Generate Schema for Selected ESDT Button

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 4.8.18-10.

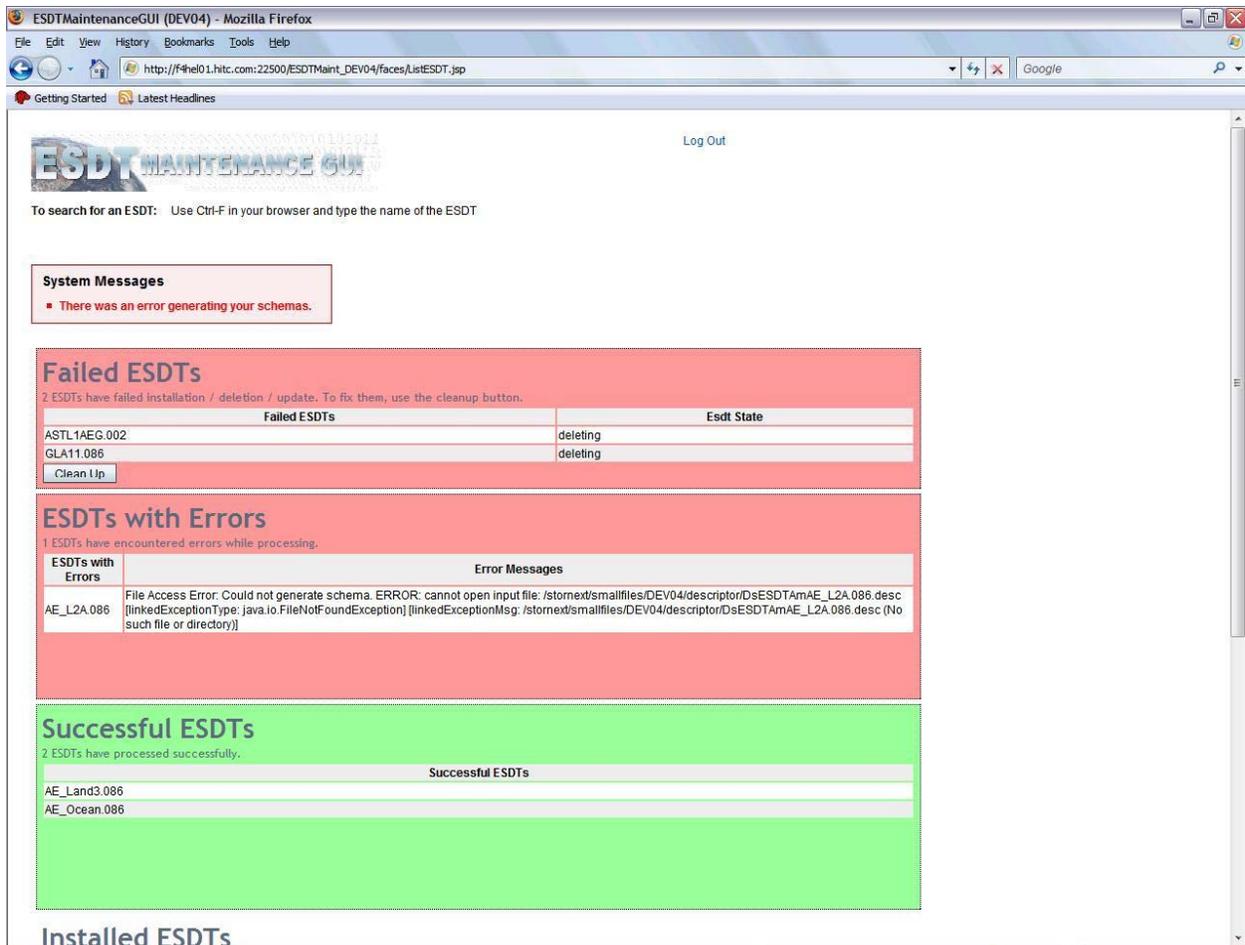


Figure 4.8.18-10. Generate Schema Result Page

4.8.18.3 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 4.8.18-11 and 4.8.18-12. 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: /stornext/smallfiles/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">
    <DLLName DATA_LOCATION="MCF" MANDATORY="TRUE" NUM_VAL="1">libDesSDTSyBASIC.0019
    <GranuleTimeDuration DATA_LOCATION="MCF" MANDATORY="FALSE" NUM_VAL="1">90000</C
    <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</F
      <ProcessingCenter DATA_LOCATION="MCF" MANDATORY="TRUE" NUM_VAL="1">AMSR-E 51
      <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</C
      <MaintenanceandUpdateFrequency DATA_LOCATION="MCF" MANDATORY="TRUE" NUM_VAL=

```

Figure 4.8.18-11. ESDT Detail Page (view XML)

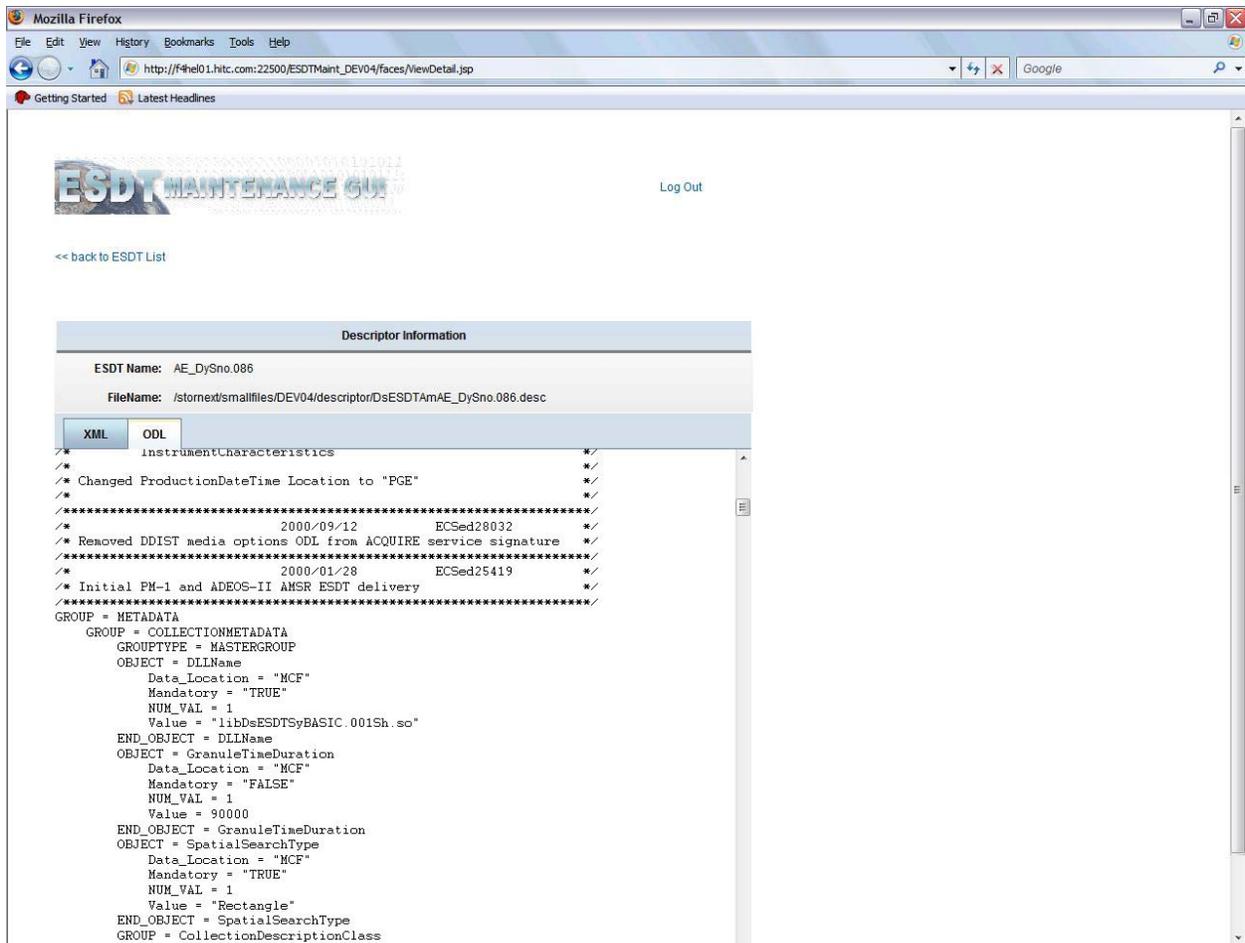


Figure 4.8.18-12. ESDT Detail Page (view ODL)

4.8.18.4 Install ESDT Page

This Install ESDT Page, as shown in Figure 4.8.18-13, 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.

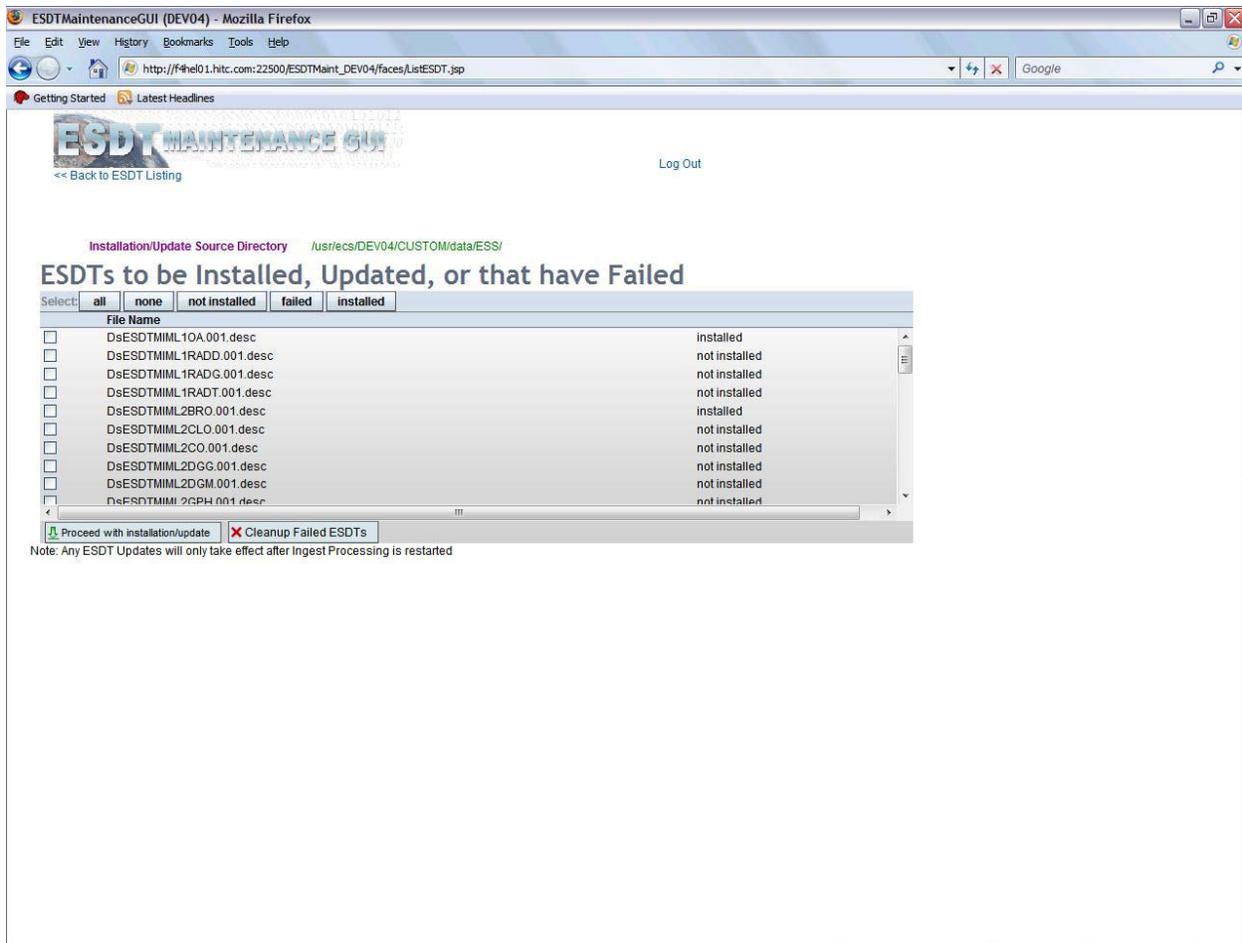


Figure 4.8.18-13. Install ESDT Page

4.8.18.4.1 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 4.8.18-14. The ESDT selected must be in the installed or not installed state.

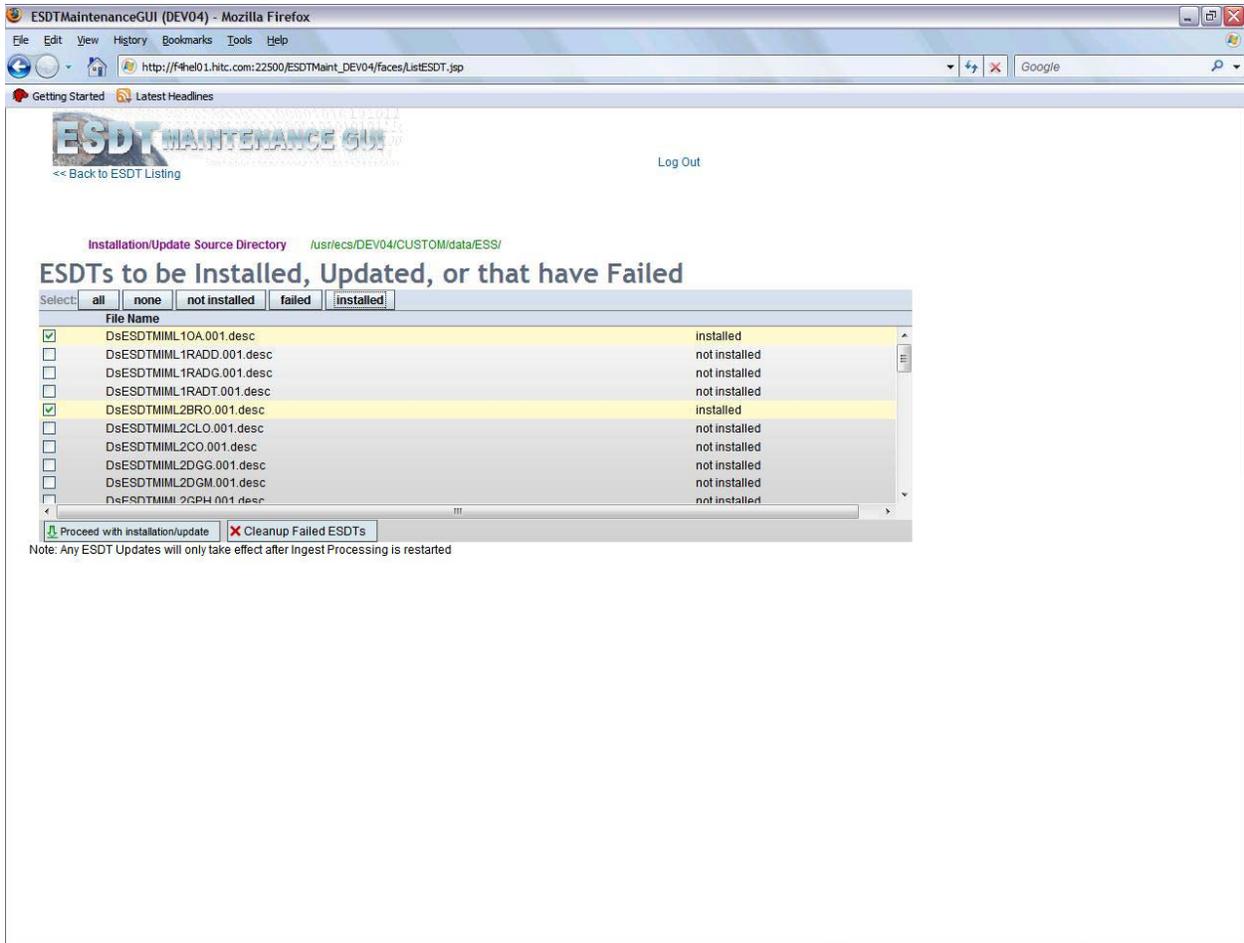


Figure 4.8.18-14. Install/Update ESDTs

Click on the “Proceed with installation/update” button to perform the installation or update action on the selected ESDTs as shown in Figure 4.8.18-15.

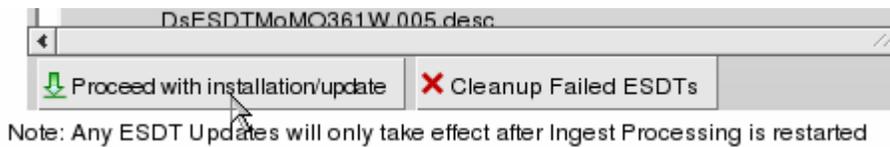


Figure 4.8.18-15. Install/Update ESDT button

An example of the result of the installation and update action is displayed in Figure 4.8.18-16.

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.

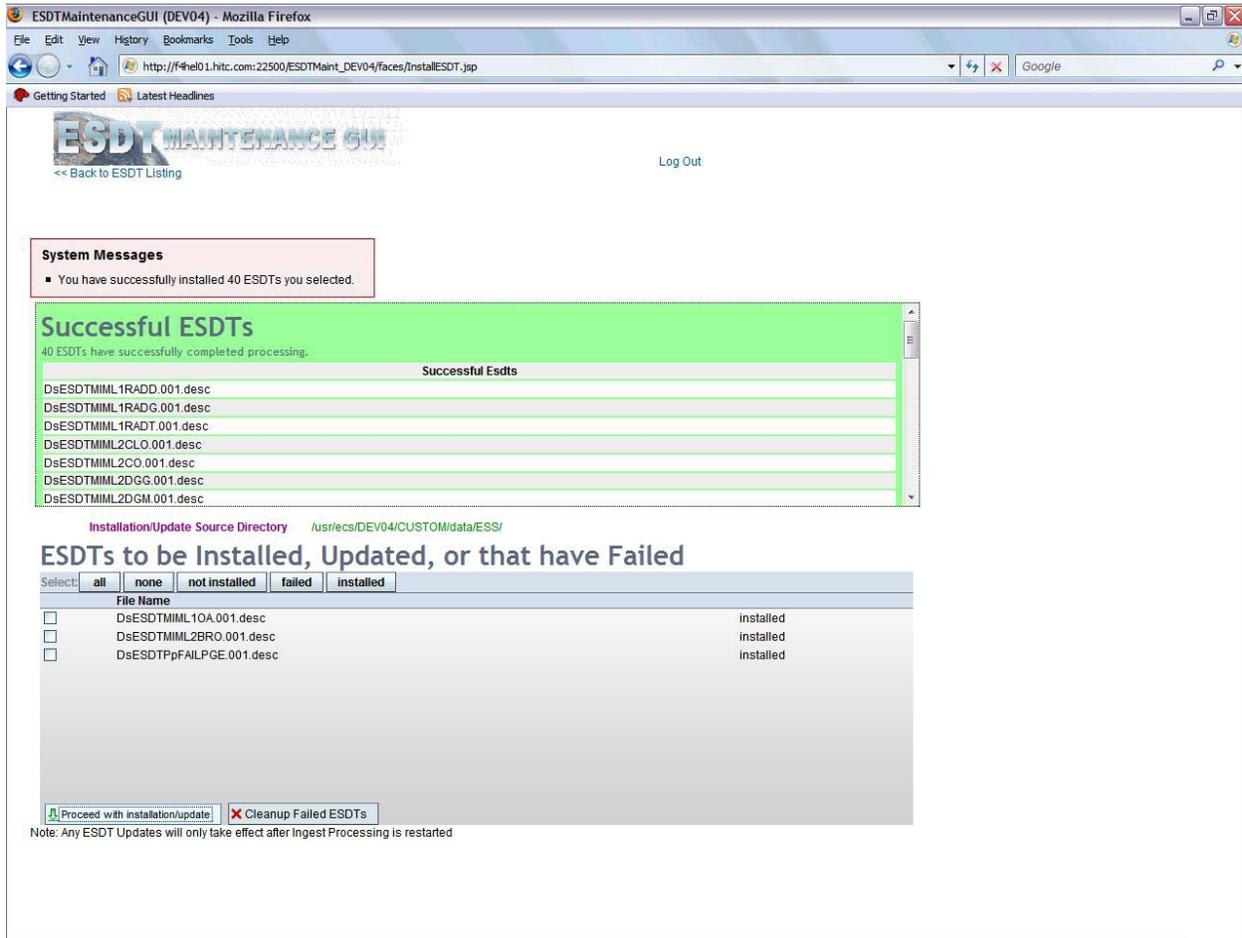


Figure 4.8.18-16. Install/Update ESDTs Result Page

4.8.18.4.2 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 4.8.18-17.

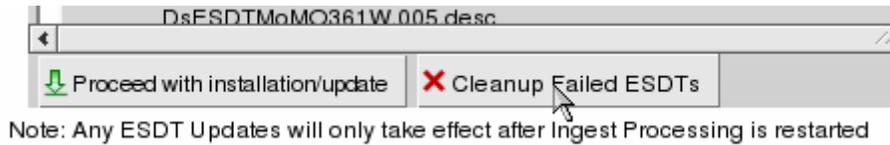


Figure 4.8.18-17. Cleanup Failed ESDTs button

An example of the result of the cleanup failed ESDTs action displayed in Figure 4.8.18-18.

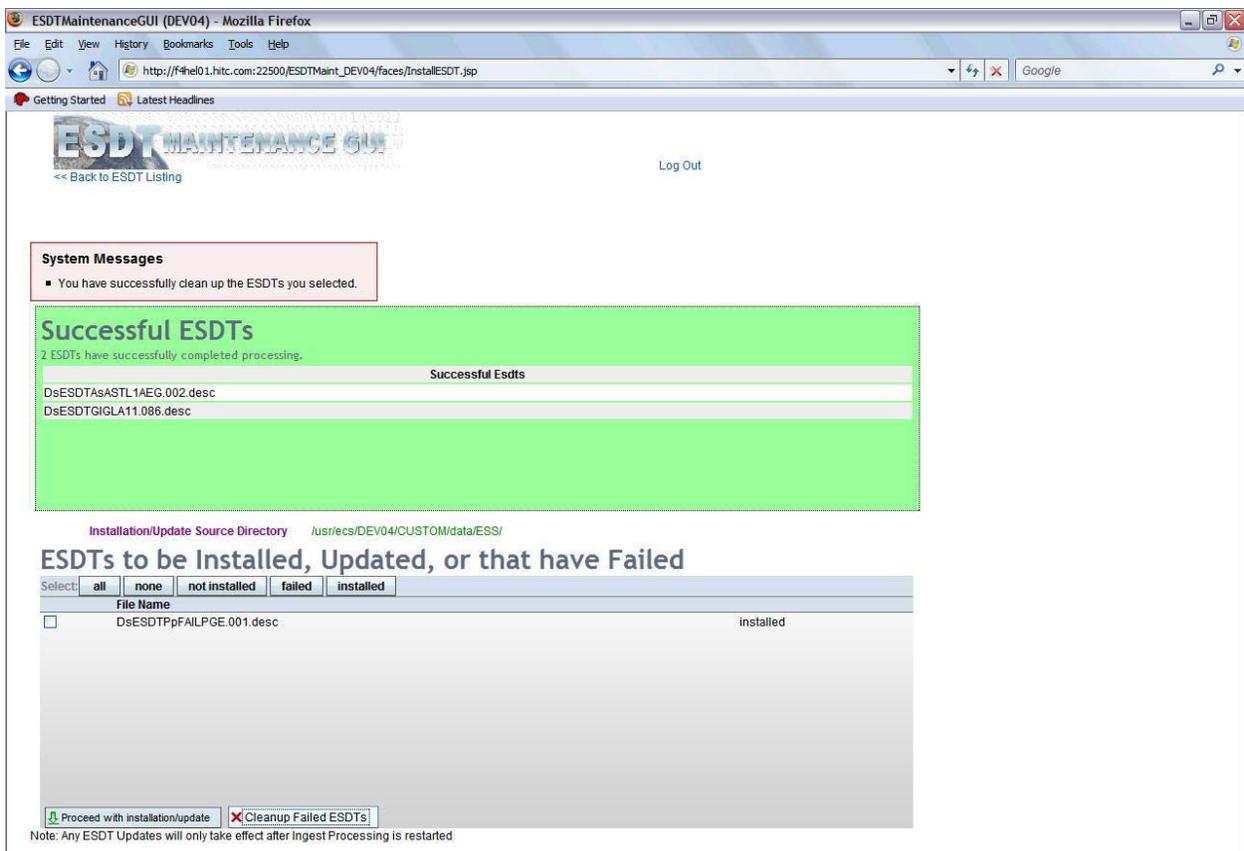


Figure 4.8.18-18. Cleanup Failed ESDTs Result Page

4.8.18.5 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.

4.8.19 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.

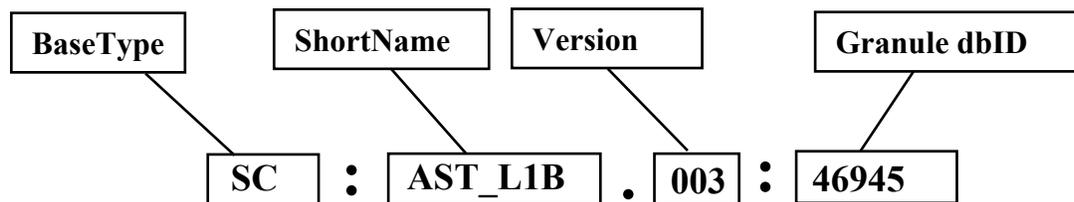
The sections below describe in detail the use of the EcDsBulkSearch, EcDsBulkDelete, EcDsBulkUndelete, and EcDsDeletionCleanup utilities.

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>

4.8.19.1 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

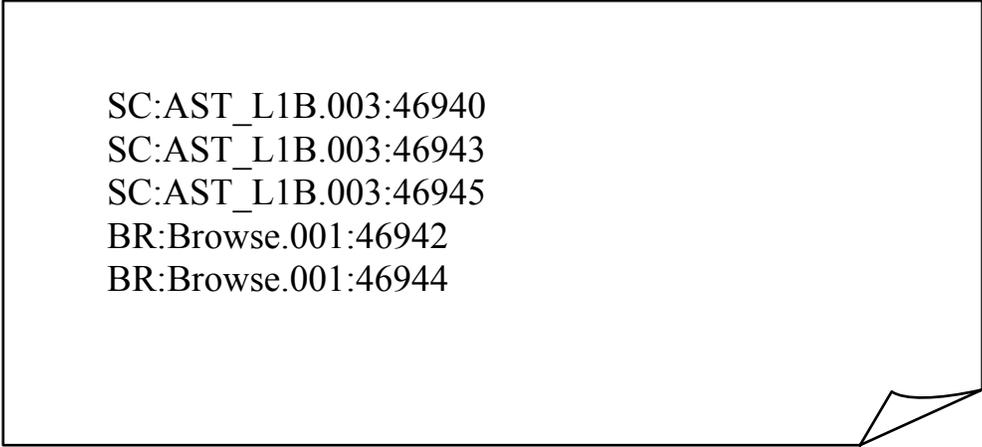
ShortName represents the data type of the granule such as “AST_L1B”.

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.

4.8.19.1.1 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 4.8.19-1 provides a description of the command line parameters for the EcDsBulkSearch utility.

Table 4.8.19-1. 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 4.8.19-1. 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 xxdbi01_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.

4.8.19.1.2 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>
```

