4.7.7 Using the Order Manager GUI

The Order Manager (OM) GUI provides operators with access to the Order Manager database. The GUI allows operators to view and modify requests that have been placed on hold by the Order Manager Server because they require operator intervention, and resubmit requests or portions of requests that have failed. It also supports the processing of physical media requests; management of HEG orders; and user configuration of ODL metadata users, external subsetter and SCP policy.

Notes on Operator Capability Levels

In accordance with new Operator GUI security standards, the OM GUI will implement 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 restricted to basic functionality as outlined in this document. To accomplish this, the OM GUI disables inputs, buttons, and access to certain pages for Limited Capability Operators.

All screenshots in this document show pages accessible to Full Capability Operators, with the understanding that certain elements will be visibly disabled in many pages. All functionality not available to Limited Capability Operators will be clearly outlined in this document.

The OM GUI provides Full Capability operators with the ability to:

- Monitor for Operator Interventions and modify request parameters associated with those interventions (such as update metadata format, SCP parameters).
- View Completed Interventions.
- View list of all Distribution Requests, Processing Service Requests, Ftp Push Distribution Requests, Staging Distribution Requests, Archived Requests and Archived Processing Requests.
- Filter Distribution Requests by individual order id, request id, e-mail address, first name, or last name. Filter Distribution Requests by combinations of status, media type, order type, user id, and creation start and end time.
- From any list of Distribution Requests, perform the following actions as appropriate: change priority, resubmit, suspend, resume, cancel or stop a request.
- View detailed distribution request information and perform the following actions as appropriate:
    - Change priority, resubmit, suspend, resume, cancel or stop the request.
    - Add or change operator notes.
    - Change address information.
- View details of an ECS Order.
- View the profile of a user associated with an ECS Order.

- View suspended Ftp Push / SCP destinations and resume dispatching.
- Suspend an active destination or view non-terminal requests for the destination.
- View details for suspended Ftp Push / SCP destinations including Ftp Push / SCP Operations that caused the suspension and Ftp Push / SCP Requests that are not in a terminal state.
- View, update and cancel bundling order information (link to NSBRV GUI).
- Monitor for Operator Alerts caused by Ftp Push / SCP operations, Data Pool File System errors, Archive Server errors.
- Monitor and suspend/resume processing queue states.
- Monitor and suspend/resume staging states.
- Monitor the current staging status by media type, FTP Push or SCP.
- Configure OM Server and OM Database parameters.
- Configure the aging parameters for each ECS Priority level.
- Configure settings for each media type.
- Configure ODL metadata users.
- Configure the parameters for each external subsetter.
- Define and configure FTP Push / SCP destinations, as well as the "policies" for those destinations.
- Configure Archive Resource parameters.
- Monitor for OM Server statistics.
- Monitor for OM Staging statistics.
- Get general and context-based help for all OM GUI functions.

The OM GUI provides Limited Capability operators with the ability to:

- Monitor for Operator Interventions.
- View Completed Interventions.
- View list of all Distribution Requests, Ftp Push Distribution Requests or Staging Distribution Requests.
- Filter Distribution Requests by combinations of order id, request id, status, order type, media type, user id, first name, last name, e-mail address, or creation time.
- View detailed distribution request information.
- View processing service request information.

- View details of an ECS Order.
- View the profile of a user associated with an ECS Order.
- View archived distribution requests.
- View suspended Ftp Push / SCP destinations.
- View details for suspended Ftp Push / SCP destinations including Ftp Push / SCP Operations that caused the suspension and Ftp Push / SCP Requests that are not in a terminal state.
- View bundling order information (link to NSBRV GUI).
- Monitor for Operator Alerts caused by FTP Push operations, Data Pool File System errors, Archive Server errors, or Archive Tape errors.
- Monitor processing queue states.
- Monitor staging states.
- Monitor the current staging status by media type, FTP Push destination or SCP destination.
- View OM Server and OM Database parameters.
- View settings for each media type.
- View email settings for ODL metadata users.
- View configuration for each external subsetter.
- View FtpPush / SCP policy settings.
- View Archive Resource parameters.
- Monitor for OM Server statistics.
- Monitor for OM Staging statistics.
- Get general and context-based help for all OM GUI functions.

4.7.7.1 Starting the OM GUI

Start the web browser and then access the URL for the OM GUI web page with the format:

`http://`**`server:port`**

Example: `http://f4dp101.hitc.com:22401`

There is no need to specify a cgi-bin directory or a specific HTML page. The GUI will open itself in a new window and will close the parent window. If run on a Windows or Linux platform, the parent window may not close.

Browser Requirements

The OM GUI is certified for use with any browser supporting the Mozilla standard. Many modern browsers support this standard, including Netscape 7+, Firefox, and others. The OMS GUI was not designed to work with MS Internet Explorer or older versions of Netscape. JavaScript is an integral part of the OM GUI, and as such it must be enabled in the client browser.

Java, other scripting languages, or plug-ins are not used in the OM GUI.

4.7.7.1.1 OM GUI Home Page

The OM GUI Home Page screen shown in Figure 4.7.7-1 explains the basic services of the OM GUI. There is a static frame to the left that allows for easy and direct access to the desired pages. Due to the nature of this navigation method, the individual pages should not be viewed outside the frame environment. The navigation frame is also resizable if so desired.

Login and Sessions

The operator has the option of recalling a session by typing a name into the Login box in the left frame. This is only to recall particular session settings and is not intended for security in any way (see the GUI Security section later in this document). If the login name does not exist, a new session is created. If the operator does not choose to login, a temporary session is created.

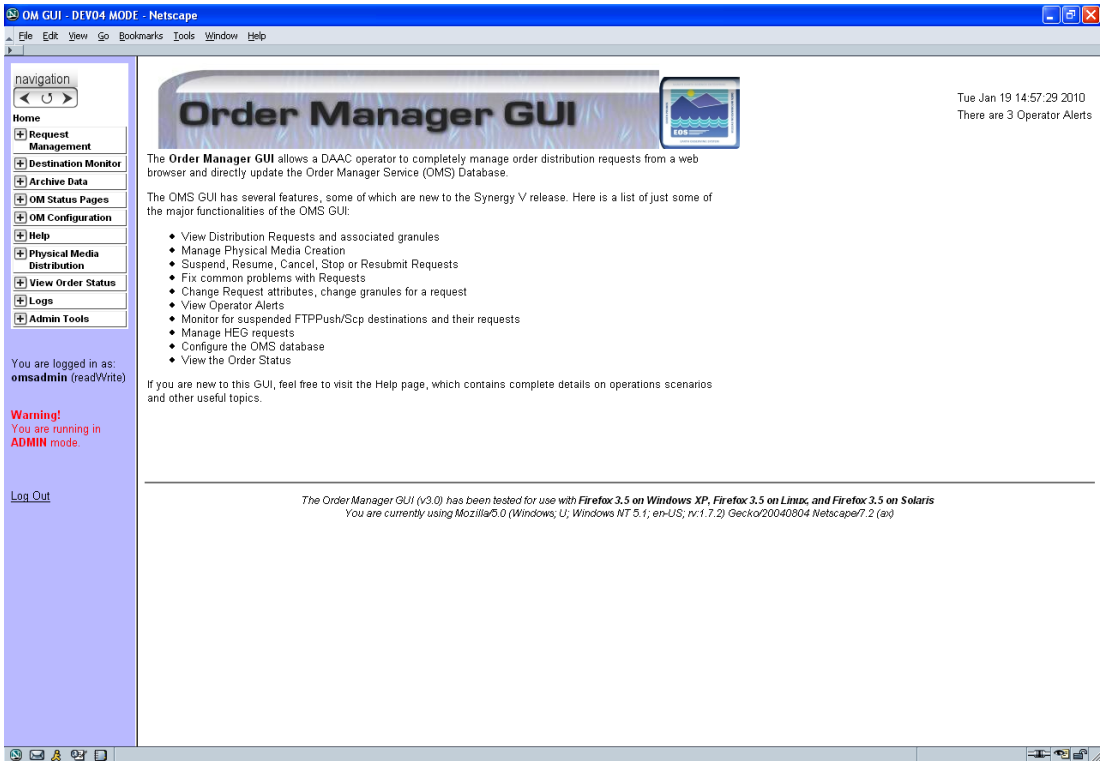


Figure 4.7.7-1. Order Manager GUI Home Page

Note: This screen shows an operator logged in using the OM GUI's non-secure login system. This only appears if the security protocols were not installed.

The operator alerts are displayed at the top of the screen.

4.7.7.1.2 GUI Security

The OMS GUI can optionally be installed with the GUI Security feature enabled. If it is, you will be prompted for a user name and password once the GUI is started. This user name will also be used as the session identifier, so that the operator can recall session settings. See Figure 4.7.7-2 for an example of the login dialog box.

User Names and Passwords

The installation team will have to create user names and passwords using special utilities. The details on this are in a different document.

GUI Security Disabled

If GUI Security has not been installed, the operator can still "log in" using the OMS GUI's proprietary login system (see "Login and Sessions" under Section 4.7.7.2). See Figure 4.7.7-2 for an example of the login dialog box.

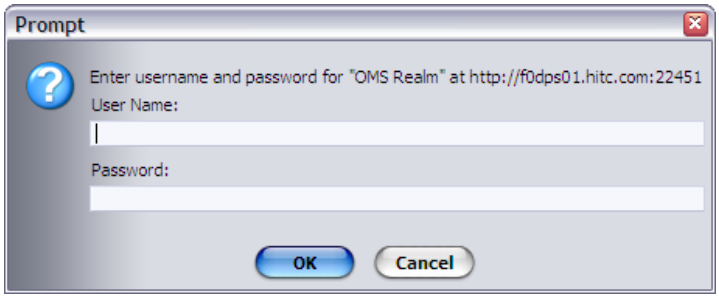


Figure 4.7.7-2. GUI Security Login

4.7.7.2 Request Management Pages

The Request Management section consists of several subsections that support a variety of capabilities allowing the operator to manage, modify, and monitor many aspects of distribution request processing. While the basic functionality of the Intervention pages remains the same, there are some enhancements, such as the ability to view Operator Interventions based on Staging errors.

In the event of a request failure, an operator intervention will appear on the "Open Interventions" page. In addition, an "Operator Alerts" page displays non-fatal warnings or errors that do not cause an Operator Intervention, but which otherwise might pose valuable to the operator. An example might be a suspended FTP Push destination.

4.7.7.2.1 Open Interventions Page

From the navigation menu, click on "Request Management" to display the available actions, and then click on **"Open Interventions"** to display the Open Interventions page, which contains a list of all the currently open Operator Interventions that require attention, as shown in Figure 4.7.7-3.

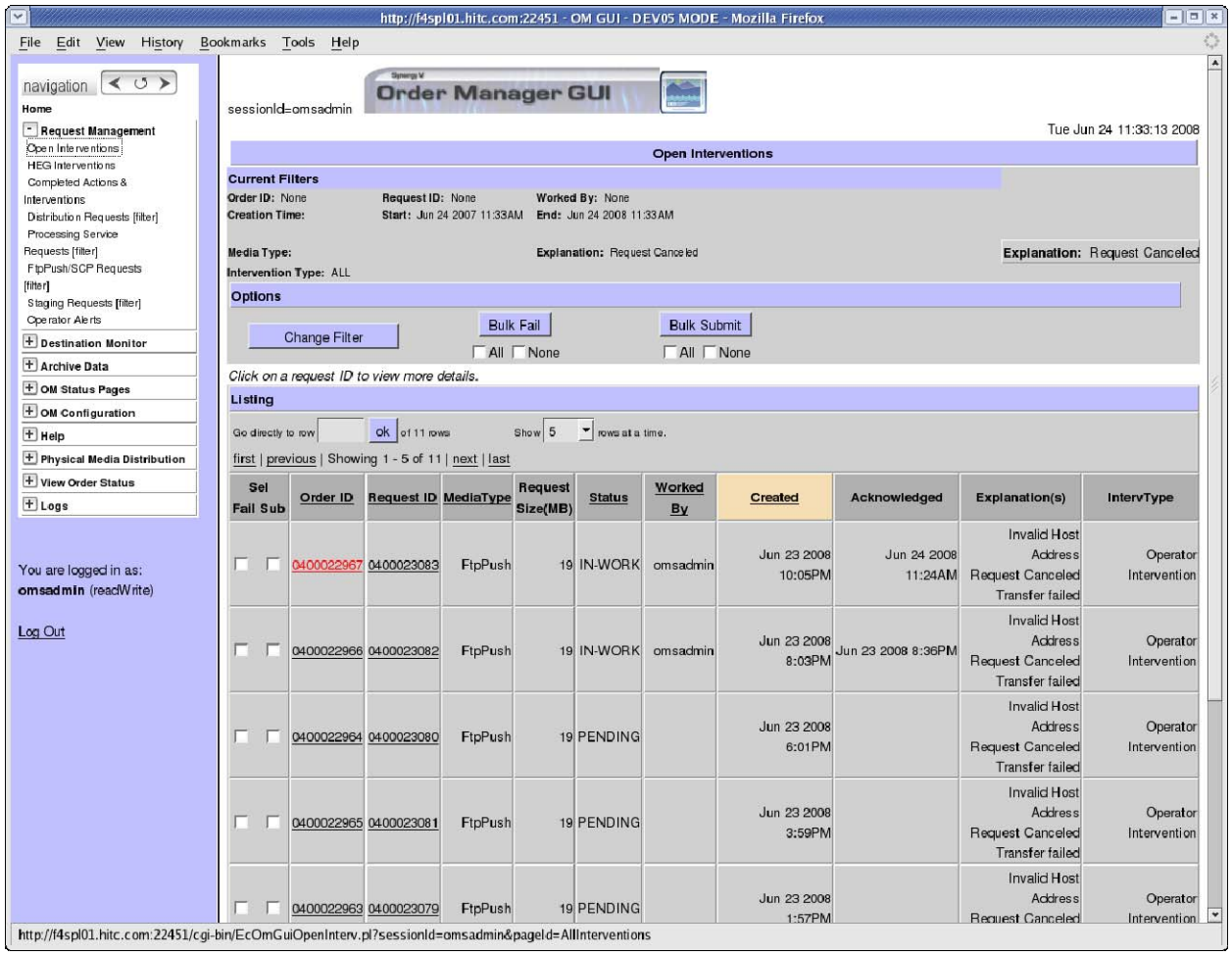


Figure 4.7.7-3. Open Interventions Page

The listing shows the Request ID that caused the intervention, as well as the associated Order ID, media type, request status, the operator who worked the intervention (no name will be shown if it has not been worked on), creation time, acknowledgement time, and the short explanation of what caused the request intervention. There are also checkboxes which can be used to select interventions to be acted upon for a **Bulk Fail** or **Bulk Submit**. Note that the highlighted column heading indicates which field is currently being used to sort the entries in the table. See Table 4.7.7-1 for descriptions of each field on this page.

Table 4.7.7-1. Open Interventions

Field Name	Description
Sel Fail	Checkbox used to indicate the intervention to be acted upon for a **Bulk Fail.** If the box is checked, the request will be failed when the **Bulk Fail** button is pressed.
Sel Sub	Checkbox used to indicate the intervention to be acted upon for a **Bulk Submit.** If the box is checked, the request will be submitted when the **Bulk Submit** button is pressed.
Order ID	The Order ID associated with the Request. Clicking on the Order ID will display a "detail" of the Order information.
Request ID	The Request ID associated with the Intervention. Clicking on the Request ID will display a detail of the Intervention.
MediaType	The media type this Order/Request uses.
Request Size(MB)	Size of the request in megabytes.
Status	The current status of the Intervention. This can be one of: PENDING: No operator has been assigned nor any action has yet been taken for the Intervention. IN-WORK: An operator has been assigned to an Intervention. This does not necessarily mean an action has been taken.
Worked By	The operator currently working the intervention. If no name appears, the Intervention has not been worked or reviewed. An operator must assign a name to the intervention before any modifications can be made.
Created	The Creation Date/Time of the Intervention.
Acknowledged	The Date/Time that an action was first taken or when an operator assigned the intervention to a worker.
Explanation(s)	A description of the nature of the error. In the case of an FTP Push failure or Staging error, a special icon will appear to make it easily recognizable.
IntervType	Intervention type (new field). For normal interventions, this is simply "Operator Intervention". Three types have been added: • HEG – Interventions related to HEG processing errors • Media Creation Error – Interventions resulting from an error at the creation stage of a physical media volume or volumes • QC Failed – Interventions resulting from an error at the QC Verification stage of a physical media volume or volumes

Interventions List Bulk Actions

The interventions list bulk actions allow the operator to act on more than one intervention at the same time. Buttons are shown on the **Options** bar for the **Bulk Submit** and **Bulk Fail** actions. When the operator clicks the **Bulk Fail** button, any intervention whose **Sel Fail** checkbox has been checked will be failed. When the operator clicks the **Bulk Submit** button, any intervention whose **Sel Sub** checkbox has been checked will be submitted.

When the operator clicks the All checkbox below the **Bulk Fail** or **Bulk Submit** buttons, the corresponding checkboxes in the interventions list will be checked. When the operator clicks the None checkbox below the **Bulk Fail** or **Bulk Submit** buttons, the corresponding checkboxes in the interventions list will be unchecked.

Intervention List Filters

As with the Distribution Request pages, the Intervention pages have a filtering capability. To access this filter, click on the "Change Filter" button at the top of the page. This will display a pop-up window, as shown in Figure 4.7.7-5, in which the user can change the filter settings. The top of the page also displays your current filtering options, as shown in Figure 4.7.7-4.

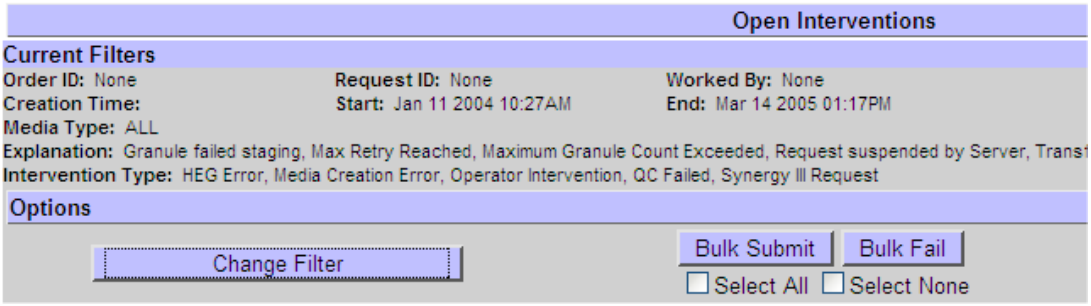


Figure 4.7.7-4. Current Intervention Filters

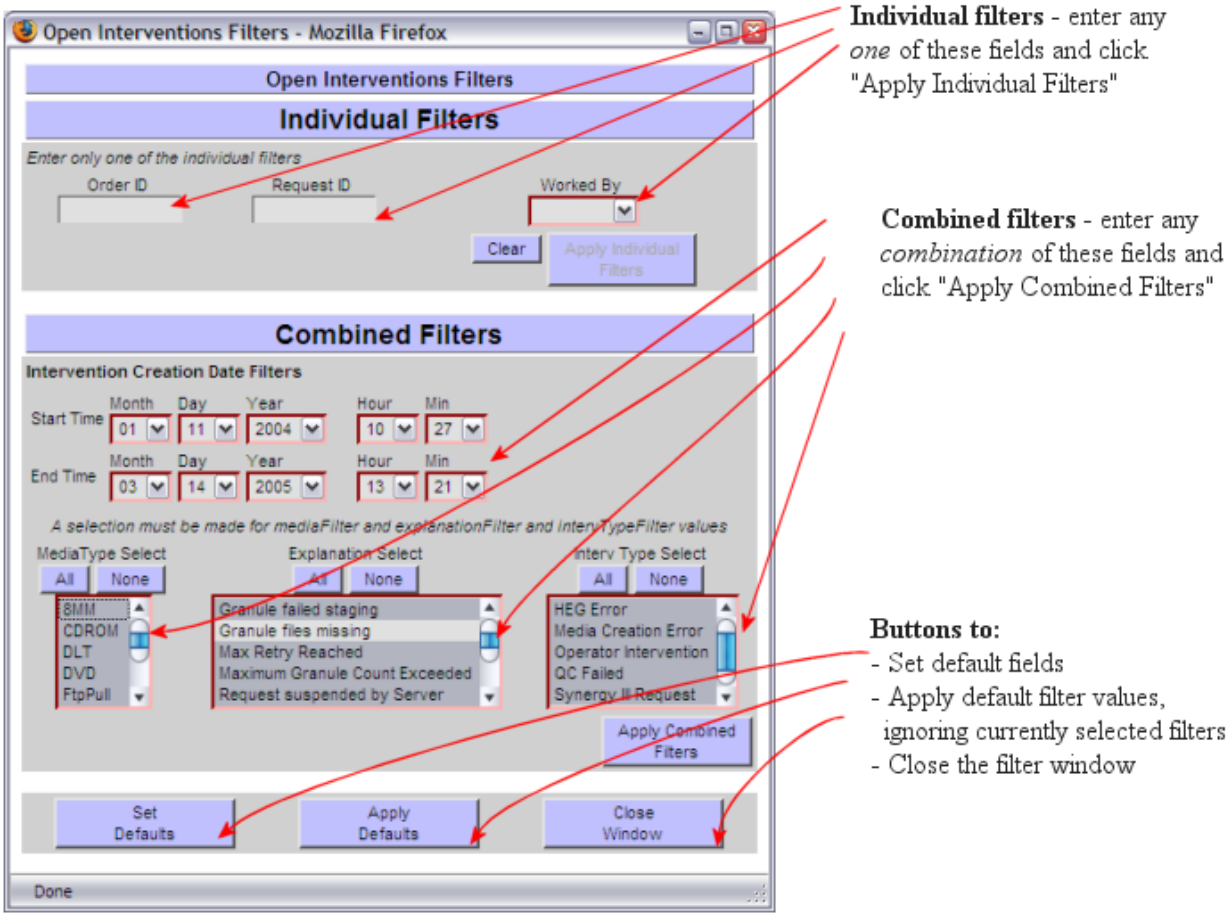


Figure 4.7.7-5. Filter Window diagram

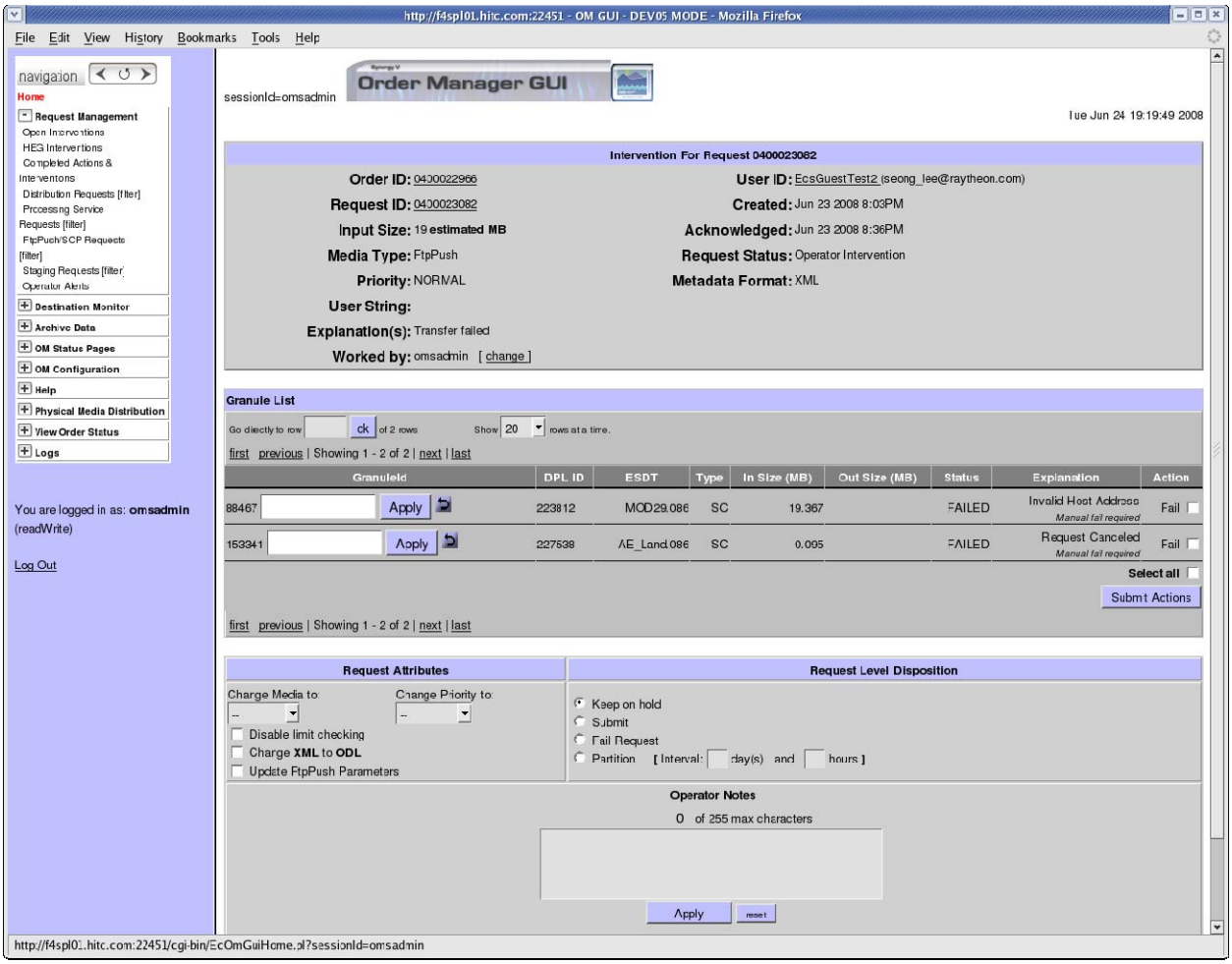


Figure 4.7.7-6. Open Intervention Detail

Note for Limited Capability Operators: The Open Intervention Detail page is limited to viewing the details of the intervention. Modifications may *not* be made to the Request or Granules for the Request. The operator is also prevented from taking any action on the Intervention.

To view the details of an intervention, click on its Request ID. This will bring you to a separate page (Figure 4.7.7-6) displaying all of the information on the previous listing, plus the user string (which would show the external request ID if the order source is the MTMGW), and the list of granules associated with the request.

From this page, the operator may take several actions to modify the request. First, any granule may be replaced with another by typing in a new granule ID and clicking “Apply”. The granules may also be failed by clicking the “Fail” button in the far right column on the row for that granule.

Please note that modifications to the granules are independent of the request attributes – i.e., any changes made will not affect the status of the request, and the request will still be in "Intervention" status until the operator submits the request. See Table 4.7.7-2 for a description of each field on this page.

Legend:

FC = Full Capability operator only (the operator can only view this field or control)

all = This field or control does not have any restrictions

Table 4.7.7-2. Open Intervention Detail Page (1 of 4)

Field Name	Perm. Level	Description
User ID	all	The "owner" of this order, in most cases the person who originated the order. Clicking on the User ID will display a complete profile of the User. In parentheses, also displayed is the e-mail address to which information about this order will be sent (e.g., a granule is failed or changed).
Priority	all	The ECS Priority level associated with this Request. These Priority levels are predetermined in the Data Pool. For example, a LOW priority might have a priority of 75. The Priority Levels can be viewed in the OM Configuration Pages under "Aging Parameters".
Order ID	all	The Order ID associated with the Request.
Request ID	all	The Request ID associated with the Intervention.
Input Size	all	The estimated size in MB of the Request.
Media Type	all	The media type this Order/Request uses.
Request Status	all	The current processing status of the Request. The Status can be one of "Intervention" or "Suspended" (this applies only FTP Push destination errors that have caused an Operator Intervention).
Worked by	FC	The operator currently working the intervention. If no name appears, the Intervention has not been worked or reviewed. An operator must assign a name to the intervention before any modifications can be made.
Created	all	The Creation Date/Time of the Intervention.
Acknowledged	all	The Date/Time that an action was first taken or when an operator assigned the intervention to a worker.
Explanation(s)	all	A description of the nature of the error. In the case of an FTP Push failure or Staging error, a special icon will appear to make it easily recognizable.

Table 4.7.7-2. Open Intervention Detail Page (2 of 4)

Field Name	Perm. Level	Description
Granule List		
GranuleId	FC	The ECS Granule ID for the granule. This is not the full Granule ID as stored in the MSS or Order Manager Database, rather it is the 16-digit ID as stored in the Data Pool database. The operator can change the GranuleId by entering the new one in the text box next to the current GranuleId and clicking apply. Granule IDs must be changed one at a time. Maximum length is 16 digits.
ESDT	all	The ESDT the granule is associated with, consisting of the ESDT short name and version ID.
Type	all	The type of granule, displayed as a two-character code. For example, SC is Science, BR is Browse, etc.
In Size (MB)	all	The input size in MB of the granule, before any processing (e.g. HEG). This field is always displayed, not matter what type of granule it may be.
Out Size (MB)	all	The output size in MB of the granule, after it has underwent processing (e.g. HEG). This field is only displayed if an output size exists in the database.
Status	all	The current status of the granule. Statuses can be: SKIPPED: The granule has been skipped because it has failed validation (e.g., the granule was not found). ***Note that FAILED and SKIPPED granules may be failed by the operator. Granules in any other state can not be failed.*** NULL: This is the initial state, essentially meaning the status is OK. TRANSFERRING: The granule is in the process of being pushed to a destination. SHIPPED: The granule has been delivered to the PDS to be put of a physical medium, or the granule has been pulled. FAILED: There are several explanations for failed granules. ***Note that FAILED and SKIPPED granules may be failed by the operator. Granules in any other state can not be failed.*** HOLD: The granules may be placed on “HOLD” if it has failed validation or there are problems writing the granules to the media.
Explanation	all	Provides a more detailed explanation of the granule Status.
Action	FC	If the granule is eligible to be failed a “Fail” button will be provided in this column.

Table 4.7.7-2. Open Intervention Detail Page (3 of 4)

Field Name	Perm. Level	Description
Request Attributes		
Disable limit checking	FC	When the request is submitted, the request size will not be taken into consideration. If the request was too small or too large, this option should be used to bypass these checks.
Change Media to	FC	Select the desired new media type for this request. If FtpPush is selected, the operator will be prompted for the FtpPush destination details on the next page.
Change Priority to	FC	Select the desired new priority for this request.
Change XML to ODL	FC	This option will only appear if the metadata format was XML. When the option is checked, the operator will receive metadata in ODL format.
Change ODL to XML	FC	This option will only appear if the metadata format was ODL. When the option is checked, the operator will receive metadata in XML format which is the default metadata format.
Update FTP Push Parameters	FC	This option will only appear if the media type was originally FtpPush. When this option is checked, the operator will be prompted to change the existing FtpPush parameters on the next page.
Update SCP parameters	FC	This option will only appear if the media type was originally SCP. When this option is checked, the operator will be prompted to change the existing SCP parameters on the next page.
Request Level Disposition		
Keep on hold	FC	This will keep the request on “Hold” – i.e., in Intervention status, and will stay on hold until the operator submits or fails the request. This option also saves the operator notes.
Submit	FC	This is in effect re-submitting the request with the altered attributes. Once the request is submitted, the Intervention is closed out. When this option is selected, the operator will be prompted to confirm the disposition on the next page (and will possibly be prompted for further details of an altered Request Attribute).
Fail Request	FC	Selecting this option will fail the entire distribution request and close out the intervention. The operator will be prompted for confirmation on the next page. A DN option is presented on the Close Confirmation page when this disposition is selected. By default, a DN will be sent, unless the operator selects the option not to send it.

Table 4.7.7-2. Open Intervention Detail Page (4 of 4)

Field Name	Perm. Level	Description
Partition	FC	This is in effect submitting the request but with the specification to partition it over the current partition size (see the Media Configuration section for more details on partitioning). If days and/or hours are provided, the request will be partitioned in this time interval. The days and hours fields must be whole numbers with no decimal fractions.
Operator Notes	FC	Up to 255 characters can be stored for notes. The notes will only be saved if a disposition is taken on the request, even if a request is failed. When a granule ID is changed, a record of the change is automatically appended in the notes.

Close Confirmation

When the actions have been finalized, click "Apply" at the bottom of the screen. This brings up the Close Confirmation page, where the operator will be prompted to verify any actions s/he wishes to take. If the action warrants an e-mail (failed request, partition, modified granules) the operator may add text to the standard e-mail preamble that will be sent to the configured e-mail address for that user. If the media type has been changed from FtpPush to a physical media type, the operator will be prompted for the shipping address. If the media type has been changed to either FtpPush or SCP, the operator will be prompted for the destination details; some of this destination information is dependent on the media type.

Note: Since Limited Capability operators cannot work on Interventions, the Close Confirmation screen will not be accessible to them.

Note: This screen is not visible to limited-capability operators.

After the operator has verified and confirmed the actions, the next screen shows the status of the submitted disposition. Figure 4.7.7-7 shows an example of a successful submission and verifies that the database has been updated with the changes. To get back to the Open Interventions listing, click OK.

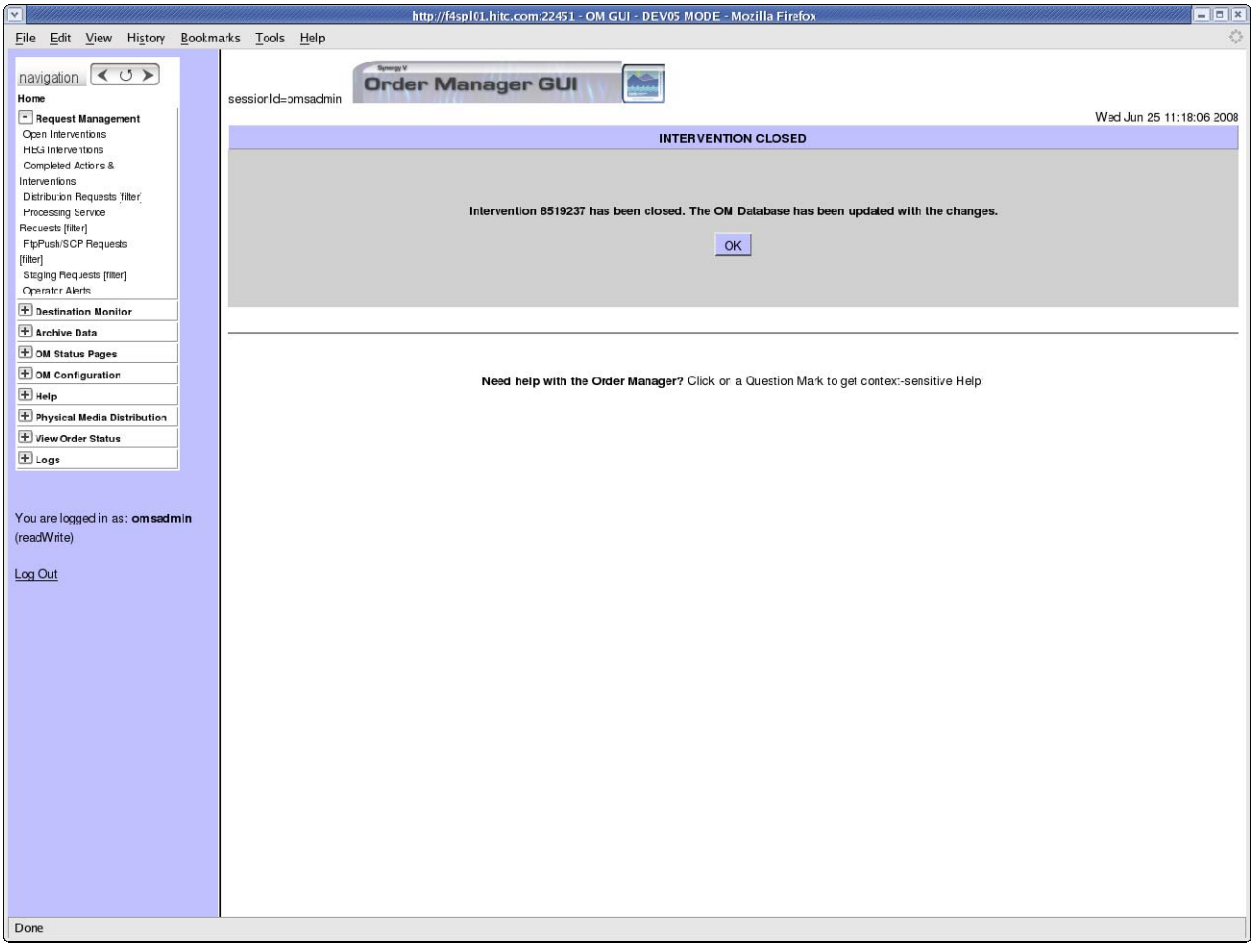


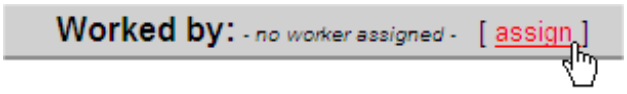
Figure 4.7.7-7. Close Confirmation Success Screen

Note: This screen is not visible to limited-capability operators.

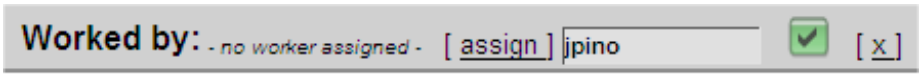
Instructions for Working an Intervention

The following are the operator steps to work on an intervention.

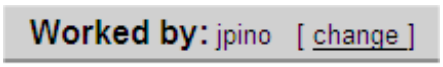
1. If a worker is not currently assigned to the intervention, *-no worker assigned-* will be displayed in the **Worked by** field. Click [assign]:



A text box will appear with your currently logged-on ID. You can also use a different ID. Click the green checkbox to assign the worker:



The page reloads with the new worker ID:



To assign a different worker, click [change] and put a new name in the textbox.

2. First, the operator can choose to fail or edit granules. For example, some granules that are inaccessible can be replaced by a new granule ID (the GranuleId). It is the operator's responsibility to obtain a suitable replacement, as the GUI/database will not automatically do this. Simply click the **Apply** button next to the granule to change it. Note: All granule changes are permanent. A granule cannot be un-failed, and no record is kept of previous granule IDs when changing the granule.
3. Next, the operator may change the request attributes, then select a disposition to close the intervention. There are four possible attributes the operator may change:
   a. **Disable limit checking –** *If this is selected, the request size limit checking will be disabled.*
   b. **Change Media to** – *Allows the request's media type to be changed to any physical or electronic media type. Some additional issues to be aware of:*
      - **Changing to SCP or FtpPush** – *When confirming the resubmission of the request, the operator will be prompted to enter destination parameters, some of which will depend on the new media type.*
   c. **Change Priority to –** *This changes the request's ECS priority level. A higher priority moves the request through the system more quickly.*
   d. **Change XML to ODL** – *This will appear if the metadata format for the request is XML. It allows the operator have the metadata to be delivered in ODL format.*
   e. **Change ODL to XML** – *This will appear if the metadata format for the request is ODL. It allows the operator have the metadata to be delivered in XML format.*
   f. **Update FtpPush Parameters** – *This will appear if the media type is FTP Push. It allows the operator to update any FTP parameters, including the destination information.*
   g. **Update SCP Parameters** – *This will appear if the media type is SCP. It allows the operator to update any SCP parameters, including the destination information.*

Dispositions

The available dispositions, or actions, the operator may make on the request are:

- **Keep on hold** – *Normally, the operator can use this disposition to add or update the operator notes on the intervention. The intervention will not be closed.*
- **Submit**. – *The operator can use this disposition to release the intervention, thus applying any new request attributes. Once the intervention is submitted, the request is no longer in*

Operator Intervention and will be sent back through validation and normal processing by the OMS Server.

- **Fail Request** – *Completely fails the distribution request, at which point it is not sent back through validation, nor will it be processed by the OMS Server.*
- **Partition** – *For cases when a request size exceeds the maximum size limit. This is effectively submitting the request (see the **Submit** option above).*

4. The operator can also add to or edit the operator notes. **(Note**: there is a 255-character limit)
5. Then click the **Apply** button. A confirmation page will display to show the disposition information. For a failed request and granules, the additional e-mail text box will display to allow operator to optionally add additional e-mail text. The default is to send e-mail for failed request or granules. However, the operator can choose not to send e-mail.

4.7.7.2.2 Operator Alerts Page

From the navigation menu, click on "Operator Alerts" to open the **Operator Alerts** page (Figure 4.7.7-12). By default, the filter is set to display all types of Alerts and the operator can filter the list for the various Alert types. The types of Operator Alerts that can be displayed are:

- FTP Push / SCP Destination Alerts (problems with the destination not causing an Operator Intervention)
- Data Pool File System Alerts
- Archive Server Alerts
- ECS Server Alerts – warnings about OMS resource errors

The list of alerts will also be sorted in ascending order by date (i.e., the oldest Alerts will appear first). For FTP Push Destination Alerts, the destination could be either a configured or a non-configured destination (not one in the Frequently Used Destinations list, as configured in the FTP Push Policy Configuration page).

The Alert Info will be shown in the column adjacent to the Alert Type. This column will contain more specific information about the nature of the problem. For example an FTP Push Alert would show the IP address (or configured alias, if appropriate) and why the destination is having problems.

For FTP Push or SCP Alerts, a link will appear in the Alert Details Column, and the operator may click on this to view a listing of all requests associated with the suspended destination. The operator can then modify the request attributes manually. No detail page is available for other types of alerts, as all of the pertinent details are already displayed.

Unlike an Operator Intervention, no specific action can be taken to close an alert. The Order Manager Server will automatically clear an alert once all conditions related to the problem have been resolved.

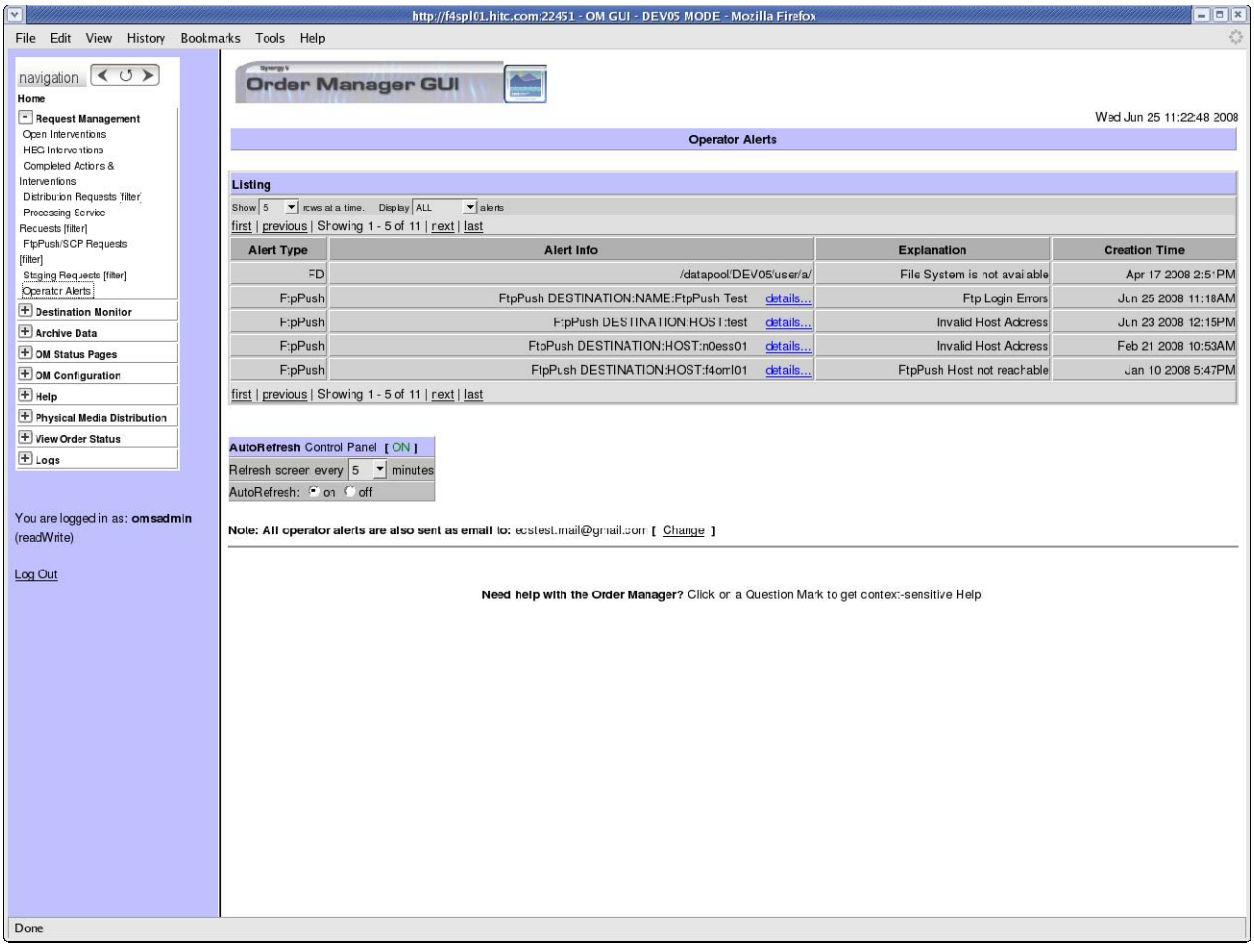


Figure 4.7.7-12. Operator Alerts

4.7.7.2.3 Completed Interventions Page

4.7.7.2.3.1 Completed Operator Actions and Interventions Page

From the navigation menu under the **Request Management** subheading, the operator can click on "Completed Actions & Interventions" to open the **Completed Operator Actions and Interventions** page (see Figure 4.7.7-13). This page displays all completed and closed Operator Interventions and Actions. Once the operator has completed work on an intervention or action, the item in that list is moved to this page. Table 4.7.7-4 describes all the fields on this page.

