

4.4 Security and Accountability

This section describes the security and accountability tools used by DAAC operators:

1. TCP Wrappers
2. Crack
3. Tripwire
4. Cryptographic Management Interface (CMI)

This page intentionally left blank.

4.4.1 TCP Wrappers

TCP Wrappers allow the operator to control access to various network services through the use of access control lists. They also provide logging information of wrapped network services, which can be used to prevent or monitor network attacks. It intercepts incoming network connections and verifies if the connection is allowed before passing the connection onto the actual network daemon. TCP Wrappers allows the operator to monitor and filter incoming requests for the systat, finger, ftp, telnet, rlogin, rsh, exec, tftp, talk, and other network services. Full descriptions of these Unix services can be obtained using the “man” command, e.g., man systat. TCP Wrappers perform the following functions automatically:

- **Access control:** access can be controlled per host, per service, or combinations thereof.
- **Host name spoofing:** verifies the client host name that is returned by the address->name DNS server, by asking for a second opinion from a local DNS server.
- **Host address spoofing:** the wrapper programs can give additional protection against hosts that claim to have an address that lies outside their own network.
- **Client username lookups:** the protocol proposed in RFC 931 provides a means to obtain the client user name from the client host. The requirement is that the client host runs an RFC 931-compliant daemon. The information provided by such a daemon is not used for authentication purposes but it can provide additional information about the owner of a TCP connection.
- **Multiple ftp/gopher/www archives on one host:** `daemon@host' access control patterns can be used to distinguish requests by the network address that they are aimed at. Judicious use of the `twist' option (see the hosts_options.5 file supplied with TCP Wrappers, `nroff -man' format) can guide the requests to the right server. These can be servers that live in separate chroot areas, or servers modified to take additional context from the command line, or a combination.
- **Sequence number guessing:** client username lookup protocol can help to detect host impersonation attacks. Before accepting a client request, the wrappers can query the client's IDENT server and find out that the client never sent that request.

Additional information on TCP Wrappers can be obtained at the following URL:

<http://www.alw.nih.gov/Security/prog-firewall.html>

TCP Wrappers is used to perform the operator functions listed in Table 4.4.1-1.

Table 4.4.1-1. Common EMD Operator Functions Performed with TCP Wrappers

Operating Function	Command/Action	Description	When and Why to Use
Monitor potentially malicious attempts to access network services.	Check TCP Wrappers log using a text editor.	Program continuously runs in the background appearing to malicious external client service requests as a normal inetd daemon process.	To check for evidence of an attempt of breaking-in.

4.4.1.1 Quick Start Using TCP Wrappers

TCP Wrappers provides a library of tiny daemon wrapper programs. The daemons each correspond to a service provided by the host operating system. The daemons are registered with the service, which results in the operating system invoking the daemon each time that service is invoked. The daemons perform their function(s) and terminate. A common function is to log the name of the client host and requested service. They do not exchange information with client or server applications, and impose no overhead on the actual conversation between the client and server applications. Optional features include: access control to restrict what systems can connect to what network daemons; client user name lookups with the RFC 931 protocol; additional protection against hosts that pretend to have someone else's host name; and additional protection against hosts that pretend to have someone else's host address.

4.4.1.1.1 Command Line Interface

The TCP Wrappers cannot be invoked or accessed from the command line. The TCP Wrapper daemons are invoked by the operating system service to which they are registered. The daemons terminate upon completing their function.

4.4.1.2 TCP Wrapper Main Screen

TCP Wrapper does not have a graphical user interface.

4.4.1.3 Required Operating Environment

For all COTS packages, appropriate information on operating environments, tunable parameters, environment variables, and a list of vendor documentation can be found in a CM controlled document for each product. To find the documentation for AMASS, refer to the EMD Baseline Information System web page, URL <http://pete.hitc.com/baseline/index.html>.

4.4.1.4 Databases

None.

4.4.1.5 Special Constraints

None.

4.4.1.6 Outputs

Table 4.4.1-2 describes TCP Wrappers output.

Table 4.4.1-2. TCP Wrapper Outputs

Output	Disposition	Description and Format
Event log	The disposition of the wrapper logs is determined by the system configuration file parameter for the system log file name. In the ECS Release 4 Development Environment (host = mss1) the parameter is local3.info and the log file is /etc/syslog.conf.	The Wrapper daemons log the event/service request that caused their invocation. The log provides sufficient information to describe the event/service request and response from the system. Log records are output in ASCII text format, each record containing the following fields: Date Time host Service Response to request Event

Figure 4.4.1.1 shows an example of a log file created by TCP Wrappers. The log file can be examined with available tools like the **vi** editor. The contents of the log file can be used to generate reports on Service Request activity for the host.

```
Nov 5 14:49:37 tle2sun in.telnetd[584]: connect from tle2sun.HITC.COM
Nov 5 19:59:26 tle2sun in.rshd[5283]: connect from neptune.HITC.COM
Nov 5 19:59:29 tle2sun in.rshd[5318]: connect from neptune.HITC.COM
Nov 6 10:46:20 tle2sun in.telnetd[6417]: connect from ins1.HITC.COM
Nov 6 11:18:38 tle2sun in.rlogind[6608]: connect from ins1.HITC.COM
Nov 6 13:03:24 tle2sun in.rlogind[6739]: connect from mss1.HITC.COM
Nov 6 13:06:33 tle2sun in.rlogind[6758]: connect from neptune.HITC.COM
Nov 6 13:06:40 tle2sun in.rshd[6761]: connect from neptune.HITC.COM
Nov 6 13:07:03 tle2sun in.rshd[6763]: connect from neptune.HITC.COM
Nov 6 13:07:22 tle2sun in.rshd[6765]: connect from neptune.HITC.COM
Nov 6 14:37:42 tle2sun in.rshd[6860]: connect from stargazer.HITC.COM
Nov 6 16:19:17 tle2sun-e0 in.rlogind[381]: connect from ins1.HITC.COM
Nov 6 19:39:50 tle2sun-e0 in.rlogind[674]: connect from ins1.HITC.COM
Feb 14 16:21:09 vulcan in.rlogind[163]: connect from akashi.HITC.COM
Feb 14 17:17:01 vulcan in.rlogind[367]: connect from akashi.HITC.COM
Feb 14 17:28:35 vulcan in.rshd[371]: connect from huckfinn.HITC.COM
Feb 14 17:40:48 vulcan in.rlogind[372]: connect from dss1.HITC.COM
Feb 14 17:47:28 vulcan in.telnetd[379]: connect from dss1.HITC.COM
Feb 14 18:43:07 vulcan in.telnetd[395]: refused connect from spg-as14s67.erols.com
Feb 14 19:29:00 vulcan in.rlogind[405]: connect from dss1.HITC.COM
Feb 16 05:52:42 vulcan in.rlogind[705]: connect from boston.HITC.COM
Feb 16 19:45:33 vulcan in.telnetd[676]: refused connect from spg-as27s43.erols.com
Feb 17 07:40:49 vulcan in.rshd[1040]: connect from neptune.HITC.COM
Feb 17 19:33:36 vulcan in.rshd[595]: connect from transam.gsfc.nasa.gov
Feb 18 09:34:11 vulcan in.telnetd[858]: connect from hobbes.HITC.COM
Feb 18 10:11:38 vulcan in.telnetd[958]: connect from hobbes.HITC.COM
Feb 18 12:39:20 vulcan in.rlogind[1025]: connect from klingon.HITC.COM
Feb 18 12:40:37 vulcan in.rshd[1046]: connect from neptune.HITC.COM
Feb 18 12:40:44 vulcan in.rshd[1049]: connect from neptune.HITC.COM
Feb 18 12:40:50 vulcan in.rshd[1051]: connect from neptune.HITC.COM
Feb 18 12:40:55 vulcan in.rshd[1054]: connect from neptune.HITC.COM
```

Figure 4.4.1-1. Example of TCP Wrappers Log

The log file provides the following information for each entry: data and time; host sever name; type of service requested and port that provides that service; answer given to the request connection (connect/refused); client host name.

4.4.1.7 Event and Error Messages

None.

4.4.1.8 Reports

None.

4.4.2 Crack

Crack is a COTS freeware product used in compliance management. Crack is designed to find standard Unix eight-character Data Encryption Standard (DES) encrypted passwords by standard guessing techniques outlined below. It is flexible, configurable and fast, and able to make use of several networked hosts via the Berkeley rsh program (or similar), where possible.

Crack takes as input a series of password files and source dictionaries. It merges the dictionaries, turns the password files into a sorted list, and generates lists of possible passwords from the merged dictionary or from information gleaned about users from the password file. It does not attempt to remedy the problem of allowing users to have obvious passwords, which can be guessed, and it should not be used in place of getting a really good, secure password program replacement.

Crack works by making many, individual passes over the password entries supplied to it. Each pass generates password guesses based upon a sequence of rules, supplied to the program by the user. The rules are specified in a simplistic language in the files `gecos.rules` and `dicts.rules`, to be found in the Scripts directory. The rules are written as a simple string of characters, with one rule to a line. Blank lines, and comment lines beginning with a hash character `#` are ignored. Trailing white space is also ignored. The instructions in the rule are followed from left to right, and are applied to the dictionary words one by one, as the words are loaded. Some simple pattern-matching primitives are provided for selection purposes, so if the dictionary word does not match the pattern, it is ignored.

Additional information on Crack can be obtained at the following URL:

<http://www.alw.nih.gov/Security/prog-firewall.html>

Crack is used to perform the operator functions listed in Table 4.4.2-1.

Table 4.4.2-1. Common EMD Operator Functions Performed with Crack

Operating Function	Command	Description	When and Why to Use
Ensure passwords used are not easy to guess.	<code>Crack filename</code>	Scans the password file and makes guesses of passwords.	Regularly review systems for weak passwords.

4.4.2.1 Quick Start Using Crack

For more information on Crack, see the `readme.txt` file in the Docs directory where Crack is installed. The directory where Crack is installed can be found in the ReadMe file described in Section 4.4.2.3, Required Operating Environment.

The documentation of Crack, used as a basis and referenced in this section, is for version 5.0.

4.4.2.1.1 Invoking Crack From the Command Line Interface

To execute Crack from the command line prompt use:

```
>Crack [filename]
```

Where [*filename*] is the name of the password file.

4.4.2.2 Crack Main Screen

Crack does not have a GUI. Crack output is returned to the command line interface where Crack was started. The Crack startup message and initialization is shown in Figure 4.4.2-1.

```

X mss2
mss2{rsnyder}>71: ypcat passwd > ypfile
mss2{rsnyder}>72: ./Crack -nice 10 ypfile
Crack 4.1f RELEASE, The Password Cracker (c) Alec D.E. Muffett, 1992
Invoked as: ./Crack -nice 10 ypfile
Making dictionary Dicts/bigdict - This may take some time...
touch Dicts/.lockfile
Binary directory: /home/rsnyder/COTS/security/crack_4.1/10
( cd ../Sources ; make clean )
make[1]: Entering directory `/home/rsnyder/COTS/security/crack_4.1/Sources'
rm -f *.o *.u *.a *.pixie *.Addrs *.Counts
rm -f crack-pwc tester bytesex testrule
rm -f speedcrypt speedfcrypt speedxform speedufc
make[1]: Leaving directory `/home/rsnyder/COTS/security/crack_4.1/Sources'
( cd ../Sources ; make crack-pwc,which )
make[1]: Entering directory `/home/rsnyder/COTS/security/crack_4.1/Sources'
Choosing between Crack.fcrypt and Crack.ufc
((../Scripts/do_ufc && make crack-pwc,ufc) || make crack-pwc,fcrypt)
Looking for UFC-crypt in /home/rsnyder/COTS/security/crack_4.1/ufc-crypt
Cannot find /home/rsnyder/COTS/security/crack_4.1/ufc-crypt - cannot use UFC-cry
pt on this platform
make[2]: Entering directory `/home/rsnyder/COTS/security/crack_4.1/Sources'
cc -O -c crack-lib.c -o crack-lib.o
cc -O -c crack-glob.c -o crack-glob.o
cc -O -c crack-supp.c -o crack-supp.o
cc -O -c crack-sort.c -o crack-sort.o
cc -O -c bytesex.c -o bytesex.o
"bytesex.c", line 19; warning: shift count negative or too big: <=<= 32
cc -O -o bytesex bytesex.o
cc -O `./bytesex` -c crack-fcrypt.c
cc -O -c crack-pwc.c -o crack-pwc.o
"crack-pwc.c", line 1103; warning: argument #2 is incompatible with prototype:
prototype: pointer to function(int) returning void : "/usr/include/signa
l.h", line 37
argument : pointer to function(void) returning void
cc -O -o crack-pwc crack-pwc.o crack-lib.o crack-glob.o crack-supp.o crack-sort.
o crack-fcrypt.o
make[2]: Leaving directory `/home/rsnyder/COTS/security/crack_4.1/Sources'
make[1]: Leaving directory `/home/rsnyder/COTS/security/crack_4.1/Sources'
cp ../Sources/crack-pwc .
Sorting data for Crack.
Flags: -nice -i /tmp/pw.24133 Dicts/bigdict
Running program in background
Output will be written to a file in directory /home/rsnyder/COTS/security/crack_
4.1
named 'out<something>'
mss2{rsnyder}>73: █

```

Figure 4.4.2-1. Xterm Window of Crack Startup Message and Initialization

4.4.2.3 Required Operating Environment

Crack can run on any EMD platform. However, due to the possibility of security breaches and the high overhead in using the tool, it is implemented on only a few select machines.

Additional information regarding Crack is stored in a CM controlled ReadMe file. To find the ReadMe file for Crack, use the ClearCase Baseline Manager to determine where in ClearCase the ReadMe file resides.

4.4.2.4 Databases

The source dictionaries used by Crack are gecos.rules and dicts.rules. These files are maintained at the SMC and updates periodically forwarded to the DAACs. Instructions for updating these files are included in the Crack Readme file located in the Crack directory.

4.4.2.5 Special Constraints

None

4.4.2.6 Outputs

Table 4.4.2-2 describes the output for Crack.

Table 4.4.2-2. Outputs for Crack

Output	Description and Format
Notification of a person's user id and password used to get into the system.	The results of running Crack are written to the file specified when Crack is invoked and can be shown on the screen in ASCII format.

4.4.2.7 Event and Error Messages

None.

4.4.2.8 Reports

Crack 4.1 does not support reports.

4.4.3 Tripwire

Tripwire is an intrusion detection tool that aids system administrators and users in monitoring a designated set of files for any changes. File systems may be altered without authorization in a number of ways, including an intruder, an authorized user violating a DAAC policy, or malicious code-altering system executables as others are run. Using Tripwire, unauthorized changes are tracked in a very short amount of time.

Tripwire automates the creation of input lists and output lists of files. Tripwire uses the file `tw.config` to maintain the list of tested files. File attributes such as file size, ownership, inode number, inode values and timestamps are compared between the input and output lists. For each file, Tripwire computes a digital signature, which is a fixed-sized output generated by a signature function whose input is an arbitrary file. If the contents of a file are changed in any way, the signature also changes. One of the signature functions is to test for the integrity of a file system by generating checksums of files and comparing them with a previously generated database of checksums. Added or deleted files are flagged and reported, as are any files changed from their previously recorded state in the database. When run against system files on a regular basis, any file changes would be spotted when Tripwire is next run, giving system administrators information to enact damage control measures immediately.

Tripwire uses message-digest algorithms (one-way hash functions) to detect changes in a hard-to-spoof manner. This detects significant changes to critical files, including those caused by insertion of backdoor traps or viruses. Tripwire also monitors changes to file permissions, modification times, and other significant changes to inodes as selected by the system administrator on a per file/directory basis. Tripwire performs the following functions automatically:

- **Database Generation** -- Tripwire initializes the database based upon the entries enumerated in the `tw.config` file.
- **Database Update** -- Provides incremental database update functionality on a per-file/directory basis. This obviates having to regenerate the entire database every time a file or set of files change.
- **Integrity Checking** -- Generates a report of added, deleted, or changed files, comparing all the files described by the `tw.config` file against the files residing on the file system.
- **Interactive Update** -- Reports added, deleted, and changed files and prompts the user whether those database entries should be updated. The Interactive Update provides a method for system administrators to keep Tripwire databases ``in sync" with file systems that change.

Tripwire is used to perform the operator functions listed in Table 4.4.3-1.

Table 4.4.3-1. Common EMD Operator Functions Performed

Operating Function	Command	Description	When and Why to Use
Change the configuration file.	Edit the specific configuration file using the vi editor.	Specify which file(s) should be monitored.	When another file needs to be monitored. Checks the integrity of the file system specified when the daemon is started.
Compare file signatures with database.	Done by Tripwire "cron" run periodically.	Compares files' current signatures against the database and emails the operator a notification for changed files.	This activity is a continuous, periodic performed on a configured interval by the "cron" run.
Update the signatures data store.	Done manually in response to "Interactive Update" prompts.	Updates the signature data store when the email notification discloses legitimate changes.	As necessary to maintain a valid data store of signatures.

4.4.3.1 Quick Start Using Tripwire

The following command is used to execute Tripwire from the command line prompt:

/etc/tripwire-1.2/src/tripwire -v > {filename}

The following is the general syntax of executing Tripwire

tripwire [options ...] >filename

Where *options* are:

-initialize	Database Generation mode -init
-update entry	update entry (a file, directory, or tw.config entry) in the database
-interactive	Integrity Checking mode with interactive entry updating
-loosedir	use looser checking rules for directories
-d dbasefile	read in database from dbasefile (use '-d -' to read from stdin)
-c configfile	read in config file from configfile (use '-c -' to read from stdin)
-cfd fd	read in config file from specified fd
-dfd fd	read in the database file from specified fd
-Dvar=value	define a tw.config variable (ala @@define)
-Uvar	undefine a tw.config variable (ala @@undef)
-i # or -i all	ignore the specified signature (to reduce execution time)
-q	quiet mode
-v	verbose mode
-preprocess	print out preprocessed configuration file
-E	save as -preprocess
-help	print out interpretation help message
-version	print version and patch information

filename is a complete filename (including path) for the output report file.

Tripwire is automatically invoked on all machines by a “cron” run, which periodically executes Tripwire. The operator receives information from Tripwire by email for files whose current signature does not match the datastore signature. The operator must verify the file changes and update the datastore or report a security violation. Tripwire may be run manually to update the datastore or create reports. The Operator can also generate Tripwire reports via the command line.

Additional information on Tripwire can be obtained at the following URL:

<http://www.alw.nih.gov/Security/prog-monitor.html>

The documentation of Tripwire, used as a basis and referenced in this section, is for version 1.3.1.

The differences between the behaviors of Tripwire started from the “Cron” run and started by the operator result from the use of appropriate parameters on the start command. These parameters are listed and explained below.

4.4.3.2 Tripwire Main Screen

Tripwire does not have a GUI. The Tripwire startup message is shown in Figure 4.4.3-1.

```

X mss2
mss2{rsnyder}162: ./tripwire -i 1 -i 2
### Phase 1:  Reading configuration file
### Phase 2:  Generating file list
### Phase 3:  Creating file information database
### Phase 4:  Searching for inconsistencies
###
###                Total files scanned:           2867
###                Files added:                   0
###                Files deleted:                 0
###                Files changed:                 2677
###
###                After applying rules:
###                Changes discarded:             2675
###                Changes remaining:             2
###
changed: -rw----- rsnyder   499198 Apr 23 11:09:42 1998 /home/rsnyder/COTS/se
curity/tripwire-1.2/src/databases/tw.db_mss2
changed: -rwxr-xr-x rsnyder    4725 Apr 23 11:11:09 1998 /home/rsnyder/./cshrc
### Phase 5:  Generating observed/expected pairs for changed files
###
### Attr          Observed (what it is)           Expected (what it should be)
### =====
/home/rsnyder/COTS/security/tripwire-1.2/src/databases/tw.db_mss2
    st_size: 499198                               483328
    st_mtime: Thu Apr 23 11:09:42 1998           Thu Apr 23 11:09:36 1998
    st_ctime: Thu Apr 23 11:09:42 1998           Thu Apr 23 11:09:36 1998

/home/rsnyder/./cshrc
    st_mtime: Thu Apr 23 11:11:09 1998           Tue Apr 21 10:08:39 1998
    st_ctime: Thu Apr 23 11:11:09 1998           Tue Apr 21 10:08:39 1998

mss2{rsnyder}163: █

```

Figure 4.4.3-1. Xterm Window with Tripwire Showing Tripwire Startup Message

4.4.3.3 Required Operating Environment

Tripwire runs on all Sun and Origin servers.

For all COTS packages, appropriate information on operating environments, tunable parameters, environment variables, and a list of vendor documentation can be found in a CM controlled document for each product. To find the documentation for Tripwire, refer to the EMD Baseline Information System web page, URL <http://pete.hitc.com/baseline/index.html>.

4.4.3.4 Databases

Tripwire uses an internal data store of captured information. The user can update this data store through the command line interface. Reporting information based on the information Tripwire has gathered and placed in this data store is sent by email to the operator.

4.4.3.5 Special Constraints

None.

4.4.3.6 Outputs

Tripwire generates the outputs presented in Table 4.4.3-2 below in the filename specified on the command line invocation. A sample of the generated report is shown in section 4.4.3.8.

Table 4.4.3-2. Tripwire Outputs

Output	Description and Format
Tripwire compares the new datastore with the existing Tripwire datastore stored on the file system, reporting added or deleted files, as well as those files that have changed.	See 4.4.3.8, the Report section, for a sample of Tripwire output.
Email to the operator.	Email messages list the files examined by Tripwire whose current signature does not match the file's entry in the signature file.
Updates to the Tripwire datastore.	The operator must review the email mentioned above and determine whether it represents a data corruption problem or the Tripwire signature datastore is out-of-date. If the determination is the data store is out of date, the operator must use Tripwire with the interactive update option and update the signature file.
Security problem notification.	If the operator determines the Tripwire email indicates a security violation the operator must log the problem.

4.4.3.7 Event and Error Messages

None.

4.4.3.8 Reports

Tripwire must be started from the command line interface to request the Tripwire report. A sample of Tripwire output is shown below in Figure 4.4.3-2.

```

2:30am (mentor) 985 % Tripwire
### Phase 1: Reading configuration file
### Phase 2: Generating file list
### Phase 3: Creating file information database
### Phase 4: Searching for inconsistencies
###
###          Total files scanned:      82
###          Files added:              0
###          Files deleted:            0
###          Files changed:           80
###
###          After applying rules:
###          Changes discarded:       79
###          Changes remaining:       1
###
changed: -rw----- genek 4433 Oct 13 02:30:34 1992 /tmp/genek/Tripwire-0.92/config.h
### Phase 5: Generating observed/expected pairs for changed files
###
### Attr    Observed (what it is)      Expected (what it should be)
### =====
/tmp/genek/Tripwire-0.92/config.h
st size: 4441          4433
md5 (sig1): 0aqL1O06C3Fj1YBXz3.CPdeb 0cPX1H.DYS.s1vZdKD.ELMDR
snefru (sig2): 0PcgcK/MZvEm.8pIWe.Gbnn/ /8VoJv1JcoUA0NvoGN.k3P6E
crc32 (sig3): .EHA6x          /OuGNV
crc16 (sig4): ...9/q          ..6yu
md4 (sig5): /hQ0sU.UEbJo.UR4VZ/mNG/h .UR4VZ/mNG/h/VSG/W/Z643k
md2 (sig6): .hLwjb.VRA0O.Z72y90xTYqA 1LR0Gg1l.vqB0.1g330Pi8/p

```

Figure 4.4.3-2. Tripwire Report

4.4.4 Cryptographic Management Interface (CMI)

The Cryptographic Management Interface (CMI) GUI program, *EcSeAuthnProg*, is used by operations personnel to generate a randomized username and password (though only the password is currently used) given a key. There is one key for each EMD server and is the same as the Program ID stored in a server's configuration file. This tool is most often used to generate passwords for Sybase and FTP user accounts. It is therefore recommended that access to this tool is restricted to Sybase and Unix System Administrators only.

CMI is used to perform the operator functions listed in Table 4.4.4-1.

Table 4.4.4-1. Common EMD Operator Functions Performed with CMI

Operating Function	Command / GUI	Description	When and Why to Use
Start <i>CMI</i> program.	<i>EcSeAuthnProg</i>	This brings up the <i>ConnectAuth</i> GUI.	In order to obtain the user password for a given application key.
Generate password.	<i>CMI Main Screen (ConnectAuth GUI)</i>	This causes the program to generate a randomized username and password.	This is only needed when an EMD server requires a new user account.

4.4.4.1 Quick Start Using CMI

The CMI Main Screen is a custom developed GUI utility and should be used only by operations personnel.

To execute CMI from the command line prompt, enter:

> **EcSeAuthnProg**

4.4.4.2 CMI Main Screen

Figure 4.4.4-1 is the CMI GUI Screen, which comes up when the CMI program is run. It contains three fields:

- Application Key field
- User Id field
- Password field

Operations personnel fill out the first field with the application key. In response, CMI returns a user name and password, which are displayed in the associated fields.

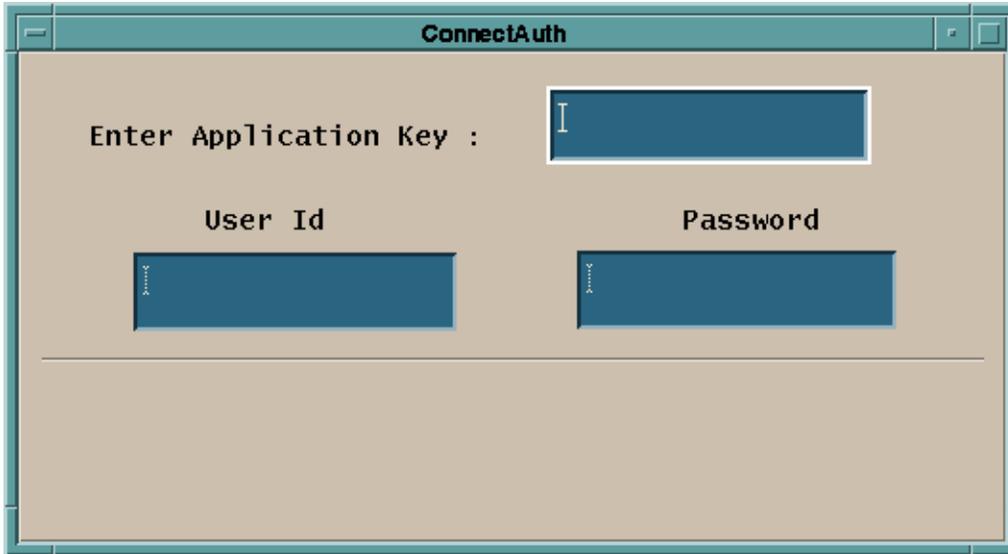


Figure 4.4.4-1. CMI Main Screen

Table 4.4.4-2 describes all the fields found in the CMI Screen in Figure 4.4.4-1.

Table 4.4.4-2. CMI Field Descriptions

Field Name	Data Type	Size	Entry	Description
Application Key	Integer	1 to 10 digits	Required	Key identifying an application.
User Id	Character	8	Generated by <i>EcSeAuthnProg</i> program	Displays the randomized user id based on the key (this field is not used).
Password	Character	8	Generated by <i>EcSeAuthnProg</i> program	Displays the password to be used when creating the account.

4.4.4.3 Required Operating Environment

The *EcSeAuthnProg* depends on a data file, which must be called “data” and must exist in the directory from which the tool is invoked. The data file is the same file as the *EcSeRandomDataFile* located in `$ECS_HOME/<mode>/CUSTOM/security`, only with a different name. CMI requires no other configuration files. The program must be run on a Sun platform.

4.4.4.3.1 Interfaces and Data Types

CMI utilizes no special data types or interfaces.

4.4.4.4 Databases

None.

4.4.4.5 Special Constraints

A data file called “**data**” must exist in the execution directory. The data file must be the same file as the EcSeRandomDataFile.

4.4.4.6 Outputs

All information is displayed on the CMI screen.

4.4.4.7 Event and Error Messages

The CMI program issues error messages, which are listed in Appendix A.

4.4.4.8 Reports

None.

This page intentionally left blank.

4.5 Science Software Integration and Test (SSI&T)

This section describes the tools used by DAAC operations personnel who are Science Software Integration and Test (SSI&T) specialists. The function of SSI&T is to prepare the science software received from the Instrument Teams for DAAC production. The tools available to the specialist cover a broad spectrum. Some are COTS tools falling into the category of “Office Automation” such as spreadsheet and word processing packages. Others are COTS but are more specialized, such as graphic display packages. All the COTS tools/products are documented in separate product specific documentation. These tools are only identified in this section. Operators must verify that COTS documentation matches the product version in use. Finally, there are custom applications that are unique to the SSI&T activity. These tools are described in the following subsections.

This page intentionally left blank.

4.5.1 Science Software Integration and Test (SSI&T) Manager

The SSI&T Manager is the starting point for use by the SSI&T specialist to check in and verify the science software delivered by the instrument teams at the Science Computing Facilities (SCFs). The SSI&T Manager application provides access to all COTS tools and custom applications that are part of the SSI&T environment. The SSI&T Manager GUI is capable of kicking off instrument-specific compilation and execution scripts, configuration management scripts, custom code checking, file display and comparison tools, and COTS tools such as analysis environment programs. The SSI&T Manager GUI contains a checklist of SSI&T steps in delivery and testing of science software, and a display of a log file recording SSI&T events. The checklist and log are the only inherent functionality to SSI&T Manager. All other programs run from the Manager can also be invoked from the Unix command line.

The terms Process Control File (PCF) and Object Description Language (ODL) are used in the following sections. The PCF is a file telling an executable where to find its various inputs and outputs, as well as the values for any specific runtime parameters. Different variants of these files are used both by the SSI&T Manager and by PGEs. ODL is a “parameter = value” format for input files. It is used to define the PGEs to the Planning and Data Processing Database.

Table 4.5.1-1 presents a summary of the capabilities provided via the SSI&T Manager GUI.

Table 4.5.1-1. Common ECS Operator Functions Performed through the SSI&T Manager GUI (1 of 3)

Operating Function	Command/Script or GUI	Description	When and Why to Use
Prepare SSI&T checklist.	SSI&T Manager GUI	<ul style="list-style-type: none"> • SSI&T Manager GUI requires a checklist it can access. • The checklist contains all the operational procedures for science software integration and test. 	<ul style="list-style-type: none"> • During normal SSI&T operations, used to keep track of activities pending and completed. • Checklist of operational procedures must first be prepared using a text editor. • SSI&T Manager must then be linked to the checklist before invoked.
Change SSI&T Checklist.	SSI&T Manager GUI	<ul style="list-style-type: none"> • SSI&T Manager GUI requires changing the checklist. • The checklist contains all the operational procedures for science software integration and test. 	<ul style="list-style-type: none"> • Update tracking of activities pending and completed. • Checklist of operational procedures must first be prepared using a text editor. • SSI&T Manager must then be linked to the checklist before invoked.

Table 4.5.1-1. Common ECS Operator Functions Performed through the SSI&T Manager GUI (2 of 3)

Operating Function	Command/Script or GUI	Description	When and Why to Use
Open xterm session	<ul style="list-style-type: none"> • Open xterm window via Tools pull-down menu. • Also can be opened via Unix command <i>xterm</i>. 	Standard Unix command line window.	As needed for ad hoc use.
Code analysis	Select Sparcworks via Tools: Code Analysis pull-down menu.	Used for ad hoc analysis of science software.	Used to debug problems (e.g., memory leaks).
Check for standards compliance	Select the following via Tools: Standards Checkers pull-down menu: <ul style="list-style-type: none"> • FORCHECK • Prohibited Function Checker • Process Control File (PCF) Checker • Prolog extractor 	<ul style="list-style-type: none"> • Check FORTRAN 77 science software. • Check if certain functions are used in the science software, which conflict with the production environment. • Check the syntax of the data in the Process Control File. • Extract prologs from science software. 	<ul style="list-style-type: none"> • To ensure that science code conforms to ECS standards. • To ensure the delivered PCF is of the proper syntax. • To extract prologs from science software.
Product Examination	Select the following via Tools: Product Examination pull-down menu: <ul style="list-style-type: none"> • IDL • EOSView 	Opened via Tools pull-down menu.	<ul style="list-style-type: none"> • Ad-hoc graphical analysis. • For viewing an arbitrary file (e.g., standard product) in HDF format.
File Comparison	Select the following via Tools: Product Examination File Comparison <ul style="list-style-type: none"> • ASCII files • Binary files • HDF files (GUI) • HDF files (hdiff) 	Compares the outputs of the science software between the DAAC and SCF.	Ensures output generated at the SCF when running the science software is the same output generated at the DAAC.

Table 4.5.1-1. Common ECS Operator Functions Performed through the SSI&T Manager GUI (3 of 3)

Operating Function	Command/Script or GUI	Description	When and Why to Use
Edit text	Select the following via Tools: Text Editors pull-down menu: <ul style="list-style-type: none"> • Emacs • Xedit 	Text editors.	Edit arbitrary text file.
PDPS Database	Select the following via Tools: PDPS Database pull-down menu: <ul style="list-style-type: none"> • PCF ODL Template • Check ODL Files • SSI&T Science Metadata Update • SSI&T Operational Metadata Update GUI 	<ul style="list-style-type: none"> • Creates an ODL file template from the science software PCF. • Check the ODL file updates PGE and ESDT SCIENCE metadata in the PDPS /SSI&T database. • Updates PGE OPERATIONAL metadata via GUI in the PDPS /SSI&T database. 	To initialize and update the Planning/Production (PDPS) databases: <ul style="list-style-type: none"> • SSI&T version • Production version
Data Server	Select the following via Tools: Data Server pull-down menu: <ol style="list-style-type: none"> 1. Acquire DAP 2. Insert Static 3. Insert Test Dynamic 4. Insert EXE TAR 5. Edit SSAP 6. Get MCF 	<ol style="list-style-type: none"> 1. Acquires Delivery Archive Package (DAP). 2. Inserts static input file. 3. Inserts test dynamic input file. 4. Inserts tar file with files needed for processing. 5. Edits Science Software Archive Package (SSAP) components. 6. Acquires Metadata Configuration Files (MCFs) from the Data Server. 	<ol style="list-style-type: none"> 1. After DAP notification received by email. 2. After ESDT is registered in Data Server, before test PGE run. 3. After ESDT is registered in Data Server, before test PGE run. 4. After PGE compilation, before test PGE run. 5. After PGE testing is complete, at time of promotion to Production, as needed to edit/review SSAP components. 6. As needed to retrieve MCF.

4.5.1.1 Quick Start Using SSI&T Manager

The SSI&T Manager provides a common interface to the SSI&T tools. An overview of the SSI&T Manager GUI is provided in Section 4.5.1.2. A more detailed discussion of the tools accessed via this GUI from an ECS standpoint is provided in subsequent sections. Further information on tools such as Xedit, Emacs, GhostView, etc. can be found on the web by doing a web search on “*toolname* manual.”

The following assumptions are made with regard to the use of the SSI&T Manager application:

- The operator is located at a workstation or server to which the SSI&T Manager has been configured
- The operator has proper authorization to access the PDPS/SSI&T database and the Data Server
- To access files in ClearCase, the operator has a ClearCase view already set
- The operator's environment (available from the main window of the SSI&T Manager (see Figure 4.5.1-1)) has been configured as documented in the pertinent sections of the Help menu. The Index submenu of the Help menu provides access, through the Netscape browser, to a number of topics that help the operator in the environment configuration. Refer to section 4.5.1.2.10 for additional details on the Help Menu. A list of topics that can be searched through the Help menu is shown in Figure 4.5.1-36.

To start SSI&T Manager at the Unix command line, enter:

\$ECS_HOME/<mode>/custom/utilities/EcDpAtMgrStart<mode>

where:

mode is the ECS mode of operation (e.g., **OPS**, **TS1** or **TS2**)

4.5.1.1.1.1 Sun Platform

Table 4.5.1-2 lists the SSI&T command line interfaces for the Sun workstation.

Table 4.5.1-2. SSI&T Command Line Interfaces (Sun) (1 of 2)

Command Line Interface	Description and Format	When and Why Used
EcDpAtMgrStart	Startup script for SSI&T Manager.	To do SSI&T, and record items accomplished in the log.
EcDpAtMgrLogDump	Used to dump/print a log file to the screen.	As needed.
xterm	Open a Unix command line window.	As needed.
sparcworks	Ad hoc code analysis	As needed.
ghostview	Postscript file viewer	As needed.
netscape	WWW browser Netscape	As needed.
acroread	PDF file viewer Adobe Acrobat	As needed.
DpAtMgrForcheck	FORTTRAN 77 code analysis	Determine whether FORTRAN 77 science software adheres to standards.
EcDpAtBadFuncGui	Prohibited function checker (GUI)	Determine whether science software adheres to standards.
EcDpAtBadFunc	Prohibited function checker (command line)	Determine whether science software adheres to standards.
EcDpAtCheckPCF	Process Control File checker (GUI)	Determine whether PCF is valid.

Table 4.5.1-2. SSI&T Command Line Interfaces (Sun) (2 of 2)

Command Line Interface	Description and Format	When and Why Used
EcDpAtMgrPrologs	Prolog extractor	Extract science software code prologs.
/data/IDL/idl_4/bin/idl	IDL	As needed.
EOSView	EOSView	HDF file viewer.
EcDpAtMgrXdiff	ASCII file comparison	Compare 2 text files.
EcDpAtBinDiffGui	Binary file difference environment	Compare 2 binary files.
DpAtCheckHdfFile	HDF file comparison (GUI)	Compare 2 HDF files.
DpAtHdiff	HDF file comparison (command line)	Compare 2 HDF files.
xedit	Text editor	As needed.
emacs	Text editor	As needed.
EcDpAtCreateODLtmplate	Create PGE metadata ODL template file.	Before running EcDpAtDefinePGE.
EcDpAtVerifyODL	Verify PGE metadata ODL template file.	Before Running EcDpAtDefinePGE.
EcDpAtDefinePGE	Update PDPS/SSI&T database with SCIENCE metadata.	Before executing PGE in SSI&T environment.
DpAtOpDbGui	Update PDPS/SSI&T database with OPERATIONAL metadata.	Before executing PGE in SSI&T environment.
EcDpAtStageDAP	Acquires DAP from the Data Server.	After email subscription notification received.
DpAtInsertStatic	Inserts static input file into the Data Server.	Before executing PGE in SSI&T or Production environment.
DpAtInsertTest	Inserts test dynamic input file into the Data Server.	Before executing PGE in SSI&T environment.
DpAtInsertExeTar	Inserts tar file of executables, etc. needed to run PGE file into the Data Server.	Before executing PGE in SSI&T or Production environment.
EcDpAtSSAPGui	Edit and inserts a single SSAP component into the Data Server.	After SSI&T is finished, before official promotion to Production.
netscape <html page name>	HTML pages for acquiring SSAP components from the Data Server, including test outputs.	During SSI&T, to get test outputs; After SSI&T is finished.
EcDpAtaCQUIREMCF	Get ESDT from the Data Server and insert MCF.	Before inserting MCF in the Data Server.

4.5.1.1.1.2 SGI Platform

It is intended that the SSI&T tools be most often run from the SSI&T Manager. A small number of SSI&T tools run only on the SGI platform. Because of security considerations, these tools cannot be run from the SSI&T Manager on the Sun. They can only be run from the Unix command line on the SGI platform as indicated in Table 4.5.1-3.

Table 4.5.1-3. Command Line Interfaces (SGI)

Command Line Interface	Description and Format	When and Why Used
usr/sbin/cvproj	ProDev Workshop: Used for ad hoc analysis of science software.	Used to determine causes of problems (e.g., memory leaks).
DpAtRusage	Measures PGE performance.	Output of this tool is to be typed into the "Performance Statistics" section of the PROFILE screen of the PDPS/SSI&T Database Update GUI.

Table 4.5.1-4 lists SGI platform tools associated with the SSI&T process.

Table 4.5.1-4. SGI Platform Tools Description

Categories/Tools	Tool Description & Use	Further Information
ProDev Workshop	<ul style="list-style-type: none"> • ProDev Workshop is a COTS package developed by SGI • This tool is targeted within ECS for applications running on the SGI science processors • ProDev Workshop is a software development support tool which includes several tools, which have applicability to SSI&T • Among these tools is the capability to perform static code analysis to aid in the detection of memory leaks 	<ul style="list-style-type: none"> • ProDev Workshop includes online documentation describing its features • Other ProDev Workshop documentation is delivered with ECS. • ProDev Workshop is not available from the SSI&T menu. This tool must be started from the Command Line Interface: see Table 4.5.1-3.
PGE Performance	<ul style="list-style-type: none"> • DpAtRusage is a custom tool developed by ECS • It measures performance parameters such as CPU time used for a PGE linked to the SDP Toolkit, SCF version. 	A help message is printed if the tool is invoked without input parameters.

4.5.1.2 SSI&T Manager

Figure 4.5.1-1 illustrates the SSI&T Manager. This screen is the starting point for SSI&T activities. It provides access to a collection of tools used in the SSI&T process. The procedures for SSI&T using these tools are defined in other documentation in conjunction with the DAAC SSI&T personnel and the instrument teams.

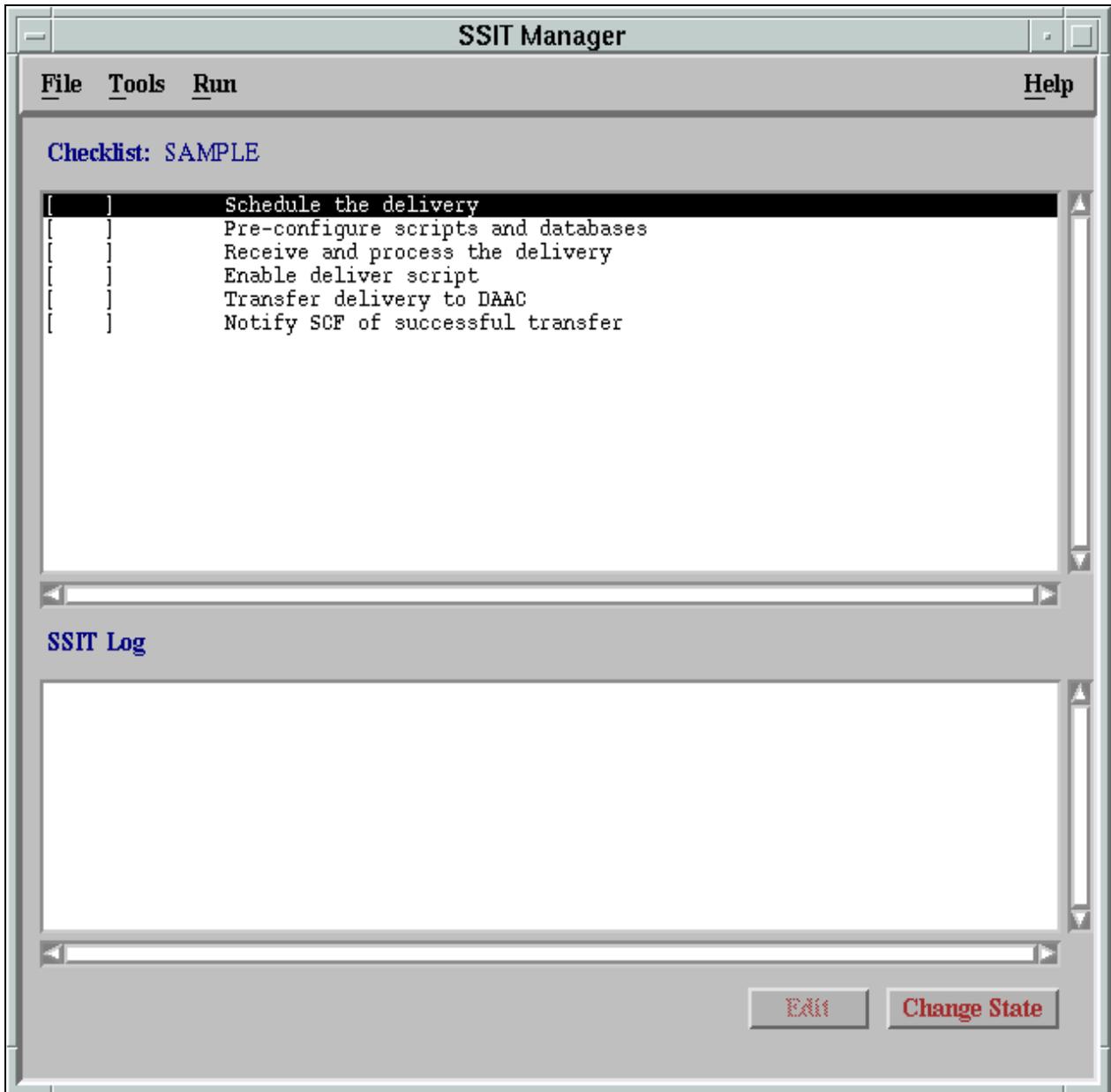


Figure 4.5.1-1. SSI&T Manager Main Screen

The SSI&T Manager Window includes two windows of information, a menu bar at the top of the screen, and two buttons at the bottom of the window.

The Checklist Window

The top window is the SSI&T Checklist. This lists a configurable set of steps to be completed during the SSI&T process. It is displayed as a line of text for each item, with a check box that is checked or unchecked. A checklist can be tailored specifically to a science software suite being delivered. The checklist is manual, in that the operator must check it with a mouse click. The checklist is configurable by editing a checklist text file as explained under the *Checklist* topic of Help/Index menu.

The SSI&T Log Window

The lower window is a log of activities accomplished in the SSI&T process. As steps in the process are accomplished, a log entry is automatically added. Each time a box on the checklist is checked or unchecked, the log is updated. The user is required to annotate any changes.

The Menu Bar

The menu bar includes a simple selection of options:

- **F**ile - Offers the option to exit from the SSI&T Manager application
- **T**ools - Provides access to a collection of tools that are of use in the SSI&T process. The tools available are described in the following section
- **R**un - User-customizable menu, intended for programs such as scripts to run science software
- **H**elp - Provides context-sensitive help

SSI&T Manager Buttons

There are also two buttons at the bottom of the screen.

- **Edit** - The SSI&T specialist, to include additional detail on the SSI&T process, can edit the checklist items. Clicking the Edit button brings up the text editor for this purpose
- **Change State** - This toggle button allows the SSI&T specialist to switch between “checked” and “unchecked” for the selected checklist item

4.5.1.2.1 SSI&T Manager Tools

There are several tools accessible through the SSI&T Manager. After selecting the Tools menu, a set of options is presented as shown in Figure 4.5.1-2.

The first item on the list, the Xterm option, starts an Xterm window session. Selecting any of the other options displays a secondary list of particular tools within that category.

Many of the items accessed through the Tools menu are COTS products for which documentation exists. Some items are custom applications from other ECS subsystems (e.g., EOSView). The complete description and documentation for these items is provided in other parts of this document. The remaining items are custom applications and are described in the

following subsections. Note, since SSI&T is a collection of various tools, there is no specific order in which they must be run. Most tools can be brought up from the SSI&T Manager Main screen as well as be started on their own. The File menu provides the capability to exit the manager, and the Tools menu gives access to various tools that make up SSI&T.