4.8.19.1.3 Search a Limited Number of Granules

The search in 4.8.19.1.2 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 4.8.19.1.1 and returns the first n granules. The following search is the same as 4.8.19.1.2 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>
```

4.8.19.1.4 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>
```

4.8.19.1.5 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>
```

4.8.19.1.6 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>
```

4.8.19.1.7 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** ($\text{EndingDateTime} \leq \text{enddate}$). 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>
```

4.8.19.1.8 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** ($\text{insertbegin} \leq \text{insertTime} \leq \text{insertend}$). 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>
-insertbegin <mm/dd/yyyy> [<hh:mm:ss>]
-insertend <mm/dd/yyyy> [<hh:mm:ss>]
-user <db_userid>
```

4.8.19.1.9 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>
```

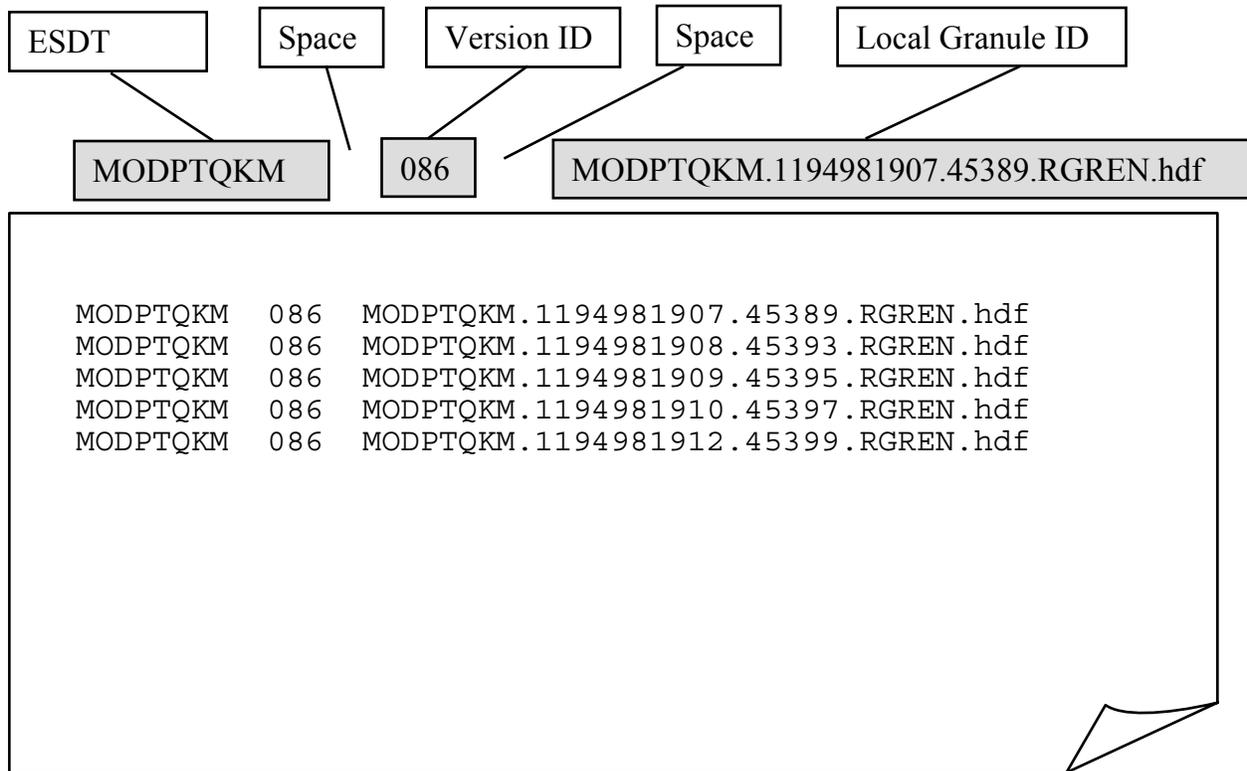
4.8.19.1.10 Search a Set of Granules for Which the insertTime is Less than or Equal to the Specified Date Time

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>
```

4.8.19.1.11 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>

```

4.8.19.1.12 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 4.8.19.2.1), 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>

```

4.8.19.1.13 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 4.8.19.2.2), 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>

4.8.19.2 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 4.8.19-2 provides a description of the parameters used in executing the EcDsBulkDelete.pl script.

Table 4.8.19-2. 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.

4.8.19.2.1 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.

4.8.19.2.2 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.

4.8.19.3 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 4.8.19-3 provides a description of the parameters used in executing the EcDsBulkUndelete.pl script.

Table 4.8.19-3. 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 4.8.19-3. 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.

4.8.19.3.1 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>]
```

4.8.19.3.2 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>]
```

4.8.19.4 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>]

Table 4.8.19-4 provides a description of the parameters used in executing the EcDsDeletionCleanup.pl script.

Table 4.8.19-4. Command Line Parameters for EcDsDeletionCleanup.pl (1 of 2)

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.
batch	No	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. 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. 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).

Table 4.8.19-4. Command Line Parameters for EcDsDeletionCleanup.pl (2 of 2)

Parameter Name	Mandatory	Description
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 <code>-batch</code> and <code>-grbatch</code>).</p> <p>If the environment variable BATCH_SIZE_GRANULE is set, and <code>-grbatch <number></code> is also specified, the value from command line parameter <code>-grbatch</code> will be used.</p> <p>If neither the environment variable BATCH_SIZE_GRANULE is set nor <code>-grbatch</code> 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 <code>-phbatch <number></code> is also specified, the value from command line parameter <code>-phbatch</code> will be used.</p> <p>If neither the environment variable BATCH_SIZE_PH is set nor <code>-phbatch</code> is specified, the user will be prompted at runtime to enter a value (a value of 5 is suggested by the prompt text).</p>

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 xxdbl01_svr
setenv SYB_DB_NAME EcInDb_<MODE>

```

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

4.8.19.4.1 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_LIBT.001 or AST_LIB.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 progress and error information will be written in the log file.

4.8.19.4.2 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 4.8.19.4.1), 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.

This page intentionally left blank.

4.9 Common Services Tools

This section describes the tools used by DAAC operators on a day-to-day basis:

1. Common Desktop Environment (CDE) Tool
2. Microsoft Office Professional
3. Netscape Communicator
4. Sun Java System Web Server
5. Batch Insert Utility
6. Data Pool Cleanup Utility
7. Update Granule
8. Data Pool Access Statistics Utility (DPASU) – Rollup Scripts
9. Data Pool Access Statistics Utility (DPASU) – Maintenance Scripts
10. Most Recent Data Pool Inserts Utility
11. Data Pool Collection-to-Group Remapping Utility
12. QA Update Utility
13. Data Pool Move Collections Utility
14. Data Pool Hidden Scrambler Utility
15. Data Pool Remove Collection Utility
16. Data Pool Band Backfill Utility
17. XML Replacement Utility

This page intentionally left blank.

4.9.1 Common Desktop Environment (CDE) Tool

The ECS uses the Common Desktop Environment (CDE) Tool COTS package to manage X windows. It is a commercial graphical user interface for UNIX supporting AIX and linux operating systems. It provides users registered at an ECS site with generalized support for performing the basic operations listed in Table 4.9.1-1.

Table 4.9.1-1. Common ECS Operator Functions Performed with CDE

Operating Function	GUI	Description	When and Why to Use
Start a desktop session	Basic login with userid and password	Invokes the CDE window manager.	Access an ECS host.
Use the Front Panel	Front Panel window	Contains set of controls for performing common tasks (i.e., calendar, email, clock, print, file management).	As needed during work session.
Manage files	File Manager	File management tool.	Perform file navigation/manipulation.
Use Application Manager	Application Manager	How to run applications using Application Manager, the main repository for applications in CDE.	Need to invoke applications.
Customize the desktop environment	Style Manager	Allow for customizing the look and behavior of desktop.	Need to customize desktop environment.
Use text editor	Text Editor	Supports creation/editing of short documents (e.g., memos, mail, resource files).	Need to create short documents.
Print	Printing	Explains how to access printers.	Need to print/change default printer.
Use Terminal	Terminal	Explains how to display and customize terminal emulator windows on desktop.	Need to access control terminal window.
Use Icon editor	Icon Editor	Creates files for use as desktop icons or backdrops.	Need to create icons/backdrops.
Use Image Viewer	Image Viewer	Allows for capture, viewing, editing, printing, and translation of monochrome/color image files.	Need to perform image manipulation.

4.9.1.1 Quick Start Using the Common Desktop Environment (CDE) Tool

After being registered as an ECS user by the site administrator, the user accesses the CDE window manager by logging into an ECS host using a defined UserID and password.

4.9.1.2 CDE Main Screen

Figure 4.9.1-1 presents an example of the type of support provided by the CDE Window Manager.

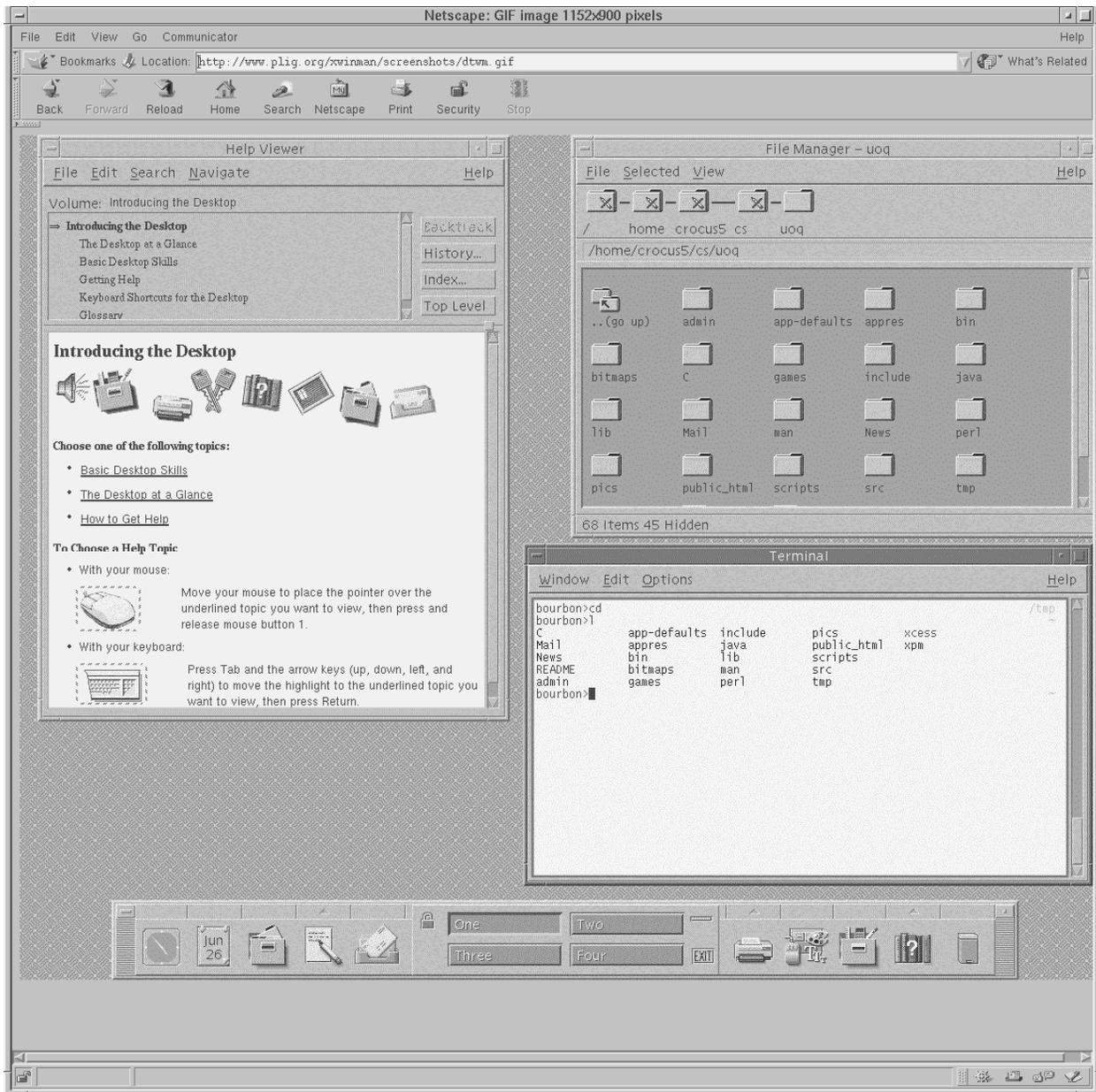


Figure 4.9.1-1. Example of CDE Window Manager Support Features

The Front Panel window at the lower part of the screen contains a set of icons allowing access to common support features. Through this panel the user can obtain time, date, monitor schedule,

access email, edit text files, print, access file manager to navigate the file system, and application manager to invoke and manage custom applications.

The Help Viewer window to the left of the screen is a support feature the user can invoke to obtain detailed online explanation of CDE support capabilities.

The File Manager window at the upper right of the screen supports navigating the file system and creating, deleting, and moving file objects.

The Terminal window below the File Manager on the screen allows Unix command line access to operating system services.

In addition to the help accessible to the online user, detailed documentation of CDE capabilities from the user standpoint and the system administrator are available from the Sun vendor at the web location:

<http://docs.sun.com>.

4.9.1.3 Required Operating Environment

Refer to the Common Desktop Environment: Advanced User's and System Administrator's Guide.

4.9.1.4 Databases

None.

4.9.1.5 Special Constraints

Access to CDE is available only to registered users of ECS sites.

4.9.1.6 Outputs

The Common Desktop only outputs event and error messages.

4.9.1.7 Event and Error Messages

CDE issues both status and error messages to the operator screen. Error messages are listed in the CDE support documentation accessible at the web link:

<http://docs.sun.com>.

4.9.1.8 Reports

None.

This page intentionally left blank.

4.9.2 Microsoft Office Professional

EMD provides Microsoft Office Professional to the DAACs to support general office automation tasks. Table 4.9.2-1 lists the operational tasks supported in the Office Professional package.

Table 4.9.2-1. Common ECS Operator Functions Performed with Microsoft Office

Operating Function	GUI Program	Description	When and Why to Use
Word processing	Microsoft Word	Allows operator to create, edit, open, save, and print documents. Allows incorporation of material generated in Excel and PowerPoint.	To create and maintain DAAC policies and procedures.
Develop a spreadsheet	Microsoft Excel	Allows operator to manage, format, chart and analyze data imported from the MSS database.	Imports data from the MSS database to create a report on an as needed basis.
Develop a presentation	PowerPoint	Allows operator to produce presentation slides, drawings, handouts, speaker notes, outlines and graphs. Allows incorporation of material generated in Excel and Word.	To develop briefings on an as needed basis.
Develop a database	Microsoft Access	Allows operator to define, create, and maintain databases. Allows operator to query database information and generate reports.	Provides support for as-needed database querying and reporting. Allows export of data to Word and Excel for analysis.

4.9.2.1 Quick Start Using Microsoft Office Professional

Microsoft's Program Manager contains the Microsoft Office Professional icon, which can be selected to provide Excel, Word, PowerPoint, and Access icons that launch the applications. Refer to the following Microsoft documentation for more details about its applications:

Microsoft Word User's Guide

Microsoft Excel User's Guide

Microsoft PowerPoint User's Guide

Microsoft Access User's Guide

The documentation of Microsoft Office Professional used as a basis and referenced in this section, is for use with the Windows XP operating system.

Microsoft Office Professional is installed exclusively on PCs.

4.9.2.2 Invoking Microsoft Office Professional

On a PC running Windows XP, the Microsoft Office Professional products can be invoked from the Office toolbar (if present) or the “Start” menu on the desktop display.

4.9.2.3 Required Operating Environment

For all COTS packages, appropriate information on operating environments, tunable parameters, environment variables, and a list of vendor documentation can be found in a Configuration Management controlled document for each product.

4.9.2.3.1 Interfaces and Data Types

The Microsoft Office Professional applications Word, Excel, PowerPoint, and Access work alike and interface with each other as if each were a single program.

4.9.2.4 Databases

The individual Microsoft Office products maintain their “products” in proprietary file structures:

MS Word = **.doc**

MS EXCEL = **.xls**

MS PowerPoint = **.ppt**

MS Access = **.mdb**

Each release of these products can accept output from previous releases of the same product, and generally, their competitor’s products available at the time of their release. However, they may not be able to use the same file extension name structures created by a later release. See the Special Constraints Section 4.9.2.5 below.

4.9.2.5 Special Constraints

The version of Microsoft Office Professional installed is Office 2003 for Windows XP. Users must take care when importing files (.doc, .xls, .ppt, and .mdb) and ensure the files are not produced by a later version of these products.

4.9.2.6 Outputs

The Microsoft Office Professional products display their outputs on screen and produce printed output appropriate to the product.

4.9.2.7 Event and Error Messages

Microsoft Office Professional provides help windows to identify and explain any Microsoft Office error messages.

4.9.2.8 Reports

Refer to the associated Microsoft User Guide for detailed information on the generation of reports using Word, Excel, PowerPoint, and Access.

4.9.3 Firefox

Firefox is a GUI interface for browsing the World Wide Web (WWW) and for obtaining information from other sources. Some of the Firefox functions are:

- View/process text/html files as well as other MIME formats
- Provide an interface to Telnet, Gopher, FTP, Email, and Newsgroups
- Read content of bulletin boards on the world-wide-web

Firefox is used to perform the operator functions listed in Table 4.9.3-1. Please refer to the Firefox Help option for additional information on functionality not explicitly mentioned here.

Table 4.9.3-1. Common ECS Operator Functions Performed with Firefox

Operating Function	Command/Action	Description	When and Why to Use
View Web Pages	Main window	<ul style="list-style-type: none">• Operator views pages written in HTML source code.• These pages provide images, text, and form templates.	To obtain information and to process user-interactive forms.
Process Forms	Main window	<ul style="list-style-type: none">• Forms are provided for operator input.• Certain operations require a password.	Used to search or manipulate the existing database (functions add, delete, modify.)
Browse Bulletin Boards (BB)	Main window	Allows for exchange of information with users and scientists that share the same interest.	To ask or provide information on the BB subject to a large community of users.

4.9.3.1 Quick Start Using Communicator

For more information, Firefox's Help option is available online (Open the "Help" pulldown menu from the Firefox main screen and select Help Contents. The main page with the contents of the Firefox Help appears. The operator can select subjects he/she is interested in by following the available links. By opening the "File" menu on the main page and selecting "Print", a hardcopy of the displayed text can be obtained.

4.9.3.1.1 Command Line Interface

To execute Firefox from the command line prompt use:

```
> /tools/bin/firefox &
```

4.9.3.2 Firefox Main Screen

Once invoked, Firefox displays the startup screen shown in Figure 4.9.3-1.



Figure 4.9.3-1. Firefox: Browser with Default Home Page

Users can set Firefox to set a preferred Home Page by clicking **Edit** → **Preferences** on the menu bar, entering a URL in the Home Page field, then clicking **Close**. An example of a home page is the ECS Data Handling System page shown in Figure 4.9.3-2.



Figure 4.9.3-2. Firefox: EDHS Home Page

From the start-up Firefox screen, the operator has several choices for loading pages in any of the MIME formats known by Firefox:

- Move the cursor to a link displayed on the page, and click on this link
- Enter or modify a URL displayed in the location bar (just below the display's menu bar).
- Click **View** → **Sidebar** → **Bookmarks** on the menu bar, and select a URL from the bookmarks sidebar that is displayed
- Click **File** → **Open File...** on the menu bar to browse for and open a file

It is recommended that operators create bookmarks of pages to be accessed frequently. Bookmarks are saved in file `bookmarks.html` in the `~/.mozilla/firefox/<profile>` directory.

4.9.3.3 Required Operating Environment

For all COTS packages, appropriate information on operating environments, tunable parameters, environment variables, and a list of vendor documentation can be found in a CM controlled

document for each product. To find the documentation for Firefox, refer to the ECS Baseline Information System web page. Contact your local CM Administrator for how to access this website at your site.

4.9.3.4 Databases

Firefox maintains, in formatted data files, a variety of information such as user preferences, security certificates, bookmarks, cookies, download history, extensions and add-ons. On Linux, these files are placed in the user's `~/mozilla/firefox/<profile>` directory.

4.9.3.5 Special Constraints

None

4.9.3.6 Outputs

Firefox provides the outputs listed in Table 4.9.3-2 below.

Table 4.9.3-2. Firefox Outputs

Output	Description and Format
Screen Display	Shows the Firefox browser GUI screen and adjusts to the screen format.
Hardcopy of Display Window	Printed version of the contents of the display window.
Display Window saved to disk	Contents of the display window can be saved to disk in Text, Source or Postscript format.
Modified, deleted or created data files	Processing of forms allows the operator to modify, delete or create data files.

4.9.3.7 Event and Error Messages

Firefox issues both status and error messages to document the status of loading a document or to display the reason for not loading a document.

4.9.3.8 Reports

None

4.9.4 Sun Java System Web Server

Sun Java System Web Server 6.1, Service Pack 5 is a multi-process, multi-threaded, secure web server built on open standards. It provides high performance, reliability, scalability, and manageability for any size enterprise, and it includes modules for creating and managing Web content, for extending or replacing functions of the server (e.g., through Java servlets and JavaServer pages), and for providing application-specific services such as security and access control.

In EMD, Sun Java System Web Server is used by several subsystems to access HTML files and to service web-based applications. It is installed locally on machines that run EMD applications relying on it. A distinct instance of a Sun Java System Web Server is created for each such application, one per mode in which the application runs. For example, EMD's Order Manager, Data Pool GUI, and BMGT all need to use Sun Java System Web Server, and each of them runs in the three modes on sites' Data Pool Server machines. Consequently, nine instances of the Sun Java System Web server are required - one for each of the three applications in each mode. Applications communicate with the appropriate instance via a unique port number. The port numbers these Sun Web Servers use can be found in the EMD baseline document, 910-TDA-002, ECS Software Port Mapping Baseline.

An additional instance of the Sun Java System Web Server known as the Administration Server is created whenever Sun Java System Web Server is installed on a machine. You use it to manage all Web Server instances.

Table 4.9.4-1 summarizes the Sun Java System Web Server functions used by EMD and references vendor guides that describe their use. Release Notes are available on the Internet at <http://docs.sun.com/app/docs/doc/819-2479>.

Table 4.9.4-1. Common EMD Operator Functions Performed with the Sun Java System Web Server (1 of 3)

Operating Function	Command/Script	Description	When and Why to Use
Administer <i>Sun Java System</i> web servers	Administration Server GUI	Allow operators to add and remove web server instances.	When applications needing web servers are installed or removed.
Set Administration Preferences	Administration Server GUI	Allow operators to: <ul style="list-style-type: none"> • Stop the Administration Server • Edit its listen socket settings • Change the user account under which its processes run • Change its superuser settings • Specify log file options, including log file rotation • Configure JRE paths 	When <i>Sun Java System</i> is installed and when the Administration Server needs reconfiguration.
Provide security and encrypt transactions	Administration Server GUI	Allow operators to: <ul style="list-style-type: none"> • Create a trust database • Request, install, and manage VeriSign and other server certificates • Install and manage certificate revocation lists (CRLs) and compromised key lists (CKLs) • Enable client authentication 	As needed to activate security features designed to safeguard data, deny intruders access, and allow access to those authorized.

Table 4.9.4-1. Common EMD Operator Functions Performed with the Sun Java System Web Server (2 of 3)

Operating Function	Command/Script	Description	When and Why to Use
Configure web servers	Server Manager GUI	Allows operators to: <ul style="list-style-type: none"> • Start and stop web server instances • Adjust performance settings • Edit configuration file (magnus.conf) settings and apply them to the server • Add and edit listen sockets • View, manage, and archive logs • Monitor server activity and quality of service • Edit file cache settings 	As needed to improve web server performance, troubleshoot problems, and support use by EMD custom code.
Analyze log files	Server Manager GUI Administration Server GUI	Allows operators to: <ul style="list-style-type: none"> • View access logs • View error logs • Set logging preferences 	As needed to monitor and troubleshoot web server activities.
Monitor servers	Server Manager GUI	Allows operators to: <ul style="list-style-type: none"> • Compile and view a variety of server performance statistics in real-time • Set bandwidth and max connections parameters for enforcing quality of service policies 	As needed to monitor, manage, and troubleshoot web server activities and to tune server performance.

Table 4.9.4-1. Common EMD Operator Functions Performed with the Sun Java System Web Server (3 of 3)

Operating Function	Command/Script	Description	When and Why to Use
Program the server	Class Manager GUI	Allows operators to: <ul style="list-style-type: none"> • Install CGI programs, Java Servlets and JavaServer Pages • Configure how the server is to run them 	When installing new server-side applications or changing how the applications are to be run.
Manage server content	Class Manager GUI	Allows operators to: <ul style="list-style-type: none"> • Set primary and additional document directories • Configure document preferences • Configure URL forwarding • Customize error responses • Specify a document footer • Restrict the use of file symbolic links • Set the server to parse HTML files • Set cache control directives 	When creating or altering web server instances, to specify where documents to be served are located. When customized responses to client requests are warranted. When restrictions are needed on information cached by proxy servers.

4.9.4.1 Quick Start Using Sun Java System Web Server

Sun Java System Web Servers are managed with the help of the following four user interfaces:

- Administration Server – contains the Java forms for managing, adding, removing, and configuring web server instances
- Server Manager - contains the Java forms for configuring individual instances of web servers and for adding and configuring classes of virtual servers
- Class Manager – contains the Java forms for managing, adding, removing, and configuring virtual servers. Virtual servers allow you, with a single installed server, to offer companies and individual domain names, IP addresses and some server administration capabilities. A default virtual server is created automatically for each web server instance
- Virtual Server Manager – contains pages that allow you to see and edit all the settings for a single virtual server

Note: EMD currently uses multiple instances of the web server rather than virtual servers. In past releases, Sun Java System Web Server’s virtual servers did not support unique configuration information.

The Server Manager, Class Manager, and Virtual Server Manager are accessed from the Administration Server GUI. The Administration Server must be running before the operator can access the Administration Server GUI.

4.9.4.1.1 Command Line Interface

The preferred method for starting the Administration Server operationally is to type the following as root:

```
# /etc/init.d/webserver01 start
```

This starts the administration server using the port specified during installation.

To start the Administration Server GUI and proceed to access the functionality discussed in Section 4.9.4.1, start a web browser then enter the URL for the administration server as follows:

```
http://<servername>.<ECSDomain>.<domain>:<portnumber>
```

The operator is then prompted for a username and a password. Once this information is entered the Administration Server web page appears as shown in Figure 4.9.4-1.

Note: The browser used for this task must be capable of supporting frames and JavaScript. Netscape Communicator 7.0, included in the EMD baseline, is capable of supporting both frames and Java Script.

4.9.4.2 Sun Java System Web Server Main Screen

The Administration Server GUI is Sun Java System's main web server screen. It is used to administer all Sun Java System Web Server instances. The screen has six tabs, each of which contains buttons for accessing Java forms to perform functions that govern the Administration Server or all the other web servers under its control. The tabs are:

- Servers tab (shown in Figure 4.9.4-1) - for adding (i.e., creating) and removing web server instances and invoking the Server Manager GUI. Operators invoke the Server Manager GUI by first selecting a web server from the tab's Select A Server pulldown menu, then pressing the Manage button
- Preferences tab – for stopping the Administration Server; setting runtime options for the Administration Server (such as process owner, port number, and logging parameters); and viewing the Administration Server's access and error logs
- Global Settings tab – for configuring or enabling a directory service, access control, Sun Java System's built-in cron facility, a Java runtime environment or development kit, and SNMP options applicable to all web servers under the Administration Server's control
- Users&Groups tab – for creating and modifying users, groups, and organizational units in an LDAP database. EMD does not presently use an LDAP database, so attempts to access this tab are rejected
- Security tab – for creating a trust database; obtaining, installing, and managing server certificates; and managing certificate revocation and compromised key lists for the Administration Server

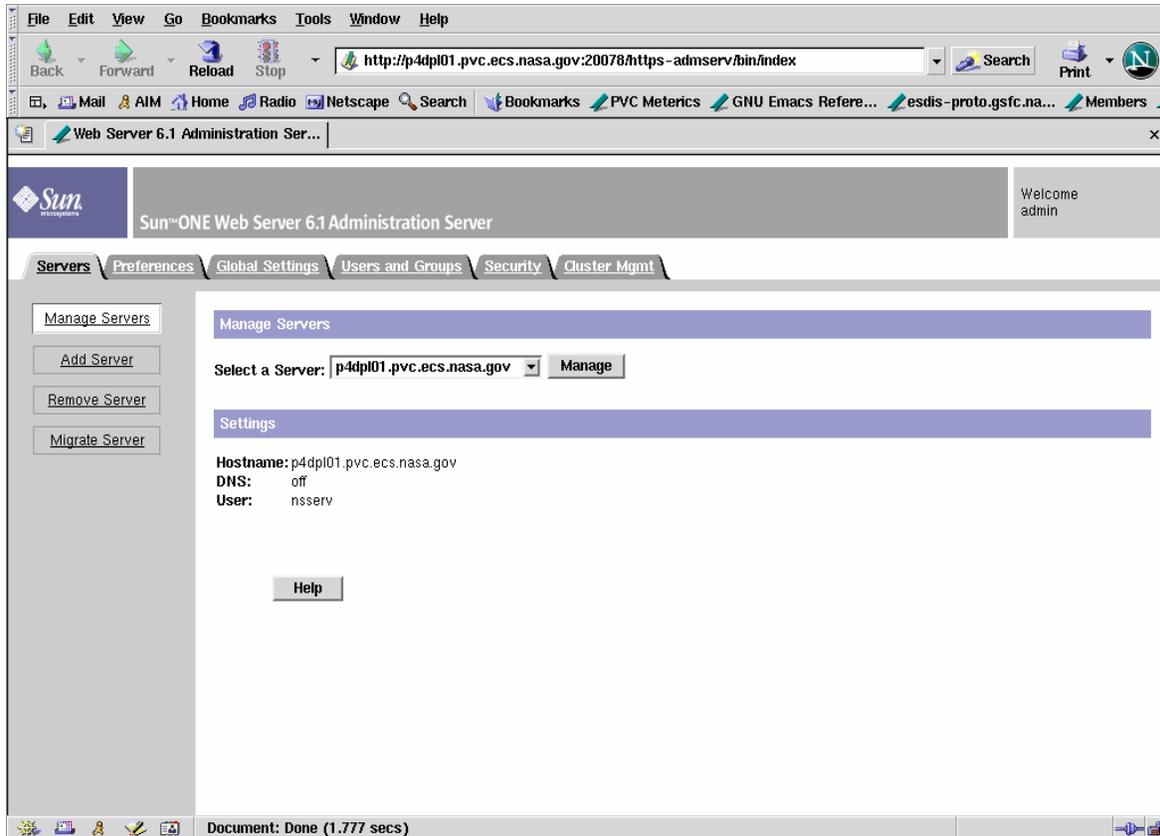


Figure 4.9.4-1. Sun Java System Web Server Administration Server Screen

- Cluster Management tab – for establishing and maintaining a group of Sun Java System Web Servers on multiple machines that can be administered by a single master Administration Server

4.9.4.2.1 Sun Java System Web Server: Server Manager Screen

Operators use the Server Manager GUI for managing a web server instance. (Web server instances are created via the Administration Server GUI.). The screen has eight tabs, each of which contains buttons for accessing Java forms to perform functions governing a single web server and its virtual web servers. The tabs are:

- Preferences tab (shown in Figure 4.9.4-2) – for starting and stopping the server and configuring runtime options for it such as port number, MIME types, caching, thread pools and performance tuning parameters
- Security tab – for creating a trust database; obtaining, installing, and managing server certificates; and managing certificate revocation and compromised key lists. This tab functions the same as the Security tab on the Administration Server GUI, but it applies to a web server other than the Administration Server

- Logs tab – for viewing and managing the web server’s access and error logs. Forms also let operators control the amount of information recorded in the access log, configure automatic rotation of log files, and generate statistics based on log contents
- Monitor tab – for monitoring a web server’s activity using built-in monitoring functions and SNMP, and for setting quality of service parameters (i.e., bytes per second and max connections)
- Virtual Server Class tab – for creating and modifying virtual server classes. It also provides access to the Class Manager GUI. By default, all virtual servers in a class have the same settings
- Java tab – for configuring Java features for a web server. This includes enabling web applications, servlets and JavaServer Pages (JSP), configuring Java Virtual Machine attributes, and managing session and JSP cache files
- Legacy Servlets tab – for specifying servlet properties as in Sun Java System
- Search tab – for searching the contents and attributes of documents on the web server. This tab supports creating and maintaining document collections, organizing and re-indexing collections at pre-determined times, and defining text search patterns, and configuring what users see when they get search results

Three buttons that appear above this screen’s tabs are:

- Server Manager – for choosing a different web servers to configure, including the Administration Server
- Class Manager – for invoking the Class Manager GUI to configure virtual servers for the web server
- Apply – for placing configuration changes into effect

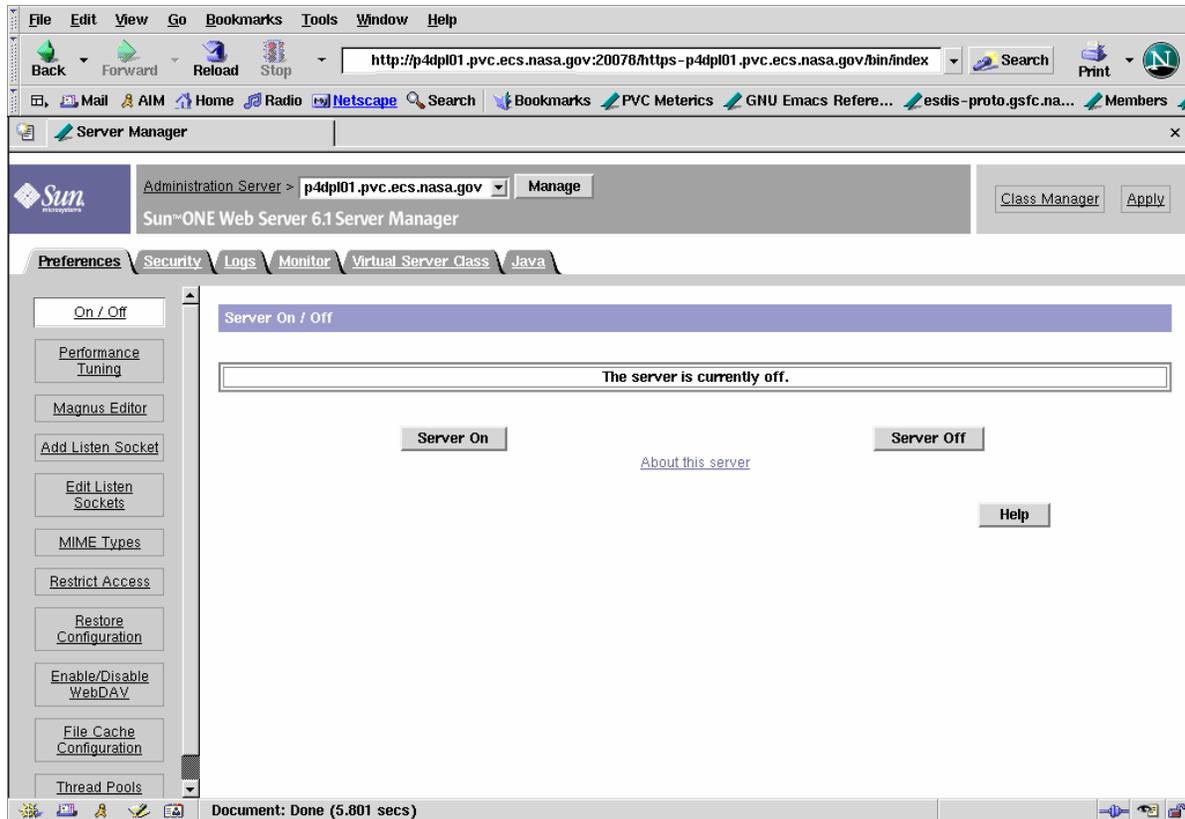


Figure 4.9.4-2. Sun Java System Web Server: Server Manager Screen

4.9.4.2.2 Sun Java System Web Server Class Manager Screen

Operators use the Class Manager GUI to create and configure virtual web servers for a web server instance. The Class Manager contains settings that affect a single class or single virtual server. You can set services for the class in the Class Manager, as well as add virtual servers (members of the class) and configure settings for an individual virtual server.

The screen has four tabs, each of which contains buttons for accessing Java forms to perform functions governing all the classes of virtual servers for the web server. The tabs are:

- Virtual Servers tab (shown in Figure 4.9.4-3) – for managing and creating virtual servers and editing basic virtual server properties. Preferences set via this tab govern MIME types, access control lists, quality of service, logging, web applications, and CGI programs used by the virtual servers. The tab also provides access to the Virtual Server Manager GUI. By default, all virtual servers in a class have the same settings.
- Programs tab – for configuring the CGI programs for the virtual server class. Use this tab to specify the location and query handler (if any) of your CGI programs as well as the prefix that URLs can use as an alias for the path to the programs.

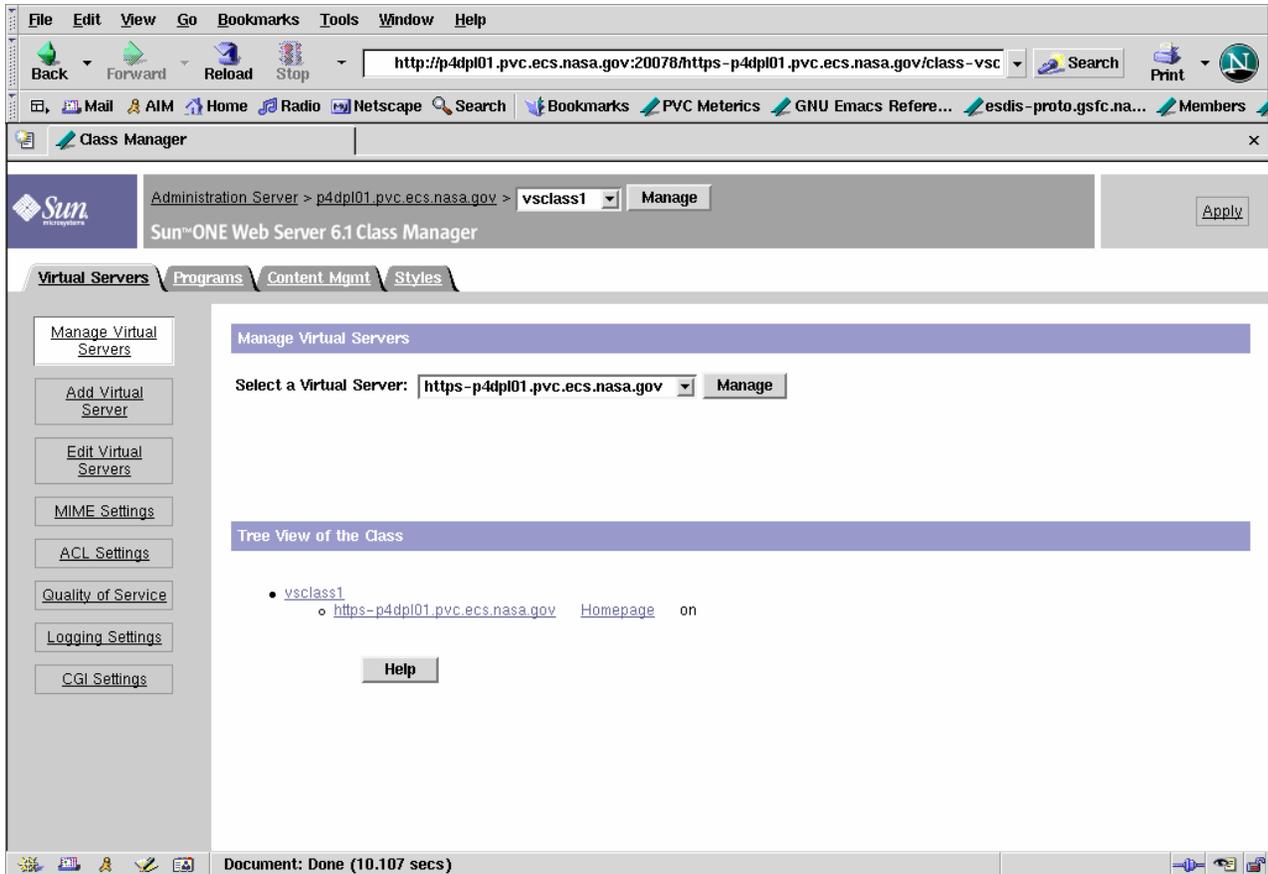


Figure 4.9.4-3. Sun Java System Web Server Class Manager Screen

- Content Management tab – for managing the web content that the class of virtual servers supports. Use this tab to specify paths to your root and additional document directories, establish directories where users can create and store their own home pages and documents, and define various properties governing how requests for content are to be handled
- Styles tab – for creating, editing, and applying configuration *styles* to files and directories. A style is a collection of predefined options that facilitates assigning properties for the web server’s handling of files and directories

The three buttons appearing above this screen’s tabs are:

- Server Manager – for returning to the Server Manager GUI
- Class Manager – for selecting a different class of virtual servers to manage
- Apply – for placing configuration changes into effect

4.9.4.2.3 Sun Java System Web Server Virtual Server Manager Screen

Operators use the Virtual Manager GUI to configure a single virtual server. The Virtual Server Manager contains settings that affect a single virtual server. These settings override those initially inherited from the server’s class. The screen has the following two tabs:

- Status tab (shown in Figure 4.9.4-4) – for viewing certain, key attributes of the virtual server. Two of the attributes are hyperlinks to pages for browsing the server’s access and error logs.
- Settings tab – for specifying values for most of the virtual server’s configurable attributes. These attributes (discussed previously), including document root, log file locations, access control lists, MIME types, and CGI parameters.

Four buttons that appear above this screen’s tabs are:

- Server Manager – for returning to the Server Manager GUI
- Class Manager – for selecting a different class of virtual servers to manage
- Virtual Server Manager – for selecting a different virtual server to manage
- Apply – for placing configuration changes into effect

4.9.4.3 Required Operating Environment

In EMD, the Sun Java System Web Server is deployed to run on Linux machines. Operators must have Netscape Communicator version 7.0 or higher and Java and cookies enabled in their browsers to use it.

4.9.4.4 Databases

The Sun Java System Web Server uses an internal, trust database to store public and private keys in support of Secure Socket Link encryption. The Administration Server and each server instance can have its own trust database. In addition, Sun Java System Web Server allows operators to define one or more Lightweight Directory Access Protocol (LDAP) databases that virtual servers can use for user authentication. The web servers themselves do not manage LDAP databases, and EMD does not currently use them. Refer to the *Sun Java System Web Server Administrator’s Guide* for further information on databases used by the Web Server.

4.9.4.5 Special Constraints

None.

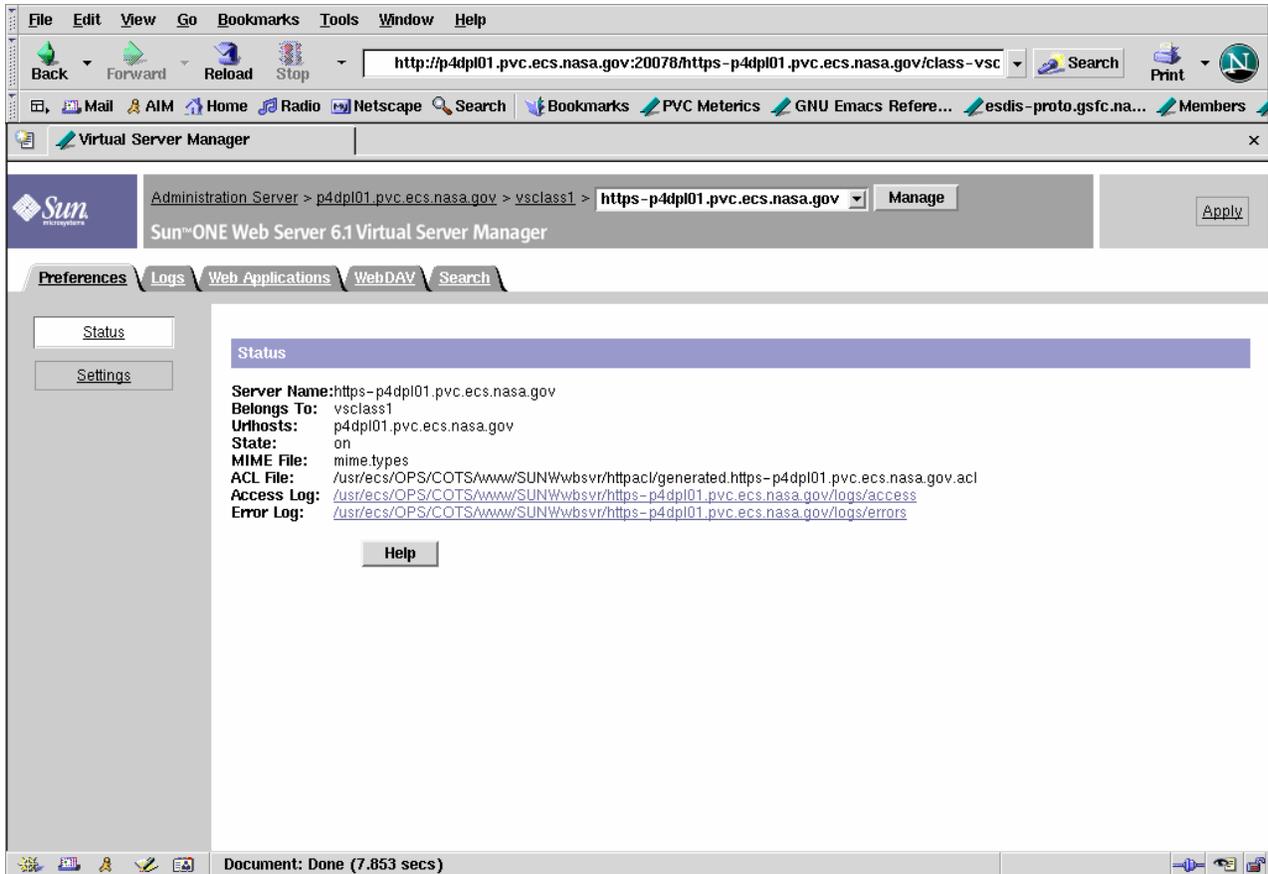


Figure 4.9.4-4. Sun Java System Web Virtual Server Manager Screen

4.9.4.6 Outputs

The Web Server supports real-time monitoring of web servers' activities. After enabling statistics, operators can view data about connections, the Domain Name Service (DNS), keep alives, cache and virtual servers. These can help them identify how many resources their servers need. Examples include:

- Number of idle (i.e., awaiting connection) threads
- Number of threads dealing with reading requests
- Number of threads dealing with writing responses
- Number of threads dealing with resolving hostnames
- Number of KeepAlive hits and flushes
- Number of DNS cache hits and misses

Additionally, server monitoring provides totals such as:

- Number of bytes received and sent
- Total requests
- Number of requests by type

Number of connections

Enable and monitor statistics using the Monitor Current Activity page, which is accessible from the Monitor tab of the Server Manager GUI.

To monitor a greater variety of server statistics, use Sun Java System's *perfdump* utility. The utility must first be "installed" by editing the web server's *obj.conf* configuration file and restarting the server. Statistics can then be viewed by pointing a browser to <http://<host>/.perf>. See the *Performance Tuning, Sizing, and Scaling Guide* for details.

4.9.4.7 Event and Error Messages

Each web server uses two files to record server activity. The *access* log file records requests to the server and server responses. The *error* log file lists errors the server has encountered. Both files typically reside in the web server's logs directory, but operators can control their location. Operators can also have the system automatically archive these files. See the *Administrator's Guide* for details.

For *access* logs only, operators can control the amount and format of what gets logged. They can specify whether to log accesses, what format to use, and whether the server should spend time looking up the domain names of the clients when they access a resource. They can also specify the file be written in common logfile format, flexible log format, or a user customizable format. Specify preferences using the Log Preferences page of the Server Manager GUI or edit the web server's configuration files directly. See the *Administrator's Guide* and the *NSAPI Programmer's Guide* for details.

4.9.4.8 Reports

Operators can generate a report about a web server's activity using Sun Java System Web Server's log analyzer. The report can contain a variety of statistics such as:

- Total hits and totals for various types of errors
- Most recently logged events
- Most frequent users
- Most commonly accessed URLs, and URL accesses exceeding some threshold amount
- Hosts most often accessing the server, and hosts whose accesses exceed some threshold amount

Operators can choose which of the statistics to include in a report, and output can be generated in HTML or text. However, results depend on what events the operators have the web servers log.

Reports are initiated from the Generate Report page, which is accessible from the Logs tab of the Server Manager GUI. Operators can also run program *flexanlg* from the command line. Refer to the *Administrator's Guide*, Chapter 9 - "Using Log Files", for additional information.

4.9.5 Batch Insert Utility

The Batch Insert Utility allows operators to insert granules residing in or outside of (non-ECS granules) the ECS archive into the Data Pool. It is a command line utility, which queues the granules up for dispatch by the Data Pool Action Driver (DPAD) for insertion by the Data Pool Insert Utility (DPIU). It accepts either a list of ECS granule identifiers or a list of non-ECS file names. A label identifying a batch of granules is specified as a command-line parameter so that operators can monitor a batch with the Data Pool Maintenance GUI. Thematic collections are also supported so the granules to be inserted can be linked to a theme.

Fault recovery capability is also supported, preventing inserts of duplicate actions inserted from a previous run.

Input is provided via input file or standard input.

4.9.5.1 Quick Start using the Batch Insert Utility

Enter the following command to start the Batch Insert Utility:

```
> EcDIBatchInsert.pl mode -ecs | -nonecs [ -file pathname ]  
[ -theme "theme_name" ] [ -label label ]  
[ -rpriority priority ] [ -rperiod period ]  
[ -dpriority priority ] [ -mdonly ]  
[ -verbose ]
```

Table 4.9.5-1 provides a description of these command line parameters.

Table 4.9.5-1. Command Line Parameters of the Batch Insert Utility (1 of 2)

Parameter Name	Description
<i>mode</i>	An input parameter specifying the mode of operation. This must be the first parameter passed, and it must be a valid, existing Data Pool mode with a format like OPS or TS1.
-ecs	Indicates that ECS granules are inserted. The input file (see -file) (or standard input) consists of a list of granule ids.
-nonecs	Indicates that non-ECS granules are inserted. The input file (see -file) (or standard input) consists of a list of XML file pathnames.
-file <i>pathname</i>	The pathname of the input file containing a list of either granule ids (if -ecs is specified) or XML pathnames (if -nonecs is specified).
-theme " <i>theme_name</i> "	Theme name to be associated with granules. <i>theme_name</i> is a character string and must match an existing theme name in the Data Pool inventory. Enclose it in quotes if embedded blanks or other special characters are part of the name.

Table 4.9.5-1. Command Line Parameters of the Batch Insert Utility (2 of 2)

Parameter Name	Description
-label <i>label</i>	An identifying label to be linked to the batch of granules being inserted. <i>label</i> is a character string. If no batch label (-label) is supplied, the label is set to the first sixteen characters of the input filename (excluding the directory name). If standard input is used in lieu of an input file, a batch label must be specified with the -label option.
-rpriority <i>priority</i>	A retention priority to be applied to all granules being inserted. $255 \geq \textit{priority} \geq 1$
-rperiod <i>period</i>	Number of days to retain all granules being inserted in inventory.
-dpriority <i>priority</i>	A dispatch priority to be applied to all granules being inserted. $255 \geq \textit{priority} \geq 1$
-mdonly	Flag indicating only metadata files are inserted for all granules being inserted.
-verbose	Directs the utility to run using verbose option. Default is non-verbose.

Mandatory parameters include *mode* and either -ecs or -nonecs. *Mode* must be the first parameter supplied.

4.9.5.1.1 Batch Insert Utility Commands

Below are some examples for invoking this utility:

1. `EcDlBatchInsert.pl OPS -ecs -file /home/fred/ECSMODISgranules1 -verbose`

Adds actions to action insert queue for all ECS granules specified by granule ids in the input file. No -label parameter specified, so label is formed from first 16 characters of input filename (ECSMODISgranules). Runs in the verbose mode.

2. `cat /home/fred/ECSfile1 | EcDlBatchInsert.pl OPS -ecs -label MODIS_batch1 -verbose`

This example is similar to example 1 but using standard input instead of -file. Note that the -label parameter must be supplied since filename is not accessible to the utility.

3. `EcDlBatchInsert.pl OPS -nonecs -file /home/fred/nonECSSVolcanogranules
-label Chig_volcano -theme "Chiginagak Volcano 2002"`

Adds actions to action insert queue for all non-ECS granules specified by XML pathnames in the input file. All granules are linked with theme name of "Chiginagak Volcano 2002" in inventory. Runs in the non-verbose mode.

4. `EcDlBatchInsert.pl OPS -ecs -file /home/fred/ECSMODISgranules1 -verbose -mdonly`

This example is similar to example 1 but only metadata files are inserted.

5. `EcDlBatchInsert.pl OPS -ecs -file /home/fred/ECSMODISgranules1 -verbose -rpriority
200`

This example is similar to example 1 with retention priority of granules to be set to 200 in the inventory.