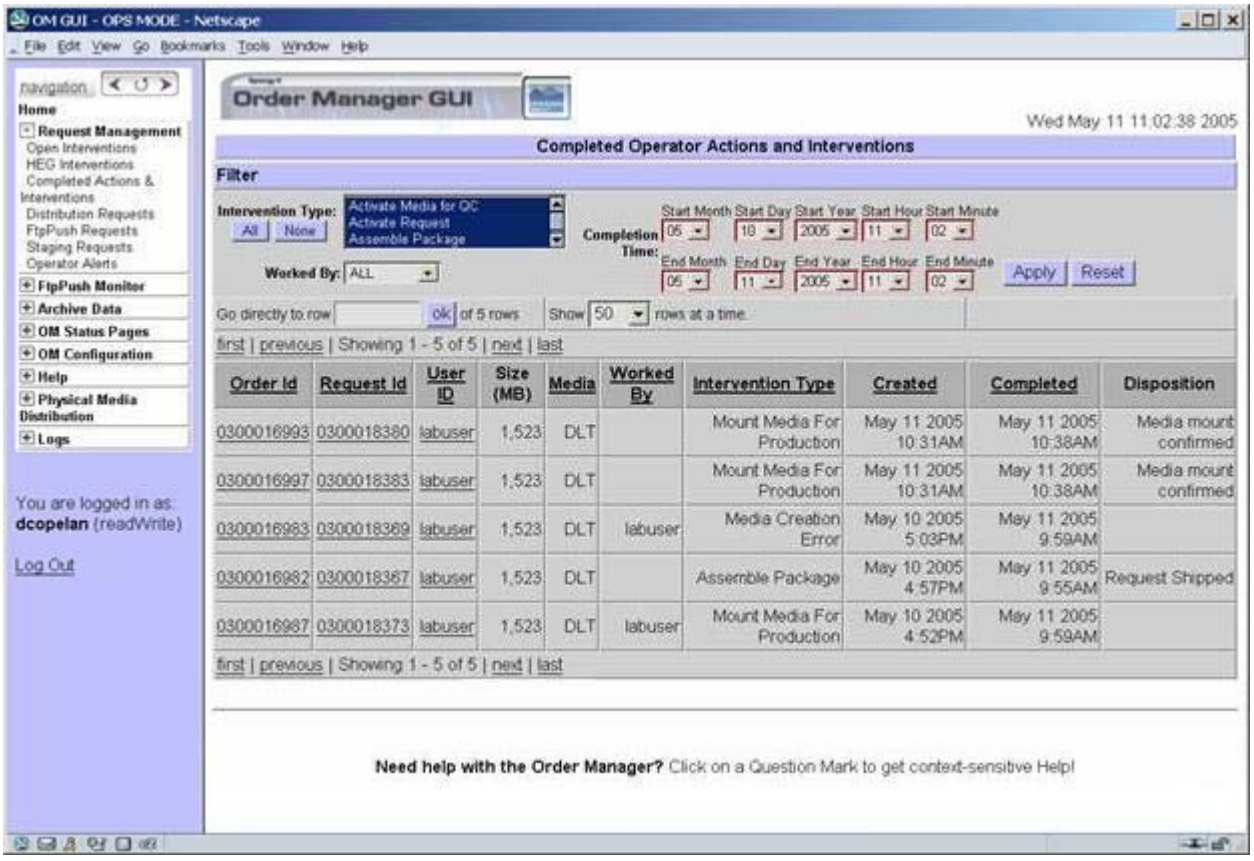


Figure 4.7.7-13. Completed Operator Actions and Interventions Page

Table 4.7.7-4. Fields on Completed Operator Actions and Interventions Page (1 of 2)

Field Name	Description
Order Id	The Order ID associated with the Request. Clicking on the Order ID will display a “detail” of the Order information.
Request Id	The Request ID associated with the Closed Intervention. Clicking on the Request ID will display a detail of the Intervention.
User ID	The “owner” of this order, in most cases the person who originated the order. Clicking on the User ID will display a complete profile of the User.
Size (MB)	The estimated size in MB of the Request.
Media	The media type this Order/Request uses.
Worked By	The operator who last worked on, resolved, or closed the Intervention.

Table 4.7.7-4. Fields on Completed Operator Actions and Interventions Page (2 of 2)

Field Name	Description
Intervention Type	The type of the Intervention or action.
Created	The Creation Date/Time of the Intervention.
Completed	The Closure Date/Time of the Interventions.
Disposition	The final action that was taken to resolve the Intervention.

Filtering the Completed Operator Actions and Interventions List

At the top of the page, the operator may select the time parameters, worker ID, and Intervention Type by which to filter the list. Once the operator clicks "Apply" in the filter window, the Completed Interventions page is reloaded with the applied filter values.

Completed Action/Intervention Detail

By clicking on a Request ID, the operator can view the same details of an Intervention or Action as contained on the Open Intervention Detail or Physical Media Console page (see Figure 4.7.7-14), except that the operator cannot take any action nor modify the Request in any way. To get back to the Completed Operator Actions and Interventions listing, the operator can click the back icon  on the top of the navigation frame.

4.7.7.2.3.2 Completed Interventions/ Actions Detail Page

When viewing the detail of a Completed Intervention, the operator can click the Order ID to view the Order information. Table 4.7.7-5 describes each field on this screen.

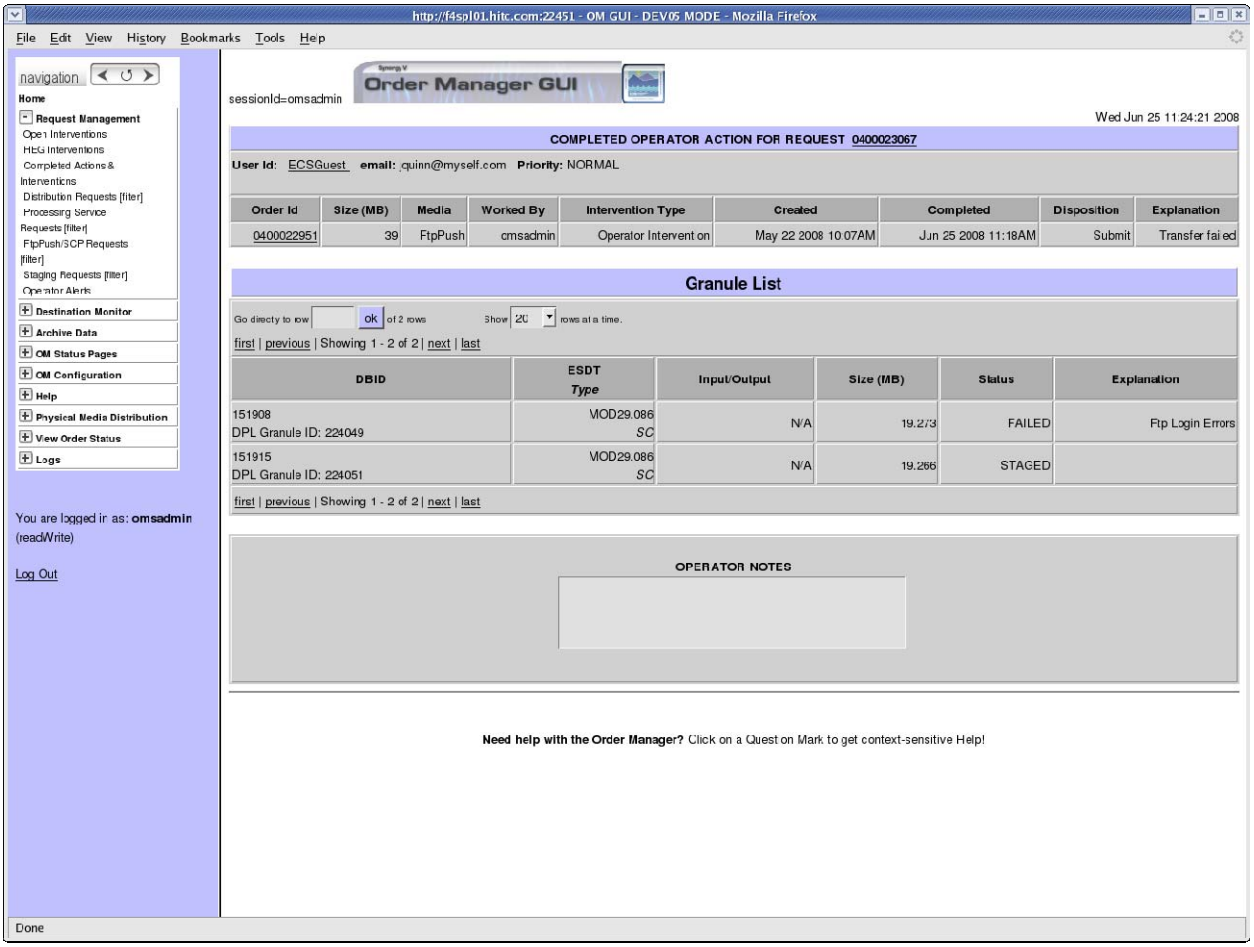


Figure 4.7.7-14. Completed Intervention/Action Detail Page

Table 4.7.7-5. Fields on Completed Intervention Detail Page (1 of 2)

Field Name	Description
User Id	The “owner” of this order, in most cases the person who originated the order. Clicking on the User ID will display a complete profile of the User.
email	The e-mail address to which information about this order will be sent (e.g., a granule is failed or changed).
Priority	The ECS Priority level associated with this Request. These Priority levels are predetermined in the Data Pool. For example, a LOW priority might have a priority of 75. The Priority Levels can be viewed in the OM Configuration Pages under “Aging Parameters”.
Order Id	The Order ID associated with the Request. Clicking on the Order ID will display a “detail” of the Order information.
Size (MB)	The estimated size in MB of the Request.

Table 4.7.7-5. Fields on Completed Intervention Detail Page (2 of 2)

Field Name	Description
Media	The media type this Order/Request uses.
Worked By	For nonphysical media requests, shows the name of the worker who last worked this intervention.
Intervention Type	The type of the Intervention or action.
Created	The Creation Date/Time of the Intervention.
Completed	The Date/Time the Intervention was completed.
Disposition	The final action that was taken to resolve the Intervention.
Explanation	This is the explanation of any errors that occurred on the granule-level.
Fields on the Granule List	
DBID	The Database ID or "Granule ID" for the granule. This is not the full Granule ID as stored in the MSS or Order Manager Database, rather it is the 16-digit ID as stored in the Data Pool database.
ESDT Type	The ESDT type the granule is associated with, consisting of the ESDT short name and version ID.
Size (MB)	The size in MB of the granule.
Status	**The current status of the granule. Statuses can be:** SKIPPED: The granule has been skipped because it has failed validation (e.g., the granule was not found). NULL: This is the initial state, essentially meaning the status is OK. TRANSFERRING: The granule is in the process of being pushed to a destination. SHIPPED: The granule has been delivered to the PDS to be put of a physical medium, or the granule has been pulled. FAILED: FTP Push transfer failure. HOLD: The granules may be placed on "HOLD" if it has failed validation or there are problems writing the granules to the media.
Processing Instructions	Will be displayed when viewing a HEG intervention. A HEG order may contain a mix of granule types (those with and without processing instructions), if there are any to display, an additional column will be shown in the granule list. This column shows a link to view the processing instructions details, if any.
Explanation	Provides a more detailed explanation of the granule Status.
Operator Notes Box	
OPERATOR NOTES	This will contain a record of the DBID changes, plus any notes the operator may have manually typed in.

Links to Other Pages

The operator may click on the Request Id, Order Id or User Id to view the Request Detail, or Order pages, respectively, associated with the request. For HEG requests, the operator may click on the **View** link to view the processing instructions associated with the granule.

4.7.7.2.4 Distribution Requests Pages

The subsections are:

- Distribution Requests
- Processing Service Requests
- Destination Monitor
    - Suspended Destinations
- Staging Requests
- Historical Requests
- Historical Processing Requests
- Order Detail

Lists of distribution requests also appear on the Order page, for bundling orders only, and on the Destination Detail page, requests not in a terminal state only. All actions that apply to other lists of distribution requests are available on these pages as well.

4.7.7.2.4.1 View Distribution Requests

There are six pages that display a Distribution Requests list. These are:

- Distribution Requests (All)
- Processing Service Requests
- FtpPush/SCP Requests
- Staging Requests
- Historical Distribution Requests
- Destination Monitor
- Order Page – Bundling Orders only

These pages share many common features. These shared features will be described in the next section, followed by descriptions of features that are unique to each page.

4.7.7.2.4.2 Distribution Requests Lists – Common Features

Request Lines

Each line of the request list shows pertinent fields for a specific request. A few fields are not shown in every list. These are specified in the unique features sections. Table 4.7.7-6 is a list of fields that appear for every request line.

Table 4.7.7-6. Fields Displayed (1 of 2)

Field Name	Data Type	Size	Description
Ord Typ	Character	8	“Regular”, “Bundled”, “MM” or “HEG”.
OrderID	Integer	8	UID for this order created internally. This is a link to the Order page for this order.
RequestID	Link/Integer	10	UID for a request. This is a link to the Request Detail page.
Request Size(MB)	Integer	8	Cumulative size of granule science/metadata files in MB. Formatted as follows: for zero value – “0”, for value > 0 and < .5 – “< .5”, for all other values -rounded to the closest integer.
Gran Cnt	Integer	8	Number of granules associated with the request.
Staging Complete (Staging Requests Page)	Integer	8	Number of granules that have completed staging.
Complete (FtpPush /SCP Requests Page)	Integer	8	Number of granules that have completed FTP Push / SCP.
Media	Character	8	Type of media associated with the request.
Priority	Character	6	This is a list of possible request priorities if the following conditions do not apply, the request: • is in a terminal state , • has been submitted to PDS, • has a status of “QC Hold” or “Waiting for Shipment”, • has a status of “Pending Media Prod" and the dispatch mode for its media type is manual, • has a status of “Transferring " and has a device assigned to it. The current priority of the request is highlighted and can be changed. If the request is in a terminated state, no priority is displayed. Otherwise, the current priority is displayed and cannot be changed.
Apply (priority) Button	Button	n/a	Click to change the priority of the request to the selected value.

Table 4.7.7-6. Fields Displayed (2 of 2)

Field Name	Data Type	Size	Description
Request Status	Character	21	Current status of the request. If the status is "Operator Intervention" and an OMS intervention exists, the status will be a link to the Intervention Detail page for the intervention.
Resource Class	Character	9	Resource class is an indicator of resource utilization based on archive resource demand. Values are: Cheap, Moderate, or Expensive.
ESDT	Character	12	Earth science data type.
UserID	Character	8	Identification of the user submitting the request.
Resub Cnt	Integer	5	Number of times specified request has been resubmitted.
Created	Date/Time	19	Date/time the request was created.
Last Update	Date/Time	19	Date/time the request was last updated.
Actions	Buttons	8	One button for each Action for which the request is eligible. See section "Actions" for details.

Navigation

The operator can scan through the list of requests by clicking on navigation links. These links permit selecting a specific starting row of requests or jumping to the **first**, **next**, **previous** or **last** block of requests. The operator can jump to a specified row by entering the row number in the box in the **Go directly to row __** of *n* **rows** line and clicking on the **OK** button. The pull down menu lets the operator select the number of requests displayed on a page; the available values are 5, 10, 20, 50 and 100 requests per page. If a value has been configured for the list, it will also be included as a selectable value. Table 4.7.7-7 provides descriptions of the navigation fields for the Distribution Requests page.

Table 4.7.7-7. Request Management Page Navigation Field Descriptions

Field Name	Data Type	Size	Entry	Description
Go directly to row (line no.) of nnnn rows	Integer	5	Optional	Line number of request to display at the top of the list.
ok	Button	n/a	Optional	Refreshes the list starting with request line entered.
first	Link	n/a	Optional	Selects first block of requests.
previous	Link	n/a	Optional	Selects previous block of requests.
next	Link	n/a	Optional	Selects next block of requests.
last	Link	n/a	Optional	Selects last block of requests.
Show nn rows at a time	Drop down list	3	Optional	Number of rows (nn) to display in the Distribution Requests listing. Default value is taken from the configuration file.

Refresh

This page will be refreshed by default every 5 minutes. The operator can change the refresh rate by selecting from the pull down menu. The operator can also choose to suspend refresh by clicking the **AutoRefresh Control Panel** on/off button. If any field is changed the new value is stored and the page refreshes immediately. See Table 4.7.7-8 for Field Descriptions.

Table 4.7.7-8. Request Management Page Refresh Field Descriptions

Field Name	Data Type	Size	Entry	Description
AutoRefresh	Toggle switch	n/a	Optional	Turns auto-refresh on or off depending upon the current state.
Auto-refresh screen every nn minutes	Drop down list	2	Optional	Interval in minutes for screen auto-refresh. Values are 1, 5, 10, 15, 30, 45, and 60.

Filters

The list of current filters for the displayed request list is shown at the top of the page. To change these filter values, the operator clicks on the **Change Filter** button. This causes a pop-up window containing fields for changing the various filters to appear. Once the operator has selected the desired filters and clicks the **Apply Individual Filters, the Apply Combined Filters** or the **Apply Defaults** button, the Distribution Requests list will be refreshed with the new filters. The Distribution Requests Filters page (Figure 4.7.7-15) field descriptions are shown in Table 4.7.7-9.

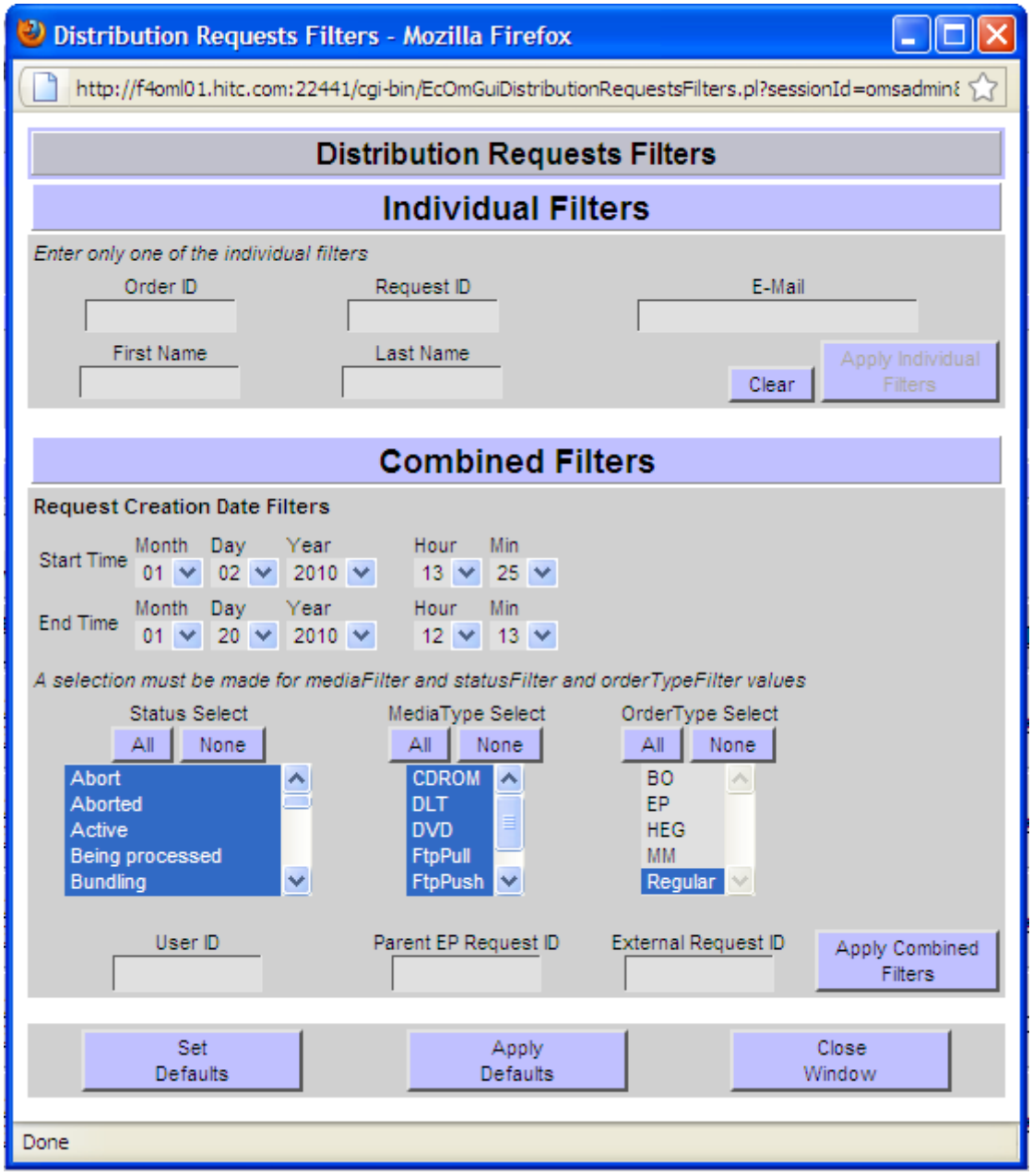


Figure 4.7.7-15. Distribution Request Filter Popup

Table 4.7.7-9. Distribution Requests Filter Page Field Descriptions (1 of 3)

Field Name	Data Type	Size	Entry	Description	Default Value
Individual Filters – only one item from this group may be entered					
Order ID	Integer	11	Optional	Order ID of requests to be selected.	None
Request ID	Integer	11	Optional	Request ID of request to be selected.	None
E-Mail	Character	15	Optional	E-Mail address of requests to be selected.	None
First Name	Character	12	Optional	First Name of requests to be selected.	None
Last Name	Character	12	Optional	Last Name of requests to be selected.	None
Clear Button	Button	n/a	Optional	Clears value in any field in this group and disables the Apply Individual Filters button.	n/a
Apply Individual Filters	Button	n/a	Optional	Applies the field in Individual filter group which has text entered.	n/a
Combined Filters – these filters will be “anded”. At least one value for Status and Media Type is required.					
Creation time from/to	Character	n/a	Required Defaults need not be changed	Select from pull-down lists to specify a starting date and time and an ending date and time for filtering.	To: current date/time. From current date/time minus 24 hours.
Status Select - All	Button	n/a	Optional	Selects all status values in the status scrolling list.	n/a
Status Select - None	Button	n/a	Optional	De-selects all status values in the status scrolling list. The warning message “A selection must be made..” is highlighted until a selection for status is made.	n/a
Status Select List	Scrolling List	n/a	Optional	Clicking on an entry in the list selects it if it is de-selected or de-selects it if it is selected. Any number of entries may be selected.	All statuses are selected.

Table 4.7.7-9. Distribution Requests Filter Page Field Descriptions (2 of 3)

Field Name	Data Type	Size	Entry	Description	Default Value
MediaType Select - All	Button	n/a	Optional	Selects all media type values in the media type scrolling list.	n/a
MediaType Select - None	Button	n/a	Optional	De-selects all media type values in the media type scrolling list. The warning message “A selection must be made...” is highlighted until a selection for media type is made.	n/a
MediaType Select List	Scrolling List	n/a	Optional	Clicking on an entry in the list selects it if it is de-selected or de-selects it if it is selected. Any number of entries may be selected.	All Media Types are selected.
OrderType Select - All	Button	n/a	Optional	Selects all order type values in the media type scrolling list.	n/a
OrderType Select - None	Button	n/a	Optional	De-selects all order type values in the media type scrolling list. The warning message “A selection must be made...” is highlighted until a selection for media type is made.	n/a
OrderType Select List	Scrolling List	n/a	Optional	Clicking on an entry in the list selects it if it is de-selected or de-selects it if it is selected. Any number of entries may be selected.	All Order Types are selected.
User ID	Character	13	Optional	User ID, entered to specify a user ID for filtering.	None
Parent EP Request ID	Character	13	Optional	Request ID of the Parent EP request	None
External Request ID	Character	13	Optional	External Request ID, entered to specify an external request ID for filtering	None
Apply Combined Filters	Button	n/a	Optional	Applies above “Combined” filters to the request list.	n/a

Table 4.7.7-9. Distribution Requests Filter Page Field Descriptions (3 of 3)

Field Name	Data Type	Size	Entry	Description	Default Value
General Buttons					
Set Defaults	Button	n/a	Optional	Sets all filter selections to their default values on the Filters page.	n/a
Apply Defaults	Button	n/a	Optional	Sets all filter selections to their default values on the Filters page and applies these values to the corresponding requests List Page.	n/a
Close Window	Button	n/a	Optional	Closes the Requests Filter window.	n/a

Any attributes that the operator selects/enters will be remembered for the duration of the session and for future sessions when the operator logins in with the same User ID, but only those in the group whose Apply button has been clicked will be used to filter the distribution requests list. There are two categories of filtering attributes -- Individual Filters and Combined Filters. Either of these filter categories can be applied at one time.

To select Individual Filters, the operator enters one of the five fields displayed: Order ID, Request ID, E-Mail, First Name and Last Name. If a value is entered in one of the five fields, the other four fields become disabled. Deleting the entered value or clicking the Clear button reenables all of the Individual Filter fields. Clicking the Apply Individual Filters button applies the entered field entry and reloads the Distribution Requests window with the results.

To select Combined Filters, the operator selects or enters values for the desired attributes. The Creation Date Filters are initially set to: End Time - the current date/time, and Start Time - 24 hours before the current date/time. If initial (default) date/time values are not changed, they will update to the current time whenever they are applied. The operator can change these attributes by clicking on the down triangle, which appears next to the value of each attribute, and then clicking on a value from the drop-down list that is displayed. The drop-down lists show all possible values for month, day, hour and minute. For the year, only the current year and the previous year are shown for selection.

At least one value must be selected for each of Status, MediaType, and OrderType attributes. The selected/entered attributes are “anded” for filtering. This means that only requests having all of the selected attribute values will be displayed. If any of the Status, MediaType, and OrderType attributes is not selected, the warning message “A selection must be made …” is highlighted and the Apply Combined Filters button is disabled until the required values are selected.

The Status Select, MediaType Select, and OrderType Select lists initially display all possible statuses/media types/order types for a request with all values selected. The operator can click on the **None** button to deselect all entries in a list or **All** button to select all entries again. Also, the

operator can click on an individual status/media type entry in the scrolling list to select or deselect it. If the entry was not selected, it will be selected. If the entry was selected, it will be deselected. Any number (more than 0) or combination of statuses, media types or order types may be selected. To select multiple values from one list, hold down the Ctrl key while clicking on values after the first. To select a range of values from one list, click on the value at the start of the range and then hold down the Shift key while selecting the value at the end of the range.

All Combined Filter attributes will be applied when the operator clicks the Apply Combined Filters button at the lower right corner of the group. The Distribution Requests window will be reloaded filtered by the selected/entered attributes.

The three buttons at the bottom of the window are Set Defaults, Apply Defaults and Close Window.

- **Set Defaults** restores the default values to all filter attributes shown on the filters page to global default values. The distribution requests page is not updated. The operator can make additional changes to the filters before applying them to the distribution requests page by using the "Apply Individual Filters" or "Apply Combined Filters" buttons.

- **Apply Defaults** restores the global default values to all filter attributes on the Filter page, and applies these values to the distribution requests page. The "applied" values will be used in the future until they are changed.

- **Close Window** closes the Request Filters window. It does not affect the Distribution Requests window.

Default global values are:

For Individual Filters, all values are cleared (made empty).

For Combined Filters, Table 4.7.7-10 shows the global default values by page.

Table 4.7.7-10. Global Default Values by Page

Page	Element	Default Value
Distribution Requests	End Time	The current date and time.
	Start Time	24 hrs prior to the End Time.
	Status	All values are selected.
	Media Type	All values are selected.
	Order Type	All values are selected.
Processing Service Requests	End Time	The current date and time.
	Start Time	24 hrs prior to the End Time.
	Status	All values are selected.
	Media Type	All values are selected.
FTP Push/ SCP Distribution Requests	End Time	The current date and time.
	Start Time	24 hrs prior to the End Time.
	Status	All values are selected.
Staging Distribution Requests	End Time	The current date and time.
	Start Time	24 hrs prior to the End Time.
	Status	All values are selected.
	Media Type	All values are selected.
Historical Distribution Requests	End Time	The current date and time.
	Start Time	One(1) month prior to the End Time.
	Status	All values are selected.
	Media Type	All values are selected.
	Order Type	All values are selected.
Open Interventions	End Time	The current date and time.
	Start Time	One(1) year prior to the End Time.
	Media Type	All values are selected.
	Explanation	All values are selected.
	Interv Type	All values are selected.
Open HEG Interventions	End Time	The current date and time.
	Start Time	One(1) year prior to the End Time.
	Media Type	All values are selected.
	Explanation	All values are selected.

The ECS ORDER and Destination Detail pages have fixed filters that cannot be changed by the operator. Table 4.7.7-11 shows the filter values used for these pages.

Table 4.7.7-11. Filters for The ECS ORDER and Destination Detail Pages

Page	Element	Default Value
ECS ORDER	End Time	The current date and time.
	Start Time	Jan 1 1900
	Status	All statuses
	Media Type	All media types
	OrderId	Current orderId
Destination Detail	End Time	the current date and time
	Start Time	Jan 1 1900
	Status	All statuses
	Destination Node or Name	Current destination

The Distribution Requests Filters window remains open until the operator clicks the Close Window button at the bottom of the window or until its corresponding distribution requests page is replaced by another page.

Sorting

The request list can be sorted by clicking on the column header links **Order Typ**, **Request ID**, **Order ID**, **Destination**, **Complete**, **Media, Priority**, **Request Status, Capacity Class**, **User ID**, **Created** and **Last Update** wherever they appear. The default sort column is **Created** (creation date).

Actions

Note: Limited Capability operators are not allowed to execute actions for requests.

The operator can execute the following actions for any request that is eligible for the action by clicking on the button of the action. The action buttons will appear for only actions for which the request is eligible. Table 4.7.7-12 explains the actions and the criteria for a request to be eligible for each action.

If the request processing state is “Cancelling,” “Resuming,” “Resubmitting,” “Stopping,” "Submitted to PDS" or "Granule Canceled,” the processing state will be displayed in the action column and no actions are permitted.

Table 4.7.7-12. Eligibility Criteria for Each Action (1 of 2)

Action	Description	Criteria for Eligibility
Resubmit	Opens a new intervention for the request and loads the “Intervention Detail” page for subsequent action.	The request is in a terminated status (including cancel, abort, aborted and shipped).
Suspend	Suspends the request. The request is suspended, the distribution requests page is reloaded and the highlighted message “Suspending” is displayed in the Action column for the request until the OMS server completes the suspension of the request.	The request is not in a terminated status; And is not currently suspended and either: 1. non-failed granules still need to be staged or Ftp pushed and is not a physical media request with status "Transferring", "QC Hold" or "Waiting for Shipment"; or 2. is a physical media request with status "Pending Media Prod" and the dispatch mode for its media type is “automatic”.
Resume	Resumes the request. A small popup window, “Confirm Resume for Request ID”, appears for entry of the Worker name and Reason for Action. When login security is on, the operator’s login id is inserted in the Worker name field. When the operator clicks the “Resume” action button, the request is resumed, the distribution requests page is reloaded and the highlighted message “Resuming” is displayed in the Action column for the request until the OMS server completes the resumption of the request.	The request is not in a terminated status; is suspended; was suspended by the operator; an OMS intervention exists; Or is a new request and processing of new requests is suspended.
Stop	Stops the request. The request is stopped, the distribution requests page is reloaded and the highlighted message “Stopping” is displayed in the Action column for the request until the OMS server completes the stopping of the request.	The request is not in a terminated status; is a physical media request; the Request Status is “Transferring”; Or the Request Status is “QC Hold” and at least one volume is “Verifying.”

Table 4.7.7-12. Eligibility Criteria for Each Action (2 of 2)

Action	Description	Criteria for Eligibility
Cancel	Cancels the request. A small popup window, "Confirm Cancel for Request ID", appears for entry of the Worker name and Reason for Action. When login security is on, the operator's login id is inserted in the Worker name field. The operator is informed that any physical media volumes that are assigned to devices will be considered dismounted. When the operator clicks the 'Apply "Cancel Action"' button, an action is queued for the Order Manager server to cancel the request. The distribution requests page is reloaded and the highlighted message "Cancelling" is displayed in the Action column for the request until the OMS server finishes Cancelling the request. No other action buttons will be shown.	The request is not in a terminated status And is not suspended and has no OMS intervention.
Inactive	For external processing requests, if the request is in the terminal state or not under OMS control, the "Inactive" button is displayed which indicates no action for the request with current status.	The request is in a terminated status and is not under OMS control with status "waiting for data".

The OM GUI is designed to present to the operator only those Action buttons for which the request is eligible. However, if an action is not activated for a period of time, the Action may become "stale" if circumstances occur which change the status of the request such that it is no longer eligible for that Action. For example, the request may be canceled by an operator using a different instance of the OM GUI or the request may have terminated normally. In that case, when the operator clicks the Action button, an error message will be displayed by the database procedure which executes the action. After reading the message to understand the cause of the error, the operator may return to the original page (by using the Link provided) and refresh/reload that page to see the currently available actions.

Change Priority

Note: Limited Capability operators are not allowed to change the priority of a request.

The priority of a request can be changed while the request is eligible to have its priority changed. The criteria which determine when a request is eligible to have its priority changed are described in Table 4.7.7-6 The operator can change the priority of a distribution request by clicking on its Priority value and selecting the desired new priority value from the drop-down list. Then the operator must click on the associated **Apply** button. Once the new priority has been applied, the priority cell will display the highlighted message "Priority Changed".

Links

OrderID The operator can view the detailed information for the order to which a distribution request belongs by clicking on its OrderID.

RequestID The operator can view the detailed information for a distribution request by clicking on its Request ID.

UserID The operator can view the detailed information about the user who submitted the order containing the distribution request by clicking on its UserID.

Refresh Control

The operator can also choose to enable/disable auto-refresh by clicking on the corresponding AutoRefresh radio button. The operator can also change the refresh rate by selecting a rate from the pull down list (default 5 minutes).

4.7.7.2.4.3 Distribution Requests Lists – Unique Features

Distribution Requests Page

The following additional option buttons are available on the Distribution Requests page as shown in Figure 4.7.7-16b.

- **Bulk Cancel**
  - If there are Physical Media requests that are bulk cancelled the media volumes for these requests will be considered dismounted. The Bulk Cancel pop-up window is shown in Figure 4.7.7-16a.
- **Bulk Claim**
  - Operator Interventions can be bulk claimed using this button.
- **Bulk Resubmit**
  - Select all eligible requests for **Bulk Cancel** or **Resubmit**
  - Select no eligible requests for **Bulk Cancel** or **Resubmit**

Figure 4.7.7-16a. Bulk Cancel Popup

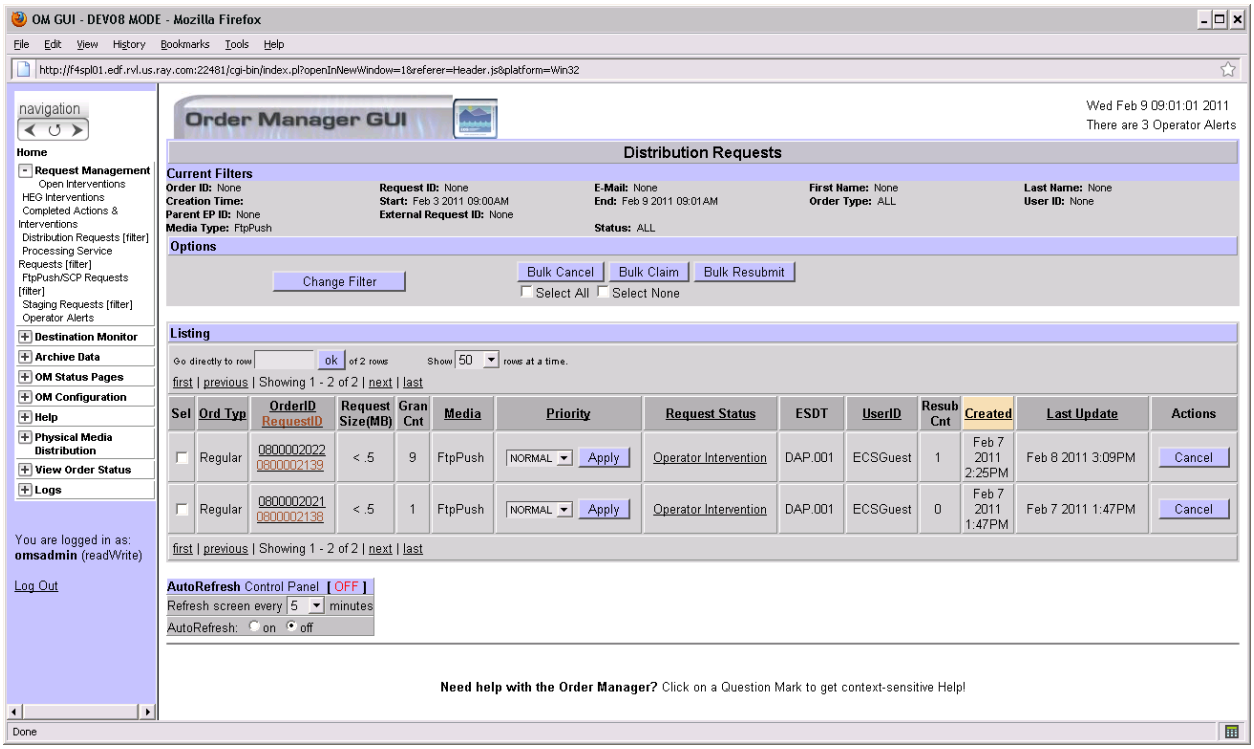


Figure 4.7.7-16b. Distribution Requests List Page

Processing Services Requests

The Processing Services Requests page includes HEG and all external subsetter requests and "Processor" column is going to indicate the processor name. It does not have a filter for media type and order type. It will have processing service filter instead. All the external processing requests do not have any actions (cancel or suspend) while they are under the control of the external system.

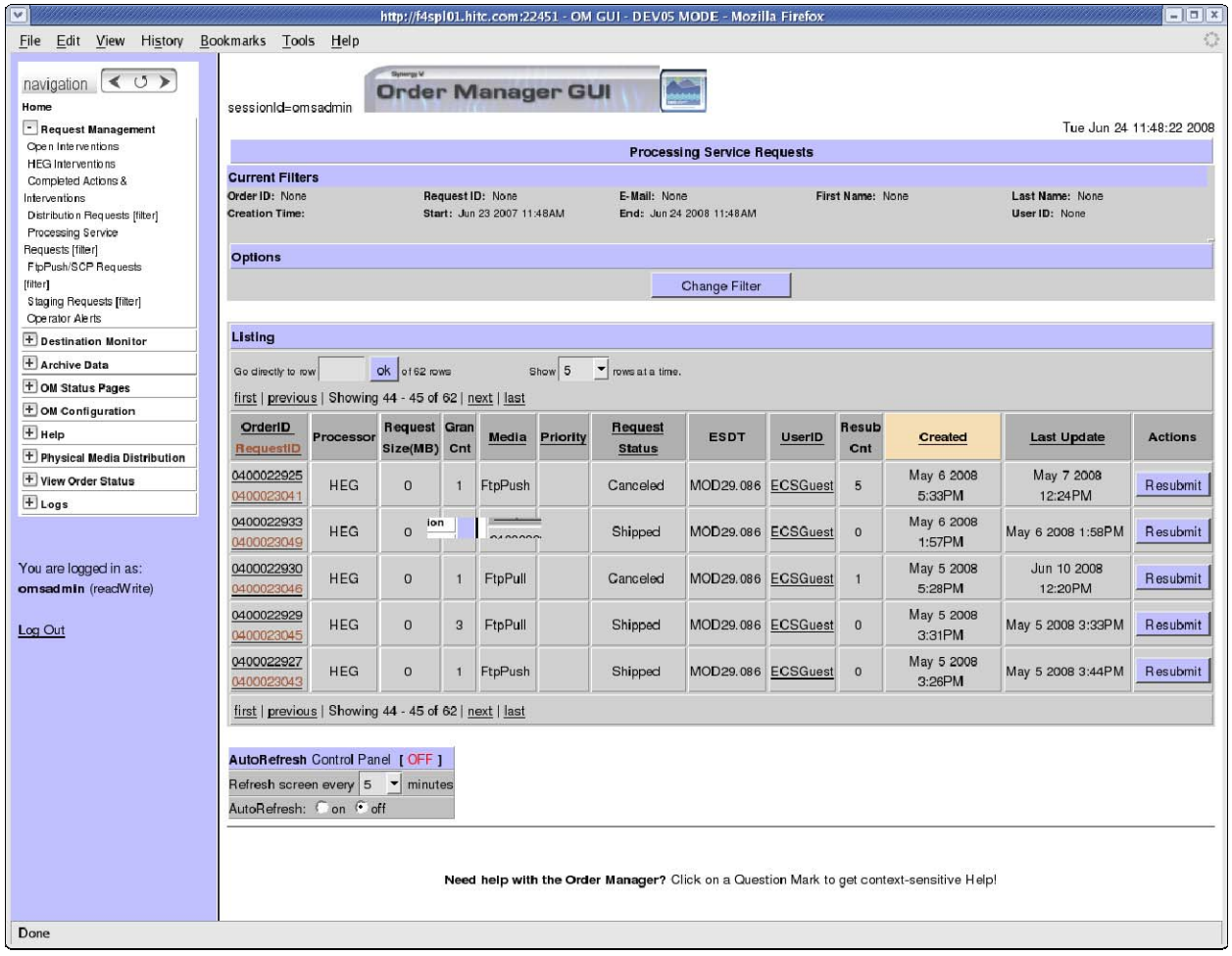


Figure 4.7.7-17a. Processing Services Requests Page

Figure 4.7.7-17a displays the similar filter and sort capabilities for the external processing requests as for the general list of distribution requests except for the “Inactive” action button.

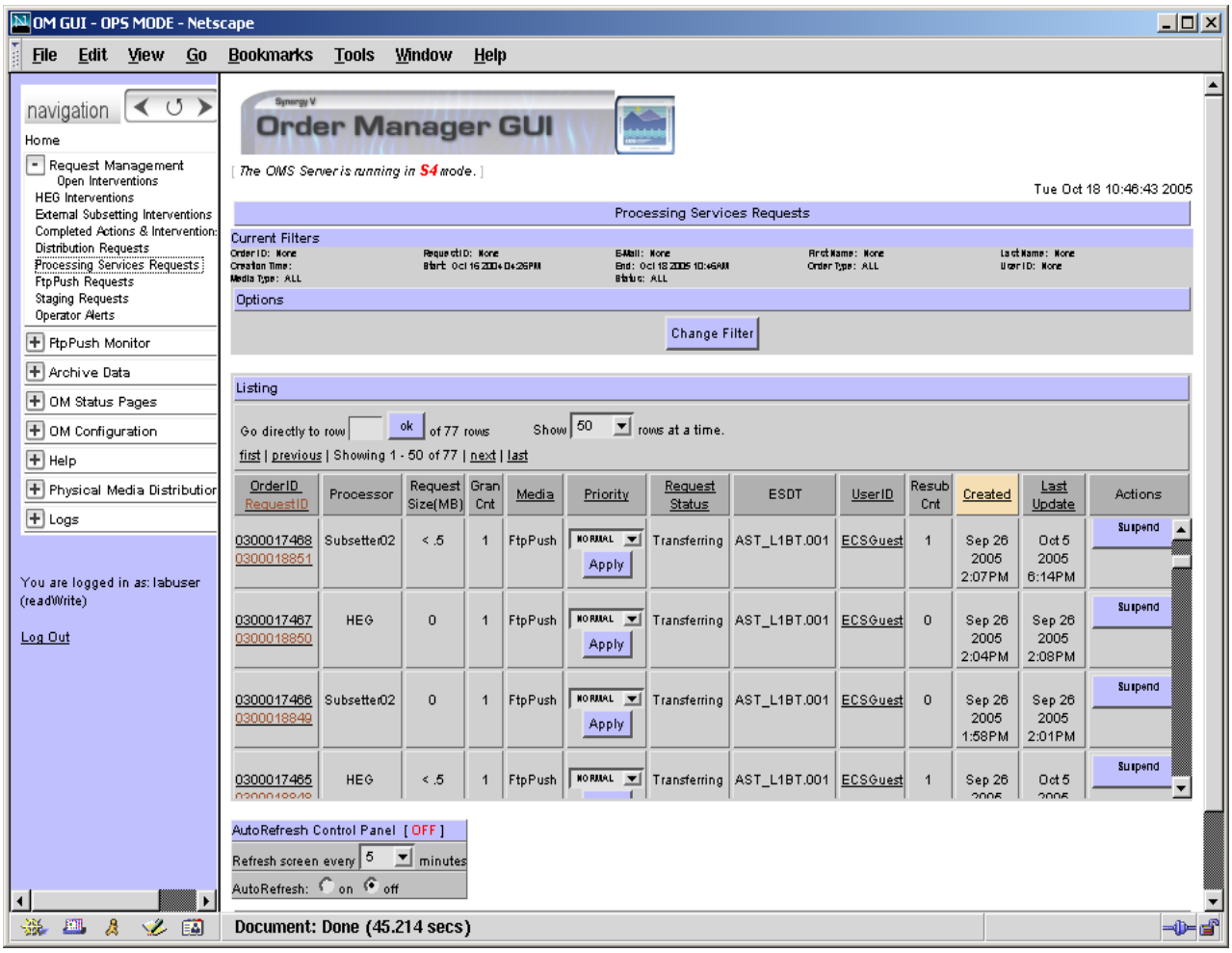


Figure 4.7.7-17b. Processing Services Requests Page

Figure 4.7.7-17b shows OMS GUI allows operator to cancel or suspend the external processing requests while those requests are under OMS control.

Processing Services Requests Filters

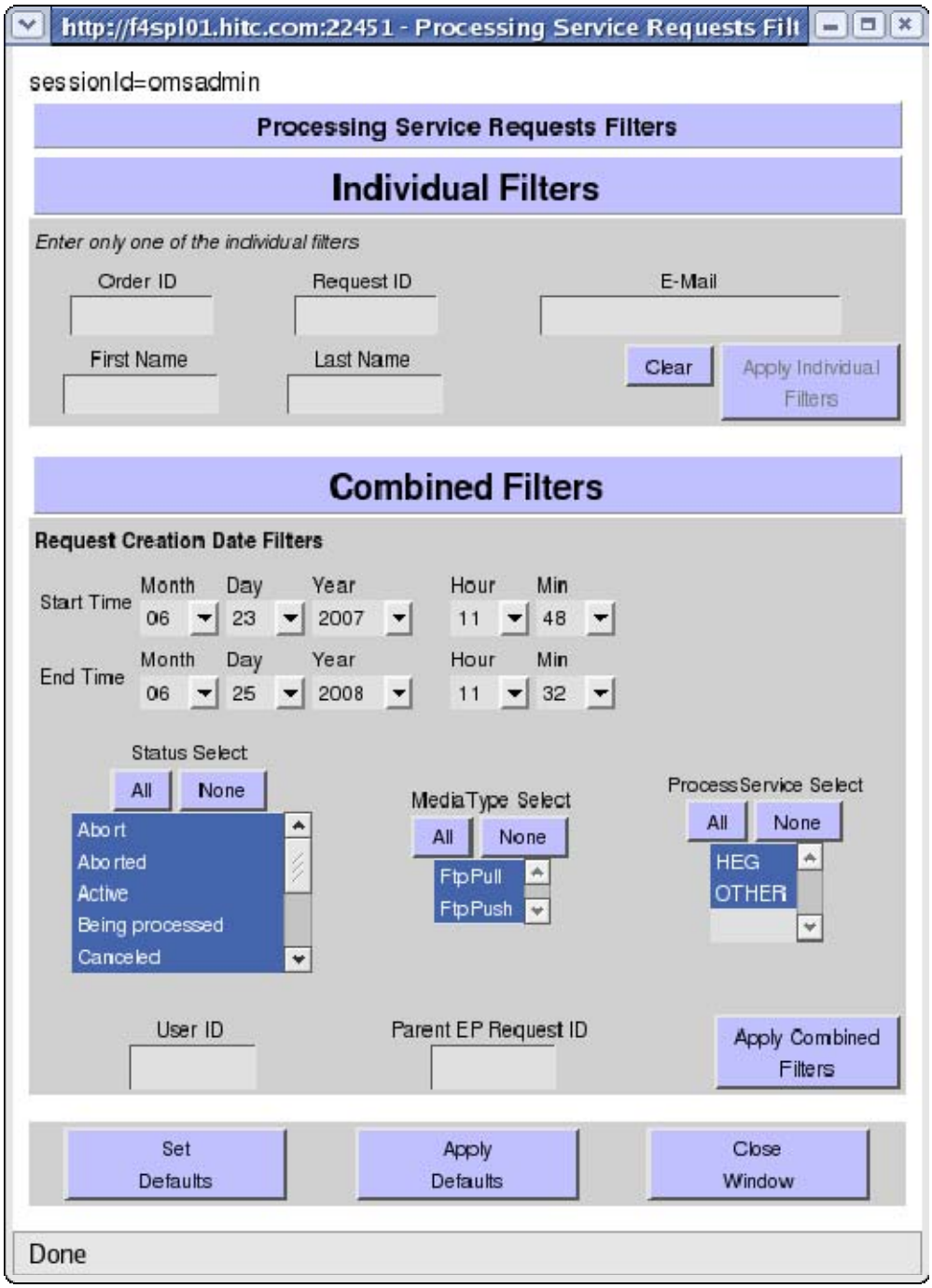


Figure 4.7.7-18. Processing Services Requests Filters Popup

Figure 4.7.7-18 shows the operator can filter any selected external processing service or HEG.