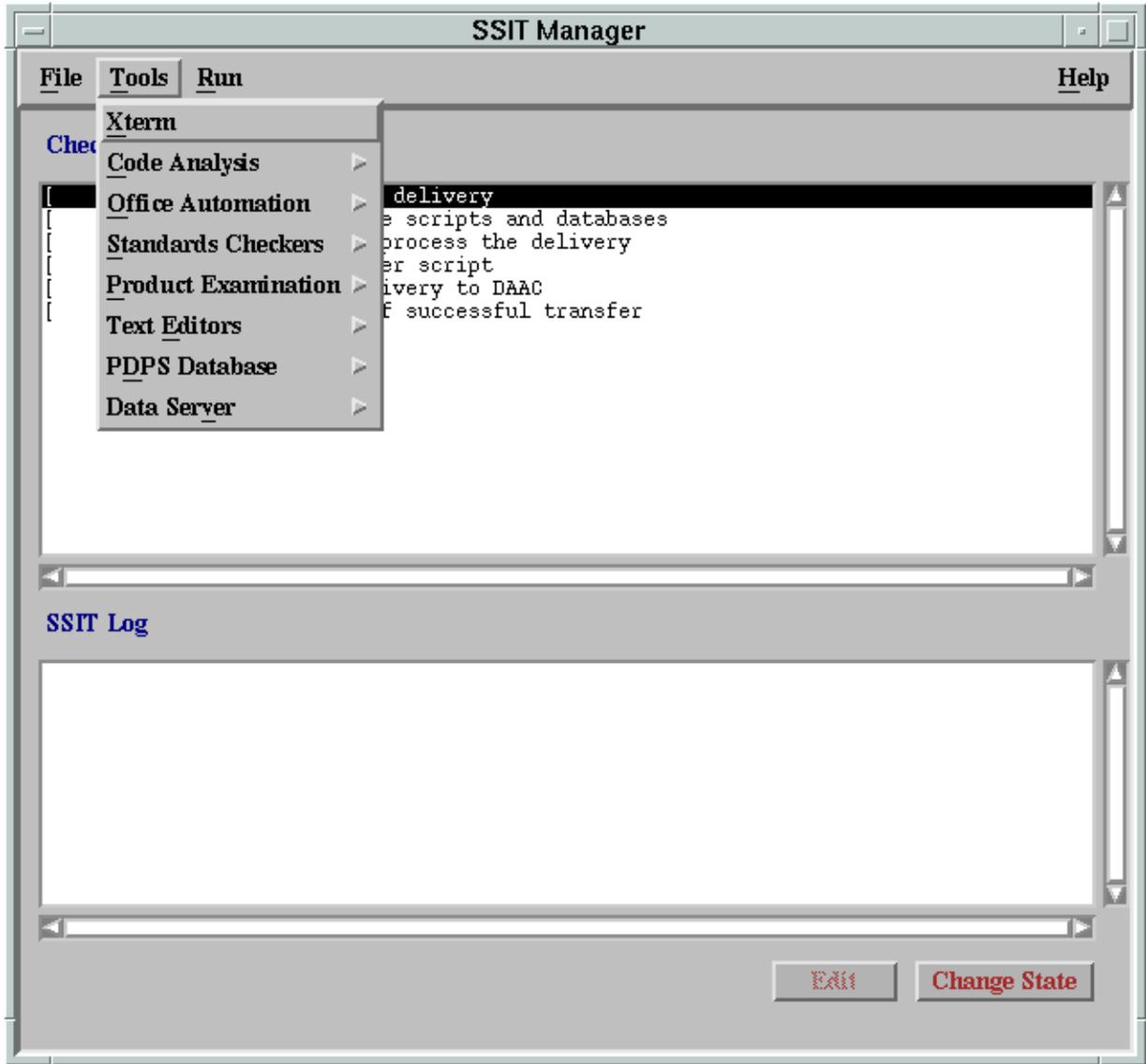


Figure 4.5.1-2. SSI&T Manager Main Screen - Tools Menu

Table 4.5.1-5 describes the SSI&T Manager Tools.

Table 4.5.1-5. SSI&T Manager Tools Description (1 of 5)

Category	Tools	Tool Description & Use	Further Information
UNIX Access	Xterm	<p>This tool's option allows the operator to open an Xterm window and enter Unix commands, as needed.</p> <p>The other SSI&T Manager windows remain available while an Xterm window is open.</p>	<p>Unix documentation is available online via 'man' command.</p>
Code Analysis	SPARCworks	<p>SPARCworks is a COTS package developed by Sun Micro Systems. Selecting this option from the Tools list starts the SPARCworks tool, which opens a separate window.</p> <p>Among the tools available in SPARCworks is the code analysis tool, which allows for static code analysis to aid in detecting memory leaks.</p> <p>For the SGI systems the ProDev Workshop product provides similar functionality.</p>	<ul style="list-style-type: none"> • SPARCworks includes online documentation describing its features • Other SPARCworks documentation is delivered with ECS. This documentation includes: <ul style="list-style-type: none"> – SPARCworks Browsing Source Code – Sparcworks Debugging a Program 3.0.1 – SPARCworks Tutorial 3.0.1 – Sparcworks Performance Tuning an Application 3.0.1
	ProDev Workshop	<p>ProDev Workshop is a COTS package developed by SGI. This tool is targeted within ECS for applications running on the SGI science processors.</p> <p>ProDev Workshop is a software development support tool, which includes several tools having applicability to SSI&T.</p> <p>Among these tools is the capability to perform static code analysis to aid in the detection of memory leaks</p>	<ul style="list-style-type: none"> • ProDev Workshop includes online documentation describing its features • Other ProDev Workshop documentation is delivered with ECS. • ProDev Workshop is not available from the SSI&T menu. This tool must be started from the Command Line Interface: see Table 4.5.1-3.
Office Automation	GhostView	<p>Selecting Ghostview from the Tools menu item starts the Ghostview application.</p> <p>Ghostview is a freeware tool. This is used for displaying Postscript format files.</p>	<p>Ghostview includes online documentation on its use.</p> <p>Since this is a freeware product, there is no other documentation provided by ECS for Ghostview.</p>

Table 4.5.1-5. SSI&T Manager Tools Description (2 of 5)

Category	Tools	Tool Description & Use	Further Information
	Netscape	<p>Netscape is a COTS package developed by Netscape Corp.</p> <p>This tool is a 'Browser', which is used to access Internet World Wide Web (WWW) sites.</p> <p>EOSDIS has adopted the WWW and Netscape as an internal standard for the publication and exchange of various types of documentation, schedules, etc.</p> <p>ECS documentation is available through the WWW, including documentation related to SSI&T.</p>	<p>Documentation on Netscape is available online.</p> <p>Documentation on ECS use of Netscape is available in this document in Section 4.12.3, Netscape Communicator, and 4.12.4, iPlanet Web Server.</p>
	Acrobat	<p>Adobe Acrobat Reader is a COTS package developed by Adobe Systems Inc.</p> <p>This product is used to view electronic documents created in the Portable Document Format (PDF) file format.</p> <p>Many of the ECS documents have been created or converted to this format.</p>	<p>Documentation on Adobe Acrobat products is available online.</p> <p>Other documentation for this application is delivered with ECS.</p>
Standards Checkers	FORCHECK	<p>FORCHECK is a COTS tool used to verify that software follows certain coding standards.</p> <p>FORCHECK analyzes source code written in FORTRAN 77 with ANSI extensions.</p> <p>When selected from the SSI&T Manager window, and xterm window is created in which the program runs.</p> <p>There is no GUI associated with FORCHECK.</p> <p>Note that FORCHECK is configured to check for the non-ANSI FORTRAN 77 extensions, which are approved for use in ECS science code, according to "Data Production and SCF Standards and Guidelines."</p>	<p>Documentation on FORCHECK is available online.</p> <p>Other documentation for this application is delivered with ECS. This includes FORCHECK for Sun/SunOS, A FORTRAN Verifier and Programming Aid, User Guide.</p>

Table 4.5.1-5. SSI&T Manager Tools Description (3 of 5)

Category	Tools	Tool Description & Use	Further Information
	Prohibited Function Checker	<p>The Prohibited Function Checker is a custom application used to determine if the science software has been developed using coding standards established by the ESDIS Project (“Data Production and SCF Standards and Guidelines”) in conjunction with the instrument teams responsible for developing the software.</p> <p>These coding standards are limited to prohibitions against the use of certain types of functions that can cause problems in the production environment.</p> <p>The Prohibited Function Checker checks for functions defined in a science software source code.</p>	<p>On-line help is provided with the Prohibited Function Checker.</p> <p>A description of the use of the tool is provided in the following sections.</p> <p>This tool is described in detail in section 4.5.1.2.5.2.</p>
Standards Checkers	Process Control File Checker	<p>The Process Control File (PCF) checker is a custom application used to verify that the format of the PCF is correct.</p> <p>A PCF is a part of the delivered science software delivery package. The PCF specifies the names and locations of files used by the science software executables, and define the correspondence between (1) the file specifications and (2) the logical identifiers used by the science software to reference the specified files.</p>	<p>On-line help is provided with the PCF Checker.</p> <p>A description of the use of the tool is provided in the following sections.</p> <p>Note that this tool consists of a GUI front end added to an ECS SDP Toolkit utility.</p> <p>Documentation of the utility is found in the SDP Toolkit Users Guide for the ECS Project.</p> <p>This tool is described in detail in section 4.5.1.2.5.3.</p>
	Prolog extractor	Custom application, which extracts code prologs from science software.	Online help is provided.
Product Examination	IDL	<p>IDL (Interactive Data Language) is a COTS product, developed by Research Systems Inc.</p> <p>It is used to interactively visualize and analyze scientific and engineering data products.</p> <p>IDL can be used with data files of an arbitrary format.</p>	<p>Documentation on IDL is available online.</p> <p>Other documentation for this application is delivered with ECS.</p>

Table 4.5.1-5. SSI&T Manager Tools Description (4 of 5)

Category	Tools	Tool Description & Use	Further Information
	EOSView	<p>EOSView is a custom application used to display HDF format files and the metadata associated with those files.</p> <p>EOSView is expected to be primarily used by the EOS scientific community, but is also used by the operations community to support the SSI&T process (for file inspection, etc.) and by the DAAC quality assurance groups to view and analyze problems arising from production operations.</p>	<p>Documentation on EOSView is available online.</p> <p>EOSView is integral with the ECS Desktop; for more information, see Section 4.12.5, "EOSView".</p>
Product Examination	File Comparison	<p>File comparison tools, which are custom products of ECS, are provided with the SSI&T Manager.</p> <p>Tools are provided to support file comparison in three different formats: ASCII, Binary, and HDF.</p> <p>These tools would be used by the SSI&T Specialists to compare the benchmark data products generated at the SCFs with the data products generated as a part of the SSI&T process at the DAAC.</p> <p>All standard data products are produced in HDF format, for which the HDF file comparison utility is used.</p> <p>For intermediate or other data products not adhering to the standards, the binary or ASCII file comparison tools can be used.</p> <p>In the case of binary files, the DAAC operator must write code to do the comparison, for which a support environment is provided.</p>	<p>Documentation on the file comparison tools is available online.</p> <p>These tools are described more fully in following subsections of this document.</p>
Text Editors	Emacs	<p>Emacs is a freeware text-editing tool most frequently used by the SSI&T specialists in developing scripts or ad hoc programs to be run in the SSI&T environment.</p>	<p>Documentation on Emacs is available online.</p> <p>Since this is a freeware product, there is no other documentation provided by ECS.</p>
	Xedit	<p>Xedit is a freeware text-editing tool most frequently used by the SSI&T specialists in developing scripts or ad hoc programs to be run in the SSI&T environment.</p>	<p>Xedit includes online documentation on its use</p> <p>Since this is a freeware product, there is no other documentation provided by ECS.</p>

Table 4.5.1-5. SSI&T Manager Tools Description (5 of 5)

Category	Tools	Tool Description & Use	Further Information
PDPS Database	PCF ODL Template	Creates a PDPS PGE SCIENCE metadata ODL file from science software PCF.	This tool is described in detail in section 4.5.1.2.8.1.
	Check ODL	Parses ODL file and records errors. This tool accesses the PDPS DB for information on PGE/ESDT for which ODL is already available.	This tool is described in detail in section 4.5.1.2.8.2.
	SSI&T Science Metadata Update	Updates PDPS/SSI&T database with PGE and ESDT SCIENCE metadata read from ODL files.	This tool is described in detail in section 4.5.1.2.8.3.
	SSI&T Operational Metadata Update GUI	Updates PDPS/SSI&T database with PGE OPERATIONAL metadata input by user.	This tool is described in detail in the following 4.5.1.2.8.4 section.
Data Server	Acquire DAP	Acquires a DAP from the Data Server.	This tool is described in detail in section 4.5.1.2.9.1.
	Get MCF	Gets information about the ESDT the MCF is acquired for and interface with the Data Server to get the desired MCF.	This tool is described in detail in section 4.5.1.2.9.2.
	Insert Static	Inserts a static input file to the Data Server.	This tool is described in detail in section 4.5.1.2.9.3.
	Insert Test Dynamic	Inserts a test dynamic input file to the Data Server.	This tool is described in detail in section 4.5.1.2.9.4.
	Insert EXE TAR	Inserts tar file of executables, etc. needed to run science software to the Data Server.	This tool is described in detail in section 4.5.1.2.9.5.
	SSAP Editor	Read/Write Metadata and record errors.	This tool is described in detail in section 4.5.1.2.9.6.

4.5.1.2.2 Xterm

Xterm is a COTS product that activates an Xterm window. This allows the operator to use the Unix environment and resources from a PC. The Xterm window as shown in Figure 4.5.1-3 has all the properties of a PC window while the “screen” portion is essentially a Unix terminal. The user needs environment parameters for the PC, such as the IP Address, to use Xterm.

There is no submenu for the Xterm menu.



Figure 4.5.1-3. Xterm Unix Terminal Window

4.5.1.2.3 Code Analysis

The Code Analysis menu as shown in Figure 4.5.1-4 provides the SPARCworks tool to assist in analyzing C code.

4.5.1.2.4 Office Automation

The Office Automation tool menu as shown in Figure 4.5.1-5 provides tools to assist operators in documentation and administrative tasks.

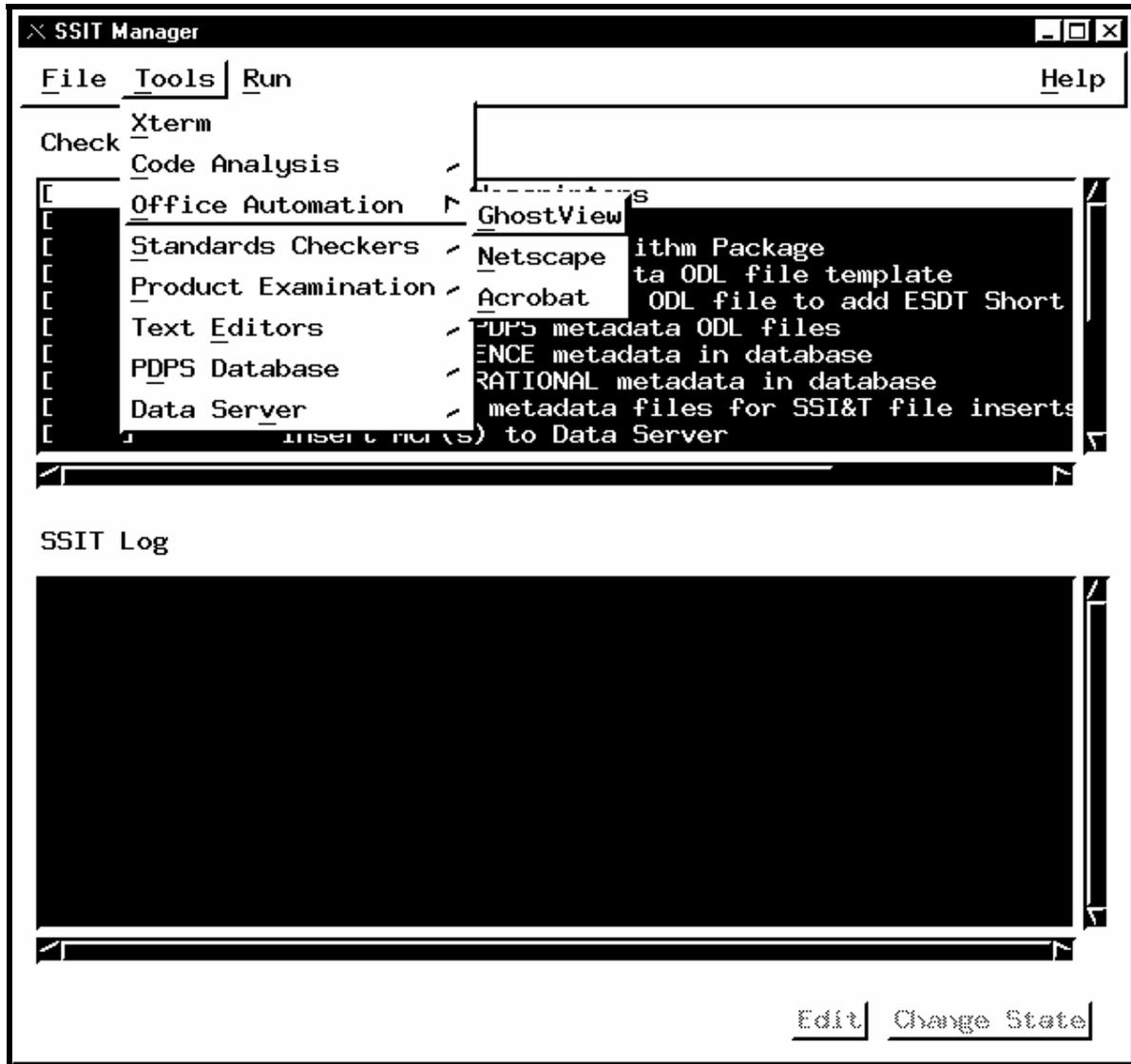


Figure 4.5.1-5. SSI&T Manager Main Screen with Office Automation Tool Menu Displayed

4.5.1.2.4.1 GhostView

GhostView is a COTS product. GhostView as shown in Figure 4.5.1-6 provides the operator with a tool for viewing and editing graphics files. The popular PostScript format (.ps) is used to capture screen images and other pictures used in documentation. The operation of this tool follows standard PC application rules. Images displayed in GhostView can be cut & pasted into MS Word® documents.

More information about GhostView is available on the Web free from the URL, “<ftp://prep.ai.mit.edu/pub/gnu/>”

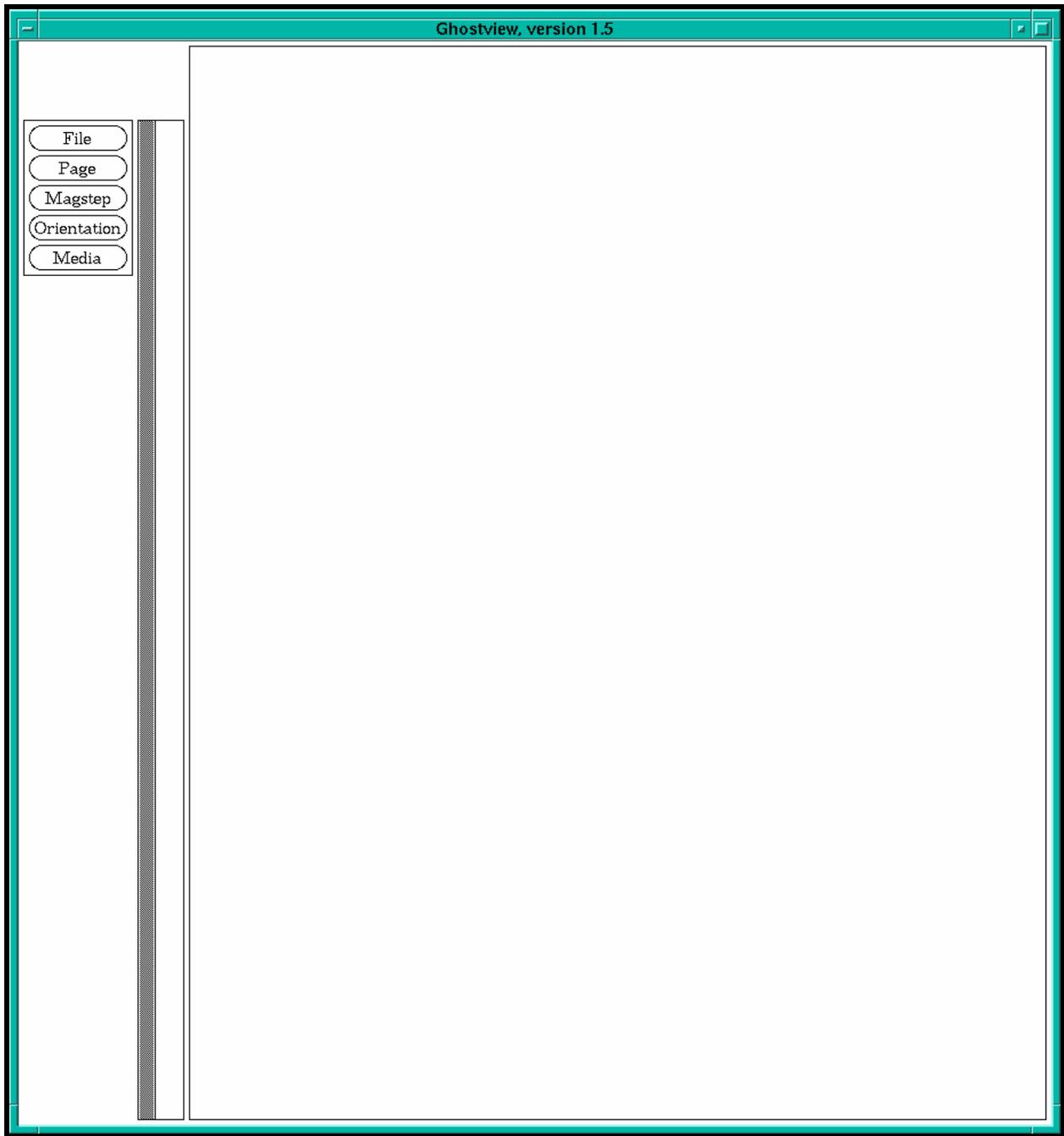


Figure 4.5.1-6. GhostView Main Screen

4.5.1.2.4.2 Netscape

Netscape is a COTS product. Netscape is the project Internet Browser, it is described in Section 4.12.3, “Netscape Communicator.”

4.5.1.2.4.3 Acrobat

Acrobat is a COTS product. Adobe Acrobat is a documentation package oriented to storing data as graphics files. The main window is shown in Figure 4.5.1-7. The package is geared to fast printing by not requiring translation of text files to printer format. The popular Acrobat (.pdf) file format is native to this package. Images or text displayed in Acrobat can be cut & pasted into Word® documents.

More information about Acrobat is available on the Web free from <http://www.adobe.com/>.

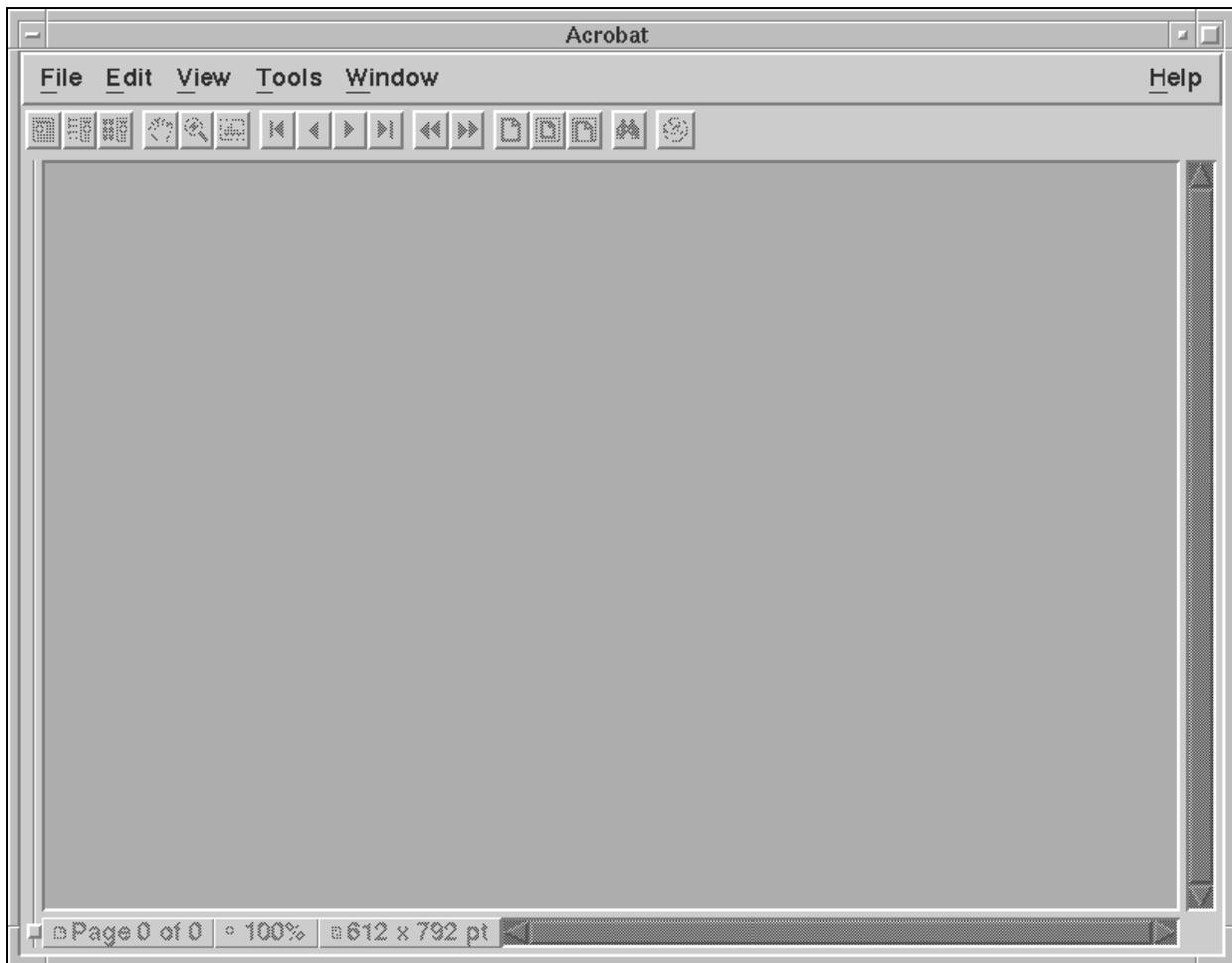


Figure 4.5.1-7. Acrobat Main Screen

4.5.1.2.5 Standards Checkers

The Standards Checkers menu shown in Figure 4.5.1-8 provides tools to assist operators in validating Delivery Algorithm Packages (DAPs).

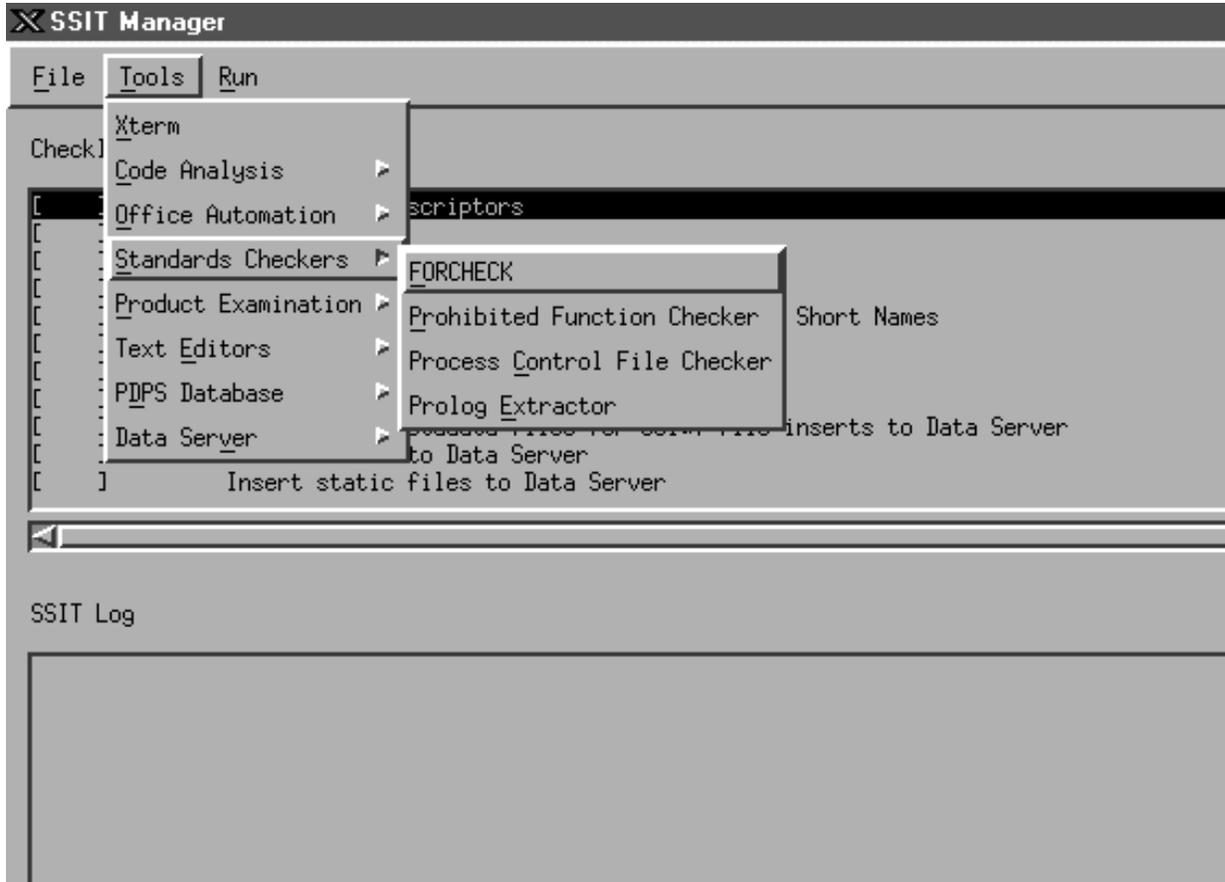


Figure 4.5.1-8. SSI&T Manager, Tools Menu, Standards Checker Submenu Choices

4.5.1.2.5.1 FORCHECK

FORCHECK, shown in Figure 4.5.1-9, is a COTS tool that checks science software written in FORTRAN 77. FORCHECK can be configured to determine whether the ANSI FORTRAN 77 extensions used in science software are allowed by the official NASA/ESDIS standard. The NASA/ESDIS standards are implemented through use of a FORCHECK configuration file, which is pointed to by the \$FCKCNF parameter.

FORCHECK does not have a GUI, it is command line driven. The FORCHECK interface is via a spawned Xterm window.



```
SSIT: FORCHECK
FORCHECK (R) V12.30
Copyright (c) 1984-1998 Leiden University. All rights reserved

Licensed to: EDF Landover Facility, Upper Marlboro, MD - Sun/Solaris ()

-- Sun compiler emulation
global option(s) and list file? █
```

Figure 4.5.1-9. FORCHECK Program Running in an Xterm Window

When invoked, FORCHECK produces a query prompt to solicit changes to the default global options and the list (output) file. The default output file is the UNIX **stout** file, normally configured as the Xterm window. FORCHECK provides numerous Global and Local options to tailor the code analysis and the resulting output. See the *FORCHECK for Sun/SunOS A Fortran Verifier and Programming Aid version 12.83 User's Guide* for more information.

4.5.1.2.5.2 Prohibited Function Checker

The document, *Data Production Software and Science Computing Facility (SCF) Standards and Guidelines, Revision A, October 1996 (423-16-01)*, mandates that science software delivered to the DAACs for integration into the ECS should be free of prohibited functions which can cause errors in the production environment. The Prohibited Function Checker (PFC) is an SSI&T tool offering a GUI interface to check source code for prohibited function calls.

The PFC GUI provides the following functionality:

- Allows the user to select file(s) to be checked for presence of prohibited functions.
- Provides a toolkit to scan program and text files for prohibited functions.
- Writes output to a log.
- Allows the user to scan through any prohibited functions found.

Table 4.5.1-6 lists the language and associated file extensions that can be checked for prohibited function calls using the PFC Tool.

Table 4.5.1-6. File Name Extensions

Language	File Name Extensions
Ada	.a, .ada
C	.c, .h
C++	.cc
Fortran 77	.f, .f77, .ftn, .inc
Fortran 90	.f90
C shell	.csh
Korn shell	.kh
Bourne Shell	.sh
Perl	.pl
Text files	.txt

4.5.1.2.5.2.1 Checking for Prohibited Functions

From the *SSI&T Manager Main Screen* (Figure 4.5.1-8), select the **Tools->Standards Checkers->Prohibited Function Checker** option to bring up the screen shown in Figure 4.5.1-10. This displays the *Prohibited Function Checker Pop-up*.

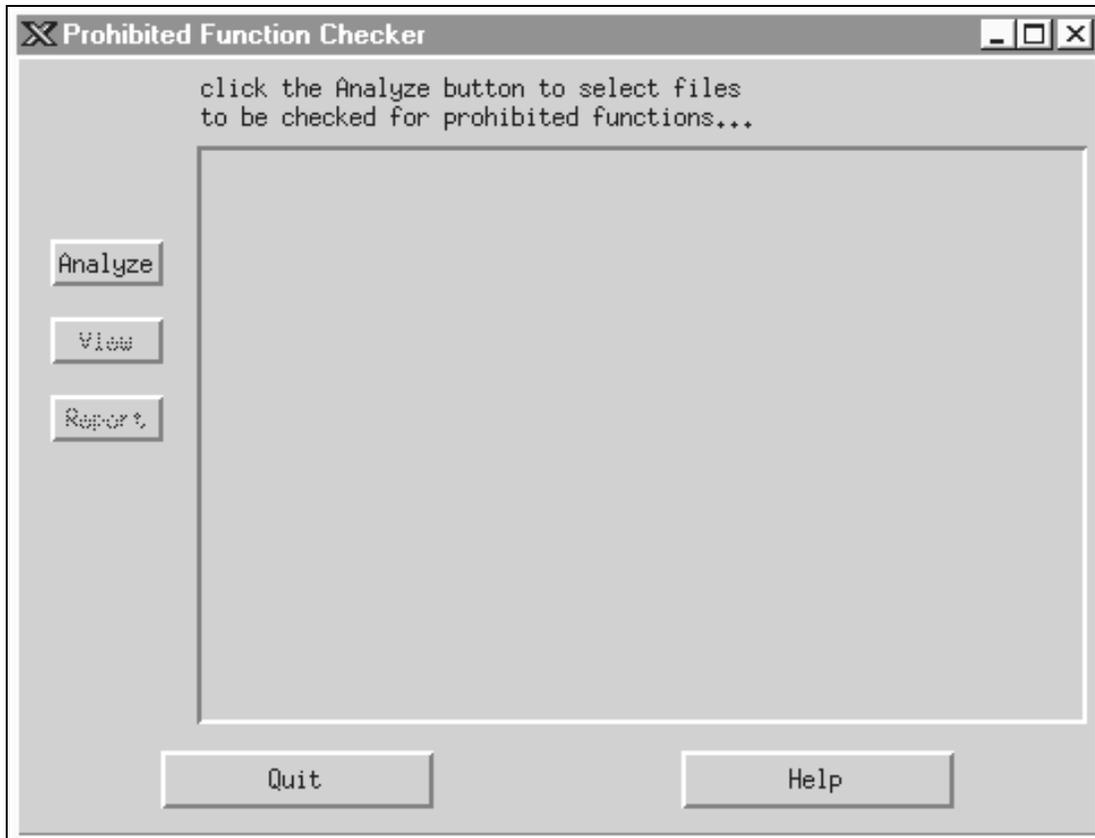


Figure 4.5.1-10. Prohibited Function Checker Pop-up

Table 4.5.1-7 describes the fields on the Prohibited Function Checker Pop-up.

**Table 4.5.1-7. Prohibited Function Checker Pop-up Field Descriptions
(1 of 2)**

Field Name	Data Type	Description
Analyze	Button	Clicking the Analyze button brings up the File Selector Pop-up for selecting files to be checked for prohibited functionality. Following selection of files and analysis, the files containing prohibited functionality are listed in the data area of the Prohibited Function Checker pop-up window.
View	Button	Highlighting a particular file in the data area of the window and then clicking the View button brings up the Source Code Pop-up window shown in Figure 4.5.1-12. Prohibited functionality found in the file is highlighted.
Report	Button	Clicking the Report button brings up the Report Pop-up shown in figure 4.5.1.13. This window allows for display/printing of a summary of the prohibited functionality found.

**Table 4.5.1-7. Prohibited Function Checker Pop-up Field Descriptions
(2 of 2)**

Field Name	Data Type	Description
Data area containing a list of files that contain prohibited functions	File names	The untitled large blank area in the Prohibited Function Checker window contains a list of those files analyzed and found to contain prohibited functionality.
Quit	Button	Clicking Quit terminates the Prohibited Function Checker.
Help	Button	The Help button brings up the Help Pop-up for obtaining an explanation of the use of this tool.

The *File Selector* pop-up, shown in Figure 4.5.1-11, is brought up when the user hits the **Analyze** button on the *Prohibited Function Checker* pop-up. With the *File Selector* pop-up, the user specifies a list of the files to be checked for the presence of prohibited functionality.

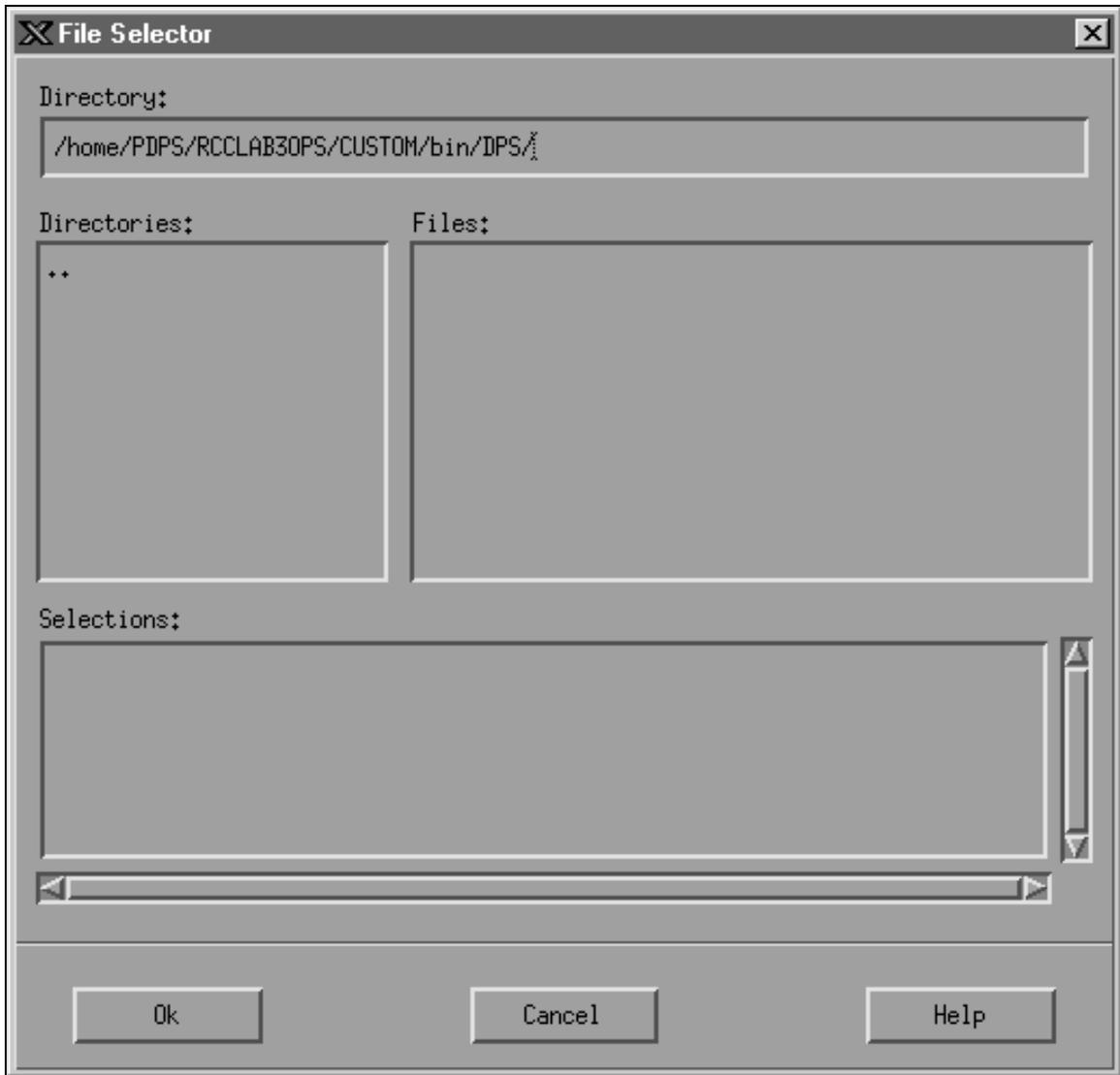


Figure 4.5.1-11. File Selector Pop-up

Table 4.5.1-8 describes the fields on the File Selector pop-up window.

Table 4.5.1-8. File Selector Pop-up Field Descriptions

Field Name	Data Type	Description
Directory:	Directory name	User enters the name of the directory where files to be checked for prohibited functions are located.
Directories:	List of directories	Displays the list of subdirectories within the current directory. Double clicking on a subdirectory results in moving into that directory and displaying its subdirectories in the Directories field and its file content in the Files field.
Files:	File names within directory	Contains the list of files within the currently selected directory. Clicking on a listed file designates it for analysis and moves it to the Selections field. To choose groups of contiguous files, hold down the left mouse button and drag downward. To choose non-contiguous files, hold down the Ctrl key while clicking on desired file names.
Selections:	List of file names	Contains the list of files selected for analysis.
Ok	Button	On building the list of files to be analyzed, click on the Ok button to initiate analysis. The File Selector pop-up disappears and the Prohibited Function Checker pop-up appears with the list of files checked and found with prohibited functionality present.
Cancel	Button	To abort the building of a list of files to be analyzed, click on the Cancel button.
Help	Button	Brings up a Help window explaining File Selector operation.

Following selection of the file(s) to be analyzed and clicking **Ok**, the *File Selector* pop-up disappears, the prohibited function analysis is performed on each source file, and the list of selected files found to contain prohibited functions is displayed on *the Prohibited Function Checker* pop-up.

The results of the *Prohibited Function Checker* (see Figure 4.5.1-10) can be examined using the **View** button (only one source file can be selected at a time to view the code) or **Report** button. To initiate a view, highlight one of the files listed and click on the **View** button. The *Source Code* pop-up, shown in Figure 4.5.1-12, appears.

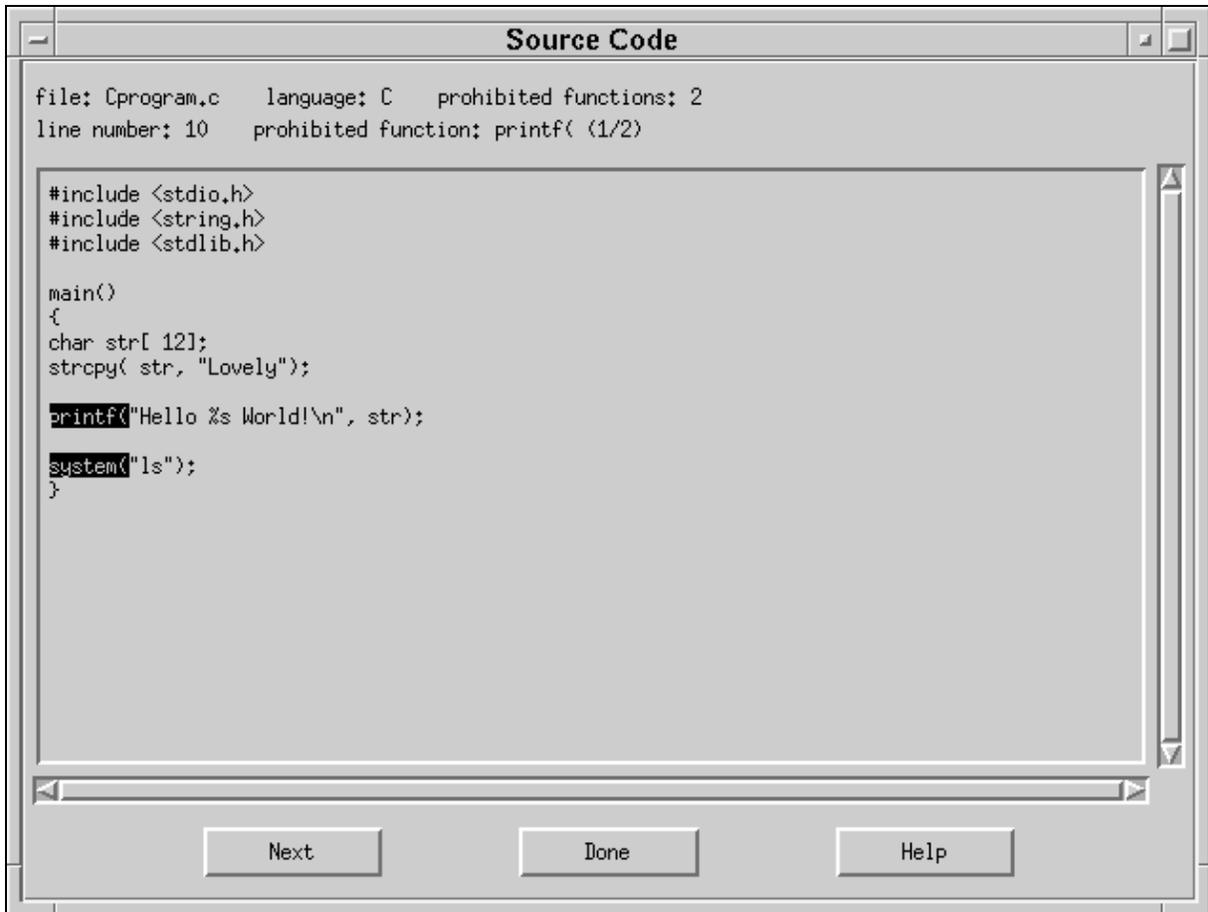


Figure 4.5.1-12. Source Code Pop-up

As shown in the *Source Code* pop-up, occurrences of prohibited functions found in the source code are highlighted. Click on the **Next** button to bring into the window successive occurrences of prohibited functions within the same source file. Click on the **Done** button to close the *Source Code* pop-up. Click on the **Help** button to explain the use of the window. All prohibited functions can be viewed similarly, by examining one source file at a time.

Use the **Report** button on the *Prohibited Function Checker* pop-up to display the *Report Pop-up* shown in Figure 4.5.1-13. It displays for each source file, a list of prohibited functions found.

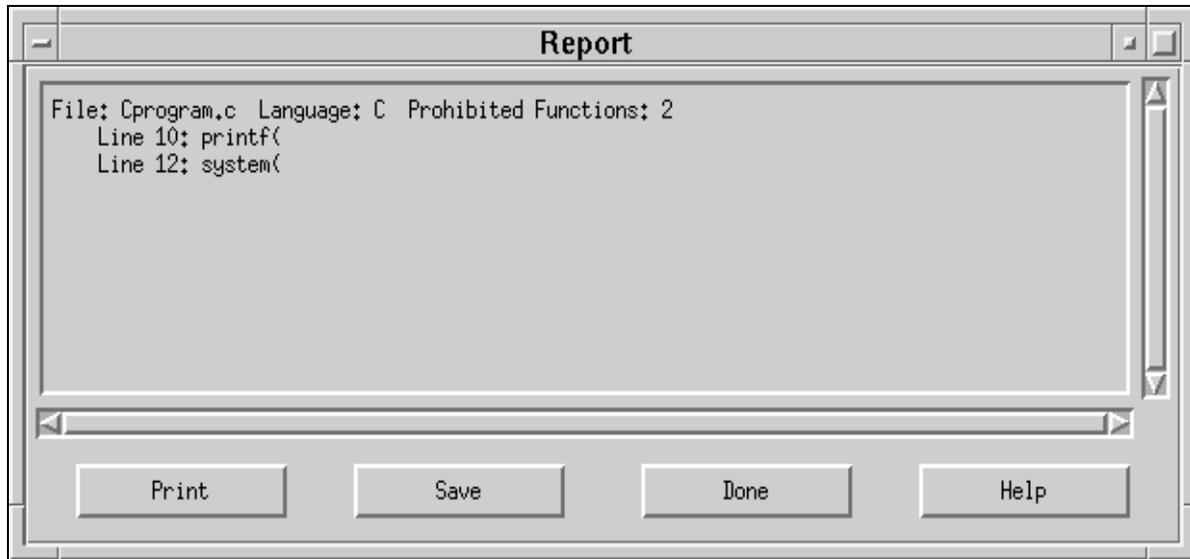


Figure 4.5.1-13. Report Pop-up

The results of the prohibited function analysis showing prohibited functions can be saved or printed from the *Report Pop-up*.

Clicking on the **Save** button brings up the *Save To File Pop-up* for specifying where to save the report.

Clicking on the **Print** button prints the results on the default printer.

Clicking on the **Done** button will close the *Report Pop-up*.

The **Help** button within any window brings up the HELP pop-up explaining the use of the window.

4.5.1.2.5.3 Process Control File Checker

This section describes how the SSI&T Specialist uses the Process Control File (PCF) Checker tool during the SSI&T process.

The PCF Checker verifies that the PCF provided by the Science Software development teams, as part of the Delivery Archive Package (DAP), is syntactically correct.

Before this program is used, the following condition must be met:

The PCFs to be checked are available, accessible, and have read permission for the user.