```
6. EcDlBatchInsert.pl OPS -ecs -file /home/fred/ECSMODISgranules1 -verbose -rpriority
   200 -rperiod 10 -dpriority 5
```

This example is similar to example 1 with retention priority of granules to be set to 200 in the inventory, retention period to last 10 days, and dispatch priority set to 5.

4.9.5.2 Batch Insert Utility Main Screen

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

4.9.5.3 Required Operating Environment

The Batch Insert Utility runs on Linux platforms.

4.9.5.4 Databases

Table 4.9.5-2 lists the supporting products this tool depends upon to function properly.

Table 4.9.5-2. Interface Protocols

Product Dependency	Protocols Used	Comments
Data Pool and AIM databases	SQL	Via SQL server machines
Perl DBI	DBD::Sybase	Requires proper install of base-lined version of Perl.

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 the database server is unavailable, the connection to the database was dropped, or there was an error executing the 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 may repeatedly attempt to connect to the database, depending on how the configuration file was set (see Section 4.9.5.4.1). If, for example, NUM_RETRIES was set to 5 and SLEEP_SEC was set to 10, this means it tries to connect 5 times, and waits 10 seconds before each attempt – a total of 50 seconds if all attempts are unsuccessful.

4.9.5.4.1 Configuration File Format – EcDlBatchInsert.CFG

The “config” file contains vital details about how to connect to the Sybase database. Without this file, the utility cannot run. The “config” file must be a single-entry plain text ASCII file, which has the following format:

```
SYB_USER = EcDlBatchInsert
SYB_SQL_SERVER = <string>
SYB_DBNAME = <string>
PGM_ID = <string>
NUM_RETRIES = <integer>
SLEEP_SEC = <integer>
```

Table 4.9.5-3. Individual Parameter

Parameter Name	Description
SYB_USER	The user name for the Sybase connection.
SYB_SQL_SERVER	The name of the SQL server for this Sybase connection.
SYB_DBNAME	The name of the Data Pool database you intend to connect to
PGM_ID	Program ID used for connecting to the Data Pool database.
NUM_RETRIES	The number of times the utility attempts to connect to the database before exiting. The recommended default is 5.
SLEEP_SEC	The number of seconds the utility waits ('sleep') between connection attempts. The recommended default is 10.

4.9.5.5 Special Constraints

The Batch Insert Utility runs only if the Data Pool and AIM database servers are running and if the databases are available. It also assumes the stored procedures are present.

4.9.5.6 Outputs

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

4.9.5.7 Event and Error Messages

Events and error messages are written to the log file. A usage message is displayed to the screen when command-line parameters are incorrectly specified.

The utility produces a log file called EcDIBatchInsert.log in the /usr/ecs/<mode>/CUSTOM/logs directory. 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.

4.9.5.8 Reports

None.

4.9.6 Data Pool Cleanup Utility

4.9.6.1 Overview

The Data Pool Cleanup utility provides a mechanism by which the ECS Operations Staff can remove granules and their associated metadata and browse files from the Data Pool disks and corresponding Data Pool database inventory. Qualification of a granule for cleanup is usually based on two criteria: expiration date/time and retention priority. If a granule's expiration date is prior to midnight of the previous day (plus or minus a specified optional offset), and its priority is less than a specified threshold, it will be eligible for removal. ECS Operations Staff may wish to run Data Pool Cleanup at regular intervals via a cron job, removing granules which have expired and have a retention priority below a certain threshold. With the proper options, this would enable ECS Operations Staff to prevent the Data Pool file systems from filling up with little to no intervention by the operator.

In addition to the nominal, non-interactive scenario outlined above, ECS Operations Staff may wish to run the utility manually, perhaps with more control over which granules are removed. To this end, the operator may provide a file to the Data Pool Cleanup utility containing either geoids (Datatype:Shortname.Versionid:dbID entries) or Data Pool granule IDs of the granules they wish to remove. In addition, they may desire to remove from the Data Pool any granules which are scheduled for deletion from the ECS Inventory. The Data Pool Cleanup utility can automatically find and remove such granules, optionally along with expired or specified granules. The Data Pool Cleanup utility can also be run in "validation" mode to clean up orphans on the datapool disks (i.e. files which are not reflected in the database), or phantoms in the database (i.e. granules in the database whose associated files do not exist on disk). These inconsistencies accumulate over time for many reasons, such as granules being cleaned up by Order Manager, and must be removed periodically. The Cleanup utility will also remove "Most Recent Data Pool Insert" utility files prefaced with naming convention "*DPRecentInserts*" that exist in the top level and collection-level directories of each Data Pool file system.

When a granule is removed from the Data Pool, the ECS Clearing House (ECHO) must be informed that the granule is no longer available for ftp download. The Data Pool Cleanup utility provides two options for doing this. First, the utility can be run in "**deleteall**" mode, which will remove the granules and export their deletion all at once. Since this would likely leave a short window between the granule being deleted, and ECHO processing the deletion, the utility offers a second option. The Data Pool Cleanup utility can be run in "**predelete**" mode, which will mark the granules for deletion, and export their deletion to ECHO, but not actually complete the deletion. The utility must then be run in "**finishdelete**" mode to complete the deletion. Splitting the cleanup up into two operations allows ECS Operations Staff to remove the granules from the ECHO list of granules available for FTP download, and then delete them at a later time to minimize the likelihood of broken FTP links.

If, as a result of granules being deleted by the Data Pool Cleanup utility, the free space in a file system which is currently flagged as being full is caused to exceed an operator configured limit, the utility will set the freeSpaceFlag to "Y" in the DIFileSystems table. The filesystem on which a given granule resides is transparent to the operator, so the granules being cleaned up or

validated may reside on any number of different file systems. If a file system is marked as unavailable, the utility will automatically skip the cleanup or validation of any granules belonging to collections residing on that file system, and log a message explaining why the collection was skipped in the log file.

4.9.6.2 Using the Data Pool Cleanup Utility

The Data Pool cleanup utility is started using the following parameters:

1. For a "cleanup only" run:

a. predelete/finishdelete run sequence

In this case, two runs are required to perform a cleanup, the first with **-echomode predelete** and the second with **-echomode finishdelete**. For example:

```
EcDlCleanupDataPool.pl <mode> -echomode predelete [-noprompt] [-offset  
<offset # of hours>] [-limit <priority limit>] [-theme <themeName>] | [-  
file <fileName>] | [-geoidfile <geoidFileName> | -ecsgrandel
```

and

```
EcDlCleanupDataPool.pl <mode> -echomode finishdelete [-noprompt]
```

Note: No other parameter can be specified with the **finishdelete** parameter, with the exception of the **noprompt** parameter.

b. deleteall run

Only one cleanup run is required with **-echomode deleteall**. For example:

```
EcDlCleanupDataPool.pl <mode> -echomode deleteall [-noprompt] [-offset  
<offset # of hours>] [-limit <priority limit>] [-theme <themeName>] | [-  
file <fileName>] | [-geoidfile <geoidFileName> | -ecsgrandel
```

c. theme cross-reference run

For example:

```
EcDlCleanupDataPool.pl <mode> -themexref <themeName> [-noprompt]
```

Note: No other parameter can be specified with the **themexref** parameter, with the exception of the **noprompt** parameter.

2. For a "validation only" run:

```
EcDlCleanupDataPool.pl <mode> -orphan | -phantom [-collgroup <groupList>] [-maxorphanage <age in # of days>] [-nofix]
```

OR

3. For a "cleanup followed by validation" run:

Specify a command line option *-cleanvalidate* along with the options described in 1 and 2 above.

There are various command line parameters that are used in combination with each other. Table 4.9.6-1 provides a description of these parameters.

Table 4.9.6-1. Command Line Parameters (1 of 3)

Parameter Name	Required	Description
<i>noprompt</i>	No	Suppresses all confirmations and warnings normally displayed on the screen. Requires no value. May be used with any other option.
Parameters specific to performing cleanup		
<i>echomode</i>	Yes (for cleanup and clean/validate, not for validation only)	Specifies the method by which the Cleanup utility will report deletion candidates to the ECS Clearing House (ECHO). The echomode parameter can take 1 of 3 values: predelete , finishdelete or deleteall . predelete signifies that the cleanup utility will build the list of items to clean up from the Data Pool and will report their removal from the DataPool to ECHO through the BMGT utility. No actual data will be cleaned up from the disks or database inventory using predelete . A value of finishdelete signifies that the cleanup utility will now delete all of the data that was marked for deletion during a previous run with the predelete parameter. A third value of deleteall will indicate that the Cleanup utility should build its list of items to cleanup, actually clean them up and to <i>also</i> notify ECHO via the BMGT utility. deleteall does not allow for a time lag between Cleanup deleting the granules and ECHO performing its own clean up of granule FTP URLs. The predelete/finishdelete run sequence can be viewed as a logical run done in two parts. The normal sequence will be to run Cleanup first with predelete and then with finishdelete . Note that an echomode parameter with a value of finishdelete can only be specified by itself since the list of items to delete will have already been determined by the previous run. If a predelete run is performed, the subsequent run <i>must</i> specify finishdelete in order to perform the actual deletions. This requirement is enforced by the utility to avoid operator error and end-user confusion. The values of predelete and deleteall may be used with each of the other parameters specific to performing Cleanup except themexref (see themexref parameter description).

Table 4.9.6-1. Command Line Parameters (2 of 3)

Parameter Name	Required	Description
<i>offset</i>	No	<p>Specifies hours before (negative) or after (positive) midnight of the previous day from which to delete. Defaults to zero.</p> <p>(Some examples: -offset 5 would delete all granules which had expired as of 5 AM of the current day; -offset -5 would delete all granules which had expired as of 7 PM yesterday -offset 72 would delete all granules which will be expiring in 72 hours measured from the previous day's midnight).</p>
<i>limit</i>	No	<p>Specifies limiting value used for determining which granules will be deleted. Will delete all granules with priority less than or equal to the specified limit. Must be within the range 1–255, 1 being the lowest priority and 255 being the highest priority. Defaults to value specified in configuration file.</p>
<i>file</i>	No	<p>Specifies name of file containing Data Pool granule ids to be deleted. May not be used with any other options other than the <i>noprompt</i> and <i>echomode</i> options.</p>
<i>geoidfile</i>	No	<p>Specifies the name of the file containing geoids which are a combination of science type, esdt short name and version id and ECS Inventory database id. Granules in this file whose ECS id match those in the data pool are candidates for data pool cleanup if specified by this option. May not be used in conjunction with any other options other than the <i>noprompt</i> and <i>echomode</i> options. Note that the geoid file can contain science granules as well as non-science granule because the ECS Inventory may delete these types of granules. The input value for this parameter is logically defined to be the output of any ECS Inventory phase 1 (EcDsBulkDelete.pl) granule deletion run. This will cause the Data Pool cleanup utility to clean up any ECS Inventory granules found in the geoid input file to be removed from the Data Pool database.</p>
<i>ecsgrandel</i>	No	<p>Indicates that only granules removed from the ECS Inventory Database will be removed from the data pool if they exist. This option may not be used in conjunction with any other options other than the <i>noprompt</i> and <i>echomode</i> options. No other cleanup will occur.</p>
<i>theme</i>	No	<p>Specifies the name of a theme for which cleanup is to be performed. The Cleanup Utility will clean up granules that would otherwise qualify for cleanup only if the granules are associated with that theme, and remove the granules entirely if they are not associated with any other theme, otherwise only remove the cross references with that theme. The theme name must be enclosed in quotes ("").</p>

Table 4.9.6-1. Command Line Parameters (3 of 3)

Parameter Name	Required	Description
<i>themexref</i>	No	Specifies the name of a theme all cross-references of which are to be removed from the Data Pool inventory. This option is specified to remove the theme cross references only. It does not remove any granules. This command line option cannot be used with any other options other than the noprompt option. The theme name must be enclosed in quotes (").
Parameters specific to performing validation		
<i>orphan</i>	No	Specifies that Data Pool validation be performed by performing orphan checking. Orphans are defined as files that are on disk but are no longer part of the Data Pool inventory in the database. The Data Pool inventory validation function will remove all orphan files and links from the Data pool disks unless nofix option was specified. (In order to perform validation either orphan or phantom or both must be provided on command line.)
<i>phantom</i>	No	Specifies that phantom checking be performed. Phantoms are defined as files that exist in the Data Pool inventory in the database, but do not exist on disk. The Data Pool inventory validation function will remove granules affected by any phantom from the inventory and all its remaining files and links from the Data pool disks unless nofix option was specified. (In order to perform validation either orphan or phantom or both must be provided on command line.)
<i>maxorphanage</i>	No	Specifies the maximum orphan age in days. The value specified must be greater than or equal to 10 days. The Data Pool inventory validation function will consider only those files on disk as orphans whose age is equal to or larger than the maximum orphan age specified. If this parameter is omitted, a default value specified in the configuration file will be used.
<i>nofix</i>	No	Specifies that a Data Pool validation be performed, but do not attempt to reconcile the discrepancies found. The validation results will be logged.
<i>collgroup</i>	No	Limit the Data Pool validation to the collection group(s) specified. Single or multiple collection groups can be specified on the command line. Multiple collection groups if provided must be separated by commas, with the string enclosed in double quotes ("), e.g. "MOAT, ASTT". By default, all collection groups in the Data Pool inventory will be included in the validation if this option is not specified.
<i>cleanvalidate</i>	No	Specifies that a cleanup run should be followed by a validation run.

The Data Pool Cleanup utility performs the following as part of the "cleanup" processing:

- Removes all data pool granules along with the associated browse files (if no other granules are cross-referenced to them) and the browse links that meet the specified cleanup criteria, from both the Data Pool inventory and the disks. This occurs when the **echomode** parameter has a value of **finishdelete** or **deleteall**.
- Removes all recent insert files prefixed with "*DPRecentInsert*" that are older than 7 days. These files are found in `/datapool/<fs1>/<mode>/user/` and `/datapool/<fs1>/<mode>/user/<group>/<esdt>`.
- If the **echomode** parameter is set to **predelete** or **deleteall**, DataPool Cleanup will add a row to the `DsMdBmgtPkgCycles` table in the ECS Inventory database to request the export of DataPool deletions to ECHO. The Bulk Metadata Generation Tool (BMGT) Monitor Server will periodically poll the database for such requests, and initiate a BMGT export cycle on DataPool Cleanup's behalf when one is found. This BMGT "*CLEANUP*" export cycle will export the deletions of any granules which are being cleaned up by the current run of the DataPool Cleanup utility, as well as any granules which have been removed from the public DataPool holdings by any other means (*e.g.* replacement, logical delete/hiding, etc) since the last BMGT metadata export. Note: The CLEANUP cycle may not be picked up by BMGT immediately as it depends on the BMGT Monitor polling interval which is configured via the BMGT GUI (See Section 4.8.16). In **predelete** or **deleteall** mode, the DataPool Cleanup utility will wait up to twice the length of the BMGT Monitor Server polling interval for the cycle to be picked up before returning. No granules will actually be cleaned (*i.e.*, **finishdelete** will fail) until the cycle is picked up.

The Data Pool Cleanup utility performs the following as part of the "validation" processing:

- Validates the Data Pool inventory and disk content by checking for the existence of orphans and/or phantoms and removing or just logging them depending on the command line options specified.

The Cleanup Utility has the ability to continue from where it stopped in the event of interruption. Additionally, to alleviate database contention, it allows only one instance of itself to execute concurrently. The Cleanup Utility also provides the **noprompt** option to suppress operator prompts/messages.

In normal operations, the Cleanup Utility will be run once a day as a cron job as a "cleanup only" run executing with **-echomode predelete**. This will build the list of cleanup candidates (based on the expiration date and retention priority) that will be reported to ECHO as those which will be deleted in the next run of cleanup. Also, those granules that have been entered in the ECS Inventory database's deleted granules table will be reported. On a subsequent run within the same 24-hour period, the cleanup utility will be run with **-echomode finishdelete** to perform the actual cleanup processing that was reported to ECHO in the **predelete** mode. A validation run can be time consuming and should not be run as often, since it potentially involves the checking of all files in the entire Data Pool inventory against those on the Data Pool disk in order to find

and remove the discrepancies. It is advised that the validation function be run using *collgroup* option whenever possible to limit the validation to the user specified collection groups.

Section 4.9.6.4 provides some examples along with detailed explanations for executing this utility.

4.9.6.3 Cleanup Configuration File

The Data Pool Cleanup utility uses a configuration file, *EcDICleanupDataPool.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 4.9.6-2 describes the configuration parameters.

Table 4.9.6-2. Configuration Parameters

Parameter Name	Value Description
SYB_USER	Sybase login name for the user of the Data Pool database.
SYB_SQL_SERVER	Name of Sybase SQL Server hosting Data Pool database.
SYB_DBNAME	Name of Data Pool database.
SYB_INV_DB_NAME	Name of the ECS Inventory Database.
PGM_ID	Program identifier used as seed to generate database password.
DEFAULT_LIMIT	Default priority limit if a limit (-limit) is not provided via command line.
NUM_RETRIES	Number of times database operation will be attempted.
SLEEP_SEC	Number of seconds between retries.
MAX_ORPHAN_AGE	Maximum age in days in qualifying a file as an orphan. A file must have an age greater than or equal to this value in order to be considered as an orphaned file. The parameter value must be 10 days or greater.
ORDER_OUTPUT_COLLECTIONS	A list of collection groups that should not be processed during orphan validation. Files in these directories will be ignored when determining orphans. The suggested value for this field is "OUTPUTS BRWS"
MAX_ORDER_AGE	The number of days for which a file in the order only directory will not be counted as an orphan. Suggested value for this field is 15.

4.9.6.4 Examples

4.9.6.4.1 Examples for "Cleanup Only" Run

Note: Each invocation of the utility that involves cleanup will also automatically clean up recent insert files. Each **predelete** or **deleteall** run will cause the export of a BMGT package to report the deletions of any cleaned up granules to ECHO.

1. Initiate Cleanup suppressing all user prompts and display messages, and report candidates to ECHO

```
EcDlCleanupDataPool.pl OPS -echomode predelete -noprompt
```

This command will generate a list of all granules in OPS mode with retention priorities less than or equal to the configured default limit (see Table 4.9.6-2) and expiration date/times on or before midnight of the previous day and report their removal from the DataPool to ECHO through a BMGT export package. It will also turn off all prompts, removing the need for operator intervention. This form is suitable for running cleanup as a background job on as a cron.

To actually clean up the data from the inventory and disks, a subsequent run must be made as follows using the echomode parameter of **finishdelete**:

```
EcDlCleanupDataPool.pl OPS -echomode finishdelete
```

2. Initiate Cleanup with default limit and date/time

```
EcDlCleanupDataPool.pl OPS -echomode predelete
```

This command will generate a list of all granules in OPS mode with retention priorities less than or equal to the configured default limit and expiration date/times on or before midnight of the previous day and report their removal from the DataPool to ECHO through a BMGT export package. To actually clean up the data from the inventory and disks, a subsequent run must be made as follows using the echomode parameter of **finishdelete**:

```
EcDlCleanupDataPool.pl OPS -echomode finishdelete
```

3. Initiate Cleanup with a specific priority limit

```
EcDlCleanupDataPool.pl OPS -echomode predelete -limit <priority  
limit>
```

This command will generate a list of granules with retention priorities less than or equal to < priority limit > and expiration date/times on or before midnight of the previous day and report their removal from the DataPool to ECHO through a BMGT export package.

For example, to clean up granules which had expired on or before midnight of the previous day and had a retention priority less than or equal to 100, the command would be the following:

```
EcDlCleanupDataPool.pl OPS -echomode predelete -limit 100
```

To actually clean up the data from the inventory and disks, this run would be followed by a subsequent invocation utilizing the **-echomode** parameter with a value of **finishdelete**:

```
EcDlCleanupDataPool.pl OPS -echomode finishdelete
```

4. Initiate Cleanup using a list of granules

```
EcDlCleanupDataPool.pl OPS -echomode predelete -file <file name>
```

This command will generate a list of granules whose granule ids are specified in <file name> and report their removal from the DataPool to ECHO through a BMGT export package. The granule IDs are Data Pool granule IDs. Note that the **-limit** and **-offset** parameters can not be specified with **-file** option.

For example, to clean up two specific granules with IDs 1234 and 5678, a file "twograns.dat" could contain the following:

```
1234
```

```
5678
```

and the command would be the following:

```
EcDlCleanupDataPool.pl OPS -echomode predelete -file twograns.dat
```

To actually clean up the data from the inventory and disks, this run would be followed by a subsequent invocation utilizing the **-echomode** parameter with a value of **finishdelete**:

```
EcDlCleanupDataPool.pl OPS -echomode finishdelete
```

5. Initiate Cleanup using a list of geoids

```
EcDlCleanupDataPool.pl OPS -echomode predelete -geoidfile  
geoidlist.dat
```

This command will generate a list of all the granules in the Data Pool whose ECS Inventory database IDs match the IDs that are enumerated in the geoid list and report their removal from the DataPool to ECHO. For example to clean up two specific ECS granules with geoids of SC:MISL0DF.001:27056 and SC:MOD01.004:27057, a file geoidlist.dat could contain the following:

```
SC:MISL0DF.001:27056
```

To actually clean up the data from the inventory and disks, this run would be followed by a subsequent invocation utilizing the **-echomode** parameter with a value of **finishdelete**:

```
EcDlCleanupDataPool.pl OPS -echomode finishdelete
```

6. Initiate Cleanup limiting the cleanup utility to only clean up granules found in the ECS granule event history table

```
EcDlCleanupDataPool.pl OPS -echomode predelete -ecsgrandel
```

This command will generate a list of all of the granules in the Data Pool that have entries in the ECS granule event history table that were marked as candidates for ECS Inventory granule deletion. The list will be reported to ECHO as cleanup candidates through a BMGT export cycle. No other Data Pool granules will be cleaned up regardless of the configured retention priority or expiration date.

To actually clean up the data from the inventory and disks, this run would be followed by a subsequent invocation utilizing the **-echomode** parameter with a value of **finishdelete**:

```
EcDlCleanupDataPool.pl OPS -echomode finishdelete
```

7. Initiate Cleanup with an offset from the previous midnight

```
EcDlCleanupDataPool.pl OPS -echomode predelete -offset <hours>
```

This command will generate a list of all granules with retention priorities less than or equal to the configured default limit and expiration date/times on or before midnight of the previous day plus or minus the offset. It will report the cleanup candidates to ECHO via a BMGT export cycle.

For example, let's suppose we wanted to clean up all granules that expired on or before 7pm yesterday. The command for this would be:

```
EcDlCleanupDataPool.pl OPS -echomode predelete -offset -5
```

Similarly, if we wanted to clean up all granules that will expire 48 hours in the future (actually 48 hours from the previous midnight), the command would be:

```
EcDlCleanupDataPool.pl OPS -echomode predelete -offset 48
```

As in previous examples, to actually clean up the data from the inventory and disks, each of these runs would be followed by a subsequent invocation utilizing the **-echomode** parameter with a value of **finishdelete**:

```
EcDlCleanupDataPool.pl OPS -echomode finishdelete
```

8. Limit the cleanup to a specific theme

```
EcDlCleanupDataPool.pl OPS -echomode predelete -offset <hours> -  
theme <themeName>
```

This command is the same as the one described in the Example 7 except the cleanup will be limited to a specified theme <themeName>.

As in previous examples, to actually clean up the data from the inventory and disks, the run would be followed by a subsequent invocation utilizing the **-echomode** parameter with a value of **finishdelete**:

```
EcDlCleanupDataPool.pl OPS -echomode finishdelete
```

9. Remove all cross-references to a specific theme

```
EcDlCleanupDataPool.pl OPS -noprompt -themexref <themeName>
```

This command removes all granule cross references to a specified theme <themeName> from the Data Pool inventory. It does not delete the granules or the physical granule files.

10. Initiate a Data Pool cleanup run to report cleanup candidates to ECHO and to subsequently remove the granules from the Data Pool database inventory and disks

```
EcDlCleanupDataPool.pl OPS -echomode deleteall
```

The **deleteall** value of the **echomode** parameter indicates the cleanup utility will report all cleanup candidates to ECHO and remove the inventory entries from the Data Pool database as well as clean up the physical files from the disks. The **echomode** parameter value of **deleteall** can be used with all other cleanup parameters except **-themexref**.

Note; The **deleteall** option operates in the same manner that Cleanup did prior to Synergy IV in that it reports to ECHO and deletes in one step rather than two.

4.9.6.4.2 Examples for "Validation Only" Run

1. Initiate a Data Pool Validation run suppressing all user prompts and display messages

```
EcDlCleanupDataPool.pl OPS -orphan -phantom -noprompt
```

This command validates the Data Pool by checking for orphans and phantoms and removes any discrepancies found from the Data Pool inventory and the disks. In checking for orphans, a default maximum orphan age provided via the configuration parameter **MAX_ORPHAN_AGE**

will be used. In addition it will ignore any files in the directories specified in the configuration parameter ORDER_OUTPUT_COLLECTIONS. It will also clean up order only orphans that are older than the number of days specified in the configuration parameter MAX_ORDER_AGE.

- Orphans: Files which exist in the DataPool file system, but not in the DataPool database.
- Phantoms: Granules which exist in the DataPool database, but whose files do not exist in the DataPool filesystem.

2. Initiate a Data Pool Validation without fixing discrepancies found

```
EcDlCleanupDataPool.pl OPS -orphan -phantom -nofix
```

This command performs the same Data Pool validation run as in Example 8, except that the discrepancies found will not be fixed but will be logged.

3. Initiate a Data Pool Validation but limit the validation to specified collection group(s)

```
EcDlCleanupDataPool.pl OPS -orphan -phantom -collgroup  
"MOAT,ASTT"
```

This command performs the same Data Pool validation run as in Example 8, except that the validation checking will be limited to the collection groups MOAT and ASTT.

4.9.6.4.3 Examples for "Cleanup followed by Validation" Run

1. Initiate a Data Pool Cleanup and Validation suppressing all user prompts and display of messages

```
EcDlCleanupDataPool.pl OPS -echomode predelete -cleanvalidate -  
orphan -phantom -noprompt
```

This command will first generate a list of all granules in OPS mode with retention priorities less than or equal to the configured default limit and expiration date/times on or before midnight of the previous day. It will report the cleanup candidates to ECHO via a BMGT export cycle. It will then validate the Data Pool by checking for both orphans and phantoms and will remove all discrepancies found from the Data Pool inventory and the disks.

As in previous examples, to actually clean up the data from the inventory and disks, the run would be followed by a subsequent invocation utilizing the **-echomode** parameter with a value of **finishdelete**:

```
EcDlCleanupDataPool.pl OPS -echomode finishdelete
```

2. Initiate a Data Pool Cleanup and Validation specifying maximum orphan age via command line

```
EcDlCleanupDataPool.pl OPS -echomode predelete -cleanvalidate -  
orphan -maxorphanage 5
```

This command is the same as the one described in Example 2 of Section 4.9.6.1.2.2, except that the validation step will only check for orphans using the maximum orphan age of 5 days provided via command line, and no phantom checking will be performed.

As in previous examples, to actually clean up the data from the inventory and disks, the run would be followed by a subsequent invocation utilizing the **-echomode** parameter with a value of **finishdelete**:

```
EcDlCleanupDataPool.pl OPS -echomode finishdelete
```

3. Initiate a Data Pool Cleanup and Validation limiting the validation to specified collection group(s)

```
EcDlCleanupDataPool.pl OPS -echomode predelete -cleanvalidate -  
orphan -phantom -collgroup "MOAT,ASTT"
```

This command is the same as the one described in Example 2 of Section 4.9.6.1.2.2, except that the validation step will limit the Data Pool validation to the two collection groups MOAT and ASTT specified via command line.

As in previous examples, to actually clean up the data from the inventory and disks, the run would be followed by a subsequent invocation utilizing the **-echomode** parameter with a value of **finishdelete**:

```
EcDlCleanupDataPool.pl OPS -echomode finishdelete
```

4.9.6.5 Data Pool Cleanup Utility Main Screen

The Data Pool Cleanup Utility does not have a main screen. It has a command line interface only.

4.9.6.6 Required Operating Environment

The Cleanup Utility will run on a Linux platform.

4.9.6.7 Databases

Table 4.9.6-3 lists the supporting products that this tool depends upon in order to function properly.

Table 4.9.6-3. Product Dependencies

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

4.9.6.8 Special Constraints

The Data Pool Cleanup utility runs only if the Data Pool database server is running and if the database is available. It also assumes the stored procedures are present.

The DAACs currently run the cleanup utility as a cron job that runs at a set time each evening. To accommodate the pre-delete capability, a script could be written to call Cleanup with **predelete** and then with **finishdelete**. A sleep statement could be inserted between the two invocations. This script could then be run as a cron job. Optionally, Cleanup could be run manually: DAACs may choose to run with **predelete** from the command line, wait for the email notification of success from ECHO, and then run with **finishdelete**.

In order for the deletions to be exported to ECHO, the BMGT servers must be running. The Cleanup utility does not actually perform the export to ECHO, but only marks the granules to be deleted and then informs BMGT that there are deletions pending export to ECHO. If BMGT is not currently running, the actual deletion of granules will not be possible until it is started.

Note: If a run that used an echomode parameter of **predelete** is not followed by a run that specifies an echomode parameter value of **finishdelete** (before a run with any other option), Cleanup will fail and log an error. This is done to avoid the situation where granules reported to ECHO in the first run as deleted are not actually deleted in the next run of cleanup. The ECHO system could easily get out-of-sync with the Data Pool.

4.9.6.9 Outputs

Output of update events and errors will be always appended to a single log file.

4.9.6.10 Event and Error Messages

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

4.9.6.11 Reports

None.

4.9.6.12 Logs

The utility produces a log file called EcDICleanup.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.

4.9.6.13 Recovery

The Data Pool Cleanup Utility provides a capability to recover from an execution failure caused by situations such as the system faults or database errors leaving all or some of the deletes unprocessed. The utility will detect such failure upon the next run and continue processing the deletes that were left unprocessed in the previous run. The operator is given no choice as to recovery. Recovery will proceed or the Data Pool inventory and disk files will be in a corrupted state. Most Recent Insert files that may have not been cleaned up during previous runs due to utility interruption or failure will be cleaned up on succeeding runs.

4.9.6.14 Sybase Error Handling

If a Sybase error occurs, you will most likely see the actual Sybase error string displayed on the screen and in the log. Some errors can be that the database server is unavailable, that the connection to the database was dropped, or that there was an error executing the 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 or ECS Inventory database can not 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 5 and SLEEP_SEC was set to 10, this means it will try to connect 5 times, and will wait 10 seconds before each attempt – a total of 50 seconds if all attempts are unsuccessful.

This page intentionally left blank.

4.9.7 Update Granule

The Update Granule Utility provides the ECS Operations Staff with a command-line interface to update the expiration date and optionally the retention priority of granules in the Data Pool inventory. The granules in the Data Pool inventory can be ECS or non-ECS granules. The utility can perform updates for

1. A single Data Pool granule
2. Multiple Data Pool granules
3. Granules associated with a theme name.

A single granule update can be performed by providing the granule ID, expiration date, and an optional retention priority via command-line input. Multiple granules can be updated by providing, via command line an input file, which contains a list of granule IDs, expiration date and an optional retention priority for each granule to be updated. To update the granules associated with a specific "theme", the operator must specify a valid theme name, expiration date and the optional retention priority via command line. When updating the granules associated with a theme, the utility updates the expiration date of a granule associated with that theme if and only if the new expiration date specified is later than the current expiration date of the granule, and the retention priority of a granule associated with that theme if and only if the new retention priority specified is higher than the current retention priority of the granule.

The utility, by default, displays summary information to the operator and prompts for confirmation before executing the update. All warning/error messages and confirmation prompts can be suppressed using the *noprompt* option. This option is suitable for running the utility as a background process, as a cron job, or other automated tasking.

No specific automatic recovery mechanism is provided for this utility. If there is a database fault, system fault, or the utility was inadvertently interrupted, it is sufficient for the operator to simply re-run the utility providing the same command-line parameters.

Input is provided via an input file.

4.9.7.1 Quick Start using the Update Granule Utility

The Update Granule Utility is a Perl script residing in `/usr/ecs/<mode>/CUSTOM/utilities` directory. The utility can be run using the following command line syntax.

```
EcDlUpdateGranule.pl <mode> [-noprompt] -file <fileName>
```

OR

```
EcDlUpdateGranule.pl <mode> [-noprompt]
    -grnid <granuleID> | -theme <themeName>
    -exp <expirationDate>
    [-ret <retentionPriority>]
```

Table 4.9.7-1 provides a description of the command line parameters.

Table 4.9.7-1. Command Line Parameters of the Update Granule Utility

Parameter Name	Description
<mode>	An input parameter specifying the mode of operation. This must be the first parameter passed, and it must be a valid, existing Data Pool mode with a format like OPS or TS1.
-noprompt	Directs the utility to run using the "noprompt" display option. The utility suppresses the displaying of all messages to the screen and updates the granules without prompting the operator for confirmation. This option is suitable for running the utility as a background process.
-file <filename>	An input parameter specifying the file to be used to input granule data to the utility. <filename> is the name of a file containing a list of granule triplets. Each triplet is a separate line entry in the input file and consists of granule ID, expiration date, and an optional retention priority. The granule ID in each triplet corresponds to the granule ID in the DIGranules table in the Data Pool database.
-grnid <granule ID>	An input parameter specifying the granule information to be entered manually at the command line. The <granule ID> parameter must be a positive integer containing no more than 16 digits and corresponds to the granuleid in the DIGranules table in the Data Pool database.
-theme <theme name>	An input parameter specifying a valid theme name. The theme name is a character string and must match an existing theme name in the Data Pool inventory and is case sensitive. Note: If the theme name contains spaces like "Volcano Eruption", it <i>must</i> be enclosed in single or double quotes, or only the first word of the title is used (an error could also occur when running the script).
-exp <expiration Date>	An input parameter specifying the new expiration date to be applied to the science granule indicated by the <granule ID>. The <expiration Date> has the format YYYY/MM/DD.
-ret <retention priority>	An input parameter specifying the new retention priority to be applied to the science granule indicated by the <granule ID>. The <retention priority> must be a positive integer within the range of 1 to 255, 1 being the lowest retention priority and 255 being the highest retention priority.

4.9.7.1.1 Update Granule Utility Commands

Below are some examples for invoking this utility:

1. `EcDlUpdateGranule.pl <mode> -file <file name>` Updates the granules provided via an input file. The utility displays all summary information and prompts the operator to confirm the update.
2. `EcDlUpdateGranule.pl <mode> -grnid <granule id> -exp <expiration date> -ret <retention priority>` Updates the expiration date and retention priority for a single

Table 4.9.7-2. Configuration Parameters

Parameter Name	Description
SYB_USER	The user name for the Sybase connection.
SYB_SQL_SERVER	The name of the SQL server for this Sybase connection.
SYB_DBNAME	The name of the database you intend to connect to.
PGM_ID	Program ID used for connecting to the Data Pool database.
NUM_RETRIES	The number of times the utility attempts to connect to the database before exiting. The recommended default is 5.
SLEEP_SEC	The number of seconds the utility waits ('sleep') between connection attempts. The recommended default is 10.

4.9.7.2 Update Granule Utility Main Screen

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

4.9.7.3 Required Operating Environment

The Update Granule Utility runs on Linux platforms.

4.9.7.4 Databases

Table 4.9.7-3 lists the supporting products that this tool depends upon in order to function properly.

Table 4.9.7-3. Interface Protocols

Product Dependency	Protocols Used	Comments
Data Pool database	SQL	Via SQL server machines
Perl DBI	DBD::Sybase	Requires proper install of Perl 5.005

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 of the errors that can occur are the database server is unavailable, the connection to the database was dropped, or there was an error executing the stored procedure. In the event of a Sybase-sourced error, the utility immediately stops running.

In the event a connection to the Data Pool database can not be established, the utility can repeatedly attempt to connect to the database, depending on how the configuration file was set (see Section 4.9.7.4.2). If, for example, NUM_RETRIES was set to 5 and SLEEP_SEC was set to 10, this means it tries to connect 5 times, and waits 10 seconds before each attempt – a total of 50 seconds if all attempts are unsuccessful.

4.9.7.5 Special Constraints

The Update Granule Utility runs only if the Data Pool database server is running and if the database is available. It also assumes the stored procedures are present.

4.9.7.6 Outputs

Output of update events and errors is always appended to a single log file. Unless the *noprompt* option is specified on the command line, confirmation prompts and information are always displayed to the screen.

4.9.7.7 Event and Error Messages

By default, confirmation messages are displayed to the operator, as well as any error conditions that may arise. Error messages are also written to the log files. With the *noprompt* option, all messages to the screen are suppressed and only written to the log file. Command-line syntax errors are displayed regardless of the *noprompt* option.

The utility produces a log file called `EcDIUpdateGranule.log` in the `/usr/ecs/<mode>/CUSTOM/logs` directory. 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.

4.9.7.8 Reports

None.

This page intentionally left blank.

4.9.8 Data Pool Access Statistics Utility (DPASU) – Rollup Scripts

The Data Pool Access Statistics Utility (hereafter referred to as “DPASU”) provides the ECS Operations Staff with several capabilities related to collecting access statistics for the Data Pool database. The DPASU encompasses two types of scripts: rollup and maintenance. The rollup scripts read and parse access logs to compile statistics and store those records in the Data Pool database, while the maintenance scripts backup, restore, and delete data in the related Data Pool database tables.

These scripts may be run on the command-line, and must be run with an operations mode. Details and instructions on how to run and configure these scripts are provided in subsequent sections.

4.9.8.1 Data Pool Access Rollup Scripts

The Data Pool access rollup scripts provide the ECS Operations Staff with the capability to parse the Data Pool web access and FTP logs for Data Pool access information and store the access information in the Data Pool database. For each Data Pool file access found in the FTP or web access logs, the rollup scripts store into the Data Pool database (in the DIGranuleAccess table) the time of access, the corresponding granule ID, the file type (metadata, browse or science), the file size (in bytes), the access type (FTP or HTTP), and the age of granule at access time (i.e., the number of days the granule has been in the Data Pool at the time of access). Such information collected over a period of time can provide useful statistical information regarding the Data Pool access patterns and provides insight into planning future support.

There are three Data Pool access rollup scripts: one for rolling up accesses to the Data Pool via the Web Access GUI (EcDIRollupWebLogs.pl), and the other two for rolling up accesses to the Data Pool via anonymous ftp. The DAAC chooses which of the two ftp rollup scripts to use depending on the kind of firewall in place at that DAAC. The EcDIRollupFwFtpLogs.pl script is used at DAACs with an EMD-supplied PORTUS firewall; the EcDIRollupWuFtpLogs.pl script is used at DAACs with any other kind of firewall (e.g., at ASDC). . All three rollup scripts are installed and run on the Data Pool host x4dpl01.

For DAACs with a PORTUS firewall, an additional EMD-supplied script runs once per day on the firewall host. This script produces a subset of the firewall log containing only ftp accesses to the x4dpl01 host. This subset of the firewall log is transferred to the `usr/ecs/OPS/COTS/firewall/logs` directory on the x4dpl01 host, with a file name of `datapoolftplog.<n>`, where `<n>` is an integer from 0 to 6, representing the day of the week when the log extract process was run. The EcDIRollupFwFtpLogs.pl script on the x4dpl01 host parses this `datapoolftplog.<n>` file.

For DAACs which do not have use a PORTUS firewall, the EcDIRollupWuFtplogs.pl parses the wu-ftp log on the x4dpl01 host (`/var/adm/xferlog`).

Each rollup script is a command-line utility allowing the operator to optionally pass input parameters. Operationally, the rollup script is run in a cron job, with the crontab file specifying when the rollup script shall start its daily execution. Each time the rollup scripts are run, they roll

up the Data Pool accesses that occurred over a specified 24-hour period in the past and store them into the Data Pool database.

By default, the start date of the 24-hour rollup period is one day prior to the date when the rollup script is executed. However, the rollup scripts allow a non-default start date of the rollup period to be specified via command line. This capability is provided to allow the DAAC operator to run the rollup scripts manually to compile statistics for a particular date for which the regular cron job may not have run for some reason.

To allow flexibility for each DAAC to specify the start time of the 24-hour rollup period, the start time is provided as a configuration parameter (ROLLUP_START_TIME). (See Section 4.9.8.3.3 for description of configuration parameters.) For example, if the start time is configured as 0:00 hours (midnight), then the rollup period will always cover a 24 hour period starting from 0:00 on the rollup start date. If the start time is configured as 6:00 hours, the rollup period always covers a 24-hour period starting from 6AM of the rollup start date.

Each of the Data Pool access rollup scripts work as follows. The script first parses the specified FTP or Web access log(s) for Data Pool access events. After the log files are parsed, the captured data is written to a temporary “flat file” – a tab-delimited text file. This file gets exported to the Data Pool database, where it is stored in a temporary table (DIftpAccessLog or DIWebAccessLog). The rollup script then uses information in the temporary access log tables and other Data Pool database tables to determine the information to be written to the DIGranuleAccess table. The flat file is removed and an entry is made into the DIAccessRollup table to keep a record of which periods have been successfully “rolled up” to prevent the accidental reprocessing of that period.

Normally the Data Pool access rollup scripts are run by cron. Unless the cron job was completely successful, no entry is made into the DIAccessRollup record table to indicate the rollup period was processed. Therefore, the DAAC operator is able to reprocess that period by manually running the rollup scripts from the command line.

4.9.8.1.1 Invoking the Data Pool Access Rollup Scripts from the Command Line Interface

Entering the following commands run the rollup scripts:

% EcDIRollupWebLogs.pl <command line parameters>

% EcDIRollupFwFtpLogs.pl <command line parameters>

% EcDIRollupWuFtpLogs.pl <command line parameters>

There are various optional and required command line parameters used in combination with each other. Table 4.9.8-1 provides a description of these parameters.

Table 4.9.8-1. Command Line Parameters of the DPASU

Parameter Name	Necessity	Description
<MODE>	Required	Indicates the Data Pool MODE the script is to run in. This parameter has the following constraints: It must be the first parameter with no label The <MODE> must imply a valid directory path An example of a <MODE> is OPS, TS1, TS2, et cetera.
-noprompt	Optional	Turns on the “noprompt” display mode – suppressing all output to the screen. This should be used in cron jobs or other scenarios where output to a display is not desired. The default display mode writes messages to the screen.
-flatfile <path/file>	Optional	Provides an alternative path/file name for the flat file produced by the parser. This is only useful with the –nodelete option.
-nodelete	Optional	Prevents the flat file from being deleted once the DPASU completes its run.
-fwftp <path/file(s)>	Optional	Indicates an alternative FTP log path/file(s) to be used instead of the configured default path/file(s). Use of a wildcard character is permitted in the file name, but if a wildcard is used, the path/file name must be enclosed in quotes (e.g. “/usr/ecs/OPS/COTS/firewall/logs/datapoolftplog.*” or “/home/allmode/archive/xferlog.0”). For EcDIRollupFwFtpLogs.pl or EcDIRollupWuFtpLogs.pl only.
-web <path/file(s)>	Optional	Indicates an alternative web log path/file(s) to be used instead of the configured default path/file. Use of a wildcard character is permitted in the file name, but if a wildcard is used, the path/file name must be enclosed in quotes (e.g., “usr/ecs/OPS/CUSTOM/logs/WebAccess.log”). For EcDIRollupWebLogs.pl only
-start <date>	Optional	Indicates the alternative start date for the rollup period with the format YYYY/MM/DD. This can be used to process previously uncovered periods. The default date is the date cron actually runs the DPASU on a day-to-day basis.

4.9.8.1.2 Default Rollup Period

By default, the rollup period begins 24 hours before the current date (i.e., the date on which the rollup script is run, either manually or by cron) plus the configured rollup start time (see Section 4.9.8.3.3 for configuration of the ROLLUP_START_TIME parameter). For example, if the rollup script is run on September 23, and the configured rollup start time is “2:00”, the Rollup period begins on September 22 at 2:00 a.m. and ends on September 23 at 1:59 a.m.

This means the Rollup script scans the specified log(s) for all entries having an access time between Sep 22 2:00 a.m. and Sep 23 1:59 a.m. Note that the 24-hour rollup period must be in the past as compared to the time the rollup script is run. The rollup script does not execute and terminates with an error if it detects the rollup period that was specified spans into a future time.

4.9.8.1.3 Specifying an Optional Start Date of Rollup Using `-start` Option

The rollup scripts allow an optional rollup period start date to be specified via command line using the `-start` option. With this option, a valid date must be entered in the following ordinal format:

YYYY/MM/DD

Because this is an optional *start* date, the end date gets 24 hours *added* to it. Therefore, if at the command-line the operator enters `-start 2002/09/18` (presumably because cron failed to run the DPASU Rollup on that day), the rollup scripts look for all entries with access times between 09/18/2002 02:00 and 09/19/2002 01:59.

The scripts check the input date at the command line to make sure it is valid before it parses the logs.

4.9.8.1.4 Running Data Pool Access Rollup Scripts with *cron*

The Data Pool access rollup scripts are run by cron on a daily basis at a consistent time of day. There are a number of factors to consider when determining at what time to run the rollup scripts each day. Factors are:

- a. The rollup scripts should be run at a time of day that is **AFTER** the configured rollup start time. (A good rule of thumb is to have the scripts run at least one half hour after the rollup start time.) This ensures the 24-hour rollup period has completed at the time the rollup scripts are run.

Example 1: If the rollup start time is 2:00 a.m., the cron should run the rollup scripts at a time after 2:30 a.m.

Example 2: If the rollup start time is 22:00, the cron should run the rollup scripts at a time after 22:30, but *not* after 23:59 because any time after that is the next day.

- b. It is recommended the rollup scripts be run by cron at a time of day when Data Pool access activity is low – e.g., during the early morning hours.
- c. The rollup scripts should be run **BEFORE** the daily Data Pool Cleanup script is run, to minimize chances that information about files accessed during the 24-hour rollup period has been removed from the Data Pool database. (If this information has been removed, the rollup scripts are unable to write information for those files in the `DIGranuleAccess` table.)
- d. The rotation/renaming times of the Web Access and FIREWALL FTP log files and the time the corresponding rollup script is run must be taken into consideration in determining, which log files to parse and whether to use a wildcard in the specification of the log file path.

For example, consider the case where the FIREWALL FTP log is rotated/renamed each day at 01:00, and the FTP rollup script is run at 03:00 with a rollup start time of 02:00. When the rollup script is run at 03:00 on September 22, 2002, the rollup period is

September 21, 2002 02:00 through September 22, 2002 01:59. The FIREWALL FTP log (e.g. datapoolftplog.1) which was rotated/renamed at 01:00, now only contains accesses for the time period September 22, 2002 01:00 through September 22, 2002 03:00 (the current time). The previous FIREWALL FTP log (e.g. datapoolftplog.0), contains accesses for the time period September 21, 2002 01:00 through September 22, 2002 00:59. To capture information for the entire rollup period, the ftp rollup script must be configured to parse both the datapoolftplog.1 and datapoolftplog.0 .

This may be accomplished either by running the ftp rollup script twice, once against datapoolftplog.1 and once against datapoolftplog.0, or by running the script once and using a wildcard to specify the ftp log path. (Note that wildcard path names must be enclosed in quotes if used on the command line with the `-web` or `-fwftp` command line parameters, but do NOT need to be enclosed in quotes if used with the configuration parameters `WEB_LOG_PATH` or `FTP_FIREWALL_LOG_PATH`. See sections 4.9.8.1.5 and 4.9.8.1.6)

- e. To prevent or minimize the chances of database contention, it is recommended the daily cron job for rolling up FIREWALL FTP access logs and the daily cron job for rolling up web access logs be staggered, so the two rollup scripts do not run at the same time.

In the case that *cron* fails to run the Data Pool access rollup scripts on a given day, the operator can manually run either script, specifying the date(s) missed using the `-start` command line parameter.

4.9.8.1.5 Specifying Alternative Paths for FIREWALL FTP or Web Access Logs

The operator can specify an access log file path different than that specified in the configuration file by using the `-web` and/or `-fwftp` command line parameters. If alternative access log file paths are used with the command line options `-web` and `-fwftp`, any wildcards used to indicate multiple files matching a pattern need to be enclosed in quotes. If they are not, the rollup scripts cannot use the files you intended. The scripts *internally* (i.e., not the shell) match all files indicated by wildcards on the command line.

For example, the path

```
/usr/ecs/OPS/COTS/firewall/logs/datapoolftplog.*
```

must be enclosed in quotes as follows

```
"/usr/ecs/OPS/COTS/firewall/logs/datapoolftplog.*"
```

to ensure the wildcard character (*) is properly passed

Keep in mind that quotes around wildcard path names are only required on the command line; they are NOT required in the configuration file.

4.9.8.1.6 Intermediate Flat File

The rollup scripts create an intermediate flat file from the log entries that contain all the data that will be exported to the database via bulk copy procedure (bcp). Normally, this file is temporarily placed in a data directory and then deleted, once the scripts have completed running. The operator can keep that flat file by specifying the **-nodelete** option. By default, the intermediate flat file is created in the following directory:

```
usr/ecs/<MODE>/CUSTOM/data/DPL/
```

The operator can specify an alternate path and name for this file on the command line using the **-flatfile** option.

4.9.8.1.7 Command-line Examples

Here are some examples of executing the Data Pool access rollup scripts from the command line.

Example 1:

```
EcDlRollupWebLogs.pl OPS -noprompt -nodelete -start 2002/12/22
```

Run Web rollup script in -noprompt display mode for an optional 24-hour rollup period starting from December 22, 2002, at the configured rollup start time. The -nodelete option prevents the flat file from being erased upon completion.

Example 2:

```
EcDlRollupFwFtpLogs.pl OPS -noprompt
```

```
EcDlRollupWuFtpLogs.pl OPS -noprompt
```

Run FIREWALL FTP / wuftp rollup script in -noprompt display mode for the default 24-hour rollup period starting from yesterday at the configured rollup start time. This example is typical of syntax used in the crontab file.

Example 3

```
EcDlRollupFwFtpLogs.pl OPS -start 2002/02/15 -fwftp "/usr/logs/*.log "
```

```
EcDlRollupWuFtpLogs.pl OPS -start 2002/02/15 -fwftp  
"/home/allmode/archive/xferlog.0"
```

Run FIREWALL FTP Rollup script in prompted mode, for an optional 24-hour rollup period starting from February 15, 2002, at the configured rollup start time, but use the FIREWALL FTP access logs stored in an alternative path /usr/logs.

4.9.8.2 Data Pool Access Statistics Main Screen

The Data Pool Access Statistics does not have a main screen. It is a command line interface.

4.9.8.3 Required Operating Environment

The Data Pool access rollup scripts run in a Linux operating environment.

4.9.8.3.1 Interfaces to supporting products

Table 4.9.8-2 lists the supporting products that this tool depends upon in order to function properly.

Table 4.9.8-2. Interface Protocols

Product Dependency	Protocols Used	Comments
Data Pool database	SQL	Via SQL server machines
Perl DBI	DBD::Sybase	Requires proper install of Perl

4.9.8.3.2 Access Log File Formats

The Data Pool access rollup scripts are dependent on a particular format of both the FIREWALL FTP and Web access logs. If the format of these log files changes, it is quite possible the scripts can incorrectly read certain fields and consequently provide incorrect or misleading rollup reports, and can even prevent the scripts from running at all. It is important to have the rollup script code adjusted if the FIREWALL FTP or WEB access log formats change in any way. Subsequent sections provide format details for these access logs.

4.9.8.3.3 Configuration Files for Data Pool Access Rollup Scripts

The Data Pool access rollup scripts use configuration files containing details about how to connect to Sybase and about where the log files exist. The file *EcDIWebRollup.CFG* contains the configuration parameters for the Web Rollup script. The file *EcDIftpFwRollup.CFG* contains the configuration parameters for the Firewall FTP Rollup script. The file *EcDIWuFtpRollup.CFG* contains the configuration parameters for the wu-ftp Rollup script. Without the configuration files, the scripts can not run. Each configuration file must be a plain text ASCII file, which has the following format, not necessarily in this order:

```
SYB_USER = <string>
SYB_SQL_SERVER = <string>
SYB_DBNAME = <string>
NUM_RETRIES = <integer>
SLEEP_SEC = <integer>
WEB_LOG_PATH = <path and file name>
FTP_FIREWALL_LOG_PATH = <path and file name> (for EcDIRollupFwFtpLogs.pl)
FTP_LOG_PATH = <path and file name> (for EcDIRollupWuFtpLogs.pl)
ROLLUP_START_TIME = <time of day>
```

Table 4.9.8-3 describes the individual configuration parameters mentioned above.

Table 4.9.8-3. Data Pool Access Configuration Parameters for Rollup Scripts

Parameter Name	Description
SYB_USER	The user name for the Sybase connection.
SYB_SQL_SERVER	The name of the SQL server for this Sybase connection.
SYB_DBNAME	The name of the Data Pool database you intend to connect to.
NUM_RETRIES	The number of times the utility attempts to connect to the database. The recommended default is 5.
SLEEP_SEC	The number of seconds the utility waits ('sleep') between connection attempts. The recommended default is 10.
WEB_LOG_PATH	The path and file name for the Data Pool Web Access custom code log. This parameter is valid for EcDIWebRollup.CFG only. The web rollup script automatically uses this path (and file or files) if an alternative one is not explicitly provided. Wildcards are permitted and do not need to be enclosed in quotes.
FTP_FIREWALL_LOG_PATH	The path and file name for the default ftp access log. This parameter is valid for EcDIFwFtpRollup.CFG only. The FIREWALL FTP rollup script automatically uses this path (and file or files) unless an alternative one is explicitly provided using the – fwftp command line parameter. Wildcards are permitted and do not need to be enclosed in quotes.
FTP_LOG_PATH	The path and file name for the default wu-ftp access log. This parameter is valid for EcDIWuFtpRollup.CFG only. The rollup script for the wu-ftp log automatically uses this path (and file or files) unless an alternative one is explicitly provided using the – fwftp command line parameter.
ROLLUP_START_TIME	The configurable time of day the rollup script uses as the start time of the 24 hour rollup period. 24-hour time must be used for this entry, e.g., "3:00" or "18:00".

4.9.8.4 Databases

The Data Pool Access Statistics utility uses the Sybase ASE Server.

4.9.8.5 Special Constraints/Dependencies

The Data Pool access rollup scripts function only if the Data Pool database server is running and if the Data Pool database is available. The rollup scripts also assume the required stored procedures are present in the Data Pool database.

Special modules are also required to run this utility. If those modules are not present or are located in an unfamiliar directory, it fails to run. Table 4.9.8-4 describes the modules required to run the rollup scripts.

Table 4.9.8-4. Data Pool Access Special Modules

Name	Description
EcDIDbInterface.pm	Database interface and connection module.
EcDIDateTime.pm	Date/time grabber with millisecond resolution.

4.9.8.6 Outputs

Rollup information is entered in the Data Pool database in the DIGranuleAccess table. If the `-noprompt` option is not on, examining status and other messages are printed to the screen. Log messages are also recorded (see below).

4.9.8.7 Event and Error Messages

All event and error messages generated from the rollup scripts are written to the respective log files. When the scripts are run in the prompted mode (default), the messages are also displayed to the screen in addition to writing to the logs.

4.9.8.8 Reports

None.

4.9.8.9 Recovery Procedures

In the case that *cron* fails to run the Data Pool access rollup scripts on a given day, the operator may manually run either script, specifying the date(s) missed using the `-start` command line parameter. See Section 4.9.8.1.3 for details.

This page intentionally left blank.

4.9.9 Data Pool Access Statistics Utility (DPASU) – Maintenance Scripts

The Data Pool Access Statistics Utility (hereafter referred to as “DPASU”) provides the ECS Operations Staff with several capabilities related to collecting access statistics for the Data Pool database. The DPASU encompasses two types of scripts: rollup and maintenance. The rollup scripts read and parse access logs to compile statistics and store those records in the Data Pool database, while the maintenance scripts backup, restore, and delete data in the related Data Pool database tables.

These scripts may be run on the command-line, and must be run with an operations mode. Details and instructions on how to run and configure these scripts are provided in subsequent sections.

4.9.9.1 Data Pool Access Maintenance Scripts

The Data Pool access maintenance scripts are operational support tools used for archiving, deleting, and backing up granule access data in the Data Pool database. Each of these scripts can be run on the command line and connects to the Data Pool database to process data contained therein. These scripts are installed and run on the Data Pool database host (x0acg0n), in the /usr/ecs/<mode>/CUSTOM/dbms/DPL directory. All of these scripts involve access to the Data Pool tables DIGranuleAccess, DIGranuleSubscription, and DIAccessRollup.

Archive Utility - *DI DbArchiveAccessStat*

This script archives data contained in DIGranuleAccess, DIGranuleSubscription, and DIAccessRollup by writing this data to an ASCII file based on an operator-specified time range.

Delete Utility - *DI DbDeleteAccessStat*

This script removes data contained in DIGranuleAccess, DIGranuleSubscription, and DIAccessRollup based on an operator-specified time range.

Restore Utility - *DI DbRestoreAccessStat*

This script restores data archived by the archive utility (contained in the ASCII file) into DIGranuleAccess, DIGranuleSubscription, and DIAccessRollup.

4.9.9.2 Invoking the Maintenance Utilities from the Command Line Interface

Entering the following commands start the maintenance utilities:

> **UtilityName <command line parameters>**

There are various command line parameters used in combination with each other. Table 4.9.9-1 provides a description of these parameters

Table 4.9.9-1. Command Line Parameters of the DPASU Access Maintenance Scripts

Parameter Name	Description
<MODE>	The mode in which the utility is being executed.
<STARTDATE>	The beginning date time range for archiving, deleting, or restoring the data. The format for this parameter is yyyyymmdd.
<STOPDATE>	The ending date time range for archiving, deleting, or restoring the data. The format for this parameter is yyyyymmdd.
<ARCHIVEDIR>	The absolute path where the generated ASCII files are stored when archiving or restoring data (this parameter only applies to the archiving and restoring scripts). The ASCII files are generated from the archive utility. The file name follows the convention <tablename>.dat.<startdate><stopdate>.
<USERNAME>	The Sybase login name.
<SERVER>	The Sybase Server where the Data Pool database located.
<DBNAME>	The name of the Data Pool database.

The parameters shown here are those used for all of the maintenance scripts. See the “Utility Commands” section for each script for specific usage. Please note that these parameters must be provided in the exact order as shown in the examples below.

Each of the scripts prompts the user to enter the password for the Sybase login.

4.9.9.3 Archive Utility Commands

The archive utility must be run with the following parameters in this exact order. There is only one command-line permutation:

```
DlDbArchiveAccessStat <MODE> <STARTDATE> <STOPDATE> <ARCHIVEDIR> <USERNAME> <SERVER> <DBNAME>
```

Example:

```
DlDbArchiveAccessStat OPS 20020405 20020505 /home/DBArchive/DataPool/ Labuser01
SybSQL_srvr DataPool_DB
```

The above example archives data to files and stores them in a specified directory.

4.9.9.4 Delete Utility Commands

The Delete Utility must be run with the following parameters in this exact order. There is only one command-line permutation:

```
DlDbDeleteAccessStat <MODE> <STARTDATE> <STOPDATE> <USERNAME> <SERVER> <DBNAME>
```

Example:

```
DlDbDeleteAccessStat OPS 20020912 20020913 Labuser01 SybSQL_srvr DataPool_DB
```

The above example deletes data in a specified time range.

4.9.9.5 Restore Utility Commands

The Restore utility must be run with the following parameters in this exact order. There is only one command-line permutation:

```
DlDbRestoreAccessStat <MODE> <STARTDATE> <STOPDATE> <ARCHIVEDIR> <USERNAME> <SERVER>
<DBNAME>
```

Example:

```
DlDbRestoreAccessStat OPS 20020405 20020505 /home/DBArchive/DataPool/ Labuser01
SybSQL_srvr DataPool_DB
```

The above example restores data in a specified time range from a specified archive directory.

4.9.9.6 Data Access Statistics Main Screen

The Data Pool Access Statistics utility does not have a main screen. It has a command line interface.

4.9.9.7 Required Operating Environment

The maintenance utilities run on a Linux platform.

4.9.9.7.1 Interfaces to Supporting Products

Table 4.9.9-2 lists the supporting products that these tools depend upon to function properly.

Table 4.9.9-2. Interface Protocols

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

4.9.9.8 Databases

The Data Pool Access Statistics utility uses the DataPool database.

4.9.9.9 Special Constraints

The maintenance utilities run only if the Data Pool database is available and the Sybase server is running.

4.9.9.10 Outputs

There are no outputs from the maintenance scripts, except the error messages to the log.

4.9.9.11 Event and Error Messages

All error messages are written to the log files, which are DlDbRestoreAccessStat.log, DlDbArchiveAccessStat.log and DlDbDeleteAccessStat.log.

4.9.9.12 Reports

None.

4.9.10 Most Recent Data Pool Inserts Utility

The most recent data pool insert utility provides the ECS Operations Staff with a command-line interface for listing the most recent additions to data pool. Output of this utility is a set of files that a user could download and quickly inspect for new Data Pool additions. In operation, this utility would be configured to run as a cron job.

Utility takes in a date command-line parameter representing the day user is interested in. Files inserted into Data pool on this day would be listed in the output files. If a date is not provided, the utility uses the previous date as a default with a time range of midnight to midnight.

Since this utility requires connection to database, there will be a configuration file containing all information needed for accessing database. Further more, all error messages would be written to an error log file in /usr/ecs/<MODE>/CUSTOM/logs directory.

4.9.10.1 Using Most Recent Data Pool Inserts Utility

Utility would mainly be run as a cron job.

For command line usage, utility is started by entering the following:

```
> EcDIMostRecentInsert.pl <MODE> [-insertDate <YYYY/MM/DD>]
```

Command line parameters and corresponding descriptions that could be used with Most Recent Data Pool Inserts utility are listed in Table 4.9.10-1.

**Table 4.9.10-1. Command Line Parameter
Most Recent Data Pool Inserts Utility**

Parameter Name	Required	Description
MODE	Yes	An input parameter that specifies the mode of operation. This must be the first parameter passed, and it must be a valid, existing Data Pool mode such as OPS or TS1.
insertDate	No	An optional parameter specifying date in which user is interested. If date parameter is not present, previous day date is used by default. Date format is YYYY/MM/DD.

Executing this utility requires the mode as the first input parameter else a fatal error would be returned. If the date parameter is present, it must conform to the following format YYYY/MM/DD. Incorrect input parameters would result in errors being written to log file.

4.9.10.2 Most Recent Data Pool Inserts Utility Commands

Examples of how to use this utility is shown below:

1. **EcDIMostRecentInsert.pl OPS -insertDate 2003/02/28**

Queries database and creates file containing listings Data Pool additions for day 2003/02/28 for OPS mode.

2. **EcDIMostRecentInsert.pl OPS**

Since “-insertDate” command line parameter is not entered, the previous day is

used by default. Queries database and creates files listing recent additions to database for previous day for OPS mode.

4.9.10.3 Required Operating Environment

The O/S requirements is Linux 2.x.

4.9.10.4 Interfaces and Data Types

Table 4.9.10.2 lists the supporting products that this tool depends upon in order to function properly.

Table 4.9.10-2. Interface Protocols

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

4.9.10.5 Configuration File Format – EcDIMostRecentInsert.CFG

The “config” file contains vital details about how to connect to the Sybase database. Without this file, the utility can not run. The config file must be a single-entry plain text ASCII file, which has the following format:

```
SYB_USER = <string>
SYB_SQL_SERVER = <string>
PGM_ID = <string>
SYB_DBNAME = <string>
NUM_RETRIES = <integer>
SLEEP_SEC = <integer>
```

Breakdown of the individual parameters:

Parameter Name	Description
SYB_USER	The user name for the Sybase connection.
SYB_SQL_SERVER	The name of the SQL server for this Sybase connection.
SYB_DBNAME	The name of the database you intend to connect to
PGM_ID	Program ID used for connecting to the Data Pool database.
NUM_RETRIES	The number of times the utility will attempt to connect to the database before exiting. The recommended default is 5.
SLEEP_SEC	The number of seconds the utility will wait ('sleep') between connection attempts. The recommended default is 10.

4.9.10.6 Special Constraints

EcDIMostRecentInsert utility runs only if the Data Pool database server is running and if the database is available. It also assumes the stored procedures are present.

4.9.10.7 Outputs

Output of this utility is a set of files. One file located at top level Data Pool directory named DPRecentInserts_<YYYYMMDD> and a file in each of the collection level directories named DPRecentInserts_<ShortName>_<VersionID>_<YYYYMMDD>.

File DPRecentInserts_<YYYYMMDD> contains distinct ShortNames and VersionIds while file DPRecentInserts_<ShortName>_<VersionID>_<YYYYMMDD> contains ShortName, VersionId and fully qualified.

Note: The EcDIMostRecentInsert.pl would shut down and log an error message if it is unable to create a file at the top level data pool directory. If it is unable to create file at the collection level directory, program would log an error message and continue with processing other valid directories. Also, each time utility runs with the same input argument, the contents of the previously created file are over written.

4.9.10.8 Event and Error Messages

Usage and processing error messages are written to log file.

4.9.10.9 Reports

None.

4.9.10.10 Logs

The utility produces a log file called *EcDIMostRecentInsert.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.

4.9.10.11 Recovery

If there is an execution failure as a result of database server or system shut down, operator simply re-runs the script. This would create a new set of files (i.e. over writing previous ones) listing additions to Data Pool for the specified insert date.

4.9.10.12 Sybase Error Handling

The utility is highly dependent on Sybase server. Connection failure to Sybase Server would simply result in program termination and error logged to log file.

Note: The utility may repeatedly attempt to connect to the database, depending on how the configuration file was set. As an example, NUM_RETRIES set to 5 and SLEEP_SEC set to 10 in configuration file would mean the utility would try to connect 5 times, and will wait 10 seconds before each attempt.

This page intentionally left blank.

4.9.11 Data Pool Collection-to-Group Remapping Utility

The Data Pool Collection-to-Group Remapping Utility will allow DAAC Operations to re-assign a Data Pool collection to a collection group different from the one to which the collection was assigned originally. This command line utility will be used to remap collections between groups.

Note: Prior to using this utility you must set the “Insert Enabled Flag” to off using the Data Pool Maintenance GUI for the source collection.

4.9.11.1 Using the EcDIRemap utility

The Data Pool Collection-to-Group Remapping Utility is invoked as follows:

```
>EcDIRemap.pl <mode> -esdt <source collection name> -version <source collection version> -oldgrp <group to which the collection currently belongs> -newgrp <group to which the current collection will be mapped>
```

There are various command line parameters that are used in combination with each other. Table 4.9.11-1 provides a description of these parameters.

Table 4.9.11-1. Command Line Parameters

Parameter Name	Required	Description
mode	Yes	The mode in which the utility will run.
esdt	Yes	Specifies the name of the source collection that is being remapped
version	Yes	Specifies the version of the source collection that is being remapped
oldgrp	Yes	Specifies the name of the source collection group that contains the source collection
newgrp	Yes	Specifies the destination group where the source collection is to be mapped

Section 4.9.11.3 provides some examples along with detailed explanations for executing this utility.

4.9.11.2 Data Pool Collection-to-Group Remapping Utility Configuration File

The Data Pool Collection-to-Group Remapping utility uses a configuration file, EcDIRemap.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 4.9.11-2 describes the configuration parameters.

Table 4.9.11-2. Configuration Parameters

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

4.9.11.3 Examples for Remapping a Collection

1. Remap a collection MOD29, Version 4 from the group MOST to the group MOSS in the OPS mode:

```
EcDlRemap.pl OPS -esdt MOD29 -version 4 -oldgrp MOST -newgrp MOSS
```

The utility will remap the directory from the old collection MOD29.004 beneath the MOST group to the collection MOD29.004 under the MOSS group. The Data Pool database inventory will be updated to reflect the new location of the files.

Previous directory structure before remapping (example):

```
/datapool/OPS/user/MOST/MOD29.004/2000.10.31/MOD29.A2000305.h11v11.004.20012651  
13249.hdf
```

New directory structure following the remapping:

```
/datapool/OPS/user/MOSS/MOD29.004/2000.10.31/MOD29.A2000305.h11v11.004.20012651  
13249.hdf
```

4.9.11.4 Required Operating Environment

The Group remapping utility will run on a Linux platform.

4.9.11.5 Interfaces and Data Types

Table 4.9.11-3 lists the supporting products that this tool depends upon in order to function properly.

Table 4.9.11-3. Interface Protocols

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

4.9.11.6 Special Constraints

The Data Pool Collection-to-Group Remapping utility requires that the “Insert Enabled Flag” be turned off using the Data Pool Maintenance GUI for the source collection prior to running the

utility. If this is not done, the utility will provide an error message to the user indicating this condition and promptly exit. Also the utility expects that the group to which the user is mapping the collection exists in the Data Pool database. In addition, the user is to be aware that the utility does not allow remapping the Browse (Browse.001) collection. Data Pool functionality assumes that the browse collection is always located in the group “BRWS”. Also, the utility doesn’t allow users to map any collection to the BRWS group. The user is given an error message and the utility exits if either of these cases is true. The utility checks to see if the given collection is part of the old or source group. If not, the utility informs the user and exits. The Group Mapping utility runs only if the Data Pool database server is running and if the database is available. It also assumes the stored procedures are present.

4.9.11.7 Outputs

Output of update events and errors will be always appended to a single log file.

4.9.11.8 Event and Error Messages

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

4.9.11.9 Reports

None.

4.9.11.10 Logs

The utility produces a log file called EcDIRemap.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.

4.9.11.11 Recovery

The EcDIRemap.pl utility will be able to recover from aborted runs by using the DIRecoveryParameters table to checkpoint its progress. In the event of an aborted run, the utility will read the recovery parameters table to determine at which point the utility left off when it aborted. This will ensure that remappings that were taking place prior to the abort will finish correctly. After recovery processing takes place, the utility will then process the current run by acting on the latest input parameters. For example, if the original command line was:

```
EcDIRemap.pl OPS -esdt MOD29 -version 4 -oldgrp MOST -newgrp MOSS
```

If this run were aborted and a new invocation of the utility was initiated with a different collection and different source and destination groups as follows:

```
EcDIRemap.pl OPS -esdt AST_L1A -version 3 -oldgrp ASTT -newgrp ASTA
```

Then the utility would give a message to the standard output and log indicating a recovery was in process for remapping MOD29.004 collections from the old group (MOST) to the new group (MOSS) was taking place. As soon as the recovery was finished, the utility would indicate that it would now process the remapping of AST_L1A.003 from the ASTT group to the ASTA group.

4.9.11.12 Sybase Error Handling

If a Sybase error occurs, you will most likely see the actual Sybase error string displayed on the screen and in the log. Some errors can be that the database server is unavailable, that the connection to the database was dropped, or that there was an error executing the 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 can not 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 5 and SLEEP_SEC was set to 10, this means it will try to connect 5 times, and will wait 10 seconds before each attempt – a total of 50 seconds if all attempts are unsuccessful.

4.9.12 QA Update Utility

The QA Update Utility (QAUU) is an operational support tool used for updating the values of the QA flags in the ECS inventory. QA updates are applied to the XML metadata files in the metadata archive and DataPool file systems as well as to metadata tables in the DataPool database.

4.9.12.1 Using the QA Update Utility

The QAUU is started by executing the following:

```
$ EcDsAmQAUUStart <mode>
  [ -file <filename> ] [ -noprompt ] [-noExitonError ]
  [-recoverOnly | -abortRecovery | -skipRecovery | -recoverInvestigated]
```

All parameters, except for mode, are optional

<mode>: the mode to run the utility in

-file <filename>: run with a single request file where <filename> is the name of the request file containing the QA updates to be applied. If omitted, all request files in the configured request directory are processed.

-noprompt: if specified, the utility will not prompt the user for confirmations

-noexitonerr: if specified, the utility will not exit on the first error. This allows the operator to determine all errors that may occur during processing.

Recovery options:

These are all mutually exclusive; only one may be specified. (For a discussion of the “investigating” flag, see section 1.1.1.5.) Note that if none of these options are specified, the utility will recover, if necessary, and process new requests:

-recoverOnly: recover and do NOT process new requests (assume we do NOT recover failures flagged as “investigating”)

-abortRecovery: delete all failures (EXCEPT failures flagged as “investigating”) in working table and process new requests

-skipRecovery: flag (don't process) failures to be investigated (InvestigateFlag = 'Y') and process new requests

-recoverInvestigated: recover (including “investigating” failures) and process new requests

4.9.12.2 Required Operating Environment

Linux

4.9.12.3 Input File Format

FileName

The input, or “request”, file name has to follow the following naming convention, depending on the MODE and site where the request is from and when the request is generated:

```
<MODE>_<Site>_QAUPDATE<description>.<YY><MM><DD><hh><mm><ss>
```

where <description> is optional.

The following example shows the filename from site LDOPE for OPS mode at 12:20:30 on Feb. 29, 2008.

```
OPS_LDOPE_QAUPDATE_SetToPassed.20080229122030
```

Note: All the files in the request directory will be processed alphabetically by file name. This guarantees that all the requests coming from the same site will be processed in the right order.

When the DAAC operator copies files from the MODE-dependent FTP site to a MODE-dependent QAUURequest directory, he or she needs to copy all the files with filenames starting with that particular MODE. This way, we can minimize the risk of processing requests intended for a different MODE. QAUU also double checks the file name it processes in regard to the MODE.

In the situation when email is used to send a request, a file that follows the same naming convention should be created and attached to the email. Requests for different MODEs should be sent to different email aliases at each DAAC. The email script (EcDsQAUUEmailScript.pl) checks if the attachment file exists and if the file name follows the naming convention. If failed, the email script sends an email back to the requester indicating an error; otherwise it saves the email in the MODE-dependent QAUURequestDir directory using the same file name as the attachment file.

Note: We can not guarantee that email requests can arrive in sequence according to the time they are produced. Therefore, if the SCF needs to update the same granule(s) again for some reason, it needs to contact the DAAC operator to make sure that the previous requests for the same granule(s) have been finished.

Each request file must contain a **header**, a **body**, and a **footer**.

Request Header

For requests from FTP sites, the header contains one line indicating which site the request is from:

```
From <Site>
```

Example:

```
From LDOPE
```

For requests sent through emails, the headers are automatically generated, standard email headers.

Request Body

The body of the request contains a single **begin** statement, followed by the actual QA update **data** statement(s) that will be applied to the metadata.

The **begin** statement must be the first line in the body and is of this form:

```
begin QAMetadataUpdate {Science | Operational} {LGID | GranuleUR | ESDT}
```

The “Science” or “Operational” column tells the QAUU whether to update science or operational QA flags and explanations. The “LGID”, “GranuleUR” or “ESDT” column tells the QAUU whether the lines that follow the **begin** statement contain local granule ids (LGIDs), granule URs, or ESDT temporal ranges as criteria to look up the granules to be updated.

Each **data** statement takes the following form:

```
<ShortName>TAB<VersionID>TAB<granule search criteria>TAB<measured parameter name> |  
ALL TAB<QA flag value>TAB<QA flag explanation value>
```

Note that the columns for the **data** statement must be delimited by tabs and not spaces, since some of the columns can themselves contain spaces.

<granule search criteria> takes one of the following forms, depending on the LGID/GranuleUR/ESDT column in the **begin** statement.

For LGID, <granule search criteria> = the LGID of a specific granule to be updated.

For GranuleUR, <granule search criteria> = the GranuleUR of a specific granule to be updated.

For ESDT, <granule search criteria> = the temporal range of the granules to be updated.

<measured parameter name> can be an actual parameter name for the granule or the string “ALL”, which indicates that the QA flag change will be applied for all the granule’s parameters.

Examples:

(<**TAB**> indicates a required tab character in these examples.)

Request to update science QA flags based on LGID for specific parameters:

```
begin QAMetadataUpdate Science LGID  
GLA06<TAB>28<TAB>GLA06_428_2119_002_0221_4_01_0001.DAT<TAB>Surface Elevation<TAB>  
Passed<TAB>Comment about something  
GLA06<TAB>28<TAB>GLA06_420_2119_002_0221_4_02_0001.DAT<TAB>Surface Elevation<TAB>  
Failed<TAB>Another comment  
.....repeat for each LGID
```

Request to update operational QA flags based on GranuleUR for a specific parameter:

```
begin QAMetadataUpdate Operational GranuleUR  
AE_Land<TAB>86<TAB>UR:10:DsShESDTUR:UR:15:DsShSciServerUR:10:[ :DSSDSRV]:17:SC:AE_Land.086:74735<TAB>Surf  
ace Soil Moisture<TAB>Passed<TAB>Comment  
.....repeat for each GranuleUR
```

Request to update science QA flags based on ESDT and temporal range for all parameters:

```
begin QAMetadataUpdate Science ESDT
GLA06<TAB>28<TAB>Mar 31 2007<TAB>Apr 1 2007<TAB>ALL<TAB>Being Investigated<TAB>Comment
AE_Land<TAB>86<TAB>Mar 31 2007<TAB>Apr 1 2007<TAB>ALL<TAB>Failed<TAB>Comment on failure
.....repeat for each ESDT
```

Request Footer

The footer is simply a statement that indicates the end of the request file and must take the following form:

```
end QAMetadataUpdate
```

4.9.12.3.1 Configuration File Format

Table 4.9.12-1 shows the configuration file parameters.

Table 4.9.12-1. Configuration File Parameters (1 of 2)

Parameter Name	Description
SYBASE_SQL_HOST	The host for the Inventory and DataPool databases
SYBASE_SQL_SERVER	The name of the Sybase server for the Inventory and DataPool databases
SYBASE_JDBC_DRIVER_CLASS	The Java class used for connecting the QAUU Java application to Sybase
SYB_DBNAME	The name of the Inventory database
SYB_DPL_DBNAME	The name of the DataPool database
SYB_PORT	The port number used to connect to the Inventory database
SYB_USER	The username used to connect to and perform queries for the Inventory and DataPool databases
PGM_ID	The ECS Program ID for the QAUU user (SYB_USER)
DB_NUM_RETRIES	The number of times to retry failed DB operations
DB_SLEEP_SEC	The number of seconds between DB operation retries
EMAIL_SERVER_HOST	Host name where email server runs
EMAIL_SMTP_USER	Email SMTP user name
EMAIL_QAUU_FROM_ADDRESS	Email notification sender address
FILE_NUM_RETRIES	The number of times to retry failed file operations
FILE_SLEEP_SEC	The number of seconds between file operation retries
QA_REQUEST_DIR	Path of directory containing QA update requests
QA_ERROR_REQUEST_DIR	Path of directory containing QA update requests that have failed.
QA_COMPLETED_REQUEST_DIR	Path of directory containing successfully completed QA update requests
QA_TEMP_DIR	Path of directory containing temporary files

Table 4.9.12-1. Configuration File Parameters (2 of 2)

Parameter Name	Description
QA_HISTORY_DIR	Path of directory containing QA update history files
DAAC_EMAIL_ADDRESSES	List of valid DAAC email notification addresses
<SCFSite>_EMAIL_FROM_ADDRESSES	List of valid email notification from addresses for a <SCFSite>
<SCFSite>_EMAIL_REPLY_ADDRESS	Valid email notification reply address for a <SCFSite>
<SCFSite>_NOTIFICATION_ON_SUCCESS	Flag indicating (if = "Y") that email notification should be sent upon successfully processing QA update requests for a <SCFSite> or for requests that fail. If = 'N', email should only be sent for requests that fail.
VALID_SCIENCE_QA_FLAGS	List of valid science QA flag values
VALID_OPERATIONAL_QA_FLAGS	List of valid operational QA flag values
NUM_XML_THREADS	The number of threads to be used. One thread will operate upon an UPDATE_BATCH_SIZE of QA updates.
MAX_NUM_GRANULES	The maximum number of granules that can be updated per run
UPDATE_BATCH_SIZE	The number of QA updates to be performed at a time.
XML_ARCHIVE_DIRECTORY	Pathname of XML Archive file system
SOCKS_PROXY_HOST	SOCKS proxy hostname
SOCKS_PROXY_PORT	SOCKS proxy port
BCP_EXEC_PATH	Path to Unix bcp executable
SHELL_PATH	Path to Unix sh shell needed to perform Unix commands
application.name	Name of this application
log.operations.level	Level of logging desired in operational log: NONE, INFORMATION, VERBOSE or XVERBOSE
log.debug.level	Level of logging desired in debug log: NONE, INFORMATION, VERBOSE or XVERBOSE
log.performance.level	Level of logging desired in performance log: NONE, INFORMATION, VERBOSE or XVERBOSE
log.overwrite	If true, log file will be overwritten for each run
log.threshold	Size of log files before new ones are created.
log.rotation.number	Number of log files that will be rotated through.

4.9.12.3.2 Recovery

The QA Update Utility offers the following mechanism for recovering from failures in a previous run. A working table is used to store the rows of QA updates to be worked off for a run. Rows are deleted when their updates are successfully applied. At the end of an uninterrupted run, all successfully updated rows will have been deleted from the working table. Rows for any updates which failed or did not complete will remain in the working table, and the reason for failure or incompleteness will be stored in a column of the table. After an error-free run, this working table will be empty. During startup, if the working table contains one or more rows,

recovery is attempted. The following failure/incompletion states are recorded in the working table:

FAILED XML UPDATE: could not update the XML file in the XML Archive

FAILED COPY: could not copy the XML Archive file to the Datapool

FAILED DPL UPDATE: could not update the Datapool MeasuredParameter table

COPIED XML: successfully copied updated XML file to the Datapool

UPDATED XML: successfully updated the XML file in the XML Archive

These states allow the utility to avoid duplicate work during recovery because they indicate what processing remains to be done. FAILED XML UPDATE indicates that all processing remains to be done. UPDATED XML and FAILED COPY indicate that the XML updates have already been applied, but the copy failed and needs to be retried. COPIED XML indicates that the update of the DataPool MeasuredParameter table is all that remains to be done. If a failure is attempted and fails a second time, it gets flagged as “investigating” in the working table. These are handled via various command-line options (see 4.9.12.1 Using the QA Update Utility above).

4.9.12.3.3 QA Update Email Script

There are two ways to place QA update request files in the QAUU request directory (the configured QA_REQUEST_DIR directory): manually and via email. Remote sites can submit update request files via email as attachments. With proper configuration, the email server will detect the emails as QAUU requests and invoke a Perl script - EcDsQAUUEmailScript.pl. The script parses the request, grabs the attached request file and moves it to the QAUU request directory. It resides on the central mail servers while the QAUU resides on other boxes. The directories containing the email script output (/usr/ecs/<mode>/CUSTOM/data/DSS/QAUU and its subdirectories) will be created on the hosts running the QAUU and remote mounted on the central mail servers. Special QAUU email aliases need to be set up in the /etc/aliases file on the email server host to direct email QAUU update request to the email script. One email alias is required for each mode supporting QAUU:

```
QAUU_<MODE>: "| /usr/ecs/<MODE>/CUSTOM/utilities/EcDsQAUUEmailScript.pl"
```

4.9.12.3.4 Outputs

Output of update events and errors will be always appended to a single log file. If specified as an option, a confirmation prompt will be displayed to the screen.

4.9.12.3.5 Event and Error Messages

If prompting is specified on the command line (default), confirmation messages are displayed to the operator. Otherwise, all prompts and screen displays are suppressed. All the error messages are written in the log file except for command line syntax errors which are displayed on the screen.

4.9.12.3.6 Reports

None.

4.9.13 Data Pool Move Collections Utility

The Move Collections Utility provides the EMD Operations Staff with a command-line interface to move collections from one file system to another. The utility requires command-line parameters that specify the collection (shortname and version id) to be moved and the target file system path. The utility also supports a verbose option. This option allows for enabling or suppressing detailed information displayed to the screen and log for the operator. The default is non-verbose, which allows an operator to run the utility as a background process. Fault recovery is also supported, allowing completion of a partially moved collection due to a database server fault or an operating system error.

The utility relies on the fact that symbolic links will be set from the collection's old filesystem to its new filesystem. For example, before a move, a collection might be located here: /datapool/OPS/user/FS1/MOAT/AIRABRAD.007. After invoking the utility with a target filesystem of FS2, it will be moved to /datapool/OPS/user/FS2/MOAT/AIRABRAD.007 with a symbolic link from its old location, i.e. /datapool/OPS/user/FS1/MOAT/AIRABRAD.007 → /datapool/OPS/user/FS2/MOAT/AIRABRAD.007

These links will be persistent and allow for backward compatibility via a "shadow directory". The shadow directory will be set up during the transition to StorNext and multiple file systems, as explained in ticket OP_S4_06.

The file system move is implemented as a copy operation to the new collection directory location, followed by removal of the old collection directory and its contents. The utility then establishes a link to the new location in place of the old directory. As a result, existing URL will not be invalidated by the move and no updated URLs need to be exported to ECHO. However, any existing URLs and file pointers will be invalid from the time when the utility starts deleting the existing directories until the time the link is established. During this time:

- A Data Pool ftp user clicking on a URL might experience a temporary error when trying to access files and directories associated with the moving collection. File transfers that are already in progress when deletion begins should complete normally.
- FTP Pull users could experience similar temporary problems when they try to access links in FTP Pull directories that were established by the OMS and that point to granules in the moving collection.

In addition, the following errors may occur during a collection move:

- During the time a collection is being moved, the Data Pool Web GUI will return an error if a user tries to access the collection via a bookmark. It will flag the collection and not display it as an active link on the collection drill down web page, the temporarily preventing drill down access to the collection.
- The Data Pool insert service will look up the collection path in the Data Pool database during the insert process. The collection path is updated once the copy step is complete. Any Data Pool insert processes that looked up the copy path BEFORE it was updated will insert their granules into the old directory location. If these granules are not copied

but then removed, they would become phantoms and could trigger additional errors downstream, e.g., in distribution; alternatively, if they are not removed they would cause the move process to fail, as the Data Pool insert service would re-create the deleted directories.

- The OMS looks up granule file locations immediately before performing an FTP Push operation. If the lookup occurs just before the collection information in the Data Pool database is updated, but the copy operation starts after the file was deleted, the FTP Push operation will fail and cause an operator intervention. Since the time window between file location lookup and ftp push start is small, the chances for that occurring are very small. The operator would need to resubmit the request, and since the directory entry will now have been updated, the ftp push operation will succeed.

If the above impacts are not acceptable, operators can suspend inserts and web access for the original file system by marking it as "unavailable" in the DPM GUI. This would also halt staging operations for that file system in OMS. To prevent errors, operators would need to verify before activating an order that it does not reference granules from the collection that is being moved. Operations will need to use a different mechanism to alert FTP users of the unavailability; and to prevent access, operations would need to take other measures, such as changing the directory permissions.

4.9.13.1 Using the Move Collections Utility

The Move Collections Utility is started by entering the following command:

```
> EcDIMoveCollection.pl <mode> -shortname <shortname>  
                        -versionid <versionid>  
                        -targetfs <file system path>
```

There are various command line parameters that are used in combination with each other. Table 4.9.13-1 provides a description of these parameters.

Table 4.9.13-1. Command Line Parameters of the Move Collections Utility

Parameter Name	Description
<mode>	An input parameter that 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 like OPS or TS1. This parameter is mandatory. Note: The user will be prompted if the utility is run in OPS mode to prevent any accidental loss of data.
-verbose	Directs the utility to run using verbose option. Some information will be displayed to the screen and detailed information will be written to the utility's log. Default is nonverbose. (See Note)
-shortname <shortname>	An input parameter that specifies the shortname of the collection to be moved. This parameter is mandatory.
-versionid <versionid>	An input parameter that specifies the version identifier of the collection to be moved. Do not specify leading zeros. This parameter is mandatory.
-targetfs <file system path>	An input parameter that specifies the relative target file system path to which the collection is being moved. Note that all Data Pool file systems must be mounted under the Data Pool root (e.g. (/datapool/OPS/user)). This parameter is mandatory.

There is no required ordered sequence of the parameters except for the <MODE> - this must be first parameter or a fatal error will be returned. The combination of these remaining inputs must be valid. A command line input error results in a 'usage' syntax display, and in most cases will also explain why the input was incorrect.

4.9.13.2 Move Collections Utility Commands

Below are some examples for invoking this utility:

```
1. EcDlMoveCollection.pl <mode> -shortname MODVOLC
   -versionid 1 -targetfs fs1 -verbose
```

Moves the files, browse links, and inventory information for the collection **MODVOLC.001** from its current directory as specified in the database to the new filesystem **fs1**. The collection will be moved from /datapool/<mode>/user/MOAT to /datapool/<mode>/user/fs1/MOAT. The utility will be run using verbose option, which displays information to screen and to the log.

```
2. EcDlMoveCollection.pl <mode> -shortname MODVOLC
   -versionid 1 -targetfs fs1
```

Same as 1) but in non-verbose mode. No output to the screen and less detail in the log.

4.9.13.3 Required Operating Environment

The Move Collections Utility will run on a Linux platform.

4.9.13.4 Interfaces and Data Types

Table 4.9.13-2 lists the supporting products that this tool depends upon in order to function properly.

Table 4.9.13-2. Interface Protocols

Product Dependency	Protocols Used	Comments
Data Pool database	SQL	Via SQL server machines
Perl DBI	DBD::Sybase	Requires proper install of baselined version of Perl.

4.9.13.5 Input File Format

N/A

4.9.13.6 Configuration File Format – EcDIMoveCollection.CFG

The “config” file contains vital details about how to connect to the Sybase database. Without this file, the utility can not run. The config file must be a single-entry plain text ASCII file, which has the following format:

```

SYB_USER = <string>
SYB_SQL_SERVER = <string>
SYB_DBNAME = <string>
PGM_ID = <string>
NUM_DB_RETRIES=<integer>
DB_SLEEP_SEC=<integer>
NUM_DELETE_RETRIES=<integer>
DELETE_SLEEP_SEC=<integer>

```

See Table 4.9.13-3 for a breakdown of individual parameters.

Table 4.9.13-3. Configuration File Parameters (1 of 2)

Parameter Name	Description
SYB_USER	The user name for the Sybase connection.
SYB_SQL_SERVER	The name of the SQL sever for this Sybase connection.
SYB_DBNAME	The name of the database you intend to connect to
PGM_ID	Program ID used for connecting to the Data Pool database. The value of this parameter must be set to 10000022 for this program.
NUM_DB_RETRIES	The number of times the utility will attempt to connect to the database before exiting. The recommended default is 5.
DB_SLEEP_SEC	The number of seconds the utility will wait ('sleep') between connection attempts. The recommended default is 10.
NUM_DELETE_RETRIES	The number of times the utility will rescan the old collection directory prior to deleting it. If the delete fails, it is most likely because the directory is not empty because some granules were inserted after the move started. The repeated rescanning for these files handles this case. The recommended default is 5.

Table 4.9.13-3. Configuration File Parameters (2 of 2)

Parameter Name	Description
DELETE_SLEEP_SEC	The number of seconds the utility will wait ('sleep') between old collection directory rescans/deletes. The recommended default is 10.

4.9.13.7 Special Constraints

The Move Collections Utility runs only if the Data Pool database server is running and if the database is available. It also assumes the stored procedures are present. The Move Collections Utility and the QA Update Utility are not allowed to run simultaneously. Each utility queries the database to see if the other is running and will shut down if the other is running.

4.9.13.8 Outputs

Output of update events and errors will be always appended to a single log file.

4.9.13.9 Event and Error Messages

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

4.9.13.10 Reports

None

4.9.13.11 Logs

The utility produces a log file called EcDIMoveCollection.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.

4.9.13.12 Recovery

The Move Collections Utility provides a capability to recover from an execution failure caused by situations such as the system faults or database errors leaving all or some of the file moves unprocessed. The utility will detect that an execution failure occurred at startup, whereupon the operator will be prompted as to whether recovery will be attempted or not. If the operator chooses to recover, the utility will complete the processing of file moves that were left unprocessed in the previous run. Upon completion of the recovery, the utility will run again with the current command-line parameters. An operator may not wish to recover (e.g. if the target filesystem has become corrupted or full). In this case, recovery will not be attempted, and the utility will run with the current command-line parameters. The moveFlag will automatically get reset to "N". Any files that were copied to the target file system would have to be manually deleted by the operator.

4.9.13.13 Sybase Error Handling

If a Sybase error occurs, you will most likely see the actual Sybase error string displayed on the screen and in the log. Some errors can be that the database server is unavailable, that the connection to the database was dropped, or that there was an error executing the 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 can not 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 5 and SLEEP_SEC was set to 10, this means it will try to connect 5 times, and will wait 10 seconds before each attempt – a total of 50 seconds if all attempts are unsuccessful.

4.9.14 Data Pool Hidden Scrambler Utility

The Data Pool Hidden Scrambler utility provides a mechanism by which the ECS Operations Staff can encrypt or re-encrypt the names of Data Pool hidden directories, both on the file system and in the Data Pool database.

The Data Pool Hidden Scrambler utility may be run with either the “transition” option (one time only, when hidden directory names are first created in the database for all Data Pool collections), or the “rename” option (when hidden directory names need to be re-encrypted, either to respond to a security breach, or on a scheduled basis at the DAAC, depending on DAAC security policy). This utility should be run as `cmshared`, `cm<mode>`, or similar.

4.9.14.1 Using the Data Pool Hidden Scrambler Utility

The Data Pool Hidden Scrambler utility should be started by the user `cmshared` (or similar). The Data Pool Hidden Scrambler utility is started by entering the following command:

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

There are four command line parameters that may be used. Table 4.9.14-1 provides a description of those parameters.

Table 4.9.14-1. Command Line Parameter

Parameter Name	Required	Description
transition	No	This parameter may not be used with any of the other command line parameters. Specifies that the utility should be run with the transition option.
collgroup	No	This parameter may not be used with the “transition” parameter, nor with the “shortname”/“versionid” parameters. Specifies that the utility should be run with the rename option, for all collections in the indicated collection group.
shortname	No	This parameter may not be used with the “transition” parameter, nor with the “collgroup” parameter. It must be used with the “versionid” parameter. Specifies that the utility should be run with the rename option for the indicated collection only.
versionid	No	This parameter may not be used with the “transition” parameter, nor with the “collgroup” parameter. It must be used with the “shortname” parameter. Specifies that the utility should be run with the rename option for the indicated collection only.

The Hidden Scrambler utility performs the following as part of the “rename” processing:

- Generates a new random `orderOnlySNDirName` and `orderOnlyGrpDirName` for each requested Collection and Collection Group in the Data Pool, and saves these names to the Data Pool database.

Note: If the `collgroup` parameter is used then the utility generates a new random `orderOnlyGrpDirName` for the collection group supplied and generates a new random

orderOnlySNDirName for each collection in that collection group. If the shortname and versionid parameters are used, then the utility will only generate a new random orderOnlySNDirName for the specified collection, and save that to the database. If neither the collgroup nor shortname/versionid parameters are used, the utility generates a new random orderOnlyGrpDirName for all collection groups in the mode and generates a new random orderOnlySNDirName for all collections in the mode.

- Creates new hidden directories based on the new orderOnlySNDirName and orderOnlyGrpName for each requested collection.
- Copies all files from the old hidden directories to the newly created hidden directories.
- Updates the FTPpull links for existing orders referencing the old hidden directories, to point to the new hidden directories
- Removes the old hidden directories.
- Reports the time it takes to update the FTPpull links.

The Data Pool Hidden Scrambler utility performs the following as part of the "transition" processing:

- Generates a new random orderOnlySNDirName and orderOnlyGrpDirName for each Collection and Collection Group in the Data Pool, and saves these names to the Data Pool database

4.9.14.1.1 Hidden Scrambler Utility Command Line Examples

1. For a "rename" run:

Note: For a "rename" run, the Hidden Scrambler utility should only be run during Data Pool downtime. The script must be run with a user account with privileges to rename directories on the Data Pool.

```
EcDlHiddenScramblerDataPool.pl OPS
```

The Hidden Scrambler Utility will perform rename processing for all collection groups and all collections in the Data Pool in OPS mode.

```
EcDlHiddenScramblerDataPool.pl OPS -collgroup MOAT
```

The Hidden Scrambler Utility will perform rename processing for the MOAT collection group and for all collections in the MOAT collection group, in OPS mode.

```
EcDlHiddenScramblerDataPool.pl OPS -shortname AST_L1B -versionid 3
```

The Hidden Scrambler Utility will perform rename processing only for the AST_L1B.003 collection in OPS mode. (Note that the corresponding collection group (ASTT) hidden directory name will not be re-encrypted).

2. For a "transition" run:

Note: Transition may be used while Data Pool is up. It should be used only once, the first time the utility is run in any given mode.

EcDIHiddenScramblerDataPool.pl TS1 -transition

The Hidden Scrambler Utility will generate encrypted directory names for all Data Pool collections and collection groups in TS1 mode, and save the names in the Data Pool database.

4.9.14.2 Hidden Scrambler Configuration File

The Data Pool Hidden Scrambler utility uses a configuration file, EcDIHiddenScrambler.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 4.9.14-2 describes the configuration parameters.

Table 4.9.14-2. Configuration Parameters

Parameter Name	Value Description
SYB_USER	Sybase login name for the user of the Data Pool database.
SYB_SQL_SERVER	Name of Sybase SQL Server hosting Data Pool database.
SYB_DBNAME	Name of Data Pool database.
PGM_ID	Program identifier used as seed to generate database password.
NUM_RETRIES	Number of times database operation will be attempted.
SLEEP_SEC	Number of seconds between retries.
PULL_DIR	Location of the FTP Pull Directory in appropriate mode. NOTE: Be sure to use the full path to the FTP Pull Directory, not a linked path (e.g. /datapool/<mode>/user/<fs>/PullDir).

4.9.14.3 Data Pool Hidden Scrambler Utility Main Screen

The Data Pool Hidden Scrambler Utility does not have a main screen. It has a command line interface only.

4.9.14.4 Required Operating Environment

The Hidden Scrambler Utility will run on a Linux platform.

4.9.14.5 Databases

Table 4.9.14-3 lists the supporting products that this tool depends upon in order to function properly.

Table 4.9.14-3. Product Dependencies

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

4.9.14.6 Special Constraints

The Data Pool Hidden Scrambler utility runs only if the Data Pool database server is running and if the database is available. It also assumes the stored procedures are present.

With the `rename` option, the utility must be run during Data Pool downtime.

The utility should only be run once with the `transition` option, the first time the utility is run in any given mode.

4.9.14.7 Outputs

Output of update events and errors will be always appended to a single log file.

4.9.14.8 Event and Error Messages

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

4.9.14.9 Reports

None.

4.9.14.10 Logs

The utility produces a log file called `EcDIHiddenScrambler.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.

4.9.14.11 Recovery

The Data Pool Hidden Scrambler Utility provides a capability to recover from interruptions caused by situations such as the system faults or database errors leaving all or some of the directories unprocessed. 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 is given no choice as to recovery. Recovery will proceed so that the Data Pool inventory and disk files will not be left in a corrupted state.

4.9.14.12 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 `SLEEP_SEC` 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.

4.9.15 Data Pool Remove Collection Utility

The Data Pool Remove Collection utility provides a mechanism by which ECS Operations staff can remove collections from the Data Pool database that are no longer of interest to the end users.

4.9.15.1 Using the Data Pool Remove Collection Utility

The Data Pool Remove Collection utility is started using the following parameters:

```
EcDIRemoveCollection.pl <MODE> -ShortName <SHORTNAME> -VersionId <VERSIONID> [-debug]
```

OR

```
EcDIRemoveCollection.pl <MODE> -inpfile <INPUTFILENAME> [-debug]
```

Table 4.9.15-1 lists the descriptions of the command line parameters.

Table 4.9.15-1. Command Line Parameters

Parameter Name	Required	Description
debug	No	Helps developers debug the app by printing copious debug information
ShortName	Yes	ShortName of the collection to be deleted
VersionId	Yes	VersionId of the collection being deleted
inpfile	Yes	The full path to an input file specifying multiple collections. Please note that either an input file or a ShortName/VersionId combination should be used but not both

The input file contains a list of ShortName VersionId pairs, one pair per line, as shown below:

```
ShortName1 VersionId1
```

```
ShortName2 VersionId2
```

...

There should be at least one space or tab between the ShortName and VersionId on each line in the input file. Other white space will not affect the utility.

If there are any active granules associated with the collection, or if there are any other database errors, the utility will print an appropriate error message on the screen and log the message too.

The Remove Collection utility removes collections only from the Data Pool database. The ECS Operations staff is responsible for removing any directories (public and hidden) associated with the collection from the Data Pool file system.

4.9.15.2 Remove Collection Configuration File

The Data Pool Remove Collection utility uses a configuration file, EcDIRemoveCollection.CFG, located in the /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 4.9.15-2 describes the configuration parameters.

Table 4.9.15-2. Configuration Parameters

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

4.9.15.3 Data Pool Remove Collection Utility Main Screen

The Data Pool Remove Collection Utility does not have a main screen. It has a command line interface only.

4.9.15.4 Required Operating Environment

The Data Pool Remove Collection Utility will run on a Linux platform. It assumes that perl with the Sybase DBI modules is already installed.

4.9.15.5 Databases

Table 4.9.15-3 lists the supporting products that this tool depends upon in order to function properly.

Table 4.9.15-3. Product Dependencies

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

4.9.15.6 Special Constraints

The Data Pool Remove Collection utility runs only if the Data Pool database server is running and if the database is available. It also assumes the stored procedures are present.

4.9.15.7 Outputs

Output of collection removal events and errors will be always appended to a single log file. See Section 4.9.16.10.

4.9.15.8 Event and Error Messages

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

4.9.15.9 Reports

None.

4.9.15.10 Logs

The utility produces a log file called EcDIRemoveCollection.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.

4.9.15.11 Recovery

Since the removal of a collection is handled in one database transaction, it either works or it does not. Hence there is no need for recovery.

4.9.15.12 Sybase Error Handling

If a Sybase error occurs, the operator will most likely see the actual Sybase error string displayed on the screen and in the log. Some errors can be that the database server is unavailable, that the connection to the database was dropped, or that there was an error executing the 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 can not 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 5 and SLEEP_SEC was set to 10, this means it will try to connect 5 times, and will wait 10 seconds before each attempt – a total of 50 seconds if all attempts are unsuccessful.

This page intentionally left blank.

4.9.16 Data Pool Band Backfill Utility

The DPL Backfill Utility is a command line tool that can correct band extraction problems that occurred during DPL registrations. Granule registrations cannot fail if band extraction problems are encountered but the subsequent publications on convert-enabled data types must fail if the band information is not present in the DPL database at publication time.

The Band Backfill utility was developed to correct the problems above. It will:

- backfill the band information in the DPL database for the registered granules specified in its input file.
- request the publication of the backfilled granules via the new Data Pool Action driver.

The DAAC Operations staff can identify the granules that need band backfill via the Data Pool Maintenance GUI or by inspecting the EcDINewInsertUtilityDPAD.log file. In both cases, the type of error encountered is:

```
ERROR publreg operation encountered a convertEnabled granule with no  
band information, granuleState
```

For each Data Pool granuleId in its input file, the utility will perform the following steps:

1. Validate that the granule is in the hidden Data Pool. The granules can belong to DPL Ingest (isOrderOnly = H) or to OMS (isOrderOnly = Y).
2. Validate that the granule belongs to a convert-enabled ESDT.
3. Validate that the DPL database contains no band information for this granule.
4. Extract the band information from the granule data files and produce a .BandHeader file. This step is performed by invoking an external script (./custom/utilities/EcDIAdHEGStart). The same script is also used by the new Data Pool Action Driver to create the .BandHeader file during granule registrations. Note: for a multi-file granule, the first file that contains band information will be used.
5. Parse the .BandHeader file and insert the necessary information in the Data Pool database. The .BandHeader file will be removed once it has been parsed.
6. Request the publication of the backfilled granule by inserting a record in the DIInsertActionQueue table in the Data Pool database.
7. Process the next granule in the input file. Note: if an error is encountered during the processing of a granule, the error is logged and the utility continues with processing of the subsequent granules.

4.9.16.1 Using the Data Pool Band Backfill Utility

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

```
EcBandBackfillUtilityStart -mode <mode> -file <input file>
```

There are two command line parameters that are used in combination with each other. Table 4.9.16-1 provides a description of these parameters.

Table 4.9.16-1. Data Pool Band Backfill Utility Command Line Parameters

Parameter Name	Description
-mode <mode>	Specifies the mode of operation (OPS, TS1, etc.)
-file <input file>	Specifies the full path and file name of the file containing the Data Pool granule IDs of the granules that need to be populated with band information. The file is a flat ASCII file and it contains one Data Pool granuleId per line.

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

4.9.16.2 Data Pool Band Backfill Utility usage examples

Below is an invocation example:

```
1. EcBandBackfillUtilityStart           -mode           OPS           -file
   /home/cmshared/granuleIds.txt
```

Backfills the band information and requests the DPL publication for the granuleIds contained in the specified file. The file contains one Data Pool granuleId per line.

4.9.16.3 Required Operating Environment

The Data Pool Band Backfill 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.

4.9.16.4 Interfaces and Data Types

Table 4.9.16-2 lists the supporting products that this tool depends upon in order to function properly.

Table 4.9.16-2. 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.

4.9.16.5 Input File Format

One granuleId per line.

4.9.16.6 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.

4.9.16.7 Special Constraints

The mode specific database needs to be up and running and the installation platform need to have access to the Data Pool file system.

4.9.16.8 Outputs

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

4.9.16.9 Event and Error Messages

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

4.9.16.10 Reports

None.

4.9.16.11 Logs

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

4.9.16.12 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.

4.9.16.13 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.

4.9.17 XML Replacement Utility

The XML Replacement Utility (XRU) is an operational support tool used for replacing granule xml metadata files in the ECS inventory. The main role of this utility is to allow the DAAC operator to validate the new metadata file and to inform ECHO of the metadata change. The utility will only replace the granule xml metadata file in the ECS archive. The operator is responsible for making any needed changes to the Inventory database, the DataPool database, and the granule xml metadata file within the DataPool file systems.

4.9.17.1 Using the XML Replacement Utility

The XRU is started by executing the following:

```
$ EcDsAmXruStart <mode> [ -xmlfile <filename> | -xmlmdir <directory> ]
```

The mode parameter is required and either `-xmlfile` or `-xmlmdir` needs to be supplied.

`<mode>`: the mode to run the utility in

`-xmlfile <filename>`: if specified, the utility will perform xml replacement on a single xml file. `<filename>` is the full directory path to the file which has been edited.

`-xmlmdir <directory>`: if specified, the utility will go to the supplied directory path and perform xml replacement on all xml files within the directory

The user running the utility must have read and write privileges in the source and destination (the smallfile archive) directory. The directory must be visible on the host where the utility is run.

4.9.17.2 Required Operating Environment

Linux

4.9.17.3 XML Filename Format

Filename

The input granule xml metadata file must follow the following naming convention:

`<shortname>.<versionid>.<ecsid>.xml`

e.g. MOD29P1D.004.48903.xml, where `<ecsid>` is the dbID in the DsMdGranules table in the Inventory database.

This naming convention is the same naming convention used by the granule xml metadata file in the ECS inventory. The utility assumes that the operator will copy the granule xml metadata file out of ECS and modify the contents for replacement. For any xml files that do not follow the naming convention the replacement will fail and an error will be logged. Any files not ending with an extension of “.xml” will be ignored.

4.9.17.4 Configuration File Format

Table 4.9.17-1 shows the configuration file parameters.

Table 4.9.17-1. Configuration File Parameters

Parameter Name	Description
PROGRAM_ID	The ECS Program ID for the XRU user (SYB_USER)
DB_INVENTORY_HOST	The host for the Inventory database
DB_INVENTORY_PORT	The port number used to connect to the Inventory database
DB_SPATIAL_PORT	The port number used to connect to the sqs server
DB_NAME	The name of the Inventory database
DB_INVENTORY_USERNAME	The username used to connect to and perform queries for the Inventory database
DB_RETRIES	The number of times to retry failed DB operations
DB_SLEEP_SECONDS	The number of seconds between DB operation retries
DEBUG_MESSAGES	The flag that signifies if debug log messages will be logged or not. Valid values are "Y" and "N".
DB_JDBC_CLASS	The JDBC class used for connecting the XRU application to Sybase
DB_TIMEOUT_SECONDS	The number of seconds that a database operation will execute before timing out

4.9.17.5 Outputs

Output will be displayed on the terminal (standard out) where the utility is executed. The output will display one line for each file being processed showing the name of the file and the result for that file; (either "done" or "failed", with "done" meaning successful replacement and "failed" meaning unsuccessful replacement). An example of the output is displayed below:

```
f4dpl01{cmshared}114: EcDsAmXruStart DEV04 -xmlmdir /home/cmshared/source
Initializing...
Processing...
AE_Land.086.55515.xml...failed!
AE_Land.086.55516.xml...failed!
AST_L1A.003.48619.xml...done!
MOD29P1D.004.48909.xml...done!