FtpPush/SCP Requests

For each request in the list, values for destination, completion status and resource class are shown. The operator can sort the list by completion status, media and resource class by clicking on the corresponding column headings as shown in Figure 4.7.7-19.

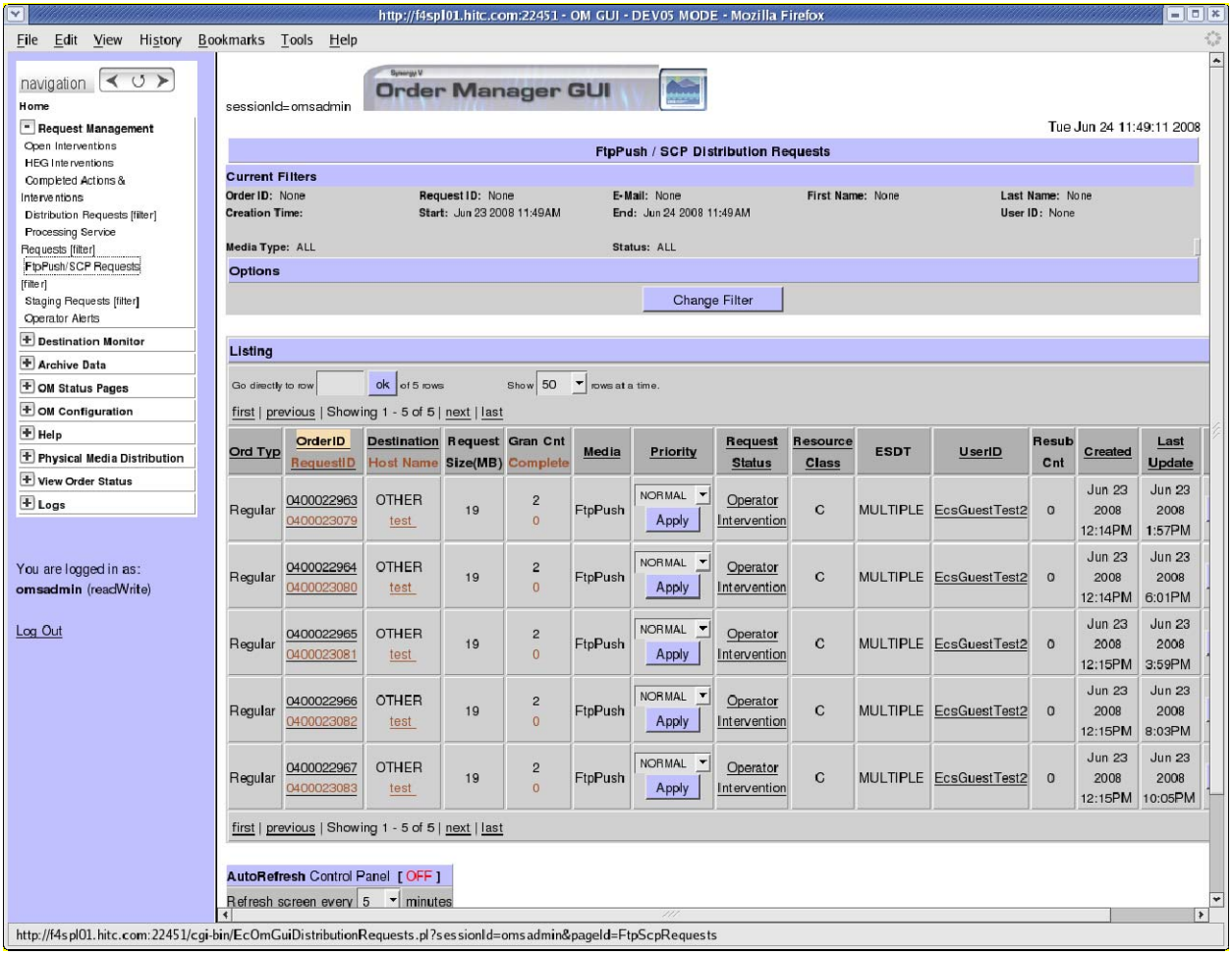


Figure 4.7.7-19. FtpPush/SCP Distribution Requests Page

Staging Requests

The Staging Requests page, shown in Figure 4.7.7-20, displays the completion status and resource class for each request in the list. The operator can sort the list by completion status or resource class by clicking on the corresponding column headings.

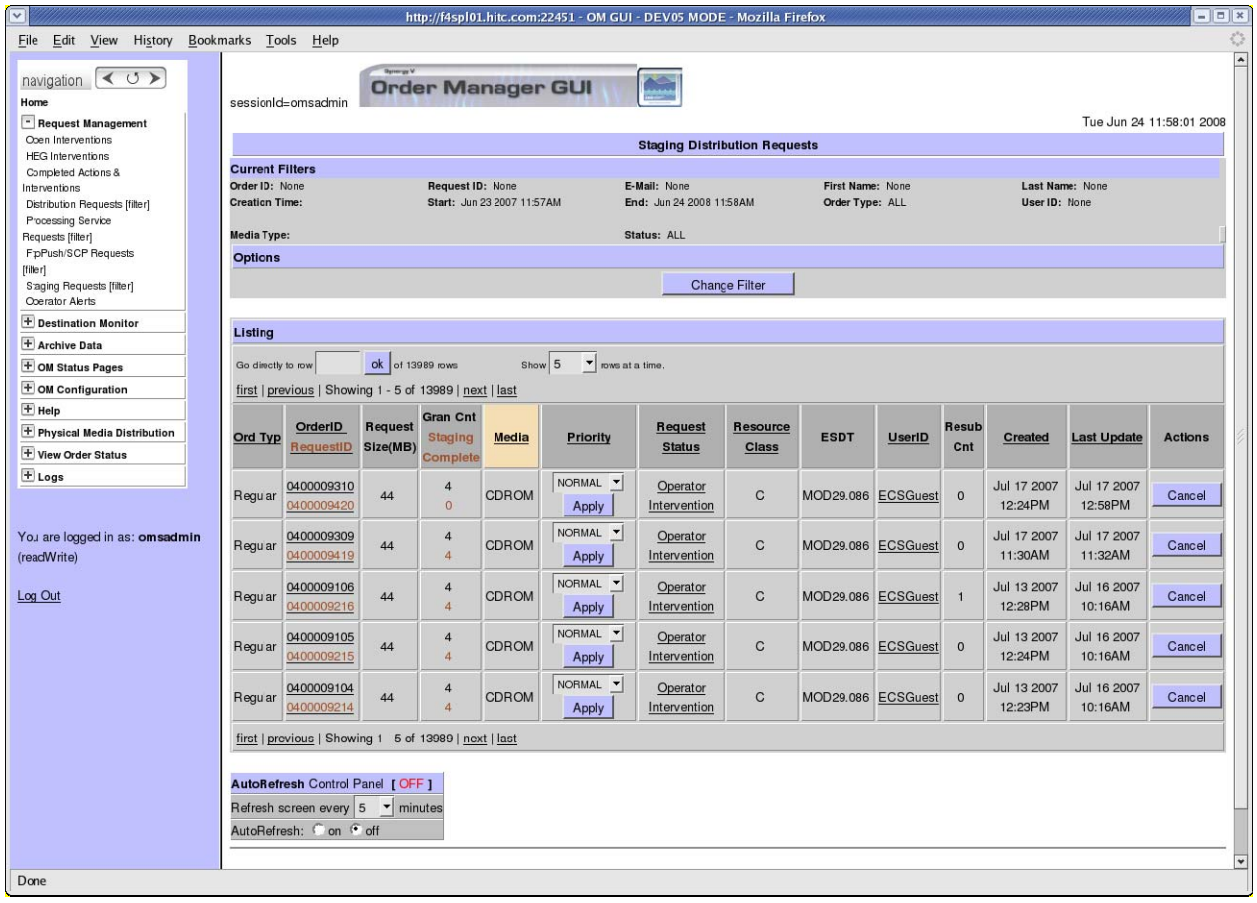


Figure 4.7.7-20. Staging Requests List Page

Historical Requests

The Historical Requests page, shown in Figure 4.7.7-21, does not allow any operator actions. Therefore, the Priority and Actions columns are not displayed.

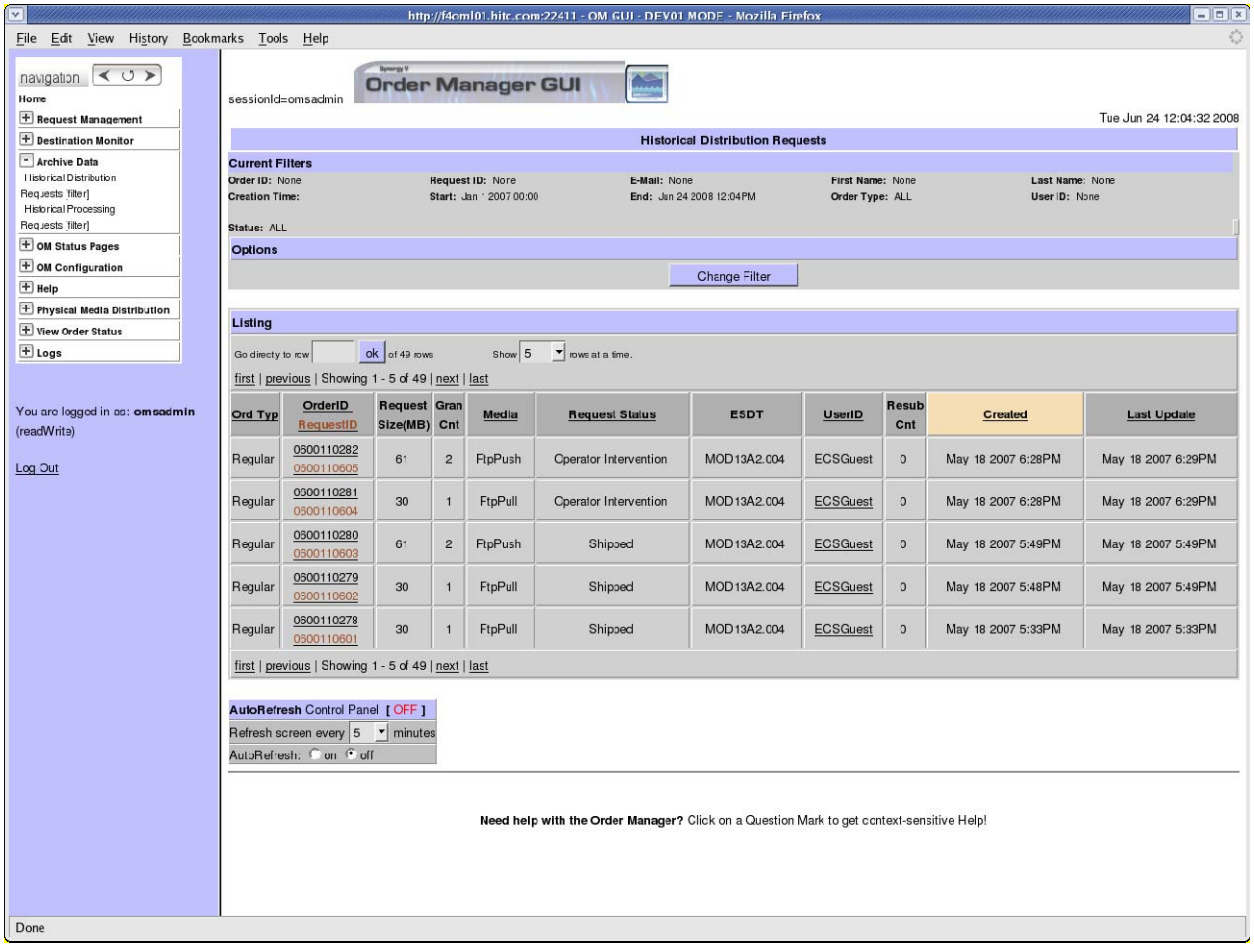


Figure 4.7.7-21. Historical Requests List Page

Historical Processing Requests

The Historical Processing Requests page, shown in Figure 4.7.7-22, allows the operator to search for and display archived external processing requests.

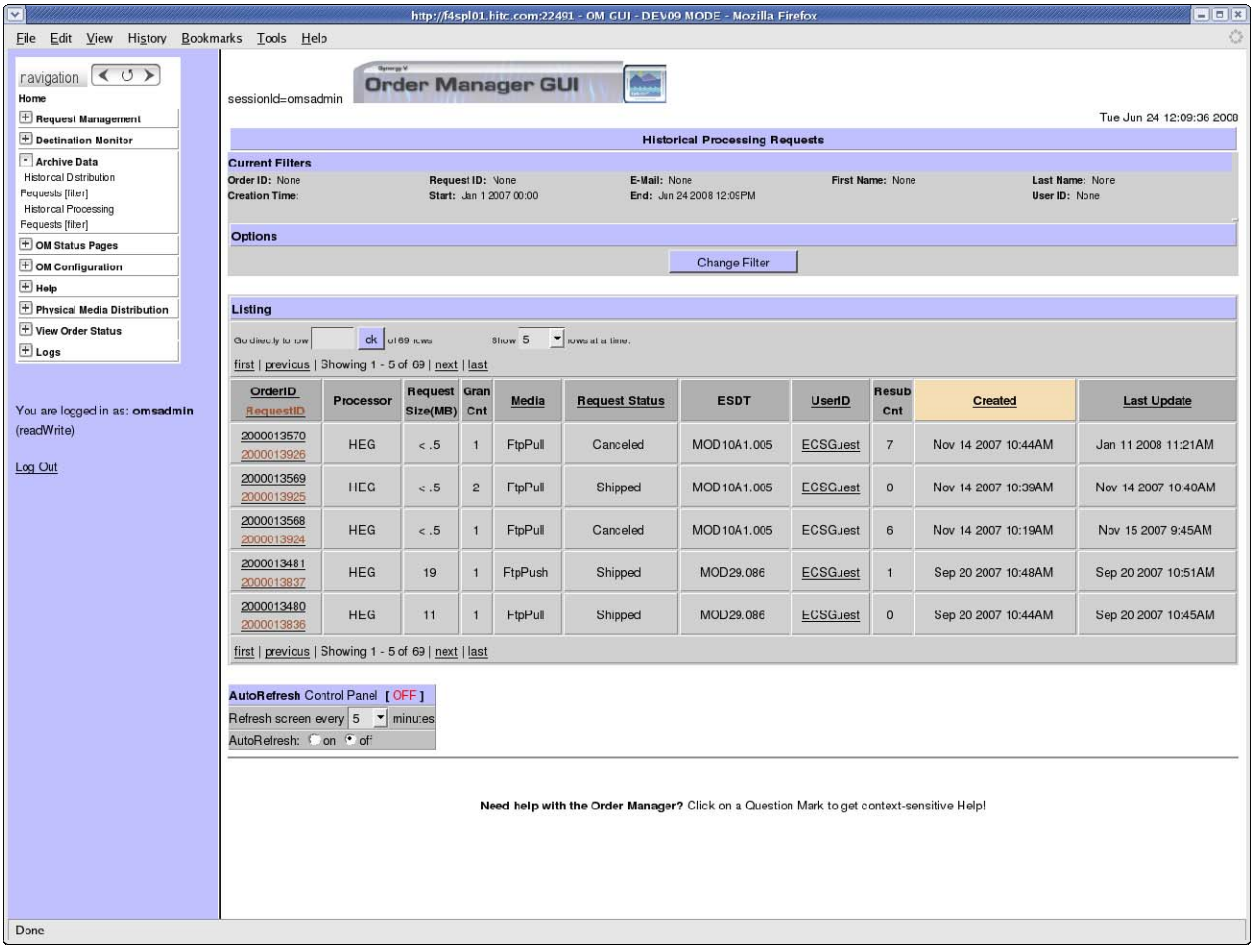


Figure 4.7.7-22. Historical Processing Requests Page

Historical Processing Requests Filter

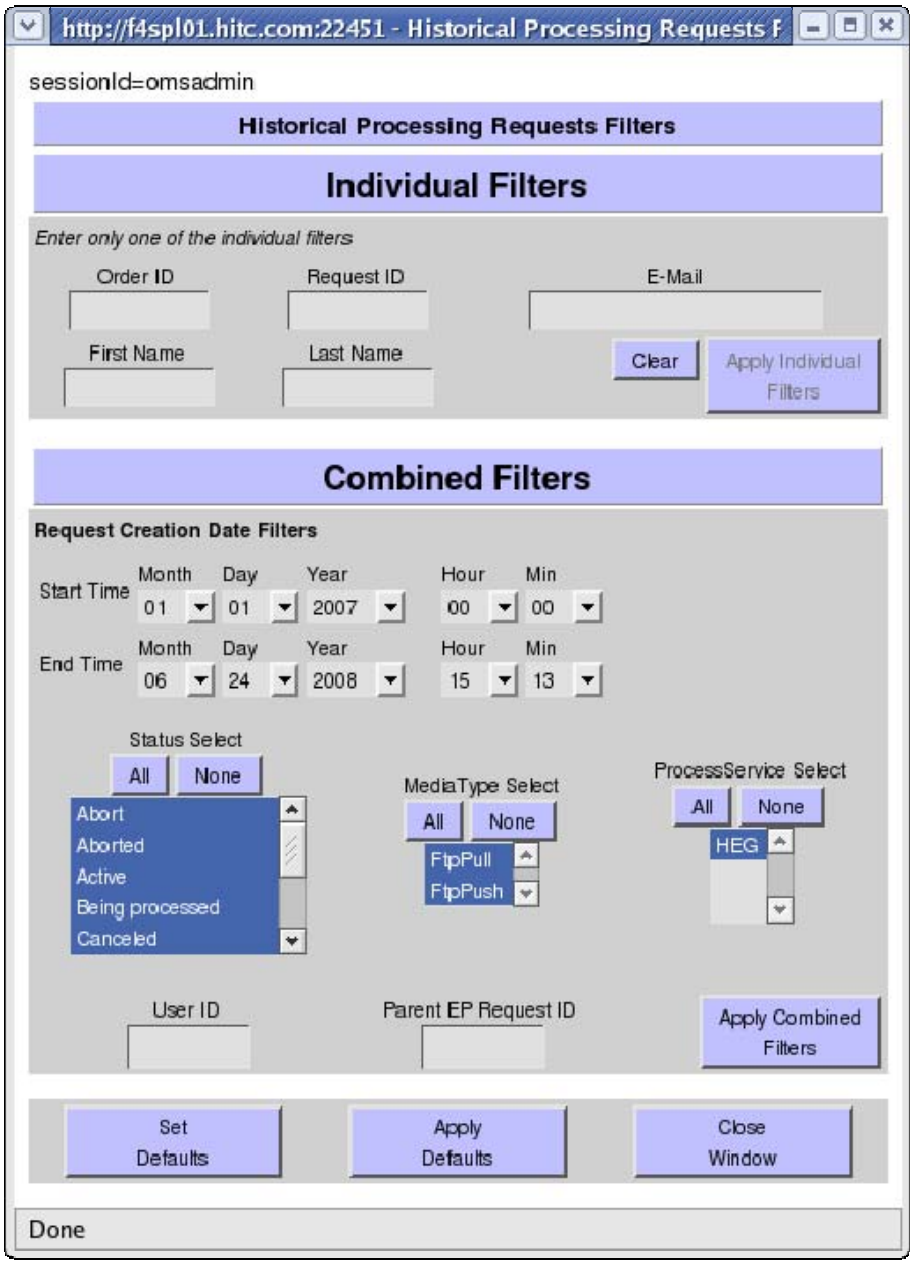


Figure 4.7.7-23. Historical Processing Requests Filter Popup

Figure 4.7.7-23 shows the historical requests filter popup window, which the operator can use to filter the list of historical external processing services and HEG requests, based on various criteria.

4.7.7.2.4.4 Distribution Request Details Page

The operator can click the request ID in any **Distribution Requests, Open Intervention, Order, or Completed Operator Actions and Interventions** page to display the detailed information for a request. Figures 4.7.7-24a and 4.7.7-24b display distribution request details screens for non-physical media requests

For all requests, the operator can perform the following functions:

- Click on the **OrderId** link to view the ECS order page.
- Change the priority of certain requests. **For a complete description of this feature see Section** 4.7.7.2.4.2 **Distribution Requests Lists – Common Features. Note:** Limited Capability operators cannot change the priority of a request.
- For Ftp Push requests, Edit FtpPush Parameters by clicking on the corresponding button. This causes the Edit FtpPush Parameters page to be displayed. Table 4.7.7-13 provides field descriptions for the entry of these values. **Note:** This feature is disabled for Limited Capability *operators*. The operator can also click Destination/Host Name to view the Destination Detail page.
- Perform actions for which the request is eligible. See Section 4.7.7.2.4.2 **Distribution Requests Lists – Common Features** for a description of actions and the types of requests they apply to.
- Scan through the granule list by clicking on navigation links. These links permit jumping to the **first**, **next**, **previous** or **last** block. The number of granules displayed in the table can be changed by selecting a value from the "Show *n* rows at a time" drop-down list. If the Distribution Request information at the top of the page indicates that the request is associated with a bundling order, the Granule List at the bottom reflects the contents of the current bundle.
- Annotate the request.
- Change any mailing, shipping address, or billing address field.

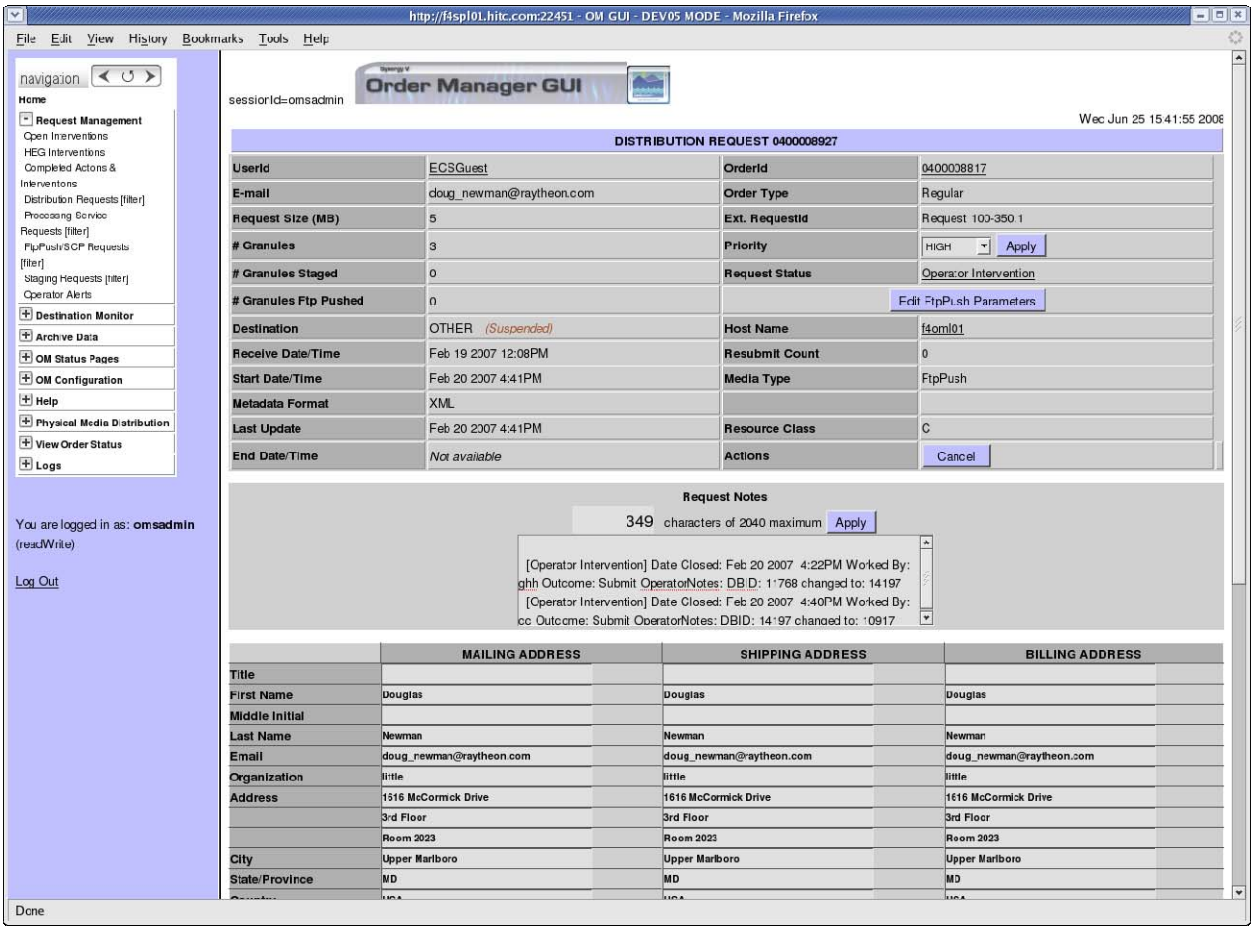


Figure 4.7.7-24a. Distribution Request Details Page for Non-Physical Media Request (1 of 2)

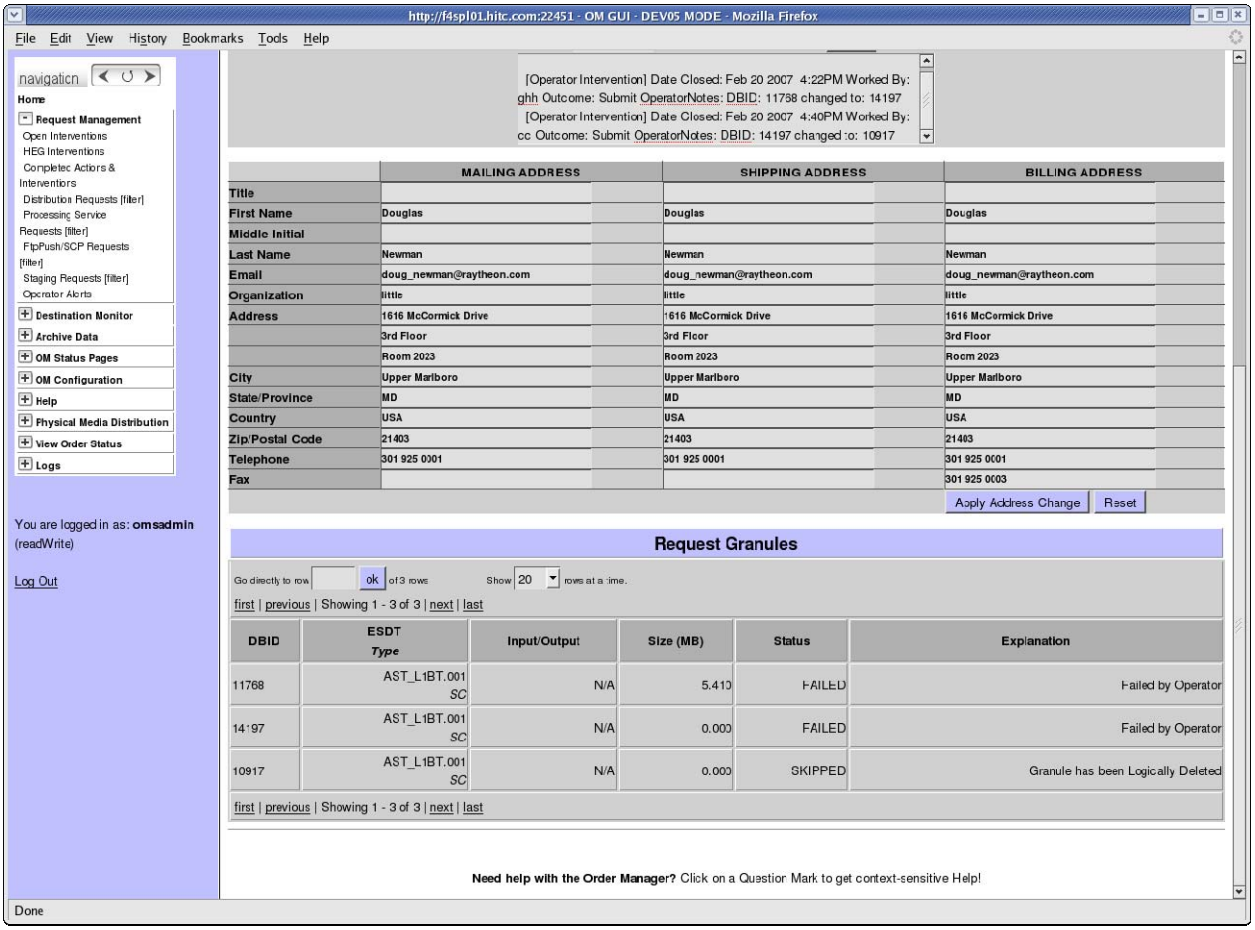


Figure 4.7.7-24b. Distribution Request Details Page for Non-Physical Media Request (2 of 2)

4.7.7.2.4.5 Edit FtpPush Parameters Page

Note: This page is not accessible to Limited Capability operators.

The Edit FtpPush Parameters Page displays a list of FTP Push parameters which can be edited by the operator. The operator can enter or change the value of any of the parameters displayed. The operator then clicks on one of the buttons at the bottom of the page. Button actions are:

- Change This Request – changes the FtpPush Parameters for the request listed and returns to the Request Detail Page.
- Change All Active Requests - changes the FtpPush Parameters for all requests for the destination listed that are not in a terminal state and returns to the Request Detail Page.
- Cancel – cancels all changes to FtpPush Parameters and returns to the Request Detail FtpPush Page.

Figure 4.7.7-25 shows the Edit FtpPush Parameters Page. Table 4.7.7-13 provides the descriptions of the fields on the Edit FtpPush Parameters Page.

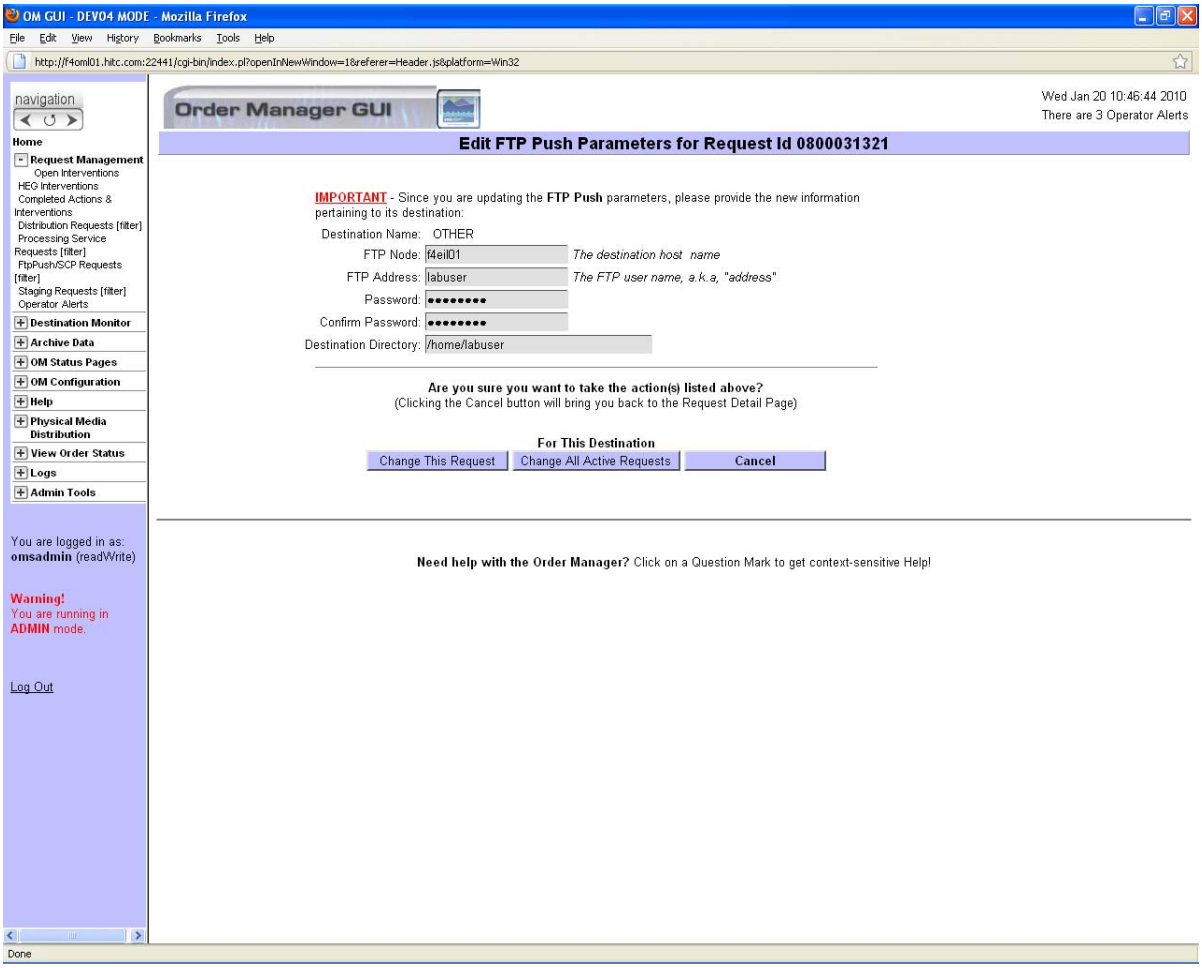


Figure 4.7.7-25. Edit FtpPush Parameters Page

Table 4.7.7-13. Field Descriptions for Edit FtpPush Parameters Page

Field Name	Data Type	Size	Entry	Description
FTP Node	Varchar	20	Required	The Unix hostname of the FTP recipient
FTP Address	Varchar	14	Required	The Unix login ID of the FTP recipient
Password	Varchar	15	Required	The Unix password for the FTP recipient
Confirm Password	Varchar	15	Required	The Unix password verification for the FTP recipient
User String	Varchar	255	Optional	String to be inserted into the FTP parameters
Destination Directory	Varchar	255	Required	The pathname of the Unix directory where the acquired files are to be stored

4.7.7.2.4.5 Edit SCP Parameters Page

Note: This page is not accessible to Limited Capability operators.

- The Edit SCP Parameters Page displays a list of SCP parameters which can be edited by the operator. The operator can enter or change the value of any of the parameters displayed.

Figure 4.7.7-26 shows the Edit SCP Parameters Page. Table 4.7.7-14 provides the descriptions of the fields on the Edit SCP Parameters Page.

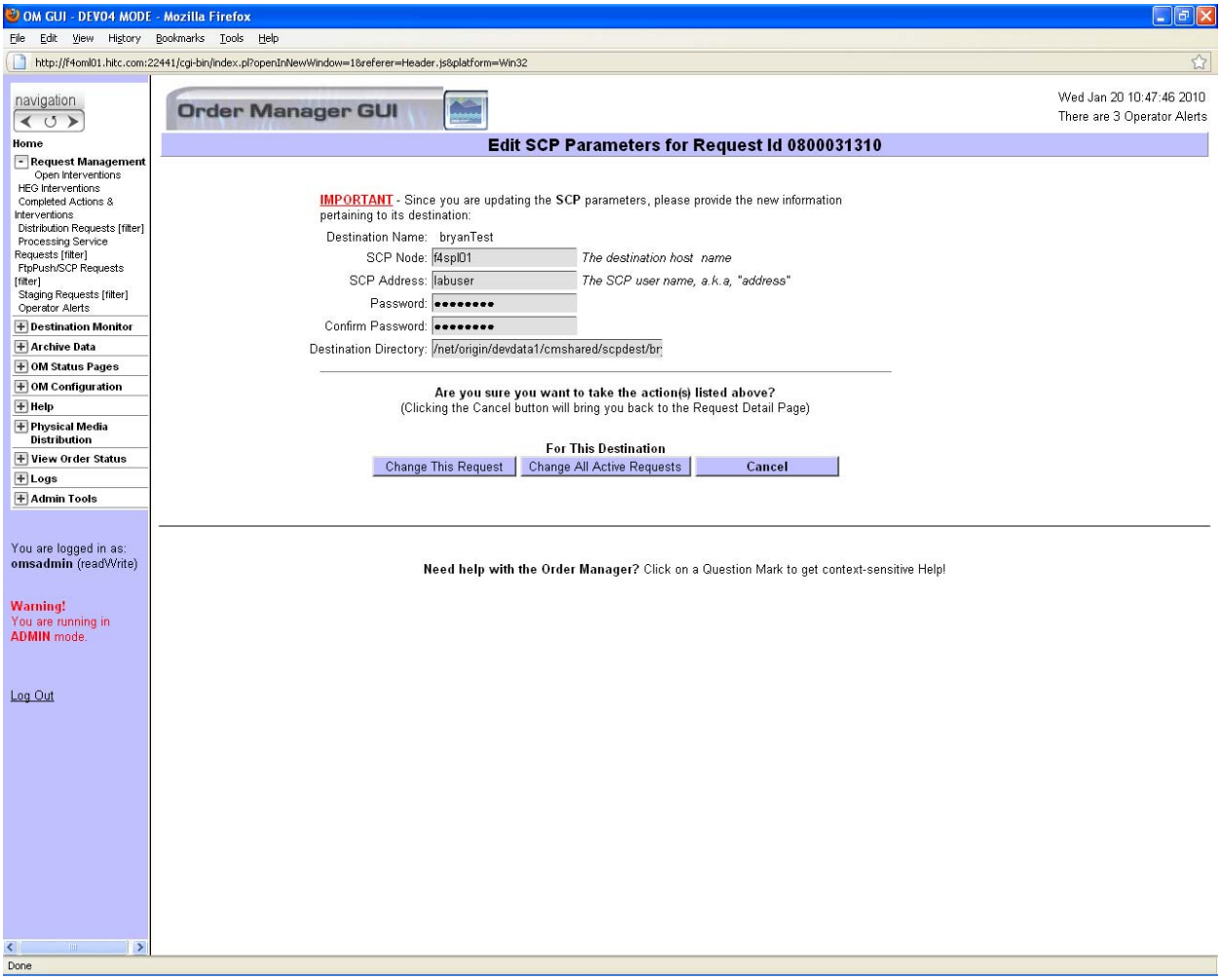


Figure 4.7.7-26. Edit SCP Parameters Page

Table 4.7.7-14. Field Descriptions for Edit SCP Parameters Page

Field Name	Data Type	Size	Entry	Description
Host Address	Varchar	20	Required	The Unix hostname of the SCP recipient.
SCP user	Varchar	14	Required	The Unix login ID of the SCP recipient.
Password	Varchar	15	Required	The Unix password for the SCP recipient.
Confirm Password	Varchar	15	Required	The Unix password verification for the SCP recipient.
Destination Directory	Varchar	255	Required	The pathname of the Unix directory where the acquired files are to be stored.

4.7.7.2.4.6 ECS Order Page

Note: Limited Capability operators are limited to viewing the details of an ECS Order. They cannot change the priority of or take actions for Requests.

The operator can click on the **Order ID** link in the Distribution Requests list page or the Distribution Request details page to open the **ECS Order** detailed information page, as illustrated in Figure 4.7.7-27. If the order is a bundling order, the operator can click the **Spatial Subscription Server** link to go to the Spatial Subscription Server Web page to view and update the Bundling Order as illustrated in Figure 4.7.7-28. The operator can click a **Request ID** to go to **Distribution Request** details page for that request (see Figure 4.7.7-29).

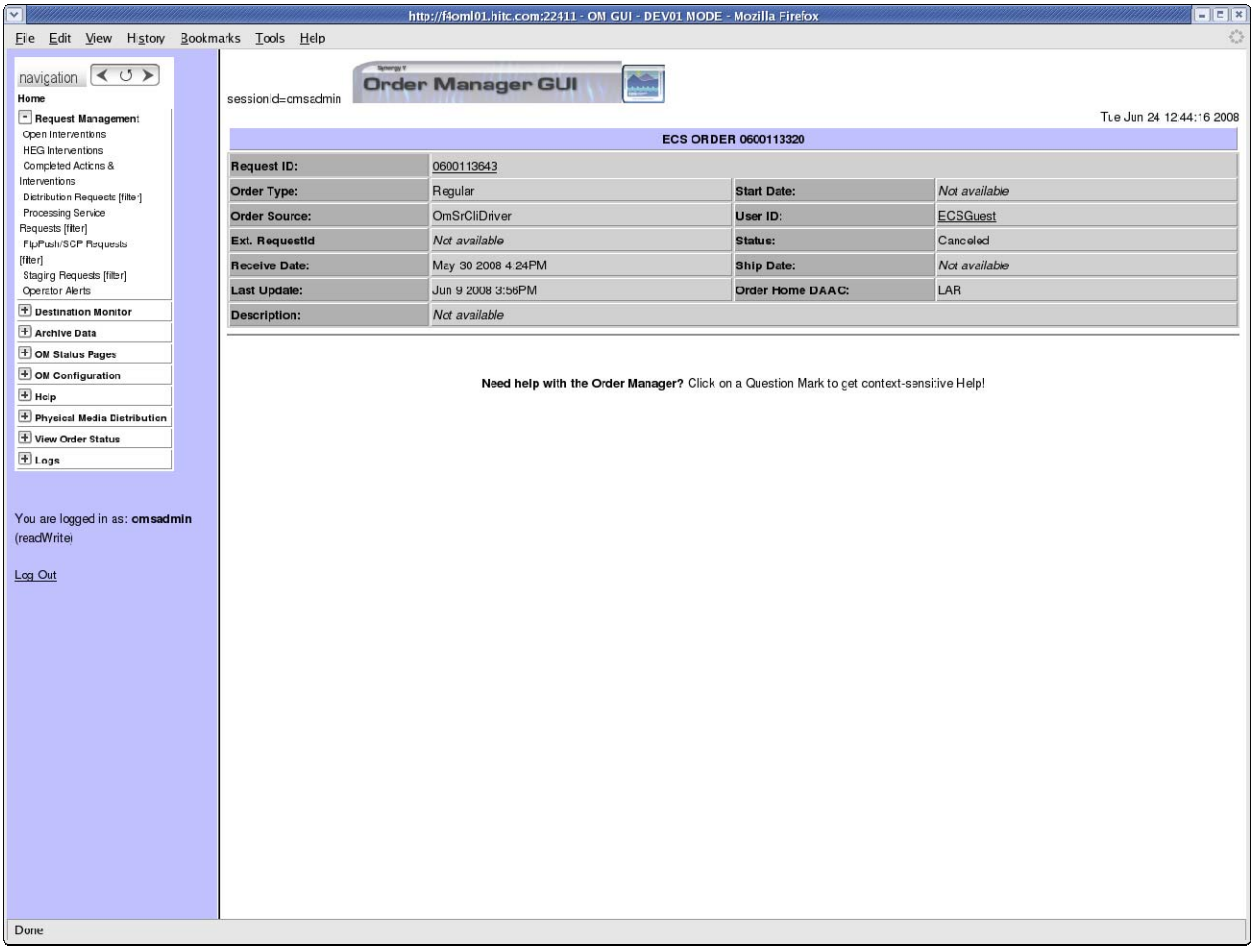


Figure 4.7.7-27. ECS Order Information Page

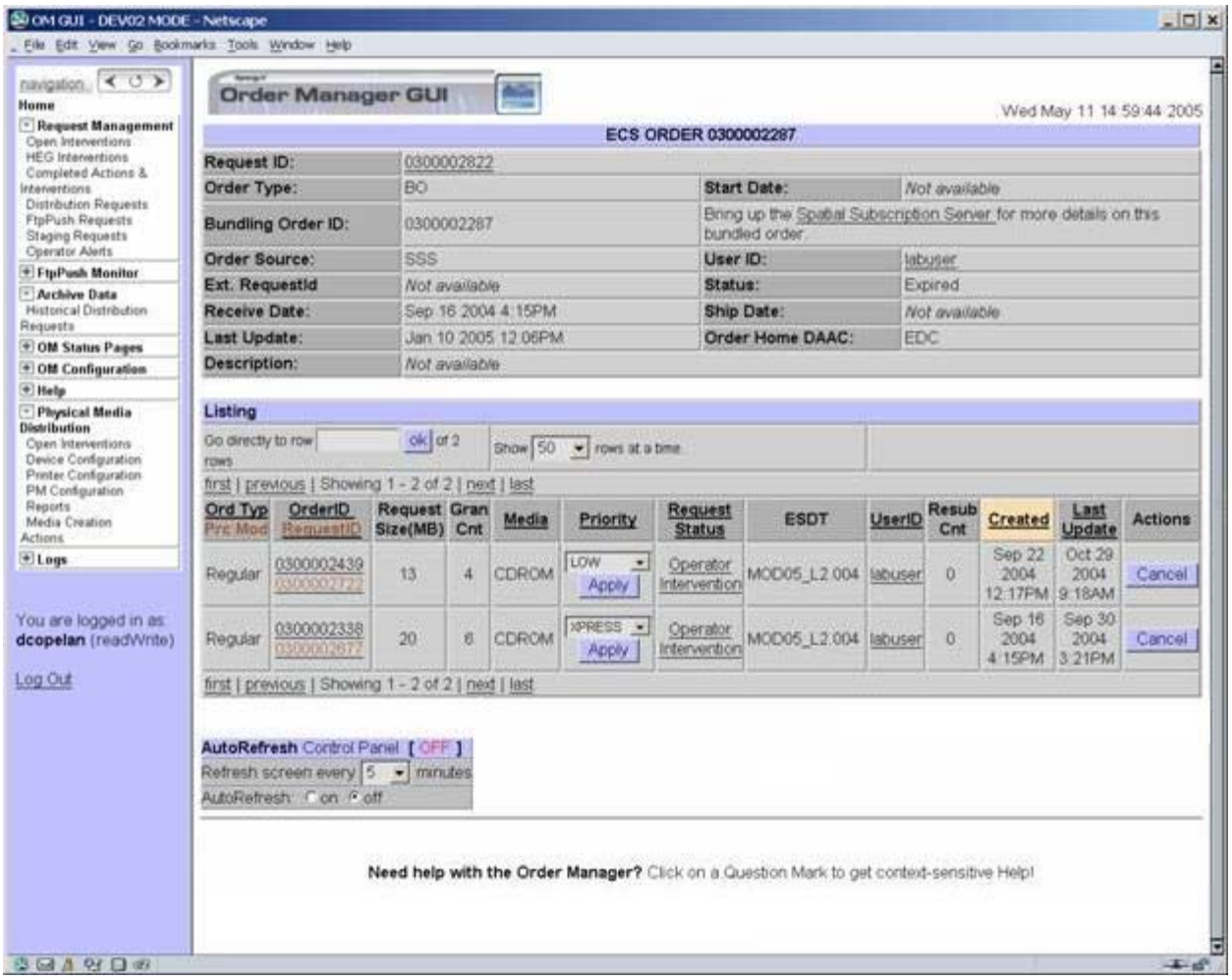


Figure 4.7.7-28. ECS Order Information Page for Bundling Order

4.7.7.3 Destination Monitor

4.7.7.3.1 Suspended Distribution Destinations Page

Note: Limited Capability operators cannot take any actions on this page.