4.5.1.2.5.3.1 Process Control File Checker

The PCF Checker Pop-up allows the user to select a PCF and have it scanned by Science Data Processing (SDP) Toolkit routines for any irregularities such as format or structure.

4.5.1.2.5.3.2 Quick Start Using This Tool

The Process Control File (PCF) Checker tool, shown in Figure 4.5.1-14, is started from the SSI&T Manager by selecting the Tools-> Standards Checker-> *Process Control File Checker* option as shown in Figure 4.5.1-8.

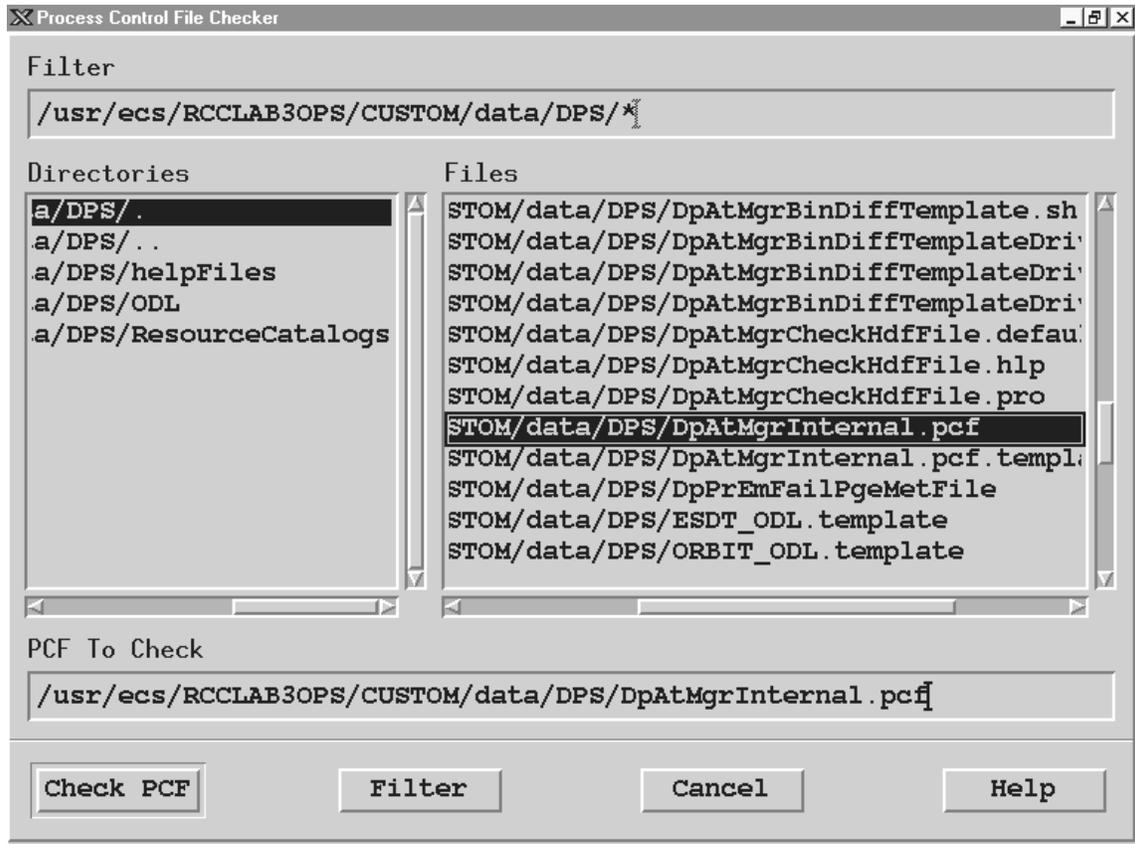


Figure 4.5.1-14. PCF Checker Pop-up

Table 4.5.1-9 describes the PCF Checker's fields.

Table 4.5.1-9. PCF Checker Field Description

Field Name	Data Type	Size	Entry	Description
Filter	ASCII characters	N/A	Read from configuration file/ user input (optional).	Displays the selected directory (from the Directories field).
Directories	ASCII characters	N/A	Read from database.	Displays the existing directories in the database.
Files	ASCII characters	N/A	Read from database (local disk).	Displays the files existing in the selected directory.
PCF To Check	ASCII characters	N/A	Read from database (local disk).	Displays the selected file to be checked.

The PCF Pop-up allows the user to work through the directory structure on the local machine in order to select PCFs to be checked, by selecting the desired directory in the Directories window. The Filter window allows users to limit the files displayed. From the File window, the user can select the PCF file to be checked. Only one PCF can be checked at a time. Selecting the **Check PCF** button initiates the checking process on the PCF specified in the PCF To Check window. The result is displayed in a Pop-up labeled PCF Checker Results, Figure 4.5.1-15. Clicking on the **Filter** button will display the selected directory from the Directories field as specified in the Filter field. Clicking the **Cancel** button will terminate the PCF Checker. Clicking the **Help** button will explain the use of the window.

If the selected file does not exist, an error dialog is displayed to inform the user. Similarly, if the specified PCF exists but cannot be opened, an error dialog pops up to inform the user. Dialogs also pop up to inform the user of any fatal system errors or errors encountered in calls to toolkit functions.

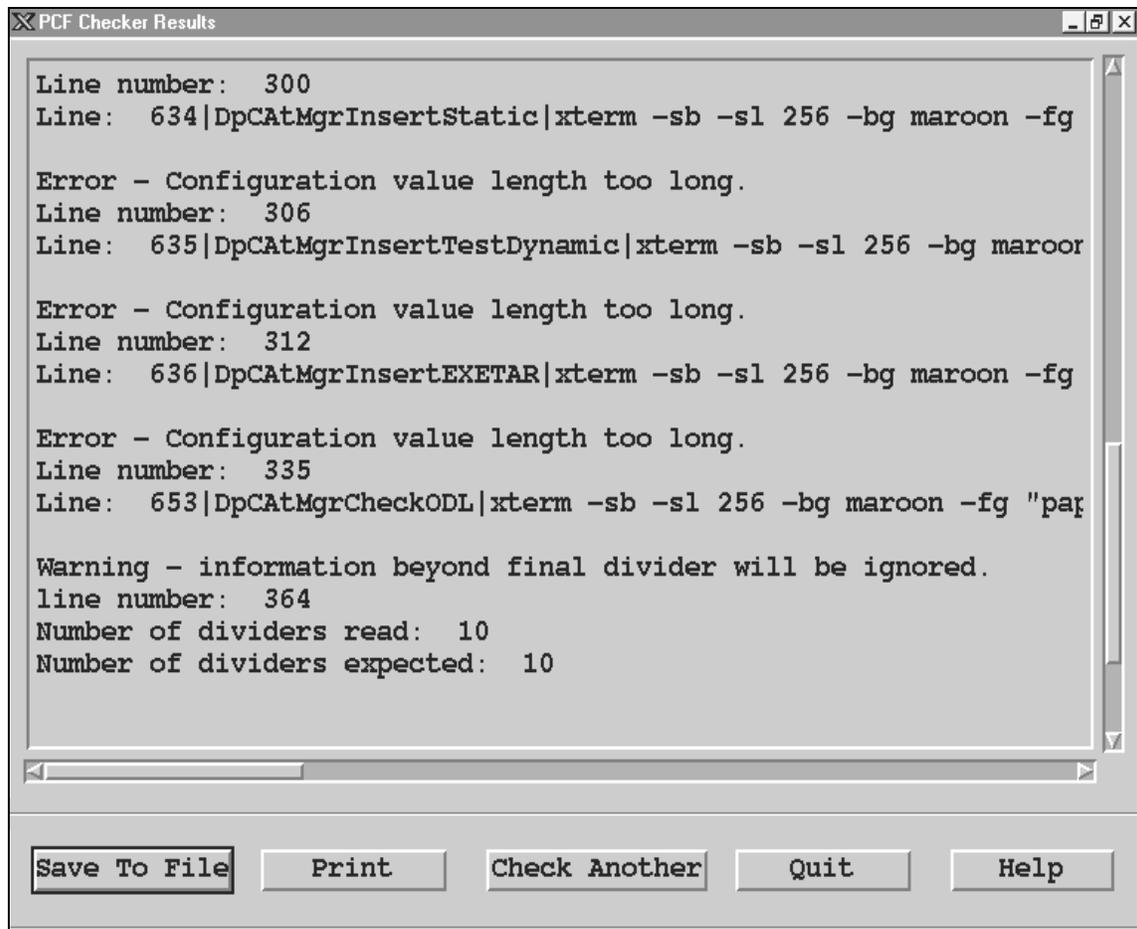


Figure 4.5.1-15. PCF Checker Results Pop-up

From the PCF Check Result Pop-up the user has options to save the results to a file, send the results to the default printer, check another PCF, or quit the Results Pop-up. Selecting the Save button brings up a pop-up labeled Save (shown in Figure 4.5.1-16) that allows the user to specify the directory and file name in which to save the results file.

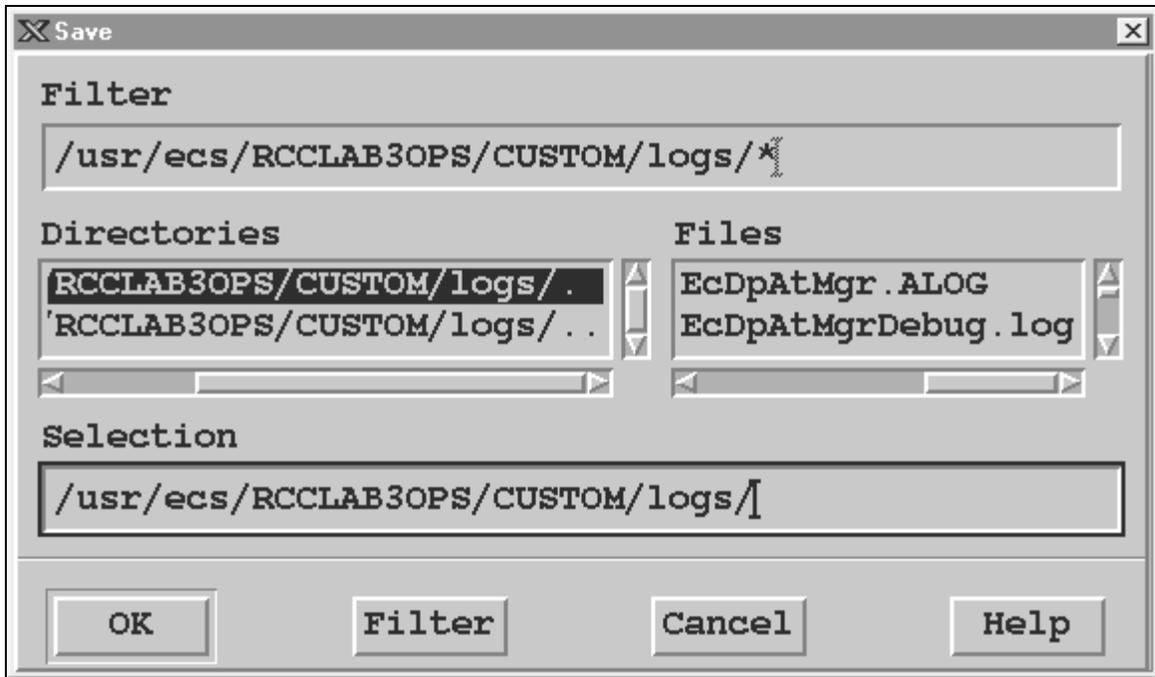


Figure 4.5.1-16. The PCF Checker Save Pop-up

The PCF Check Save window displays a default output directory and its contents on the file selector. The description fields of the Save Pop-up are identical to the PCF Checker Pop-up's (see Table 4.5.1-14). The user can select a listed file or can enter his/her own selection. Selecting the **Filter** button brings up another screen allowing the user to limit the number of files listed. Selecting the **OK** button will save the files to the specified directory. Figure 4.5.1-17 shows the filter screen.

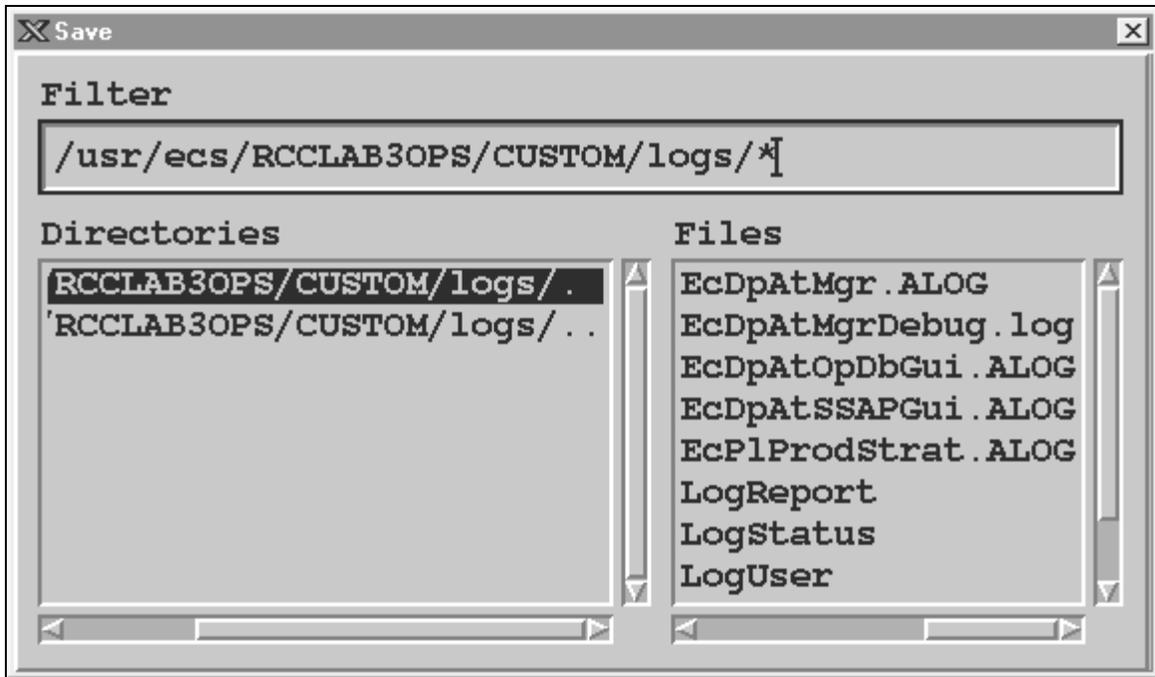


Figure 4.5.1-17. The Pop-up Display for Filtering Files from the Save Pop-up

Messages are displayed to inform the user of successful/unsuccessful completion of save operations.

The Cancel button terminates the running (open) application.

The Help button in all of the PCF Checker Pop-ups provides a description of all the functional buttons for the pop-up. For example, Figure 4.5.1-18 shows the Help on PCF Checker Results screen.

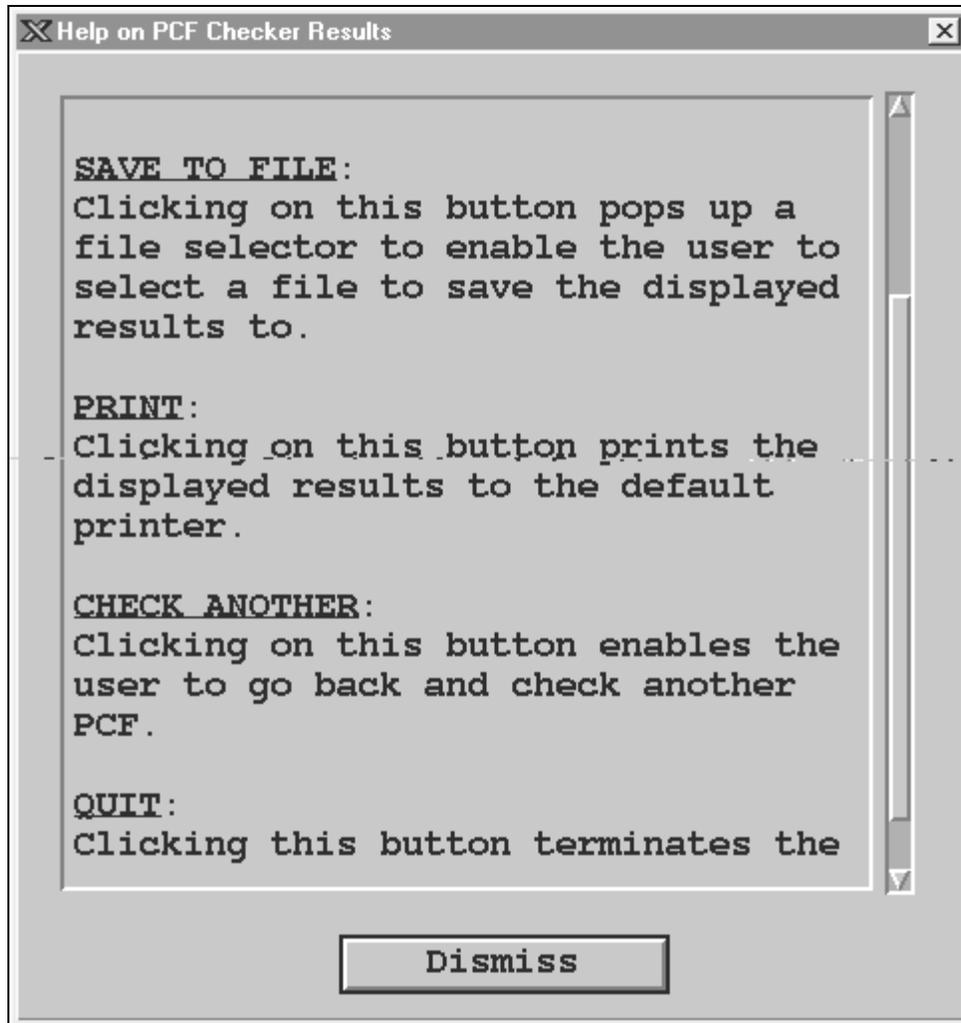


Figure 4.5.1-18. The Help on PCF Checker Results Pop-up

4.5.1.2.5.4 Prolog Extractor

The Prolog Extractor extracts prologs from science software source code modules.

By default, these prologs are assumed to be surrounded by standard delimiters, according to the official NASA/ESDIS standards document, *Data Production Software and Science Computing Facility (SCF) Standards and Guidelines, Rev A, October 1996, 423-16-01*. The starting delimiters are listed in Table 4.5.1-10.

Table 4.5.1-10. Prolog Extractor Standard Delimiters

Language	Type	Delimiter
FORTRAN 77	source	!F77
FORTRAN 90	source	!F90
C	source	!C
Ada	source	!ADA
FORTRAN 77	include	!F77-INC
FORTRAN 90	include	!F90-INC
C	include	!C-INC
any language	any	!PROLOG

The end delimiter is always !END.

Other delimiters can be specified through use of command line flags (see below).

If the module contains no prologs at all, the Prolog Extractor program returns a warning. It does not warn of functions within a particular file missing prologs.

In order for this tool to work, the file must have one of the file extensions listed in Table 4.5.1-11.

Table 4.5.1-11. Prolog Extractor File Extensions

Language	extension
FORTRAN77:	f, f77, ftn, for, F, F77, FTN, FOR
FORTRAN90:	f90, F90, f, F
FTN. INCLUDE:	inc, INC
C:	c
C include	h
Ada:	a, ada

The Prolog Extractor is invoked from the SSI&T Manager under the TOOLS menu item Standards Checkers.

Output from this tool is always written to a file.

By default, the SSI&T Manager internal PCF specifies that this output file be named prologs.txt in the local directory. You can change this by editing the internal PCF.

The PCF value of the output filename is overridden by the "-o" option on the command line.

4.5.1.2.6 Product Examination

Product examination tools are used to view, examine, or compare files used in science processing. Figure 4.5.1-19 shows the Product Examination tool menu options on the SSI&T Manager Main screen. These tools assist in the maintenance of the science software by allowing

the operator to view programs in different languages, view science data, or compare program or data output files.

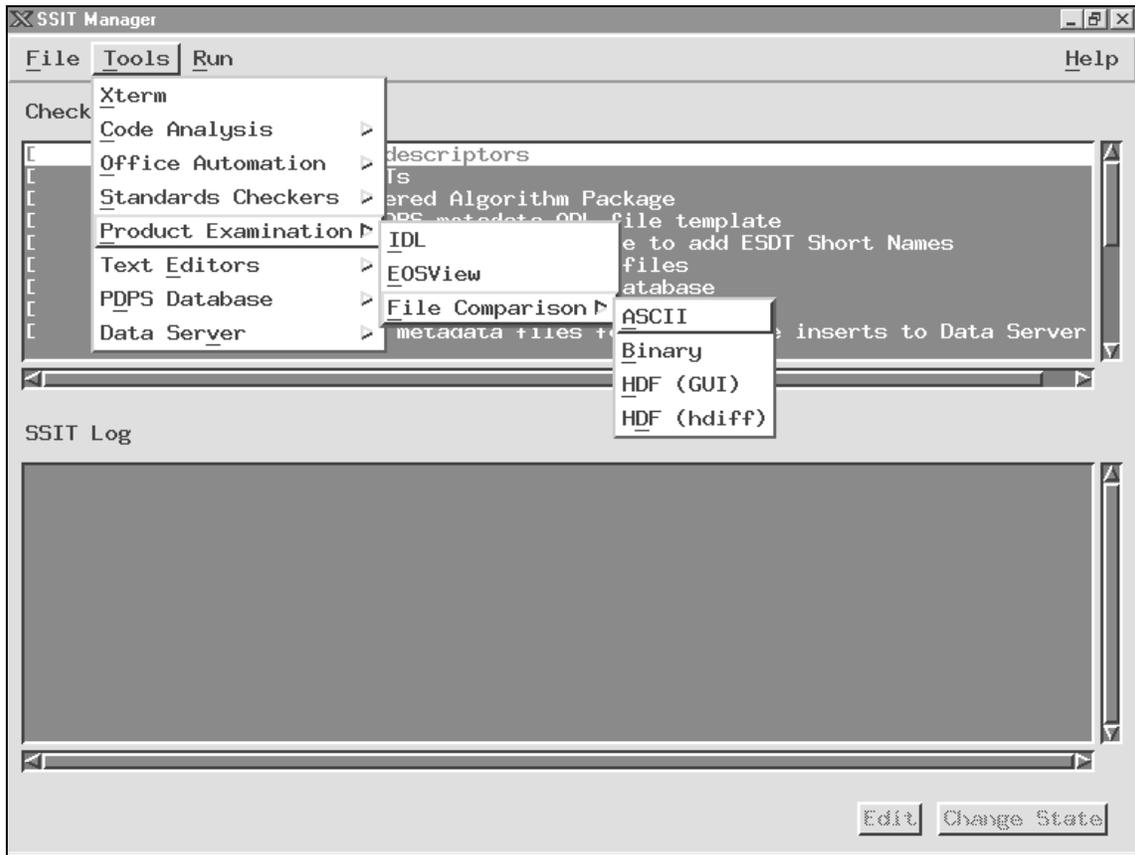


Figure 4.5.1-19. SSI&T Manager Main Screen with Product Examination Tools Displayed

4.5.1.2.6.1 IDL

IDL does not provide a GUI. IDL is a powerful visualization package in common use throughout the science community. This COTS package includes a language, also called IDL, providing capabilities for data visualization and comparison of various file formats. IDL is provided so DAAC operators can write data visualization and comparison tools, and execute them on an "as needed" basis.

The IDL language is an interpreted computer language, allowing applications (written in IDL) to look at binary (unformatted) files as well as HDF files.

Once inside this environment, user written programs can be compiled and executed.

Documentation for IDL is obtained by typing `idlhelp` on the UNIX command line.

4.5.1.2.6.2 EOSView

EOSView is a custom Hierarchical Data Format (HDF) file verification tool. It is for use by anyone who wishes to verify or inspect EOS data products in HDF EOS or HDF format. Users include EOS instrument team science software developers and data product designers, DAAC personnel, and end users of EOS data products, i.e., scientists and researchers.

EOSView is described in Section 4.12.5, “EOSView.”

4.5.1.2.6.3 File Comparison Tools

This section describes how the SSI&T specialist uses the File Comparison tools during the SSI&T process.

The File Comparison tools are selected from the Tools item on the menu bar, and appear under the Product Examination option. Four alternatives are then offered:

- *ASCII* – Text file comparison tool; X windows based xdiff utility
- *Binary* – Custom development environment which assists in the generation of DAAC-written code which compares binary files
- *HDF (GUI)* -- IDL program which compares HDF format files
- *HDF (hdiff)* – “hdiff” CHUI (CHaracter oriented User Interface) utility which compares HDF format files, written by MODIS

These tools are most frequently used to insure the data products generated from the PGEs run in the SSI&T environment are identical (within tolerance) to the files produced by the PGEs at the SCF.

4.5.1.2.6.3.1 ASCII File Comparison Tool

When the ASCII File Comparison tool is started by selecting the **T**ools->**P**roduct Examination->**F**ile Comparison->**A**SCII menu option from the SSI&T Manager, an Xterm is spawned, which prompts the user for input. The user enters two ASCII filenames, and the differences between the files are displayed. The comparison tool is based on the X-Window application, “xdiff”.

4.5.1.2.6.3.2 Binary File Difference Environment

When the Binary File Difference Environment is started by selecting the **T**ools->**P**roduct Examination->**F**ile Comparison->**B**inary menu option from the SSI&T Manager, the pop-up shown in Figure 4.5.1-20 is displayed. The button functions are described below in the order they should be used.

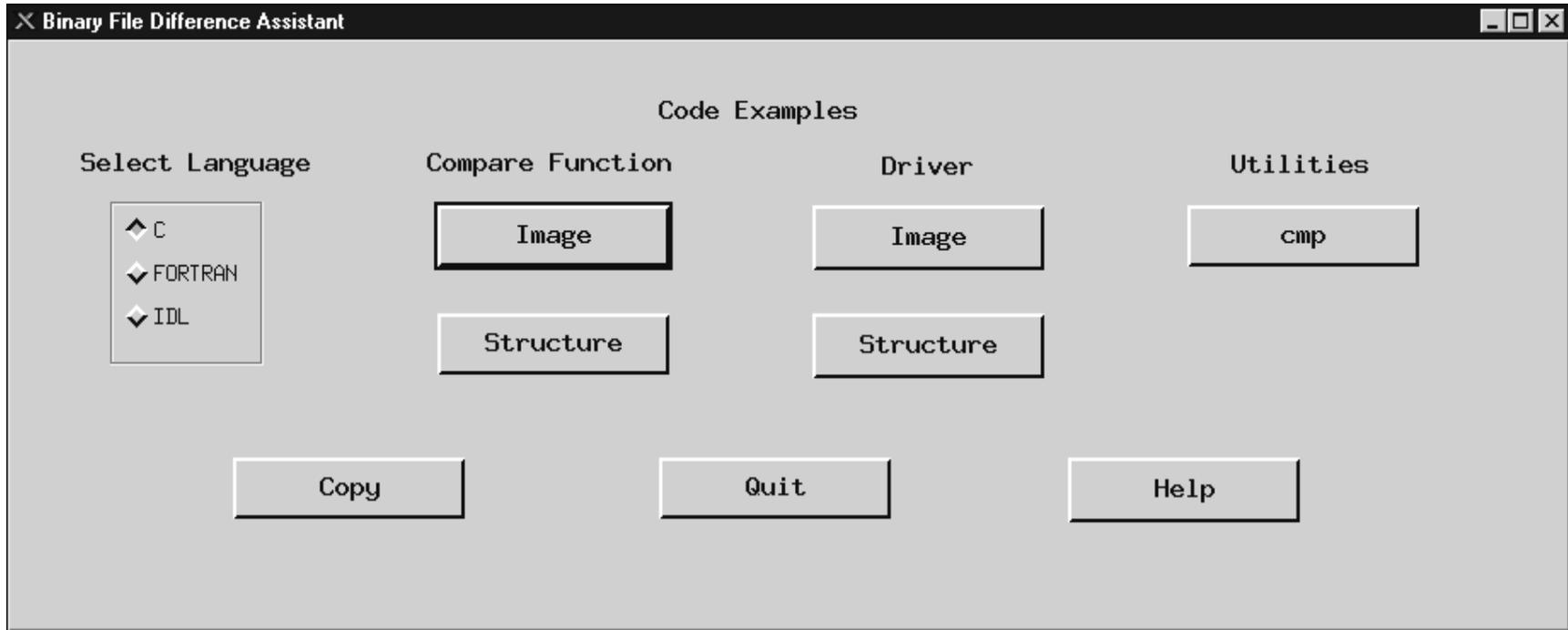


Figure 4.5.1-20. Binary File Comparison Pop-up

The Binary File Difference Assistant assists the user in development of custom code that compares two binary files.

Since there are an unwieldy number of possibilities for binary file formats, this tool cannot compare two binary files without some custom code written at the DAAC, hence, the “Assistant” in the name.

- The **Select Language** box provides a choice of languages.
- The choice depends largely on preference.
- It does not necessarily have to be the language used to create the files being compared. The user clicks “**COPY**” to generate three files: a main driver code module, a template comparison code module, and a makefile in the selected language [C, FORTRAN 77 or Interactive Data Language (IDL)].
- The operator then edits this template to add code that compares files for the specific binary file format, compiles the code, and executes it as appropriate; these operations are all independent of the pop-up.

The **Code Examples** section of the pop-up window displays simple code examples about binary file comparison. The operator can copy and paste these in their code if desired. Examples have been compiled and tested. This sample code is provided to help the operator in visualizing a completed comparison module. This code is displayed when the user presses one of the “**Compare Function**” buttons, either **Image** or **Structure**. **Image** displays an example of binary_file comparison code for files containing images. **Structure** displays an example of binary file comparison code for files containing arrays of structures, or records.

Also provided is sample driver code, when the operator presses a **Driver** button. Finally, code for a simple compare utility is provided for reference, when the user presses **cmp**.

4.5.1.2.6.3.3 HDF GUI

All the ECS standard data products are generated in HDF format. A tool is provided to compare HDF data files obtained running some algorithm(s) in a testing versus an operational processing environment. The HDF file difference tool is invoked from the SSI&T Manager by selecting **Tools->Product Examination->File Comparison->HDF (GUI).**

The tool is also available from the UNIX command line by typing *EcDpAtCheckHdfFile*. Both methods result in the display of the screen shown in Figure 4.5.1-21 (*note that this program is written in IDL, so it invokes the IDL environment to run.*)

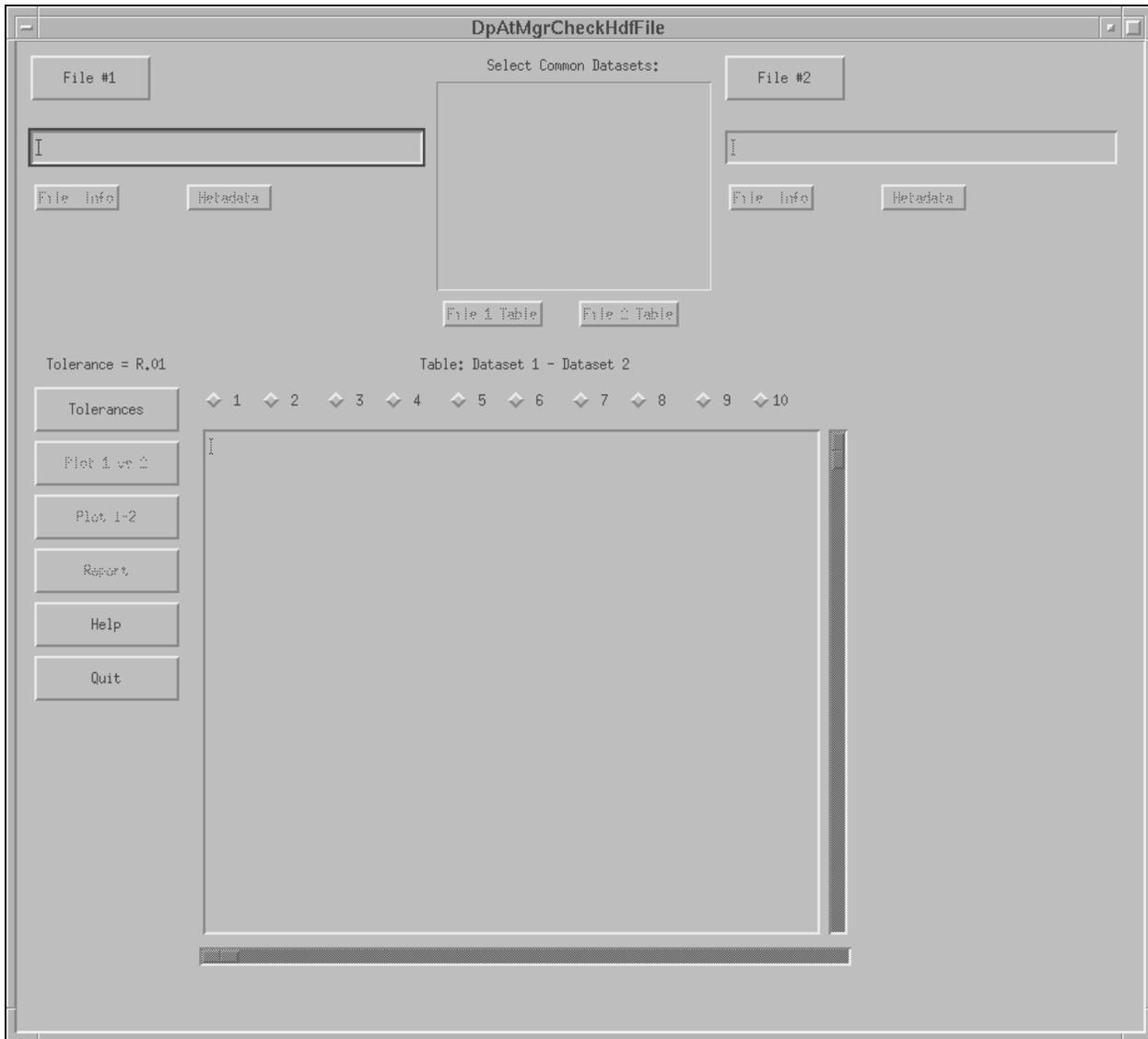


Figure 4.5.1-21. HDF Pop-up

The HDF File Difference Tool Pop-up allows the operator to compare and contrast 2 HDF files via a graphic user interface. The button functions on this pop-up are:

- Clicking the File #1 or File #2 buttons does the selection of the files to be compared. This selection activates the File Selection Dialogue Pop-up shown in Figure 4.5.1-22. The appropriate filename is displayed in the Selection textbox of the Dialogue Pop-up. The “Ok” button returns to the HDF File Comparison Pop-up and inserts the selected filename in the originally activated file selection display textbox.

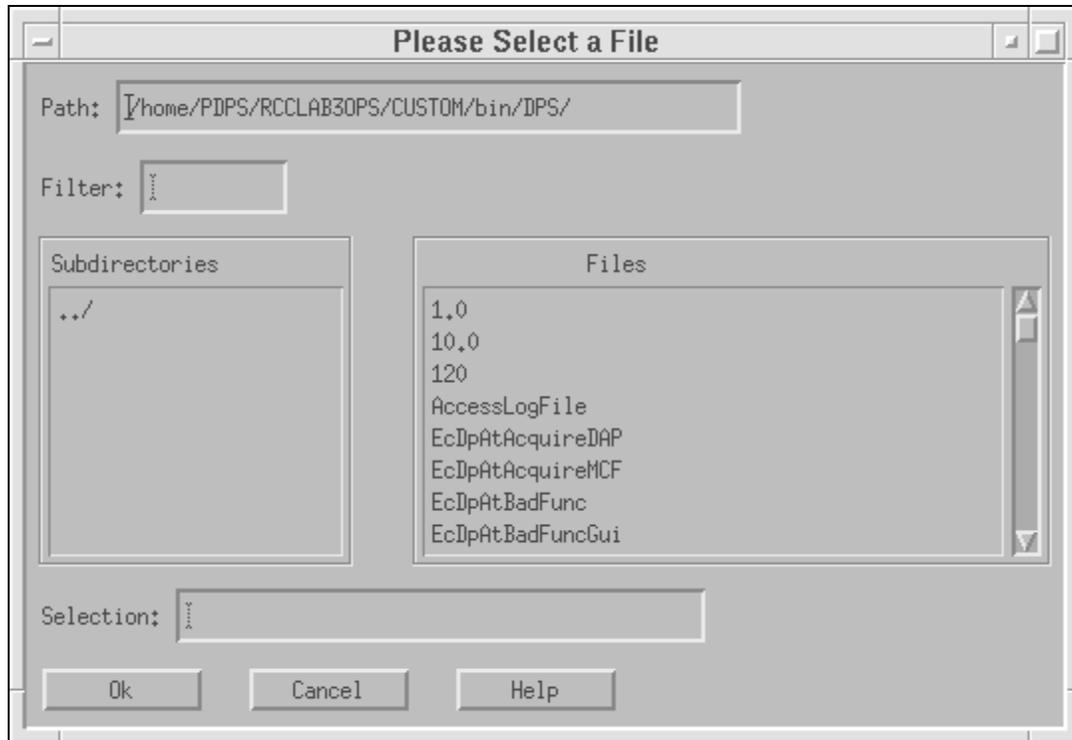


Figure 4.5.1-22. File Selection Dialogue Pop-up

- For each selected HDF file, **File Info** and Associated **Metadata** can be displayed by clicking on the available buttons.
- The tool also provides the facility to select the **Common Datasets** of both files from the list available in the center upper part of the screen.
- Once the data sets are identified, the operator selects the type of tolerance (relative or absolute) used to compare the numerical values of the two data sets. To accomplish this, click on the **Tolerances** button (Figure 4.5.1-21) and type the appropriate value of tolerance as shown in Figure 4.5.1-23.

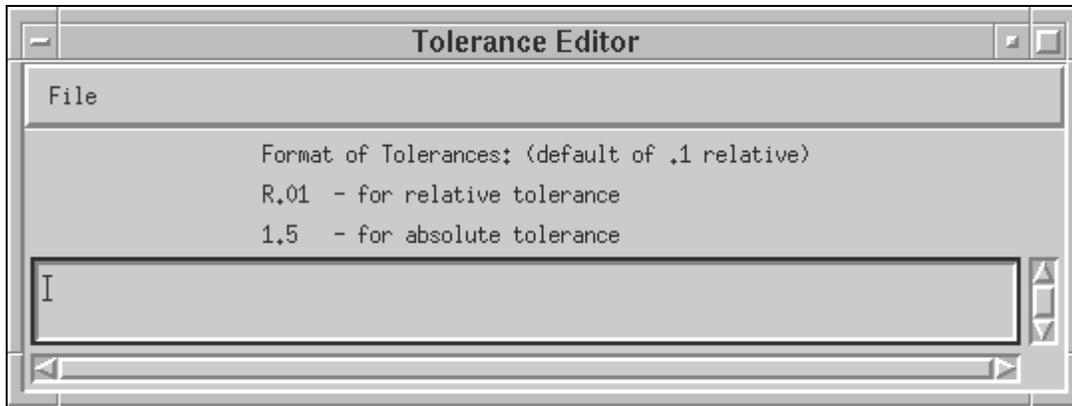
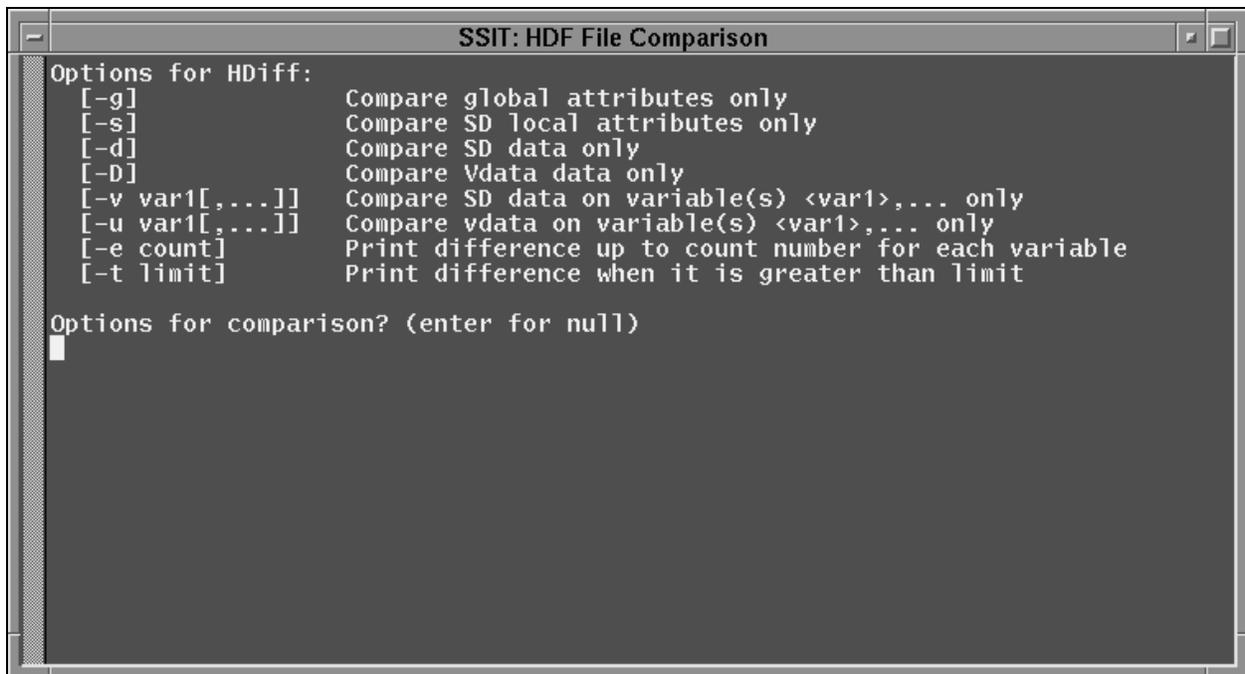


Figure 4.5.1-23. Tolerance Editor Pop-up

- The HDF pop-up allows the operator to input comparison coefficients (shown in Figure 4.5.1-21 as radio buttons **1** through **10** next to the **Tolerances** button on the HDF File Comparison Pop-up).
- Finally, two types of graphs can be selected by the operator by clicking on either one of the following buttons:
 - **Plot 1 vs. 2** shows a plot with both data sets.
 - **Plot 1-2** shows the difference between the two data sets.
- Documentation for the HDF file difference tool is obtained by clicking on the **Help** button on the main screen.
- Clicking the **Report** button will generate a report of the HDF file comparison results.
- Clicking on the **Quit** button closes this Tool.

4.5.1.2.6.4 HDF File Comparison - hdiff

The HDF File Comparison hdiff tool is started from the SSI&T Manager screen by selecting the **Tools->Product Examination->File Comparison->HDF (hdiff)** option. An Xterm appears prompting the user for input. There is no graphics screen for this function. It is run through the command line interface. The operator is also provided with a list of options for different kind of comparisons the tool can perform on HDF files (See Figure 4.5.1-24). After the operator enters two HDF filenames, the differences between the files are displayed.



```
SSIT: HDF File Comparison
Options for HDiff:
[-g]          Compare global attributes only
[-s]          Compare SD local attributes only
[-d]          Compare SD data only
[-D]          Compare Vdata data only
[-v var1[,...]] Compare SD data on variable(s) <var1>,... only
[-u var1[,...]] Compare vdata on variable(s) <var1>,... only
[-e count]    Print difference up to count number for each variable
[-t limit]    Print difference when it is greater than limit

Options for comparison? (enter for null)
█
```

Figure 4.5.1-24. HDF (hdiff) Options

4.5.1.2.7 Text Editors

The Text Editors menu provides tools to assist operators in documentation and administrative tasks (Figure 4.5.1-25).



Figure 4.5.1-25. SSI&T Manager Main Screen with Text Editors Tool Menu Displayed

4.5.1.2.7.1 Emacs

Emacs, as shown in Figure 4.5.1-26, is a COTS product providing SSI&T users with Editor capabilities. More information about Emacs can be downloaded from the web at <ftp://prep.ai.mit.edu/pub/gnu/>.

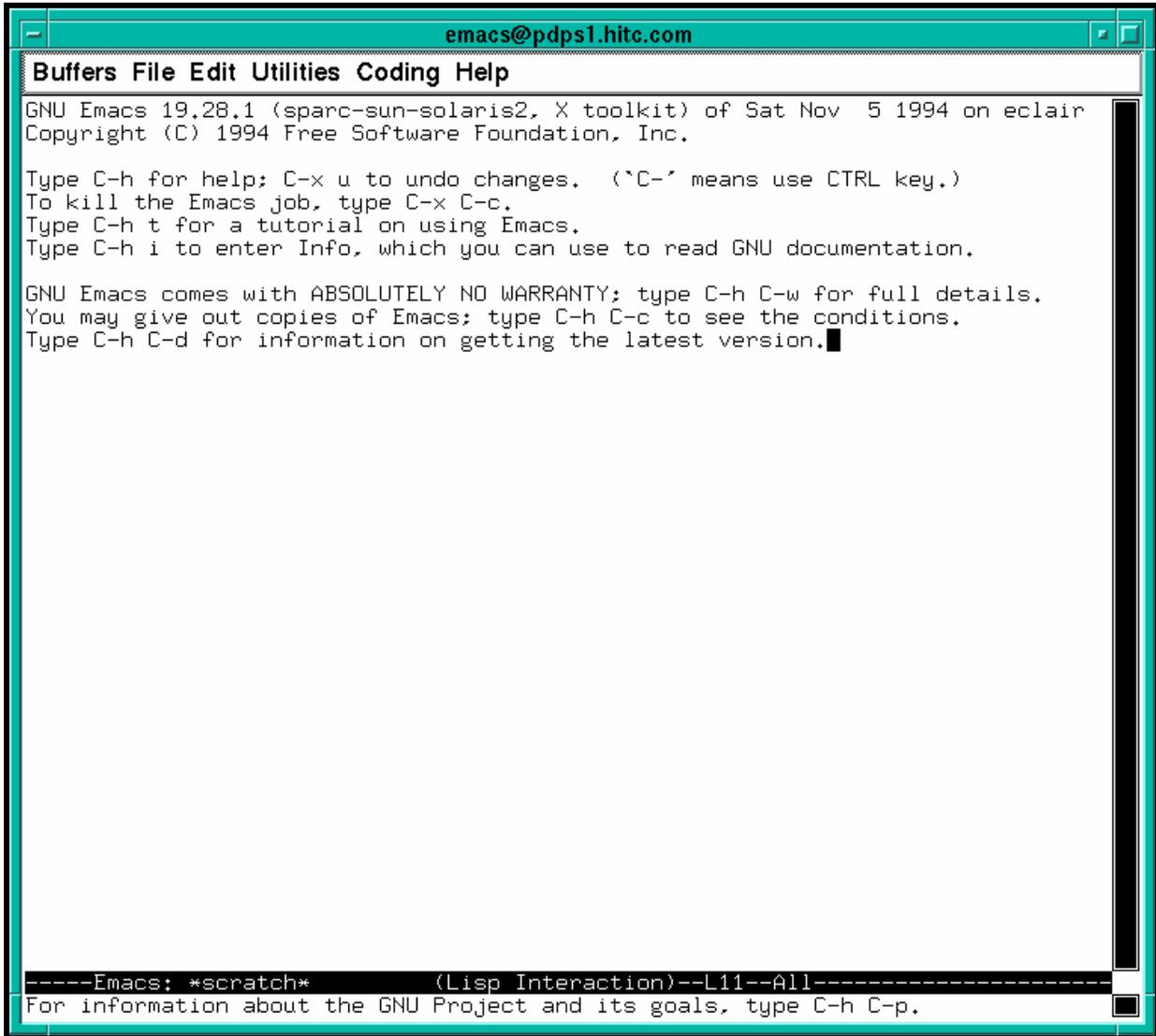


Figure 4.5.1-26. Emacs Pop-up

4.5.1.2.7.2 Xedit

Xedit, shown in Figure 4.5.1-27, is a COTS product that provides SSI&T users with file editing capabilities. For a detailed description of the Xedit tool, invoke the Unix man(ual) command:

>**man xedit**



Figure 4.5.1-27. Xedit Pop-up

4.5.1.2.8 PDPS Database

The PDPS Database Update Tools are accessed using the **Tools->PDPS Database** menu item of the SSI&T Manager as shown in Figure 4.5.1-28. These tools are used to register a PGE with the PDPS database. Critical information about the PGE and under what conditions it should be run is captured. This information is used by the PDPS prior to and during PGE execution, in both the test environment and the production environment.