Success: 2, Fail: 2, Total: 4
```

A log file will be generated to log the execution of the utility and any errors that occurred during the xml replacement run. The log file will be located in the mode's log directory with the name, EcDsAmXru.log. Older logs will have a timestamp appended to the end of the log filename.

Glossary

AutoSys/AutoXpert	COTS software that provides job scheduling and management. Also provides graphics to monitor, analyze, forecast and plan AutoSys implementations.
Baseline Manager	Baseline Manager package used to maintain records of baselined operational system configurations. (see also XRP-II)
Batch Insert Utility	The Batch Insert Utility is a command line interface that allows operators to insert granules residing in or outside of (non-ECS granules) the ECS archive into the Data Pool.
Bulk Metadata Generation Tool	The EcOsBulkURL Utility allows operators to make available the File Transfer Protocol (FTP) Universal Resource Locators (URLs) in the Data Pool to the ECS Clearing House (ECHO).
ClearCase	Software change manager that stores ECS custom software and science software, regulates access to the files, controls and logs file changes, performs software builds, and maintains a record of the build. Maintains a library of software deployed to sites.
CMI	Cryptographic Management Interface. Used to create accounts for given user names and passwords.
Crack	Used to determine if passwords are secure.
DAR	Data Acquisition Request for ASTER instrument data.
Database Installation and Maintenance Scripts	A set of eleven standard database scripts have been created for the DDIST, INGEST, MSS, PDPS, SDSRV, STMGT, and SUBSRV subsystems to facilitate database installation and database administration activities. These scripts are designed to be accessible from both the command line and the Stage Install function of ECSAssist.
Data Dictionary Maintenance Tool	Tool that allows the operator to maintain the ECS Data Dictionary.
Data Distribution Requests GUI	Monitors and controls the request for data distribution (for FTP Pushes and FTP Pulls only).
Data Ingest	Provides a means for external providers to ask for ECS ingest services.

Data Pool Ingest GUI	The Data Pool Ingest Graphical User Interface allows the operators to view past ingest activities, monitor and control ingest requests, modify system and external data provider parameters, and initiate hard media ingest.
Data Pool Access Statistics Utility – Rollup Scripts	The Data Pool Access Statistics Utility (DPASU) provides the ECS Operations Staff with several capabilities related to collecting access statistics for the Data Pool database. The DPASU encompasses two types of scripts: rollup and maintenance. The rollup scripts read and parse access logs to compile statistics and store those records in the Data Pool database, while the maintenance scripts backup, restore, and delete data in the related Data Pool database tables.
Data Pool Access Statistics Utility – Maintenance Scripts	The Data Pool Access Statistics Utility (DPASU) provides the ECS Operations Staff with several capabilities related to collecting access statistics for the Data Pool database. The DPASU encompasses two types of scripts: rollup and maintenance. The maintenance scripts backup, restore, and delete data in the related Data Pool database tables.
Data Pool Cleanup Utility	The Data Pool Cleanup utility provides a mechanism for the ECS Operations Staff to remove expired granules and their associated metadata and browse files from the Data Pool disks and corresponding Data Pool database inventory.
Data Pool Maintenance GUI	The DPM GUI provides an operator interface to monitor the current status of Data Pool Inserts and to maintain specific Data Pool parameters. This GUI manages ECS and Non-ECS data collections.

Data Products	<p>Designated as standard or special data products, generated as a part of research investigation using EOS data. The various levels of data are defined as follows (1995 MTPE/EOS Reference Handbook):</p> <p><i>Level 0</i> - Reconstructed, unprocessed instrument/payload data at full resolution; any and all communications artifacts, e.g., synchronization frames, communications headers, duplicate data removed.</p> <p><i>Level 1A</i> - Reconstructed, unprocessed instrument data at full resolution, time-referenced, and annotated with ancillary information, including radiometric and geometric calibration coefficients and geo-referencing parameters, e.g., platform ephemeris, computed and appended but not applied to the Level 0 data.</p> <p><i>Level 1B</i> - Level 1A data that have been processed to sensor units (not all instruments will have a Level 1B equivalent).</p> <p><i>Level 2</i> - Derived geophysical variables at the same resolution and location as the Level 1 source data.</p> <p><i>Level 3</i> - Variables mapped on uniform space-time grid scales, usually with some completeness and consistency.</p> <p><i>Level 4</i> - Model output or results from analyses of lower level data (e.g., variables derived from multiple measurements).</p>
Data Server	<p>Software associated with storing earth science and related data, searching and retrieving the data, and supporting the administration of the data, hardware devices, and software products.</p>
DDTS	<p>Change request manager used to compose, submit, report and track status of proposals to change ECS resources electronically.</p>
ECS Assistant	<p>The ECS Assistant (ECSAssist) is a custom program that simplifies the process of installation, testing and management of ECS. The tool is for use in installing software and maintaining the information related to that software. Only the Subsystem Manager function of ECSAssist should be used in the ECS operational environment.</p>
ECS Desktop	<p>Simulates Common Desktop Environment (CDE); interface that acts like a file manager, allowing launch of applications, creation of directories and moving/copying/ deleting files.</p>

ECS Registry GUI	The ECS Registry GUI is a management tool for ECS applications allowing users to create and update parameter information. Registry data is stored in a registry database.
Email	Service that manages electronic mail messages for DAAC operators.
EOSView	A custom HDF file verification tool. Displays HDF files and HDF-EOS data.
Event Log	The Event Log Database resides at each ECS site. It records status and error messages generated by the various ECS applications at the site. The Event Log Browser is used to view the status and error messages.
Event Log Directory	This directory resides on every computer platform and contains the log files used by applications to report status and error messages. Log files in the Event Log Directory are loaded into the Event Log Database on a periodic basis.
FLEXIm	COTS for the administration of licenses.
GCMD Data Export	Extracts Data Interchange Format (DIF) from the SDSRV inventory database to the Global Change Master Directory (GCMD).
Granule Deletion Administration Tool	The Granule Deletion Administration Tool provides the ECS Operations Staff with the capability to delete granules using a command line interface. The granules can be deleted from both the inventory and archive or just the archive. Granules are not physically deleted from the archive. The directory entry is deleted so the files cannot be accessed. The physical storage occupied by the deleted granules is not reclaimed through this operation.
IDL	Interactive data language used to interactively visualize and analyze scientific and engineering data products.
Ingest GUIs	Allows monitor and control of Ingest requests, modification of system and external data-provided parameters and initiate hard media ingest. An HTML interface allows for submission of ingest requests for processing.
Inventory, Logistics and Maintenance (ILM) Manager	Supports M&O in maintaining records that describe all inventory components, structures, and interdependencies.
ISQL	SQL command parser utility used to interact with a SQL server and databases on a SQL server.

Java System Web Server	This COTS product is a multi-process, multi-threaded, secure web server built on open standards. It provides high performance, reliability, scalability, and manageability for any size enterprise, and it includes modules for creating and managing Web content, for extending or replacing functions of the server (e.g., through Java servlets and JavaServer pages), and for providing application-specific services such as security and access control. In ECS, the Web Server is used by several subsystems to access HTML files and to service web-based applications.
Main Window Manager	Provides login to UNIX and ECS, authenticates the user and brings up the appropriate ECS Desktop based upon the operator role.
Microsoft Office Professional	A collection of applications working together as if it were a single program. The collection includes Word (for text and graphical processing), Excel (a spreadsheet) and PowerPoint (making graphics/presentations) programs.
Netscape Communicator	World Wide Web (WWW) browser. Used to obtain information from other sources.
Networker	Tool used by system administrators to perform site-wide system backups, except databases.
Order Manager GUI	The Order Manager (OM) Graphical User Interface (GUI) provides the operators with direct access to the OM database. The GUI allows operators to view and modify requests that have been placed on hold by the Order Manager because they require operator intervention and resubmit requests or portions of a request that failed. For Synergy III, the GUI is an addition to the existing System Management Subsystem (MSS) Order Tracking GUI and the Data Distribution (DDIST) GUI rather than a replacement for them.
Order Manager Command Line Utility	The Order Manager Command Line utility provides a mechanism by which the ECS Operations Staff can submit order requests into the Order Manager System (OMS) database directly without knowing whether the Order Manager Server is up or down.
Order Tracking	User services tool that tracks order status and request status.

PIPRGenerator User Interface	The PIPRGenerator is the command line interface for the Production Request Editor. The PIPRGenerator allows the user to create and activate a number of Routine Production Requests using information contained in an input file. The input file contains the PgeIds and GEOIds for the PGEs and primary input granules, respectively, for the Production Requests to be created.
Process Control File	Specifies the names and locations of files used by science software executables, and defines the correspondence between the file specifications and the logical identifiers used by the science software to reference the specified files.
Quality Assurance Monitor	The ECS Quality Assurance (QA) Monitor processing capabilities enable DAAC operators to perform duties associated with DAAC QA activities. The ECS QA Monitor GUI is the user-interface for entering data requests and displaying data, status, and error messages. The QA Monitor does not produce data products, but communicates with the science data server to retrieve data that have been previously archived. The ECS QA Monitor assists in manual quality assurance activities such as querying and retrieving data granules, visualizing data products and updating metadata.
Regenerate Failed PDR Tool	The Regenerate Failed PDR tool provides the ECS Operations Staff with the capability to generate a Product Delivery Record (PDR) for each failed granule in a PDR and copy the generated PDRs to an Ingest polling directory using a command line interface. The purpose of the tool is to provide a means for the ECS Operations Staff to easily resubmit only failed granules to Ingest polling, rather than having to manually edit the original PDR file or resubmit all of the granules, which would create duplicate granules in the archive.
Replication Server	Maintains warm standby copies of application data and replicates changes among databases at different sites.
Resource Planning	Used to plan the allocation of DAAC resources.

Restricting ESDT and Granule Access	The two scripts <i>EcDsSrUpdateESDTAccess</i> and <i>EcDsSrUpdateQATimeRange</i> provide DAAC operations staff the capability to adjust how the Science Data Server restricts <i>Acquire</i> access to granules. When evaluating a user's permission to <i>Acquire</i> a granule, the Science Data Server uses the value of the NASA user attribute stored in the User Profile system. The first script, <i>EcDsSrUpdateESDTAccess</i> , allows the DAAC operator to restrict an entire ESDT/Data Collection to one or more of the specific NASA user types stored in the User Profile system. The second script, <i>EcDsSrUpdateQATimeRange</i> , allows individual granules to be restricted based upon the granule's QA flag values and the type of NASA user making the request.
Science Data Server GUI	The Science Data Server GUI provides the operator two major functions, the management of Earth Science Data Types (ESDTs) and the management of all types of requests that the Science Data Server operator is involved with.
Science Data Server Command Line Interface	The Science Data Server Command Line Interface (SCLI) tool provides the Product Distribution System (PDS) with the capability to acquire Landsat and Non Landsat products via an FtpPush, in fulfillment of orders placed by ECS users for those products via DTF tape, 8mm tape, Digital Linear Tape and CD-ROM/DVD. The SCLI tool accomplishes this by providing a command line interface for the Science Data Server acquire request.
Sniffers	Monitors network traffic for collisions and troubleshooting.
SQL Server	A SQL Server is a set of cooperating processes that manage multiple Sybase databases and multiple users.
SSI&T Manager	Allows check in and verification of science software delivered by the instrument teams at the Science Computing Facilities. Provides access to all COTS tools and custom applications that are part of the SSI&T environment.
StorNext	StorNext Storage Manager (SNSM) is a hierarchical storage management (HSM) system for managing data on multiple storage tiers consisting of disk and tape resources.
Subscription Editor	Allows the operator to manually enter Subscriptions to the Subscription Server.
Subscription Server	Allows users to register their events related to a certain type of data.

Sybase Replication Server	Maintains warm standby copies of application data and replicates changes among databases at different sites.
TCP Wrappers	Monitors and controls access to network services on a host.
TestTrack Pro	TestTrack Pro (TTPro) provides a trouble ticketing service that furnishes both ECS users and operations personnel at the DAACs a common environment for classifying, tracking, and reporting the occurrence and resolution of system-related problems.
Tripwire	An intrusion detection tool that monitors files for changes.
Tuple	Data reflecting unique strings of information associated with and descriptive of an event (e.g., names, identifier numbers, data types).
Update Granule	The Update Granule Utility provides the ECS Operations Staff with a command line interface to update the expiration date and optionally the retention priority of granules in the Data Pool inventory. The granules in the Data Pool inventory can be ECS or non-ECS granules.
User Account Management GUI	Tool used by DAAC operators to process new accounts and manage existing ones.
V0 Gateway Metadata Import	Extracts ESDT data into an export file, maps the valids to the V0 domain and stores it in the V0 Gateway database.
WhatsUp Professional	WhatsUp Professional is a graphical network mapping, monitoring, and notification COTS application. The DAAC network administrators use it to monitor network devices and the services on those devices and to get feedback on their network's performance.
Whazzup GUI	The Whazzup GUI is a tool that monitors and displays the execution status and related performance statistics associated with ECS programs. It is implemented using the Perl language and uses a CGI-based web interface to display information to the user.

Abbreviations and Acronyms

A

ACS	Automated Cartridge System
ACSLs	Automated Console System for Library Services
ADC	Affiliated Data Center
AI&T	Algorithm Integration and Test
AITTL	Algorithm Integration and Test CSCI
ALOG	Application Log file
AM-1	See Terra
AMASS	Archival Management and Storage System
AML	Automated Media Library
AMU	Automated Management Unit
ANSI	American National Standards Institute
AOI	Area of Interest
AOS	Area of Search
API	Application Program (or programming) Interface
AR	Action Request
AS	ASTER
ASBP	AIRS Summary Browse Products
ASCII	American Standard Code for Information Exchange
ASTER	Advanced Spaceborne Thermal Emission and Reflection Radiometer

B

BB	Bulletin Boards
BIS	Baseline Information System
BLM	Baseline Manager
BOM	Bill of Material
BMGT	Bulk Metadata Generation Tool

C

CAP	Cartridge Access Port
CCB	Configuration Control Board
CCR	Configuration Change Request
CCS	Control Center System Middleware, a custom code replacement for DCE
CD	Cartridge Drives
CD-ROM	Compact Disk -- Read Only Memory
CDDTS	Clear Distributed Defect Tracking System
CDE	Common Desktop Environment
CDRL	Contract Data Requirements List
CE	CERES
CFG	Configuration File
CGI	Common Gateway Interface
CHUI	Character-based User Interface
CI	Configuration Item
CID	Control Item Identifier
CIDM	Client, Interoperability and Data Management Subsystem group
CLI	Command Line Interface
CLS	Client Subsystem
CM	Configuration Management
CMI	Cryptographic Management Interface
COTS	Commercial Off-The-Shelf
CPIO	Copy In and Out
CPU	Central Processing Unit
CRM	Change Request Manager
CSCI	Computer Software Configuration Item
CSDT	Computer Science Data Type
CSMS	Communications and Systems Management Segment

CSS Communications Subsystem (of CSMS)
CSV Comma Separated Variable

D

DAAC Distributed Active Archive Center
DAO Data Assimilation Office (at GSFC)
DAP Delivery Archive Package
Delivery Algorithm Package
DAR Data Acquisition Request
DAS Data Availability Schedule
Distributed Archive Server
DB DataBase
DBA Database Administrator
DBMS DataBase Management System
DBO Database Owner
DCCI Distributed Computing Software CSCI (of CSS)
DDICT Data Dictionary CSCI (of DMS)
DDMT Data Dictionary Maintenance Tool
DDTS Distributed Defect Tracking System (COTS)
DEG Degrees
DES Data Encryption Standard
DHWM Data High Water Mark
DID Data Item Description
DIF Data Interchange Format
DLL Dynamically Linked Library
Data Link Library
DLT Digital Linear Tape
DLWM Data Low Water Mark

DMS	Data Management Subsystem (of SDPS)
	Degrees, Minutes and Seconds
DNS	Domain Name Service
DO	Derived Objects
DPAD	Data Pool Action Driver
DPASU	Data Pool Access Statistics Utility
DPIU	Data Pool Insert Utility
DPL	Data Pool
DPM	Data Pool Maintenance
DPR	Data Processing Request
DSKT	Desktop CSCI (of CLS)
DSS	Data Server Subsystem (of SDPS)
DTS	Distributed Time Service

E

EA	External Ancillary
EASI	ECS Assist System Installation
EBIS	ECS Baseline Information System
ECHO	ECS Clearing House
ECN	Equipment Control Number
ECS	EOSDIS Core System
ED	EDOS
EDC	EROS Data Center (DAAC)
EDF	ECS Development Facility
EDGRS	ESDIS Data Gathering and Reporting System
EDHS	ECS Data Handling System
EIF	Entry Interface Facility
EIN	Equipment Inventory Number
EOC	EOS Operations Center (ECS)

EOS	Earth Observing System
EOS-AM	EOS Morning Crossing (Descending) Mission -- see Terra
EOSDIS	Earth Observing System Data and Information System
ESDIS	Earth Science Data and Information System
ESDT	Earth Science Data Type
ESN	EOSDIS Science Network
ESSM	Enterprise SQL Server Manager
ETAC	EMASS Technical Assistance Center

F

FDDI	Fiber Distributed Data Interface
FIFO	First In-First Out
FQDN	Fully Qualified Domain Name
FSMS	File Storage Management System
FTP	File Transfer Protocol

G

GB	Giga-Byte
GCMD	Global Change Master Directory (not developed by the ECS project)
GFE	Government Furnished Equipment
GMT	Greenwich Mean Time
GSFC	Goddard Space Flight Center (DAAC)
GTWAY	V0 Interoperability Gateway CSCI (of DMS)
GUI	Graphical User Interface
GV	Ground Validation

H

HAIF	HDF ASCII Interchange Format
HDF	Hierarchical Data Format

HDF-EOS	an EOS proposed standard for a specialized HDF data format
HEG	HDF-EOS to GeoTIF Converter
HLE	High Level Event
HMI	Human Machine Interface
HTML	HyperText Markup Language
HTTP	Hypertext Transport Protocol
HWCI	Hardware Configuration Item

I

I&T	Integration and Test
I/O	Input/Output
ICD	Interface Control Document
ICMP	Internet Control Message Protocol
ID	IDentification
IDG	Infrastructure Development Group
IDL	Interactive Data Language
ILM	Inventory, Logistics and Maintenance Manager
IMSL	International Math and Statistics Library
INCI	Internetworking CSCI (of CSMS)
INGST	Ingest Services CSCI (of INS)
INS	Ingest Subsystem (of SDPS)
IOS	Interoperability Subsystem
IP	Internet Protocol (address)
IR1	Interim Release One
IRD	Interface Requirements Document
ISO	International Standards Organization
ISQL	Interactive Structured Query Language
ISS	Internetworking Subsystem (of CSMS)
IT	Instrument Team

J

JDT	Java Data Acquisition Request (DAR) Tool
JIL	Job Information Language
JM	Job Management
JPL	Jet Propulsion Laboratory

L

L0-L4	Level 0 (zero) through Level 4 data
LAMS	Landsat 7 Archive Management System
LAN	Local Area Network
LaRC	Langley Research Center (DAAC)
LCU	Library Control Unit
LDAP	Lightweight Directory Access Protocol
LMU	Library Management Unit
LS	LandSat
LSM	Library Storage Module
LTM	Log Transfer Manager

M

MB	MegaByte (10^6 bytes)
MCF	Metadata Configuration File
MCI	Management Software CSCI
MDA	Management Data Access
MFR	Manufacturer
MI	MISER
	Machines Impacted
MIN	Minutes
MISR	Multiangle Imaging SpectroRadiometer
MLCI	Management Logistics CSCI

MM	Millimeter
MO	MODIS
MODIS	Moderate-Resolution Imaging Spectrometer
MOPITT	Measurements of Pollution in the Troposphere
MP	MOPITT
MSS	System Management Subsystem (of CSMS)
MTPE	Mission to Planet Earth
MUA	Mail User Agent
MWO	Maintenance Work Order

N

N/A	Not Applicable
NBSRV	Spatial Subscription Server
NCR	Non-Conformance Report
NCS	Network Computing System Netscape Commerce Server
NESDIS	National Environmental Satellite Data and Information Service
NFS	Network File System
NMC	National Meteorological Center (NOAA)
NOAA	National Oceanic and Atmospheric Administration
NSBRV	Spatial Subscription Server CSCI
NSIDC	National Snow and Ice Data Center (DAAC)
NW	NetWorker

O

ODFRM	On-Demand Product Request Form (of CLS)
ODL	Object Description Language
OEM	Original Equipment Manufacturer
OODCE	Object Oriented DCE

OM	Order Manager
OMS	Order Manager Server
OPER	Operator
OPS	Operations
OS	Operating System
OSF	Open Systems Foundation
OSI	Open System Interconnect
OTS	Off-the-Shelf

P

PAN	Production Acceptance Notification
PCF	Process Control File
PCFG	Parameter Configuration File
PDF	Portable Document Format
PDR	Production Data Request
	Product Delivery Record
PFC	Prohibited Function Checker
PGE	Product Generation Executable
PLANG	Production Planning CSCI (of PLS)
PM	Preventative Maintenance
PO	Purchase Order
POSIX	Portable Operating System Interface for Computer Environments
PR	Production Request
PRS	Primary Replication Server
PRONG	Data Processing CSCI (of DPS)
PVC	Performance Verification Center
PWB	Planning Work Bench (of PLS)
	Production Planning Workbench

Q

QA	Quality Assurance
QC	Quality Control
QRU	Query/Retrieve/Update

R

RAID	Redundant Array of Inexpensive Disks
RAM	Random Access Memory
RMA	Reliability, Maintainability, Availability
ROC	Read-Only Cache
RRS	Replicate Replication Server
RSA	Replication System Administration
RSI	Replication Server Interfaces
RSM	Replication Server Manager
RSSD	Replication Server System Database
RTF	Rich Text Format
RTU	Rights To Use

S

SA	System Administrator SAGE III
SAA	Satellite Active Archive
SAGE	Stratospheric Aerosol and Gas Experiment
SCF	Science Computing Facility
SCLI	Science Data Server Command Line Interface
SCSI	Small Computer System Interface
SDP	Science Data Processing
SDPF	Sensor Data Processing Facility (GSFC)
SDPS	Science Data Processing Segment (ECS)
SDPTK	SDP Toolkit CSCI

SDS Science Data Standards (Science Data Group data used in EOSView)
 SDSRV Science Data Server CSCI (of DSS)
 SEC Seconds
 SGI Silicon Graphics, Inc.
 SMC System Management Center (ECS – at GSFC)
 SMTP Simple Mail Transport Protocol
 SNMP Simple Network Management Protocol
 SP Space Pool
 SPRHW Science Processing HWCI
 SQL Structured Query Language
 SQR SQL Report Writer
 SSH Secure Shell
 SSI&T Science Software Integration and Test
 SSM/I Special Sensor for Microwave/Imaging (DMSP)
 SSO System Security Officer
 SSS Spatial Subscription Server
 SST Sea Surface Temperature
 SSAP Science Software Archive Package
 STK Storage Tek
 SYS System

T

TAR Tape Archive
 TCP/IP Transmission Control Protocol/Internet Protocol
 TDP Tabular Datastream Protocol
 Terra EOS AM Project spacecraft 1, morning spacecraft series -- ASTER, MISR, MODIS and MOPITT instruments (formerly called AM-1 spacecraft)
 TOMS Total Ozone Mapping Spectrometer
 TONS TDRS On-board Navigational System
 TT Trouble Ticket

TTPro TestTrack Pro

U

UFS UNIX File System

UR Universal Reference

URL Universal Resource Locator

USGS United States Geological Survey

UT Universal Time

UTC Universal Time Code

UUID Universal Unique Identifier

V

V0 Version 0

VATC Verification and Test Center

VOB Version Object Base

W

WAIS Wide Area Information Server

WAN Wide Area Network

WKBCH Workbench CSCI (of CLS)

WWW World Wide Web

X

xAR (generic) Acquisition Request

XML Extensible Markup Language

Y

Y2K Year 2000