The Distribution Destinations Displays a list of suspended FTP Push / SCP Destinations as shown in Figure 4.7.7-29. The operator can see details for a destination by clicking the name of the destination (for a configured destination) or the hostname (for a non-configured destination) to be viewed. This displays the Ftp Push / SCP Monitor – Destination Details page described in Section 4.7.7.3.2 FTP Push / SCP Destinations Detail Page.

The operator can resume dispatching to a destination by clicking its **Resume** button.

The Active Destinations section allows the operator to enter either a Destination Name or Host Name and either suspend / resume the destination or see the Destination Monitor by clicking the Destination Name – Destination Details page for the destination.

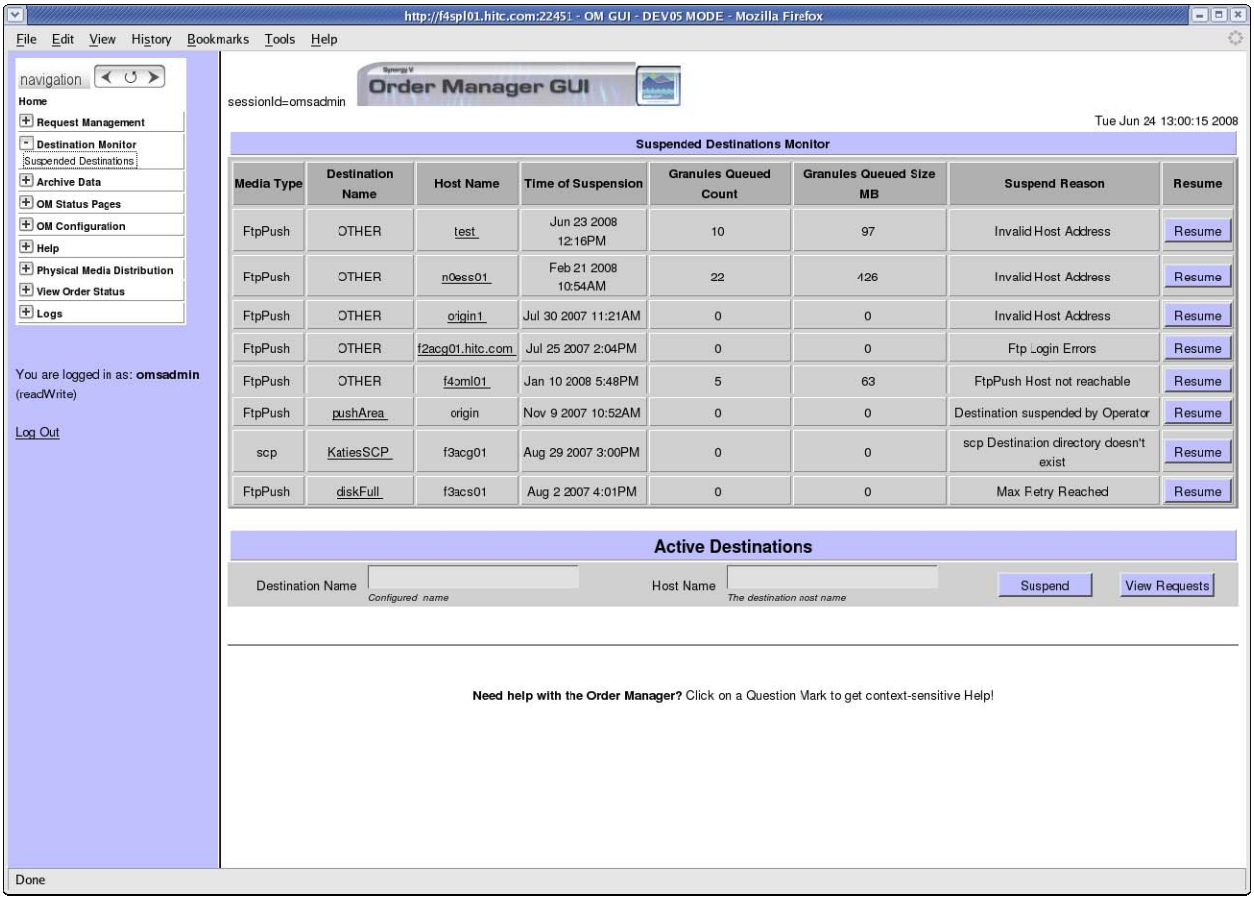


Figure 4.7.7-29. Suspended Destinations Monitor Page

Note: *Limited Capability operators cannot take any actions on this page.*

The FTP Push / SCP Distribution Destination Detail Page displays a **Suspend**/**Resume** button, a list of FTP Push / SCP Operations that Caused the Suspension of the destination and a list of FTP Push Requests That Are Not In A Terminal State for the destination.

The Suspend/Resume button is labeled **Resume** if the destination is suspended and **Suspend** if the destination is active. Clicking this button will suspend or resume the destination.

For a description of the list of FTP Push Requests, see Section 4.7.7.2.4.1. View Distribution Requests.

4.7.7.3.2 FTP Push/ SCP Distribution Destinations Detail Page

Note: Limited Capability operators cannot take any actions on this page.

The FTP Push / SCP Distribution Destinations Detail Page displays a list of FTP Push / SCP Operations that Caused the Suspension of the destination selected and a list of FTP Push Requests That Are Not In A Terminal State for the destination selected. This is displayed in Figures 4.7.7-30 and 4.7.7-31.

For a description of the list of FTP Push / SCP Requests see Section 4.7.7.2.4.2.

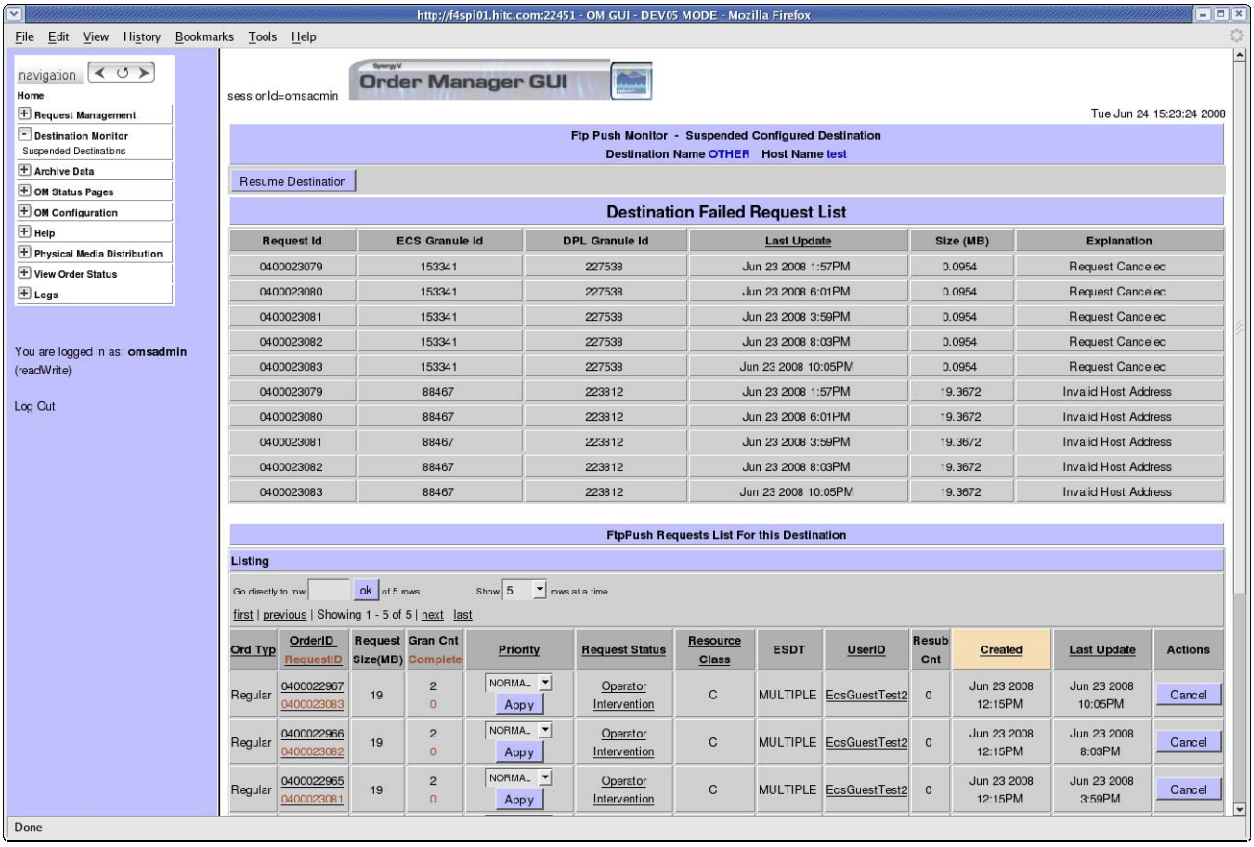


Figure 4.7.7-30. FTP Push Distribution Destinations Detail Page

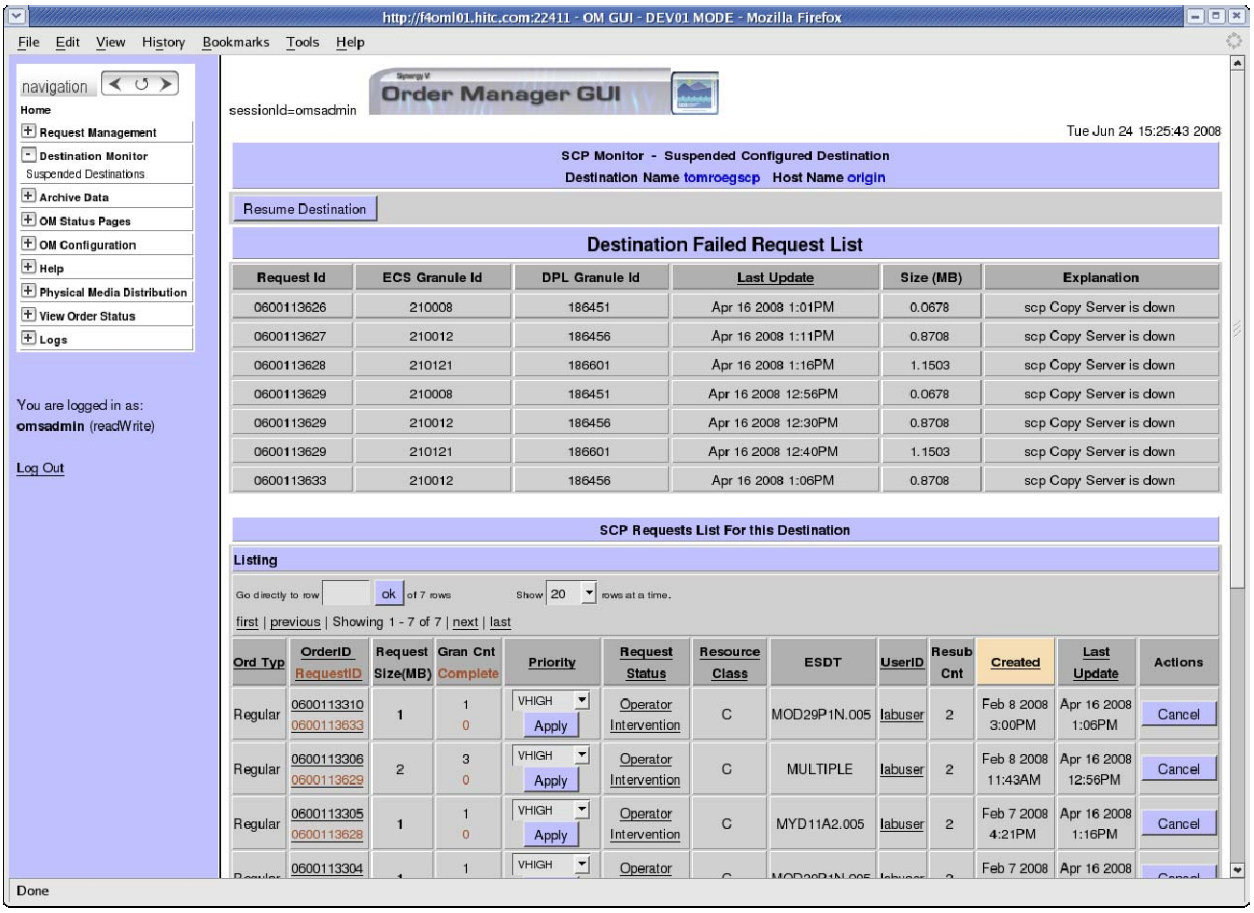


Figure 4.7.7-31. SCP Distribution Destinations Detail Page

4.7.7.4.1 Staging Status Pages

The Staging Status pages show a summary of the volume and number of granules that are currently in Staging. The Staging information is broken down into four categories:

- Granules waiting for Staging
- Granules in Staging
- Granules that have been Staged but not yet shipped
- Granules that have been staged and shipped

The Staging Status information is categorized by media type – one page for FtpPush, and another for all other media types (physical media and FtpPull).

Staging Status by Media Type

Click on "Media Type" under the **Staging Status Pages** subsection of **OM Status Pages**. This will show a detailed summary of number and volume of granules in their various Staging states, as shown in Figure 4.7.7-36. Next to each media type is also the target low and high Watermarks, see Table 4.7.7-16 for more details on Watermarks. These Watermarks are configurable by full-capability operators in the Media Configuration page.

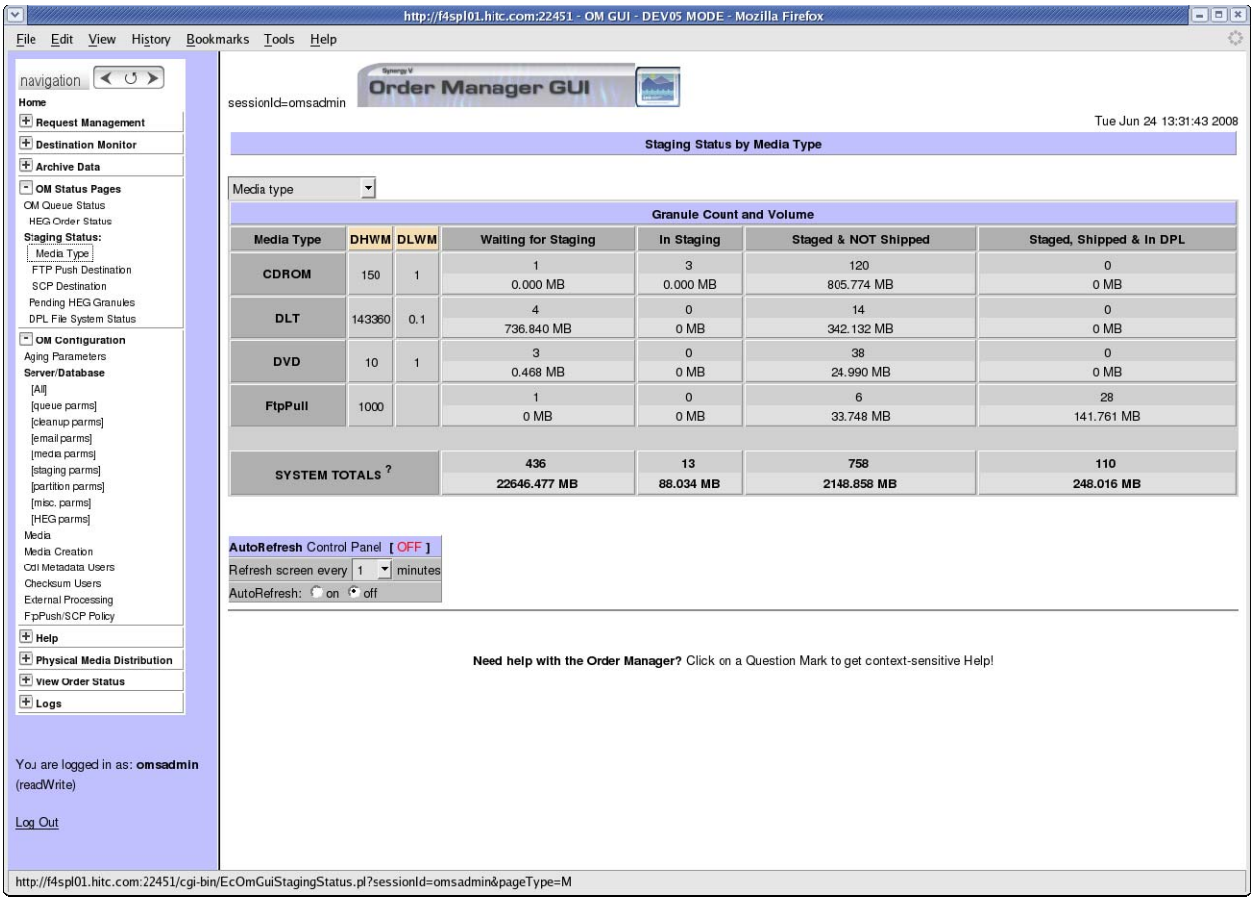


Figure 4.7.7-36. Staging Status by Media Type

Staging Status by FTP Push Destination

Click on "Ftp Push Destination" under Staging Status Pages in the OM Status Pages menu. This page will display a list of the currently configured FTP Push destination names, along with the IP address and destination directory (see Figure 4.7.7-37). Each of these destinations has individual DHWM and DLWM settings, as well their own Staging Status numbers. This screen shows the number and volume (in MB) of granules that are:

- Waiting for Staging
- In Staging
- Staged & NOT Shipped
- Staged, Shipped & In DPL

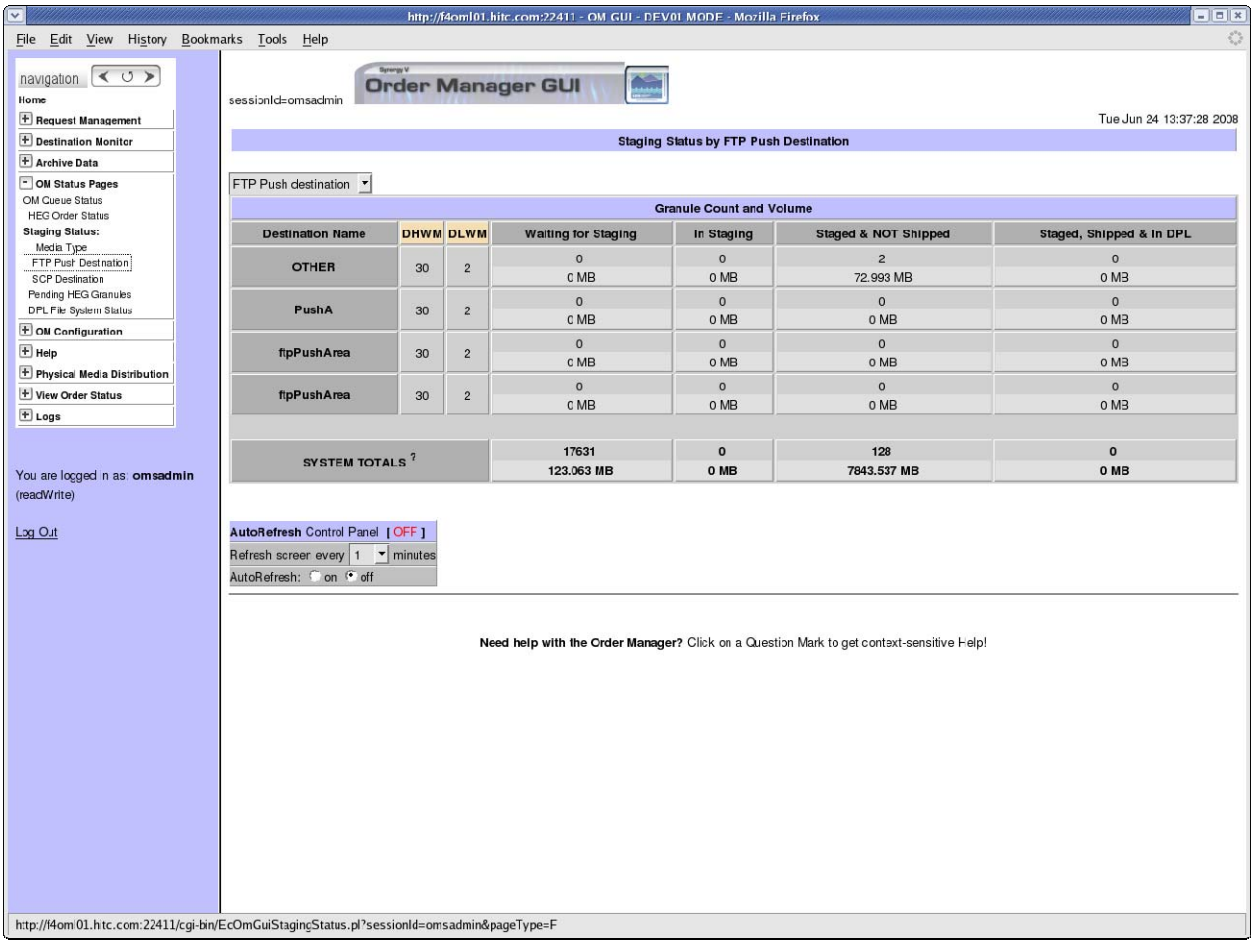


Figure 4.7.7-37. Ftp Push Destination Listing For Staging Status

Staging Status by SCP Destination

Click on "SCP Destination" under Staging Status Pages in the OM Status Pages menu. This page will display a list of the currently configured SCP destination names. Each of these destinations has individual DHWM and DLWM settings, as well their own Staging Status numbers. This screen, as shown in Figure 4.7.7-38, shows the number and volume (in MB) of granules that are:

- Waiting for Staging
- In Staging

- Staged & NOT Shipped
- Staged, Shipped & In DPL

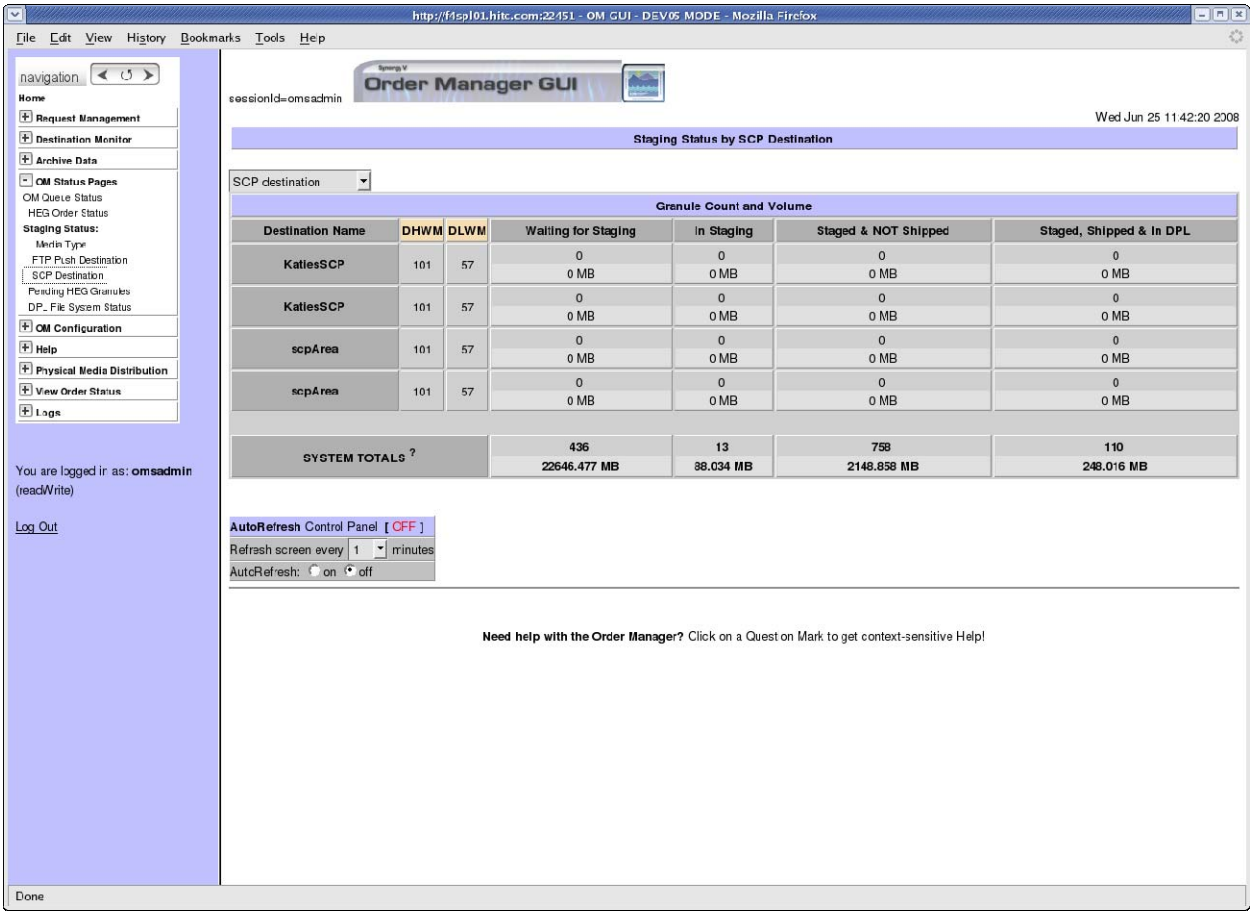


Figure 4.7.7-38. SCP Destination Listing For Staging Status

\

4.7.7.6 OM Configuration Pages

Note: For all types of configuration pages, Limited Capability operators can only view configuration parameters. The ability to update parameters will be disabled.

Aging Parameters

To access this page, click "Aging Parameters" under the **OM Configuration** menu. This page displays parameters that affect how Distribution Requests are aged over time (see

Figure 4.7.7-40). The aging parameters are configurable for each ECS Priority Level (XPRESS, VHIGH, HIGH, NORMAL, and LOW). Below is a description of each parameter.

Age Step: The aging rate by which the effective priority of a request increases for every hour it has been waiting. The range is 0-100, including decimal fractions. If this parameter is set to 0, waiting requests will never increase in priority.

For example, if the Age Step is set to 5.5 and a request with an initial priority of 100 waits 10 hours to be pushed, then the request will increase in priority by a factor of 5.5 every hour until it has been delivered:

Hour 0: priority = 100

Hour 1: priority = 105.5

Hour 2: priority = 111

.

.

Hour 10: priority = 155

Maximum Priority The maximum priority a request can attain through this aging process. For example, if Maximum Priority were set to 130, then in the example above, once the request had reached a priority of 130, it would not go any higher (i.e., at Hour 10 it would still be 130).

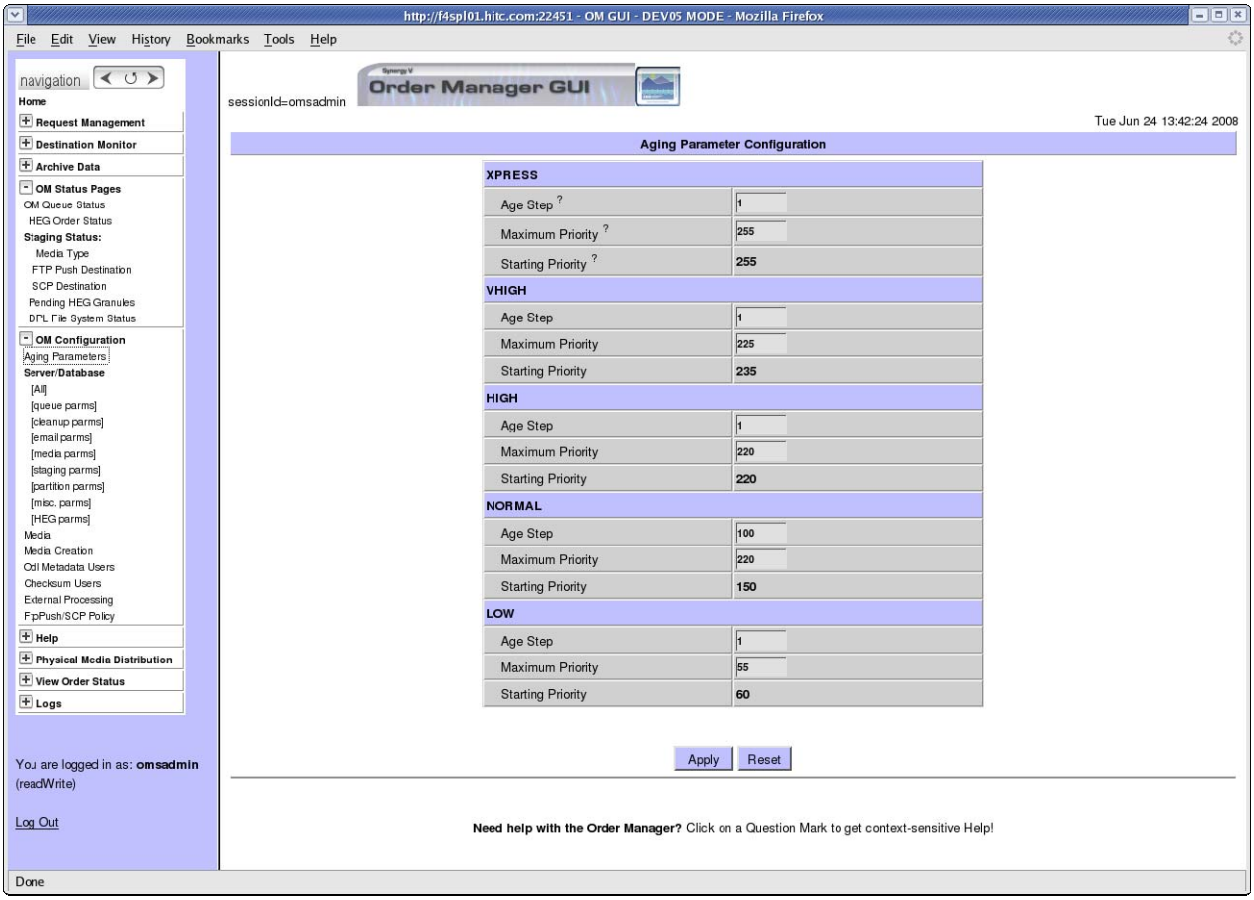


Figure 4.7.7-40. Aging Parameter Configuration

Server/Database Configuration

These are values that affect how the OM Server and Database run (see Figure 4.7.7-41 and Figure 4.7.7-42). The page displays the current value of the configuration parameters and provides a text input box to change them. To the far right is a description of each parameter.

These parameters are dynamically loaded into the page, meaning that the parameters displayed are those that the operator can modify. If a configuration parameter is added in the Database, it will also be displayed on the screen. See Table 4.7.7-18 for a description of these parameters.

Drop-Down Lists

Some parameters are not editable text fields, but drop-down lists containing the possible values for that field. This is to protect the OMS Server from acting in an undesirable way as a result of using an unexpected value. For example Global Staging Status is one such field – it *must* be "S" or "A" for the OMS Server to function properly.

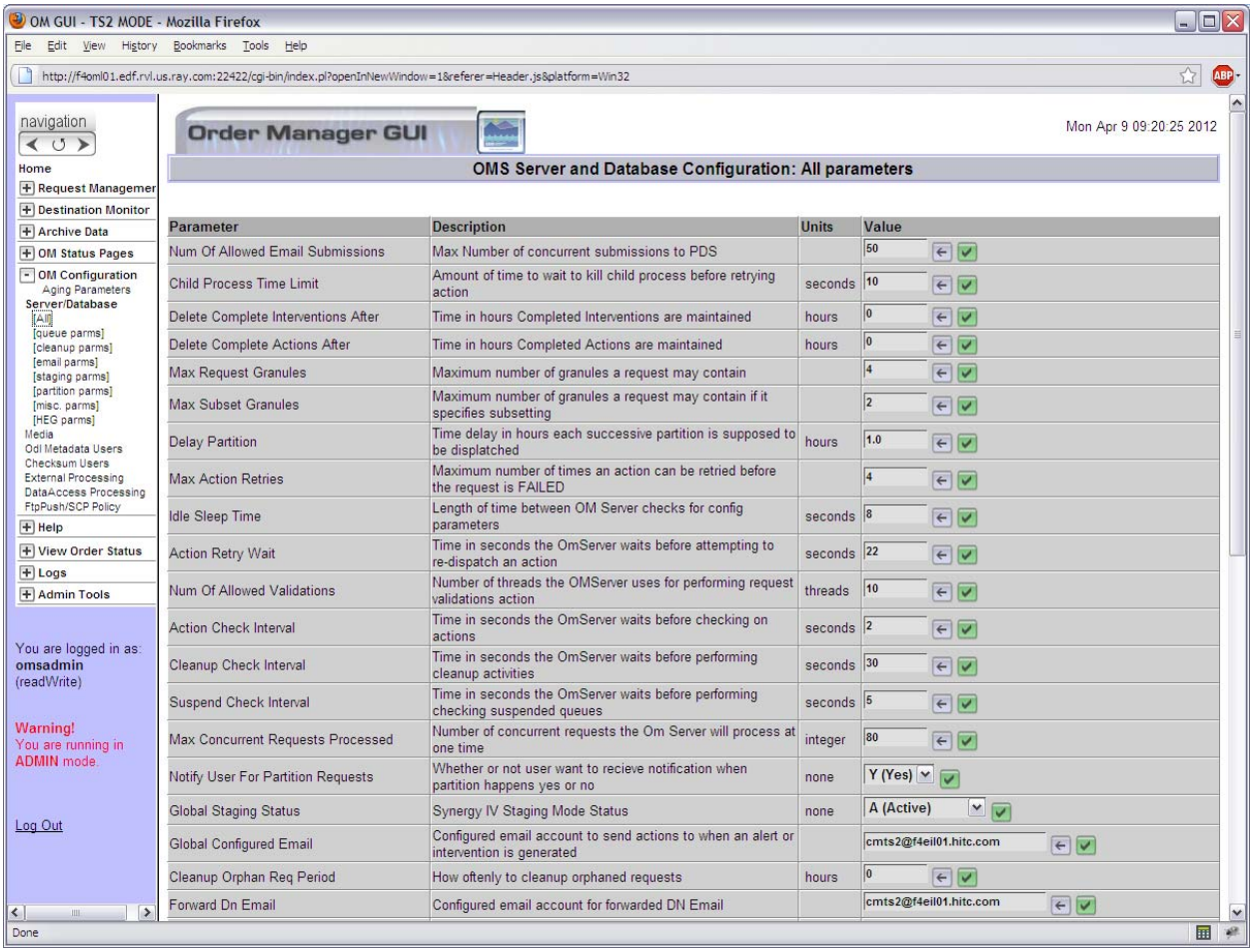


Figure 4.7.7-41. Server/Database Configuration —Part 1

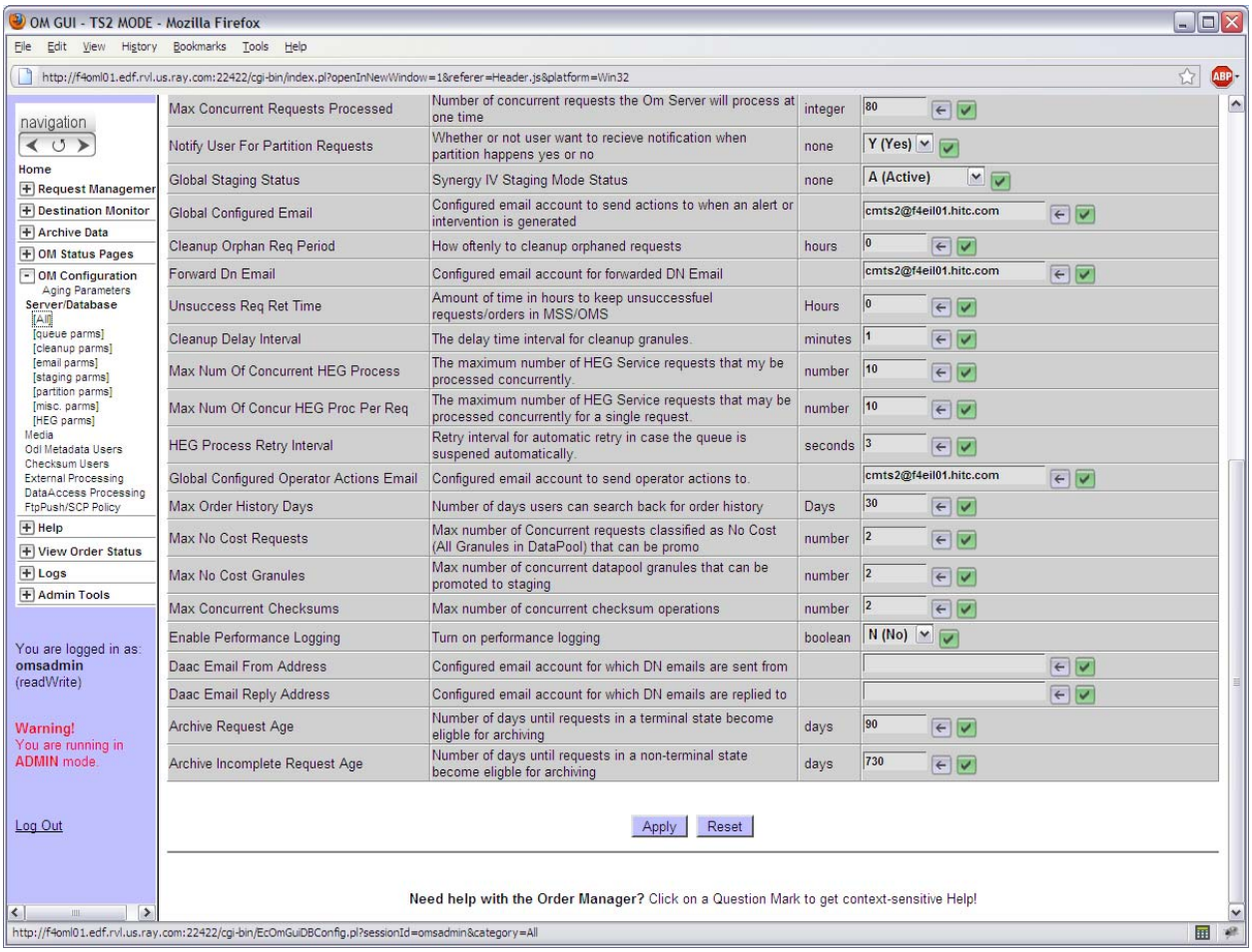


Figure 4.7.7-42. Server/Database Configuration – Part 2

Media Configuration

To access this page, click on "Media" under the **OM Configuration** menu. These configuration parameters are specific to each media type, and are dynamically loaded just as the Server/Database Configuration parameters. The page displays the current value of the parameter and provides a text box input to change it. Figure 4.7.7-43 shows an example of some of the Media Configuration Parameters. See Table 4.7.7-18 for a description of these parameters.

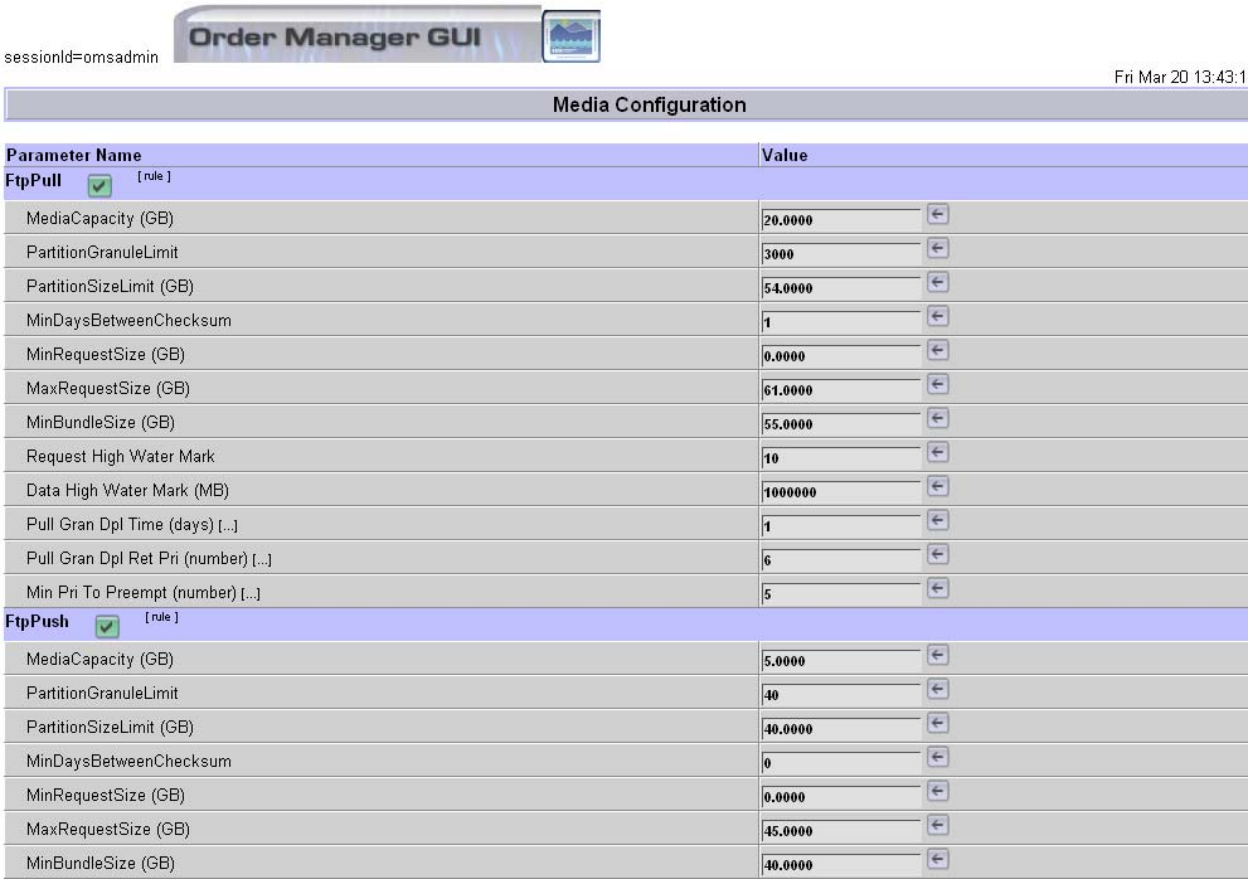


Figure 4.7.7-43. Media Configuration Page

Table 4.7.7-18. OM GUI Configuration Parameters Descriptions (1 of 2)

Field Name	Data Type	Description
Num Of Allowed Email Submissions		Max Number of concurrent submissions to Email.
Child Process Time Limit	seconds	Amount of time to wait to kill child process before retrying action.
Delete Complete Interventions After	hours	Time in hours Completed Interventions are maintained.
Delete Complete Actions After	hours	Time in hours Completed Actions are maintained.
Max Request Granules		Maximum number of granules a request may contain.
Max Subset Granules		Maximum number of granules a request may contain if it specifies subsetting.
Delay Partition	hours	Time delay in hours each successive partition is supposed to be dispatched.
Max Action Retries		Maximum number of times an action can be retried before the request is FAILED.
Idle Sleep Time	seconds	Length of time between OM Server checks for config parameters.
Action Retry Wait	seconds	Time in seconds the OmServer waits before attempting to re-dispatch an action.
Num Of Allowed Validations	threads	Number of threads the OMServer uses for performing request validations action.
Action Check Interval	seconds	Time in seconds the OmServer waits before checking on actions.
Cleanup Check Interval	seconds	Time in seconds the OmServer waits before performing cleanup activities.
Suspend Check Interval	seconds	Time in seconds the OmServer waits before performing checking suspended queues.
Max Concurrent Requests Processed		Number of concurrent requests the OmServer will process at one time.
Notify User For Partition Requests		Whether or not user want to receive notification when partition happens yes or no.
Global Staging Status		Staging Mode Status.
MinDaysBetweeenChecksum		Checksum files that haven't been checksummed for this many days.
Max Failure Archive		Allowable number of failures prior to suspending Archive.
Global Configured Email		Configured email account to send actions to when an alert or intervention is generated.
Max Orphan Req Age	hours	How long to keep an orphaned request in system before it is qualified for removal.
Cleanup Orphan Req Period	hours	How often to cleanup orphaned requests.
Forward Dn Email		Configured email account for forwarded DN Email.

Table 4.7.7-18. OM GUI Configuration Parameters Descriptions (2 of 2)

Field Name	Data Type	Description
Unsuccess Req Ret Time	hours	Amount of time in hours to keep unsuccessful requests/orders in OMS.
Cleanup Delay Interval	minutes	The delay time interval for cleanup granules.
Max Num Of Concurrent HEG Process		The maximum number of HEG Service requests that my be processed concurrently.
Max Num Of Concur HEG Proc Per Req		The maximum number of HEG Service requests that may be processed concurrently for a single request.
HEG Process Retry Interval	seconds	Retry interval for automatic retry in case the queue is suspended automatically.
Due Date For Media Request	hours	Number of hours from the time the request finished staging that request is due for distribution.
Global Configured Operator Actions Email		Configured email account to send operator actions to.
MediaCapacity (GB)	Float	Size in GB that will fit on 1 volume.
MinRequestSize (GB)	Float	Size in GB for the smallest order to be processed.
MaxRequestSize (GB)	Float	Size in GB for the largest order to be processed.
PartitionSizeLimit (GB)	Float	Size in GB for orders to be partitioned.
MinBundleSize (GB)	Float	Size in GB for smallest bundle.
PartitionGranuleLimit	Int	Number of granules per partition.
Pull Gran Dpl Time (days)	Int	For FtpPull only. Number of days to keep granule in Data Pool.
Pull Gran Dpl Ret Pri (number)	Int	For FtpPull only. Retention Priority.
Min Pri To Preempt (number)	Int	For FtpPull only. Minimum priority to preempt.

ODL Metadata Users Configuration

Note: Limited Capability operators are limited to viewing Metadata File Users configuration only. They cannot add, or delete email addresses.

This page can be accessed by clicking "Metadata File Users" under the **OM Configuration** menu as displayed in Figure 4.7.7-44. This page allows the full-capability operators to configure a list of Email addresses that signifies users that need to receive metadata in ODL .met file format: Whenever the Email address for a Distribution Notice contains one of these addresses, the metadata will be distributed in ODL .met file format. Note that if the list is changed, currently active requests' metadata format will not change. For example, if a user's email address is deleted from the list, active requests issued for that user subsequent to the deletion will still distribute the metadata files in ODL format.