The PDPS Database Update Tools consist of four components;

1. PCF ODL Template
2. Check ODL
3. SSIT Science Metadata Update
4. SSIT Operational Metadata Update (a.k.a. PGE Registration Pop-up).

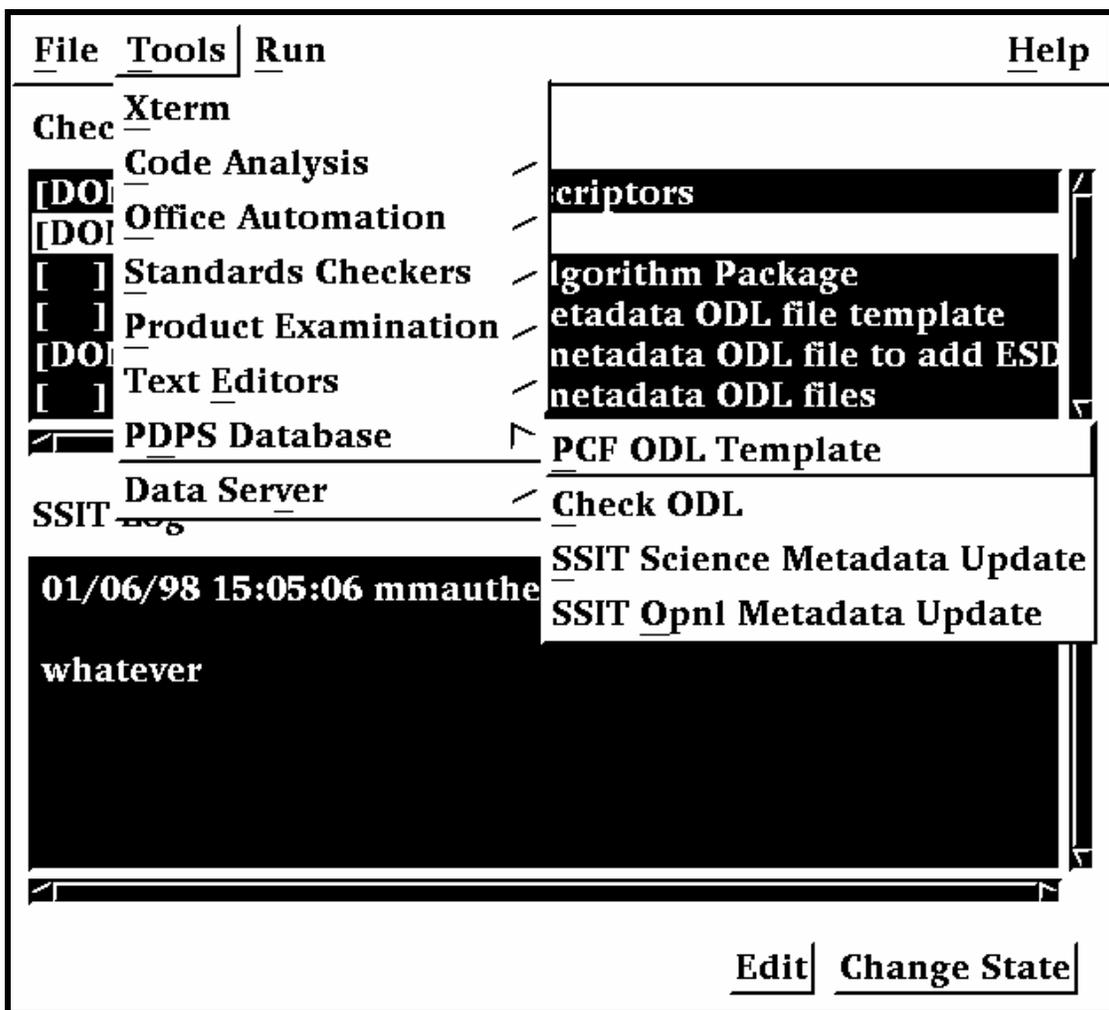


Figure 4.5.1-28. SSI&T Manager with PDPS Database Tool Menu Displayed

All programs can be started either from the **Tools->PDPS Database** menu available on the SSI&T Manager or by as executing the proper command line, as specified in the following sections.

4.5.1.2.8.1 PCF ODL Template and ODL file Checking

The first step in the process of updating the PDPS Database is to generate a template SSI&T PGE SCIENCE metadata ODL file from the PCF delivered with the science software. This is accomplished by selecting *PCF ODL Template* from the *PDPS Database* submenu.

A character based window opens where the user is prompted for various inputs, the configuration filename, the ECS mode of operations, the name of the delivered science software Process Control File (PCF), the PGE name, the PGE version, and the Profile ID.

The tool is also available by sourcing - in the executable directory *\$ECS_HOME/<MODE>/CUSTOM/bin/DPS* - the file named *buildrc*.

After the template file is generated, the SSI&T operator must edit this file, adding all information currently blank, as needed. In particular, each PCF file entry must be associated with one or more ESDTs.

The Check ODL Script, available through the SSI&T Manager by selecting the Tools->PDPS Database->Check ODL option, can check the edited file in the previous step.

The edited file is then used as an input to the next step (see Section 4.5.1.2.8.3) of the PDPS Update process.

4.5.1.2.8.2 Check ODL

The Check ODL program parses the Science Software Metadata ODL files and reports any errors found. It is used to validate ODL (make sure there are not any errors) before the SSI&T Science Metadata Update Tool is executed to populate the Science Software Metadata information in the PDPS database.

The ODL Checker program is command line driven.

The program's prompts and explanations of user responses to these prompts follow below. Note if there is a default value for an entry (in most cases this only occurs if you run the program more than once), it appears at the end of the prompt line.

**** SCIENCE Metadata ODL Files Checker Script ****

Configuration filename? (hit 'enter' for default: *.././cfg/EcDpAtCheckODL.CFG*)

In most cases hitting enter (for the default) is fine. If not, enter the correct configuration filename including full path:

ECS Mode of operations?

This is the mode (i.e. OPS, TS1) in which the tool executes. In most cases this is TS1.

PGE name (max 10 characters)?

This is the name of the PGE for which you want to check the ODL. It has a maximum length of 10 characters and must be part of the ODL file name and inside the ODL file. See PGE Metadata ODL File for more information about naming and filling out the PGE ODL file.

PGE version (max 5 characters)?

This is the version of the PGE for which you want to check the ODL. It has a maximum length of 5 characters and must be part of the ODL file name and inside the ODL file. Again, see PGE Metadata ODL File help page for more information about naming and filling out the PGE ODL file.

PGE Profile ID (0-99, 0 means null)?

This is the profile ID of the PGE who's ODL you want to check. It is an integer with a range 0-99 and must be part of the ODL file name and inside the ODL file. Again, see the PGE Metadata ODL File help page for more information about naming and filling out the PGE ODL file.

Any success or error messages are then displayed followed by:

Hit return to run again, 'q {return }' to quit:

This allows you to check another set of ODL files (just hit enter) or quit the program (enter q).

Check ODL program output.

The Check ODL program searches for any missing or invalid ODL parameters. It reports errors with the following message:

Check of PDPS Science Metadata was unsuccessful: missing or invalid data in ODL files

What follows is a list of missing or invalid parameters and why they were flagged as invalid. In many cases the parameter value is out of the expected range or is longer than the allowed maximum. For some errors, the Check ODL program cannot narrow the problem to its cause and points the user to the Toolkit Error logs as follows:

DpAtCheckOdlDpAtScienceMd::ProcessOdlFile (PGE):

ODL Error in file /ecs/formal/PDPS/DPS/SSIT/src/Metadata/Test/PGE_BAD#ODL#09.odl

Check LogStatus file

Look in the LogStatus file in the /usr/ecs/{MODE}/CUSTOM/logs directory or in the user-installed directory if you have installed a personal copy of SSI&T.

If the required ODL file(s) cannot be found, the tool reports:

DpAtCheckOdlDpAtScienceMd::ProcessOdlFile (PGE):

Unable to open PGE Science Metadata ODL file

{FILENAME}

{FILENAME} is replaced by the filename the tool was searching for. If this message appears, check the configuration settings (see Preconditions to running the Check ODL program) and the existence of the ODL file.

If the check of the ODL succeeds, the user is greeted with the message:

```
***** Check of PDPS SCIENCE metadata SUCCESSFUL *****
```

```
NO ERRORS FOUND
```

4.5.1.2.8.3 SSI&T Science Metadata Update

The next step is to update the PDPS/SSI&T database with science metadata, where the latter is defined as PDPS metadata coming from the Instrument Teams (ITs), and rarely changes. Selecting SSI&T Science Metadata Update does this from the *PDPS Database* submenu

The PGE template ODL file, which is output from the *PCF ODL Template* program of the previous section, is input to this program, after it has been edited and renamed.

An example of this file (before editing) is available in:

```
$ECS_HOME/CUSTOM/data/DPS/PGE_ODL.template .
```

The comments in the example file explain fully what data are needed.

The naming convention that should be used for the edited file is explained in the provided Help files. The location of the edited file must be set in the configuration file parameter **DPAT_PGE_SCIENCE_MD**.

An example of this file can be found in:

```
$ECS_HOME/CUSTOM/data/DPS/PGE_ODL.template .
```

In addition, if it does not already exist, a PDPS/SSI&T ESDT SCIENCE metadata ODL file must be created, one for each ESDT used by this PGE. Again the naming convention to be used for this file is documented in the Help files. The location of the edited file must be set in the configuration file parameter **DPAT_ESDT_SCIENCE_MD**. An example of this file is provided in:

```
$ECS_HOME/CUSTOM/data/DPS/ESDT_ODL.template .
```

The comments in the example file explain fully what data are needed.

In addition, if it does not already exist, a Production Rule ODL file must be created for this PGE. Again the naming convention to be used for this file is documented in the Help files. The location of the edited file must be set in the configuration file parameter **DPAT_RULE_SCIENCE_MD**. An example of this file is provided in:

```
$ECS_HOME/CUSTOM/data/DPS/ESDT_ODL.template .
```

The comments in the example file explain fully what data are needed.

Once these files (PGE, ESDT, and RULE) have been created and all the blanks have been completely filled in, the user selects *SSI&T Science Metadata Update* from the *PDPS Database*

submenu to run the program. If a PGE already exists, the operator is prompted to update it. If a PGE does not already exist, a character-based window opens and prompts the user for information needed to identify the PGE, configuration filename, ECS mode of operations, PGE name, version and profile ID. With this information the program updates the SSI&T version of the PDPS database with PGE and ESDT Science metadata read from the ODL files.

4.5.1.2.8.4 SSI&T Operational Metadata Update

The SSI&T Database can be updated with PGE operational metadata.

This pop-up is used to view/update operational parameters for a particular PGE. Operational Parameters include the performance and the resource requirements for each PGE. In addition users can also view/update PGE user-defined runtime parameter descriptions. Other options include viewing the PGE science metadata file. (The program looks in the configuration file for the **DPAT_PGE_SCIENCE_MD** and **DPAT_ESDT_SCIENCE_MD** files.)

Select View

The initial screen is as depicted in Figure 4.5.1-29, with the *SELECT* tab option displayed. This pop-up provides the following options.

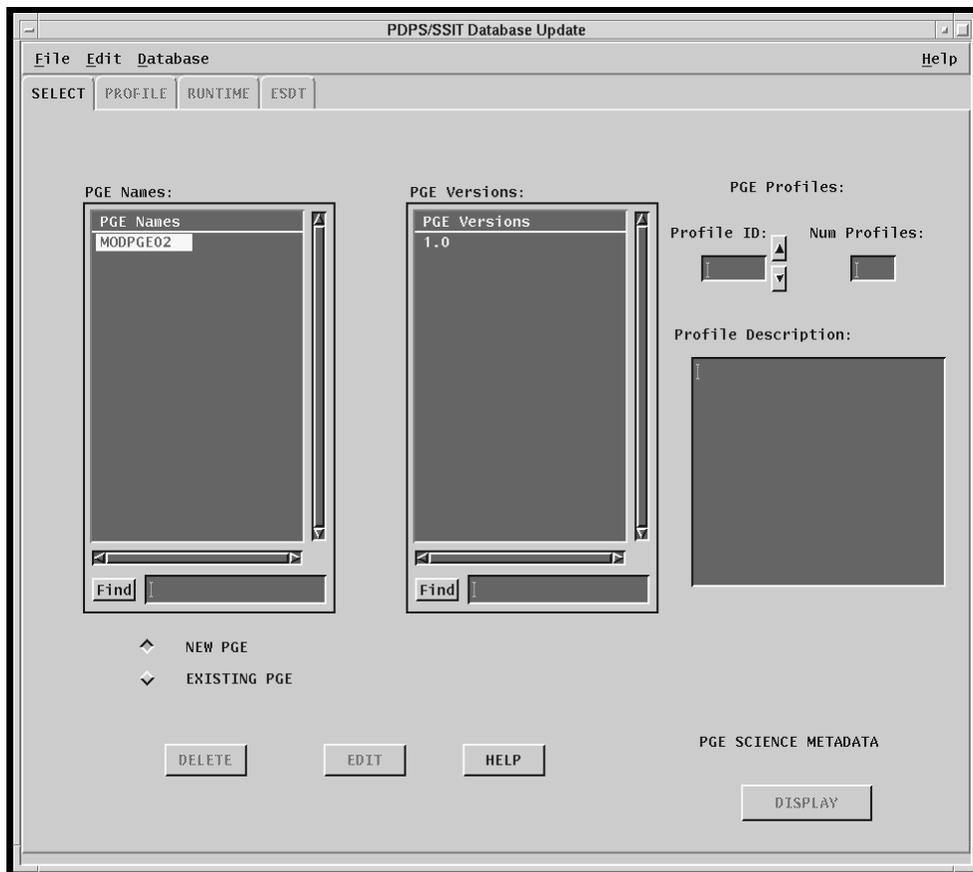


Figure 4.5.1-29. PDPS/SSI&T Database Operational Metadata Update Pop-up – Showing the SELECT Tab

- The user can select to list either new or existing PGEs by clicking either the **NEW PGE** button or the **EXISTING PGE** button. The difference is that new PGEs have not yet had their operational metadata placed into the database.
- Clicking on a PGE in the list in the **PGE Names:** field populates the version list in the **PGE Versions:** field with all the versions of the selected PGE presently defined in the PDPS database.
- Once the PGE has been selected, information on the PGE profile is also made available to the user through the **Profile Description:** field for each selected *Profile ID*.
- Clicking on the **Display** button displays PGE science metadata in a *Metadata Display* window as read-only.
- If required, selected PGEs can also be flagged as deleted in the database by clicking on the *Delete* button. (However, data is never physically deleted from the database by any SSI&T program. Only database administrators have permission to do this, using ISQL.)
- Clicking on the EDIT button allows for changing the selected PGE.

Profile View

Selecting the *PROFILE* tab displays the profile information of the selected PGE. This view is shown in Figure 4.5.1-30. The view contains fields showing Resource Requirements and Performance Statistics of the selected PGE. Descriptions of each field are presented in Table 4.5.1-10. Default values for these fields (i.e. for new PGEs) are set to null. All of the values must be filled in before the database update is completed. For existing PGEs, current settings in the database are displayed. The user can update them as needed. *Cut* and *Paste* capabilities are provided in the *Edit* pull down menu to help perform the functionality associated with this tool. Clicking on *Reset* sets the fields to the current values in the database ignoring all the changes made by the user during the current session. To record updated or new values in the database the user must click on the *APPLY* button.

Once the data on this screen has been entered into the database successfully, the PGE changes state from “New” to “Existing”. (The PGE name and version scrolling lists on the SELECT tab are now visible after the user selects *Existing* there.)

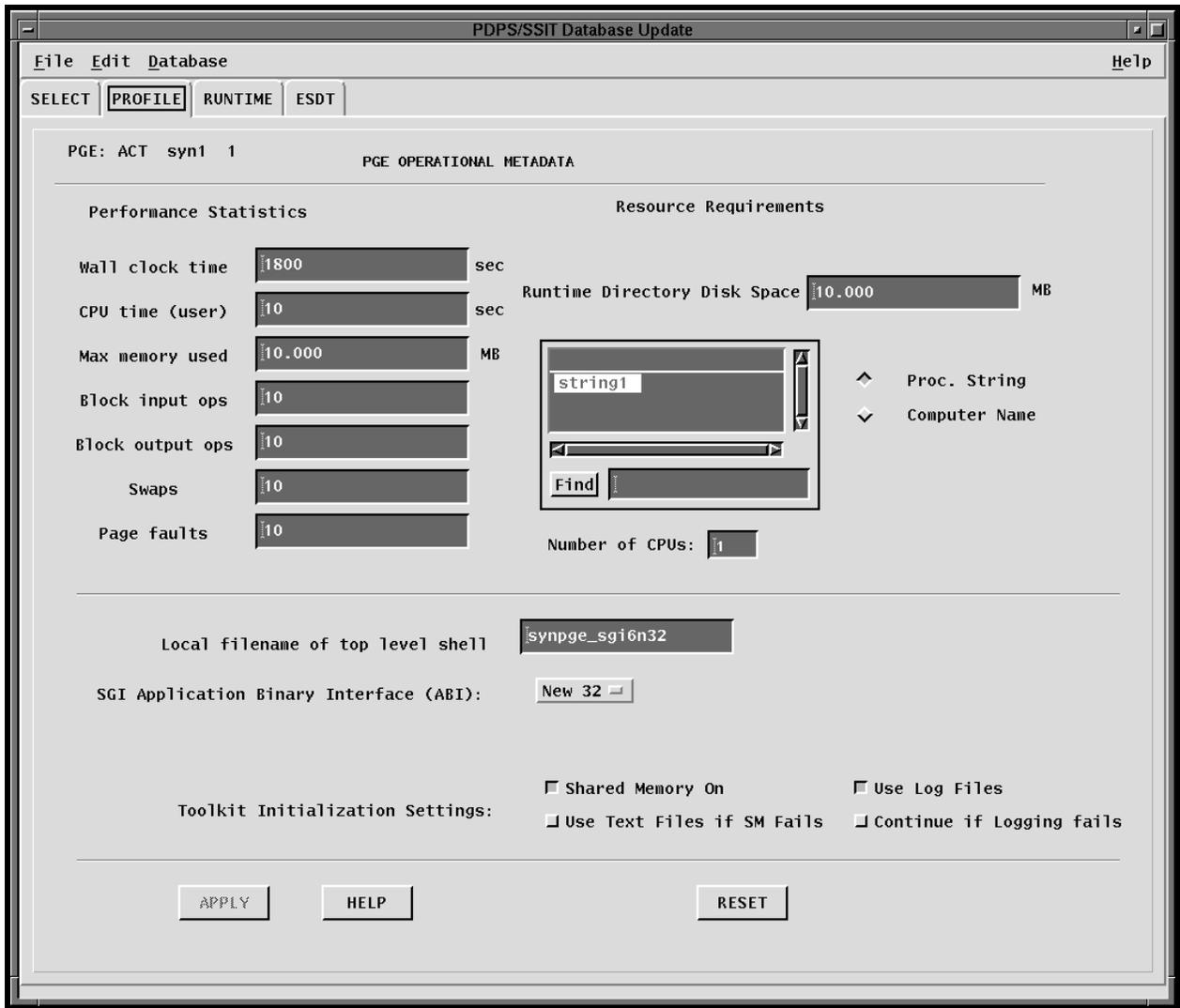


Figure 4.5.1-30. PDPS/SSI&T Database Operational Metadata Update Pop-up - Showing the PROFILE Tab

Table 4.5.1-12 describes the fields in the PDPS/SSI&T Database Operational Metadata Update pop-up.

**Table 4.5.1-12. PDPS/SSI&T Database Operational Update Profile
Field Descriptions (1 of 2)**

Field Name	Data Type	Size	Entry	Description
Wall clock Time	Float	N/A	User	Real time used by the process (in seconds); filled in both after processing with SCF Toolkit and with DAAC Toolkit (i.e., with AutoSys).
CPU Time	Float	N/A	User	CPU time used by the process in seconds.
Max Memory Used	Float	N/A	User	Maximum memory used by the process in megabytes.
Block Input Ops	Integer	N/A	User	Number of times the file system performed input in servicing.
Block Output Ops	Integer	N/A	User	Number of times the file system performed output in servicing.
Swaps	Integer	N/A	User	The number of times the process was swapped out of main memory.
Page Faults	Integer	N/A	User	The number of page faults serviced by the process.
Runtime Directory Disk Space	Float	N/A	User	Maximum disk space used by the process.
Processing String/ Computer Name display Window	Selectable	N/A	System generated	Processing string or computer name on which the PGE is executed.
Processing String	ASCII characters	30	User	A name given to collection of processors where PGE can be executed.
Computer Name	ASCII characters	30	User	Name of the computer where the process is executed.
Number of CPUs	Integer	N/A	User	Number of processors required by the PGE.
Local filename of top level shell	ASCII	30	User	Name of the executable file for the PGE.
SGI Application Binary Interface (ABI)	Selectable	N/A	System Defined	The "mode" of SGI compilation of the executable.
Toolkit Initialization Settings: Shared Memory On	On/Off	N/A	User	Determines if Toolkit shared memory is activated during PGE execution.

**Table 4.5.1-12. PDPS/SSI&T Database Operational Update Profile
Field Descriptions (2 of 2)**

Field Name	Data Type	Size (of characters)	Entry	Description
Toolkit Initialization Settings: Use Text Files if SM fails	On/Off	N/A	User	Determines if the Toolkit should use text files for shared memory if there is a failure using shared memory.
Toolkit Initialization Settings: Use Log Files	On/Off	N/A	User	Determines if the Toolkit logs information during PGE execution.
Toolkit Initialization Settings: Continue If Logging fails	On/Off	N/A	User	Determines if PGE execution should continue if logging fails.

Runtime View

Selecting the *RUNTIME* tab displays the PGE runtime parameters, i.e., static parameters used by the PGE at runtime. This view is shown in Figure 4.5.1-31. The logical ID, name and value that have already been entered in the database by the Science Metadata Update program are displayed on the left side of the screen. After selecting one of these runtime parameters the user is given the opportunity to add some descriptive text. (Use of this functionality is optional. Runtime parameter descriptions can remain as null values.) The user clicks the *OK* button when finished editing a particular parameter. Modifications are applied to the database by clicking on the *APPLY* button. As before, clicking on *RESET* sets descriptions of all parameters to current values in the database. Details of the runtime parameter fields are shown in Table 4.5.1-13.

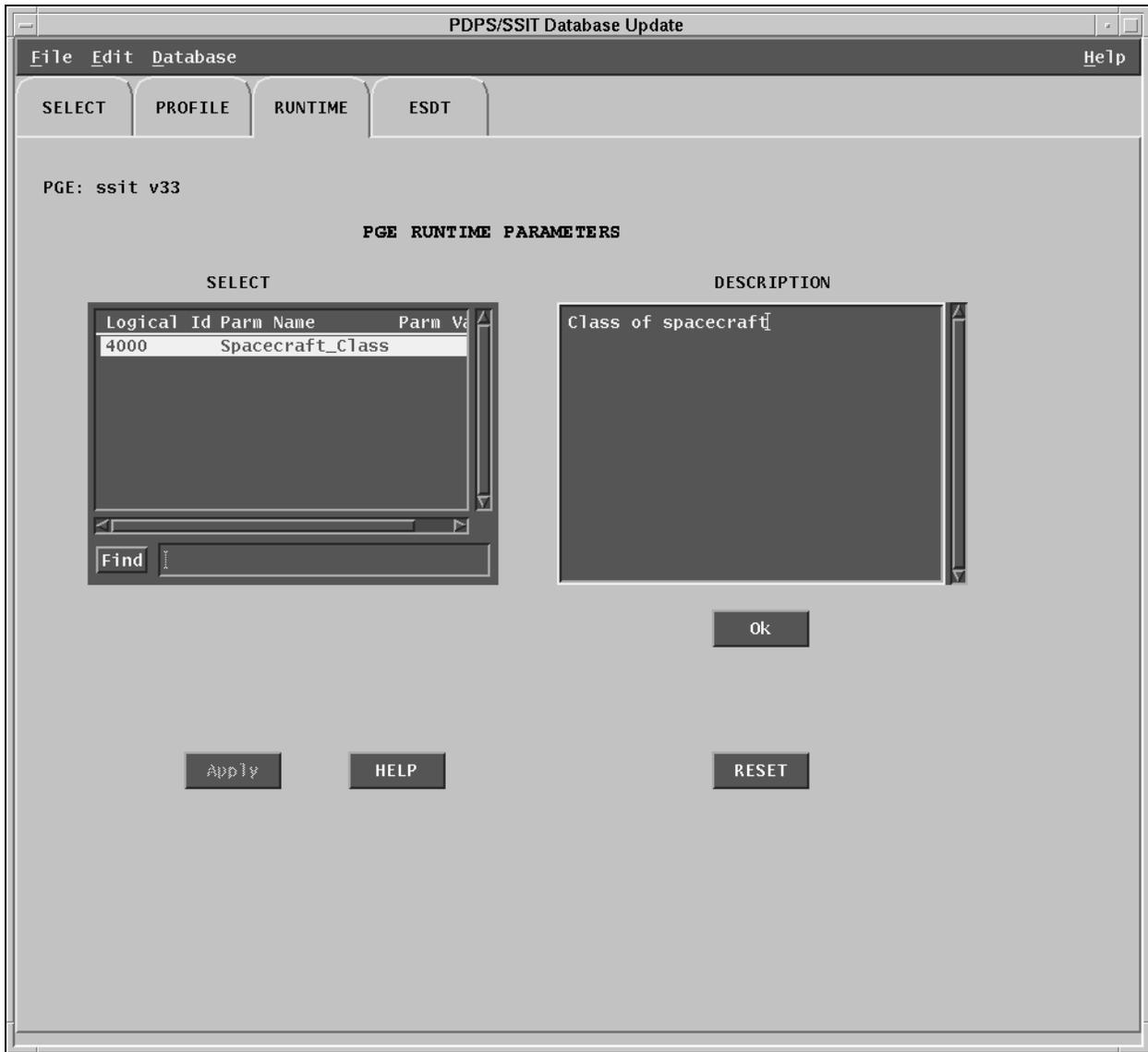


Figure 4.5.1-31. PDPS/SSI&T Database Operational Metadata Update Pop-up Showing the Runtime Tab

Table 4.5.1-13 describes the fields on the RUNTIME Tab.

**Table 4.5.1-13. PDPS/SSI&T Database Operational Metadata Update
Runtime View Field Description**

Field Name	Data Type	Size (of characters)	Entry	Description
Logical ID	Integer	N/A	Read from database	Logical ID of the user parameter in the PCF file
Parameter Name	ASCII characters	N/A	Read from database	User Parameter Name
Parameter Value	ASCII characters	N/A	Read from database	User Parameter Value
User Parameter text description	ASCII characters	<60	User	Text describing the user parameter

ESDT View

Selecting the *ESDT* tab displays the ESDT information of the selected PGE. This view is shown in Figure 4.5.1-32. Two ESDTs lists are made available to show both the PGE Input Data types and the PGE Output Data types. On selection of a particular ESDT, the associated PDPS Science ESDT metadata can be viewed by clicking on the **DISPLAY** button. The **HELP** button will explain the use of the window. This data is read only and cannot be modified through this Pop-up.

The program looks for the file it set in the configuration file parameter **DPAT_ESDT_SCIENCE_MD**.

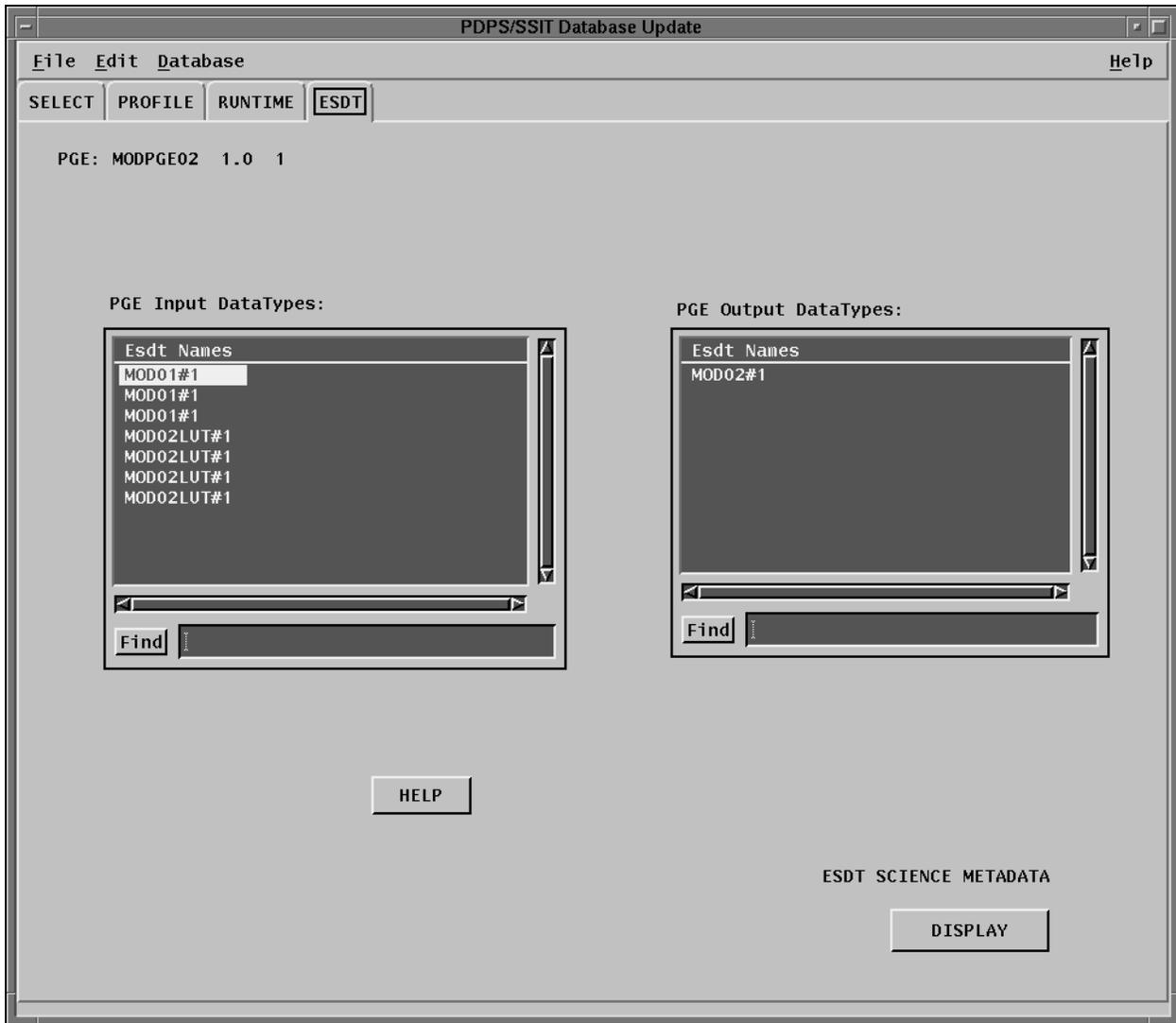


Figure 4.5.1-32. PDPS/SSI&T Database Operational Metadata Update Pop-up - Showing the ESDT Tab

Display Screen

Figure 4.5.1-33 shows a pop-up Display screen. The Display window is used to display either a PGE Science Metadata (selected from the *SELECT* tab) or the ESDT Science Metadata (selected from the *ESDT* tab).

A Display screen pops-up also if the request for modification to the database fails because of some mandatory field not being properly set (see screen of *PROFILE* view). Editing is not allowed in this window. Clicking on "DONE" closes this window.

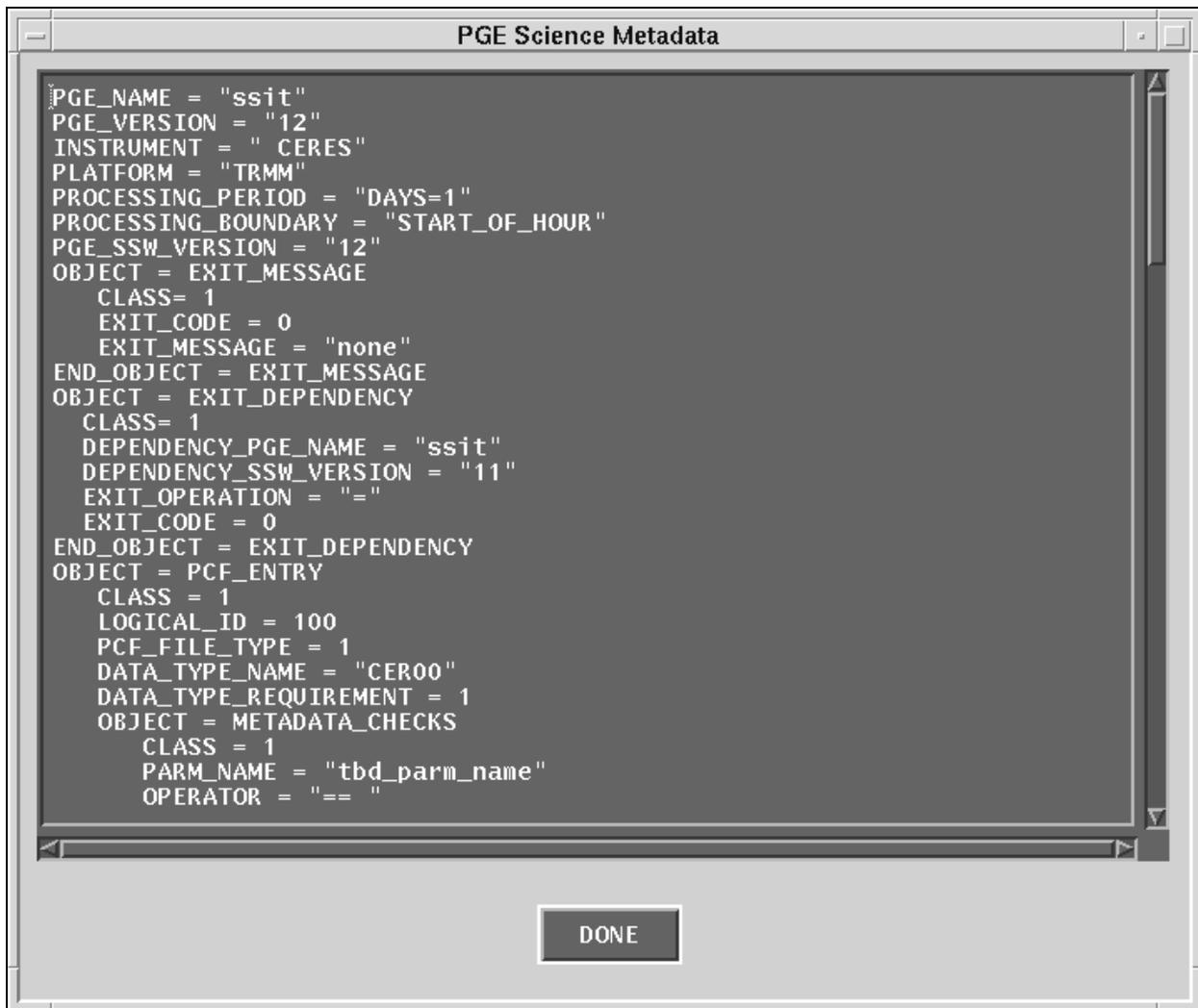


Figure 4.5.1-33. PDPS/SSI&T Database Operational Metadata Update Pop-up - Display Screen

4.5.1.2.9 Data Server

This section describes the interfaces to the Data Server used by the SSI&T Specialist during the SSI&T process. All programs can be started from the **Tools->Data Server** menu of the SSI&T Manager shown in Figure 4.5.1-34.

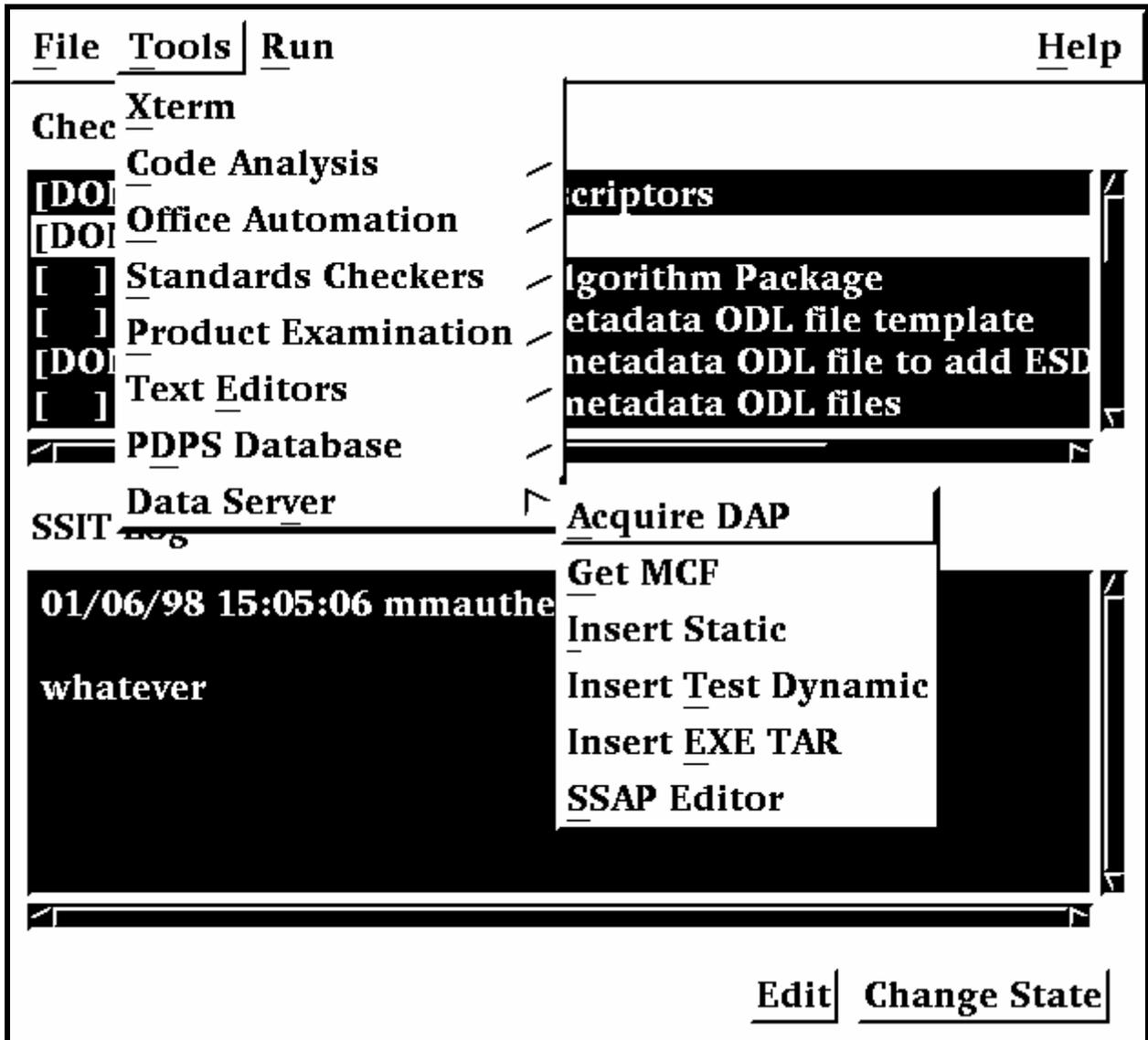


Figure 4.5.1-34. SSI&T Manager, Tools Menu, Data Server Submenu Choices

4.5.1.2.9.1 Acquire DAP

This program is used to retrieve the DAP from the Data Server.

Before its use, the following events must have occurred:

1. A subscription for the DAP must have been registered with the Data Server. The subscription delivery option is set to email, to a specified DAAC operator or mail drop.
2. The DAP must have been Ingested. There are two ways for this to occur. First, the DAP can be processed by Ingest. When this occurs, Ingest inserts the DAP into the Data Server, triggering subscription notification. Second, the DAP can be inserted with the Insert Test File.
3. After the DAAC operator received the email subscription notification, he or she must save it to a file.

To run the program, select *Acquire DAP* from the Data Server submenu. The program prompts for input parameters Process Framework configuration filename, email message filename and directory to receive staged file.

The program reads the DAP Universal Reference (UR) from the email file, acquires the DAP from the Data Server, and stages the DAP on the local disk. The name of the staged file is written to standard output. Note, this function can be used to acquire any type of granule with a known UR.

After the file is staged, the operator can unpack it and install its components in ClearCase or in other places as appropriate.

NOTE: Use of this program is optional, in the sense that the DAP can arrive at the DAAC through other means than Ingest, e.g., simple ftp.

4.5.1.2.9.2 Get MCF

This Get Metadata Configuration File (MCF) tool has a Character-based User Interface (CHUI). It is invoked by selecting **Tools->Data Server->Get MCF** menu option of the SSI&T Manager. The user is prompted to provide pertinent information about the ESDT for the MCF to be acquired from the Data Server. The information includes the configuration file name, the ECS mode of operations, the ESDT short name, the ESDT version, and, finally the location where the MCF needs to be transferred to. When the above information is provided the tool contacts the Data Server and, if the operation is successful, copies the MCF to the new location. A message to the operator either confirms the successful transfer of the MCF or issues a warning, which explains why the operation did not complete successfully.

4.5.1.2.9.3 Insert Static

The Insert Static File program inserts static input file(s) to the Data Server, for use both during SSI&T and in production. Static Files are files rarely changed between instances of PGE processing; for example, calibration files. Such files are normally part of the DAP.

After this program runs successfully, the PDPS database has been updated with the Data Server Universal Reference (UR) of the file inserted. The Planning and Processing systems read this UR at run-time so the file can be acquired and used during production.

The Insert Test Dynamic File program is command line oriented. There is no CHUI or Pop-up for this program.

Before this program is used, the following must have occurred:

- 1) An ESDT for this data type must have been created in the Data Server.
- 2) A PGE that uses the Static File must have been registered to PDPS via the SSI&T Science Metadata Update tool.
- 3) An ASCII Metadata File (.met) for this ESDT must have been created, in the format expected by the Data Server for Insert.
- 4) Both the static data file(s) in question and its corresponding ASCII Metadata file must be accessible to the local machine.
- 5) The directory where the static data file(s) and the ASCII Metadata File exist must be cross-mounted to the Data Server machines.

The Static Input File program produces prompts for the operator to enter the necessary information. These prompts and explanations follow. Note: if there is a default value for an entry (in most cases this only occurs if you run the program more than once) it appears at the end of the prompt line.

Configuration filename? (enter for default: ../../cfg/EcDpAtInsertStaticFile.CFG)

In most cases hitting enter (for the default) is fine. If not, enter the correct configuration filename including full path.

ECS Mode of operations?

This is the mode (i.e., OPS, TS1) of operations. In most cases this is TS1

ESDT short name for the file(s) to Insert?

This is the ESDT short name (max 8 characters) for the data file(s).

ESDT Version for the file(s) to insert?

This is the ESDT version (an integer) for the data file(s).

Science Group for Static file (one of {C, L, D, O} followed by a 3-digit number)?

This is the Science Group for this Static File(s). It is defined in the PGE Metadata ODL File for each PGE using this static file.

C = Coefficient File(s), L = Lookup Table/file(s), D = Database File(s), O = Other Files.

Note: using the wrong letter for the file(s) (say L for a Coefficient File) does not result in an error.

Is there more than one data file to this Static (Y = Yes, N = No)?

This indicates to the program if there is more than one file associated with the static input. Most static inputs are a single data file and a corresponding ASCII Metadata File. Other static inputs consist of more than one data file and a single ASCII Metadata file (there is always only 1 ASCII Metadata File).

If there is only one data file enter N (or just hit enter for the default). For multiple data files enter Y.

If it is NOT a Static Multi-File Granule (there is a single data file) the following prompts need to be answered:

Single Filename to Insert (including FULL path)?

This is the name of the data file to insert to the Data Server. Include the full path to the file so the tool can find it.

Associated ASCII Metadata Filename to Insert (including FULL path)?

This is the name of the ASCII Metadata File (.met) associated with the data file. The full path to the file must be included so the tool can find it. Using the Get MCF tool to get a Metadata Configuration File, which specifies the fields and type of data required for the ASCII Metadata file, can create it.

If it IS a Static Multi-File Granule (more than 1 data file) the following prompts need to be answered:

Directory where all data files and .met file exist (FULL path)?

This is the directory location where all the data files and the ASCII Metadata file is located. For Multi-File Statics all files and Metadata must reside in the same directory.

Name of MFG file (enter to end list)?

This is the name of one of the data files for a Static Multi-File Granule. Enter a file at the prompt and the prompt is then being repeated, allowing for the next file to be named. When all data files have been entered, just hit the enter key at the next prompt (thus entering nothing).

Associated ASCII Metadata Filename to Insert?

This is the name of the ASCII Metadata File (.met) associated with the data file. No path is needed because it is assumed the file resides in the directory specified above. Using the Get MCF tool to get a Metadata Configuration File, which specifies the fields and type of data required for the ASCII Metadata file, can create it.

Any success or error messages are then displayed followed by:

Hit return to run again, 'q <return>' to quit:

This allows the user to enter another Static File (just hit enter) or quit the program (q).

4.5.1.2.9.4 Insert Test Dynamic

The Insert Test Dynamic File program inserts test dynamic input files to the Data Server, for use both during SSI&T and in production. Dynamic Files are files that change at each instance of PGE processing; for example, Level 0 data files. This tool performs a function normally done by Ingest. It allows the user to insert dynamic files to the Data Server for testing purposes. Normally, Ingest takes in such files and inserts them as part of its normal processing.

After this program runs successfully, the specified file(s) have been stored at the Data Server and can be acquired by PDPS when executing PGEs. Also, any subscriptions made on the ESDT of the insert file(s) trigger, and could (via Subscription Manager) cause PGE waiting on this data to initiate execution

The Insert Test Dynamic File program is command line oriented. There is no CHUI or Pop-up for this program.

Before this program is used, the following must have occurred:

1. An ESDT for this data type must have been created in the Data Server.
2. A PGE that uses the Dynamic File(s) ESDT must have been registered to PDPS via the SSI&T Science Metadata Update tool.
3. An ASCII Metadata File (.met) for this instance of the ESDT must have been created in the format expected by the Data Server for Insert.
4. Both the test dynamic data file(s) in question and its corresponding ASCII Metadata file must be accessible to the local machine.
5. The directory where the dynamic data file(s) and the ASCII Metadata File exist must be cross-mounted to the Data Server machines.

The Insert Test Dynamic program produces prompts for the operator to enter the necessary information. These prompts and explanations follow. Note: if there is a default value for an entry (in most cases this only occurs if you run the program more than once) it appears at the end of the prompt line.

Configuration filename? (enter for default: ../../cfg/EcDpAtInsertTestFile.CFG)

In most cases hitting enter (for the default) is fine. If not, enter the correct configuration filename including full path

ECS Mode of operations?

This is the mode (i.e. OPS, TS1) of operations. In most cases this is TS1

ESDT short name for the file(s) to Insert?

This is the ESDT short name (max 8 characters) for the data file(s).

ESDT Version for the file(s) to insert?

This is the ESDT version (an integer) for the data file(s).

Is there is more than one data file to this Dynamic Granule (Y = Yes, N = No)?

This indicates to the program if there is more than one file associated with the dynamic input. Most dynamic inputs are a single data file and a corresponding ASCII Metadata File. Other dynamic inputs consist of more than one data file and a single ASCII Metadata file (there is always only 1 ASCII Metadata File).

If there is only one data file enter N (or just hit enter for the default).

For multiple data files enter Y. If it is NOT a Dynamic Multi-File Granule (there is a single data file) the following prompts need to be answered:

Single Filename to Insert (including FULL path)?

This is the name of the data file to insert to the Data Server. Include the full path to the file so the tool can find it.

Associated ASCII Metadata Filename to Insert (including FULL path)?

This is the name of the ASCII Metadata File (.met) associated with the data file. Full path to the file must be included so the tool can find it. Using the Get MCF tool to get a Metadata Configuration File, which specifies the fields and type of data required for the ASCII Metadata file, can create it.

If it is a Dynamic Multi-File Granule (more than 1 data file) the following prompts need to be answered:

Directory where all data files and .met file exist (FULL path)?

This is the directory location where all the data files and the ASCII Metadata file is located. For Multi-File Dynamics all files and Metadata must reside in the same directory.

Name of MFG file (enter to end list)?

This is the name of one of the data files for a Dynamic Multi-File Granule. Enter a file at the prompt and the prompt is then being repeated, allowing for the next file to be named. When all data files have been entered, just hit the enter key at the next prompt (thus entering nothing).