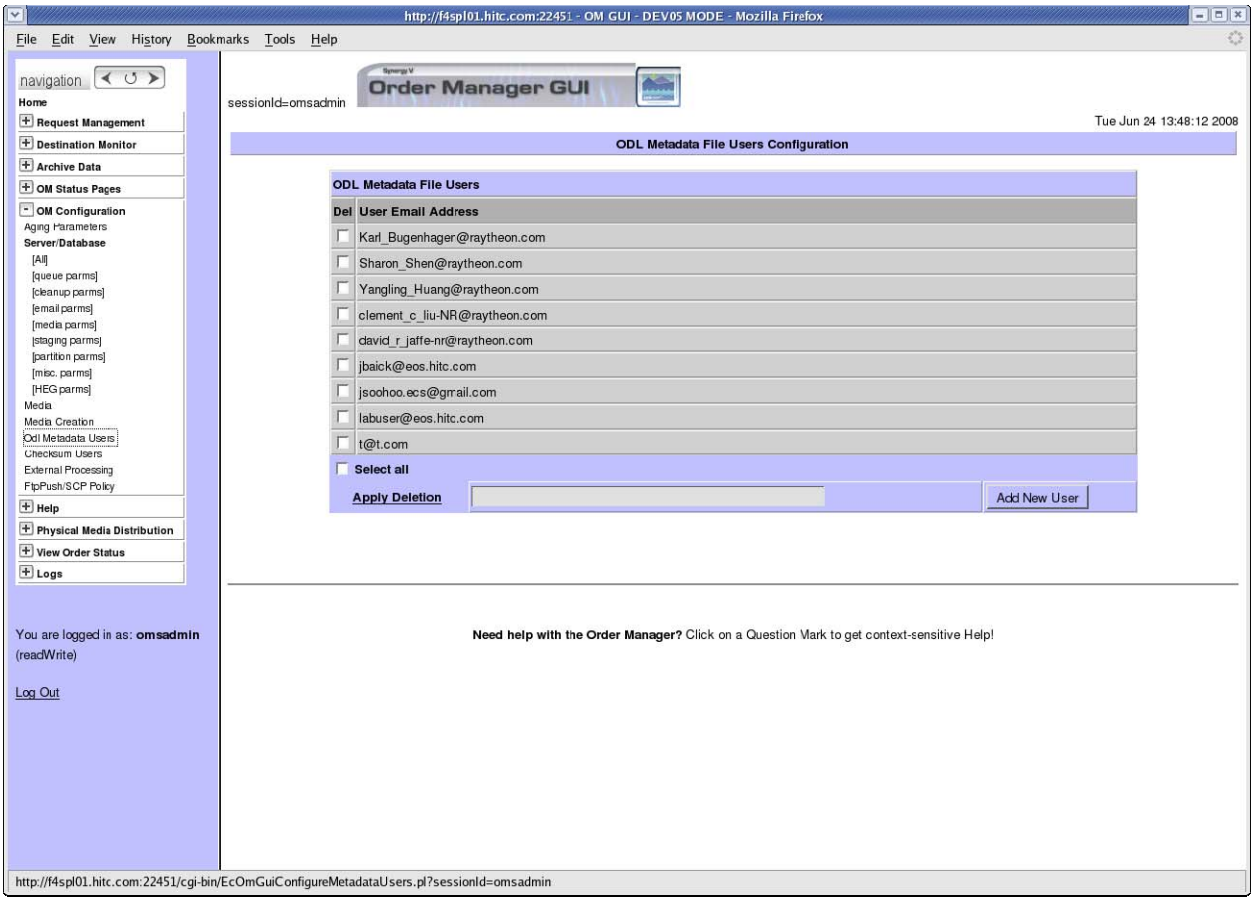


Figure 4.7.7-44. ODL Metadata File Users Configuration Page

Adding a User Email Address

Enter the email address of the user and Click on the “Add New User” button to submit changes to the database. A popup window will ask you to confirm the addition, click on “OK’ button to do so as displayed in Figure 4.7.7-45.

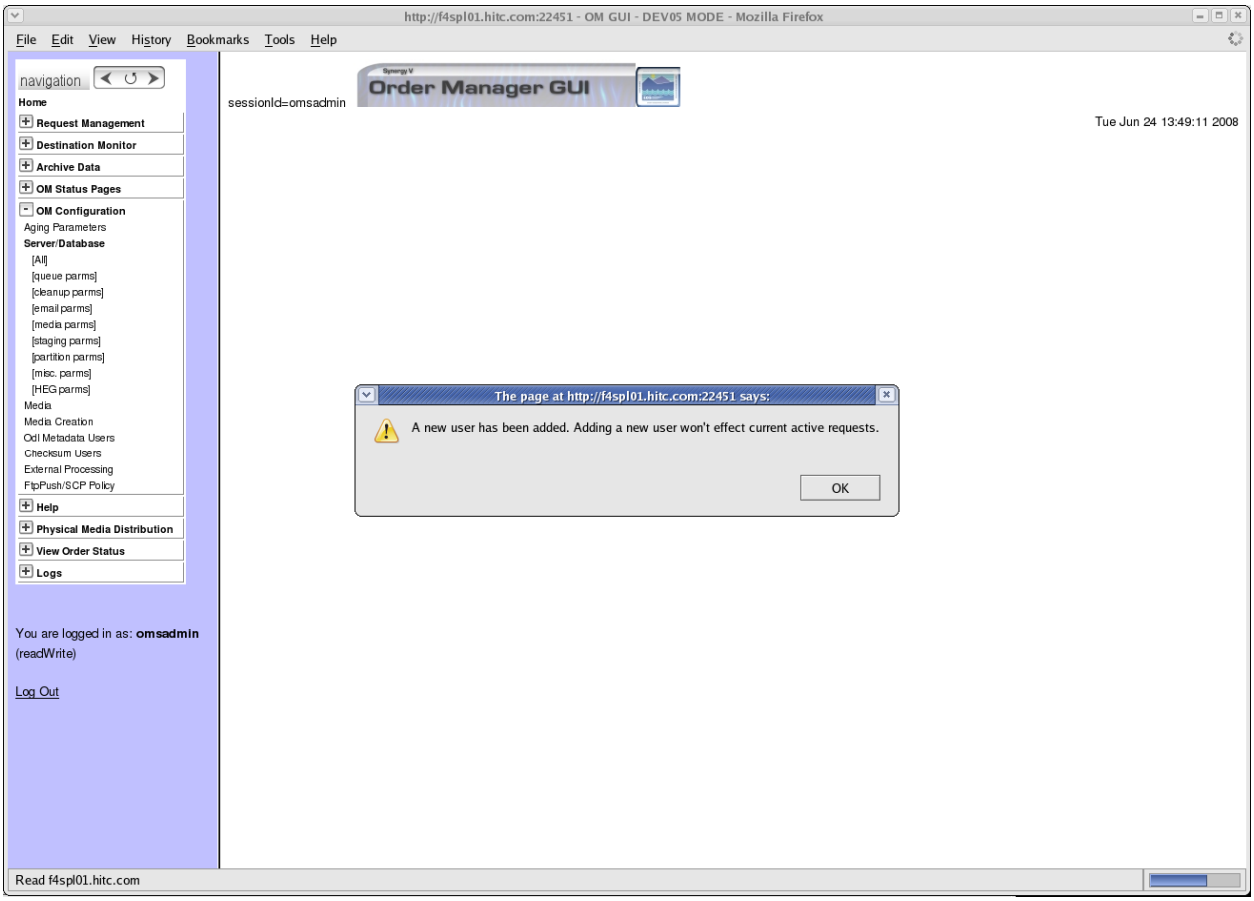


Figure 4.7.7-45. Adding a Metadata User

Deleting User Email Address(es)

Click “Select All” to check User email addresses. In addition, specific users can be selected by clicking their checkboxes individually. Then, click the “Apply Deletion” button to submit changes to the database. A popup window will ask you to confirm the deletion, click on “OK’ button to do so. Otherwise, click “Cancel” button. This is shown in Figure 4.7.7-46.

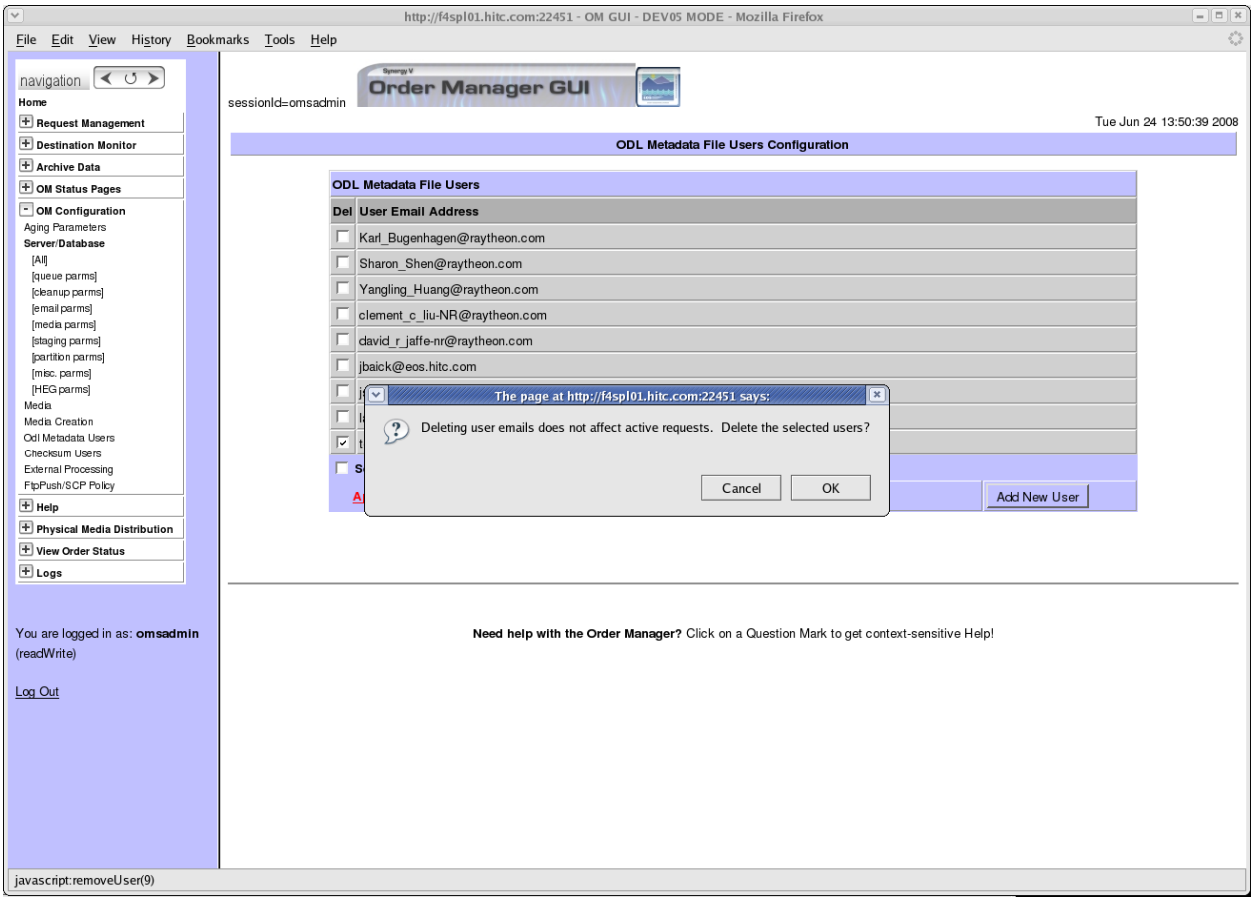


Figure 4.7.7-46. Deleting a Metadata User

External Subsetting Configuration

Note: Limited Capability operators are limited to viewing External Subsetting configuration only. They cannot edit, add, or delete destinations.

This page can be accessed by clicking “External Subsetting” under the **OM Configuration** menu. This page allows the full-capability operators to define and configure the parameters of an external subsetter.

Special configuration parameters that control external subsetting requests are displayed in the **External Subsetting Configuration** page (see Figure 4.7.7-47). Table 4.7.7-19 explains these options in detail.

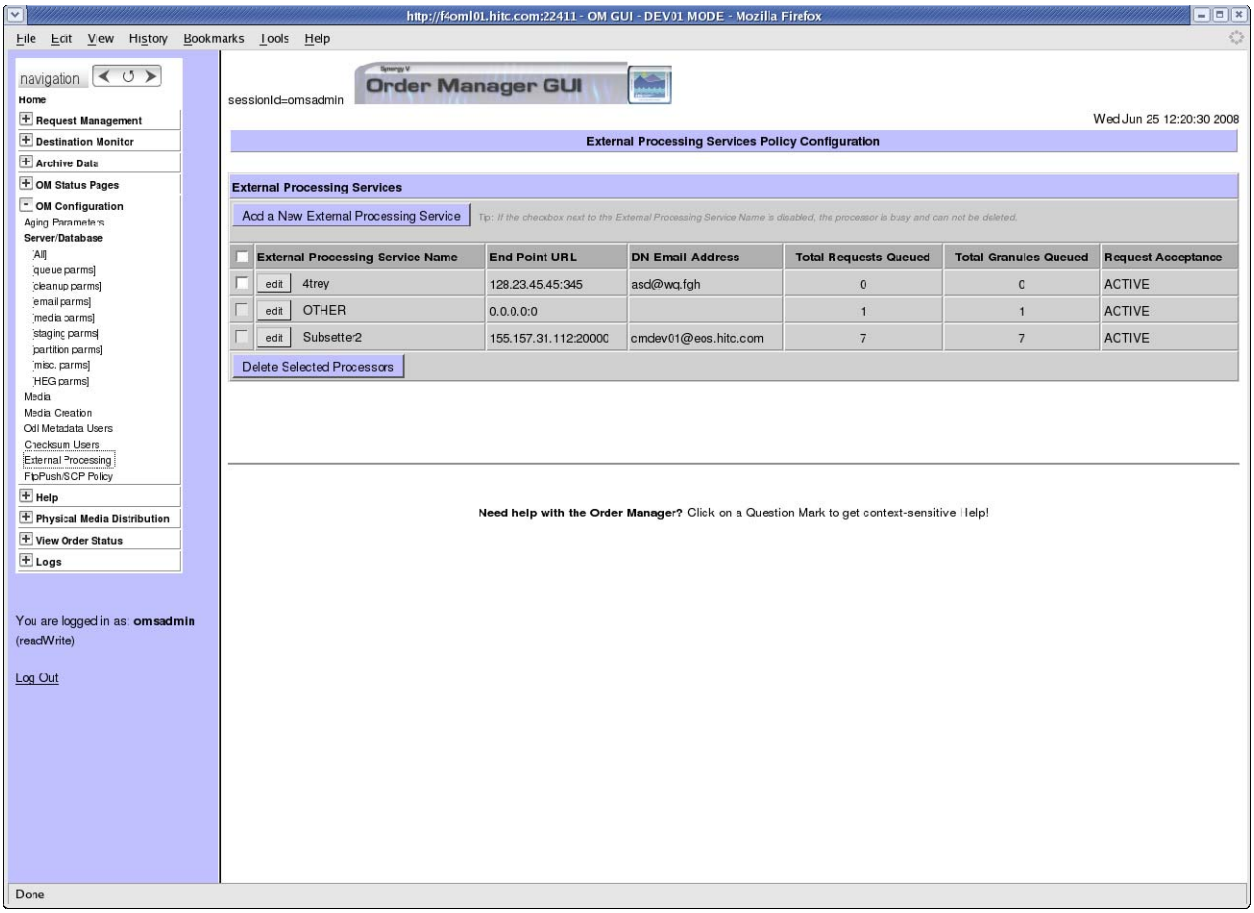


Figure 4.7.7-47. External Subsetting Configuration Page

Figure 4.7.7-47 allows an authorized operator to do the following actions:

1) View a list of external processing services: Processor Name, IP Address, Port Number, DN Email Address, Total Requests Queued, Total Granules Queued, Request Acceptance Status
2) Delete an external processing service if there is no pending request for this external processing service.
3) Add a new processing service by clicking the button
4) Edit existing processing service configuration.

Add External Subsetting Configuration

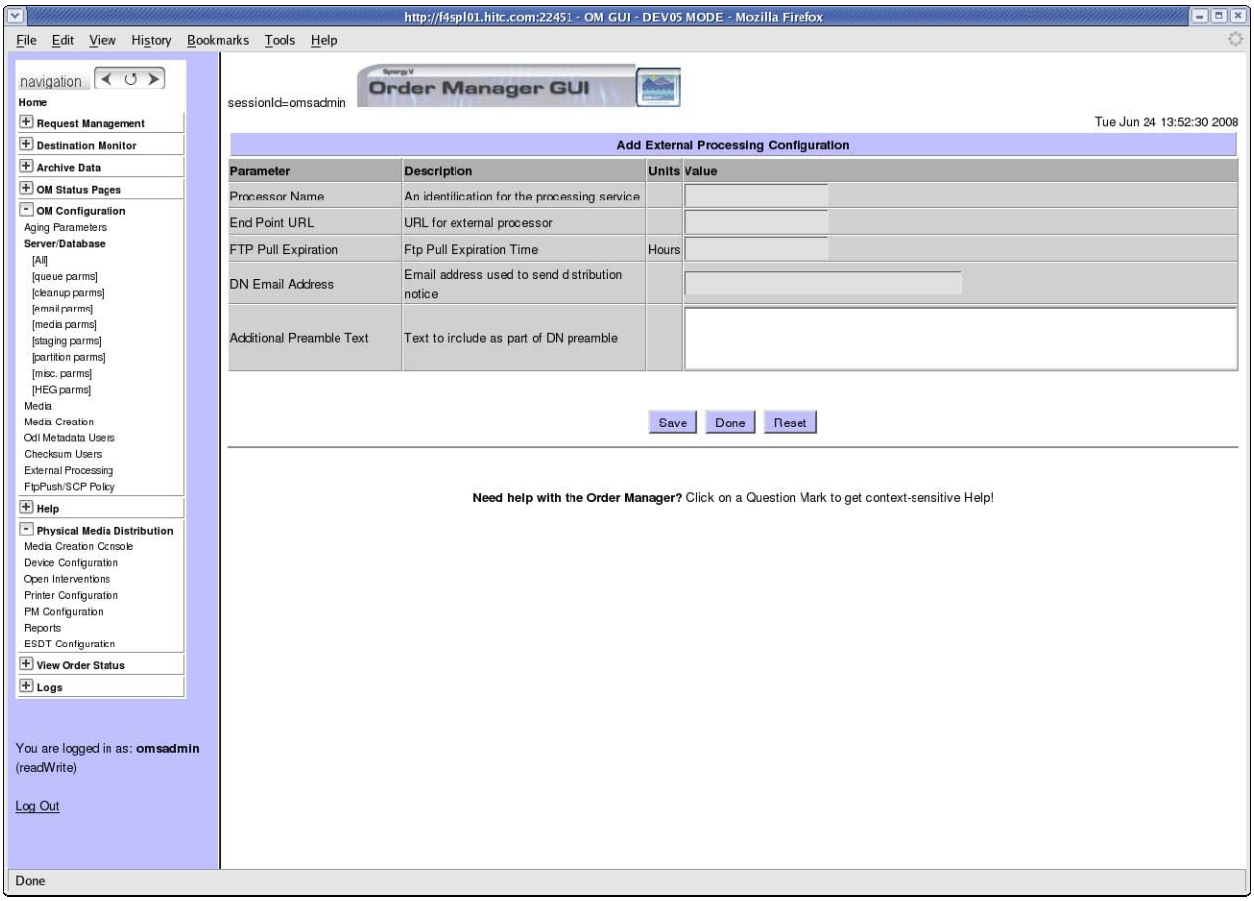


Figure 4.7.7-48. Add External Subsetting Configuration

Figure 4.7.7-48 allows an authorized operator to add a new external processing service using the parameters in Table 4.7.7-19.

Table 4.7.7-19. External Subsetting Configuration Parameters Descriptions

Parameter	Description
Processor name	A unique name for the external processing service.
IP Address	Host IP address for external processing service as configured in the ECS registry.
Port number	Port number for external processing service as configured in the ECS registry.
DN email address	DN Email address used by the external processing service.
Ftp pull expiration	Ftp pull expiration time (Not to exceed the normal FTP Pull order expiration time). The unit is hours.
Additional preamble file	Operator types the text directly in the text box which will be included as part of the preamble in any distribution notices sent to users after completing the distribution of the request for this subsetter.

View/ Edit External Subsetting Configuration

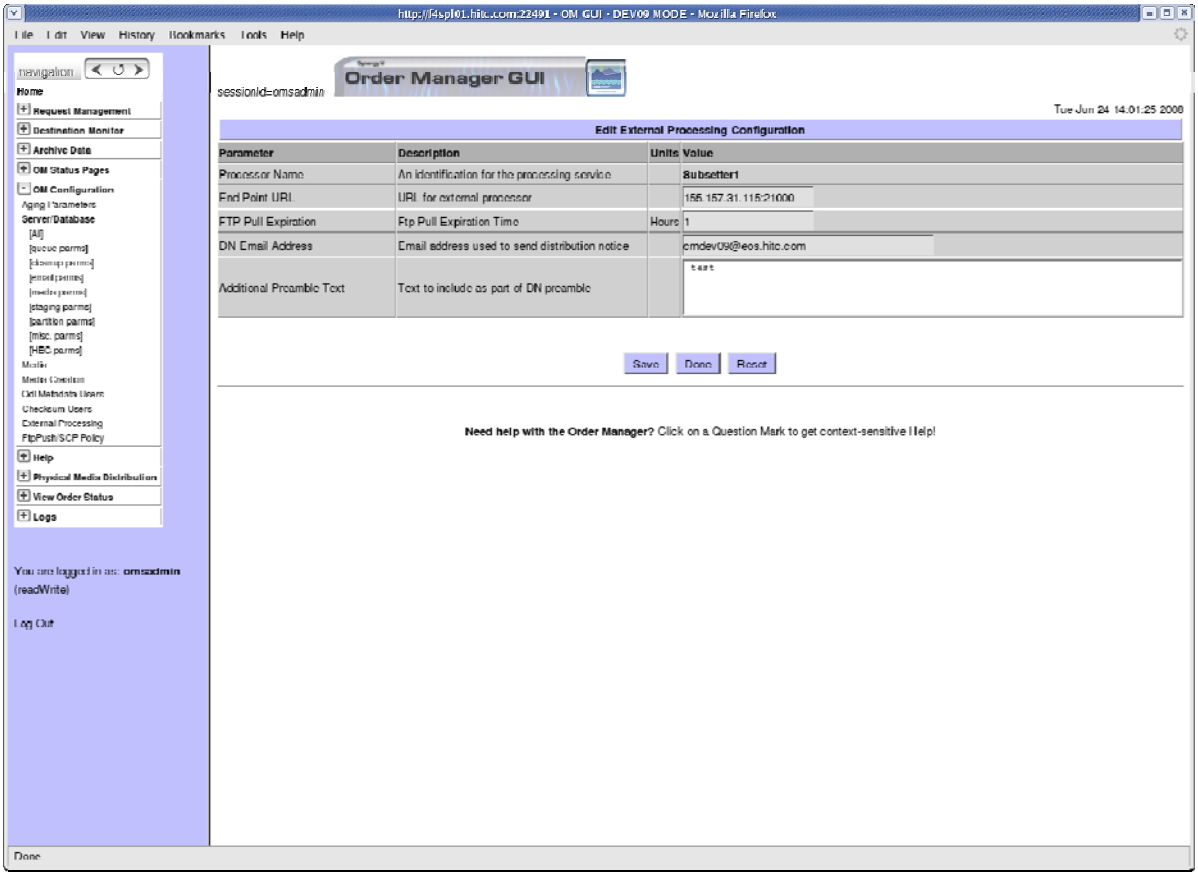


Figure 4.7.7-49. View/ Edit External Subsetting Configuration

Figure 4.7.7-49 allows the authorized operator either view or edit the existing external processing service configuration. Processor Name does not allow to be edited.

4.7.7.6.1 DataAccess Processing

Configuration parameters on the **Data Access Services Configuration** page are grouped by service (Figure 4.7.7-51).

To add a new service:

- Enter in the service name into the Service box. Examples include HEG, GDAL, etc.
- Enter the endpoint URL of the service that is being added into the box under the column labeled Endpoint.
- To set the maximum jobs allowed, enter in a value to the box under the column labeled Max Jobs.
- To set the timeout for communications between the configured service and OMS enter in a value (seconds) under the column labeled Timeout.
- To set the number of times to retry requests sent to the service, enter in a value under the column labeled Retry Interval.
- To finish, select the Add button on the far right side of the row.

In order to edit the values for service that is already configured, first delete the service then add a new service using the steps above using the desired configuration values.

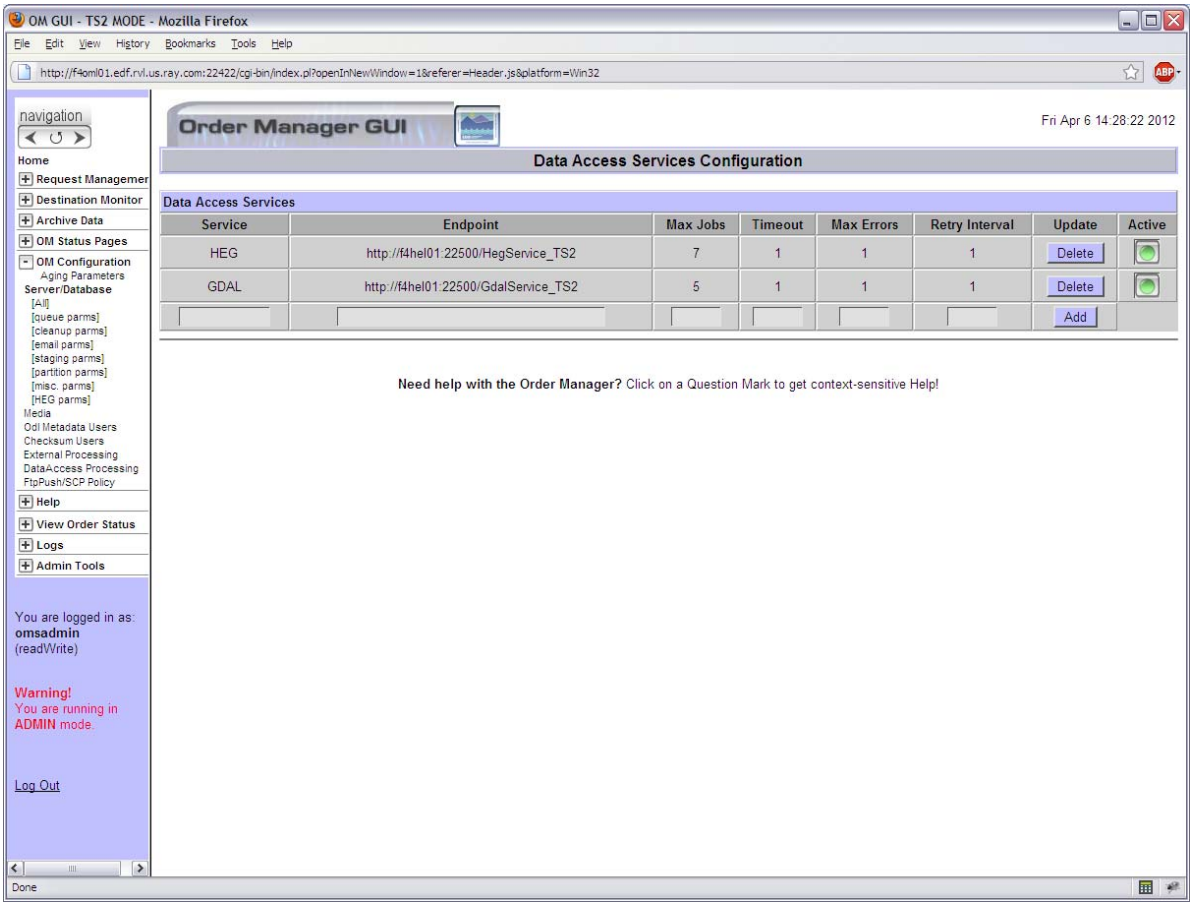


Figure 4.7.7-50. DataAccess Processing Configuration

4.7.7.6.2 FTP Push / SCP Policy Configuration

Note: Limited Capability operators are limited to viewing FTP Push / SCP Policy configuration only. They cannot edit, add, or delete destinations.

This page can be accessed by clicking “FTP Push / SCP Policy” under the **OM Configuration** menu. This page allows the full-capability operators to define and configure the fine-tuning parameters of an FTP Push / SCP destination.

Frequently Used vs. Non-configured Destinations

All FTP Push destinations belong to either the Frequently Used group, or the general non-configured group and all SCP destinations belong to the Frequently used group. All FTP Push destinations not specifically defined as a Frequently Used destination are configured on the front page (see Figure 4.7.7-51) under “Settings for Non-Configured Destinations”. These settings also serve as default values for new destinations.

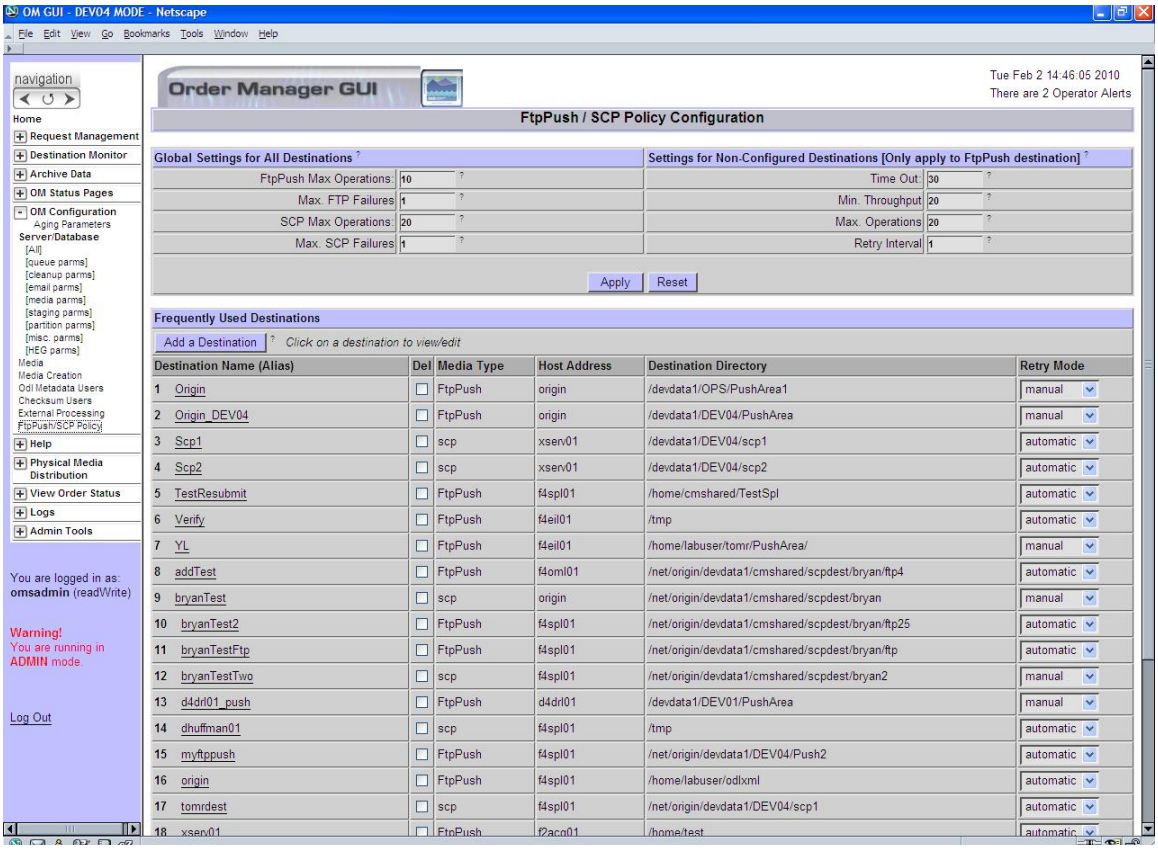


Figure 4.7.7-51. FtpPush/SCP Policy Configuration (Main Page)

Global Settings for All Ftp Push / SCP Destinations

These are two parameters that apply to all destinations regardless of their individual settings: Max Operations and Max Failures for FtpPush and SCP, respectively. Non-configured destination settings only apply to FtpPush destinations.

Adding a Destination

Click on the "Add a Destination" button under the Frequently Used Destinations section of the main page. This will open up a page, shown in Figure 4.7.7-52, which will allow the operator to define and configure a destination for either FtpPush or SCP. A destination must already exist (i.e., it must be a destination that is currently in use by one or more Orders).

The definition of a destination is:

a) Name (Alias): A descriptive name or handle by which the destination can be easily identified. Aliases must be unique.

b) Target Directory: The directory on the remote host to which files will be pushed.

c) Host/IP Address: The remote machine name or IP address.

d) Media Type: FtpPush or SCP

The combination of these attributes constitutes a Frequently Used Destination. All destinations *must* have exclusive attributes and an exclusive Alias.

The configuration parameters for the destination are already preloaded with default values from the non-configured destinations (only apply to FtpPush). The configuration parameters are described in Table 4.7.7-21.

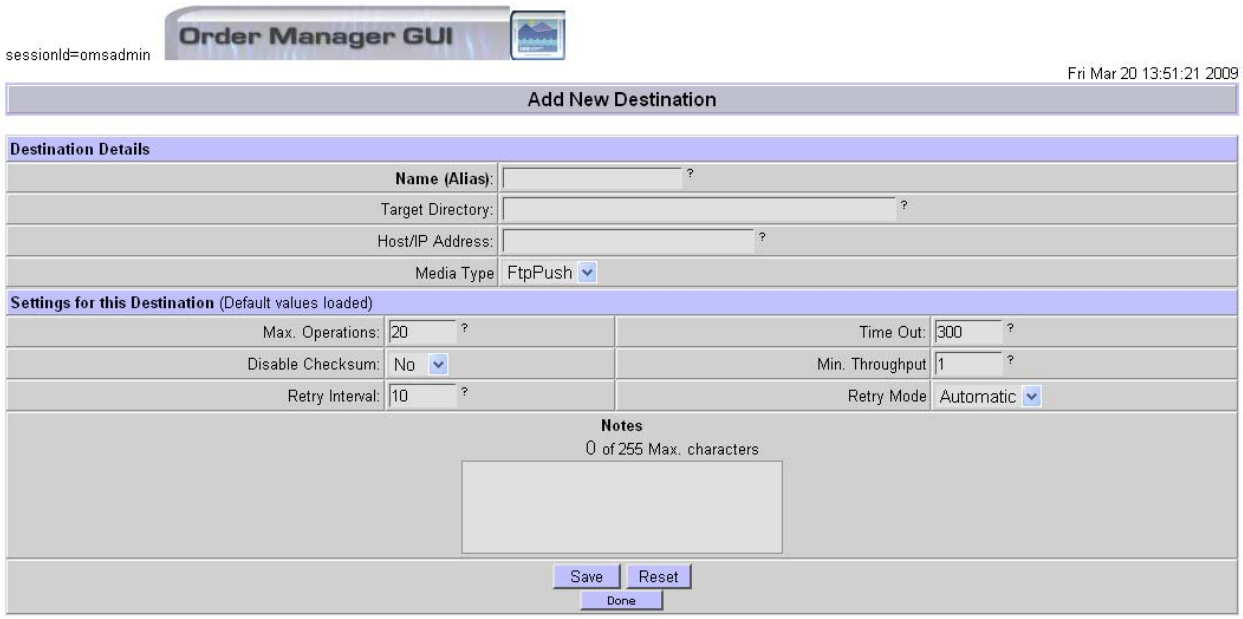


Figure 4.7.7-52. FtpPush/SCP Policy Configuration: Add New Destination

Configuring a Destination

To configure a defined Frequently Used Destination, click on the Destination Name on the main FtpPush/SCP Policy Configuration Page. This will display the details of the configuration for that destination, as shown in Figure 4.7.7-53. From there, you can modify the destination attributes (Target Directory, Host/IP Address) and the configuration parameters for that destination. The (Name (Alias) field cannot be modified. Once you are finished, click "Save" at the bottom of the screen. Click "Done" to move back to the main FtpPush/SCP Policy Configuration page. **Note:** The "Done" button will *not* save any changes made to the destination – always click "Save".

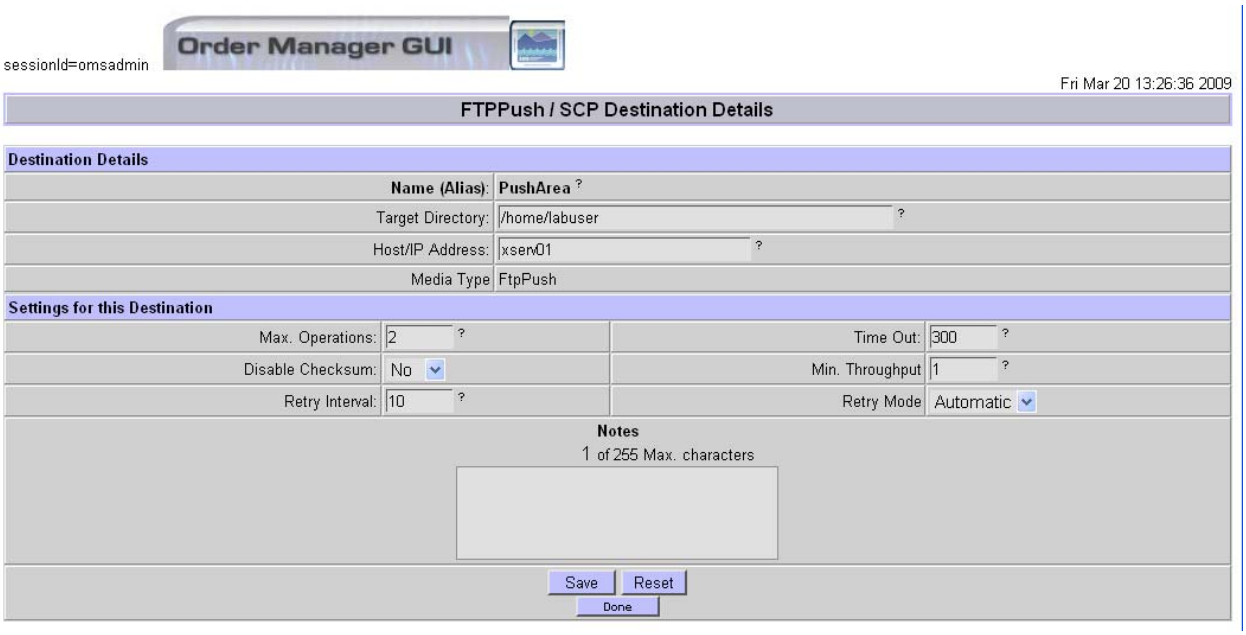


Figure 4.7.7-53. FtpPush Policy Configuration: FtpPush Destination Detail

Removing a Destination

To remove a destination from the Frequently Used Destination group, go to the main FtpPush/SCP Policy Configuration page and select the destination you wish to delete by checking the box next to the destination name in the Del column. Once you have selected the destinations you wish to remove, click on "Delete Selected Destinations" at the bottom of the screen. You will be prompted for confirmation.

Removing a destination does not actually delete the destination. Rather, it moves that destination to the non-configured group and erases its individual configuration parameters.

Table 4.7.7-20. FtpPush/SCP Policy Configuration Parameters (1 of 2)

Parameter	Scope	Data Type	Description
Max Operations	Global	Int	The maximum number of concurrent FTP Push Operations for *all* destinations added together.
Max. FTP Failures	Global	Int	The maximum number of consecutive FTP transfer failures for any destination, which, when exceeded, causes the suspension of that destination.
Max. SCP Operations	Global	Int	The maximum number of concurrent SCP Operations for all destinations added together.

Table 4.7.7-20. FtpPush/SCP Policy Configuration Parameters (2 of 2)

Parameter	Scope	Data Type	Description
Max. SCP Failures	Global	Int	The maximum number of consecutive SCP transfer failures for any destination, which, when exceeded, causes the suspension of that destination.
Disable Checksum	Destination	Yes/No	Allows user to disable checksumming of file distributed to this destination.
Time Out	Destination	Int	An extra time allotment that is applied to the expected throughput, such that: expected throughput = min. throughput + timeout.
Min. Throughput	Destination	Float	The minimum data throughput in MB/sec for a particular destination.
Max. Operations	Destination	Int	The maximum number of concurrent FTP Push Operations for a particular destination (exclusive of but subject to the global Max Operations).
Retry Interval	Destination	Int	The waiting period, in minutes, before FTP Push operations for a suspended destination are automatically retried.
Retry Mode	Destination	n/a	Specifies whether this destination should retry automatically or manually. For Non-Configured Destinations, this is always Automatic.

4.7.7.8 Help Page

The operator can view the help information on a particular page by clicking on the **Need help with the Order Manager?** link at the bottom of the page which will display a small pop-up window for help on that page. The operator may also click on the **Help** tab at the top of the page. The help information is indexed and also contains links to help on related topics. The index to available topics includes:

- About The Order Manager GUI
- Recently Added Features
- Request Management
  - Open Interventions
    - Viewing Intervention Details
    - Working an Intervention
  - Operator Alerts
  - Completed Interventions
  - Distribution Requests

- FtpPush Monitor
  - FtpPush Distributions Requests
  - FtpPush Operations
  - FtpPush Destinations
  - Staging Requests
- OM Status Pages
  - OM Queue Status
  - Staging Operations
  - Staging Status by Media Type
  - Staging by FtpPush Destination
- OM Queue Status
- OM Configuration
  - Aging Parameters
  - Server/Database Configuration
  - Media Configuration
  - FtpPush Policy Configuration
  - Archive Resources
- OM Server Statistics
- OM Log Viewer

Figure 4.7.7-54 displays a sample Help Page.

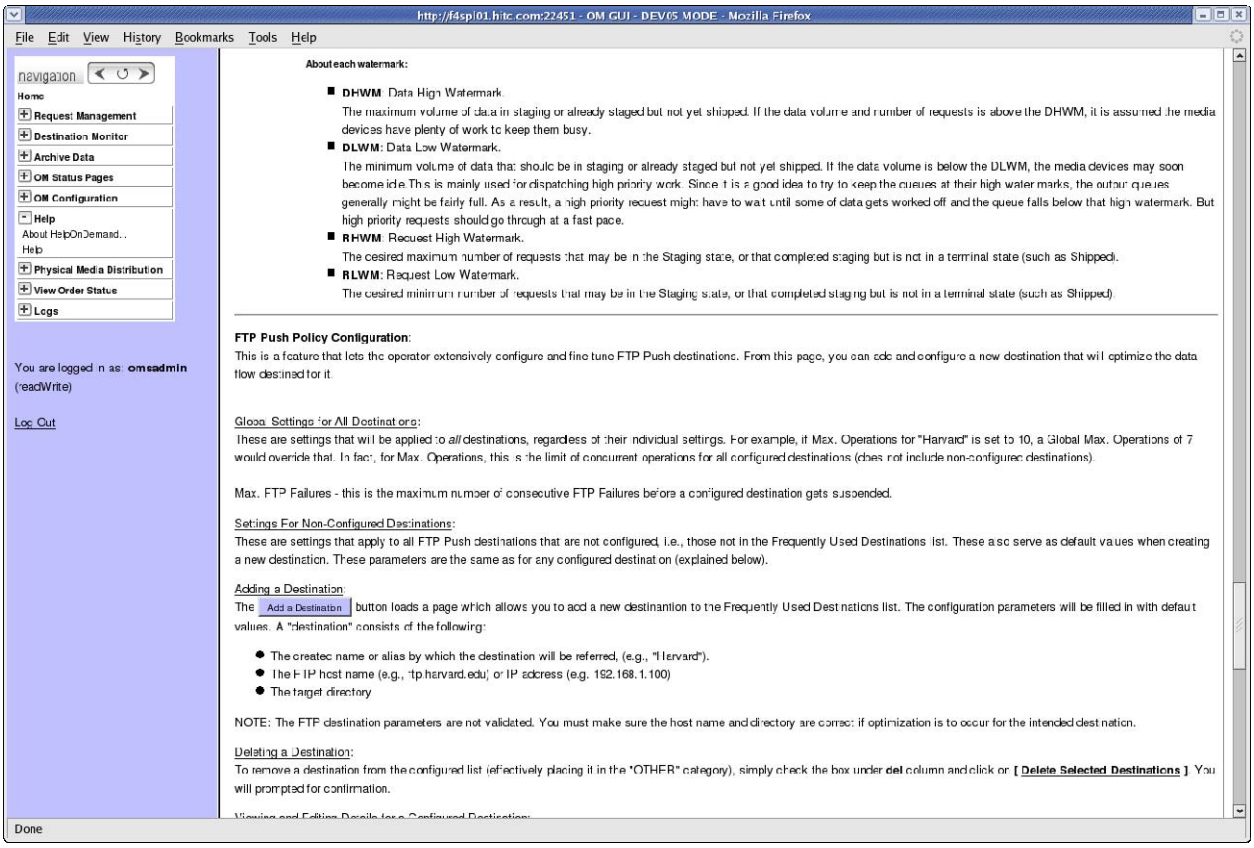


Figure 4.7.7-54. Sample Help Page

HelpOnDemand

This is a feature that gives the operator context-sensitive help for each page, but more specifically for particular controls or parameters that may not be entirely self-descriptive. Anywhere there is little question mark next to a button or text field, click on it and a dialog box describing that item will appear. Figure 4.7.7-55 shows an example of HelpOnDemand for the Time Out parameter on the FtpPush Policy Configuration page.

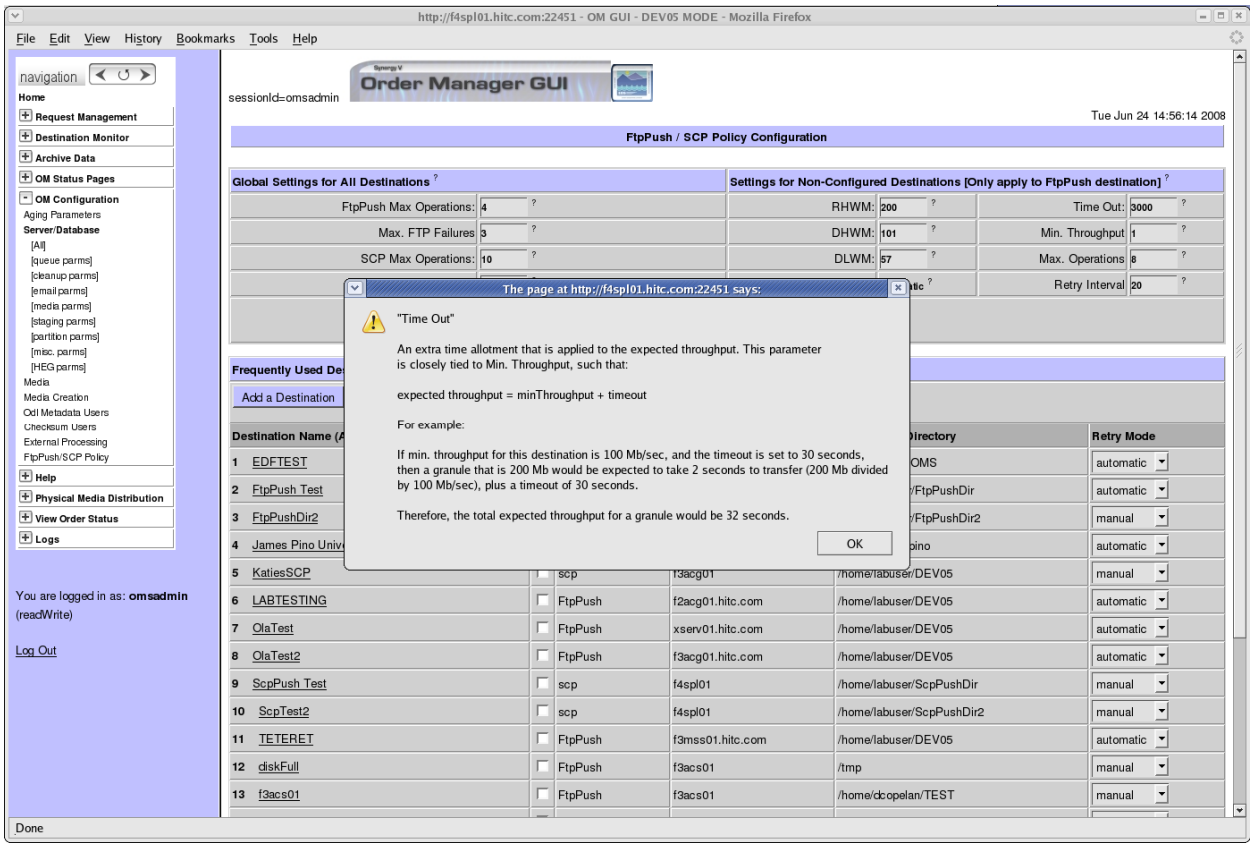


Figure 4.7.7-55. HelpOnDemand Example

4.7.7.9 OM GUI Log Viewer

The Log viewer, shown in Figure 4.7.7-56, is a simple diagnostics tool to aid the operator when an error occurs. It lets you view part or the entire Order Manager Page log file, which is a file specifically generated for the OM GUI by the OM GUI. It is usually sufficient to view the last 200-500 lines for recent activity. Simply enter the last number of lines of the log file you wish to view and click "OK". The entire log may be viewed by leaving the text box empty (or entering 0, or a number greater than or equal to the total number of lines in the file) and clicking on "OK".