Associated ASCII Metadata Filename to Insert?

This is the name of the ASCII Metadata File (.met) associated with the data file. No path is needed because it is assumed the file resides in the directory specified above. Using the Get MCF tool to get a Metadata Configuration File, which specifies the fields and type of data required for the ASCII Metadata file, can create it.

Any success or error messages are then displayed followed by:

Hit return to run again, 'q <return>' to quit:

This allows the user to enter another Dynamic File (just hit enter) or quit the program (q).

4.5.1.2.9.5 Insert EXE TAR

The Insert Exe Tar File program is used to insert the tar file containing the PGE executable (and associated files) to the Data Server for use during SSI&T and in production.

The Insert EXE TAR program is command line driven. There is no CHUI or Pop-up for this program.

Preconditions to running the Insert Test Dynamic File program

Before this program is run, the following must have occurred:

- An ESDT for EXE TAR files must have been installed at the Data Server.
- The PGE must have been registered to PDPS via the SSI&T Science Metadata Update tool.
- An ASCII Metadata file (.met) for the PGEEEXE ESDT must have been created, in the format expected by the Data Server for Insert. A template for such a file is delivered with the software and stored under /usr/ecs/{MODE}/CUSTOM/data/DPS/PGEEEXE.met.template.
- The tar file must be prepared with the files needed to execute the PGE (see below for a list).
- Both the tar file in question and its corresponding MCF must reside on the local machine.
- The directory where the tar file and the ASCII Metadata File exist must be cross-mounted to the Data Server machines.

The Exe Tar file must include

- ⇒ PGE compiled binary executable(s),
- ⇒ PGE scripts (if any),
- ⇒ SCF-generated SDP Toolkit compiled runtime message files (if any),
- ⇒ Any dynamic link libraries needed by the PGE at runtime.

...in short, anything needed to run the PGE, aside from Data Server input and output files.

The program prompts for input parameters Process Framework configuration filename, database name to update, PGE name, science software version, full path filename to insert, MCF full path filename, and top-level shell filename within tar file. (“Top level shell filename“ is the name of the script file or executable which one uses to kick off the PGE.)

After the program is run, the PGE is ready to run either in the SSI&T environment or in the production system. Then the PDPS database has been updated with the Data Server Universal Reference (UR) of the Exe Tar file inserted, the size of this file, and the top-level shell within the tar file. This data is read by the Processing system at runtime and the PGE Exe Tar file is acquired from the Data Server.

NOTE: All Exe Tar files are inserted to a single ESDT, the Short Name of which is hard-coded into the EcDpAtInsertExeTarFile software as PGEEEXE.

The Insert EXE TAR program produces prompts for the operator to enter the necessary information. These prompts and explanations follow. Note: if there is a default value for an entry (in most cases this only occurs if you run the program more than once) it appears at the end of the prompt line.

Configuration filename? (enter for default: ../../cfg/EcDpAtInsertTestFile.CFG)

In most cases hitting enter (for the default) is fine. If not, enter the correct configuration filename including full path

ECS Mode of operations?

This is the mode (i.e. OPS, TS1) of operations. In most cases this is TS1.

Name of PGE?

This is the name of the PGE to insert (max length 10 characters). It must have already been defined to the PDPS database.

Science Software version of PGE?

This is the Science Software version (version of the actual executable) of the PGE (max 5 characters). In most cases this is the same as the PGE Version, but it is possible to change the PGE Version without changing the actual executable (in which case the Science Software version and PGE Version would be different). The Science Software version is specified in the PGE Metadata ODL File.

Staged filename to Insert (including FULL path)?

This is the name of the Exe tar file to insert to the Data Server. Include the full path to the file so the tool can find it.

Associated ASCII Metadata Filename to insert (including FULL path)?

This is the name of the ASCII Metadata File (.met) associated with the tar file. The full path to the file must be included so the tool can find it. Use the /usr/ecs/{MODE}/CUSTOM/data/DPS/PGEEEXE.met.template file to create one for the tar file to be inserted.

This is the name of the top-level executable within the tar file. It is the name of the program PDPS is supposed to execute when running the PGE.

4.5.1.2.9.6 Science Software Archive Packages (SSAP) Editor

The SSAP Editor Main Pop-up is a tool that manages Science Software Archive Packages. In particular, the SSAP Pop-up allows the operator to create a new SSAP, update or delete an existing one, add or remove components, and edit Metadata associated with an SSAP.

From the SSI&T Manager menu, the operator selects the **T**ools->**D**ata Server->**S**SSAP Editor option. The SSAP Editor as shown in Figure 4.5.1-35 appears.



Figure 4.5.1-35. SSAP Editor Main Pop-up

A command line is also available to start the SSAP Editor. The operator can type the command:

EcDpAtSSAPGui ConfigFile <CONFIG_FILE> ecs_mode <MODE>,

where <CONFIG_FILE> and <MODE> are, respectively, the name of the Process Framework configuration file for this specific application and the mode of operations. The tool does not allow the operator to come up in a mode that is not consistent with the one established at startup.

Batch Mode; allows the operator to run a script including command lines for

The main window contains three pull-down menus, each one performing the following functions:

- The **File** menu is used to exit from the SSAP Editor
- The **Options** menu has two items:
 - **Check Permissions**; displays a Message window showing a list of functions the operator can perform with the SSAP Editor tool, including; Create SSAPs, Delete SSAPs, Add Files to SSAPs, Delete Files from SSAPs, Edit SSAP Metadata creating SSAPs. The operator is prompted to select the batch mode file using the **Input File Selection** screen, shown in Figure 4.5.1-36.

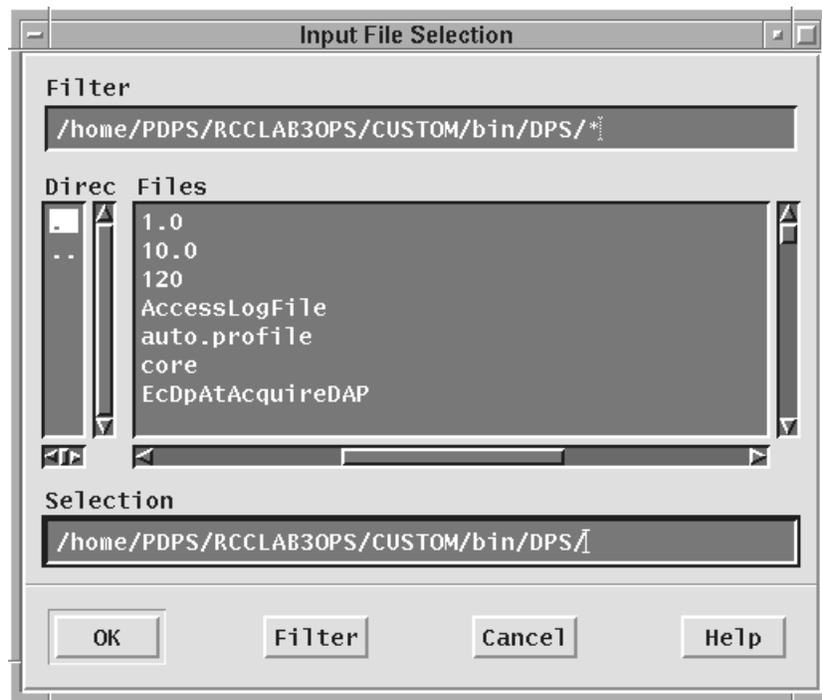


Figure 4.5.1-36. Input File Selection Pop-up

- The **Help** menu provides context sensitive help information

A toolbar at the top of the screen shows the name of the **Current SSAP** and the **Current File Type** (SSAP), shown in Figure 4.5.1-35.

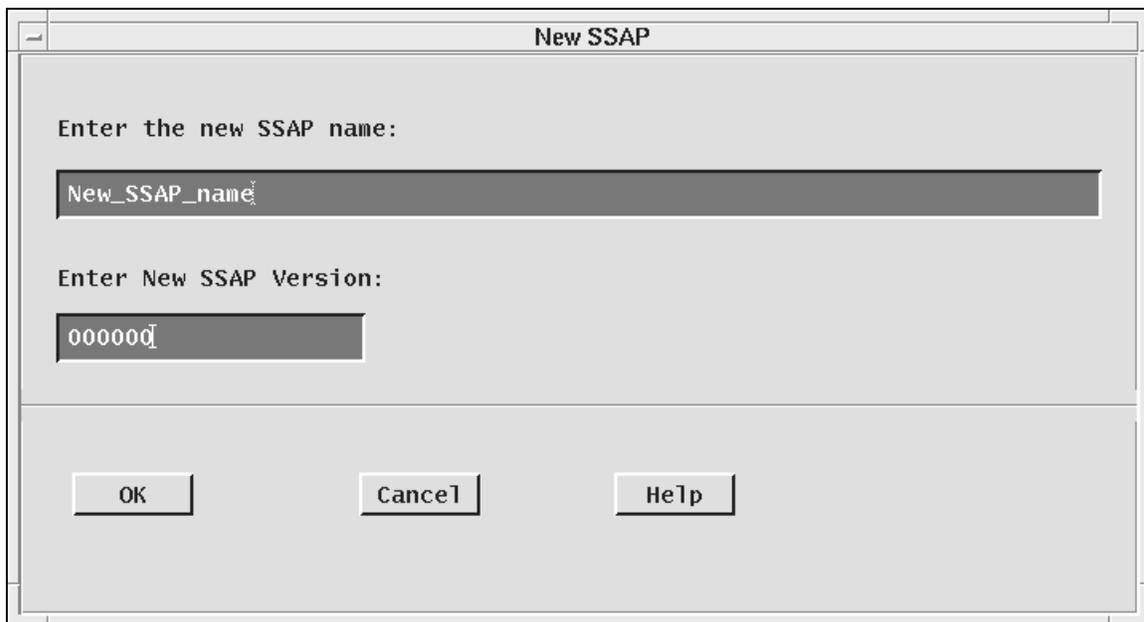
The SSAP Editor Main Pop-up screen has three tabs, i.e., **Main**, **File List** and **Metadata**.

The following three sub-sections provide details on the views opened by clicking on each of these tabs.

4.5.1.2.9.6.1 SSAP Editor Main View

The **Main** View (Figure 4.5.1-35) is the default view appearing when the SSAP Editor is started. It contains the following controls:

- A free text window listing all the existing SSAPs retrieved from the Data Server
- A **Refresh** button that updates the above SSAP Listing, regenerating the screen every time the operator needs to do so, to add recent changes to the listing
- A **Create** button allowing the operator to start the creation of a new SSAP. Upon clicking this button, the New SSAP screen appears where the operator is prompted to provide a new SSAP Name and SSAP Version, shown in Figure 4.5.1-37.



The image shows a dialog box titled "New SSAP". It has a standard window border with a title bar. Inside the dialog, there are two text input fields. The first is labeled "Enter the new SSAP name:" and contains the text "New_SSAP_name". The second is labeled "Enter New SSAP Version:" and contains the text "00000". At the bottom of the dialog, there are three buttons: "OK", "Cancel", and "Help".

Figure 4.5.1-37. New SSAP Window

- A **Create With** button is provided as an alternative for the user to create a new SSAP starting with components of an existing SSAP file. To do this, the operator should click on an existing file, and change the parameters of the old SSAP with the name, SSAP version, and other attributes of the new SSAP. The operator completes the creation of the new SSAP with

saving the changes to the new SSAP. Both the old and new SSAPs retain their different identities

- A **Submit** button allows the operator to submit the new SSAP to the Data Server for storage or update an existing SSAP
- The operator for deleting a selected SSAP from the Data Server uses a **Delete** button
- A **Reset** button allows the operator to undo the changes made for the currently selected SSAP during the session.

4.5.1.2.9.6.2 File List View

Once the new SSAP has been named or an existing one has been selected, the operator can update the list of SSAP components by clicking on the **File List** tab. Selecting the **File List** tab, the operator is presented with the screen shown in Figure 4.5.1-38.

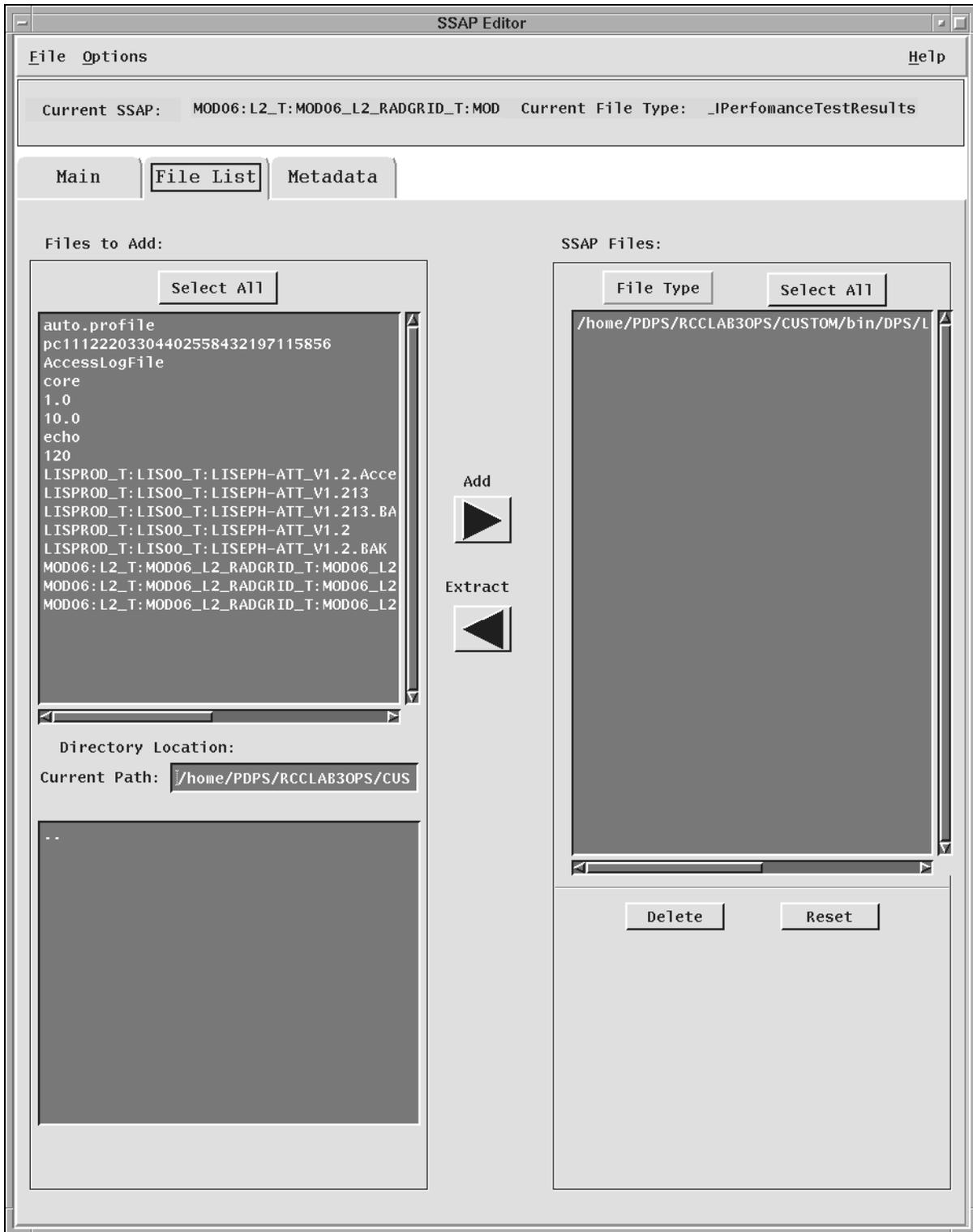


Figure 4.5.1-38. File List View Pop-up

On this screen, two main windows are shown. The first window on the left side of the screen shows all the files available on the local machine and the bottom window on this side of the screen shows the directory location, showing the current path. The other window on the right side of the screen shows the files included in the SSAP. The two buttons **Add** and **Extract** allow for the transfer between the two aforementioned sets of files between the top two windows on the screen. Once a file is selected in the list of files of the local machine, it is listed as an SSAP file component and included in a tar file.

- By clicking on the **File Type** button, selected SSAP files can be further identified as one of the following: Algorithm Description, Change Log, Context Dialogue, Delivery List, PGE Information and Software. A **File Type** must be selected before file list changes can be made

Other Options are offered to the operator to act on the list of SSAP files:

- The **Select All** button selects all the files in the **SSAP Files** window
- The **Delete** button deletes the selected files; a safety window pops-up to ask the operator to confirm the delete request
- The **Reset** button allows the operator to undo the changes just made

4.5.1.2.9.6.3 Metadata View

For each existing SSAP, the operator clicks on the Metadata tab. Pertinent information about the algorithm and the PGE can be input by the operator in several text fields, shown in Figure 4.5.1-39.

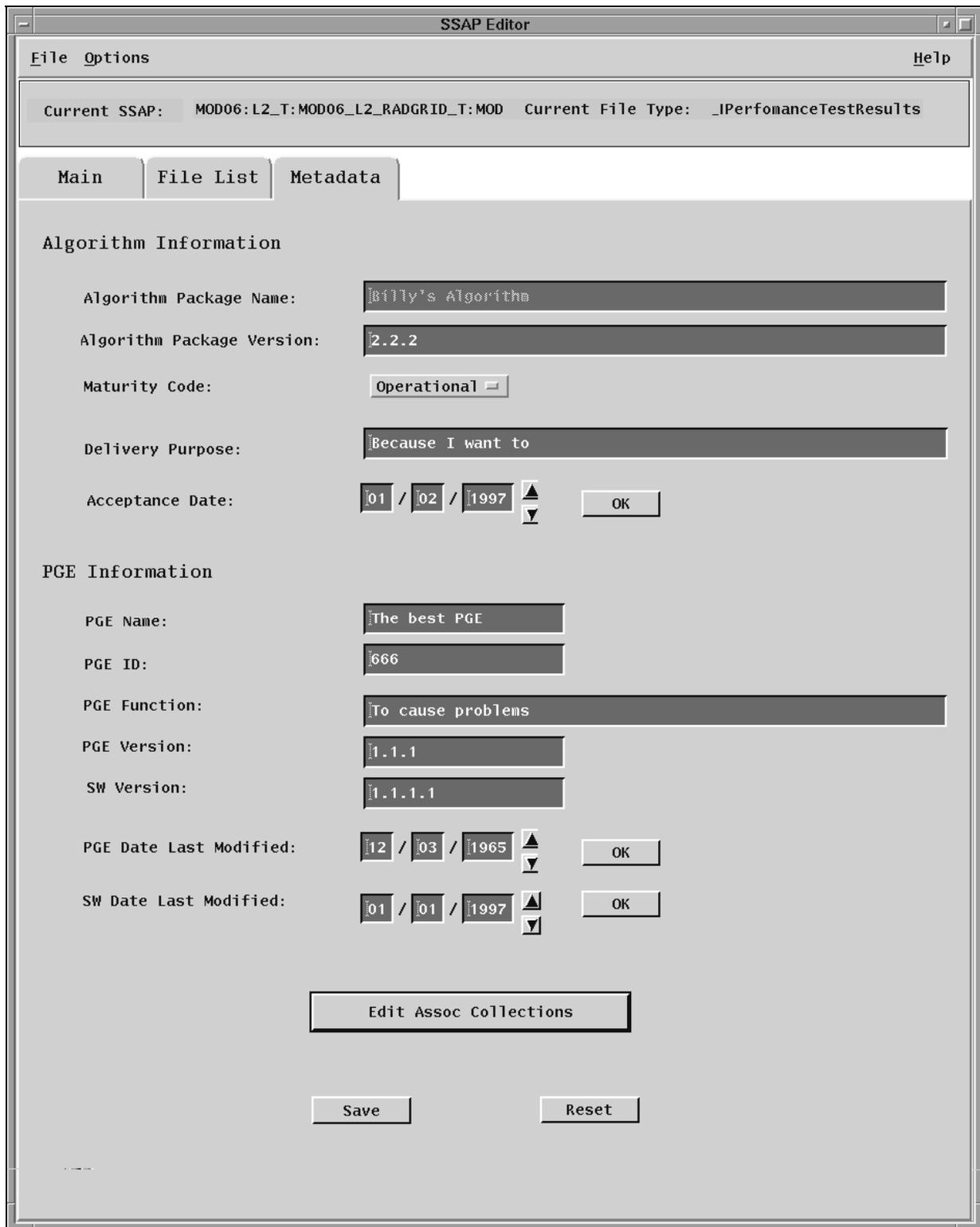


Figure 4.5.1-39. Metadata View Pop-up

Some of this information is already made available by default but can be changed by the user. Descriptions of each of the fields the operator is required to fill in are provided in the following Table 4.5.1-14.

Table 4.5.1-14. SSAP Editor - Metadata Tab Field Descriptions

Field Name	Data Type	Size	Entry	Description
Algorithm Package Name	String	Unlimited	User	Name of the Algorithm included in the SSAP.
Algorithm Package Version	String	Unlimited	User	Version of the Algorithm included in the SSAP.
Delivery Purpose	String	Unlimited	User	Reason for creating an SSAP and submitting it to the Data Server for acceptance.
Acceptance Date	Date	MM:DD:YYYY	User	The date the SSAP was accepted.
PGE Name	String	Unlimited	User	Name of the original PGE used to generate the SSAP.
PGE ID	String	Unlimited	User	ID of the original PGE used to generate the SSAP.
PGE Function	String	Unlimited	User	Metadata information on the original PGE.
PGE Version	String	Unlimited	User	Version of the original PGE used to generate the SSAP.
SW Version	String	Unlimited	User	Version of the code included in the PGE.
PGE Date Last Modified	Date	MM:DD:YYYY	User	Date the PGE was last modified.
SW Date Last Modified	Date	MM:DD:YYYY	User	Date the SW was last modified.

The Metadata view screen allows the operator to select the level of maturity or version of the code included in the SSAP. By clicking on the **Maturity Code** pull down menu, one of the following options can be selected: Pre-launch, Preliminary, Operational, Stable and Final.

The input of all the date fields is facilitated by the presence of ad-hoc up and down arrows that increment and decrement the values available as default or previously input by the operator.

In Figure 4.5.1-39, the **Edit Assoc Collection** button allows the operator to edit the ESDT's the original PGE is associated with. By clicking on this button, the Associated Collections pop-up appears as shown in Figure 4.5.1-40. Clicking the **Save** button will save changes to the SSAP Editor. Clicking the **Reset** button will allow the operator to undo the changes just made.

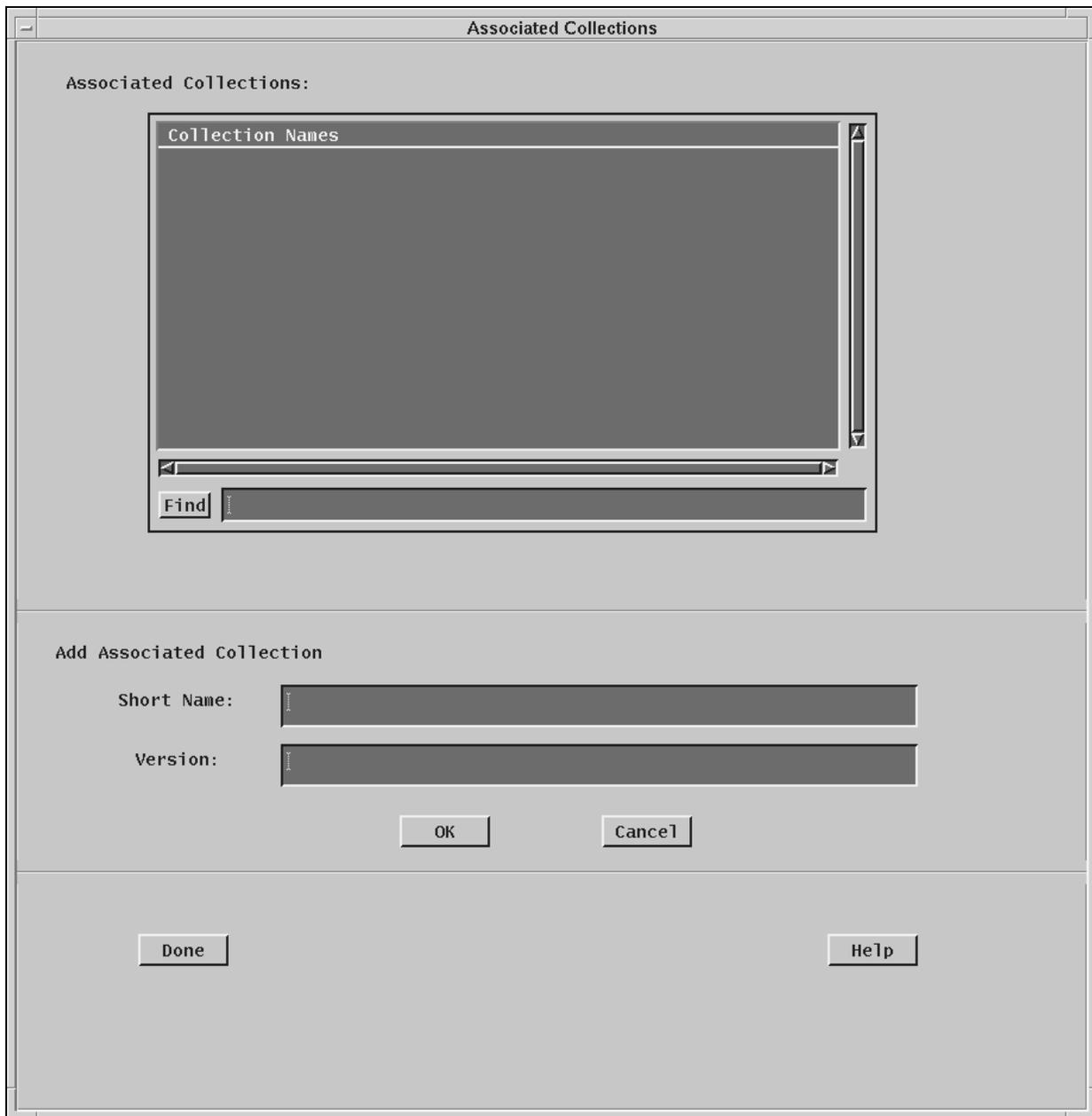


Figure 4.5.1-40. Associated Collections Pop-up

It should be noted that at least one ESDT must be associated with the original PGE in order to create a new SSAP. The Associated Collections window allows the operator to select the collection name and to input the **Short Name** and **Version** of the Associated Collections to be added. The text fields the operator is requested to provide in this pop-up are described in Table 4.5.1-15.

Table 4.5.1-15. SSAP Editor - Association Collection Field Description

Field Name	Data Type	Size	Entry	Description
Short Name	String	Unlimited	User	Name of the Associated Collection included in the SSAP.
Version	String	Unlimited	User	Version of the Associated Collection included in the SSAP.

- Once the selection and/or the addition have been completed, the operator can confirm the changes by clicking on the **Ok** button. By clicking the **Cancel** button, the Association Collections window will terminate. Further context sensitive help is available to the operator by clicking on the **Help** button
- Once **Done** is clicked, the operator is returned to the Metadata screen, shown in Figure 4.5.1-39. The **File** button at the upper left hand corner of the screen is clicked, to return to the SSI&T Manager menu, and finally to quit the operation

4.5.1.2.10 Help Menu

This appears at the far right of the SSI&T Manager Pop-up. Selecting the INDEX menu item brings up an index of all SSI&T Tools, as shown in Figure 4.5.1-41. Select a tool for which help is desired to bring up a help screen.

Selecting the ABOUT menu item brings up an instance of Netscape with a HTML file that provides basic information on the SSI&T Manager Tool, the Version number, etc.

4.5.1.3.1 Interfaces and Data Types

The SSI&T Manager exchanges data of various types through interfaces within and external to ECS. Table 4.5.1-16 lists SSI&T Manager system interfaces for Release 6.

Table 4.5.1-16. SSI&T Manager Interface Protocols

Interface (facility)	Type of Primary Interface Protocols	Type of Backup Interface Protocols	Comments
Acquire DAP	Science Data Server	None	Fixed ESDT
Insert Static	Science Data Server	None	Variable ESDT
Insert Test Dynamic	Science Data Server	None	Variable ESDT
Insert EXE TAR	Science Data Server	None	Fixed ESDT
SSAP Edit	Science Data Server	None	Fixed ESDT
Get MCF	Science Data Server	None	Fixed ESDT

4.5.1.4 Databases/ Files

The SSI&T Manager uses the Unix ndbm flat-file database facility to generate the Log/Checklist database.

More importantly, the SSI&T process is closely coupled with the PDPS database. SSI&T is carried out in a different mode than production. SSI&T activities do not interfere with the production PDPS database. The PDPS database schema information is described in Section 4.7 Resource Planning of this document.

There are many files involved with the SSI&T Manager and the associated applications. These files are summarily described in Table 4.5.1-17.

Table 4.5.1-17. SSI&T Files (1 of 2)

File Title	File Name	Purpose	Associated SSI&T Application/Tool
Installation script	\$ECS_HOME/CUSTOM/DpAtINSTALL.sh EcCoAssist	Install SSI&T software	
SSI&T Internal Process Control File	\$PGS_PC_INFO_FILE (a parameter in the configuration file)	SSI&T Manager uses a Process Control File to determine names and locations of files. This Process Control File defines correspondence between specifications for actual files, and the logical identifiers used by the SSI&T Manager to access those files. This SSI&T PCF should not be confused with the PCF used by and included with the science software.	SSI&T Manager
Process Framework Configuration	\$ECS_HOME/CUSTOM/cfg/DpAt*.CFG	Enable Process Framework for PDPS database access, ECS event logging.	All SSI&T programs
SSI&T Manager Run Menu	\$ECS_HOME/CUSTOM/data/DPS/ssit_run_menu.txt	Add items to SSI&T Manager RUN menu.	SSI&T Manager
SSI&T Manager help files	\$ECS_HOME/CUSTOM/docs/WW/DPS/DpAtMgr*.html	Online help	All tools in SSI&T Manager
Sample checklist	\$ECS_HOME/CUSTOM/data/DPS/checklist.sample	SSI&T Manager Checklist	SSI&T Manager
ECS approved FORTRAN 77 extensions	\$FCKCNF (a parameter in the configuration file)	FORCHECK is configured to check for non-ANSI FORTRAN 77 extensions approved for use in ECS science code, according to "Data Production and SCF Standards and Guidelines".	DpAtMgrForcheck.sh
Prohibited function lists	\$ECS_HOME/CUSTOM/data/DPS/prohibitedFunctions*.txt	Suggested list of prohibited functions in science software, one for each language.	Prohibited Function Checker

Table 4.5.1-17. SSI&T Files (2 of 2)

File Title	File Name	Purpose	Associated SSI&T Application/Tool
Xresources files	In \$ECS_HOME/CUSTOM/data/DPS : • XDpAtMgr • Emacs • Xdiff • Xbadfunc • DpAtMgrCheckHdfFile.defaults	Customize appearance of Pop-ups.	Pop-ups
Binary File Difference templates	\$ECS_HOME/CUSTOM/data/DPS /DpAtMgrBinDiff*	Binary File Difference templates	DpAtMgrBinDiff
Binary File Difference example code	\$ECS_HOME/CUSTOM/data/DPS /DaacBinDiff*	Binary File Difference example code	DpAtMgrBinDiff
Sample PDPS Metadata PGE ODL file	\$ECS_HOME/CUSTOM/data/DPS /PGE_ODL.template	Examples of PDPS PGE Metadata ODL files	DpAtPdpsDbUpdateScience (sample input)
Sample PDPS Metadata ESDT ODL file	\$ECS_HOME/CUSTOM/data/DPS /ESDT_ODL.template	Documentation of PDPS ESDT Metadata ODL files	DpAtPdpsDbUpdateScience (sample input)
Sample PDPS Metadata TILE ODL File	\$ECS_HOME/CUSTOM/data/DPS /TILE_ODL.template	Documentation of PDPS TILE Metadata ODL Files	DpAtPdpsDbUpdateScience (sample input)
Sample PDPS Metadata PATHMAP ODL File	\$ECS_HOME/CUSTOM/data/DPS /PATHMAP_ODL.template	Documentation of PDPS PATHMAP Metadata ODL Files	DpAtPdpsDbUpdateScience (sample input)
Sample PDPS Metadata ORBIT ODL file	\$ECS_HOME/CUSTOM/data/DPS /ORBIT_ODL.template	Example of PDPS ORBIT Metadata ODL files	DpAtPdpsDbUpdateScience (sample input)

The SSI&T process also uses the PDPS database. The PDPS database for Release 4 is ECS document 311-CD-106-005. The operator may have to identify individual data fields by examination of the descriptions in the documentation. Some data can be directly accessible through the database software.

4.5.1.5 Special Constraints

None.

4.5.1.6 Outputs

Output files explicitly associated with the SSI&T custom software have been shown in the previous sections. In addition, many of the custom tools can produce output files. See the appropriate section on the custom tools for specific information.

4.5.1.7 Event and Error Messages

The SSI&T Manager displays errors in a popup message box (see Appendix A for additional details). The SSI&T Manager also logs error and status messages; currently these are sent to standard output. Refer to the specific vendor manuals for COTS messages.

4.5.1.8 Reports

A variety of ad-hoc and canned reports are available to the DAAC operations staff to assist in the monitoring of the activities associated with the Algorithm Integration & Test. These reports are listed in Table 4.5.1-18.

Table 4.5.1-18. Reports (1 of 2)

Report Type	Report Description	When and Why Used
Algorithm Integration & Test Management Reports	Provide the operations staff information on Algorithm Integration and Test application software events that have occurred. This information is available from the MSS database.	
	Ad-hoc reports include: Production Requests and associated Data Processing Requests, Data Subscriptions, PGE Profiles, etc. This information is available from the PDPS database.	These reports can be used to track modifications and provide historical information on these data objects.
PGE Profile Reports	Catalogs the resource profile information associated with a PGE (e.g., generation size of PGE Output data, CPU Wall Clock Time Used, CPU actual time Used, I/O Operations). A profile is captured for each type of machine, i.e., Sun, SGI, etc., for which the PGE is to execute. Statistics are collected to establish standard deviations, variances, and averages of resource profile values.	These reports are used to collate information for a PGE, for a type of resource, or for a given group of PGEs used to fulfill a Production Request.
I & T Activity Report	Capture information about the activities, which have occurred and activities, which are occurring in the Algorithm Integration and Test environment.	

Table 4.5.1-18. Reports (2 of 2)

Report Type	Report Description	When and Why Used
PGE Profile Update Report	Capture information to track the updates, which have occurred in the PDPS Database to PGE Profiles.	
PGE I&T Reports	Capture information on PGEs as they progress through the AI&T process. These reports include: – Code Analysis Report – Standards Checker Report – File Comparison Reports – AI & T Discrepancy Reports – Inspection Reports – Integration Reports – Acceptance Reports	These reports are to trouble shoot problems and provide tracking and trend analysis guidance.

4.5.1.8.1 Sample Reports

These reports are described in Sections 4.7, “Resource Planning,” 4.8, “Production Planning,” and 4.9, “Production Processing” of this document.

4.6 ECS Data Ingest

ECS Data Ingest provides the software capability to acquire data by various methods and transfer the data into the ECS system. These methods include an automated transfer from prior request information, polling with or without delivery records for data, which is placed at predetermined locations, and a media transfer method, which includes reading tapes. The ECS Data Ingest subsystem also stores and manages request information, and provides for data preprocessing storage and insertion into the appropriate ECS storage location.

The ECS Data Ingest subsystem provides a GUI which allows the operator to view past ingest activities, monitor and control ingest requests, modify system and external data provider parameters, and initiate hard media ingest. It also provides the Regenerate Failed Product Delivery Record (PDR) tool, which allows the operator to generate Product Delivery Records for each granule that failed in a PDR.

This page intentionally left blank

4.6.1 Data Ingest GUI

The Data Ingest tool allows the operators to view past ingest activities, monitor and control ingest requests, modify system and external data provider parameters, and initiate hard media ingest. Table 4.6.1-1 provides details on this tool's functions.

Table 4.6.1-1. Operator Ingest Functions

Operating Function	GUI	Description	When and Why to Use
Ingest History Log Viewing ¹	History Log Tab	<ul style="list-style-type: none"> Displays a log that contains the results of past ECS ingest requests Specifies search criteria (e.g., time range), the provider ID, data set name, and request status of the Ingest History Log for log display 	As needed or for regularly scheduled Ingest performance reports or to view previously occurred Ingest activities.
Ingest Status Monitoring	Monitor/ Control Tab	<ul style="list-style-type: none"> Monitors the status of all or a subset of the ingest requests that are in progress 	As required to verify nominal system operations.
Operator Request Update	Monitor/ Control Tab	<ul style="list-style-type: none"> Updates (i.e., cancel and resume) an on going ingest request on either request or granule level 	As needed to cancel a problem request on request/granule level; or resume a suspended request on request/granule level.
Hard Media Ingest	Media Ingest Tab	<ul style="list-style-type: none"> Operator performs hard media ingest (e.g., 8mm tape) Media Ingest Session is configured on the Operator's GUI to accept the request and submit it to the ECS system 	As needed to Ingest hard media.
Modify System Parameters	Operator Tools Tab	Operator updates system parameter values including: volume threshold, request threshold, communication retry count, completed request monitor time, request monitor screen refresh time.	As needed to tune the system to current or expected daily activity thresholds.
Modify External Data Provider Information	Operator Tools Tab	<ul style="list-style-type: none"> Operator updates parameter values for each external data provider Value includes volume threshold, request threshold and priority level 	As needed to tune the system to current or expected daily activity thresholds on an external data provider basis.

¹Individual DAAC policy determines the duration for which Ingest History Log information is stored and available for viewing.

4.6.1.1 Quick Start Using Data Ingest

To execute the ECS Data Ingest GUI from the command line prompt, type:

>**EcInGUIStart** <mode> [ea_instance <instance_name>] where:

<mode> is the ECS mode for the execution (e.g., OPS, TS1 or TS2) and

<instance_name> is an optional parameter used to start a different instance of the GUI. This is used if the GUI needs to be started multiple times in a mode. There needs to be a configuration file for each instance started.

4.6.1.2 ECS Data Ingest Main Screen

The Ingest Main screen, with the "Welcome To ECS INGEST GUI Interface" screen, is shown in Figure 4.6.1-1. The major functions available by clicking on tabs on this screen are:

“**Ingest Intro**” is the default tab that welcomes the user to the tool.

"**History Log**" provides operations personnel the capability to view ingest activities that are no longer active (see Section 4.6.1.2.1).

"**Monitor/Control**" provides operations personnel the capability to monitor and update ongoing ingest activities in the system (see Section 4.6.1.2.2).

"**Operator Tools**" provides operations personnel the capability to modify interactive user information by external data provider and/or modify system parameters (see Section 4.6.1.2.3).

"**Media Ingest**" provides operations personnel the capability to perform hard media ingest. The delivery record is required for media ingest (see Section 4.6.1.2.4).

There is a Menu Bar at the top of the window for getting help and activating less-frequently used secondary functions. The Menu Bar capability is available on all Ingest GUI screens. The following menus are available:

- "**File**" provides exit capability through the **exit** or **close** command and **print** and **save** capabilities, where applicable.
- "**Help**" displays general and context-sensitive help. Select "**On Help**" from the menu to get detailed help on using help. Select "**On Keys**" to get help on keyboard and mouse usage, and for general help on interacting with user interface components.



Figure 4.6.1-1. Ingest Main Screen Shown in the Ingest Intro Tab

4.6.1.2.1 History Log Tab

The History Log tab shown in Figure 4.6.1-2 provides operations personnel with the capability to view ingest requests, which are no longer monitored by the Ingest system.

Operations personnel can selectively view the Ingest History Log based on the following search criteria:

- **Start and Stop Date/Time**
- **Data Provider**
- **Data Type**
- **Final Request Status**

Operations personnel can request a **Detailed Report** or just a **Summary Report** for the specified search criteria. The **Display** button populates the log window with the detailed or summary information selected. The “**History Log**” information is displayed in the history log information window.

Operations personnel can print requested History Log information by choosing the “**Print**” option from the **File** Menu.

Operations personnel can save requested History Log information by choosing the “**Save**” option from the **File** Menu. The file is saved in the operator’s default directory. This option is useful for saving Ingest History and Performance reports to a file for transmission to the SMC. History and Performance report generation is discussed in Section 4.6.1.8.

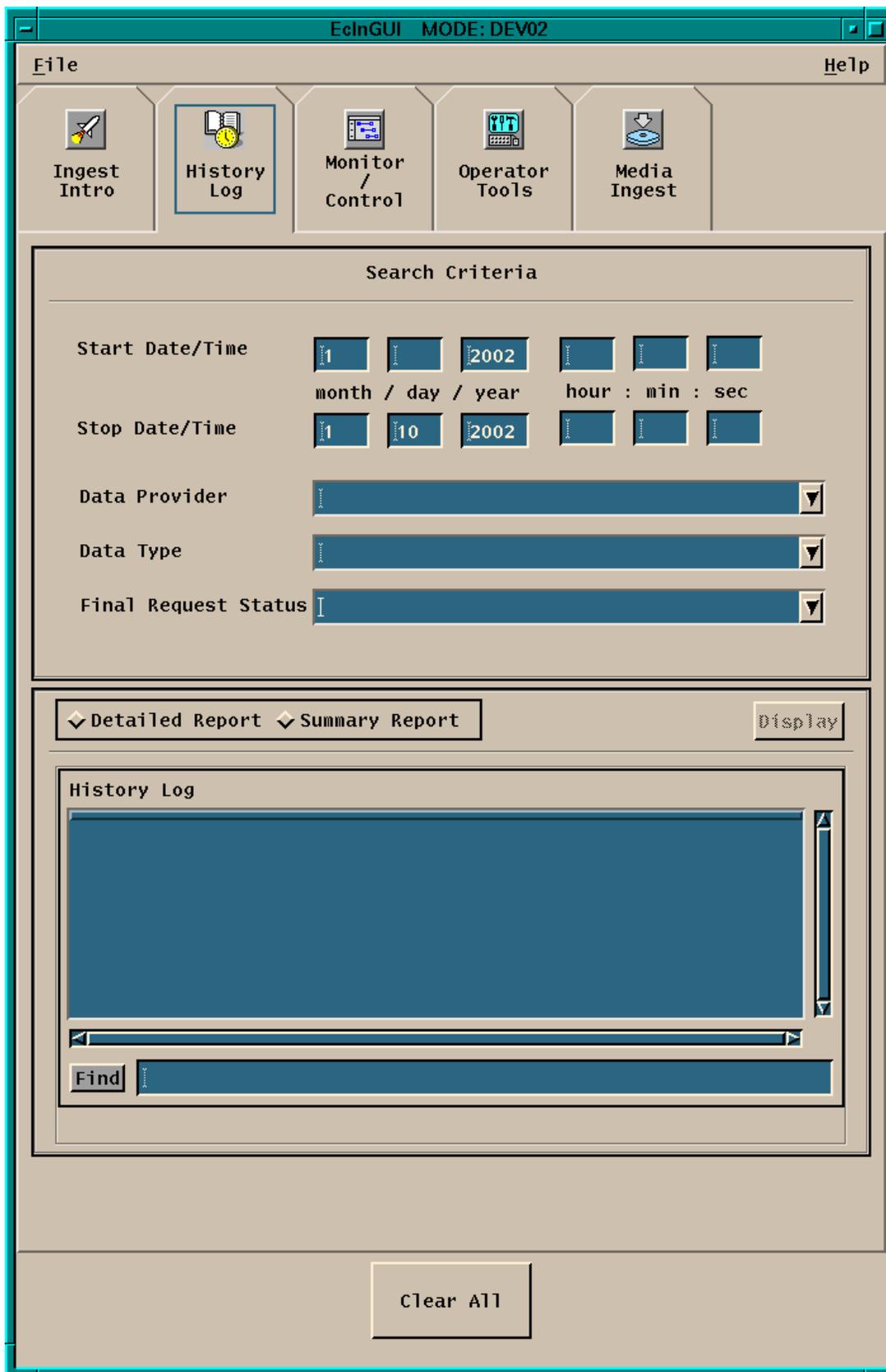


Figure 4.6.1-2. History Log Tab

Fields on the “History Log” tab are defined in Table 4.6.1-2 below.

Table 4.6.1-2. History Log Field Descriptions

Field Name	Data Type	Size	Entry	Description
Start Date/Time	Date/Time	mm/dd/yyyy hh:mm:ss	Operator Selected (Default = Current Date/Time - 24 hrs.)	Entry fields for the Start Date and Time.
Stop Date/Time	Date/Time	mm:dd:yyyy hh:mm:ss	Operator Selected (Default = Current Date/Time)	Entry fields for the Stop Date and Time.

Additional functionality is made available to the operator on the History Log tab through the following graphical components:

- The pull down menu “**File**”, that provides the capability to exit the GUI;
- The pull down menu “**Help**”, that displays general and context sensitive information;
- The radio buttons “**Summary Report**” and “**Detailed Report**”, allow toggling between the two possible optional types of report.
- The button “**Clear All**” that clears the current screen.
- The “**Display**” button will show search results from user input.

The selection of possible filters for the history log is automated through the availability of three lists: “**Data Provider**”, “**Data Type**”, and “**Final Request Status**” from which the operator can chose a number of possible options.

4.6.1.2.2 Monitor/Control Tab

Clicking on the Monitor/Control tab brings up the **Monitor/Control** tab as shown in Figure 4.6.1-3. This tab provides operations personnel with the capability to monitor ongoing ingest activities in the system. The Control feature allows the operator to cancel a request on either request or granule level. It also allows a suspended request to be resumed at either request or granule level.

The Ingest Monitor/Control tab is divided into four regions:

- Search By allows operations personnel to view all or selected ingest requests in the system.
- View Selection and Action Type button. The tab defaults to show the text view for the request information. If the graphical view button is selected, the text view region is shown as indicated in Figure 4.6.1-4.
- Information display -- allows the operator to display information in two ways based upon the selection of **Graphical View** or **Text View**
- Control Types – designed to provide operations personnel with the capability to update ongoing ingest activities in the system. Update services include **Resume** and **Cancel**.