Since the log file can grow to a very large size after continued use of the Order Manager Page, it is not recommended to load the entire log file all at once.

A helpful feature is included that shows or hides the line numbers, so that the log text can be easily cut and pasted to other places. This is especially useful for SQL:

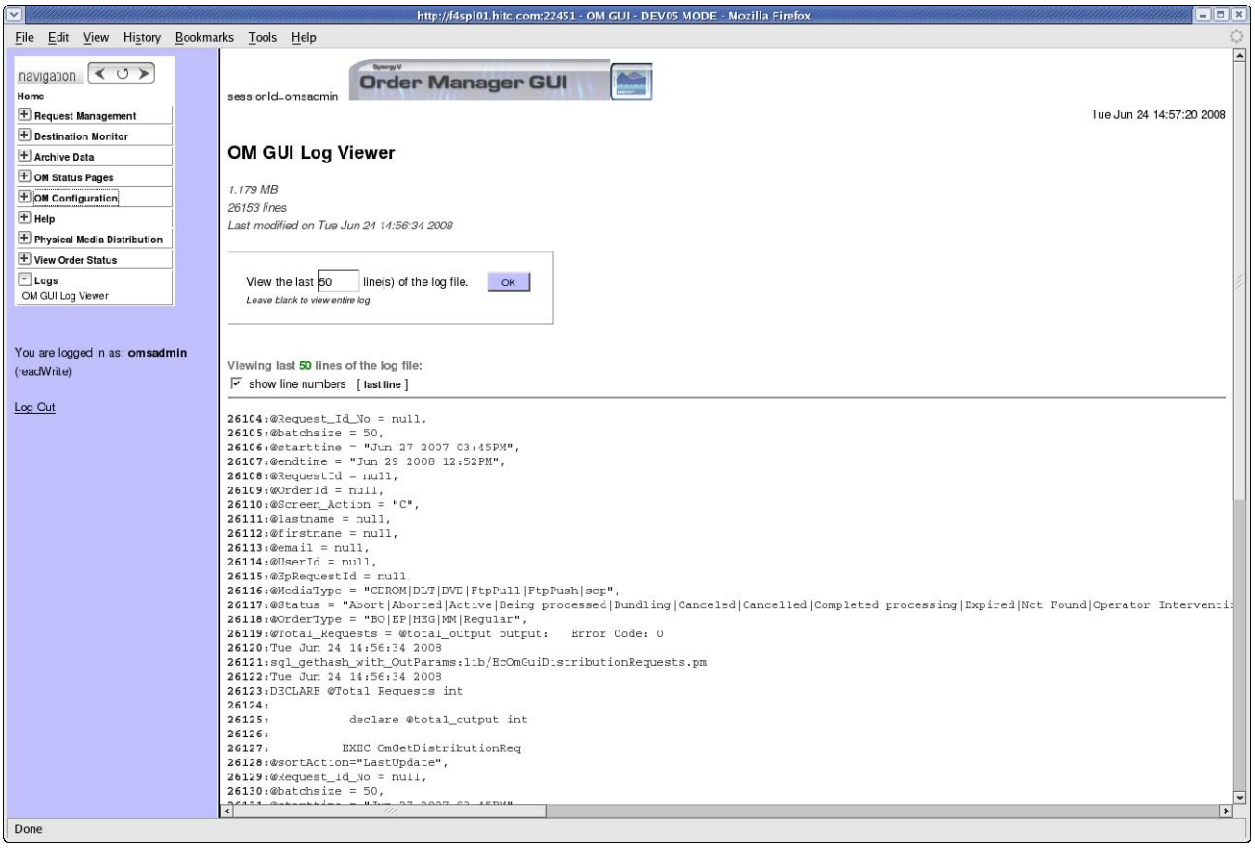


Figure 4.7.7-56. OM GUI Log Viewer Example

4.7.7.10 Required Operating Environment

The following environment is required for the OM GUI to work properly.

The O/S requirements are Linux.

The OM GUI requires the installation of Netscape 7.0 or higher.

4.7.7.11 Interfaces and Data types

The OM GUI exchanges data between the Web Browser and database, using Perl CGI and DBI Modules for the interface.

4.7.7.12 Databases

The OM GUI accesses the OMS database.

4.7.7.13 Special Constraints

There are no special constraints to running the OM GUI.

4.7.7.14 Outputs

There are no outputs from the OM GUI except for status and error messages.

4.7.7.15 Events and Messages

The OM GUI writes status and error messages to the EcOmGui.log file in the directory /usr/ecs/<MODE>/CUSTOM/WWW/OMS/cgi-bin/logs.

4.7.7.16 Reports

The OM GUI does not generate reports.

4.7.8 Science Command Line Interface (OmSCLI) in OMS

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

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

4.7.8.1 Quick Start Using the OmSCLI

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

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

Table 4.7.8-1 describes the OmSCLI command line parameters.

Table 4.7.8-1. Command Line Parameters of the SCLI Tool

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

4.7.8.2 OmSCLI Command Line Parameter Formats

-p <parameterfile>

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

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

DDISTMEDIAFMT = FILEFORMAT (required)

FTPUSER = <FTP user id> (required for FtpPush, SCP distributions)
FTPPASSWORD = <FTP Password> (required for FtpPush, SCP distributions)
FTPHOST = <FTP host name> (required for FtpPush, SCP distributions)
FTPPUSHDEST = <full path name of push directory on FTPHOST>
(required for FtpPush, SCP distributions)

ECSUSERPROFILE = <user id associated with the request> (required)
PRIORITY = HIGH | VHIGH | NORMAL | LOW | XPRESS. (required)
DDISTNOTIFYTYPE = MAIL (required)
NOTIFY=<email address> (required)

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

Example of parameter file:

1. **FTP Pull:**

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

2. **Ftp Push**

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

3. **SCP**

PRIORITY=VHIGH
DDISTMEDIATYPE=scp

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

-t <tag>

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

-f <file>

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

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

Example granule identifiers:

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

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

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

4.7.8.3 SCLI Command Line Utility Configuration File

The Command Line Utility has an associated configuration file with values stored in a basic PARAMETER = VALUE format. The configuration file is called EcOmSCLI.CFG, and is stored in the /usr/ecs/<mode>/CUSTOM/cfg directory for the mode. Table 4.7.8-2 describes its contents:

Table 4.7.8-2. OmSCLI Configuration File Parameters

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

4.7.8.4 Required Operating Environment

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

4.7.8.5 Interfaces and Data

Table 4.7.8-3 lists the supporting products this tool depends upon to function properly.

Table 4.7.8-3. Interface Protocols

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

4.7.8.6 Databases

Table 4.7.8-4 lists the databases stored procedures and tables used by the command line utility.

Table 4.7.8-4. 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>, EcInDb_<MODE>, and DataPool_<MODE> databases. In addition, the name EcOmSCLI must be added as a login.

4.7.8.7 Special Constraints

The acquire wrapper script and EcCoEnvPerl must be located in the

/usr/ecs/<mode>/CUSTOM/utilities directory of the mode.

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

4.7.8.8 Outputs

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

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

Example of sample output from OmSCLI:

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

11/09/09 10:31:58: Thread ID : 32374 : Configuring connection pool parameters. Server=f4dbl03_srvr, database=EcOmDb_DEV04, user=EcOmSCLI
About to execute SQL: exec OmGetGranulesByLGID
'MODPTQKM.A2005310.h00v09.004.2005312161035.hdf.0220.1193165254.72679.RGEN.hdf'

11/09/09 10:31:58: Thread ID : 32374 : EcDbInterface::Execute SQL=exec OmGetGranulesByLGID
'MODPTQKM.A2005310.h00v09.004.2005312161035.hdf.0220.1193165254.72679.RGEN.hdf'
The tag is: yl1238
OmSCLI: The new rpcId Id is : 9fa8bfe4-1dd1-11b2-ab93-9d1177396254:SCLI:yl1238
11/09/09 10:31:58: Thread ID : 32374 : EcDbInterface::Execute SQL=exec OmValidateRpcId '9fa8bfe4-1dd1-11b2-ab93-9d1177396254:SCLI:yl1238'
11/09/09 10:31:58.171769 32374: 0xff94fa50OmSrDistrRequest()::ctor(void):
11/09/09 10:31:58.171912 32374: ENTRY: OmSrListManager::GetFtpPushMediaType
11/09/09 10:31:58.171936 32374: ENTRY: OmSrListManager::AddFtpPushMediaType
11/09/09 10:31:58.171995 32374: SetPriority: new priority 230 for request
11/09/09 10:31:58.172098 32374: 0xff94fa50OmSrRequest()::AddGranule Id=0
11/09/09 10:31:58.172134 32374: 0xff94fa50OmSrRequest()::Adding to PENDING Id=0
11/09/09 10:31:58.172165 32374: 0xff94fa50OmSrRequest()::AddGranule Id=0
11/09/09 10:31:58.172181 32374: 0xff94fa50OmSrRequest()::Adding to PENDING Id=0
11/09/09 10:31:58.172203 32374: 0xff94fa50OmSrRequest()::AddGranule Id=0
11/09/09 10:31:58.172220 32374: 0xff94fa50OmSrRequest()::Adding to PENDING Id=0
11/09/09 10:31:58.172661 32374: Declare @o_RequestId VARCHAR(10), @o_OrderId VARCHAR(10)
exec OmCreateNonBundlingOrder 'labuser', '', null, null, null, 'FtpPush', 'Yangling_Huang@Raytheon.com', 'MAIL', 'HIGH', 'Yangling_Push', 'labuser', 'ad46f72447533e65', 'f4eil01', '/home/labuser/tomr/PushArea/', null, '9fa8bfe4-1dd1-11b2-ab93-9d1177396254:SCLI:yl1238' ,@o_RequestId output, @o_OrderId output
11/09/09 10:31:58: Thread ID : 32374 : EcDbInterface::Execute SQL=Declare @o_RequestId VARCHAR(10), @o_OrderId VARCHAR(10)
exec OmCreateNonBundlingOrder 'labuser', '', null, null, null, 'FtpPush', 'Yangling_Huang@Raytheon.com', 'MAIL', 'HIGH', 'Yangling_Push', 'labuser', 'ad46f72447533e65', 'f4eil01', '/home/labuser/tomr/PushArea/', null, '9fa8bfe4-1dd1-11b2-ab93-9d1177396254:SCLI:yl1238' ,@o_RequestId output, @o_OrderId output
11/09/09 10:31:58.180993 32374: CREATE TABLE #TEMPGRANS (ECS_GranuleId NUMERIC(16,0) NULL, DPL_GranuleId NUMERIC(16,0) NULL, EsdtType VARCHAR(12) NULL, ShortName VARCHAR(8) NULL, VersionID tinyint NULL, ShortName_Char CHAR(8) NULL, GranType CHAR(2) NULL)
CREATE TABLE #GRANS (granuleId NUMERIC(16,0) NULL, esdtType CHAR(12) NULL, granSize FLOAT NULL, granType CHAR(2) NULL)
CREATE TABLE #REQGRANS (granuleId NUMERIC(16,0) NOT NULL)
11/09/09 10:31:58: Thread ID : 32374 : EcDbInterface::Execute SQL=CREATE TABLE #TEMPGRANS (ECS_GranuleId NUMERIC(16,0) NULL, DPL_GranuleId NUMERIC(16,0) NULL, EsdtType VARCHAR(12) NULL, ShortName VARCHAR(8) NULL, VersionID tinyint NULL, ShortName_Char CHAR(8) NULL, GranType CHAR(2) NULL)
CREATE TABLE #GRANS (granuleId NUMERIC(16,0) NULL, esdtType CHAR(12) NULL, granSize FLOAT NULL, granType CHAR(2) NULL)
CREATE TABLE #REQGRANS (granuleId NUMERIC(16,0) NOT NULL)
11/09/09 10:31:58.211691 32374: insert #GRANS values (88829, 'MODPTQKM.086', NULL, 'SC')
11/09/09 10:31:58: Thread ID : 32374 : EcDbInterface::Execute SQL=insert #GRANS values (88829, 'MODPTQKM.086', NULL, 'SC')
11/09/09 10:31:58.222394 32374: insert #GRANS values (170750, 'AE_Land.002', NULL, 'SC')
11/09/09 10:31:58: Thread ID : 32374 : EcDbInterface::Execute SQL=insert #GRANS values (170750, 'AE_Land.002', NULL, 'SC')
11/09/09 10:31:58.224531 32374: insert #GRANS values (90426, 'MODPTQKM.086', NULL, 'SC')

11/09/09 10:31:58: Thread ID : 32374 : EcDbInterface::Execute SQL=insert #GRANS values (90426, 'MODPTQKM.086', NULL, 'SC')
11/09/09 10:31:58.225442 32374: exec OmPreValidate "0800031139"
11/09/09 10:31:58: Thread ID : 32374 : EcDbInterface::Execute SQL=exec OmPreValidate "0800031139"
11/09/09 10:31:58.275541 32374: The granules are all valid
11/09/09 10:31:58.275632 32374: ENTRY: OmSrDbInterface::BeginTransaction
11/09/09 10:31:58: Thread ID : 32374 : EcDbInterface::Execute SQL=begin transaction
11/09/09 10:31:58.275818 32374: exec OmInsBulkGranules_nondp '0800031139'
11/09/09 10:31:58: Thread ID : 32374 : EcDbInterface::Execute SQL=exec OmInsBulkGranules_nondp '0800031139'
11/09/09 10:31:58.297726 32374: exec OmCompleteReqSubmission '0800031139'
11/09/09 10:31:58: Thread ID : 32374 : EcDbInterface::Execute SQL=exec OmCompleteReqSubmission '0800031139'
11/09/09 10:31:58.300033 32374: ENTRY: OmSrDbInterface::EndTransaction
11/09/09 10:31:58: Thread ID : 32374 : EcDbInterface::Execute SQL=commit transaction
Order Submitted to OMS Successfully.
11/09/09 10:31:58.303448 32374: 0xff94fa50OmSrRequest(0800031139)::dtor: 0800025127
OrderId is: 0800025127
RequestId is: 0800031139

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

4.7.8.9 Event and Error Messages

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

4.7.8.10 Reports

None

This page intentionally left blank.

4.7.9 Overview of the ESDT Maintenance GUI

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

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

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

4.7.9.1 Login Page

This page first appears when the application is loaded. The operator will be required to enter a pre-assigned password, as shown in Figure 4.7.9-1. Once the operator is logged in, the Install ESDT Page will be displayed and the application will be enabled.

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

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

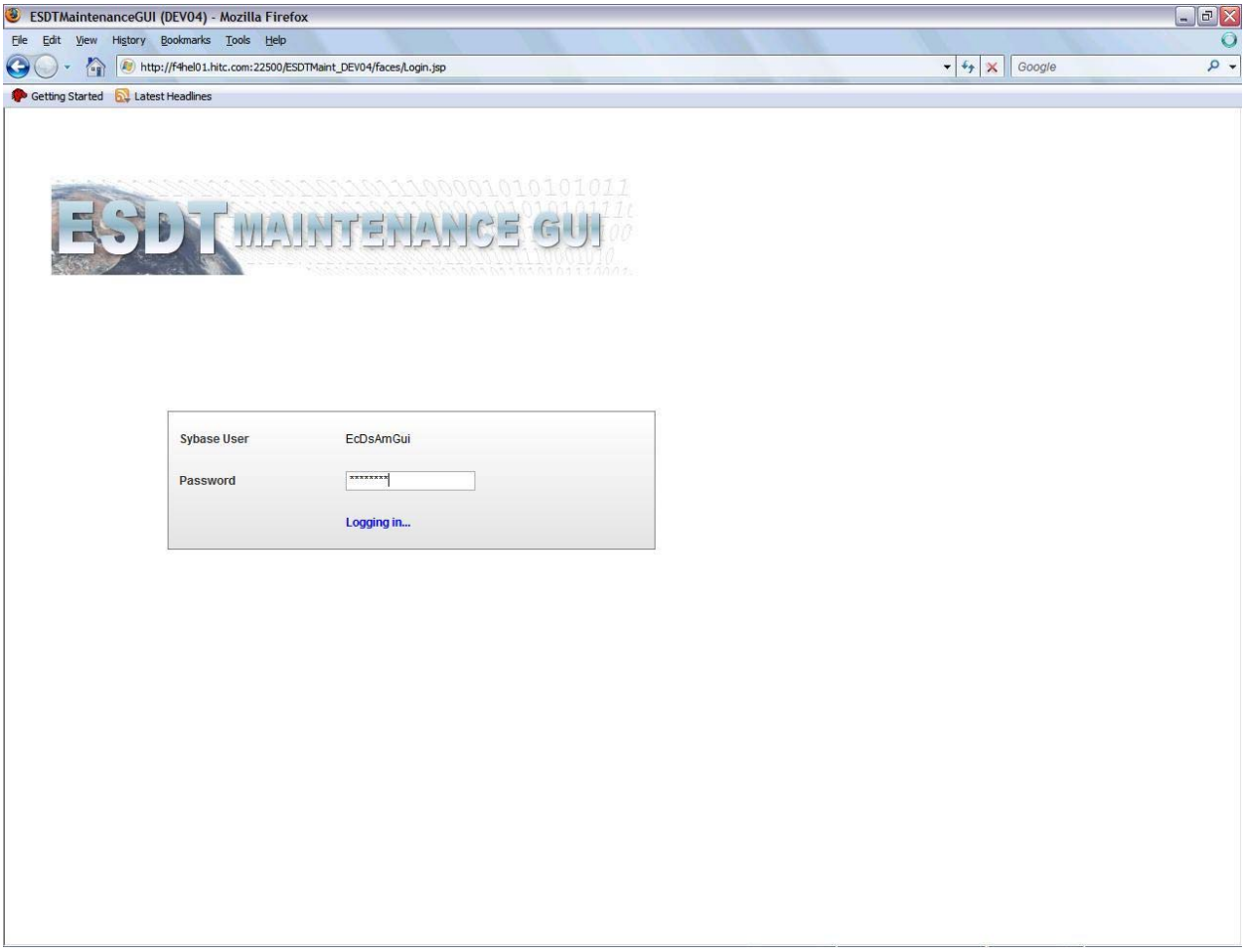


Figure 4.7.9-1. Login Page

4.7.9.2 List ESDT Page

This page first appears when the operator is logged in. The page lists the currently installed ESDTs, as shown in Figure 4.7.9-2.

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

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

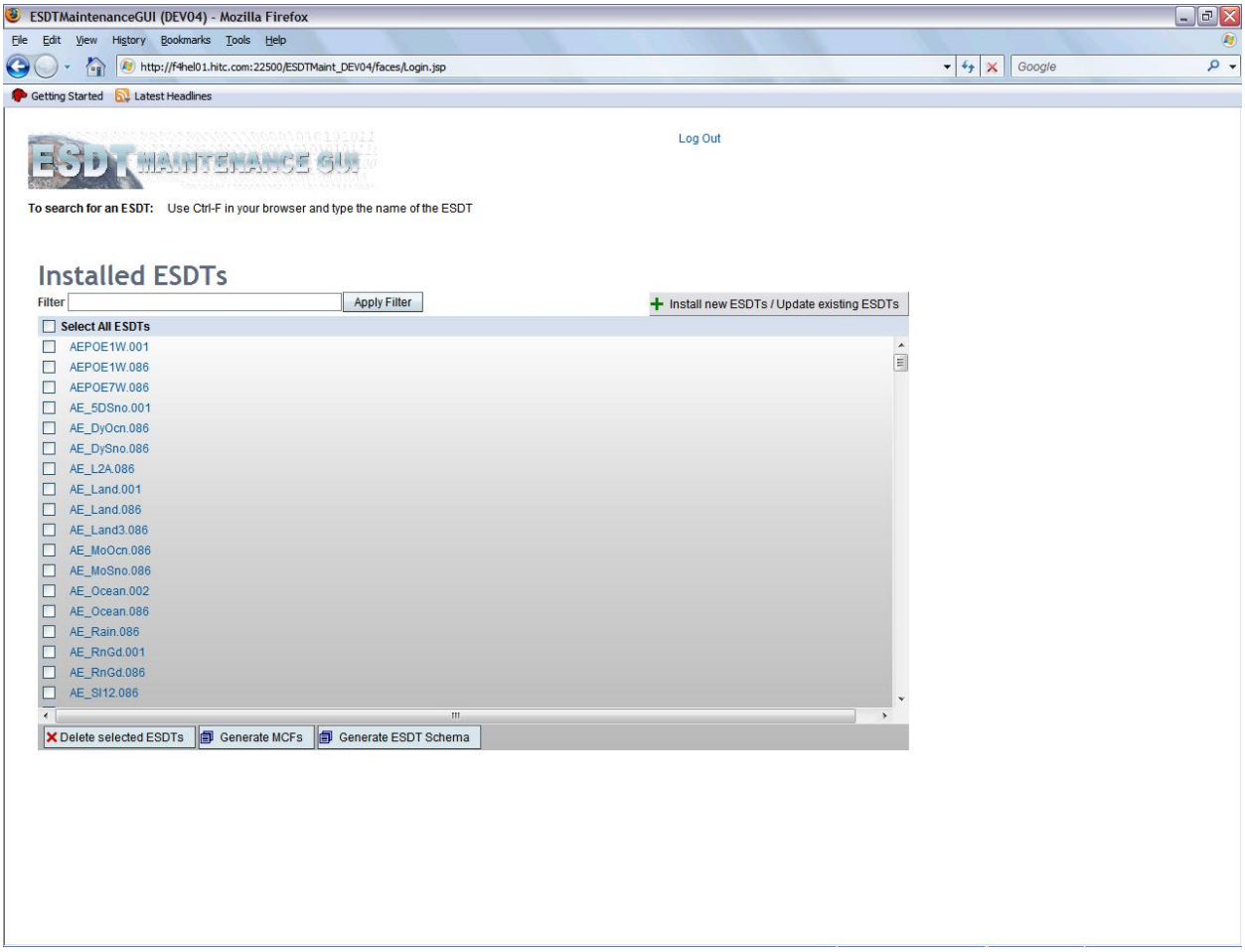


Figure 4.7.9-2. List ESDT Page

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

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

4.7.9.2.2 Filtering the Install ESDT Page

The List ESDT Page includes a filter that can be applied to the list of installed ESDTs. This is useful for selecting particular types of ESDTs for bulk action (i.e., deletion, MCF or schema generation). This is a simple text search and will search based upon the ESDT short name. As shown in Figure 4.7.9-3, "GLA" would return any ESDT with the "GLA" anywhere in the short name. The search is also case-sensitive.

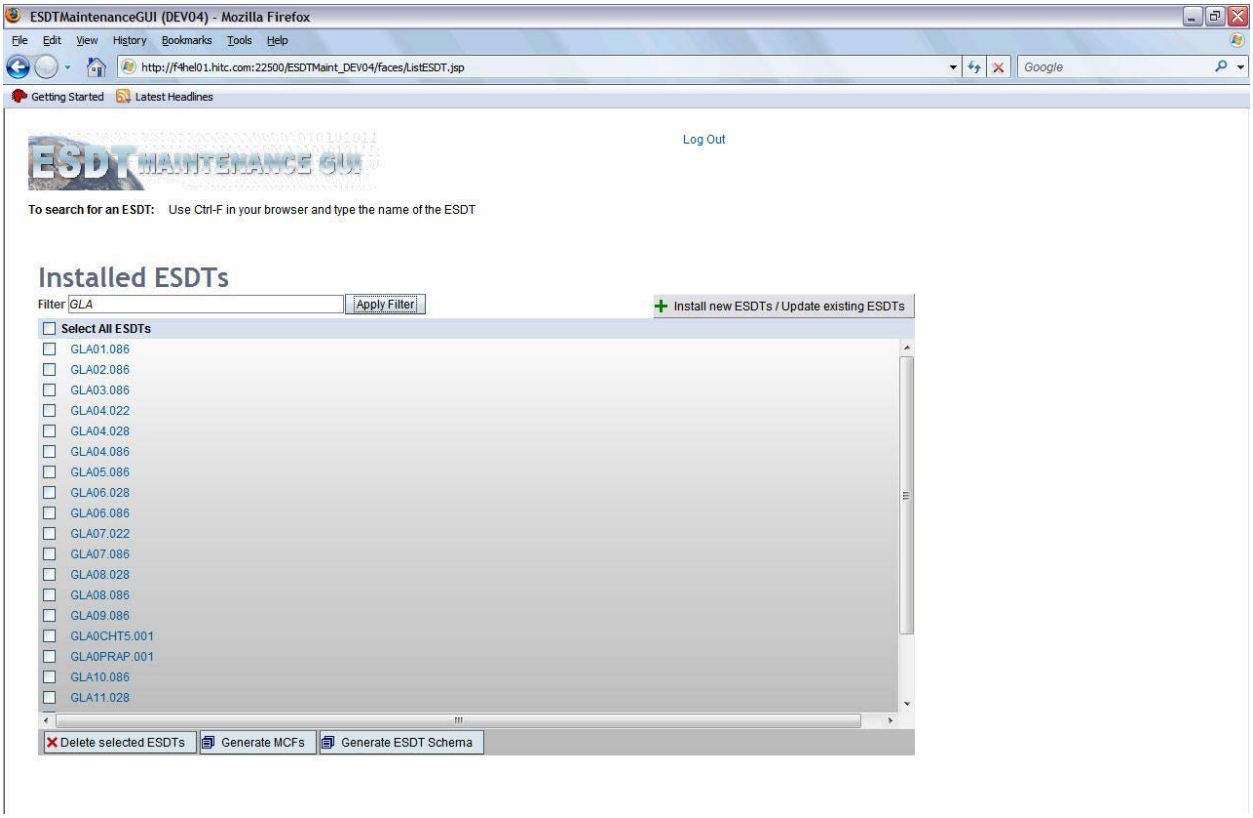


Figure 4.7.9-3. Filtering the List ESDT Page

4.7.9.2.3 View Descriptor Information for a Specific ESDT

From the List ESDT Page, click an ESDT link as shown in Figure 4.7.9-4 to view the ODL and XML descriptor information for a specific ESDT, and this will direct you to the ESDT Detail Page (see section 4.7.9.3 for more details).

4.7.9.2.4 Delete Selected ESDT Action

From the List ESDT Page, the operator can select a list of ESDTs and delete them by clicking on the “Delete selected ESDTs” button as shown in Figure 4.7.9-5.

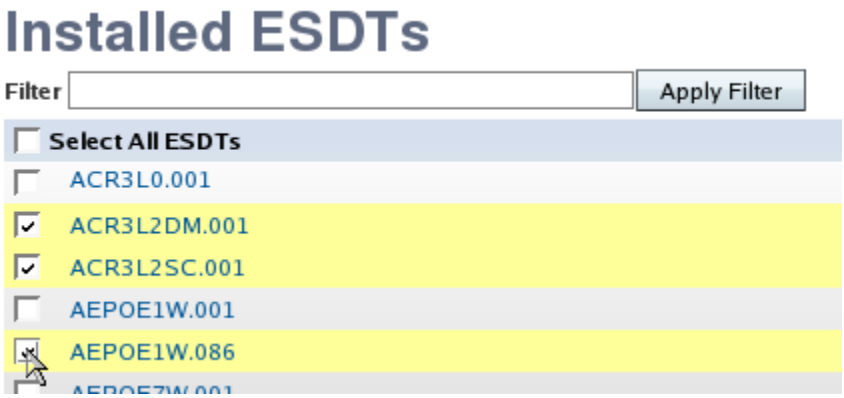


Figure 4.7.9-4. Selecting a list of ESDTs

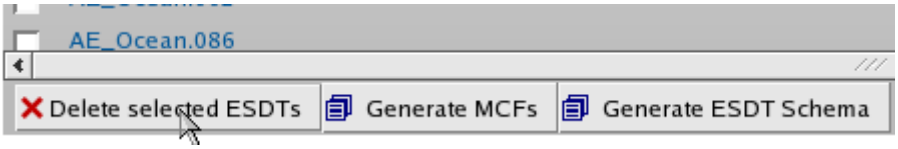


Figure 4.7.9-5. Delete selected ESDTs button

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

Once the delete action finishes, the result will be displayed at the top of the page as shown in Figure 4.7.9-6. Depending on the result, one to three possible result tables will be displayed as listed in Table 4.7.9-1.

Table 4.7.9-1. 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 (For non-ISO 19115 collections: descriptor, ESDT-specific schema, and MCF. For ISO 19115 collections: descriptor, ISO 19115 collection metadata XML, collection XPath, and granule XPath.) from the physical directory. If the ESDTs are in the updating state, the cleanup action will rollback the ESDT to its previous installed state.
ESDTs with Errors	The color of this table is red. This table displays any ESDTs that have processing errors from the most recent action submitted.
Successful ESDTs	The color of this table is green. This table displays a list of ESDTs that have completed successfully from the most recent action submitted.

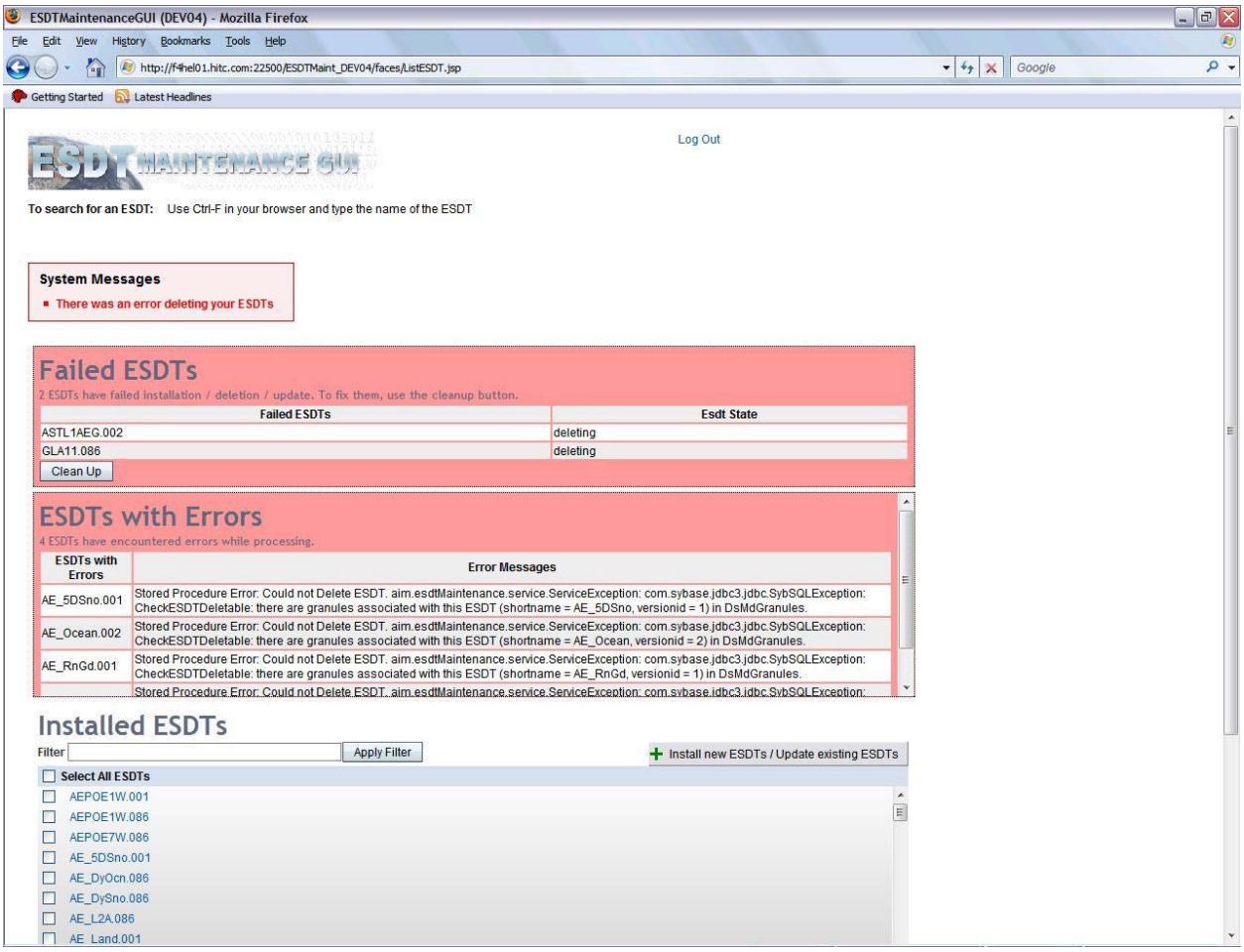


Figure 4.7.9-6. Delete ESDT Result Page

4.7.9.2.5 Generate MCFs for selected ESDTs

From the List ESDT Page, the operator can select a list of ESDTs and generate MCFs for them by clicking on the "Generate MCFs" button as shown in Figure 4.7.9-7. This option has no function for ESDTs that have ISO 19115 metadata.

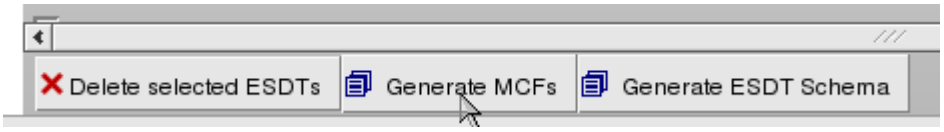


Figure 4.7.9-7. Generate MCF for Selected ESDT Button

The directory the MCFs will be generated to can be found in the EcDsAmESDTMaint.properties file. The result page for the generate MCF action is similar to the delete ESDT action as shown in Figure 4.7.9-8.

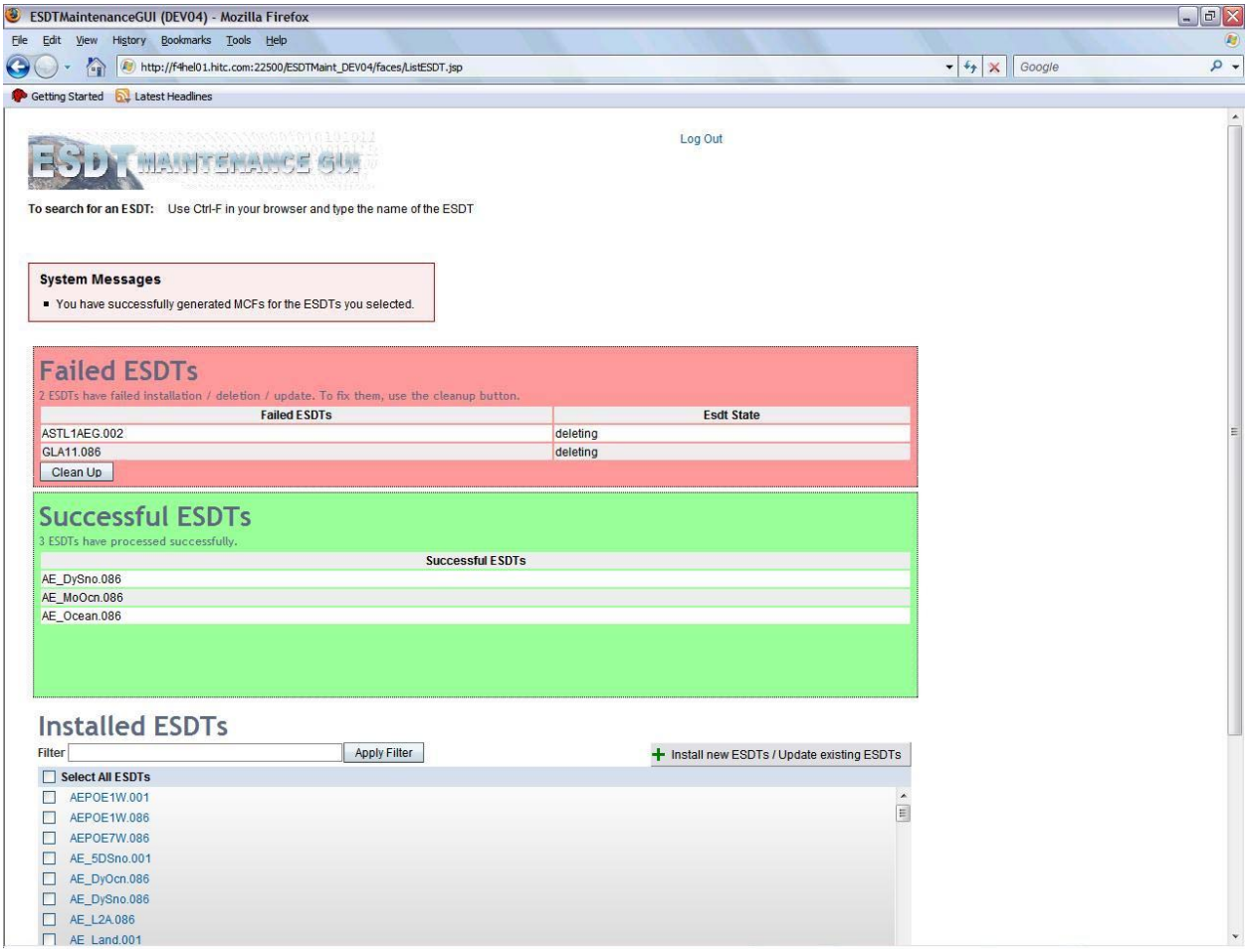


Figure 4.7.9-8. Generate MCF Result Page

4.7.9.2.6 Generate Schema for selected ESDTs

From the List ESDT Page, the operator can select a list of ESDTs and generate schema for them by clicking on the "Generate ESDT Schema" button as shown in Figure 4.7.9-9. This option has no function for ESDTs that have ISO 19115 metadata.

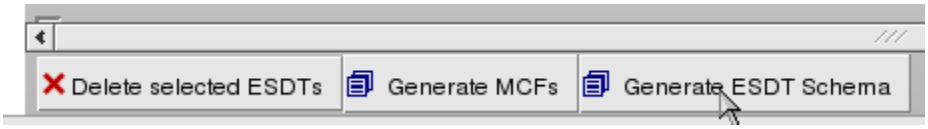


Figure 4.7.9-9. Generate Schema for Selected ESDT Button

The directory the schemas will be generated to can be found in the EcDsAmESDTMaint.properties file. The result page for the generate schema action is similar to the delete ESDT action as shown in Figure 4.7.9-10.

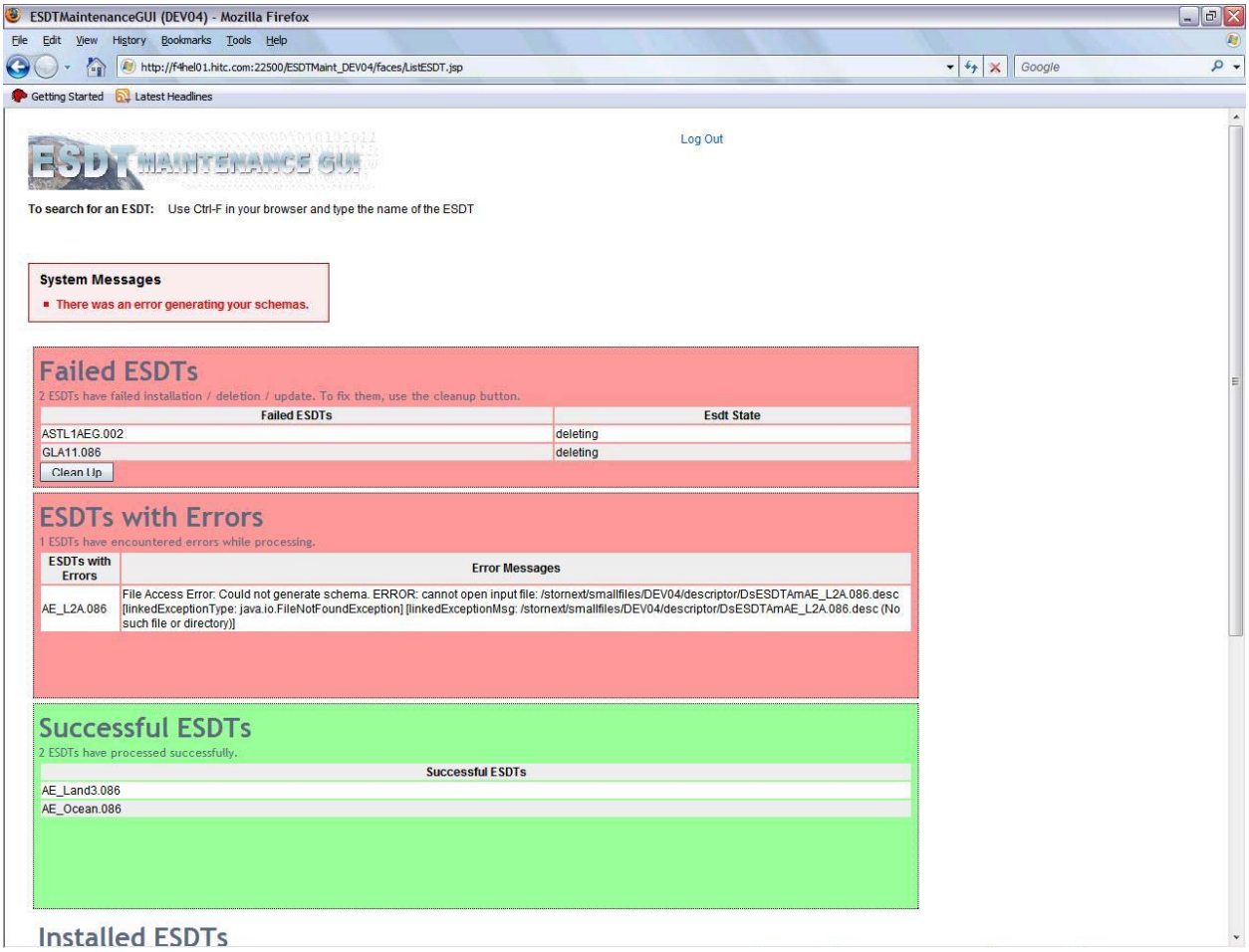


Figure 4.7.9-10. Generate Schema Result Page

4.7.9.3 ESDT Detail Page

From the List ESDT Page, the operator can click on the link for a particular ESDT to get more information about the installed ESDT. Once the link is clicked, the operator will be directed to the ESDT Detail Page as shown in Figures 4.7.9-11, 4.7.9-12, and 4.7.9-13. 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.