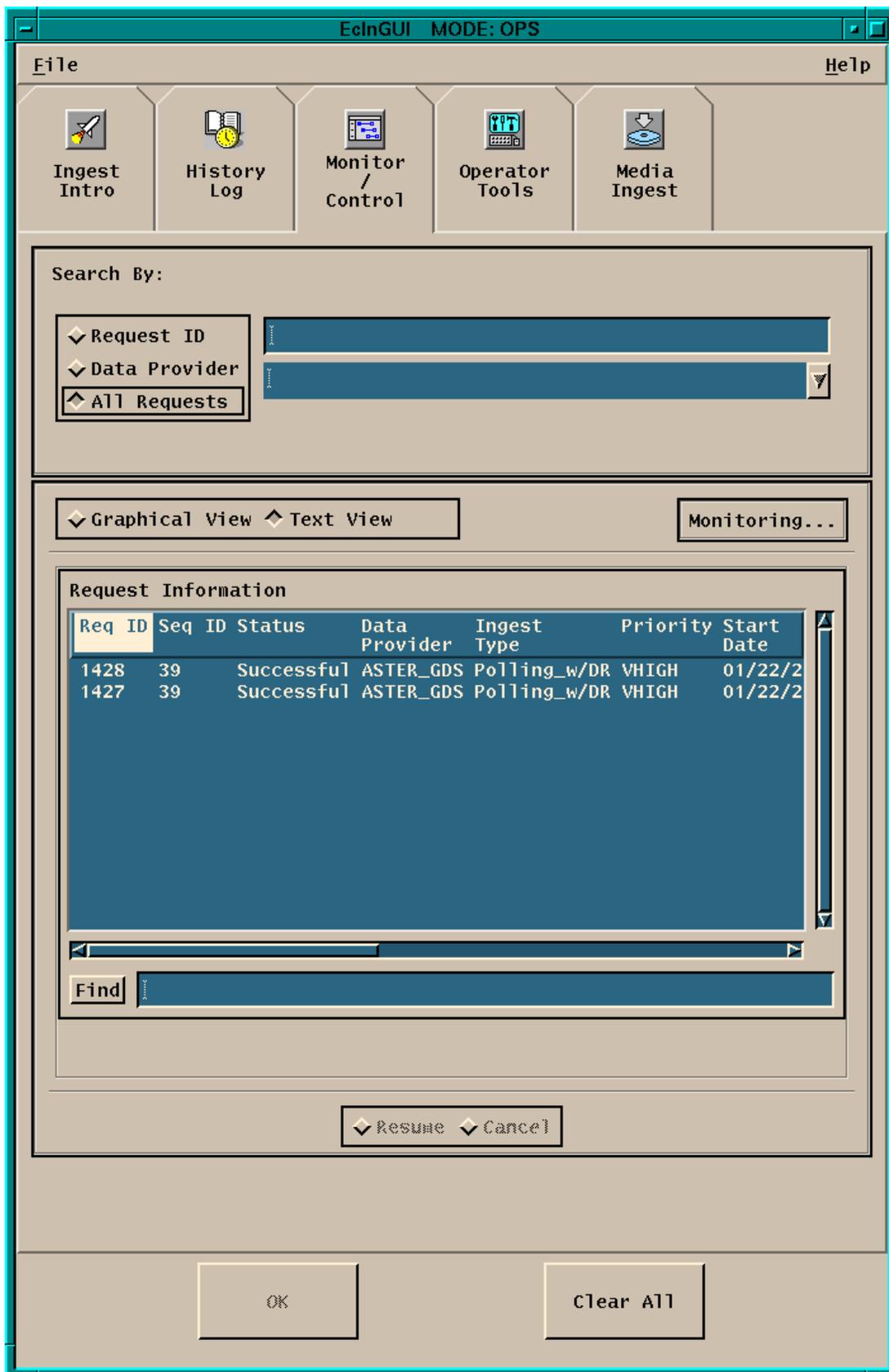


Figure 4.6.1-3. Monitor/Control Tab (Text View)

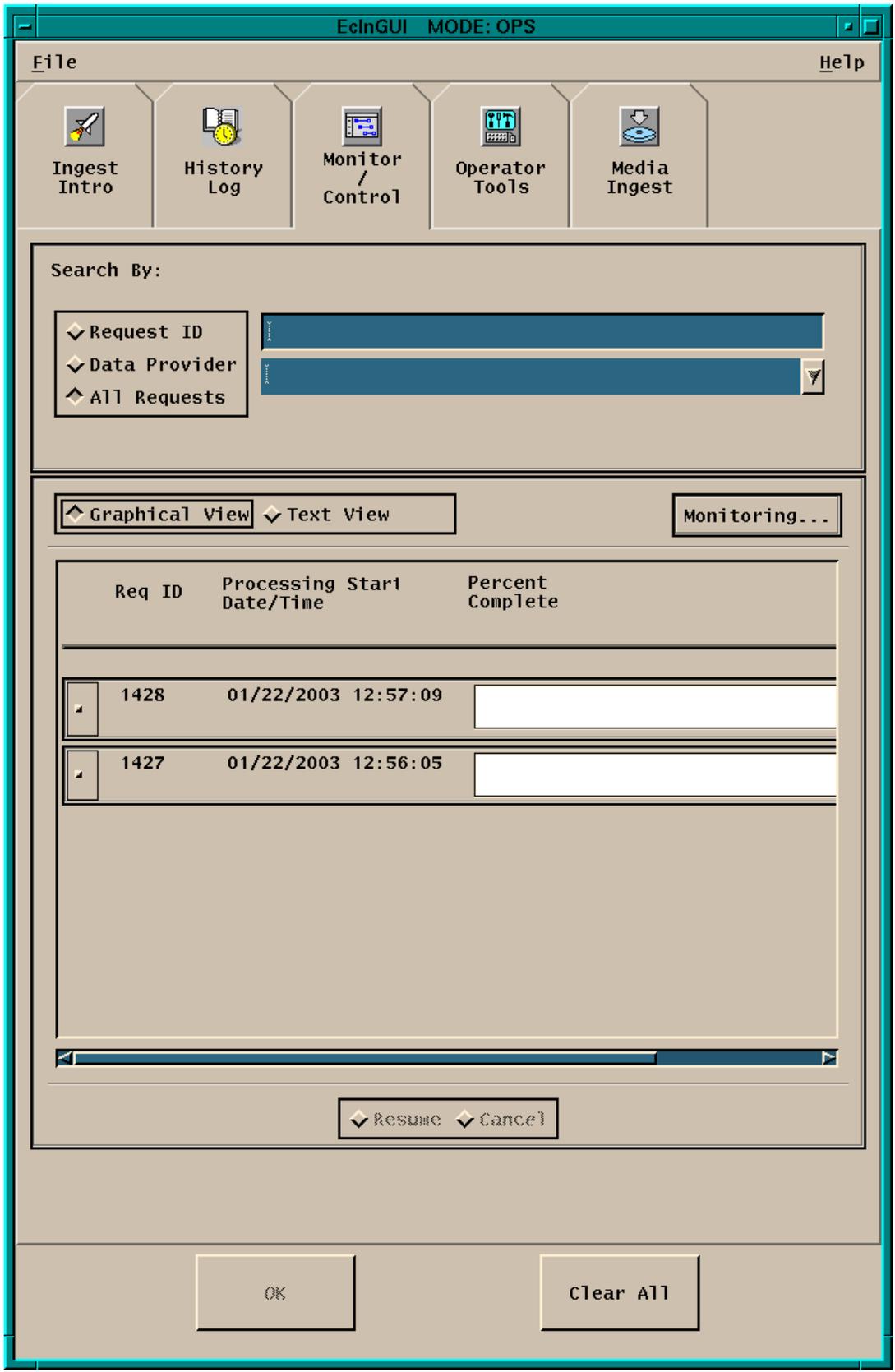


Figure 4.6.1-4. Monitor/Control Tab (Graphical View)

Table 4.6.1-3 describes the fields available on the Monitor/Control tab.

Table 4.6.1-3. Ingest Monitor/Control Tab Field Descriptions

Field Name	Data Type	Size	Entry	Description
Request ID	Integer	N/A	Operator Selected	Searches for a specific request currently in the ingest system.

Additional functionality is offered to the operator through the following graphical components:

- The “**File**” pull down menu that provides **exit** capability;
- The “**Help**” pull down menu that displays general and context-sensitive help;
- The “**Monitoring/Control...**” event label is used to display the current state of the underlining process. “**Monitoring...**” lets the operator view the list of the current requests, with information including the **Request ID**, the **Processing Start Date/Time**, and the **Percent Complete** in the Graphical view and the **Request Id**, the **Status**, the **Data Provider**, the **Ingest Type**, the **Priority**, and the **Start Date** in the Text View. “**Control...**” allows the operator to cancel or resume a request or a granule from a list;
- The pull down list “**Data Provider**”, which allows the operator to select from a list of valid external data providers;
- The “**All Requests**” radio button that displays all requests currently in the system;
- The “**Resume**” radio button that resumes the processing of a previously suspended ingest request on either request or granule level; multiple requests can be selected and resumed at one time on the request level; the resume button is active if the first highlighted request can be resumed;
- The “**Cancel**” radio button that aborts the processing of an ongoing ingest request on either request or granule level; multiple requests can be selected and canceled at one time on the request level; the cancel button is active if the first highlighted request can be cancelled;
- The sort capability at the request level is on all fields except “**End Date**” and “**End Time**” in text view; fields sort in descending order, except the “**Req ID**” field which sorts in ascending order.

4.6.1.2.3 Operator Tools Tab

Clicking on the Operator Tools tab brings up the Operator Tools window shown in Figure 4.6.1-5. This tab provides the means to set system-level and data provider-level threshold and setup information for simple throttling of the ingest process. In particular, operations staff may control both the number of requests to be processed concurrently and the amount of data to be ingested concurrently (for the entire site and for each data provider individually). Note: All system-level and data provider-level thresholds and setup information have initial values at the time of delivery.

The **Operator Tools** tab is divided into three sub-tabs:

- **Modify External Data Provider / User Information** sub-tab allows setup of thresholds and user information for external data providers, including authorized science users who interactively request data ingest (see Figure 4.6.1-5)
- **Modify System Parameters** sub-tab allows setup of thresholds and system information on a system-wide basis (see Figure 4.6.1-7)
- **File Transfer** sub-tab allows setup of external node information (see Figure 4.6.1-8)

See Tables 4.6.1-4, 4.6.1-6, and 4.6.1-7 below for a description of fields that appear, respectively, in the **Modify External Data/User Information**, **Modify System Parameters**, and **File Transfer** sub-tabs.

In addition to the above input fields, the three Operator Tools tabs include other graphical elements that provide additional functionality to the operator.

“**File**” and “**Help**” pulldown menus allow for exiting the capability and displaying of context sensitive help information, respectively.

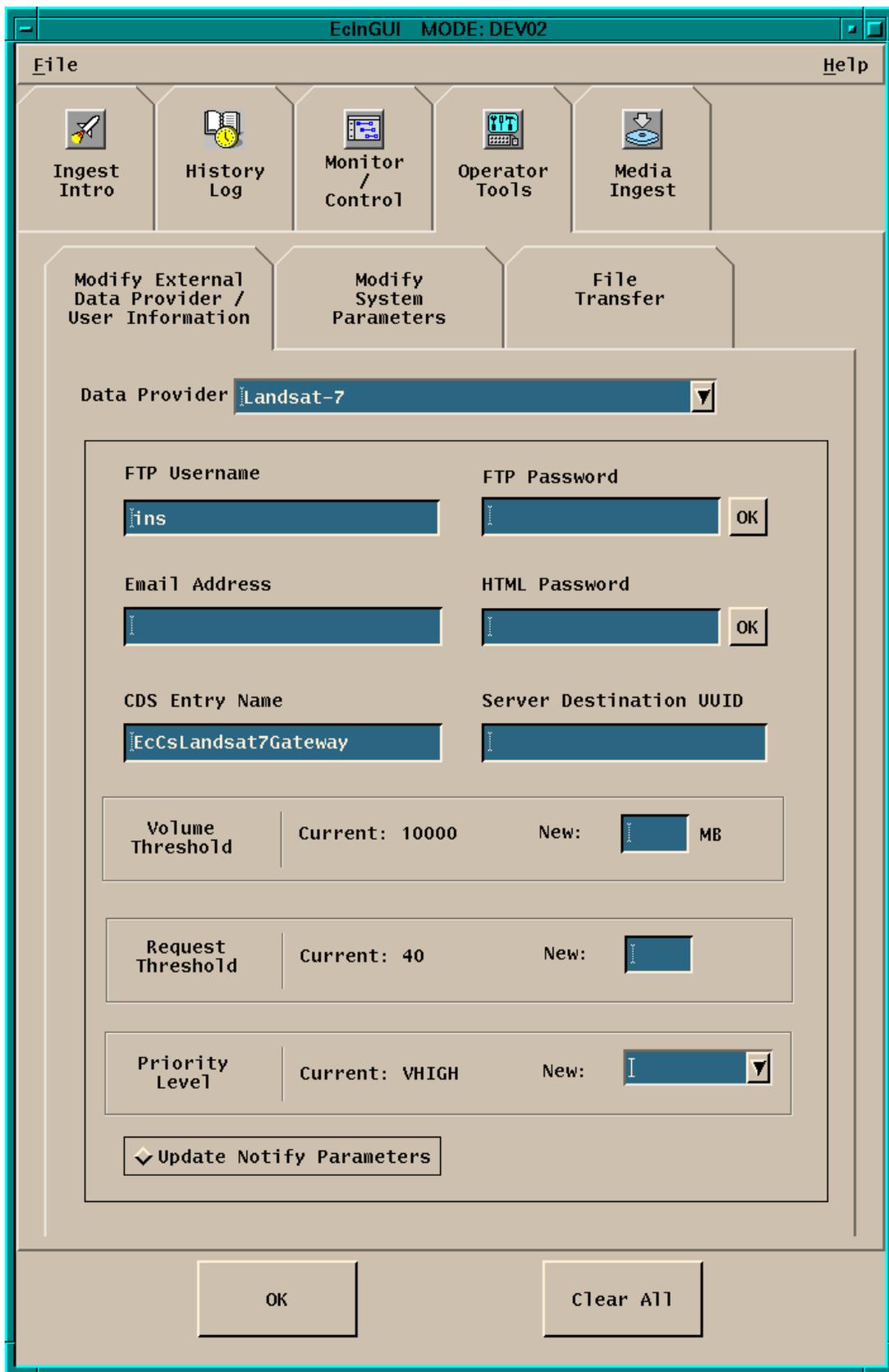


Figure 4.6.1-5. Operator Tools - External Data Provider/User Information Sub-tab

The Modify External Data Provider/User Information sub-tab includes a pull down list that allows for the selection of valid External **Data Provider** IDs. The operator clicks on **OK** to implement the changes or **Clear All** to delete all the modifications operated so far.

**Table 4.6.1-4. Operator Tools - External Data/User Information Tab
Field Descriptions**

Field Name	Data Type	Size	Entry	Description
FTP Username	Text Field	10	Operator Selected (Default = current value or blank, if no current value exists)	Username of the external data provider at a remote node holding data.
FTP Password	Text Field	30	Operator Selected (Default = blank)	<ul style="list-style-type: none"> • Provides an input entry field for FTP Password • The FTP Password is NOT reflected to the screen when the operator inputs information (and is encrypted prior to storage)
Email Address	Text Field	255	Operator Selected (Default = blank)	Email address of external data provider.
HTML Password	Text Field	30	Operator Selected (Default = blank)	<ul style="list-style-type: none"> • Not used
CDS Entry Name	Text Field	255	Operator Selected (Default = current value or blank, if no current value exists)	Provides the CDS entry for the given data provider.
Server Destination UUID	Text Field	36	Operator Selected (Default = current value or blank, if no current value exists)	Provides the UUID for the given data provider.
Volume Threshold	Integer	N/A	Operator Selected	Provides the current ingest volume threshold for the given data provider and an input area to set a new value.
Request Threshold	Integer	N/A	Operator Selected	Provides the current ingest request threshold for the given data provider and an input area to set a new value.

The **Priority Level** of the Data Provider can be changed using a list of valid priority levels (default is VHIGH).

If the toggle button **Update Notify Parameters** is selected in the **External Data Provider/User Information** sub-tab the **Notify Parameter** pop-up appears as shown in Figure 4.6.1-6.

Table 4.6.1-5 describes the fields contained in this pop-up. With the **OK** and **Cancel** buttons on the Notify Parameter, the operator can accept or delete the changes made.

The image shows a dialog box titled "Notify Parameters". It contains four text input fields:

- Notify FTP Node
- Notify FTP Directory
- Notify FTP Username
- Notify FTP Password

There is a small "OK" button to the right of the "Notify FTP Password" field. At the bottom of the dialog, there are two larger buttons: "OK" and "Cancel".

Figure 4.6.1-6. Update Notify Parameters Pop-up

Table 4.6.1-5 describes the fields on the Update Notify Parameters Pop-up.

Table 4.6.1-5. Update Notify Parameters Field Descriptions

Field Name	Data Type	Size	Entry	Description
Notify FTP Node	Text Field	255	Operator Selected (Default = blank)	Provides Notify FTP node for the given data provider.
Notify FTP Directory	Text Field	255	Operator Selected (Default = blank)	Provides Notify FTP directory for the given data provider.
Notify FTP Username	Text Field	10	Operator Selected (Default = blank)	Provides Notify FTP username for the given data provider.
Notify FTP Password	Text Field	30	Operator Selected (Default = blank)	<ul style="list-style-type: none"> • Provides an input entry field for Notify FTP Password • The Notify FTP Password is NOT reflected to the screen when the operator inputs information (and is encrypted prior to storage)

Figure 4.6.1-5 is the Modify System Parameters sub-tab.

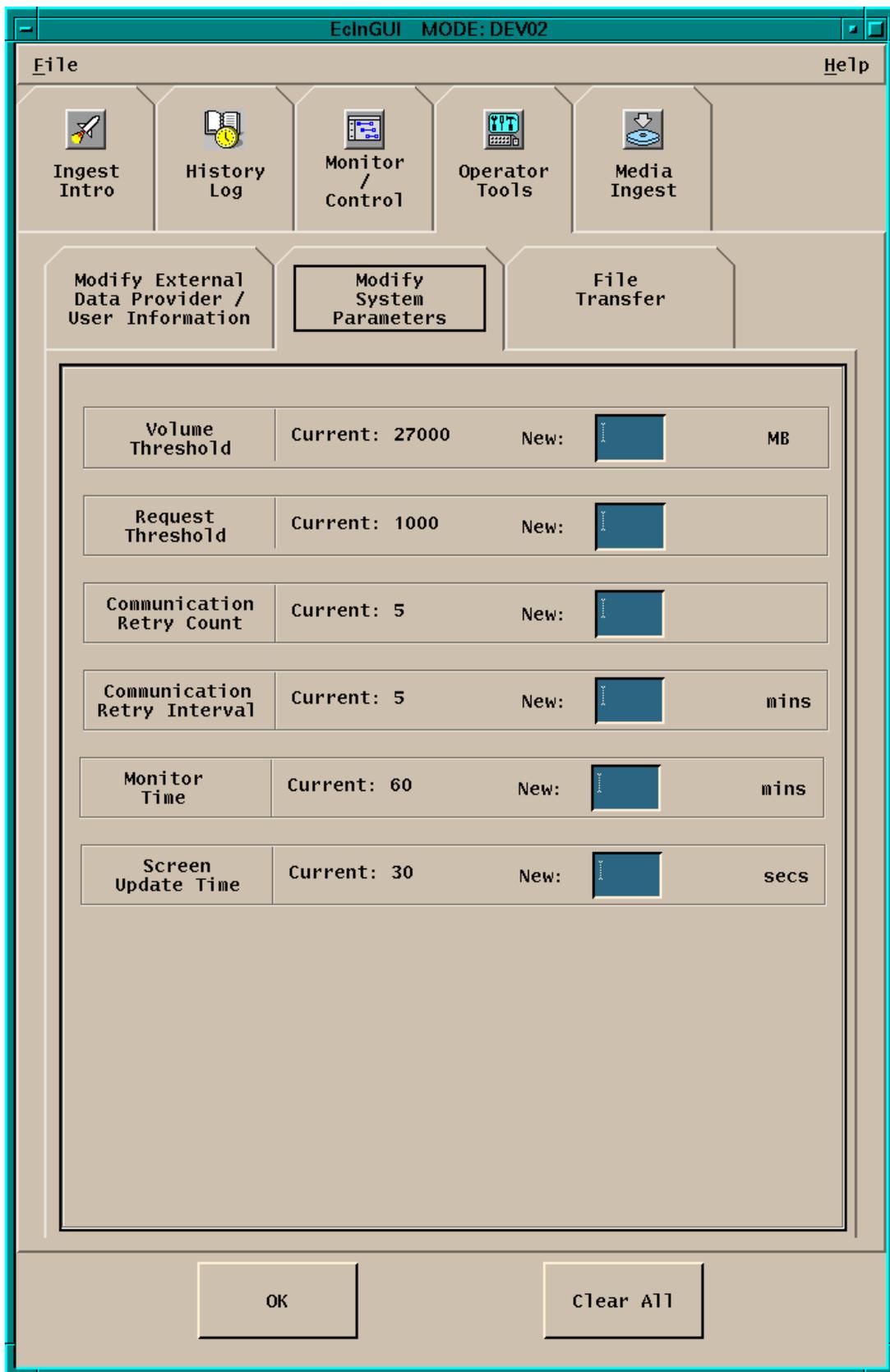


Figure 4.6.1-7. Operator Tools - Modify System Parameters Sub-tab

Table 4.6.1-6 describes the fields on the Modify System Parameters sub-tab.

**Table 4.6.1-6. Operator Tools Modify System Parameters
Field Descriptions**

Field Name	Data Type	Size	Entry	Description
Volume Threshold	Integer	N/A	Operator Selected (Default = current value)	Provides the current system volume threshold and an input area to set a new value.
Request Threshold	Integer	N/A	Operator Selected (Default = current value)	Provides the current system setting for ingest maximum requests and an input area to set a new value.
Communication Retry Count	Integer	N/A	Operator Selected (Default = current value)	Provides the current Communication Retry Count and an input area to set a new Communication Retry Count.
Communication Retry Interval	Integer	N/A	Operator Selected (Default = current value)	Provides the current system setting for communication retry interval in minutes and an input area to set a new value.
Monitor Time	Integer	N/A	Operator Selected (Default = current value)	<ul style="list-style-type: none"> • Provides the current amount of time (in minutes) that a completed request is retained on the monitoring display prior to removal • Provides an input area to set a new monitor time
Screen Update Time	Integer	N/A	Operator Selected (Default = current value)	<ul style="list-style-type: none"> • Provides the current system setting for screen update interval in seconds and an input area to set a new value

The **OK** and **Clear All** buttons allow operators to accept or delete changes to the System Parameters.

Figure 4.6.1-8 is the File Transfer sub-tab.

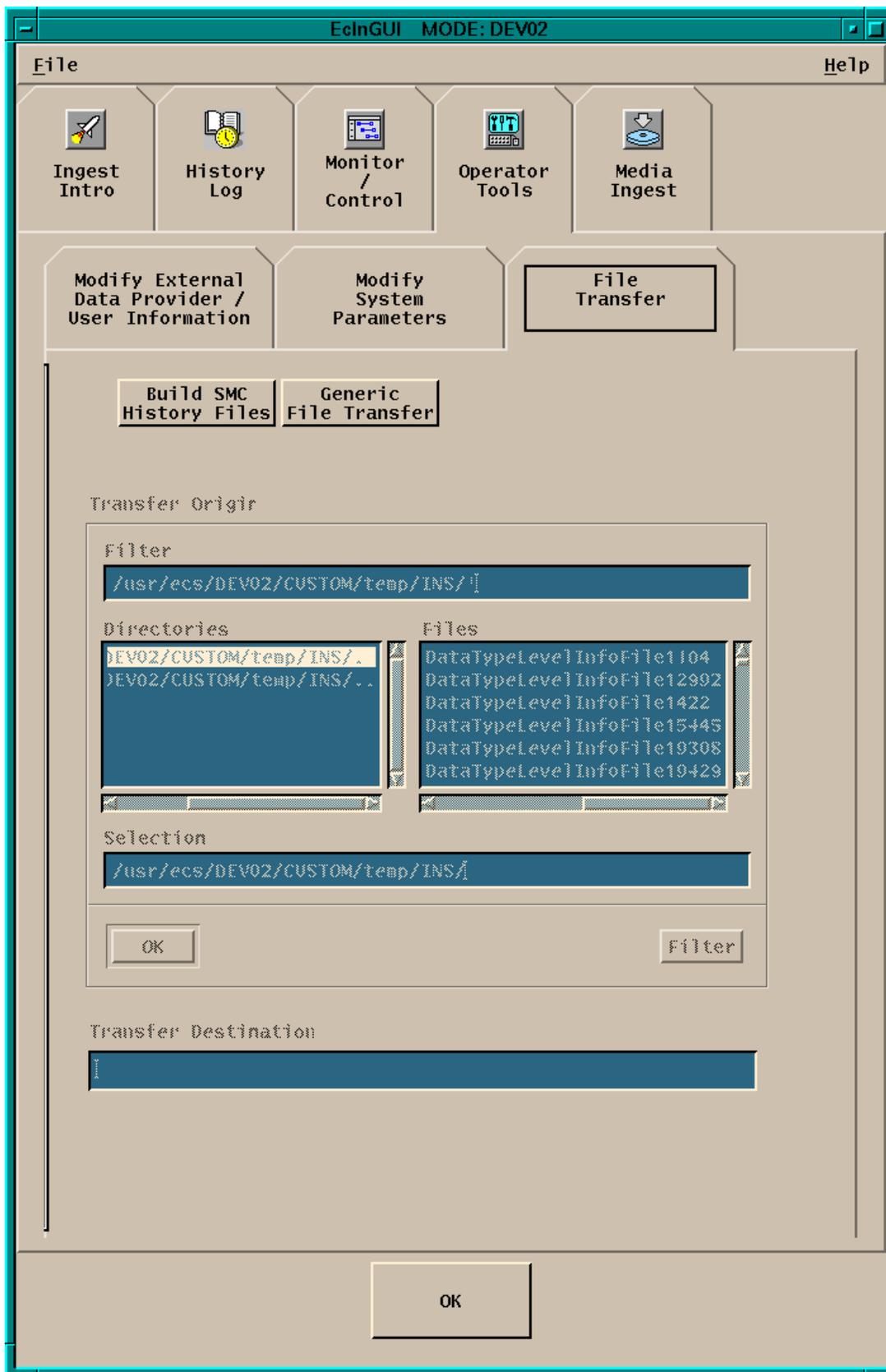


Figure 4.6.1-8. Operator Tools - File Transfer Sub-tab

Table 4.6.1-7 describes the fields on the File Transfer sub-tab.

Table 4.6.1-7. Operator Tools - File Transfer Field Descriptions

Field Name	Data Type	Size	Entry	Description
Filter	Text Field	Unlimited	Operator Selected	Provides a default directory for file selection.
Selection	Text Field	Unlimited	Operator Selected	Provides the name of file for transfer.
Transfer Destination	Text Field	Unlimited	Operator Selected	Provides the file transfer destination.
Directories	Character	Unlimited	System Generated	List of directory available for filtering.
Files	Character	Unlimited	System Generated	List of files available for filtering.

The buttons at the top of the window allow specifying if the operation objective is to perform a **Generic File Transfer** or to **Build SMC History Files**. With the **OK** button, the operator initiates the desired file transfer.

4.6.1.2.4 Media Ingest Tab

Clicking on the Media Ingest tab brings up the **Media Ingest** window as shown in Figure 4.6.1-9. This tab provides DAAC operations personnel with the capability to perform media ingest (e.g., specifying media for ingest, such as 8mm tape).

The Ingest Media Ingest tab is divided into three regions:

- Media Type – allows the operator to select two types of media (8mm tape and DTF tape)
- Data Provider and Media ID – allows the operator to specify data provider ID and Media ID. The Media ID needs to match the Media ID loaded into the stacker/drive using the STMGT GUI.
- Data Delivery Record File Location– allows the operator to specify the location of the media data delivery record file, which is required for media ingest. The file could either be embedded in the hard media or be made available electronically. If not embedded on the hard media, the Delivery Record file must be in an ECS disk location. The external data provider must connect with that location via ftp prior to delivering the hard media to ECS.

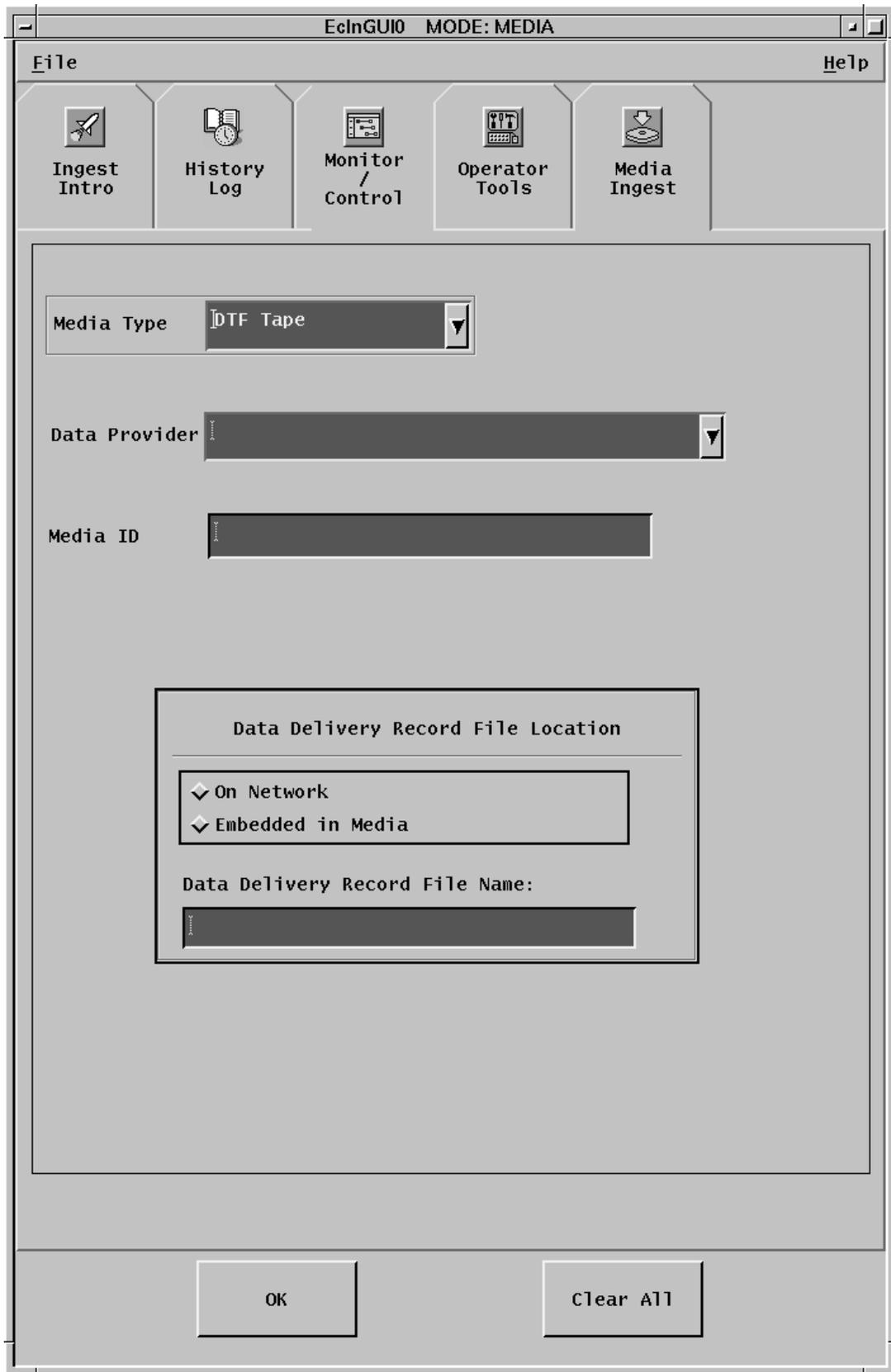


Figure 4.6.1-9. Media Ingest Tab

Table 4.6.1-8 describes the fields on the Ingest Media tab.

Table 4.6.1-8. Ingest Media Field Descriptions

Field Name	Data Type	Size	Entry	Description
Media ID	Integer	N/A	Operator Selected	Media Volume ID input.
Data Delivery Record File Name	Character	Unlimited	Operator Selected	Name of the Data Delivery Record file.

Media Type and **Data Provider** can be selected from lists of valid names. The location of the Data Delivery Record file can be selected through by toggling the **On Network** and **Embedded in Media** buttons.

With the **OK** and **Clear All** buttons the Operator can accept start the Ingest session or delete the entries input in any of the fields.

4.6.1.3 Required Operating Environment

For information on the operating environment, tunable parameters, and environment variables refer to the 920-TDx-013 “Custom Code Configuration Parameters” documentation series. The “x” refers to the installed location (e.g., 920-TDG-013 is for the GSFC DAAC). These documents can be found in the Technical Documents of the ECS Baseline Information System.

The following Table 4.6.1-9 identifies the supporting products the Data Ingest depend upon in order to function properly in the ECS environment:

Table 4.6.1-9. ECS Data Ingest Product Dependency

Product Dependence	Protocol Used	Comments
INS GUI	X-11	Via client libraries.
INS GUI and ReqMgr	TCP/IP	Via RPC call.

4.6.1.4 Databases

The ECS Data Ingest Tool interfaces with the Ingest Database. For details about the design and schema of the Ingest Database refer to 311-CD-601-001, *Ingest Database Design and Schema Specifications*.

4.6.1.5 Special Constraints

None.

4.6.1.6 Outputs

Operations staff has access to two output logs, described in Table 4.6.1-10.

Table 4.6.1-10. Outputs

Output	Description and Format
Ingest history log	The Ingest history log contains summary information about ingest request status. Stored in Sybase.
Ingest event log	The Ingest event log contains critical and detailed events of interest to DAAC operations staff. Stored in a flat file.

4.6.1.7 Event and Error Messages

The Ingest interface issues both status and error messages to the event log file. Error messages fall under the following categories: common messages, history log messages, ingest request controller messages, and ingest threshold controller messages. Both event and error messages are listed in Appendix A.

4.6.1.8 Ingest Reports

In addition to ad hoc ongoing request status displays discussed above, the Ingest subsystem provides the standard reports described in Table 4.6.1-11. These reports can be invoked through the History Log Tab discussed in Section 4.6.1.2.1.

Table 4.6.1-11. Standard Ingest Production Reports

Report Type	Report Description	When and Why Used
Ingest Request History Report	The report supplies operations staff with a view of ingest request completion performance. It provides a detailed log of the ingest requests in the reporting period (including requester, data source, data type, the times of various ingest events such as request receipt and completion, data volume, etc.). The report also provides summary statistics for the reporting period, such as completed vs. unsuccessful requests, backlog development, average ingest volumes and processing times broken down by various categories.	The report may be generated for specified time periods and executed on a regular basis (based on site policy).
Ingest Request Performance Report	The report supplies operations staff with a view of ingest performance. It provides summary statistics for the reporting period, such as completed vs. unsuccessful requests, backlog development, average ingest volumes and processing times broken down by various categories.	The report may be generated for specified time periods and executed on a regular basis (based on site policy).
Ingest Granule Performance Report	The report supplies operations staff with a view of ingest granule-level performance. It provides summary statistics for the reporting period, such as completed vs. unsuccessful requests, backlog development, average ingest volumes and processing times broken down by various categories.	The report may be generated for specified time periods and executed on a regular basis (based on site policy).

4.6.1.8.1 Sample History Reports

An example Ingest Request History Report is shown in Figure 4.6.1-10.

Req ID	Data Provider	Compl. Status	Ingest Type	Start Time	End Time	Total Granules	Success Granules	Data Volume	File Count	Transfer Time	Pre - Proc Time	Archive Time	Priority	Restart Flag
00100	TSDIS	OK	PD	01-05-97 12:31:06	01-05-97 13:51:16	1	1	112.5	5	53.6	3.2	23.4	----	---
.....	-----	-----

Figure 4.6.1-10. Sample Ingest Request History Report

An example Ingest Request Performance Report is shown in Figure 4.6.1-11.

Data Provider	Total Requests	Total Request Errors	Granules		Files		Size		Transfer Time		Pre - Proc Time		Archive Time	
			Avg	Max	Avg	Max	Avg	Max	Avg	Max	Avg	Max	Avg	Max
SDPF	112	2	24	112	62	33 6	106. 5	114.5	51	105	7	23	12	212
TSDIS												

Figure 4.6.1-11. Sample Ingest Request Performance Report

An example Ingest Granule Performance Report is shown in Figure 4.6.1-12.

Data Provider	Data Type	Total Granules	Total Granules with Errors	Files Avg Max		Granule Vol. Avg Max		Transfer Time Avg Max		Pre - Proc Time Avg Max		Archive Time Avg Max	
SDPF	CER00	112	2	62	336	106.5	114.5	51	105	7	23	12	212
SDPF

Figure 4.6.1-12. Sample Ingest Granule Performance Report

This page intentionally left blank.

4.6.2 Regenerate Failed PDR Tool

The Regenerate Failed PDR tool provides the ECS Operations Staff with the capability to generate a Product Delivery Record (PDR) for each failed granule in a PDR and copy the generated PDRs to an Ingest polling directory using a command line interface. This tool can be used when a PDR fails with a long Production Acceptance Notification (PAN) message file. A long PAN means that the request had more than one granule and not all granules had the same error. The purpose of the tool is to provide a means for the ECS Operations Staff to easily resubmit only failed granules to Ingest polling, rather than having to manually edit the original PDR file or resubmit all of the granules, which would create duplicate granules in the archive.

4.6.2.1 Quick Start Using the Regenerate Failed PDR Tool

Entering the following command starts the regenerate failed PDR tool:

```
>EcInRegenFailedPDRStart <mode>
```

The **mode** parameter specifies the mode in which the program is to run. The tool can run in any mode (e.g., OPS, TS1 or TS2). The DAAC operations staff establishes the modes.

4.6.2.2 Regenerate Failed PDR Tool Main Screen

There is no Graphical User Interface for the Regenerate Failed PDR tool. The DAAC operations user interacts with the Regenerate Failed PDR Tool by responding to the following prompts output by the program.

```
1. Generate PDRs
2. Quit
>> 1
Please enter PDR filename with path
>>
Please enter PAN filename with path
>>
Please enter the path of the Polling directory into which the PDRs should be copied
>>
The new PDR file <filename> was created successfully.
Please inspect this PDR file and correct any errors found.
Do you want this PDR to be moved to the Polling directory (y/n)?
If n is entered:
    Do you want to delete the PDR file <filename> (y/n)?
```

4.6.2.3 Required Operating Environment

For information on the operating environment, tunable parameters, and environment variables refer to the 920-TDA-022 “Custom Code Configuration Parameters” documentation series.

4.6.2.3.1 Interfaces and Data Types

The Regenerate Failed PDR tool is a stand-alone tool, so it has no dependencies on supporting products.

4.6.2.4 Databases

The Regenerate Failed PDR tool does not include the direct managing of or interfacing with any database.

4.6.2.5 Special Constraints

The Regenerate Failed PDR Tool needs to have access to the PDR and PAN files, which are to be used. It also needs to have access to the Ingest polling directory into which the generated PDRs are to be moved. Note also that the Regenerate Failed PDR Tool is started through a start script, which specifies that the EcInRegenFailedPDR.CFG configuration file be used to initialize the environment.

4.6.2.6 Outputs

The Regenerate Failed PDR Tool's output consists of prompts returned to the command line interface, error messages returned to the command line interface as described in Section 4.6.2.7, messages written to the application log file (EcInRegenFailedPDR.ALOG), and the generated PDR files.

4.6.2.7 Event and Error Messages

Refer to Appendix A for the error messages displayed by the Regenerate Failed PDR Tool.

4.6.2.8 Reports

None.

4.7 Resource Planning

This section describes the Resource Planning tools used by DAAC operators. These tools are accessible through Resource Planning GUIs.

This page intentionally left blank.

4.7.1 Resource Planning

Resource planning is accomplished by operations through a Resource Editor tool used to define the resources allocated to run production data and a Resource Scheduler GUI used to create resource reservations. The Resource Scheduler can also display resource reservations on a time line.

The Resource Editor GUI can define hardware resources such as disks and host computers (‘real computers’) as well as virtual resources such as ‘virtual computers,’ which are the collections of CPUs and disks. The list of resources can be created from scratch or they can be initially obtained from a resource configuration in MSS.

A resource plan is defined by a set of approved resource reservations. The resource planner to create, validate and approve reservations until a conflict-free plan is achieved uses the Resource Scheduler GUI. At this point, the reservations are “committed” and the reservations become ground events. The Ground Event activities include maintenance, test, and training.

Table 4.7.1-1 provides a high level summary of the activities of the Resource Planning GUIs.

Table 4.7.1-1. Common ECS Operator Functions Performed with Resource Planning GUIs (1 of 2)

Operating Function	GUIs Involved	Description	When and Why to Use
Ingest /create and modify resource configuration.	Resource Editor, Virtual Computer Details, Disk Details, Hardware Details, String Details, Real Computer Details, AutoSys Details.	Obtains from the MSS CM system a list of system resources that are used for resource planning and to edit/add to these resource items.	<ul style="list-style-type: none"> • To define system resources. • To add and modify resources.
Enter a resource reservation request.	Resource Scheduler, Reservation Edit/Definition, Resource Selection, Intervals Selection.	The resource planner enters a request for a resource reservation.	To define the resource reservation for testing or maintenance.
Review resource reservation requests in the system.	Resource Scheduler, Reservation Edit/Definition, Resource Selection, Intervals Selection.	The resource planner reviews the resource reservations that have been made.	To support resource allocation process, to aid in preparing a resource reservation request, or to inspect the use of system resources.

Table 4.7.1-1. Common ECS Operator Functions Performed with Resource Planning GUIs (2 of 2)

Operating Function	GUIs Involved	Description	When and Why to Use
Commit approved resources.	Resource Scheduler	The resource planner allocates a resource to a particular activity.	To register the reservation as the ground event.
Review Timeline.	Resource Reservation Planning Master Timeline	A timeline oriented view of resource allocation for all configured system resources.	To aid in planning resource use within the system.

4.7.1.1 Quick Start Using Resource Planning

There are two main applications associated with Resource Planning:

- **Resource Scheduler GUI** is used to enter a new resource reservation, to modify an existing resource reservation request, to approve a resource reservation, to commit approved resource reservations, to view a timeline format display of all approved resource reservations, and to delete an existing reservation.
- **Resource Editor GUI** is used to add or delete resources or modify the characteristics of resources.

To invoke the Resource Editor GUI from the command line, enter:

EcPIRpreStart <mode> <MSGSVR_ID>

<mode> is the ECS mode for the execution.

<MSGSVR_ID> is the TBS Server Identifier.

To execute the Resource Planning Scheduler from the command line, type:

EcPIRpsistart <mode> <MSGSRV_ID>

<mode> is the ECS mode for the execution.

<MSGSVR_ID> is the TBS Server Identifier.

4.7.1.2 Resource Scheduler Main Screen

Figure 4.7.1-1 is the Resource Scheduler GUI, which appears when the Resource Scheduler Icon is clicked. This GUI presents a list of the resource reservations entered into the system, with one line of information per resource reservation. The list can be filtered by activity type by clicking on the button displayed next to “Activity Type” at the top left of the screen. The selection button lists each of the activity types defined to resource planning, e.g., ‘Testing’. Selecting an activity from this list limits the display to resource reservations with that type of activity specified.

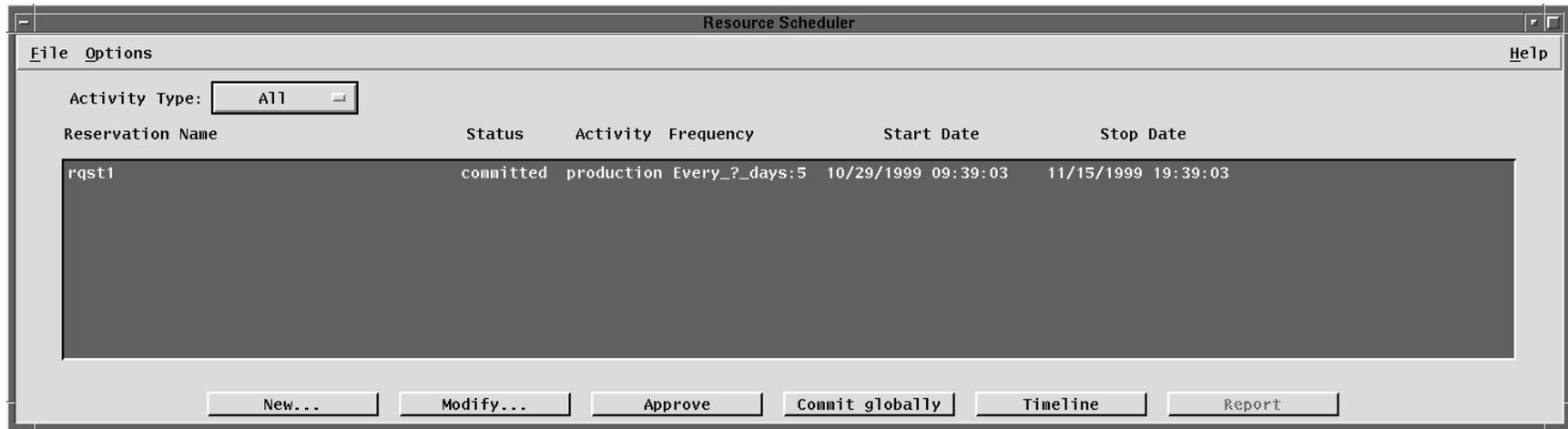


Figure 4.7.1-1. Resource Scheduler GUI

There are several buttons at the bottom of the GUI, which affect the status of the reservations. The functions associated with these buttons are also accessible from the menu bar. These buttons and their actions are:

- **New ...** - Used to enter a new resource reservation request. This brings up the Resource Reservation Request Edit/Definition GUI described in the Section 4.7.1.1.3.
- **Modify ...** - Used to edit or review the details of an existing resource reservation request. First select a resource reservation entry from the list in the Resource Planning GUI, and then click on 'Modify...' This brings up the Resource Reservation Request Edit/Definition GUI for the selected resource reservation
- **Approve** - Used to indicate that all reviews associated with the resource planning process have taken place and there are no objections to the resource usage as described by the request. Clicking on this button checks if there is any conflict between this resource reservation and other reservations. Error messages are displayed if conflicts are detected
- **Commit globally** - Commits all approved resource reservations. At this point, the information related to the activity and its reserved resources is accessible by the production planning software.
- **Timeline** - Displays a timeline-oriented view of the resource plan in the Resource Reservation Plan Timeline GUI. This GUI is discussed further in Section 4.7.1.1.4.
- In addition, on the menu bar, the pulldown menus provide the following capabilities.
- 'File' Pulldown:
 - **New** - Same as the 'New' button on the GUI - used to enter a new resource reservation request.
 - **Open** - Same as the 'Modify...' button on the GUI - used to update information for an existing resource reservation request.
 - **Delete** - deletes a resource reservation request.
 - **Exit** - exits the application.
- 'Options' Pulldown:
 - **Timeline** - Same as the 'Timeline' button on the GUI - used to display a timeline-oriented view of the resource plan.
 - **Report** – Generates a report with information about a resource reservation request.

4.7.1.2.1 Resource Reservation Request Edit/Definition GUI

From the Resource Planning GUI, a user enters a new resource reservation, or reviews an existing resource reservation. Either selection displays the Resource Reservation Request Edit/Definition GUI shown in Figure 4.7.1-2.

Resource Reservation Request Edit/Definition - Modify

Request Name:

Edited Date:

Originator:

Sponsor:

Activity: Priority:

Description:

Start Day as "MM/DD/YYYY" Start Time as "HH:MM:SS"

Stop Day as "MM/DD/YYYY" Stop Time as "HH:MM:SS"

Frequency:

Rejected
 Validated
 Status:

Comments:

Figure 4.7.1-2. Resource Reservation Request Edit/Definition GUI

Table 4.7.1-2 describes the fields of the Resource Reservation Request Edit/Definition GUI.

**Table 4.7.1-2. Resource Reservation Request Edit/Definition GUI
Field Description**

Field Name	Data Type	Size (# of characters)	Entry	Description
Request Name	ASCII characters	<= 40	User input, required	A name for the resource request.
Edited Date	Date	40	System generated	Date of resource request entry.
Originator	ASCII characters	<= 30	System generated	Userid of the user entering the resource request.
Sponsor	ASCII characters	<= 30	User input	Name of the individual designated to review and validate the Resource Request for completeness, etc.
Activity:	ASCII characters	N/A	User provided, required	Activity performed by the resource reservation request.
Priority	Integer	3	User provided	The priority for the activity.
Description	ASCII characters	30	User provided, optional	User description of the activity for which the resource is required.
Start Day	Date	10	User provided, required	The start date of the resource request.
Start Time	Time	8	User provided, required	The start time of the resource request.
Stop Day	Date	10	User provided, required	If the reservation is to be repeated at some frequency, the stop date specifies the end date for the repeated request.
Stop Time	Time	8	User provided, required	The stop time of the resource request.
Frequency:	ASCII characters	N/A	User provided, required	The Frequency: pulldown menu offers the options listed in Table 4.7.1-3 to specify how the resource request should be repeated over the term from Start Time to Stop Time. If the "Every_?_days" is selected, the user enters the interval in days in the associated text field to the right.
Status	ASCII characters	<= 20	System generated	Displays the status of the request, e.g., 'approved', 'validated', etc.
Comment	ASCII characters	Unlimited	User provided, optional	User comment on the resource reservation request.

Selecting the '**Resources ...**' button displays the Resources Selection GUI, shown in Figure 4.7.1-3, that displays two lists, one containing all of the available resources and the other listing the

resources that the operator selects for incorporation into the resource reservation. The user selects the desired resource by means of the arrow buttons. The name of the resource request is displayed, once the reservation is saved.

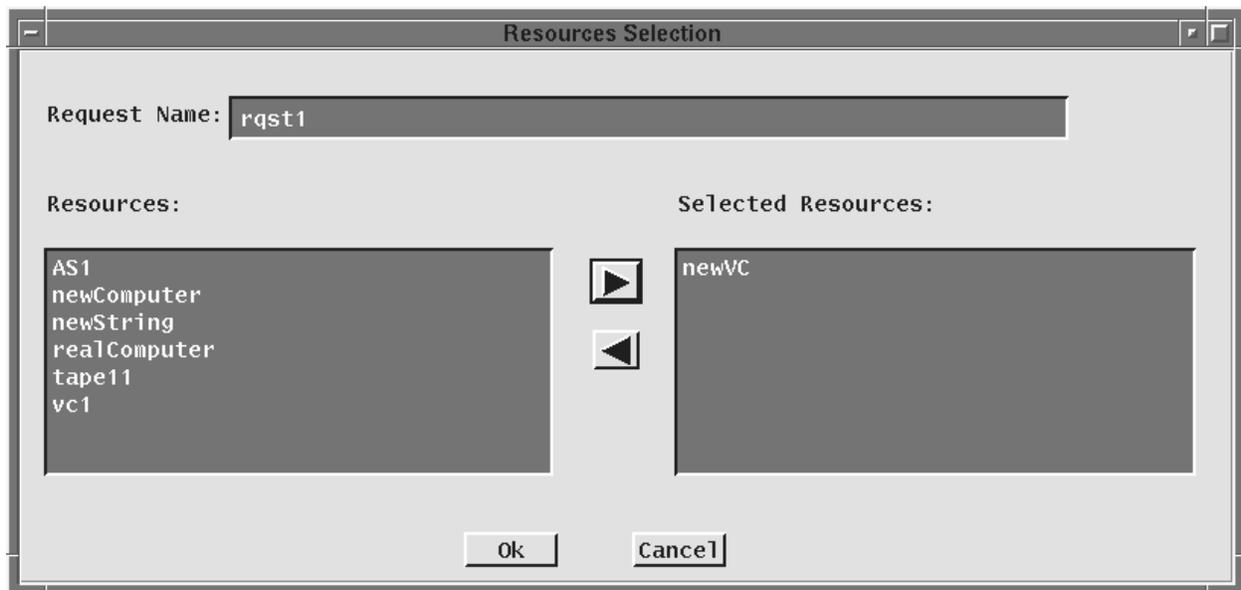


Figure 4.7.1-3. Resources Selection GUI

In the time period specification region of the Resource Reservation Request Edit/Definition GUI, the user can specify the frequency of occurrence of a repeating resource request. Several options for expressing the frequency are available in the ‘**Frequency**’ selection list box combined with a text field that provides a qualifier. The available frequency options are listed in Table 4.7.1-3.

Table 4.7.1-3. Frequency Qualifiers for Resource Reservation Request Edit/Definition GUI (1 of 2)

Frequency Specifier	Text Qualifier	Result
Once	--	The default. Resource reservation covering the period from the start time and stop time for the start date specified.
Monthly	--	Resource reservation for every month on the start day of the month, repeated until the end date as specified.
Daily	--	Resource reservation for every day, between the start date and end date, for the start time and end time specified.
Every M-W-F	--	Resource reservation for every Monday, Wednesday, and Friday, between the start date and end date, for the start time and end time specified.
Every M-through-F	--	Resource reservation for every Monday through Friday, between the start date and end date, for the start time and end time specified.
Every T, Th	--	Resource reservation for every Tuesday & Thursday, between the start date and end date, for the start time and end time specified.

Table 4.7.1-3. Frequency Qualifiers for Resource Reservation Request Edit/Definition GUI (2 of 2)

Frequency Specifier	Text Qualifier	Result
Weekends	--	Resource reservation for every Saturday & Sunday, between the start date and end date, for the start time and end time specified.
'Every_?_days'	N-days	Resource reservation for every n-days, between the start date and end date, for the start time and end time specified.

The **'Intervals ...'** button provides the mechanism to tailor a 'Frequency' based request by overriding selected intervals. If the user selects the 'Intervals ...' button, the Intervals Selection GUI shown in Figure 4.7.1-4 is displayed, providing two lists. The window on the left displays a list of dates selected through the 'Frequency' button. The window on the right shows dates that the operator selects for exclusion from the initial interval. Items are moved from one list to the other by selecting the item and using the arrow keys. The name of the resource request is also displayed.

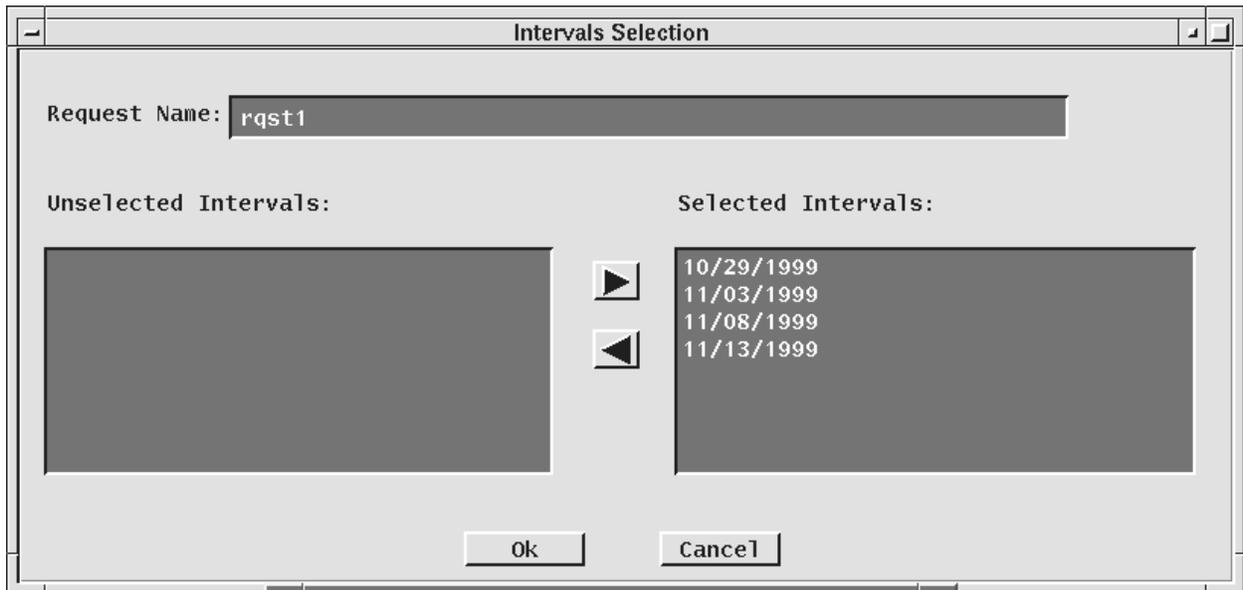


Figure 4.7.1-4. Intervals Selection GUI

4.7.1.2.2 Resource Reservation Planning Master Timeline GUI

From the Resource Planning GUI, the user can select the **'TimeLine'** button to view a timeline format display of all approved resource reservations, similar to the example displayed in Figure 4.7.1-5.

The Resource Reservation Planning Master Timeline GUI represents a set of resources, arranged along the left side of the GUI, and some period of time as indicated across the top edge of the GUI. One or more 'resource reservation' bars represent the use of a resource over a period of time across the GUI. A bar represents a time period during which a resource reservation has

been approved for the resource. Each bar has the name of the resource reservation. At those times where a reservation has not been placed against a resource, that resource is allocated to a default activity (which can vary by resource). For example, science-processing computers are used for science processing unless a reservation has been placed against them. Scroll bars allow scrolling up and down through the full list of resources and left and right in time. A select list of time span viewing options (e.g., 24-hours, 48-hours) at the lower left of the screen is available for selecting the time span of interest.



Figure 4.7.1-5. Resource Reservation Planning Master Timeline GUI

The capabilities to adjust some of the timeline display features (e.g., colors) are available through the pulldown menus located at the top of the GUI. These capabilities include:

- **'File'** Pulldown:
 - **Quit** - Exits this application and returns to the Resource Planning GUI
- **'Display'** Pulldown:
 - **Change Color** - Changes the color setup of the display.
 - **Change Resource** - Selects or filters the resources displayed in the timeline.

- **Change Timescale** - Changes the plan window start and stop time

4.7.1.2.2.1 Report Generator GUI

This functionality is documented in the Production Planning Generator command line interface (see Section 4.8.4).

4.7.1.3 Resource Editor GUI

The resource list can be created from scratch or be initialized from the MSS-managed configuration list, through a 'fetch baseline' operation. Resources can be added to or deleted from the resource list without affecting the MSS configuration list. This is useful, for example, for identifying resources available in the future for the purposes of planning. The Resource Editor GUI, shown in Figure 4.7.1-6, is the tool used for adding resources or modifying the characteristics of existing resources. On start-up, the Resource Editor GUI displays a list of the resources known to the resource-planning tool. To add a new resource the operator first selects a resource type using the **Resource Type** selection button I. Then the operator clicks on the 'New...' button. The characteristics of an existing resource can be modified clicking on the resource in the resource list and clicking on 'Modify...'. A resource can be removed by selecting it and clicking on 'Delete'.

The retrieval of configuration information is a two-step process. First, press the '**Fetch Baseline**' button to generate a file that contains configuration information that is used by Resource Planning. Then press the '**Load Baseline**' button to load this information into the Planning and Data Processing Subsystems (PDPS) database.

The GUIs for the definitions of specific resources are discussed in the following subsections.



Figure 4.7.1-6. Resource Editor GUI

In addition, the 'File' pulldown located on the menu bar provides an 'Exit' option to leave the application.

4.7.1.3.1 Hardware Details GUI

The Hardware Details GUI, shown in Figure 4.7.1-7, is used to define the characteristics of generic hardware resources at the DAAC, such as communications equipment.



Figure 4.7.1-7. Hardware Details GUI

The characteristics of the other fields are described in Table 4.7.1-4.

Table 4.7.1-4. Hardware Details GUI Field Description

Field Name	Data Type	Size (characters)	Entry	Description
Resource Name	ASCII characters	<= 60	User provided, required.	A user defined name for the resource.
Comments	ASCII characters	Unlimited	User provided, optional.	User comment on the resource.

4.7.1.3.2 Disk Details GUI

The Disk Details GUI, shown in Figure 4.7.1-8, is used to define the characteristics of disk partition type resources.

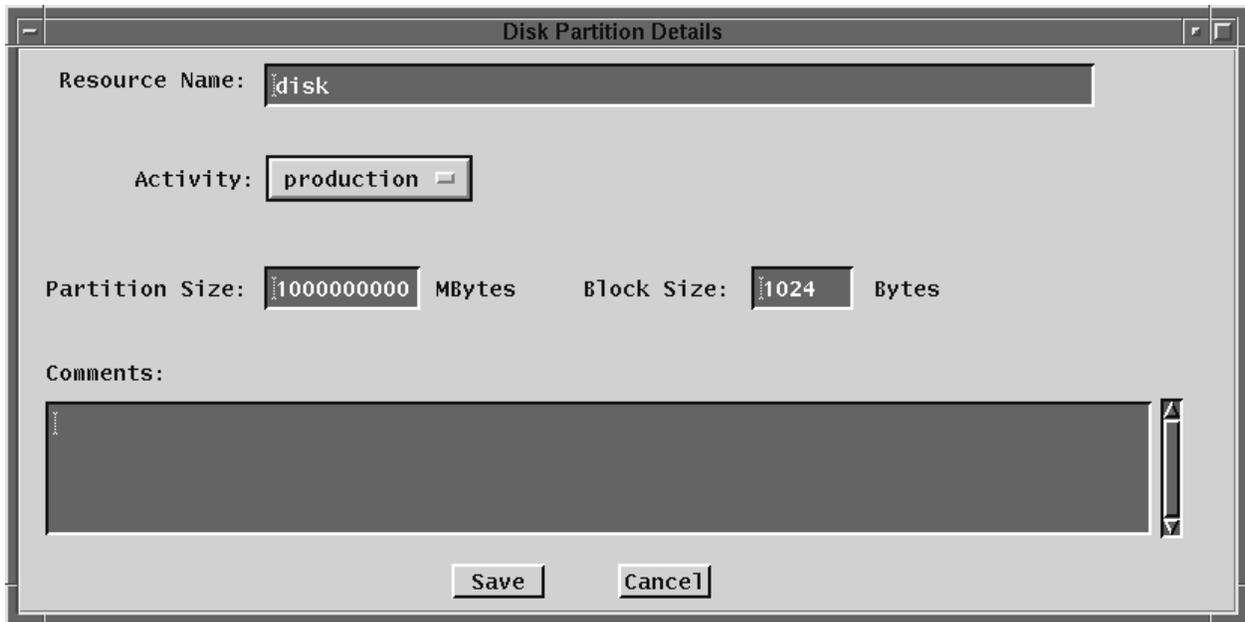


Figure 4.7.1-8. Disk Details GUIs

The characteristics of the other fields are described in Table 4.7.1-5.

Table 4.7.1-5. Disk Resource Details GUI Field Description

Field Name	Data Type	Size (characters)	Entry	Description
Resource Name	ASCII characters	<= 60	User provided, required.	A user defined name for the resource.
Activity	ASCII characters	N/A	User-provided, required.	Activity performed by the resource
Partition Size	Float	<= 10	User provided, required.	The size of the disk partition, in bytes.
Block Size	Integer	<= 6	User provided, required.	Block size used for the disk in bytes.
Comments	ASCII characters	Unlimited	User provided, optional.	User comment on the resource.

4.7.1.3.3 Virtual Computer Details GUI

The Virtual Computer Details GUI shown in Figure 4.7.1-9 is used to define the characteristics of virtual computer resources.

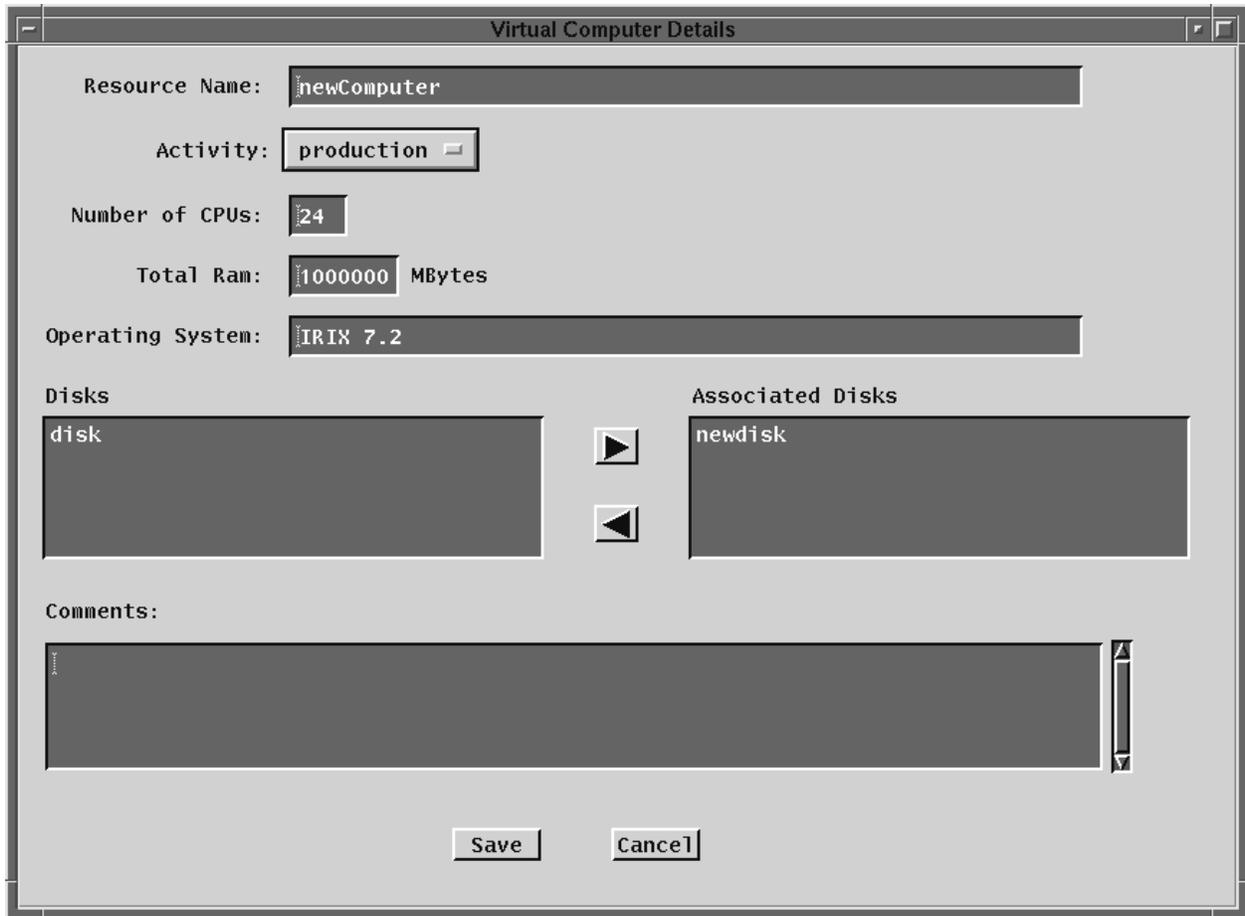


Figure 4.7.1-9. Virtual Computer Details GUIs

The characteristics of the other fields are described in Table 4.7.1-6.

Table 4.7.1-6. Virtual Computer Details GUI Field Description

Field Name	Data Type	Size (characters)	Entry	Description
Resource Name	ASCII characters	<= 60	User provided, required	A user defined name for the computer.
Activity	ASCII characters	N/A	User-provided, required.	Activity performed by the resource
Number of CPUs	Integer	<= 3	User provided, required.	The number of CPUs within the computer.
Total RAM	Integer	<= 7	User provided, required.	The total memory for the computer, in Mbytes.
Operating System	ASCII characters	<= 60	User provided, required.	The operating system name/version for the computer.
Disks	ASCII characters	<= 60	N/A	<ul style="list-style-type: none"> • A list of the disks previously defined for that site • This list of disks is used to select from when a disk is associated (or disassociated) with the computer using the arrow buttons
Associated Disks	ASCII characters	<= 60	User provided, required.	Disks in this list are associated with the computer.
Comments	ASCII characters	Unlimited	User provided, optional.	User comment on the resource.

4.7.1.3.4 String Details GUI

The String Details GUI, shown in Figure 4.7.1-10, is used to define the characteristics of a string, or a collection of virtual computer resources.

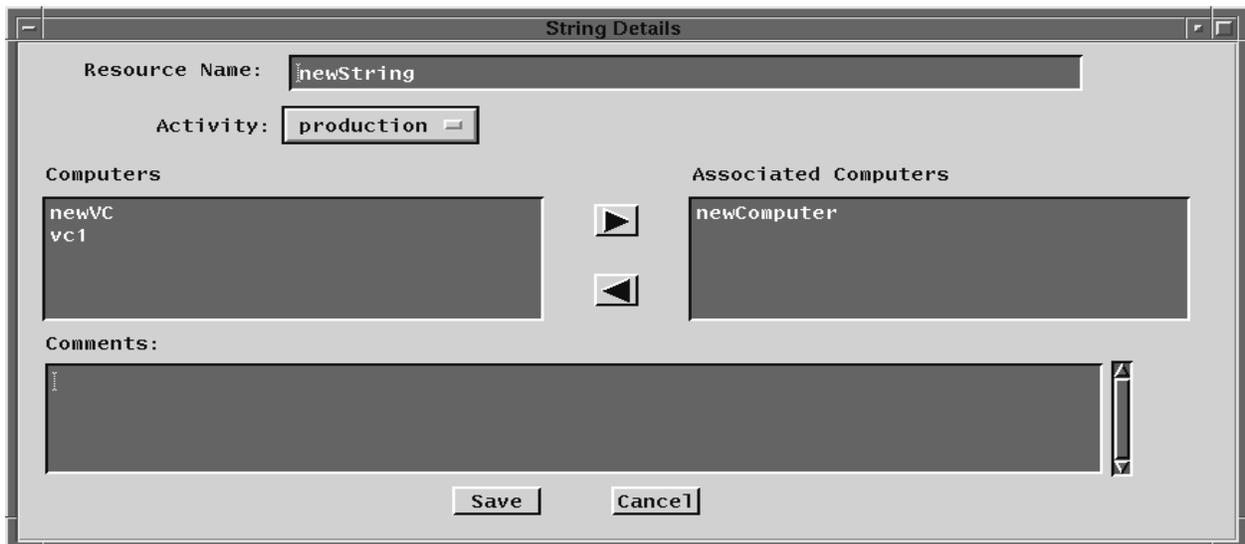


Figure 4.7.1-10. String Details GUIs

The characteristics of the other fields are described in Table 4.7.1-7.

Table 4.7.1-7. String GUI Field Description

Field Name	Data Type	Size (characters)	Entry	Description
Resource Name	ASCII characters	<= 60	User provided, required.	A user defined name for the resource.
Activity	ASCII characters	N/A	User-provided, required.	Activity performed by the resource
Computers	ASCII characters	<= 60	N/A	A list of the computers previously defined for that site. This list of computers is used to select from when a computer is associated (or disassociated) with the string using the arrow buttons.
Associated Computers	ASCII characters	<=60	User provided, required.	Computers in this list are associated with the string.
Comments	ASCII characters	Unlimited	User provided, optional.	User comment on the resource.

Note that the Real Computer Details GUI and the Autosys Details GUI are similar to the String Details GUI. They are shown, respectively, in Figures 4.7.1-11 and 4.7.1-12.

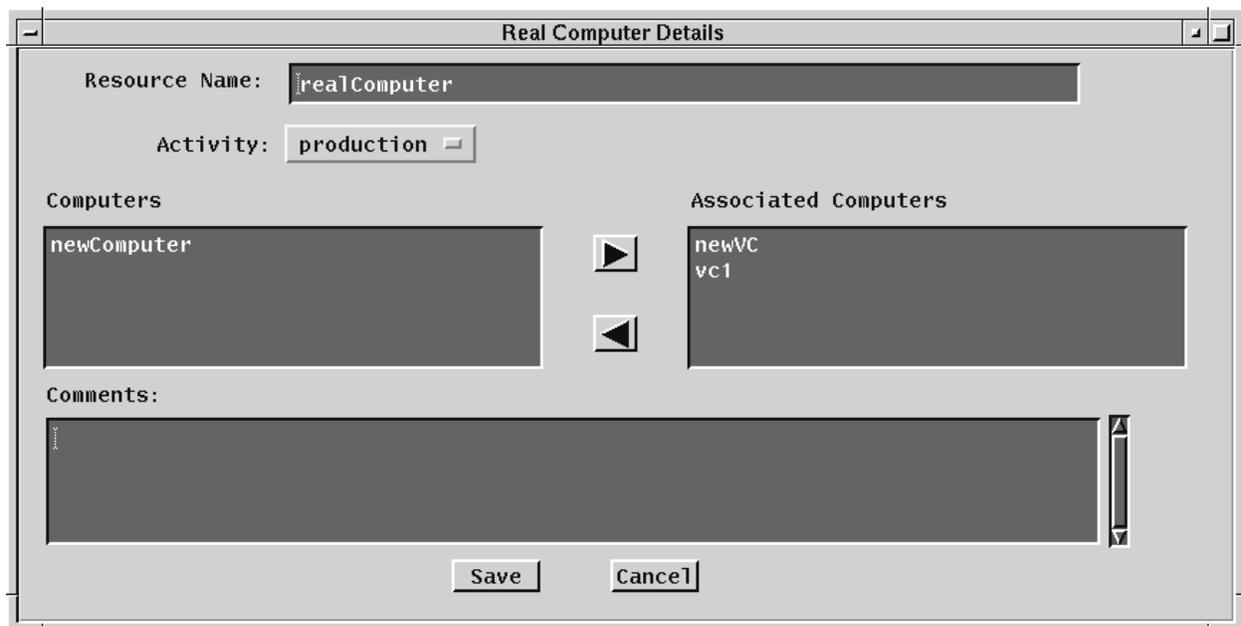


Figure 4.7.1-11. Real Computer Details GUIs

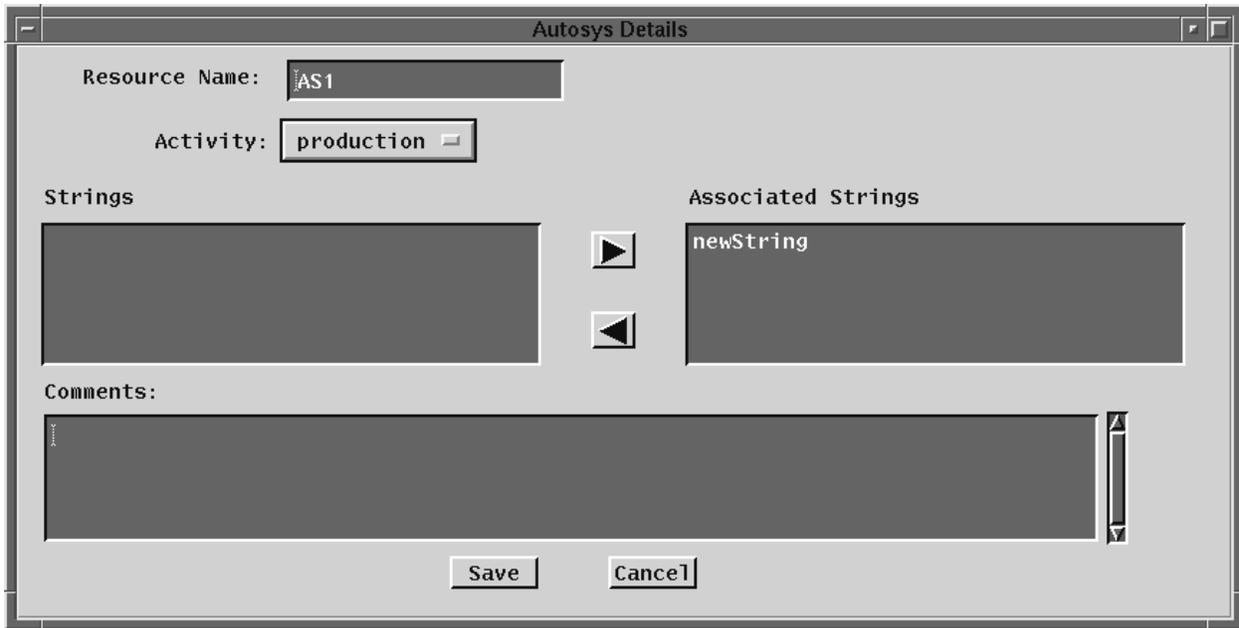


Figure 4.7.1-12. AutoSys Details GUIs

4.7.1.4 Required Operating Environment

For information on the operating environment, tunable parameters, and environment variables refer to the 920-TDA-022 “Custom Code Configuration Parameters” documentation series.

4.7.1.5 Databases

The Resource Planning GUIs relies on the PDPS database to provide all the necessary information at the start time as well as to save all the updated input data about Resource Planning and Resource Definition.

Information on the PDPS database is contained within the Production Planning tools portion of this document, Section 4.8.1.3.

4.7.1.6 Special Constraints

There are no special constraints associated with the Resource Planning tools.

4.7.1.7 Outputs

The Resource Planning tools provide output in three forms:

- Displays to the GUIs (described above)
- Updates to the PDPS database for planning (described in Section 4.8.1.3)
- Generates reports (described in Section 4.7.1.8).

4.7.1.8 Event and Error Messages

The resource planning tools provide informational messages or warnings for minor errors. Significant resource planning events or errors are logged to the MSS Event Log.

Both event and error messages are listed in Appendix A.

4.7.1.9 Reports

There are no reports generated.

4.8 Production Planning

This section describes the Production Planning GUIs used by DAAC operators.

The production planner to define the science processing jobs to be run at a DAAC uses the Production Planning GUIs. The jobs are defined in terms of Production Requests (PRs). A PR is essentially an order for data to be produced by the data processing subsystem. A single PR may lead to several jobs being run over time, or to a single job producing a single set of data. PRs apply to the processing of new data (standard PRs, or standing orders) or the reprocessing of existing data (reprocessing PRs).

The Planning subsystem uses the PR and information on the science processing software (known as a Product Generation Executive, or PGE) to prepare a Data Processing Request (DPR). A DPR corresponds to a single processing job.

Planning subsystem GUIs are used to enter or modify PRs, review DPRs, and produce a selection of production planning related reports. The production planner uses the GUIs to create plans for data processing from the PRs defined in the Production Planning Subsystem. The production planner to activate or cancel a plan also uses the GUIs.

The Production Planning GUIs are packaged into three applications: the Production Request Editor, the Production Planning Workbench, and the Production Strategies User Interface. The Production Request Editor and Production Planning Workbench are accessible through separate icons from the desktop.

This page intentionally left blank.

4.8.1 Production Request Editor

The Production Request Editor allows the user to create production requests, which produce data products. The Production Request Editor is used to perform the functions described in Table 4.8.1-1.

Table 4.8.1-1. Common ECS Operator Functions Performed with Production Request Editor

Operating Function	GUI	Description	When and Why to Use
Generate/Review /Edit Production Requests (PRs)	Production Request Editor -- PR Edit, PR List	<ul style="list-style-type: none"> The operator enters Production Request information Production Requests define processing over a period of time 	To process or reprocess ECS data. When a PR expires, a new one can be input.
Review Data Processing Requests (DPRs)	Production Request Editor – DPR View, DPR List	<ul style="list-style-type: none"> The operator can review the characteristics of a particular data processing request DPRs are created automatically from PRs 	A DPR can be reviewed to inspect job parameters.

4.8.1.1 Quick Start Using Production Request Editor

To execute the Production Request Editor from the command line prompt use:

EcPIPRE_IFStart <mode> where:

<mode> is the ECS mode for the execution (e.g., OPS, TS1 or TS2)

Refer to the 910-TDA-022 “Custom Code Configuration Parameters” documentation series for a listing of EcPIPRE_IFStart.

4.8.1.2 Production Request Editor Main Screen

The Production Request Editor is a collection of display areas in a ‘tab stack’ selection arrangement – clicking on a tab along the top causes the associated tab “page” (also referred to as the tab) to be displayed. There are “pop-ups” associated with some of the tabs, which expand the displayed area and provide GUI interaction for sub-functions. The first tab in the ‘Planning’ window, shown in Figure 4.8.1-1, is the ‘cover page’ or default of the tab stack, which lists and describes the other tabs. The primary activities associated with the Production Request Editor activity are:

- Production Request Edit Tab (PR Edit)
- Production Request List Tab (PR List)

- Data Processing Request View Tab (DPR List)
- Data Processing Request List Tab



Figure 4.8.1-1. Production Request Editor GUI Showing the Planning Tab

In addition, on the menu bar, the pulldown menus provide the following capabilities.

- **File** Pulldown:
 - **Exit** – To exit the application

The **PR List** tab, shown in Figure 4.8.1-2, and **PR Edit** tab, shown in Figure 4.8.1-3, have a lot of commonality. The data displayed on the **PR List** tab is a subset of the data included on the **PR Edit** tab. These data fields are described in Table 4.8.1-2.

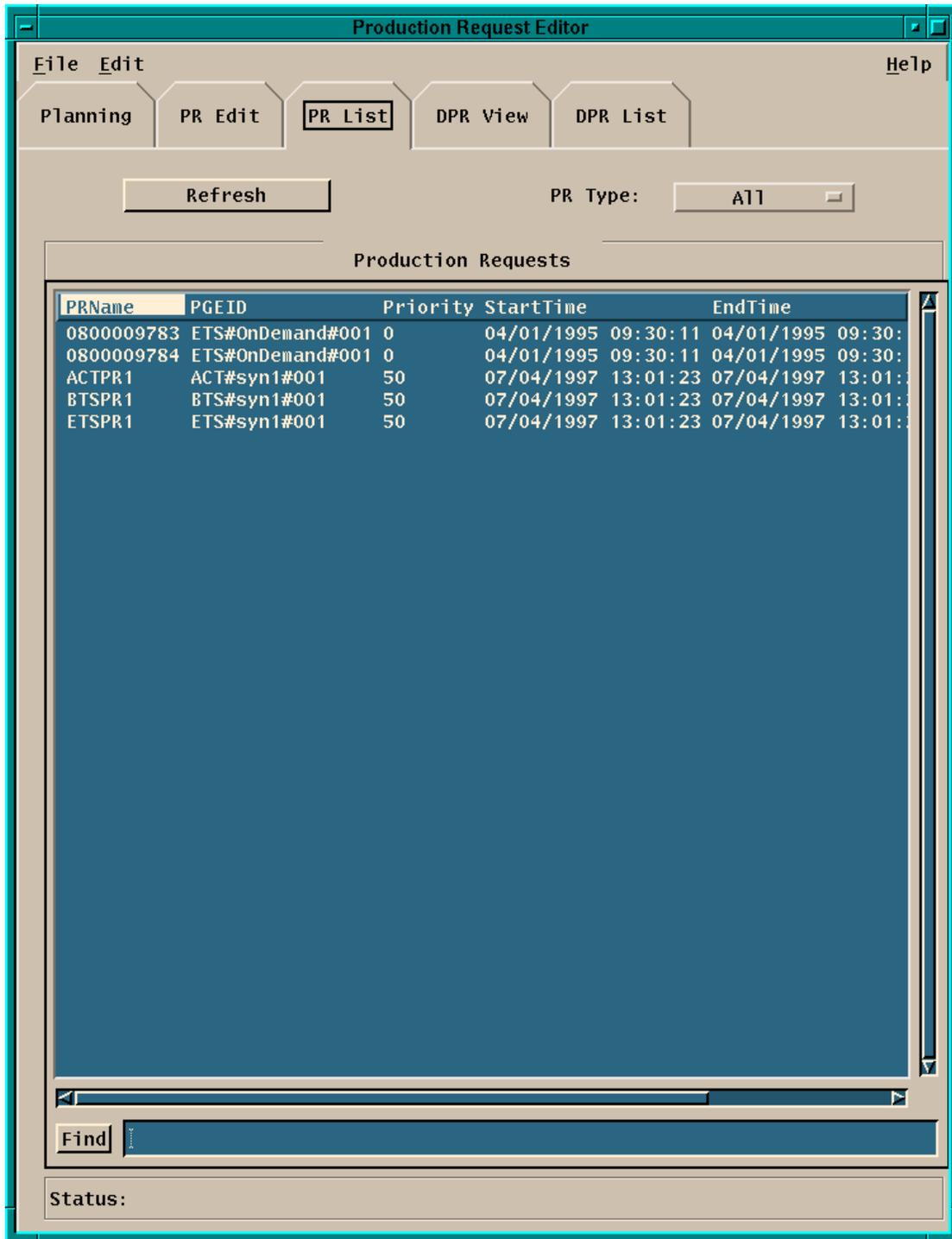


Figure 4.8.1-2. PR List Tab

The **PR List** tab allows the operator to review those PRs already entered into the system. The PRs are presented in the order in which they appear in the PDPS database as a one line summary display for each PR. Entering a search string in the field next to the 'Find' button and then clicking on the button can locate a particular PR. The first occurrence of the search string is highlighted. By clicking on (highlighting) a PR and then selecting the **File: Open** pull-down option, the **PR Edit** tab is initialized with information for that selected PR.

The menu bar for the **PR List** tab and its pull-down menus provide the following capabilities.

- **'File'** Pull-down:
 - **Open** – Allows the operator to open a highlighted, existing Production Request for review or editing in the **PR Edit** tab
 - **Exit** – To exit the application
- **'Edit'** Pull-down:
 - **Delete** – To delete a production request

Clicking on the **Refresh** button will display an up-to-date list of Production Requests in the editor. Selecting an option from the **PR Type** item allows users to choose a specific Production Request type.

The data on the **PR List** tab are identified by the column headings at the top of the display. These data descriptions are given in Table 4.8.1-2.

Production Request Editor

File Edit **Help**

PR Name:
 Origination Date:
 (UTC)

PR Type:
 Originator:

User Type:
 Priority (1 to 10):

Satellite Name:

Instrument Name:

PGE Name:

PGE Version:

Profile Id:

Collection Time
 Insertion Time

Duration
 UTC Time
 Orbit

Collection Time

Begin / / - : :

End / / - : :

From

To

Tile Id

PGE Chain Head
 Yes
 No
 Computer

Intermittent DPR

Skip
 Keep
 SkipFirst

Comment:

Figure 4.8.1-3. PR Edit Tab

On the menu bar for the **PR Edit** tab, the pulldown menus provide the following capabilities.

- **'File'** Pulldown:
 - **New** – When selected, the fields are cleared to allow entry of a new Production Request
 - **Open** – Allows the operator to select an existing Production Request for review or editing. This function uses the File Selection Pop-up (Figure 4.8.1-4) invoked by **Save As**
 - **Save As...** - Allows the operator to save the displayed Production Request and give it a new PR name. This function uses the File Selection Pop-up (Figure 4.8.1-4) invoked by **Open**
 - **Exit** – To exit the application
- **'Edit'** Pulldown:
 - **Delete** – To delete a production request

Most PRs are slight modifications of existing PRs, for example, to change the time duration of a PR. The process for entering a new PR is usually to select an available PR via the **File: Open** feature. The user then would modify the parameters as necessary and save the new PR through the **'Save As...'** option on the **'File'** menu bar of **PR Edit**. This action brings up a separate pop-up to name the new PR. Also, a completely new PR can be entered directly via the **PR Edit** tab.

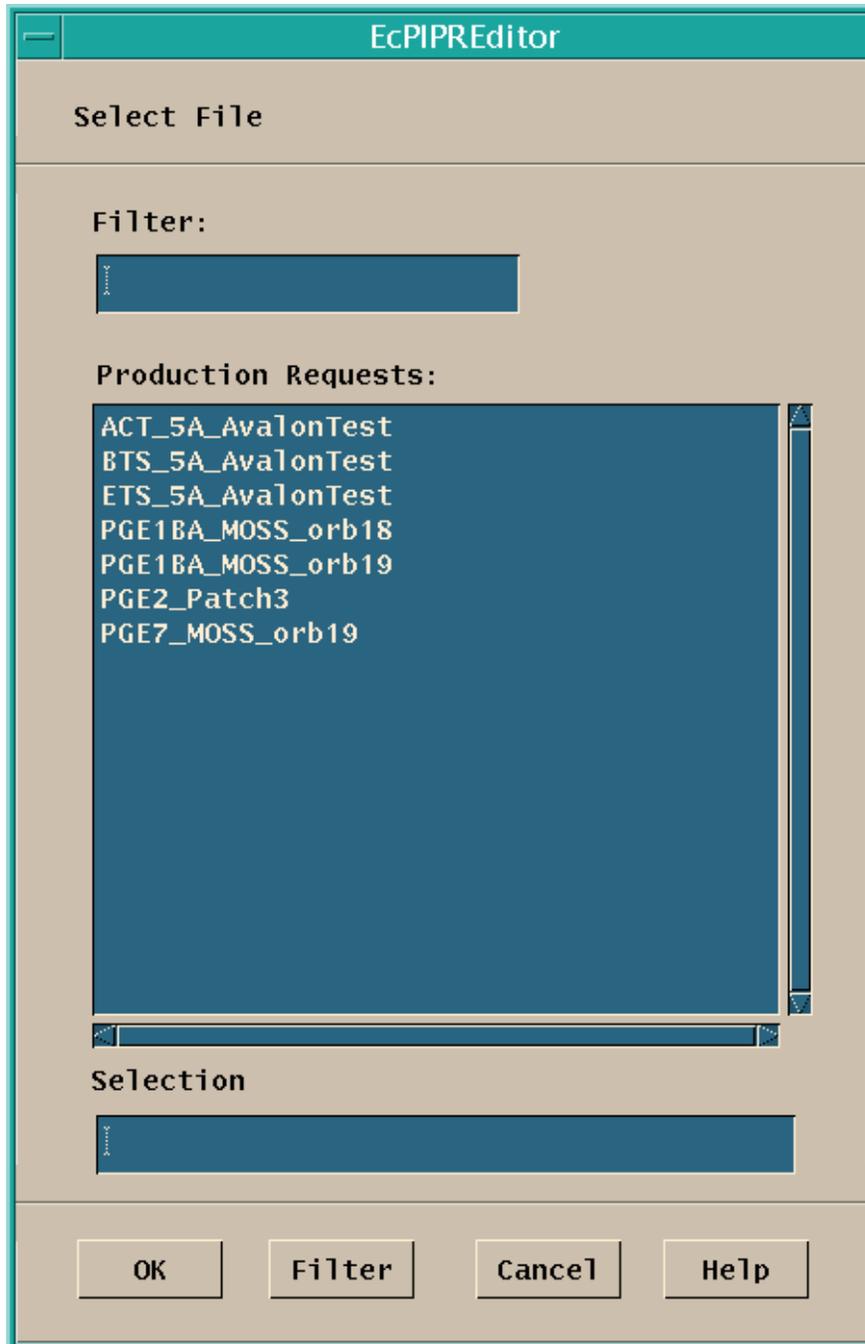


Figure 4.8.1-4. File Selection Pop-up

The **PR Edit** tab fields are organized into six regions: Production Request Identification, Request Definition, Duration, PGE Chain Head / Computer, Intermittent DPR, and Comment. The individual fields of the **PR Edit** tab are described in Table 4.8.1-2.

Table 4.8.1-2. PR Edit Field Description (1 of 2)

Field Name	Data Type	Size	Entry	Description
Production Request Identification	--	--	--	Information used to identify the PR and the originator.
PR Name	ASCII characters	<27	User input, required	A name for the PR.
PR Type	GUI selection	N/A	Click	Routine, On Demand or Reprocessing
User Type	GUI selection	N/A	Click	Operator, DAAC Manager, Scientist or Researcher
Origination Date (UTC)	Date	10	System generated	Date of PR entry.
Originator	ASCII characters	<25	System generated	User ID of the user entering the PR.
Priority (1 to 10)	Integer	<3	User input, required	Priority to be associated with the PR; 1≤value≤99.
Production Request Definition	--	--	--	Information defining the PR.
Satellite Name	ASCII characters	<25	System generated	Satellite name, if applicable, with which the PR/PGE is associated can be displayed when the PGE is selected.
Instrument Name	ASCII characters	<20	System generated	Instrument name, if applicable, with which the PR/PGE is associated can be displayed when the PGE is selected.
PGE Name	ASCII characters	<12	System generated	Name of PGE to be used in the PR can be displayed when the PGE is selected.
PGE Version	ASCII characters	10	System generated	The version number of the PGE to be associated with the PR can be displayed when the PGE is selected.
Profile ID	Integer	<3	System generated	The Profile Id of the PGE to be associated with the PR displayed when the PGE is selected.
Duration	--	--	--	Time range over which the PR is applicable.
Collection Time	Time	<22	System generated	Data start/stop time (in format: MMDDYYYY HH:MM:SS:MS)
Insertion Time	Time	<22	System generated	Time ASTER tape was inserted by the DAAC.
UTC Time	Button	N/A	User input required	UTC radio button, for use when instrument is time based. UTC start/stop date/times are then required.

Table 4.8.1-2. PR Edit Field Description (2 of 2)

Field Name	Data Type	Size	Entry	Description
Orbit	Button	N/A	User input required	Orbit radio button, for use when the instrument is orbit based, such as MISR. The user is then required to enter the orbit number instead of start/stop date/time. After the orbit number is entered, the user can hit the UTC radio button, and the start/stop date/time for the orbit selected appears in the Collection Time Begin/End display.
Begin Date	Date	8	User input, required	Start date of instrument data to be processed.
Begin Time	Time	6	User input, required	Start time of instrument data to be processed.
End Date	Date	8	User input, required	End date of instrument data to be processed.
End Time	Time	6	User input, required	End time of instrument data to be processed.
From (Orbit)	Integer	<8	User input	User can select orbit from number as an alternative to UTC Time.
To (Orbit)	Integer	<8	User input	User can select orbit to number as an alternative to UTC Time.
Tile Id	Integer	<8	User input	User enters the number of the tile requested.
PGE Chain Head	Button	N/A	User input required	Radio button indicating whether the PGE associated with displayed PR is the first in a chained PGE scenario.
Computer	ASCII characters	<60	User input required	User clicks on the down arrow to display a list of computers on which DPRs can run. Selecting a computer populates the field.
Intermittent DPR	--	--	--	Information about the number of intermittent DPRs being skipped or kept.
Skip	Integer	<6	User input	Number of DPRs to skip.
Keep	Integer	<6	User input	Number of DPRs to keep.
SkipFirst	Flag	GUI	User selected	Determines whether the first DPR is skipped or kept.
Comment	ASCII characters	<255	User input, optional	User comment associated with the PR.
Status	ASCII characters	<20	System generated	Current status of the PR.

There are 4 function selection buttons in the Request Definition section of the screen:

1. **PGE Selection button**

As a part of the **PR Edit** process, a PGE needs to be associated with the PR. The PGE Selection pop-up contains a list of the PGEs available for selection and is accessed via the '**PGE...**' button on the **PR Edit** tab. The PGE Selection pop-up is shown in Figure 4.8.1-5. The information describing each PGE is entered through GUIs used in the SSI&T process, described in Section 4.5.1. To associate a PGE to a PR, the desired PGE is highlighted from the list of PGEs, and then selected by clicking the **OK** button on the PGE Selection pop-up. Entering a search string in the field next to the 'Find' button and then clicking on the button can locate a particular PGE. The resulting PGE list matches the search string in any of the displayed fields: PGE Name, PGE Version, etc. Clicking on the **Cancel** button at the bottom of the PGE Selection pop-up returns to the previous window.

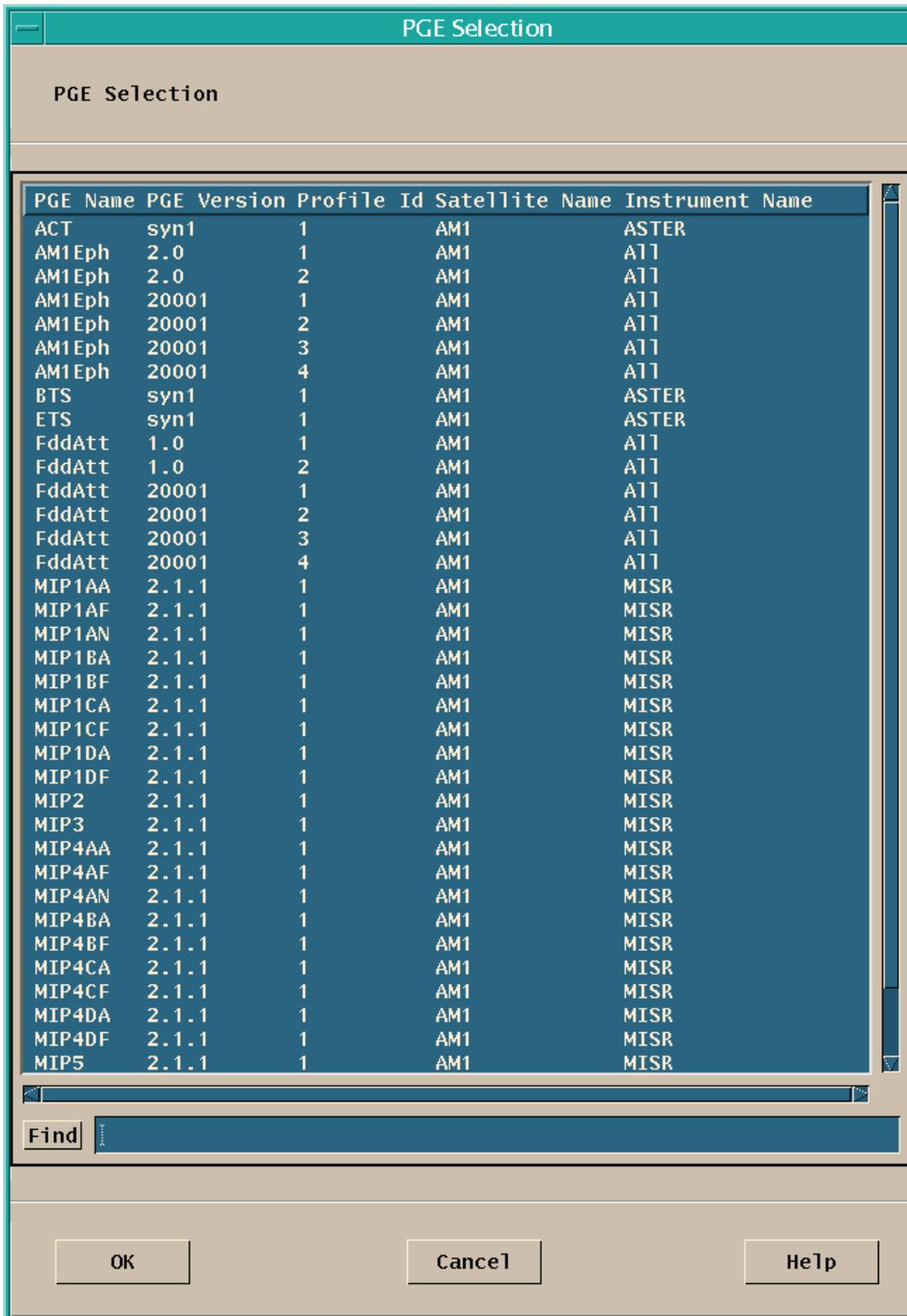


Figure 4.8.1-5. PGE Selection Pop-up

This screen has three function buttons:

- **OK** Complete the action displayed (the selection)
- **Cancel** Ignore the action displayed (the selection)
- **Help** Display a Help pop-up with information about the function of this window.