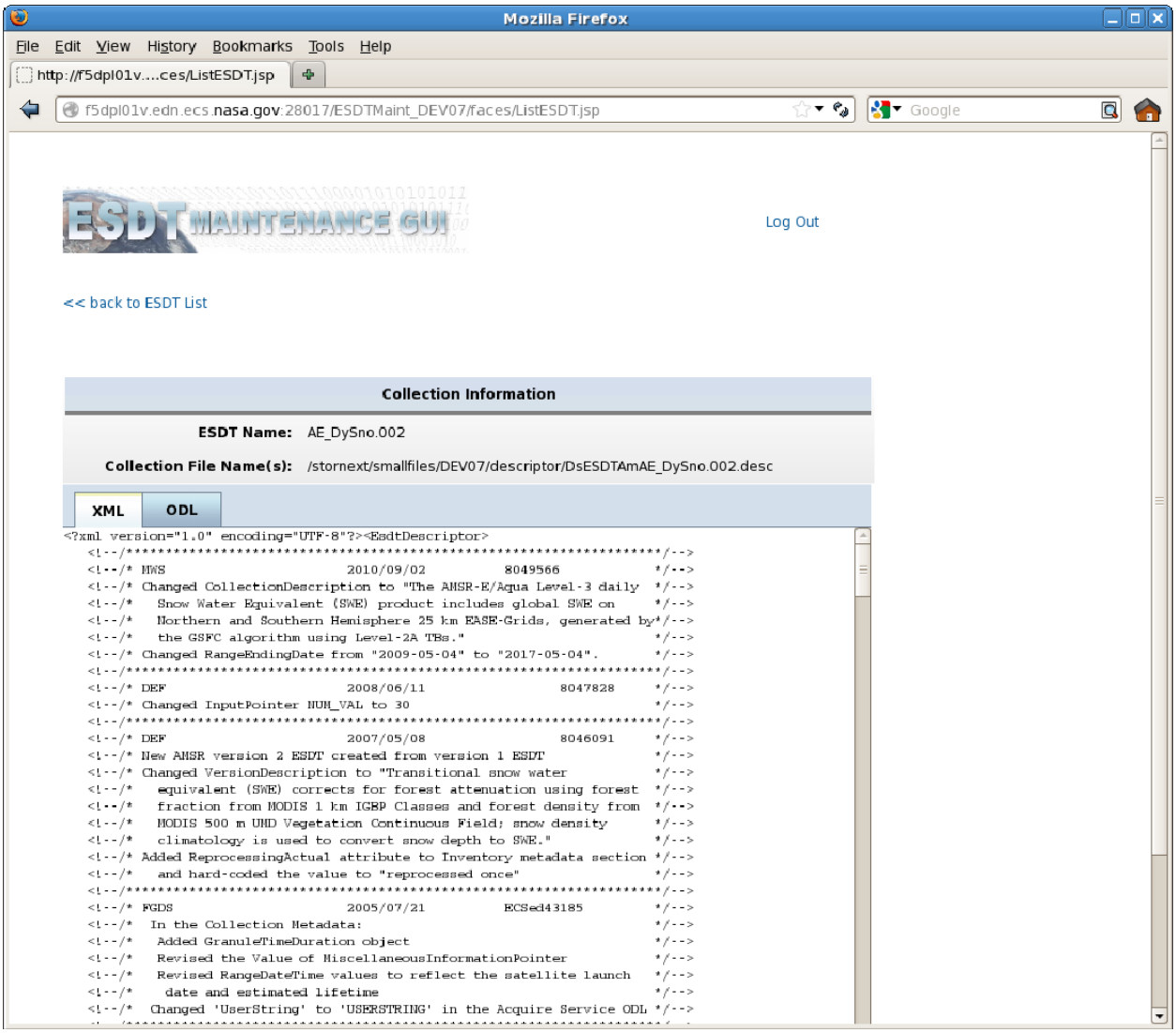


Figure 4.7.9-11. ESDT Detail Page (view XML)

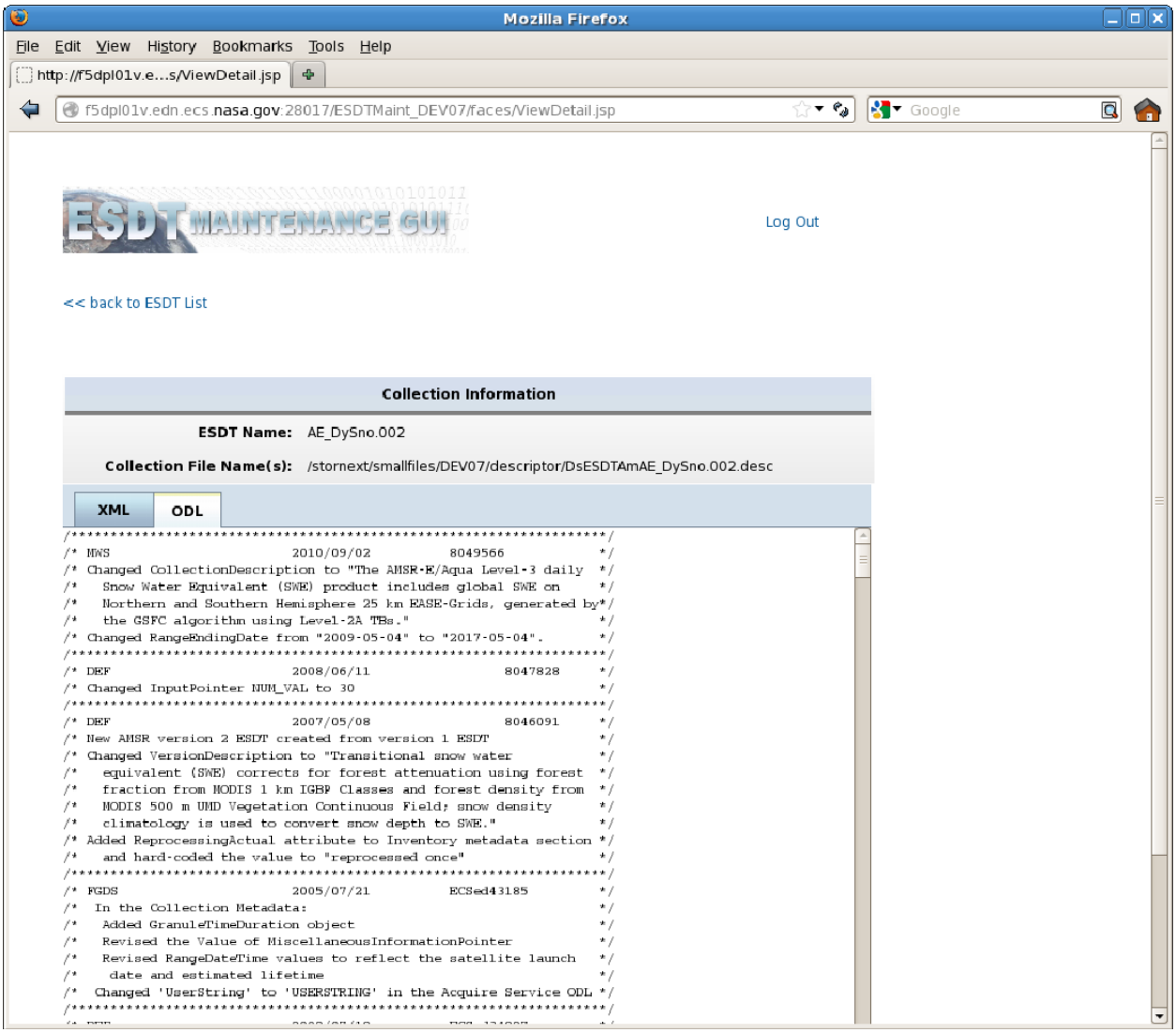


Figure 4.7.9-12. ESDT Detail Page (view ODL)

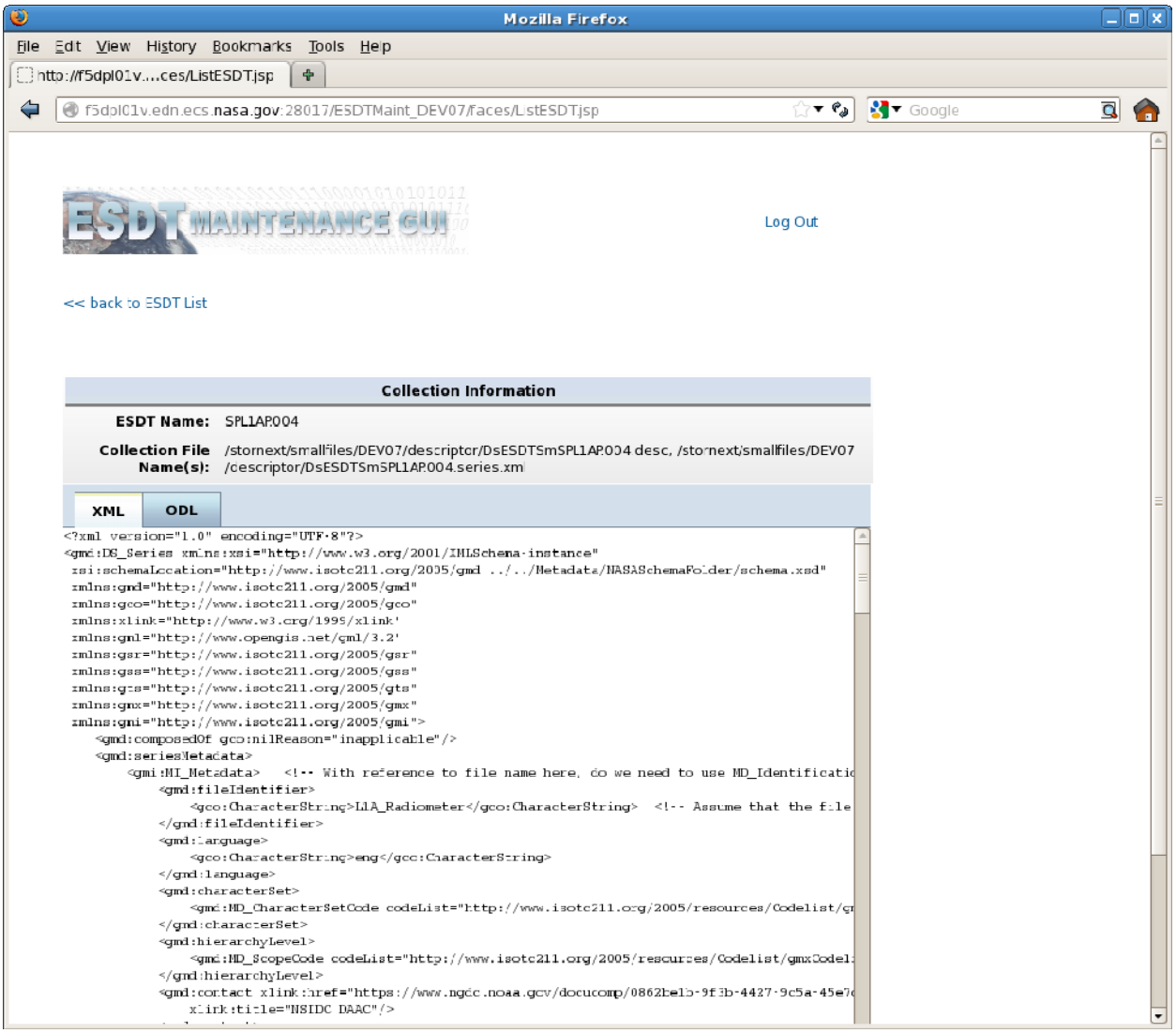


Figure 4.7.9-13. ESDT Detail Page (view XML (ISO 19115))

4.7.9.4 Install ESDT Page

This Install ESDT Page, as shown in Figure 4.7.9-14, appears when the operator clicks on the "Install new ESDTs / Update existing ESDTs" button from the List ESDT Page. From this page, the operator can install new or update existing ESDTs as well as cleanup failed ESDTs.

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

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

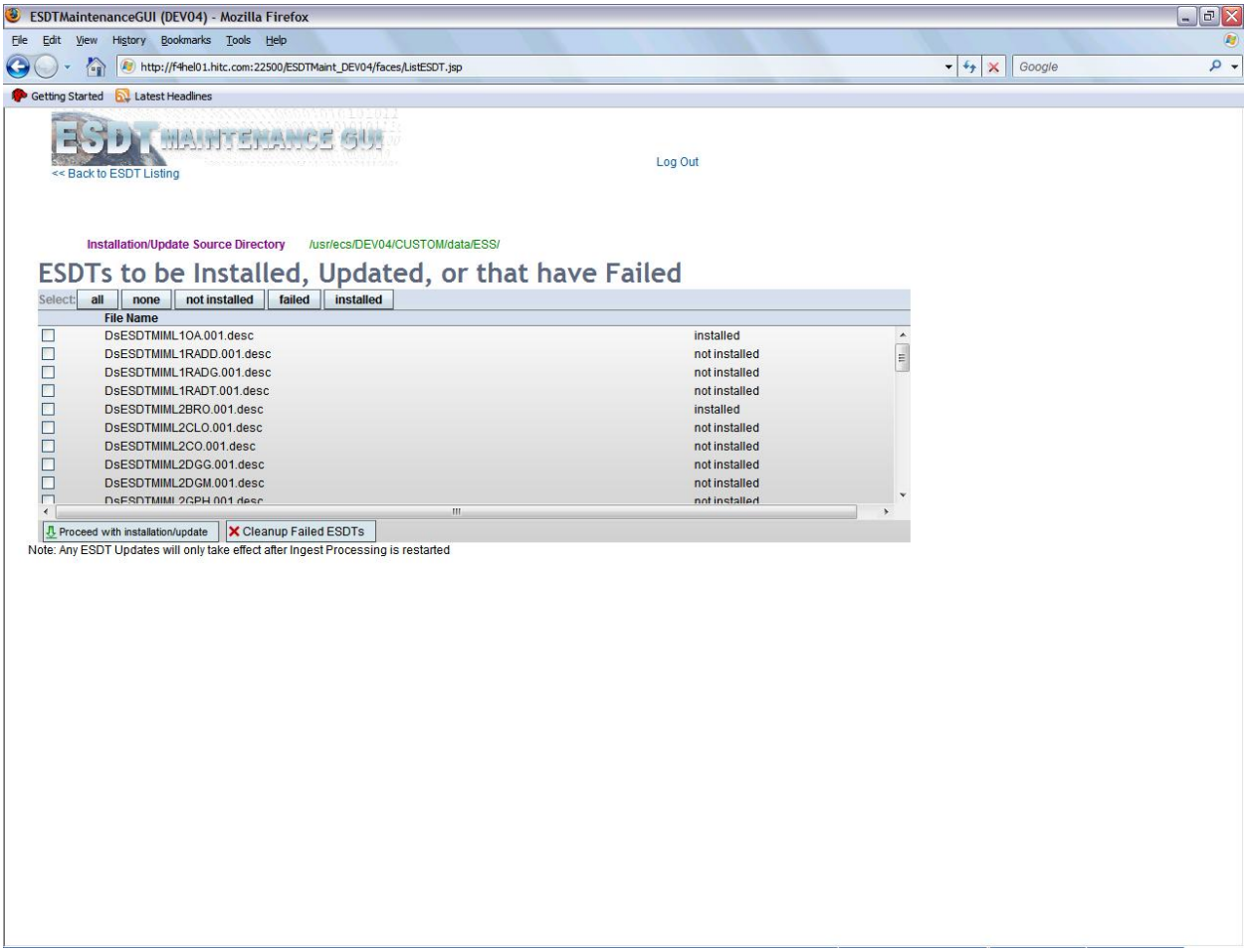


Figure 4.7.9-14. Install ESDT Page

4.7.9.4.1 Install/Update ESDTs

An operator performs installation or update on ESDTs by first selecting one or more ESDTs from the list as shown in Figure 4.7.9-15. The ESDT selected must be in the installed or not installed state.

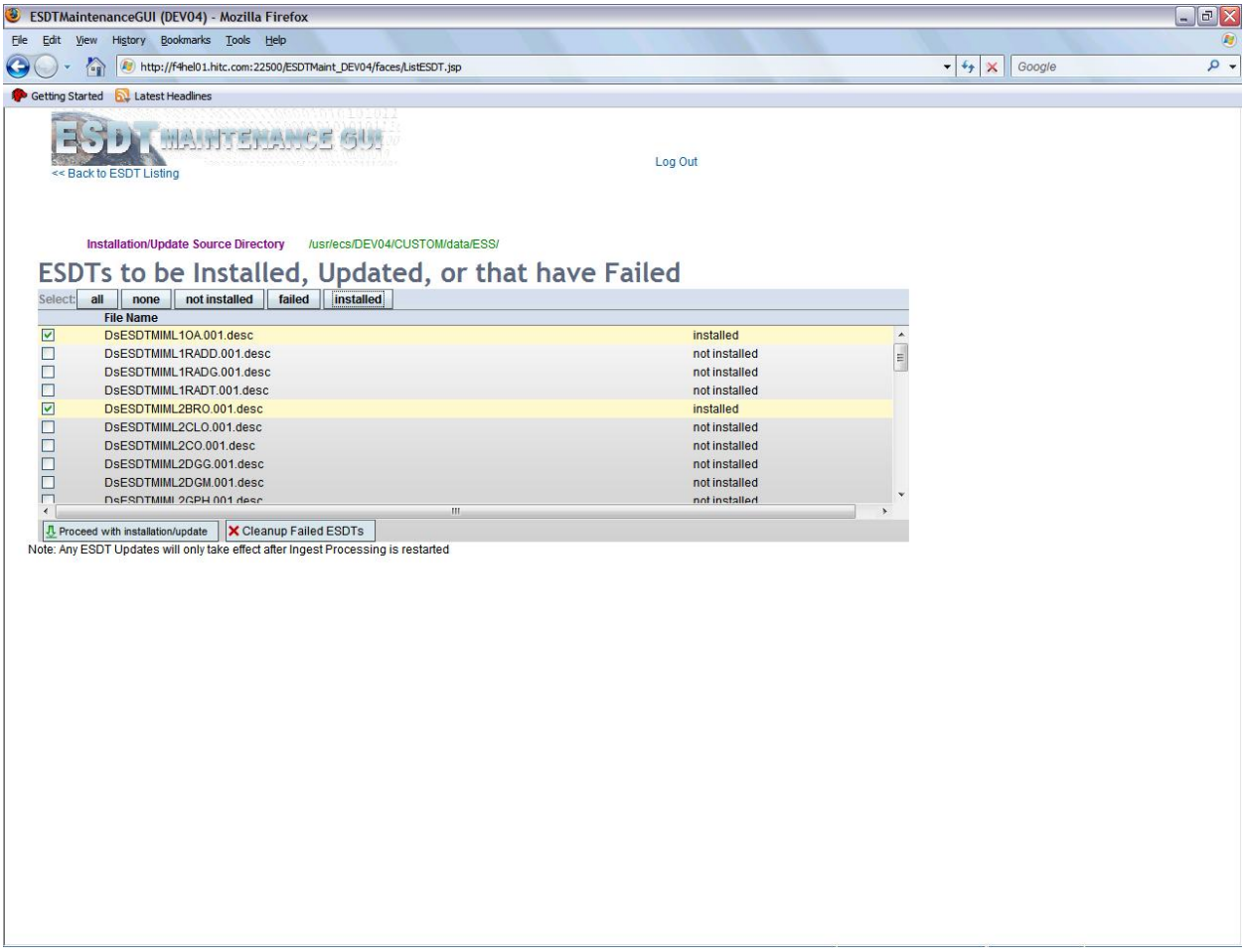


Figure 4.7.9-15. Install/Update ESDTs

Click on the "Proceed with installation/update" button to perform the installation or update action on the selected ESDTs as shown in Figure 4.7.9-16.

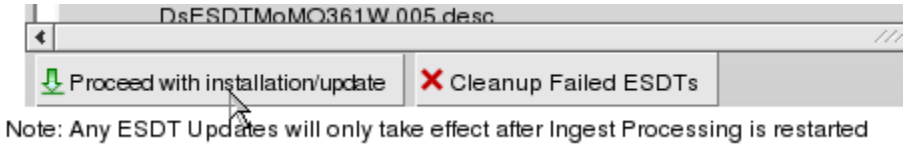


Figure 4.7.9-16. Install/Update ESDT button

An example of the result of the installation and update action is displayed in Figure 4.7.9-17.

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

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

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

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

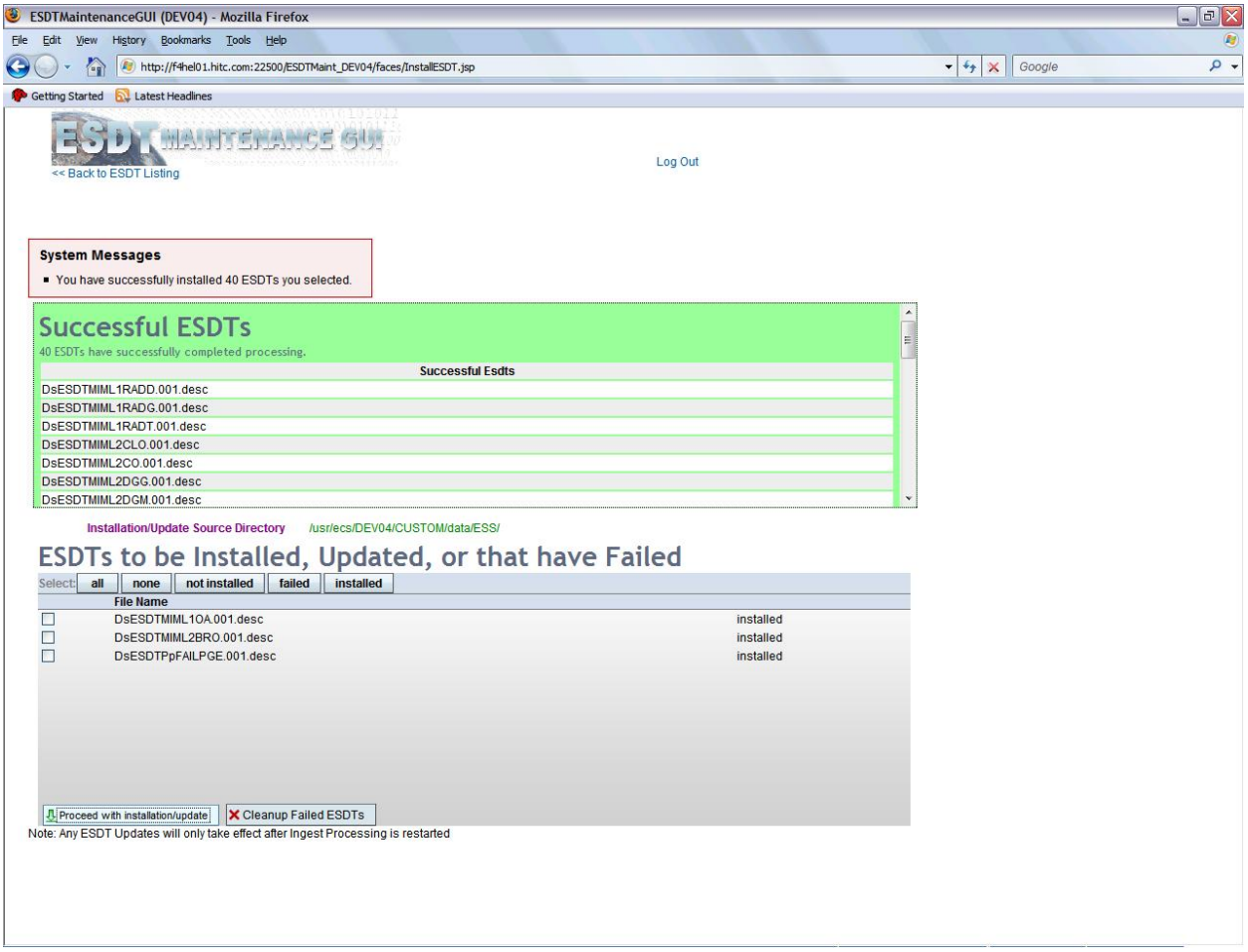


Figure 4.7.9-17. Install/Update ESDTs Result Page

4.7.9.4.2 Cleanup Failed ESDTs

An operator performs cleanup of failed ESDTs by first selecting one or more ESDTs from the list that are in the "failed" state. Then click on the "Cleanup Failed ESDTs" button to perform the cleanup action on the selected ESDTs as shown in Figure 4.7.9-18.

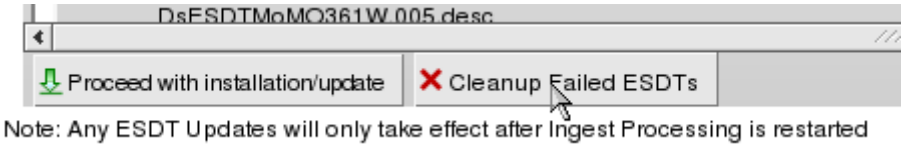


Figure 4.7.9-18. Cleanup Failed ESDTs button

An example of the result of the cleanup failed ESDTs action displayed in Figure 4.7.9-19.

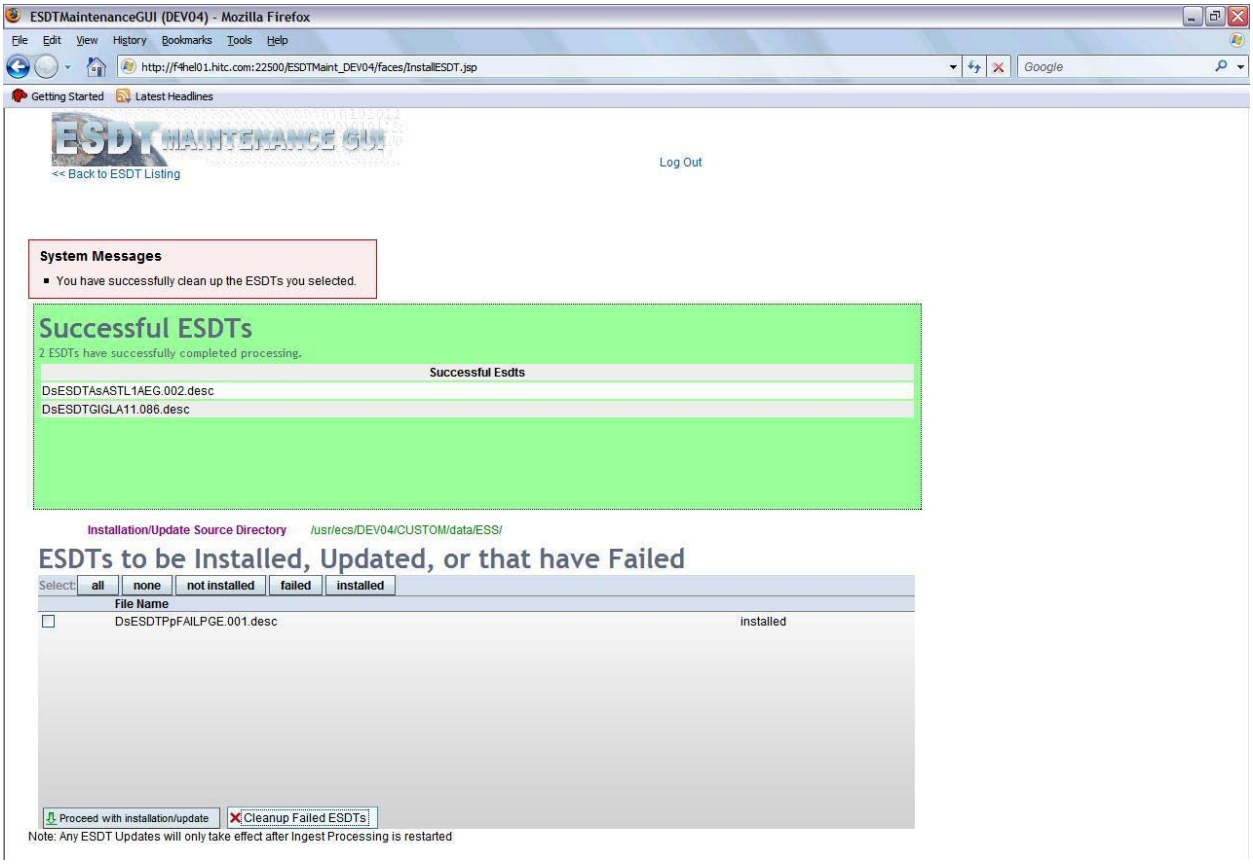


Figure 4.7.9-19. Cleanup Failed ESDTs Result Page

4.7.9.5 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.

4.7.10 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 AmGranule in the AIM RDBMS schema; or 2) by using the –dfa option of the EcDsBulkDelete utility, which sets the DeleteFromArchive flag in AmGranule in the AIM RDBMS schema. 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. Please note, Granule Deletion Phase II will skip granules that are in public Data Pool or on order. So, before the operator executes Phase II script, he/she should run Data Pool Unpublish Utility with –aim option to unpublish granules which are marked for deletion in the AIM database.

  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 AIM schema; b) the xml metadata files for the granule from the online and xml archives; and c) the science file(s) for the granule from the online and AIM/tape 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 AIM/tape archive, but will NOT remove the AIM schema 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), unpublish granules marked for deletion (Unpublish Utility with –aim option), 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, QA and MP 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, QA and MP granules altogether.

The sections below describe in detail the use of the EcDsBulkSearch, EcDsBulkDelete, EcDsBulkUndelete, and EcDsDeletionCleanup utilities. NOTE: as part of Release 8.2 all the

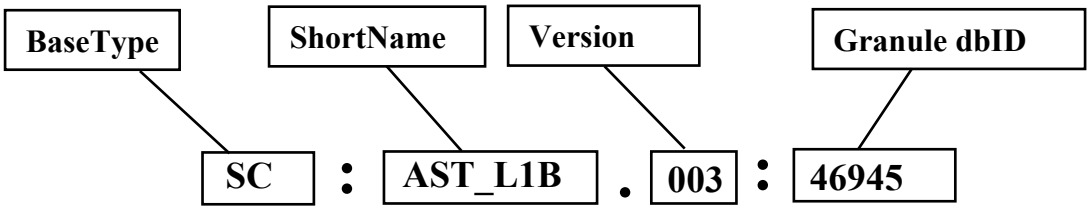
utilities use a common configuration file called EcDsAmGranuleDeletion.CFG containing parameters needed to connect to the RDBMS, the operator will still be prompted for the RDBMS password for the EcDsAmGranuleDeletion user.

Conventions of the command line used in the following sections

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

4.7.10.1 Generate a GeOID file

A GeOID, 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

MP – HDF 4 File Content Map granule

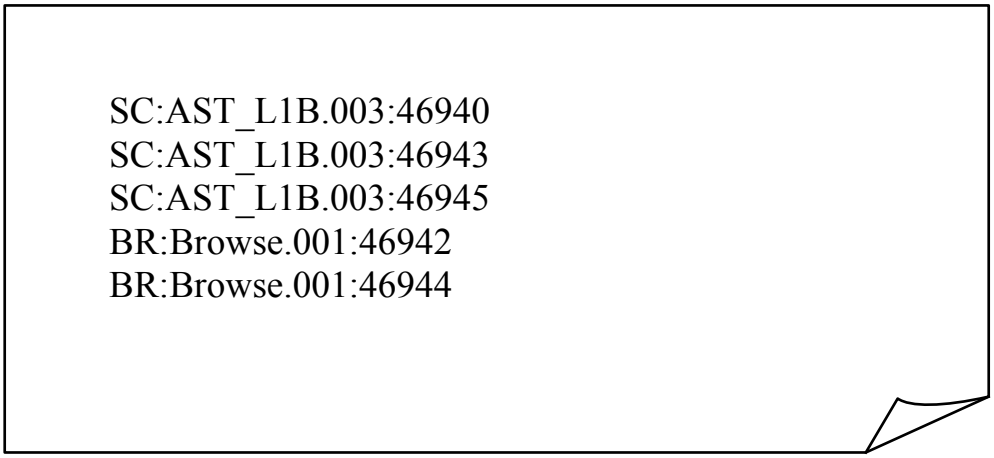
ShortName represents the data type of the granule such as “AST_L1B”.

Version 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 GeOID per line, as represented in the figure below:

GeOID File structure



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

4.7.10.1.1 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>] [**-password** <passwd>]

-geoidfile <path/filename>

–**mode** <MODE>Table 4.7.10-1 provides a description of the command line parameters for the EcDsBulkSearch utility.

Table 4.7.10-1. Command Line Parameters of the EcDsBulkSearch.pl (1 of 2)

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

Table 4.7.10-1. 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).
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 containing 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 "geoidfile" 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>

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

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

4.7.10.1.2 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>]**

4.7.10.1.3 Search a Limited Number of Granules

The search in 4.7.10.1.2 may return a large number of granules. A limited number of granules can be output by specifying option **–limit <n>** where **n** is an integer. The Bulk Search Utility sorts the granules according to the rules in 4.7.10.1.1 and returns the first n granules. The following search is the same as 4.7.10.1.2 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>] **-limit** <n>

4.7.10.1.4 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 **≥ begindate** AND EndingDateTime **≤ enddate**). The following command generates a GeOID file containing granules which meet the above search criteria, sorted in ascending order of dbID:

➤**EcDsBulkSearch.pl** **-geoidfile** <path/geoidfile_name>

-name <ShortName>

-version <VersionID>

-begindate <mm/dd/yyyy> [<hh:mm:ss>]

-enddate **<mm/dd/yyyy> [<hh:mm:ss>]4.7.10.1.5 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 ≥ **begindate**). 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>

-begindate **<mm/dd/yyyy> [<hh:mm:ss>] 4.7.10.1.6 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 <= **acquireend**). The following command generates a GeOID file containing granules which meet the above search criteria, sorted in ascending order of dbID:

➤**EcDsBulkSearch.pl** **-geoidfile** <path/geoidfile_name>

-name <ShortName>

-version <VersionID>

-acquireend <mm/dd/yyyy> [<hh:mm:ss>]

-user <db_userid>

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

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

➤**EcDsBulkSearch.pl** **-geoidfile** <path/geoidfile_name>

-name <ShortName>

-version <VersionID>

-enddate **<mm/dd/yyyy> [<hh:mm:ss>] 4.7.10.1.8 Search a Set of Granules for Which the insertTime is Within the Specified Date Time Range**

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

➤**EcDsBulkSearch.pl** **-geoidfile** <path/geoidfile_name>

-name <ShortName>

-version <VersionID>

-insertbegin **<mm/dd/yyyy> [<hh:mm:ss>]**

-insertend **<mm/dd/yyyy> [<hh:mm:ss>**

4.7.10.1.9 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** (insertTime ≥ **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>] 4.7.10.1.10 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** (insertTime ≤ **insertend**). The following command generates a GeOID file containing granules which meet the above search criteria, sorted in ascending order of dbID:

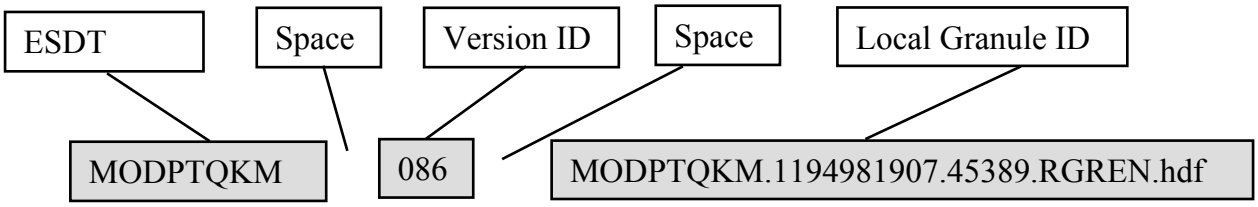
➤**EcDsBulkSearch.pl -geoidfile** <path/geoidfile_name>

-name <ShortName>

-version <VersionID>

-insertend **<mm/dd/yyyy> [<hh:mm:ss>] 4.7.10.1.11 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** :




```
MODPTQKM  086  MODPTQKM.1194981907.45389.RGREN.hdf
MODPTQKM  086  MODPTQKM.1194981908.45393.RGREN.hdf
MODPTQKM  086  MODPTQKM.1194981909.45395.RGREN.hdf
MODPTQKM  086  MODPTQKM.1194981910.45397.RGREN.hdf
MODPTQKM  086  MODPTQKM.1194981912.45399.RGREN.hdf
```

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>4.7.10.1.12 Search for Granules Marked for Deletion with –physical option**

The EcDsBulkSearch utility also allows Operations staff to search for granules that have already been marked for deletion. In this example, the EcDsBulkSearch utility returns a GeOID file of granules that have been marked for deletion using the –physical option of the EcDsBulkDelete utility (see 4.7.10.2.1), i.e., granules where the DeleteEffectiveDate is set in the AmGranule table in the inventory database:

➢**EcDsBulkSearch.pl -physical** **-geoidfile** <path/geoidfile_name>

4.7.10.1.13 Search for Granules Marked for Deletion with –dfa option

In this example, the EcDsBulkSearch utility returns a GeOID file of granules that have been marked for deletion using the –dfa option of the EcDsBulkDelete utility (see 4.7.10.2.2), i.e., granules where the DeleteFromArchive flag is set in the AmGranule table in the inventory database:

➢**EcDsBulkSearch.pl -dfa** **-geoidfile** <path/geoidfile_name>

4.7.10.2 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**] [**-password** <passwd>]
-geoidfile <path/filename>
–**mode** <MODE> [**-log** <log_file_name>]

Table 4.7.10-2 provides a description of the parameters used in executing the EcDsBulkDelete.pl script.

Table 4.7.10-2. 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, MP) 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.
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>

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

4.7.10.2.1 Mark Granules for physical deletion of science files and metadata

TheEcDsBulkDelete 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>

[**-log** <log_file_name.log>]

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

4.7.10.2.2 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>

[**-log** <log_file_name.log>]

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

4.7.10.3 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, MP) 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**]

[**-password** <passwd>]

-geoidfile <path/filename>

–**mode** <MODE>

[-**log** <log_file_name>]

Table 4.7.10-3 provides a description of the parameters used in executing the EcDsBulkUndelete.pl script.

Table 4.7.10-3. Command Line Parameters for EcDsBulkUndelete.pl

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, MP) 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.
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>

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

4.7.10.3.1 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 AmGranule table in the inventory database):

➢ **EcDsBulkUndelete.pl -physical**

[**-noassoc**]

-geoidfile <path/geoidfile_name>

[**-log** <log_file_name.log>]

4.7.10.3.2 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 AmGranule:

➢ **EcDsBulkUndelete.pl -dfa**

-geoidfile <path/geoidfile_name>

[**-log** <log_file_name.log>]

4.7.10.4 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 from both the Online Archive and the backup/tape Archive.

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 archives. If the granules were marked for deletion with the –physical option, and the –noassoc option was not specified, all files for associated Browse, PH, QA and MP granules will also be removed from the archives.

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 archives, as well.

When the EcDsDeletionCleanup utility is executed, it will check if there is any unfinished work from the 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**

–**mode** <MODE>

[-**batch** <number>]

[-**grbatch** <number>]

[-**phbatch** <number>]

[-**log** <log_file_name>]

[-xmlbatch <number>]

[-databatch <number>]

[-logbatch <number>]

Table 4.7.10-4 provides a description of the parameters used in executing the EcDsDeletionCleanup.pl script.

Table 4.7.10-4. Command Line Parameters for EcDsDeletionCleanup.pl (1 of 3)

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

Table 4.7.10-4. Command Line Parameters for EcDsDeletionCleanup.pl (2 of 3)

Parameter Name	Mandatory	Description
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).
grbatch	No	This is a tuning parameter. It represents the batch size used for physical granule file cleanup. This parameter may be omitted if the environment variable BATCH_SIZE_GRANULE is set. (Note: the environment variable BATCH_SIZE_GRANULE applies to both –batch and –grbatch). If the environment variable BATCH_SIZE_GRANULE is set, and -**grbatch** <number> is also specified, the value from command line parameter –**grbatch** will be used. If neither the environment variable BATCH_SIZE_GRANULE is set nor –**grbatch** is specified, the user will be prompted at runtime to enter a value (a value of 10000 is suggested by the prompt text).
phbatch	No	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. If the environment variable BATCH_SIZE_PH is set and -**phbatch** <number> is also specified, the value from command line parameter –**phbatch** will be used. If neither the environment variable BATCH_SIZE_PH is set nor –**phbatch** is specified, the user will be prompted at runtime to enter a value (a value of 5 is suggested by the prompt text).
xmlbatch	No	This is a tuning parameter. It represents the batch size for processing the deletion of XML files from the XML Archive. The utility will iterate through the deletion of XML files retrieving and deleting “xmlbatch” files per iteration. This controls memory growth of the utility when processing a large number of granules. If this parameter is not specified, the utility will prompt for a value (suggesting a size of 1000).

Table 4.7.10-4. Command Line Parameters for EcDsDeletionCleanup.pl (3 of 3)

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

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

setenv MODE <MODE>

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

4.7.10.4.1 Run a New Physical Cleanup

Command for starting a new run:

➤**EcDsDeletionCleanup.pl** [**-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 there are none, the utility will terminate.

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

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

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

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

3. Quit

Select 1, 2 or 3: _

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

1 – Only cleanup granules whose deletion date falls into a single day specified by the lag time;

2 – Cleanup all granules whose effective deletion date is older than the date specified by the lag time;

3 – Nothing to cleanup, exit.

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

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

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

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

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

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

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

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

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

2. Select all datatypes for deletion

3. Quit

Select 1, 2 or 3: _

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

1 – Cleanup granules marked for deletion which have an ESDT shortname and versionid in the input file. The input file lists one ESDT per line, in <shortname.versionid> format. A wildcard * may be used as part of the ESDT shortname;

2 – Cleanup all granules marked for deletion, regardless of their ESDT.version;

3 – Nothing to cleanup, exit.

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

4.7.10.4.2 Rerun unfinished Physical Cleanup

The EcDsDeletionCleanup utility always checks if there were any unprocessed granule(s) left over from a previous unfinished run(s). If so, when the EcDsDeletionCleanup utility is invoked normally (see 4.7.10.4.1), leftover information will be displayed and logged, and a menu will be displayed for the user to select how to run the cleanup:

Previous run was not completed, you can choose to:

1. Rerun unfinished run only

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

3. Quit

Select 1, 2 or 3:

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

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

4.7.11 DataPool Checksum Verification Utility

The DataPool Checksum Verification utility (DPCV) provides a mechanism by which the ECS Operations Staff can perform checksum verification for files in the Data Pool. It can be scheduled and run as a background process to proactively verify the integrity of files in the Data Pool. For example, the utility could be set up on one host as a background process that would verify the checksum of all files for a given collection that have not had their checksum verified recently. This is accomplished by specifying a checksum verification option based on time elapsed since the last time checksum was verified. On a separate ECS host, another process could be set up to run in the background, checking files from different collection every 10 seconds based on the configuration parameter SECONDS_BETWEEN_CHECKSUMS. It is important for the DAACs to set up these background processes in a way that will balance the load between hosts and adequately keep the last checksum verification time up to date for distributed collections. This will reduce the load on OMS and cause distribution of data to occur in a more timely and accurate fashion. If necessary, the utility can be run on-demand by the DAAC operator to verify checksum values for a particular set of files.

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

4.7.11.1 Using the DataPool Checksum Verification Utility

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

EcDlDPCVStart <mode> <command line parameters>

There are nine command line parameters that may be used. Table 4.7.11-1 provides a description of those parameters.

Table 4.7.11-1. Command Line Parameter

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

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

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

4.7.11.1.1 DataPool Checksum Verification Utility Command Line Examples

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

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

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

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

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

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

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

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

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

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

3. For a "file" run:

EcDlDpcvStart OPS –file dplgranuleids.txt

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

4. For a cron run:

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

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

4.7.11.2 DataPool Checksum Verification Utility Configuration File

The DataPool Checksum Verification utility uses a configuration file: EcDlDpcv.properties, located in /usr/ecs/<mode>/CUSTOM/cfg directory. The configuration parameters are stored in a PARAMETER = VALUE format with each parameter/value pair as a separate line entry in the file. Table 4.7.11-2 describes the configuration parameters.

Table 4.7.11-2. Configuration Parameters

Parameter Name	Description
PGM_ID	Program ID used for connecting to the Inventory database.
HOST_NAME	The host name where the program runs on.
DBUSERNAME	The user name for the RDBMS connection.
DBSERVER	The host name for the RDBMS server.
DBSERVERPORT	The port for the RDBMS supporting the mode
DBNAME	The DB name within the RDBMS
DBSUBSYSTEM	The RDBMS schema/subsystem (aim) hosting this utility
JDBC_DRIVER_CLASS	The Database jdbc driver class.
DB_RETRIES	The number of times the utility attempts to connect to the database before exiting. The recommended default is 5.
DB_SLEEPSECONDS	The number of seconds the utility waits (‘sleep’) between connection attempts. The recommended default is 10.
SQL_TIMEOUT_SECONDS	The number of seconds to timeout a db operation.
DPCV_EXPIRATION_TIME	The number of hours the utility uses to mark an un-finished process as expired.
DPCV_HISTORY_RETENTION_TIME	The number of days the utility uses to cleanup old DPCV run record in database.
SECONDS_BETWEEN_CHECKSUMS	The number of seconds between checksum operations
NUM_CHECKSUM_RETRIES	The number of retries on checksum failures
HOST_NAME	The host name where DPCV is running
VALIDATION_OUTPUT_DIR	The directory where the validation output files will be saved.

4.7.11.3 DataPool Checksum Verification Utility Main Screen

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

4.7.11.4 Required Operating Environment

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

4.7.11.5 Databases

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

Table 4.7.11-3. Product Dependencies

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

4.7.11.6 Special Constraints

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

4.7.11.7 Outputs

DPCV generates a log file for each run (See Section 4.7.11.10 for details). Besides the log file generated for each run, DPCV will also produce a phantom report and a checksum mismatch report if necessary under directory that is configured as VALIDATION_OUTPUT_DIR in the configuration file. The phantom report lists the DPL granule ids of the phantom granules found. The checksum mismatch lists the DPL granule ids of the granules that have checksum mismatch failures. The naming convention for the phantom report is: Phantom_dplids_RepairByRestoreOlaFromTape.<pid>.<timestamp>. The naming convention for the checksum mismatch report is: ChecksumMismatch_dplids_RepairByRestoreOlaFromTape.<pid>.<timestamp>.

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

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

4.7.11.8 Event and Error Messages

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

4.7.11.9 Reports

See outputs above.

4.7.11.10 Logs

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

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

4.7.11.11 Recovery

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

4.7.11.12 Database Error Handling

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

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

4.7.12 Inventory Validation Tool

The Inventory Validation Tool provides the EED Operations Staff with a command-line interface to verify the consistency of the ECS archive..

4.7.12.1 Using the Inventory Validation Tool

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

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

There are various command line parameters that are used in combination with each other. Table 4.7.12-1 provides a description of these parameters.

Table 4.7.12-1. Command Line Parameters of the Inventory Validation Tool

Parameter Name	Description
<mode>	Mandatory. Specifies the mode of operation. This must be the first parameter passed, and it must be a valid, existing Inventory mode with a format of OPS or TS[1-4] or DEV0[1-9].
-outputDir	Optional. Specifies the relative path under the base directory defined under parameter VALIDATION_OUTPUT_DIR in the configuration file EcDlInventoryValidationTool.CFG. Note: the base directory has to exist; The relative directory (only one level down) will be created if it doesn't exist. This is where all the output files reside. If the relative path is not provided, the output files will go to the base directory.
-suppressLDeleted	No longer used, will be ignored if provided.
-suppressDFAed	No longer used, will be ignored if provided.

The <mode> must be the first parameter. A command line input error results in a 'usage' display. The reason why the input was incorrect is also displayed.

4.7.12.2 Inventory Validation Tool Commands

Below is an example for invoking this tool:

1. **`EcDlInventoryValidationTool.pl DEV04 -outputDir inventory`**

Output files will be written to an 'inventory' subdirectory under the VALIDATION_OUTPUT_DIR directory.

4.7.12.3 Required Operating Environment

The Inventory Validation Tool will run on the same server as EcDlCleanupFilesOnDisk.pl.

4.7.12.4 Interfaces and Data Types

Table 4.7.12-2 lists the supporting products that this tool depends upon in order to function properly.

Table 4.7.12-2. Interface Protocols

Product Dependency	Protocols Used	Comments
aim schema	SQL	Via RDBMS

4.7.12.5 Configuration File Format – EcDlInventoryValidationTool.CFG

The configuration file contains details about how to connect to the RDBMS server. Without this file, the tool can not run. Table 4.7.12-3 shows a sample configuration file.

Table 4.7.12-3. Sample Configuration File

Parameter Name	Description
DBUSERNAME	The user name for the RDBMS connection.
DBSERVER	The host name for the RDBMS server.
DBSERVERPORT	The port for the RDBMS supporting the mode
DBNAME	The DB name within the RDBMS
DBSUBSYSTEM	The RDBMS schema/subsystem (aim) hosting this utility
PGM_ID	Program ID used for connecting to the RDBMS.
NUM_RETRIES	The number of times that the utility attempts to connect to the database before exiting.
SLEEP_SEC	The number of seconds the utility waits between database connection attempts.
ROWCOUNT	No longer used, but still present in the configuration file.
SKIP_MISSINGDPL	No longer used, but still present in the configuration file.
VALIDATION_OUTPUT_DIR	The base directory where output files from the utility are written. The recommended value is /workingdata/emd/<mode>/lvu

4.7.12.6 Special Constraints

The Inventory Validation tool runs only if the AIM database is available.

4.7.12.7 Outputs

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

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

The names are:

- InventoryDiscrp_registered_but_not_archived_granuleids_RepairManually.<pid>.<yyyymmddhhmmss>: granules that are registered but not archived.

- InventoryDiscrp_should_be_public_granuleids_RepairByPublish.<pid>.<yyyymmddhhmmss>: granules that are in the hidden Data Pool but are in public collections and are eligible to be public, as well as granules which are in the AIM database but not in the Data Pool database.
- InventoryDiscrp_should_be_hidden_granuleids_RepairByUnpublish.<pid>.<yyyymmddhhmmss>: granules that are in the public Data Pool but should be in the hidden Data Pool.
- InventoryDiscrp_should_be_hidden_browseids_RepairByUnPublish<pid>.<yyyymmddhhmmss>: browse granules that should not be in public Data Pool.
- InventoryDiscrp_versionnumber_filename_inconsistency_granuleids_RepairManually.<pid>.<yyyymmddhhmmss>:
  Inconsistent granule version identifier with granule file or link names.

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

However, publishing failures **can still occur** when there are several granule versions in the Data Pool of which none is public. **This can occur, for example,** if the public **version** somehow gets unpublished,. In these cases, **all** versions of the hidden granules are **considered** eligible to be published, however since they're replacements to each other some might fail (depending on the sequence of the publishing **operations**). Once the latest version is published, the remaining hidden versions will no longer **be considered candidates for publishing and subsequent runs will not include them** in their output file.

4.7.12.8 Event and Error Messages

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

4.7.12.9 Logs

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

4.7.12.10 Recovery

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

This page intentionally left blank.

4.7.13 Publish Utility

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

4.7.13.1 Using the Data Pool Publish Utility

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

EcDlPublishUtilityStart <MODE> -ecs [-file <file_name_path> | -collection <ShortName.VersionId> | -g <ecsId1>[ ][,]<ecsId2>] [-theme <themeName>] [-batchlabel <batchLabel>] [-maxnumconactions <num>] [-register] [-publish]

or

EcDlPublishUtilityStart <MODE> -nonecs -file <file_name_path> [-theme <themeName>] [-batchlabel <batchLabel>] [-maxnumconactions <num>]

Table 4.7.13-1 Provides a description of these parameters.

Table 4.7.13-1. Data Pool Publish Utility Command Line Parameters (1 of 2)

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

Table 4.7.13-1. Data Pool Publish Utility Command Line Parameters (2 of 2)

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

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

4.7.13.2 Data Pool Publish Utility usage examples

1. EcDlPublishStart OPS –ecs –file /home/cmshared/granuleIds.txt

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

2. EcDlPublishStart OPS –ecs –g 12345, 23456 -publish

Publish the two hidden granules whose ECS ids are given.

3. EcDlPublishStart OPS –ecs –collection MYD29P1D.004 –maxnumconactions 10

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

4. EcDlPublishStart OPS –ecs –g 12345 –theme "test"

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

5. EcDlPublishStart OPS –nonecs –file /home/cmshared/nonecs_xml.txt

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

4.7.13.3 Required Operating Environment

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

4.7.13.4 Interfaces and Data Types

Table 4.7.13-2 lists the supporting products that this tool depends upon in order to function properly.

Table 4.7.13-2. Interface Protocols

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

4.7.13.5 Input File Format

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

4.7.13.6 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 EcDlInsertUtility.properties.

4.7.13.7 Special Constraints

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

4.7.13.8 Outputs

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

4.7.13.9 Event and Error Messages

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

4.7.13.10 Reports

None

4.7.13.11 Logs

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

4.7.13.12 Recovery

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

4.7.13.13 Database Error Handling

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

4.7.14 Unpublish Utility

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

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

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

The Unpublish utility can also be used to unpublish granules which are marked for deletion in the AIM database (deleteEffectiveDate is set, or DFA flag is set to "Y" or "H") for example, as would occur after a run of the Granule Deletion Utility (EcDlBulkDelete.pl). Additionally, it can unpublish granules that were "restricted" in AIM by adding entries to the DsMdGranuleRestriction table.

4.7.14.1 Using the Data Pool Unpublish Utility

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

EcDlUnpublishStart.pl –mode <mode> [-f|-file <inputfile>]

| [-g|-granules <id1>,<id2>...]

| [-a|-aim -offset <offset # of hours> [-lagtime <lagtime # of hours]]

EcDlUnpublishStart.pl –help for instructions.

Table 4.7.14-1 provides a description of these parameters.

Table 4.7.14-1. Data Pool Unpublish Utility Command Line Parameters (1 of 2)

Parameter Name	Description
-file <*input_file*>	The file which contains a list of DPL granule ids for unpublish. Input_file specifies the full path and file name of the file.
-granules <*id1*>, <*id2*> …	DPL granule ids for unpublish.
-aim	Tells the unpublish utility to unpublish granules deleted from the AIM database. If this option is used, the –offset option should also be provided.

Table 4.7.14-1. Data Pool Unpublish Utility Command Line Parameters (2 of 2)

Parameter Name	Description
–offset <*#days*>	Specifies the past number of days for which to find deleted AIM granules. This option is only valid in conjunction with the –aim option. This option is useful when running from a cron; it allows the utility to not "overlap" the AIM events it processes. It can be thought of as a "start time" for retrieving events from AIM.
-lagtime <#hours>	Indicates that only granules marked for deletion in the Inventory DB prior to the specified number of hours should be unpublished. This option is only valid in conjunction with the –aim option. This option was added as part of Release 8.1 and is useful for manually delaying the unpublish action in the Online Archive until after ECHO has processed the "deletion" event from AIM. It can be thought of as the "end time" for retrieving events from AIM. This option replaces the "-exportonly" option previously provided in EcDlCleanupGranules.pl. The option should be used carefully so that it doesn't skip AIM events; it should NOT be used in a cron to process daily AIM events because the repeating –offset (start time) will cause the events to be skipped. For cron processing it is more reliable to simply run the EcDsBulkDelete.pl utility several hours earlier than the scheduled cron for this utility.
-help	Display instructions to run the utility.

An incorrect command line results in a 'usage' syntax display message. The log file for the utility is /usr/ecs/<mode>/CUSTOM/logs/EcDlUnpublish.log.PID.

4.7.14.2 Data Pool Unpublish Utility usage examples

1. EcDlUnpublishStart.pl –mode OPS –file /home/cmshared/granuleIds.txt

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

2. EcDlUnpublishStart.pl –mode OPS –granules 12345, 23456

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

3. EcDlUnpublishStart.pl –mode OPS –aim –offset 2 –lagtime 24

Unpublish granules deleted from the AIM database since the current time minus 2 days exclude the granules deleted for the last 24 hours.

4.7.14.3 Required Operating Environment

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

4.7.14.4 Interfaces and Data Types

Table 4.7.14-2 lists the supporting products that this tool depends upon in order to function properly.

Table 4.7.14-2. Interface Protocols

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

4.7.14.5 Input File Format

One granuleId per line.

4.7.14.6 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 EcDlInsertUtility.properties.

4.7.14.7 Special Constraints

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

4.7.14.8 Outputs

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

4.7.14.9 Event and Error Messages

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

4.7.14.10 Reports

None

4.7.14.11 Logs

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

4.7.14.12 Recovery

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

4.7.14.13 Database Error Handling

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

4.7.15 Archive Checksum Validation Utility

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

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

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

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

4.7.15.1 Using the Archive Checksum Validation Utility

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

EcDsAmAcvu.pl <mode> <command line parameters>

There are eight command line parameters that may be used. Table 4.7.15-1 provides a description of those parameters.

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

```
 -file <FILENAME>
 -modifytype <FILENAME>)
[-outputDir <DIRECTORY>]
```

Table 4.7.15-1. Command Line Parameter (1 of 2)

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

Table 4.7.15-2. Command Line Parameter (2 of 2)

Parameter Name	Required	Description
modifytype	Yes, if volumegroup, mediaid, granuleid, or file parameters are not present	Parameter to specify the name of an input file containing granuleids of granules, checksum origins, and checksum types whose files will have their checksum verified and modified with the new checksum origin and checksum type. Granuleid, origin, and type should be listed in the input file separated by commas. Additional Granuleids, origins, and types should be separated by newlines. Ex: <granuleid>,<origin>,<type> <granuleid>,<origin>,<type> Etc…
outputDir	No	Parameter to specify directory for error files under /workingdata/emd/<MODE>/Acvu

4.7.15.1.1 Archive Checksum Validation Utility Command Line Examples

1. For a "volumegroup" run:

   `EcDsAmAcvu.pl OPS –volumegroup /stornext/snfs1/OPS/MODIS`

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

   `EcDsAmAcvu.pl OPS –volumegroup`

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

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

2. For a "media id" run:

   `EcDsAmAcvu.pl OPS –mediaid VG7029`

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

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

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

3. For a "granuleid" run:

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

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

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

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

4. For a "file" run:

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

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

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

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

5. For a "modifytype" run:

```
EcDsAmAcvu.pl OPS –modifytype granules.txt
```

The Archive Checksum Validation Utility will perform checksumming for the files related to granules specified in granules.txt and update the Checksum origins and Checksum types.

4.7.15.2 Archive Checksum Validation Utility Configuration File

The Archive Checksum Validation utility uses a configuration file, EcDsAmAcvu.CFG, located in /usr/ecs/<mode>/CUSTOM/cfg directory. The configuration parameters are stored in a PARAMETER = VALUE format with each parameter/value pair as a separate line entry in the file. Table 4.7.15-2 describes the configuration parameters.

Table 4.7.15-2. Configuration Parameters

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

4.7.15.3 Archive Checksum Validation Utility Main Screen

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

4.7.15.4 Required Operating Environment

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

4.7.15.5 Databases

Table 4.7.15-3 lists the supporting products that this tool depends upon in order to function properly.

Table 4.7.15-3. Product Dependencies

Product Dependency	Protocols Used	Comments
aim schema	SQL	Via RDBMS

4.7.15.6 Special Constraints

The Archive Checksum Validation runs only if the RDBMS is running and if the database is available.

4.7.15.7 Outputs

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

4.7.15.8 Event and Error Messages

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

4.7.15.9 Reports

None

4.7.15.10 Logs

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

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

4.7.15.11 Recovery

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

4.7.15.12 RDBMS Error Handling

If a RDBMS error occurs, the actual RDBMS 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 RDBMS function. In the event of a RDBMS-sourced error, the utility will immediately stop running.

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

This page intentionally left blank.

4.7.16 XML Archive Corruption Check Utility (EcDsAmXcu.pl)

The XML Archive Corruption Check Utility (EcDsAmXcu.pl) provides a mechanism by which the ECS Operations Staff can periodically check for corruption in the XML Archive.

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

4.7.16.1 Using the XML Archive Corruption Check Utility (EcDsAmXcu.pl)

The XML Archive Corruption Check Utility (EcDsAmXcu.pl) should be started by the user cmshared (or similar). The XML Archive Corruption Check Utility (EcDsAmXcu.pl) is started by entering the following command:

EcDsAmXcu.pl <mode> <command line parameters>

There are seven command line parameters that may be used. Table 4.7.16-1 provides a description of those parameters.

Table 4.7.16-1. Command Line Parameter

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

4.7.16.1.1 XML Archive Corruption Check Utility (EcDsAmXcu.pl) Command Line Examples

1. For an "ESDT" run:

```
EcDsAmXcu.pl OPS –ESDT ALL
```

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

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

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

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

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

2. For a "granuleid" run:

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

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

3. For a "file" run:

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

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

4.7.16.2 XML Check Configuration File

The XML Archive Corruption Check Utility uses a configuration file, EcDsAmXcu.CFG, located in /usr/ecs/<mode>/CUSTOM/cfg directory. The configuration parameters are stored in a PARAMETER = VALUE format with each parameter/value pair as a separate line entry in the file. Table 4.7.16-2 describes the configuration parameters.

Table 4.7.16-2. Configuration Parameters (1 of 2)

Parameter Name	Value Description
DBUSERNAME	The user name for the RDBMS connection
DBSERVER	The host name for the RDBMS server.
DBSERVERPORT	The port for the RDBMS supporting the mode.

Table 4.7.16-2. Configuration Parameters (2 of 2)

Parameter Name	Value Description
DBNAME	The DB name within the RDBMS.
DBSUBSYSTEM	The RDBMS schema/subsystem (aim) hosting this utility.
PGM_ID	Program identifier used as seed to generate database password.
NUM_RETRIES	Number of times database operation will be attempted.
RETRY_INTERVAL	Number of seconds between retries.
MAX_CONCUR_CHECKS	Number of concurrent calls to xmllint that will be allowed.
VALIDATION_OUTPUT_DIR	The default directory to place error output files. The directory should be readable/writeable by cmshared. The suggested directory is /workingdata/emd/<MODE>/Xcu

4.7.16.3 XML Archive Corruption Check Utility Main Screen

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

4.7.16.4 Required Operating Environment

The XML Archive Corruption Check Utility will run on a Linux platform.

4.7.16.5 Databases

Table 4.7.16-3 lists the supporting products that this tool depends upon in order to function properly.

Table 4.7.16-3. Product Dependencies

Product Dependency	Protocols Used	Comments
aim schema	SQL	Via RDBMS

4.7.16.6 Special Constraints

The XML Archive Corruption Check Utility runs only if the RDBMS is running and if the database is available.

4.7.16.7 Outputs

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

4.7.16.8 Event and Error Messages

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

4.7.16.9 Reports

None

4.7.16.10 Logs

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

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

4.7.16.11 Recovery

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

4.7.16.12 RDBMS Error Handling

If a RDBMS error occurs, the actual RDBMS 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 RDBMS-sourced error, the utility will immediately stop running.

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

4.7.17 RestoreOlaFromTape

The ***RestoreOlaFromTape*** utility will repair individual granules or files that are lost or damaged in the on-line archive provided that the inventory entries of the corresponding granules are completely intact. This is because *RestoreOlaFromTape* does not have the capability to repair inventory (aim)schema entries. In all other cases, granules maybe restored using the Publish utility (e.g., if file entries or browse JPEG references are incorrect)..

The ***RestoreOlaFromTape*** utility shall:

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

In addition, the ***RestoreOlaFromTape*** utility shall:

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

Input is provided via an input file.

4.7.17.1 Running the RestoreOlaFromTape Utility

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

> **EcDlRestoreOlaFromTapeStart <MODE> -file** <file name and path with contents specified by –contents parameter> **-contents <granuleids | dplfiles> [-restoremisbr] [-restorelinks | -restorelinksonly] [-restorexmlonly] [-norecovery] [-email <usertoreceivestatusemail>]**

Table 4.7.17-1 provides a description of the above command line parameters.

Table 4.7.17-1. RestoreOlaFromTape Utility Parameters

Parameter Name	Description
-file <file name>	Name and path of the input file to be used by the utility
-contents <contents type>	The type of contents present in the file. Any of the following options are allowed: ⇒ granuleids: the input file contains the ECS granule IDs (GranuleIds) or browse IDs (BrowseIds) of the on-line archive granules that must be repaired ⇒ dplfiles: the input file contains the DPL filenames of the files that must be repaired. Browse files in JPG or HDF format are also accepted
[-restoremisbr]	Indicates if the utility should restore MISBR browse granule in the DPL. If the parameter is not set, the MISBR browse granule will not be restored. NOTE: This parameter will cause the utility to MISBR browse granule only when the configuration parameter MISR_SPECIAL_PROCESSING is set to "Y".
[-restorelinks | -restorelinksonly*]*	Indicates if the utility should restore browse/QA/PH symbolic linkage file for the given science granule. When '-restorelinks' is not provided in command line, only science granule metadata and data files are restored. When '-restorelinks' is specified in command line, both science granule files and browse/QA/PH symbolic links are restored. When '-restorelinksonly' is specified in command line, only browse/QA/PH symbolic links are restored. Note: A list of science granule ECS GranuleIds should be used to restore browse/QA/PH symbolic links.
[-restorexmlonly]	Indicates if the utility should restore granule metadata file only. When '-restorexmlonly' is not provided in command line, both granule metadata and data files are restored. When '-restorexmlonly' is specified in command line, only granule metadata file are restored.
[-norecovery]	Indicates if the utility should not recover from the last unsuccessful run. By default, the utility will disregard the current input file and read and complete the latest unsuccessful run (request) from the database. NOTE: if NO recovery is desired, the last unsuccessful run will be set to "Aborted" in the database.
[-email recipient_email_address]	Indicates the Email address of the user to receive the termination status of the utility. Multiple email addresses may be entered, separated by semicolons. If errors occurred, detail about the errors or how to retrieve the details will be present in the Email message.

4.7.17.2 Sample invocations of the RestoreOlaFromTape Utility

Below are some examples for invoking this utility:

1. **EcDlRestoreOlaFromTapeStart OPS -file** </home/john/granuleids.txt> **-contents granuleids –norecovery –email cmshared@ecs.nasa.gov**

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

2. **EcDlRestoreOlaFromTapeStart OPS -file** </home/john/dplfiles.txt> **-contents dplfiles –norecovery**

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

3. **EcDlRestoreOlaFromTapeStart OPS -file** </home/john/granuleids.txt> **-contents granuleids**

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

4.7.17.3 RestoreOlaFromTape Utility Main Screen

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

4.7.17.4 Required Operating Environment

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

4.7.17.5 Databases

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

Table 4.7.17-2. Interface Protocols

Product Dependency	Protocols Used	Comments
AIM schema	SQL	Located within the ecs database.
Postgres JDBC driver	JDBC	Requires proper install of Database JDBC driver.

If a PostgreSQL error occurs, you are most likely to see the actual PostgreSQL error string displayed on the screen and in the log. Some errors can be that the database server is

unavailable, the connection to the database was dropped, or there was an error executing a RDBMS function. In the event of a PostgreSQL-sourced error, the utility immediately stops running.

In the event that a connection to the RDBMS cannot be established, the utility will exit immediately.

4.7.17.6 Configuration File Format – RestoreOlaFromTape.properties

The configuration file contains vital details about how the utility will operate. The utility will exit immediately if a configuration file is not available. The file is a plain text ASCII file and has the following format as shown in Table 4.7.17-3:

Table 4.7.17-3. Individual Configuration Parameters (1 of 2)

Parameter Name	Description
PGM_ID	Sybase connectivity, the ID (10000030) is used to decrypt the DB password based on ECS standards
DBSERVER	The host name for the PostgreSQL data server
DBSERVERPORT	The port number for the PostgreSQL server on the specified host
DBUSERNAME	The user name (EcDlRestoreOlaFromTape) used to login to the PostgreSQL server
DBNAME	The name of the RDBMS database (ecs)
DBSUBSYSTEM	The name of the subsystem for this utility. This controls the RDBMS “schema path” to be used by the utility.
DB_RETRIES	Number of retries of a RETRYABLE DB operation (e.g. deadlock)
DB_SLEEPSECONDS	Number of sleep seconds between retries
SQL_TIMEOUT_SECONDS	Time in seconds that an SQL query will execute before timing out.
DB_BATCH_SIZE	The batch size for the database retrieve operations, its default value is 50
DEBUG_MESSAGES	(Y/N) indicates if detailed debugging information will be written to the log file.
CHECKSUM_SERVICE_HOSTS	The service hosts to be used for checksumming. The service hosts are configured in the format of <host_name_1>:<port_num>:<num_of_slots_1>, <host_name_2>:<port_num>:<num_of_slots_2>, …
CHECKSUM_TIMEOUT	Number of seconds before timeout a checksum operation
COPY_SERVICE_HOSTS	The service hosts to be used for copy operation. The service hosts are configured in the format of <host_name_1>:<port_num>:<num_of_slots_1>, <host_name_2>:<port_num>:<num_of_slots_2>, …

Table 4.7.17-3. Individual Configuration Parameters (2 of 2)

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

4.7.17.7 Special Constraints

The RestoreOlaFromTape Utility runs only if the ECS RDBMS server is running and the "ecs" database is online and if at least one checksum service host is available.

4.7.17.8 Outputs

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

4.7.17.9 Event and Error Messages

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

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

4.7.17.10 Reports

None

4.7.18 RestoreTapeFromOla

The ***RestoreTapeFromOla*** utility will repair individual files that are lost or corrupted on tape based on the primary file instance that is present in the on-line archive. The inventory entries of the corresponding granules must be completely intact. This is because the utility does not have capability to repair inventory database entries. The utility shall:

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

Notes:

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

Input is provided via an input file.

4.7.18.1 Running the RestoreTapeFromOla Utility

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

> **EcDlRestoreTapeFromOlaStart <mode> -file** <file name and path of input file whose contents type is specified by the –contents parameter> **-contents mediaids | granuleids | tapefiles [-volumegrouptype primary|backup] [-originalvolumegroup] [-removereadonlyfile] [-norecovery] [-email <usertoreceivestatusemail>]**

Table 4.7.18-1 provides a description of the above command line parameters.

Table 4.7.18-1. RestoreTapeFromOla Utility Parameters (1 of 3)

Parameter Name	Description
-file <file name>	Name and path of the input file to be used by the utility
-contents <contents type>	The type of contents present in the file. Any of the following options are allowed: ⇒ mediaids: the input file contains the media IDs (tape labels) of the tapes that were lost / damaged. ⇒ tapefiles: the input file contains the complete file names and paths of the tape files that must be repaired. ⇒ granuleids: the input file contains the ECS granule IDs (GranuleIds) of the tape granules that must be repaired

Table 4.7.18-1. RestoreTapeFromOla Utility Parameters (2 of 3)

Parameter Name	Description
[-removereadonlyfile]	Indicates that the utility should remove the original tape file from archive if the file cannot be restored to its original location. The utility always restores the file to the currently opened volume groups. Details below: If the option is not present, the utility will not try to remove the original tape file from archive. If the file cannot be restored to its original location, it will be restored in the currently opened volume group with the new file name, and the original file will remain on tape at the original location, without any corresponding AIM inventory record. The utility will not even try to remove the original file, regardless of the permissions on it. If the option is present, the utility will try to remove the original file. The file restored has the same name as original file. The utility will prompt the user to verify that the permissions to the RO volume group have been changed to RW if necessary: Have you changed the RO permissions to RW in the RO volume group affected by the restore (Y/N)? On Y the utility will proceed and: If the permissions to the affected files are RW, it will remove the original files that are affected. If the permissions to the affected files are RO, it will FAIL the restore of the granules involved. It is the responsibility of DAAC operations to inspect the log, identify the failed granules and rerun the utility after setting the correct RW permissions to the closed Volume Group. The reason for the failure is that if we would in fact restore the granule, the original file will remain on tape at the original location, and other application will find the bad copy. On N the utility will exit.
[-volumegrouptype <volumegroup type>]	The type of volume group the utility will restore files to. By default, the utility will restore tape files to both primary and backup volume groups. Any of the following options are allowed: ⇒ primary: the utility will restore files to primary volume groups. ⇒ backup: the utility will restore files to backup volume groups.
[-originalvolumegroup*]*	Indicates that the utility will restore files to original volume groups, the files are not renamed. No need to pass in -removereadonly file option if -originalvolumegroup is in the command line. By default, the utility will restore to currently open volume group.
[-norecovery]	Indicates that the utility should not recover from the last unsuccessful run. By default, the utility will disregard the current input file and read and complete the latest unsuccessful run (request) from the database. NOTE: if NO recovery is desired, the last unsuccessful run will be set to “Aborted” in the database.

Table 4.7.18-1. RestoreTapeFromOla Utility Parameters (3 of 3)

Parameter Name	Description
[-email recipient_email_address]	Indicates the Email address of the user to receive the termination status of the utility. Multiple email addresses may be specified, separated by semicolons. If errors occurred, detail about the errors or how to retrieve the details will be present in the Email message.

4.7.18.2 Sample invocations of the RestoreOlaFromTape Utility

Below are some examples for invoking this utility:

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

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

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

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

3. **EcDlRestoreTapeFromOlaStart OPS -file** </home/john/granuleids.txt> **-contents granuleids**

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

4.7.18.3 RestoreTapeFromOla Utility Main Screen

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

4.7.18.4 Required Operating Environment

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

4.7.18.5 Databases

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

Table 4.7.18-2. Interface Protocols

Product Dependency	Protocols Used	Comments
AIM schema	SQL	Located within the ecs database.
JDBC driver	JDBC	Requires proper install of JDBC driver

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

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

4.7.18.6 Configuration File Format – RestoreTapeFromOla.properties

The configuration file contains vital details about how the utility will operate. The utility will exit immediately if a configuration file is not available. The file is a plain text ASCII file and has the following format as shown in Table 4.7.18-3.

Table 4.7.18-3. Individual Configuration Parameters (1 of 2)

Parameter Name	Description
PGM_ID	Database connectivity, the ID (10000031) is used to decrypt the DB password based on ECS standards.
DBSERVER	Database connectivity, the host name for the database server.
DBSERVERPORT	Database connectivity, the port number for the database server on the specified host.
DBUSER	Database connectivity, the user name (EcDlRestoreTapeFromOla) used to login to the database server. AIM databases.
DBNAME	Database connectivity, the name of the RDBMS database (ecs)
DBSUBSYSTEM	Database connectivity, database schema (aim)
DB_POOL_SIZE	Database connectivity, the database connection pool size for the AIM.
JDBC_DRIVER_CLASS	Database connectivity, JDBC driver class.
DB_RETRIES	Number of retries of a RETRYABLE DB operation (e.g. deadlock).
DB_SLEEPSECONDS	Number of sleep seconds between retries.

Table 4.7.18-3. Individual Configuration Parameters (2 of 2)

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

4.7.18.7 Special Constraints

The RestoreTapeFromOla Utility runs only if the ECS RDBMS server is running and the "ecs" database is online and if at least one checksum service host is available.

4.7.18.8 Outputs

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

4.7.18.9 Event and Error Messages

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

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

4.7.18.10 Reports

None

4.7.19 EMS Dataset Extract Utility

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

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

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

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

4.7.19.1 Running the EMS Dataset Extract Utility

The EMS Dataset Extract utility is run from a CRON (see Section 4.7.19.2) or started by entering the following command from the /usr/ecs/<mode>/CUSTOM/utilities directory Linux command line:

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

Table 4.7.19-1 shows the parameters for the EMS Dataset Extract utility.

Table 4.7.19-1. Command Line Parameters of the EMS Dataset Extract Utility (1 of 2)

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

Table 4.7.19-1. Command Line Parameters of the EMS Dataset Extract Utility (2 of 2)

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

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

Table 4.7.19-2 describes datasets that are extracted and exported using the extraction utility:

Table 4.7.19-2. Datasets of the EMS Dataset Extract Utility

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

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

4.7.19.2 EMS Dataset Extract Utility Examples

Below are examples for invoking this tool:

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

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

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

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

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

3. **EcDbEMSdataExtractor.pl -m <mode> -s "start date" –e "end date"**

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

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

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

4.7.19.3 Required Operating Environment

The EMS Dataset Extract utility runs on the Linux platform.

4.7.19.4 Interfaces and Data Types

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

Table 4.7.19-3. Interface Protocols

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

4.7.19.5 Configuration File Format – EcDbEMSdataExtractor.CFG properties

The EcDbEMSdataExtractor.pl utility requires a configuration file. This configuration file, "EcDbEMSdataExtractor.CFG", is located in the /usr/ecs/<mode>/CUSTOM/cfg directory on the x4spl01 server. All edits of the EcDbEMSdataExtractor.CFG" file will be implemented using a Linux editor, such as "vi". The configuration file contains vital details about how to connect to the Postgres server and EMS host machine. Without this file, the tool cannot run. Table 4.7.19-4 describes the configuration parameters:

Table 4.7.19-4. Configuration Parameters (1 of 2)

Parameter Name	Recommended Value	Description
SERVER	<x4dbl01_srvr>	Enter Postgres server name e.g. x4dblnn_srvr.
PROVIDER	<DAAC NAME>	Enter provider name e.g. DAAC identifier.
EMSEXTRACTDIR	/usr/ecs/<mode>/CUSTOM/data/DSS	Enter EMS extraction directory location. This is the directory path specifying where data is extracted to when bcp'd out of database e.g. /usr/ecs/<mode>/CUSTOM/data/DSS.
EMSUSER	cmshared OR allmode	Enter user name to gain access to host represented by IPADDRESS - provided by EMS team.
PGMID	7000900	Static value. Same for all DAACs and Modes.

Table 4.7.19-4. Configuration Parameters (2 of 2)

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

4.7.19.6 Flat Files Naming Convention

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

1. **Timestamp**

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

2. **Root File Name**

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

3. **Extension**

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

The name of the data files is in the following format for all the Data Providers**:**
<YYYYMMDD>_<Provider>_<FileType>_<DataSource>.flt.rev<1-n>
Where: YYYY designates the 4 digit year for the time the Data Ingest Flat File was created
MM designates the 2 digit month, 01 through 12

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

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

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

4.7.19.7 Flat Files Updates

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

4.7.19.8 Flat File Format

A flat file contains the output data from DAAC operational database tables. Each line of the flat file represents one record information and each field of a record is ASCII text delimited by "|&|". The extracted flat files are located in the directory that is specified by the "EMSEXTRACTDIR" (Table 4.7.19-4) in configuration file.

1. Data Ingest Flat File

Table 4.7.19-5 describes the data ingest flat file layout information:

Table 4.7.19-5. Data Ingest Flat File Layout (1 of 2)

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

Table 4.7.19-5. Data Ingest Flat File Layout (2 of 2)

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

2. Data Archive Flat File

Table 4.7.19-6 describes the data archive flat file layout information:

Table 4.7.19-6. Data Archive Flat File Layout (1 of 2)

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

Table 4.7.19-6. Data Archive Flat File Layout (2 of 2)

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

3. Product Attribute Search Flat Files

Table 4.7.19-7 describes the product attribute search flat file layout information:

Table 4.7.19-7. Product Attribute Search Flat File Layout

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

4. Product Attribute Metadata Flat Files

Table 4.7.19-8 describes the product attribute metadata flat file layout information:

Table 4.7.19-8. Product Attribute Metadata Flat File Layout (1 of 2)

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

Table 4.7.19-8. Product Attribute Metadata Flat File Layout (2 of 2)

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

5. Electronic Media Distribution Flat Files

Table 4.7.19-9 describes the electronic/physical media distribution flat file layout information:

Table 4.7.19-9. Electronic Media Distribution Flat File Layout (1 of 2)

Field Name	Description	Table Name	Column Name	Column Datatype	Nulls
requestId	Identifier for a request	EcAcRequest	requestId	varchar(10)	No
orderId	Identifier for an order	EcAcRequest	orderId	varchar(10)	No
userId	Identification of user submitting a request for distribution; Ftp User corresponds to ftpAddress field in EcAcRequest.	EcAcOrder	userId	varchar(14)	No
orderSource	Origination of this order (MTMGW, SSS, DPLGUI, V0GW, etc).	EcAcOrder	orderSource	varchar(21)	Yes
orderType	The type of an order.	EcAcOrder	orderType	varchar(2)	Yes

Table 4.7.19-9. Electronic Media Distribution Flat File Layout (2 of 2)

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

Both FTP, HTTP, and Media methods generate the same flat files. The layouts generated from MSS/OMS database are the same as the layouts in Table 4.7.19-9.

Table 4.7.19-10 describes the physical media distribution flat file layout information generated from DataPool database for DistFTP and DistHTTP:

Table 4.7.19-10. Media Distribution Flat File Layout

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

4.7.19.9 Special Constraints

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

4.7.19.10 Outputs

Outputs will be printed to standard out if the –v flag is included with on the command line of the EMS Dataset Extract utility. Messages are also output to the EcDbEMSdataExtractor.log file (see Section 4.7.19.12). The DAAC Operations Staff should review the messages printed to the log file.

4.7.19.11 Event and Error Messages

Error messages will be displayed on standard out if the –v flag is included with the executed EMS command. Error messages will be logged in the EcDbEMSdataExtractor.log file (see Section 4.7.19.12). EMS Dataset Extract utility events are recorded in the MSS database EcEMSextractRecord table. Field descriptions for the EcEMSextractRecord table are described in Table 4.7.19-11.

Table 4.7.19-11. EcEMSextractRecord Table

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

4.7.19.12 Logs

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

4.7.19.13 Recovery

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

This page intentionally left blank.

4.7.20 DPL Checksum Server Utility

The Datapool Checksum Server Utility (DCSU) provides a mechanism by which the ECS Operations Staff can perform checksum modifications of server checksum types in the datapool. The utility allows the operator to specify which

4.7.20.1 Using the Datapool Checksum Server Utility

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

EcDsXXXxxx.pl <mode> <command line parameters>

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

```
EcDsXXADcsu.pl <MODE> [-calculate]
                      [-days <NUMBER OF DAYS>]
                      [-percent <PERCENT 1-100>]
                      [-norecovery]
                      (-volumegroup <VOLUME GROUPS> |
                       -mediaid <MEDIAIDS> |
                       -granuleid <GRANULEIDS> |
                       -file <FILENAME>
                       -modifytype <FILENAME>)
                      [-outputDir <DIRECTORY>]
```

Table 4.7.20-1. Command Line Parameter (1 of 2)

Parameter Name	Required	Description
calculate	No	Optional parameter to specify whether to calculate and store checksums for files found currently without checksums.
days	No	Optional parameter to specify days since last checked.
percent	No	Optional parameter to specify percentage of files to check.
norecovery	No	Optional parameter to specify not to recover from previous run.
volumegroup	Yes, if mediaid, granuleid, modifytype, or file parameters are not present	Parameter to specify volumegroups whose files will have their checksum verified. This is a comma separated list of one or more volume groups (no spaces). Volumegroups should be specified by full path name.

Table 4.7.20-2. Command Line Parameter (2 of 2)

Parameter Name	Required	Description
mediaid	Yes, if volumegroup, granuleid, modifytype or file parameters are not present	Parameter to specify mediaids whose files will have their checksum verified. This is a comma separated list of one or more mediaids (no spaces).
granuleid	Yes, if volumegroup, mediaid, modifytype or file parameters are not present	Parameter to specify granules whose files will have their checksum verified. This is a comma separated list of one or more granule ids (no spaces).
file	Yes, if volumegroup, mediaid, modifytype or granuleid parameters are not present	Parameter to specify the name of an input file containing granuleids of granules whose files will have their checksum verified. Granuleids should be listed in the input file separated by newlines.
modifytype	Yes, if volumegroup, mediaid, granuleid, or file parameters are not present	Parameter to specify the name of an input file containing granuleids of granules, checksum origins, and checksum types whose files will have their checksum verified and modified with the new checksum origin and checksum type. Granuleid, origin, and type should be listed in the input file separated by commas. Additional Granuleids, origins, and types should be separated by newlines. Ex: <granuleid>,<origin>,<type> <granuleid>,<origin>,<type> Etc…
outputDir	No	Parameter to specify directory for error files under /workingdata/emd/<MODE>/Acvu

4.7.20.1.1 Datapool Checksum Server Utility Command Line Examples

1. For a "volumegroup" run:

4.7.20.2 Datapool Checksum Server Utility Configuration File

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

Table 4.7.20-2. Configuration Parameters

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

4.7.20.3 Datapool Checksum Server Utility Main Screen

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

4.7.20.4 Required Operating Environment

The Datapool Checksum Server Utility will run on a Linux platform.

4.7.20.5 Databases

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

Table 4.7.20-3. Product Dependencies

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

4.7.20.6 Special Constraints

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

4.7.20.7 Outputs

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

4.7.20.8 Event and Error Messages

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

4.7.20.9 Reports

None

4.7.20.10 Logs

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

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

4.7.20.11 Recovery

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

4.7.20.12 Sybase Error Handling

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

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

This page intentionally left blank.

4.7.21 Duplicate Granule Reporting Tool

The Inventory Validation Tool provides the EED Operations Staff with a command-line interface to verify the consistency of the ECS archive..

4.7.21.1 Using the Inventory Validation Tool

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

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

There are various command line parameters that are used in combination with each other. Table 4.7.21-1 provides a description of these parameters.

Table 4.7.21-1. Command Line Parameters of the Inventory Validation Tool

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

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

4.7.21.2 Inventory Validation Tool Commands

Below is an example for invoking this tool:

1. **EcDlInventoryValidationTool.pl DEV04 -outputDir inventory -suppressLDeleted**

Output files will be written to an 'inventory' subdirectory under the VALIDATION_OUTPUT_DIR directory. The output result will not include any granules that are logically deleted (non-null deleteEffectiveDate) or deleted (DeleteFromArchive set to 'Y') in the ECS archive and missing in Data Pool.

4.7.21.3 Required Operating Environment

The Inventory Validation Tool will run on the same setver as EcDlCleanupFilesOnDisk.pl.

4.7.21.4 Interfaces and Data Types

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

Table 4.7.21-2. Interface Protocols

Product Dependency	Protocols Used	Comments
AIM database	SQL	Via Sybase server

4.7.21.5 Configuration File Format – EcDlInventoryValidationTool.CFG

The configuration file contains details about how to connect to the Sybase server. Without this file, the tool can not run. Table 4.7.21-3 shows a sample configuration file.

Table 4.7.21-3. Sample Configuration File

Parameter Name	Description
DBUSERNAME	The user name for the RDBMS connection.
DBSERVER	The host name for the RDBMS server.
DBSERVERPORT	The port for the RDBMS supporting the mode
DBNAME	The DB name within the RDBMS
DBSUBSYSTEM	The RDBMS schema/subsystem (aim) hosting this utility
DBUSERNAME	The user name for the RDBMS connection.
PGM_ID	Program ID used for connecting to the Sybase database.
NUM_RETRIES	The number of times that the utility attempts to connect to the database before exiting.
SLEEP_SEC	The number of seconds the utility waits between database connection attempts.
ROWCOUNT	Used to restrict the number of rows returned when running the Sybase query to retrieve discrepancies regarding granules which exist in the ECS Archive but not in the DataPool. This parameter may be used to prevent an out-of-memory error where there are large numbers of discrepancies between the AIM and DPL databases – e.g., before migration is complete. When set to 0, the rowcount is not restricted.
SKIP_MISSINGDPL	Set to Y or N. Allows the user to skip the checking that identifies granules which exist in ECS archive but not in Data Pool.
VALIDATION_OUTPUT_DIR	The base directory where output files from the utility are written. The recommended value is /workingdata/emd/<mode>/Ivu

4.7.21.6 Special Constraints

The Inventory Validation tool runs only if the ”aim” schema is available within the “ecs” database.

4.7.21.7 Outputs

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

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

The names are:

- InventoryDiscrp_registered_but_not_archived_granuleids_RepairManually.<pid>.<yyyymmddhhmmss>: granules that are registered but not archived.
- InventoryDiscrp_should_be_public_granuleids_RepairByPublish.<pid>.<yyyymmddhhmmss>: granules that are in the hidden Data Pool but are in public collections and are eligible to be public, as well as granules which are in the AIM database but not in the Data Pool database.
- InventoryDiscrp_should_be_hidden_granuleids_RepairByUnpublish.<pid>.<yyyymmddhhmmss>: granules that are in the public Data Pool but should be in the hidden Data Pool.
- InventoryDiscrp_should_be_hidden_browseids_RepairByUnPublish<pid>.<yyyymmddhhmmss>: browse granules that should not be in public Data Pool.
- InventoryDiscrp_versionnumber_filename_inconsistency_granuleids_RepairManually.<pid>.<yyyymmddhhmmss>:
  Inconsistent granule version identifier with granule file or link names.

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

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

4.7.21.8 Event and Error Messages

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

4.7.21.9 Logs

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

4.7.21.10 Recovery

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

4.7.22 Duplicate Granule Identification Utility

The Duplicate Granule Identification Utility provides the EED Operations staff with a command-line interface to recalculate and report duplicate granules.

There are three major activities involved: Rule changes, Recalculate, and Report. Only one instance of this utility can be run at any given time, so Recalculate and Report will never run concurrently. On the other hand, Rule changes can, and should, be able to happen anytime. Whenever the rule is changed for a given collection, all the existing duplicates become invalid, recalculation needs to be rerun, and the reporting needs to be rerun for the collection to include all the duplicates in the system for that collection. Therefore, the integrity of the duplicate granule data really depends on the operator's awareness of the timing of the three activities.

For instance, if a report is generated during or before the rule is changed, the operator needs to be aware of it and make decisions on what to do with the possible duplicates reported using the old rule. If the report is generated after the rule is changed, then no invalid duplicates will be reported. However, until the recalculation is rerun, the report will only include the duplicates related to the granules ingested after the rule is changed.

Here are some design decisions we've made:

1. We feel that locking mechanism doesn't apply in this situation, i.e. just because we implement locks and artificially delay the rule changes, doesn't make the existing old rules and the duplicate granules identified using the old rules valid.
2. The rule changing is an infrequent, well-orchestrated activity. Recalculating and reporting should also be planned activities. Therefore, there's no reason not to be able to run them in an orderly manner – Rule change, recalculate, report.
3. Regarding the request of prompting the operator that the rule is changed on a certain collection and the reporting on that collection will be skipped – First of all, reporting should be able to run as a cronjob, so no interaction is allowed. Secondly, it's not necessary to skip the whole collection because we only report duplicate granules identified by the most current rule, at the time of the reporting - all the invalid entries created during some border situation listed below will be excluded in the reporting.
4. It is important to clean up the invalid entries in the AmGranuleReplacement table before starting to recalculate. In theory, it's possible that the granules registered/recalculated right before the rule change time or rule logic change time could insert invalid duplicates both before and after the time reference point, using the old rule or old logic. But since there's always a delay between the time we run the recalculation and the time the rule changes(or the time the new logic is in place for the forced case), all the invalid entries will be removed using the following approach. After the recalculation, it's recommended to run the EcDlInventoryValidationTool.pl to identify the granules that are out of place(i.e. supposed to be in public but in hidden and vice versa) and publish/unpublish them accordingly, before running the reporting again to identify the duplicates and delete them.
    a. In the case when the recalculation is needed due to the rule changes, at the start of the recalculation, we'll remove all the entries for a given collection, that are associated

with the rules that are different from the rule indicated by DuplicateGranRuleNo in the AmCollection table, except for the entries associated with the file collision rule, and recalculate all the granules registered before the rule changing time in the AmDupGranRuleChangeEvent table.

b. In the case of a forced recalculation, which is not because of the rule changes, more likely because of logic changes in the stored proc used to identify the duplicates, we will need to remove all the entries in the AmGranuleReplacement table for the given collection, except for the entries associated with the file collision rule, and recalculate all the granules for that collection that are registered before the time when the actual recalculation happens.

4.7.22.1 Using the Duplicate Granule Identification Utility

The Duplicate Granule Identification Utility is started by entering the following command from the /usr/ecs/<mode>/CUSTOM/utilities directory:

```
> EcDsAmIdentifyDuplicateGranules.pl <command line parameters>
```

There are various command line parameters that are used in combination with each other. Table 4.7.22-1 provides a description of these parameters.

Table 4.7.22-1. Command Line Parameters of the Duplicate Granule Identification Utility (1 of 3)

Parameter Name	Description
<mode>	Mandatory. Specifies the mode of operation. This must be the first parameter passed, and it must be a valid, existing Inventory mode with a format of OPS or TS[1-4] or DEV0[1-9].
report | recalculate	Mandatory. Either report or recalculate. report indicates that the utility will perform a report on all the duplicate granules recorded in AmGranuleReplacement table, for each collection involved, taking the state of the granule into consideration. recalculate indicates that the utility will recalculate, or identify all the replacement/duplicate granule pairs, based on the duplicate granule rule for each collection involved, and insert them in AmGranuleReplacement table.

Table 4.7.22-1. Command Line Parameters of the Duplicate Granule Identification Utility (2 of 3)

Parameter Name	Description
`-c | -- collectionInfo <shortname.versionId | collection group | all >`	Optional. Specifies which collection to operate on. Used with both `report` and `recalculate` shortname.versionId case: When used with `report`, it means to only report the duplicates in the AmGranuleReplacement table that belong to the specified collection. When used with `recalculate`, it means to only recalculate for the specified collection, if it exists in the AmDupGranRuleChangeEvent table. collection group case: When used with `report`, it means to only report the duplicates in the AmGranuleReplacement table that belong to the specified `collection group`. When used with `recalculate`, it means to only recalculate for the collection within the specified collection group, if they exist in the AmDupGranRuleChangeEvent table all case: When used with `report`, It means to report all the duplicate information recorded in AmGranuleReplacements table. When used with `recalculate`, it means to recalculate for all the collections that exist in AmDupGranRuleChangeEvent table. It's also the default behavior for both `report` and `recalculate` if -c is not specified.

Table 4.7.22-1. Command Line Parameters of the Duplicate Granule Identification Utility (3 of 3)

Parameter Name	Description
`-f | --force`	Optional. Used only with `recalculate`. Could be paired with `-c`, `-g`, or `-a`, or none, which is the same as pairing with `-a` It means to force the recalculation on a specific collection, or a collection group, or all the collections in the system, regardless of whether the collection exists in the AmDupGranRuleChangeEvent table. If the collection does exist in AmDupGranRuleChangeEvent table, at the end of the recalculation, on that collection, the entry for that collection will be removed, if the NewRuleNo is the same as the rule number used for the recalculation. Exception: QA/PH/MAP/BR/DAP will not be included
`-h | --includeDFAH`	Optional. Used with `report` only. When specified, the geoid file will not exclude any duplicate granules with DFA set to "H".
`-r |--includeRestricted`	Optional. Used with `report` only When specified, the geoid file will not exclude any duplicate granules that exist in DsMdGranuleRestriction table.
`-o | --outputDir <path>`	Optional. Used with `report` only. It specifies the relative path under the base directory `defined under` parameter DUPLICATE_OUTPUT_DIR in the configuration file EcDsIdentifyDuplicateGranules.CFG. Note: the base directory has to exist; The relative directory (only one level down) will be created if it doesn't exist. This is where all the output files reside. If the relative path is not provided, the output files will go to the base directory.

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

4.7.22.2 Duplicate Granule Identification Utility Commands

Below is an example for invoking this utility:

1. `EcDsIdentifyDuplicateGranules.pl DEV03 report -g "AMSA" --outputLocation Duplicates --includeDFAH`

The utility will report all the "AMSA" duplicates, based on the information stored in the AmGranuleReplacement table, for the "AMSA" data. The output geoid file will be written to the

'Duplicates' subdirectory under the DUPLICATE_OUTPUT_DIR directory. The output geoid file will not exclude any duplicate granules that have DeleteFromArchive set to "H".

2. `EcDsIdentifyDuplicateGranules.pl DEV03 recalculate`

The utility will recalculate on all the collections existing in the AmDupGranRuleChangeEvent table.

3. `EcDsIdentifyDuplicateGranules.pl DEV03 recalculate -f -c "AST_L1BT.001"`

The utility will recalculate on collection "AST_L1BT.001", regardless if the collection exists in the AmDupGranRuleChangeEvent table.

4.7.22.3 Required Operating Environment

The utility can run on any host that has access to the AIM database.

4.7.22.4 Interfaces and Data Types

Table 4.7.22-2 lists the products that this tool depends upon in order to function properly.

Table 4.7.22-2. Interface Protocols

Product Dependency	Protocols Used	Comments
AIM schema	SQL	Via database server

4.7.22.5 Configuration File Format – EcDsIdentifyDuplicateGranules.CFG

The configuration file contains details about how to connect to the database server. Without this file, the tool can not run. Table 4.7.22-3 shows a sample configuration file.

Table 4.7.22-3. Sample Configuration File

Parameter Name	Description
DBUSERNAME	The user name for the RDBMS connection.
DBSERVER	The name of the RDBMS server.
DBSERVERPORT	The port for the RDBMS supporting the mode
DBNAME	The DB name within the RDBMS
DBSUBSYSTEM	The RDBMS schema/subsystem (aim) hosting this utility
PGM_ID	Program ID used for connecting to the database.
NUM_RETRIES	The number of times that the utility attempts to connect to the database before exiting.
SLEEP_SEC	The number of seconds the utility waits between database connection attempts.
DUPLICATE_OUTPUT_DIR	The base directory where output files from the utility are written. The recommended value is /workingdata/emd/<mode>/Idg

4.7.22.6 Special Constraints

The Duplicate Granule Identification utility runs only if the AIM database is available.

4.7.22.7 Output

Output file containing the geoids of the duplicates is created under the base directory defined in the configuration file under DUPLICATE_OUTPUT_DIR, if the `--outputLocation` parameter is not provided on the command line. Otherwise, the output files will be created under the outputLocation, relative to the base directory.

4.7.22.8 Event and Error Messages

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

4.7.22.9 Logs

The utility logs messages in the file

/usr/ecs/<mode>/CUSTOM/logs/EcDsIdentifyDuplicateGranules.log.

4.7.22.10 Recovery

If the Duplicate Granule Identification utility is interrupted by a fault, when the utility is restarted, it will just rerun everything and produce a new output file.