The data displayed on this screen is described in Table 4.8.1-2.

2. **PGE Parameters button**

The parameters associated with a PGE when it is run are normally specified for the PGE when the PR is defined. If unspecified, the parameters default to values set during the SSI&T process, as described in Section 4.5.1. The production planner can edit or review these parameters by clicking the '**PGE Parameters...**' button on the **PR Edit** tab (Figure 4.8.1-3) once a PGE has been selected. The PGE Parameter Mappings pop-up used to edit/view the parameters are shown in Figure 4.8.1-6. The name of the PGE is shown in the text region at the top of the window. The window lists each parameter on a line with its default value. Entering a search string in the field next to the 'Find' button and then clicking on the button can locate a particular PGE Parameter. The resulting list of PGE Parameters matches the search string in any of the displayed fields: Parameter Name, Logical ID, etc. To override a value, select the desired parameter line, enter the override value in the 'Override Value' text region at the bottom, click **Apply** and then click **OK**.

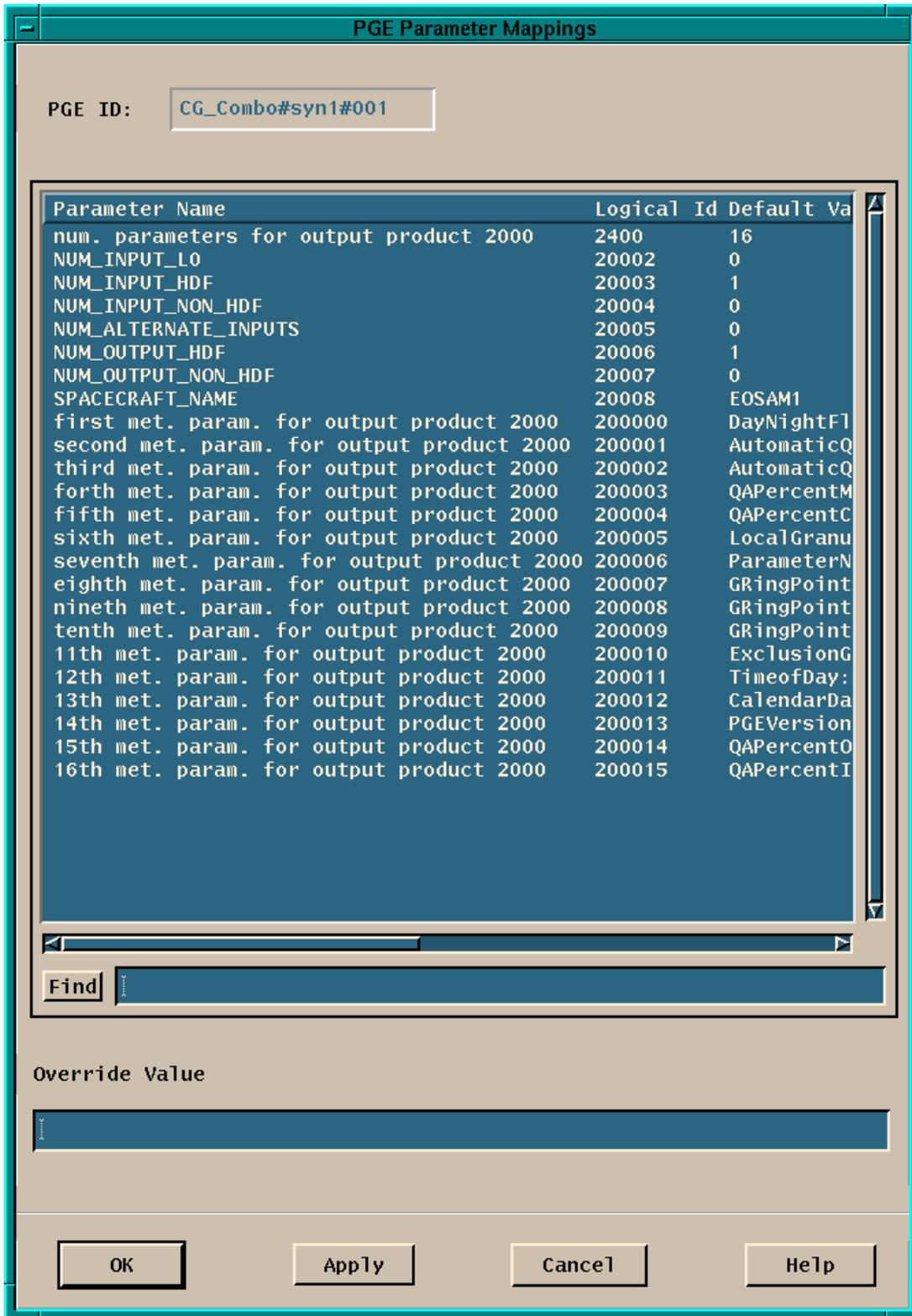


Figure 4.8.1-6. PGE Parameter Mappings Pop-up

This screen has four function buttons:

- **OK** Complete the action displayed (the selection)
- **Apply** Update the database with the value entered
- **Cancel** Ignore the action displayed (the selection)
- **Help** Display a Help pop-up with information about the function of this window.

The data displayed on this screen is described in Table 4.8.1-2.

3. Metadata checks button

The Metadata checks button on the PR Edit tab invokes the MetaDataChecks pop-up window shown in Figure 4.8.1-7. This function allows the operator to check a metadata value against a specified input data granule metadata field value to determine if the data granule can be used or a PGE performed to update the values. The production planner can edit or review these values in the MetaDataChecks pop-up window. This window is invoked by clicking the '**Metadata Checks...**' button on the **PR Edit** tab (Figure 4.8.1-3) once a PGE has been selected. The name of the PGE is shown in the text region at the top of the window. The upper section of the window lists each input data type for the PGE and the lower section displays metadata check values. Entering a search string in the field next to the 'Find' button in the upper section of the window and then clicking on the button can locate a particular input data type. Selecting a particular input data type displays any metadata field values associated with the selected input data type in the lower section of the window. A particular metadata field can be located in the lower section of the window by entering a search string in the field next to the '**Find**' button in the lower section of the window and then clicking on the button. To modify a value, select the desired metadata checks line in the lower section of the window. The metadata field name is displayed in the 'MetaDataField' text region and the current metadata value is displayed in the 'Value' text region at the bottom. Enter the modified value in the 'Value' text region, click **Apply** and then click **OK**.

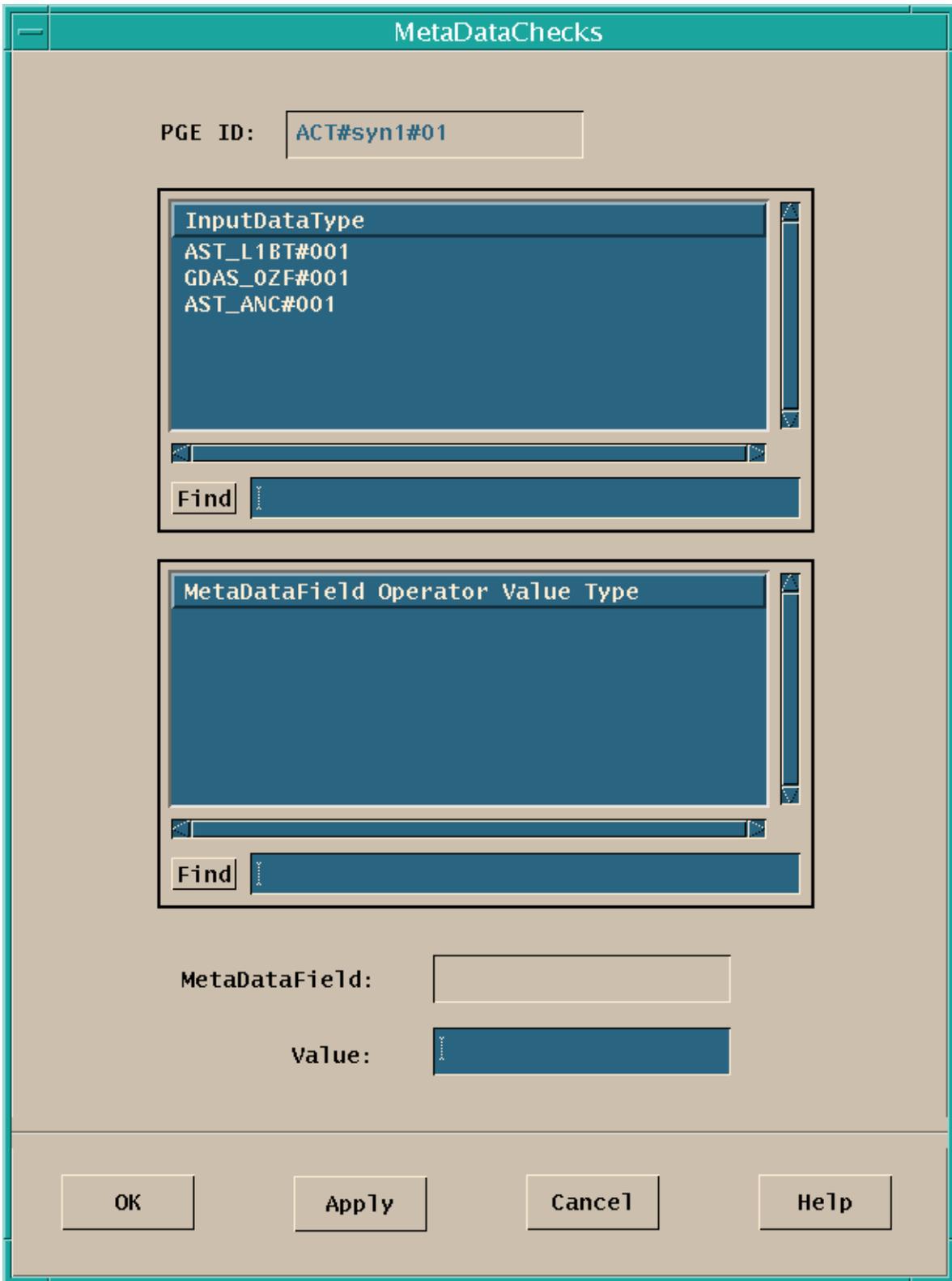


Figure 4.8.1-7. MetaDataChecks Pop-up

This screen has four function buttons:

- **OK** Complete the action displayed (the selection)
- **Apply** Update the database with the value entered
- **Cancel** Ignore the action displayed (the selection)
- **Help** Display a Help pop-up with information about the function of this window

Table 4.8.1-3 describes the fields displayed on the PR Edit-MetaDataChecks pop-up window.

Table 4.8.1-3. PR Edit-MetaDataChecks Field Description

Field Name	Data Type	Size	Entry	Description
PGE ID	ASCII characters	<17	System generated	ID of the associated PGE.
Input Data Type	ASCII characters	<20	System generated	Identifiers for the data types used as input for the selected PGE.
MetaData Field	ASCII characters	<40	System generated	Name of the metadata field.
Operator	ASCII characters	<3	System generated	The logical operator (<, >, =, <=, =>) used with the metadata field.
Value	ASCII characters	<80	System generated	The value stored in the selected metadata field.
Type	ASCII characters	<5	System generated	Metadata field characteristic (Float, Int, String).

4. Alternate Input Values button

The Alternate Input Values button on the PR Edit tab causes the AlternateInputValues pop-up shown in Figure 4.8.1-8 to appear. This function allows the operator to select a replacement for a data input to a PGE with multiple inputs. This can be necessary if inputs to the PGE are not available and alternative inputs can be used.

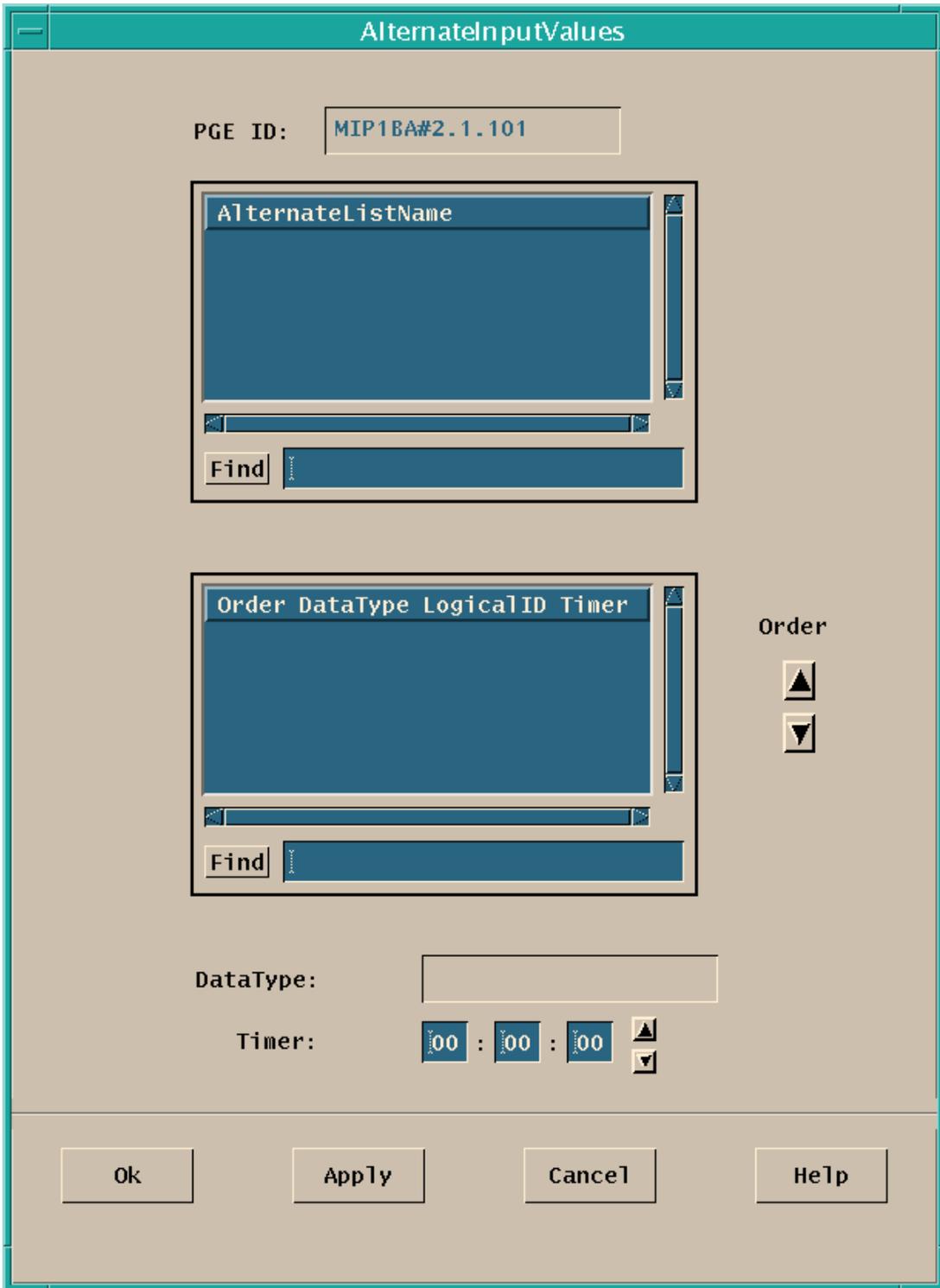


Figure 4.8.1-8. AlternatInputValues Pop-up

This screen has four function buttons:

- **Ok** Complete the action displayed (the selection)
- **Apply** Update the database with the value entered
- **Cancel** Ignore the action displayed (the selection)
- **Help** Display a Help pop-up with information about the function of this window

Table 4.8.1-4 describes the information displayed on the AlternateInputValues pop-up.

Table 4.8.1-4. PR Edit-AlternateInputValues Field Description

Field Name	Data Type	Size	Entry	Description
PGE ID	ASCII characters	<17	System generated	ID of the associated PGE.
Alternate List Name	ASCII characters	<20	System generated	Identity of the primary data type for this input.
Order	Integer	<99	System generated/ User changeable	Current order of this data type.
Data Type	ASCII characters	<20	System generated	Identity of the data type.
Logical ID	Integer	<8	System generated	The SDP Toolkit logical identifier used to reference the data type.
Timer	Long Integer	<12	System generated/ User changeable	The time period the Subscription Manager waits for an alternate input to arrive.

4.8.1.2.1 DPR View and DPR List Tabs

The **DPR List** tab, shown in Figure 4.8.1-9, and the **DPR View** tab, shown in Figure 4.8.1-10 have a lot of commonality. The data displayed on the **DPR List** tab is a subset of that included on **DPR View** tab. Also, the dropdown menus of both tabs have the same options.

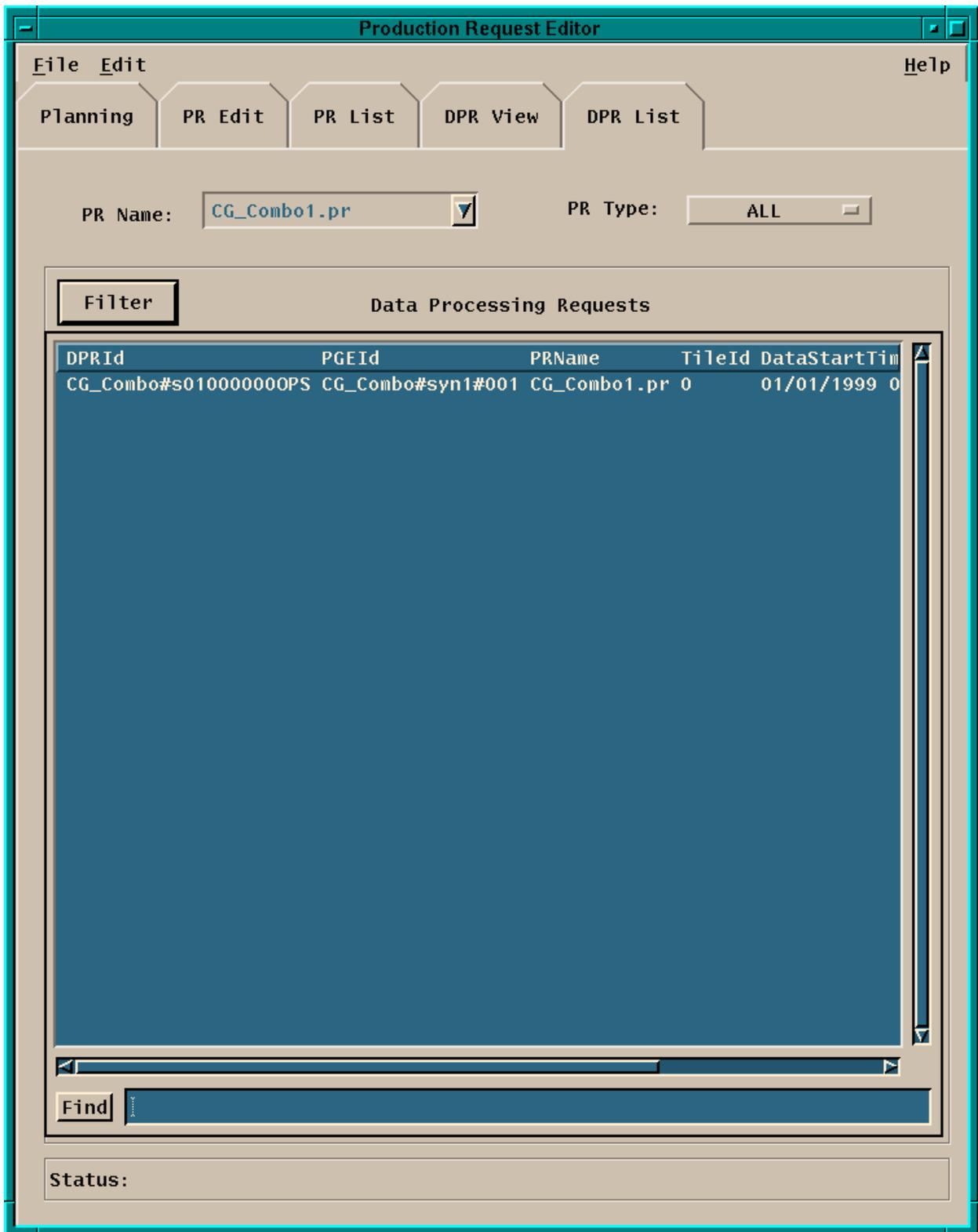


Figure 4.8.1-9. DPR List Tab

DPRs are generated automatically from the PRs described above during Planning. The Production Request Editor GUI provides capabilities to review these DPRs. By selecting the '**DPR List**' tab, a list of DPRs is displayed (in the order in which they have been entered into the PDPS database) for review in the same manner in which PRs are selected and edited. In addition, on the menu bar, the pull-down menus provide the following capabilities.

- '**File**' Pulldown:
 - **Open** - Allows the operator to select an existing DPR for review in the DPR View tab. In the **DPR View** tab (but not in the **DPR List** tab), this function uses the File Selection Popup (Figure 4.8.1-4)
 - **Exit** - To exit the application
- '**Edit**' Pulldown:
 - **Delete** - To delete a DPR

Each line of the **DPR List** display represents a DPR, i.e., a job run when all data and resource needs are satisfied. Entering in a search string in the field next to the 'Find' button and then clicking on the button can search for a particular DPR. The resulting DPR list can match the search string in any of the displayed fields: DPR ID, PGE ID, etc. In addition, DPRs can be filtered for associated PRs by selecting a PR using the Production Request selection tool at the top of the window and clicking on the '**Filter**' button. By selecting (clicking on) one of the DPR summary lines, selecting 'Open' from the 'File' pull-down, and selecting the '**DPR View**' tab, that DPR is displayed in detail.

Production Request Editor

File Edit Help

Planning PR Edit PR List DPR View DPR List

All Times In UTC

Data Processing Request Identification

DPR Name: ACT#syn1#004130123TS1 PR Name: ACT_5A_AvalonTest

Origination Date: 08/16/1999 13:58:31

Originator:

PGE ID: ACT#syn1#01 PGE Parameters...

Data Start Time: 07/04/1997 13:01:23 PGE File Mappings...

Data Stop Time: 07/04/1997 13:01:24

Request Data and Status

Predicted Start

Time: 08/16/1999 13:58:47 Priority: 250

Actual Start

Time: 08/16/1999 13:58:47 Status: SUCC_DEL

Figure 4.8.1-10. DPR View Tab

This screen has two function buttons:

- **PGE Parameters** See Figure 4.8.1-5
- **PGE File Mappings** See Figure 4.8.1-11

The **DPR View** tab fields are organized into three regions: Data Processing Request Identification, PGE Information, and Request Data and Status. The individual fields of the **DPR View** tab are described in Table 4.8.1-5.

Table 4.8.1-5. DPR View Field Description

Field Name	Data Type	Size	Entry	Description
Data Processing Request Identification	--	--	--	Information used to identify the DPR and the originator.
DPR Name	ASCII characters	<24	System generated	DPR name generated from the associated PGE.
PR Name	ASCII characters	<27	System generated	A name for the associated PR.
Origination Date	Date	8	System generated	Date of PR entry.
Originator	ASCII characters	<25	System generated	User ID of the user entering the PR.
PGE Information	--	--	--	Information describing the PGE.
PGE ID	ASCII characters	<17	System generated	ID of the associated PGE.
Data Start Time	Date & time	17	System generated	Start date and time of the data to be processed by the job.
Data Stop Time	Date & time	17	System generated	Stop date and time of the data to be processed by the job.
Request Data and Status	--	--	--	Information describing the running status of the job.
Predicted Start Time	Date & time	17	System generated	Date and time at which the associated job is expected to be run, as predicted by the planning subsystem.
Actual Start Time	Date & time	17	System generated	Date and time at which the associated job ran.
Priority	Integer	3	System generated	Priority of the job; 100≤; ≥1.
Status	ASCII characters	<20	System generated	Status of the job.

DPRs are generated automatically from PRs during the Planning process, using PGE information defined during SSI&T. The PGE information can be viewed but not edited from this screen. This is because the information is quite complex and its alteration can have considerable side effects on the PGE execution.

PGE parameters can be viewed for a particular DPR run. Selecting the 'PGE Parameters' button on the DPR View can access these parameters. This screen is similar to Figure 4.8.1-6 shown earlier.

The input and output files for a particular DPR can be viewed by selecting the '**PGE File Mappings...**' button. The associated GUI is shown in Figure 4.8.1-11. The PGE File Mappings GUI displays one line of information for each file used by or being produced by the PGE. A particular input or output file can be searched for by entering in a search string in the field next to the '**Find**' button and then clicking on the button. The resulting file list can match the search string in any of the displayed fields. Information displayed includes:

- **Logical ID** The ID or tag used within the PGE to access the file
- **Granule ID** The ID or tag used to identify the file uniquely
- **Start/Stop Time** The start or stop date and time for the data contained in this file

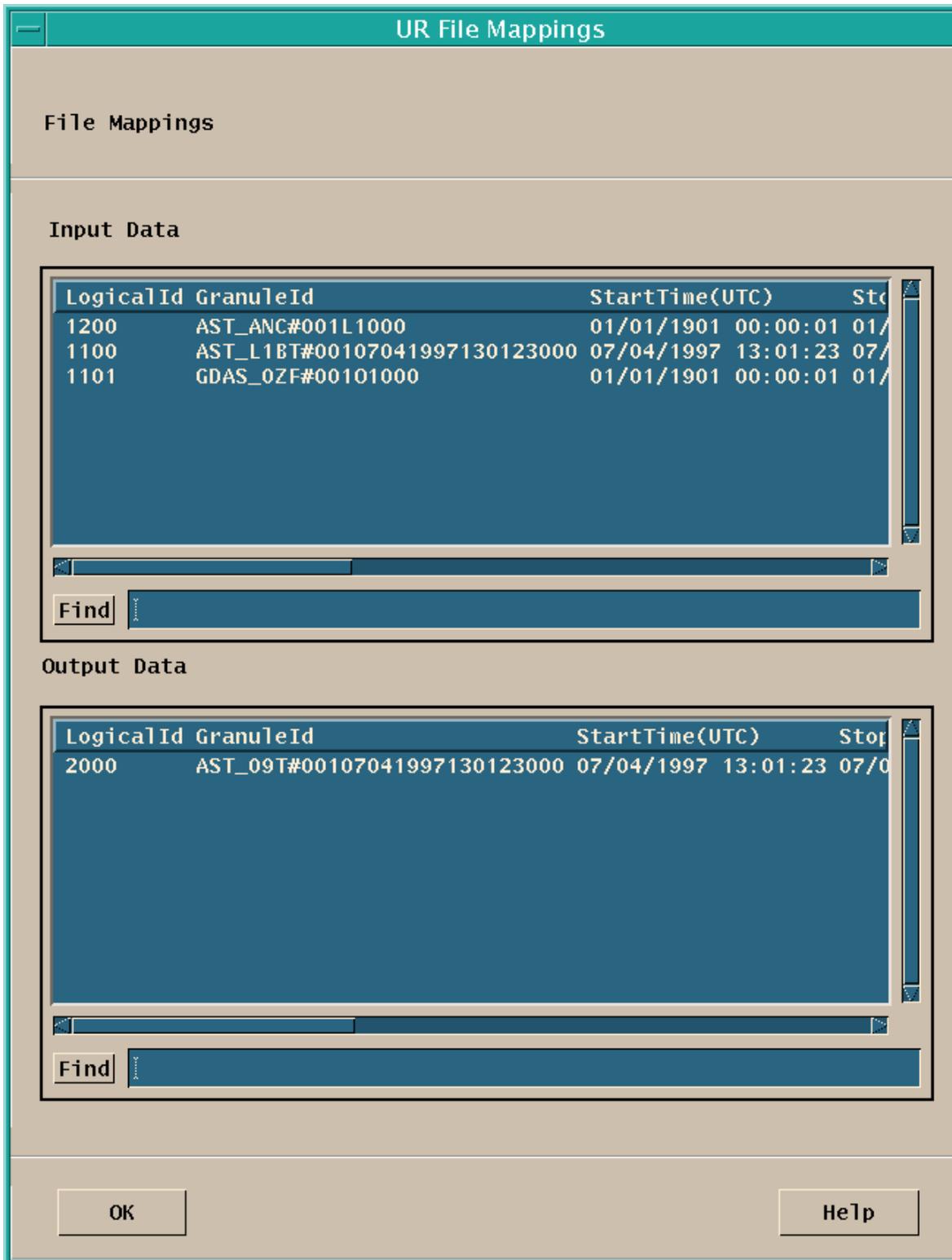


Figure 4.8.1-11. File Mappings Pop-up

This screen has two function buttons:

- **OK** Dismiss the screen
- **Help** Display a Help pop-up with information about the function of this window

The individual fields of the **File Mappings** GUI are described in Table 4.8.1-6.

Table 4.8.1-6. File Mappings Field Description

Field Name	Data Type	Size	Entry	Description
Logical ID	Integer	<8	System generated	The SDP Toolkit logical identifier used to reference the data type.
Granule ID	ASCII characters	<20	System generated	Identity of the data type.
Start Time	Time	17	System generated	Start date/time of instrument data.
Stop Time	Time	17	System generated	End date/time of instrument data.

4.8.1.3 Required Operating Environment

These GUIs can be hosted on the Planning Server and Queuing Server or Planning Workstation.

For information on the operating environment, tunable parameters and environment variables of Production Strategies User Interface refer to the 910-TDA-022 “Custom Code Configuration Parameters” documentation series. The following table (Table 4.8.1-7) identifies the supporting products this tool depends upon in order to function properly.

Table 4.8.1-7. Support Products for Production Request Editor

Product
Sun Solaris 8
MOTIF 1.2.3
PDPS Database

4.8.1.3.1 Interfaces and Data Types

The Planning Subsystem has no interfaces external to ECS. It interfaces with the Data Processing Subsystem through the Planning and Data Processing System (PDPS) database to exchange planning information. The PDPS database is populated with information from the SSI&T components after the integration of a new or updated PGE. An element of the Planning system, which does not require an operator interface, the Subscription Manager, also interfaces with the ECS Science Data Server Subsystem.

The Planning Subsystem is primarily intended for use by the operations staff of the EOSDIS DAACs. However, the capability has been requested and made available for science personnel from the SCFs to enter Production Requests into the Planning subsystem using the same

interface as described above. This is to be done with the consent of the DAAC operations managers and via the X-11 interface with procedures and equipment provided to ensure a secure interface for these updates.

4.8.1.4 Databases

The Planning Subsystem includes the PDPS database, which contains information needed to plan the processing at a DAAC. It is also used for resource planning, containing information on the configured resources and their allocation. Finally, the Data Processing Subsystem, the other major element of the PDPS, uses the database. The following subsections provide a summary description of the PDPS database tables, and tools used with the database.

The SSI&T capability, which is a design component within the Data Processing Subsystem, includes capabilities to support the integration and test of the science software, PGEs, within the production-processing environment. As a part of this activity, the SSI&T GUI, described in Section 4.5.1 of this document, provides for the entry into the databases used by SSI&T of PGE-related information needed to plan for and run science software. At the conclusion of the SSI&T process, this database information is transferred to the operational databases from the databases established for SSI&T.

The Production Request Editor allows changes to scheduling information for PRs and DPRs. The complete database schema is listed in 311-CD-623, *Planning and Data Processing Subsystem Database Design and Schema Specifications*.

4.8.1.5 Special Constraints

There are no special constraints associated with the Production Request Editor.

4.8.1.6 Outputs

Outputs of the Production Request Editor are provided in one of two ways:

1. Production Request Editor GUI responses as described above, including the File Mapping GUI.
2. Updates to the PDPS database (described in Section 4.8.1.3).

4.8.1.7 Event and Error Messages

The Production Planning GUIs provide informational messages or warnings for minor errors, which the operator can immediately correct directly via the operator interface. Significant production planning events or errors are logged to the ECS Planning Server ALOG file. The Error messages are listed in Appendix A.

4.8.1.8 Reports

None.

This page intentionally left blank.

4.8.2 Production Planning Workbench

The second major element of production planning is the Production Planning Workbench (PWB). Table 4.8.2-1 summarizes the operational functionality of this tool. Via the PWB GUI, the operator selects a Production Request (PR) to be included in the plan. The planning tool provides a forecast of the start and completion times of the jobs based upon historical experience in running these PGEs. Using the planning tool, the operator can “activate” the plan, i.e., transfer the information included in the plan to the Data Processing subsystem that loads it into the AutoSys tool where production processing is managed.

Table 4.8.2-1. Common ECS Operator Functions Performed with the Production Planning Workbench

Operating Function	Description	When and Why to Use
Create a Plan	The operator can select available PRs to be included in the plan.	When a new or revised plan needs to be prepared.
Activate a Plan	The operator can activate a plan, which transfers the planned jobs into the Data Processing System.	When a new plan is prepared and accepted.
Save a Candidate Plan	The operator can create What-if scenarios.	When plans are in the process of being created.

4.8.2.1 Quick Start Using Production Planning Workbench

To execute Production Planning Workbench from the command line prompt, enter:

>EcPIAllStart <mode> <Application ID>

Where:

<mode> is the ECS mode under which the program is to run (e.g., OPS, TS1 or TS2)

<Application ID> is an integer from 1 to 5 uniquely identifying the execution

This action results in the display of the Production Planning Workbench main window and the Planning Master Timeline GUI discussed in the following sections.

4.8.2.2 Production Planning Workbench Main Screen

When the Production Planning Workbench is started, the GUI window shown in Figure 4.8.2-1 is one of the windows displayed. Table 4.8.2-1 contains a description of the fields in this GUI. The GUI is started using information from the currently activated plan. Other plans can be opened or new plans created and saved using the ‘New’, ‘Open’, ‘Save’, and ‘Save As’ options on the ‘File’ pull down on the menu bar, much like creating, opening, and saving a text document.

Comments to the plan can be made in the indicated region. Selected PRs can be moved from one list to the other using the arrow buttons. A PR is scheduled by selecting from the pool of

available 'Unscheduled' PRs and, using the arrow buttons, the selected PRs are moved to the 'Scheduled' list. A note of clarification about Ground Events during scheduling: All Ground Events are automatically scheduled with any plan. Hence the GUI Ground Events are always being put into the 'Scheduled' list. Whenever a plan is activated, the Ground Events are also being activated. (If there is an unscheduled Ground Event in the 'Unscheduled' list, this means that this Ground Event has lost allocations.) The new plan can then be saved with a new name through the 'Save As' option on the 'File' pull-down menu.

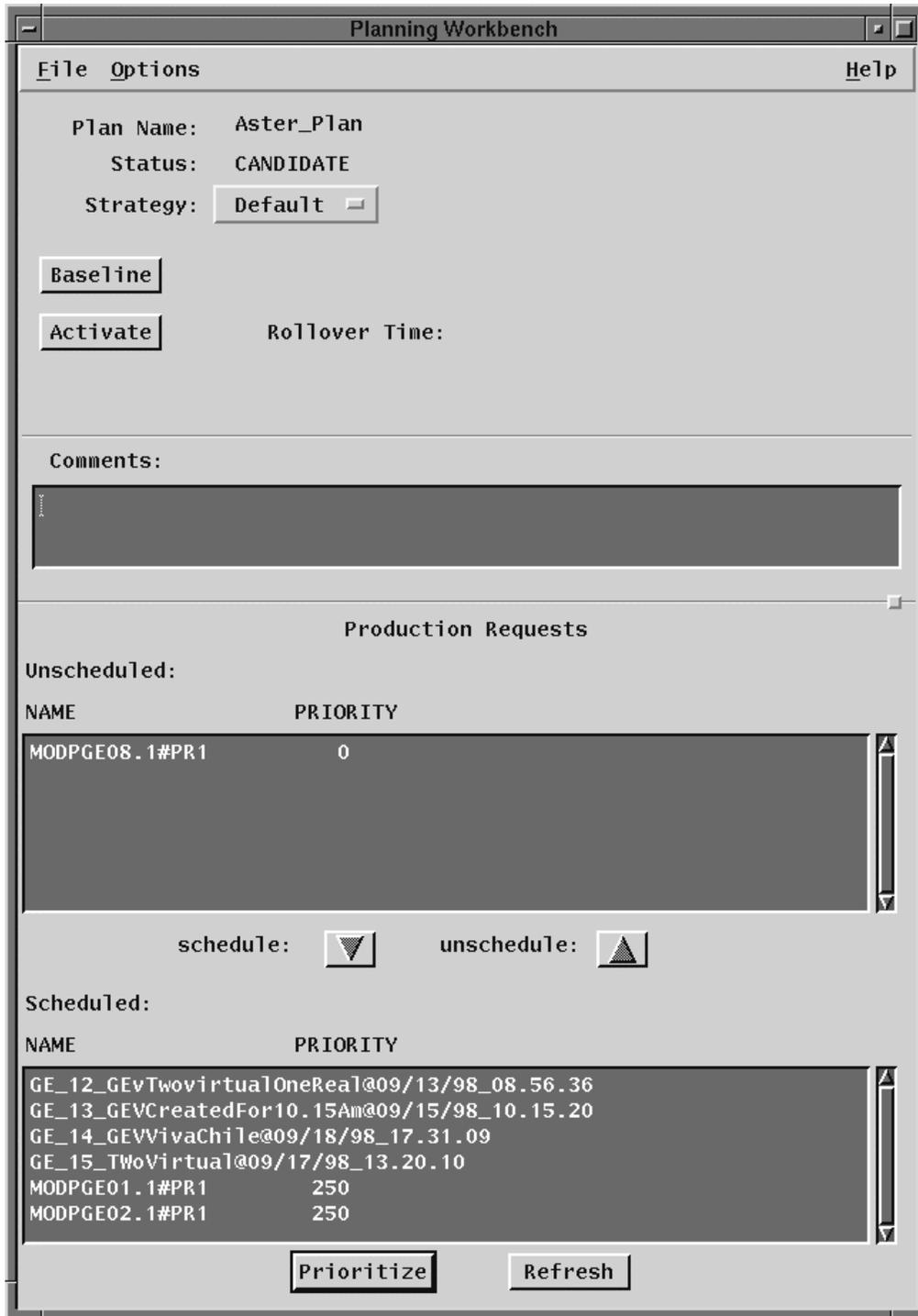


Figure 4.8.2-1. Production Planning Workbench GUI

The following describes the pulldown menu options supported by the PWB GUI:

File

New clears all the fields, and creates a new plan. The new plan name cannot be longer than 20 characters

Open allows the user to select an existing Production Plan for review or editing. This function invokes a File Selection Pop-up for specifying the file to be opened

Save saves the current plan

Save As saves the displayed Production Plan under a new plan name. This function uses the File Selection Pop-up for specifying the new name for the Production Plan.

Delete allows the user to delete a plan via a file selection pop-up. The active plan cannot be deleted but other candidate plans can be removed

Exit exits the application

Options (no functionality has been associated with this menu, as yet)

Help – Provides assistance in using the PWB GUI

The following push buttons are supported on the PWB GUI:

Baseline records the plan and the time the record was baselined. This baseline plan can be used as a point of comparison with which to compare future plans and results

Activate activates a plan and the Data Processing Requests (DPRs) associated with the planned PRs are then transferred to the Data Processing subsystem and loaded into the AutoSys production queuing system. Once the necessary data dependencies are satisfied, the jobs to be run are 'released' when processing resources are available. When the Activate button is pressed on the planning workbench, the current active plan gets "replanned over" by the selected plan. A pop-up listing any unprocessed DPRs that are in the "old" active plan and not selected in the "new" plan is displayed and the user is asked to confirm that these DPRs are to be cancelled

Schedule schedules the selected PRs in the Unscheduled List

Unschedule un-schedules the selected PRs in the Schedule List

Prioritize allows for the modification of priorities associated with PRs. To change the priority for a PR, select (highlight) the PR and press the Prioritize button. A Priority pop-up is displayed. Enter a new priority in the text region for the selected PR and press OK

Refresh re-acquires information from the PDPS database.

Also a pulldown list is available on the PWB GUI:

Strategy: selects different strategies for scheduling, which determine the priorities assigned to the individual PRs in a plan

For all of these actions, the operator is prompted for confirmation before proceeding. Table 4.8.2-2 summarizes the information fields on the PWB GUI.

Table 4.8.2-2. Production Planning Workbench Field Description

Field Name	Data Type	Size	Entry	Description
Plan Name	ASCII characters	<20	System generated	Name assigned to the plan; assigned initially when the 'New' plan is saved.
Status	ASCII characters	<20	System generated	Status of the displayed plan: 'Active', 'Candidate.'
Rollover Time	ASCII characters	19	System generated	The time at which the currently selected plan was activated.
Comments	ASCII characters	<256	User input, optional	User comments.
Production Requests: Unscheduled	ASCII characters	<256	System generated	List of available PRs, which are currently not scheduled.
Production Requests: Scheduled	ASCII characters	<256	User input	List of available PRs, which are currently scheduled. Planner selects these from 'Unscheduled' list and moves them to/ from 'Scheduled' with arrow buttons.

4.8.2.2.1 Planning Master Timeline GUI

Figure 4.8.2-2 is the Planning Master Timeline GUI main screen, which is brought up when the Production Planning Workbench is started as explained in Section 4.8.2.1.

The **Planning Master Timeline GUI** represents a set of computers, arranged along the left side of the GUI, and some period of time as indicated across the top edge of the GUI. Several bars across the GUI for that computer represent the execution of Data Processing Requests (DPRs) on a computer over a period of time. A bar represents a time period during which a PGE (as described by a DPR) is running. Each bar has the name of the PGE. Scroll bars allow scrolling up and down through the full list of computers and left and right in time. A select list of time span viewing options (e.g., 24-hours, 48-hours) at the lower left of the screen is available for selecting the time span of interest. If one exits from the **Planning Master Timeline GUI**, it can be restarted as explained in Section 4.8.2.1.

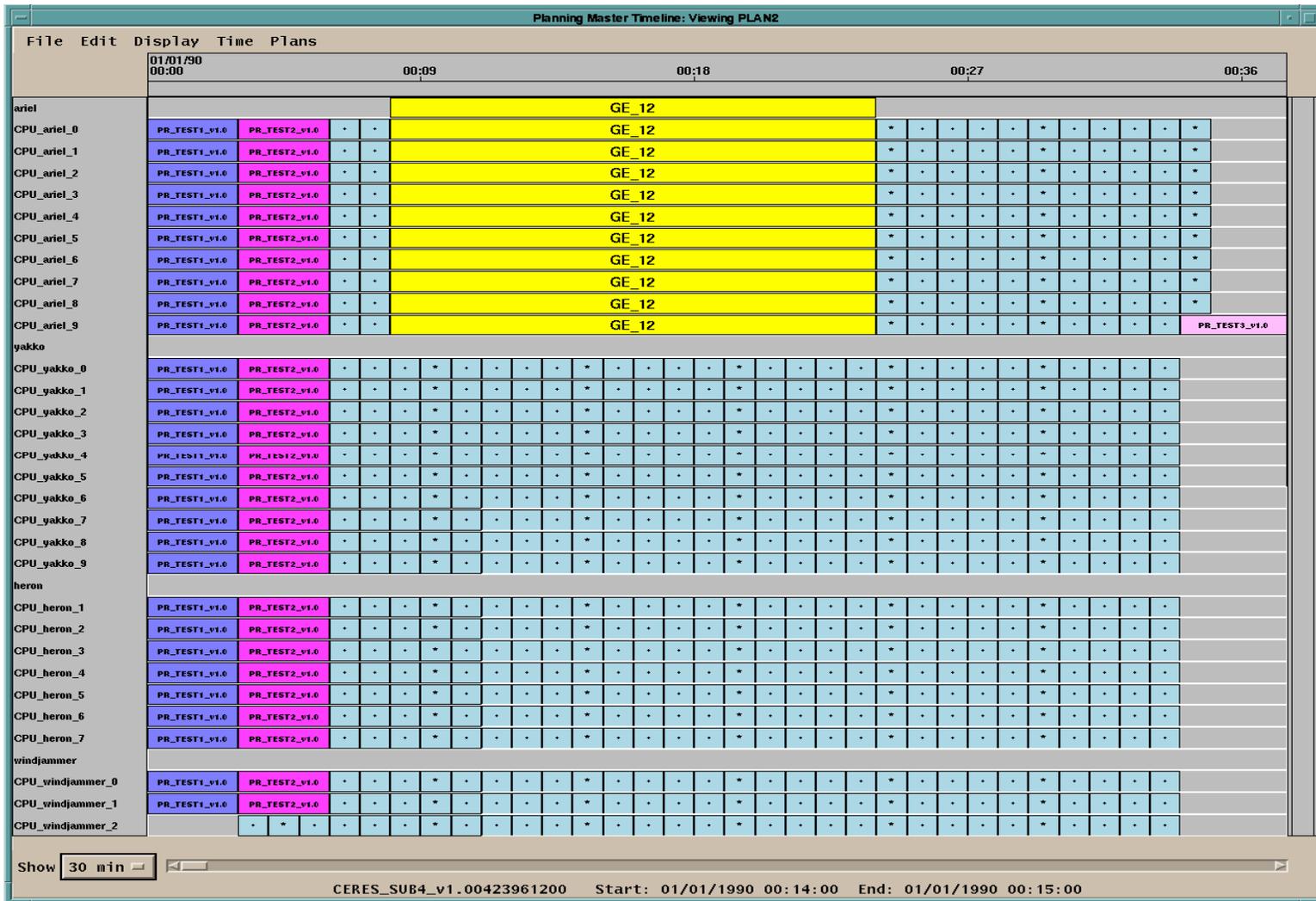


Figure 4.8.2-2. Planning Master Timeline GUI

The Planning Master Timeline Pulldown Menu Options are:

File

Open Plan: Opens a previously created plan through a file selection pop-up. It loads it in the main region of the timeline and adds it to the plans menu buttons

Load Configuration: loads a configuration for the timeline containing a list of resources and color options. It does this through a file selection

Save Configuration: saves a configuration file

Exit: quits application

Edit - Not yet supported

Display - To select resources and attributes of the display (e.g., colors)

Time - To select Start and Stop times of the plan windows

Plans - To elect plans to include

4.8.2.3 Required Operating Environment

These GUIs can be hosted on the Planning Server and Queuing Server or Planning Workstation.

For information on the operating environment, tunable parameters and environment variables of Production Strategies User Interface refer to the 920-TDA-022 “Custom Code Configuration Parameters” documentation series.

Table 4.8.2-3 identifies the supporting products this tool depends upon in order to function properly.

Table 4.8.2-3. Support Products for Production Planning Workbench

Product
Sun Solaris 8
MOTIF 1.2.3
PDPS Database

4.8.2.3.1 Interfaces and Data Types

The Production Planning Workbench has no interfaces external to ECS. It interfaces with the Data Processing Subsystem through the Planning and Data Processing System (PDPS) database.

4.8.2.4 Databases

The Production Planning Workbench updates the information in the PDPS database for subsequent use by the Data Processing software. It updates entries in DB tables such as PIDataProcessingRequest, PIPlans, and PIGroundEvent.

The complete database schema is listed in 311-CD-623, *Planning and Data Processing Subsystem Database Design and Database Schema for the ECS Project*.

4.8.2.5 Special Constraints

There are no special constraints that are associated with the production planning tools.

4.8.2.6 Outputs

Outputs of the Production Planning Workbench are provided in one of three ways:

Production Planning Workbench GUI responses as described above, including the Production Plan Timeline display, Figure 4.8.2-2.

For Updates to the PDPS database, see section 4.8.2.4.

4.8.2.7 Event and Error Messages

The Production Planning Workbench provides informational messages or warnings for minor errors, which the operator can immediately correct directly on the operator interface. Significant production planning events or errors are logged to the ECS Production Server ALOG file. Both event and error messages are listed in Appendix A.

4.8.2.8 Reports

Report generation capability is not available.

4.8.3 Production Strategies User Interface

The Production Strategies User Interface is used to tailor the priority of Production Requests (PRs). Table 4.8.3-1 summarizes the operation functionality of this tool. The priority of production requests impacts the development of processing plans. Production Strategy is a method of developing a composite priority from several factors for a production request. These factors can be assigned different proportions of the final priority. The factors themselves are composed of attributes, which can be given separate priorities. The factor attributes are then averaged to produce the priority of the factors.

The Production Strategies User Interface provides an option to display the default values for the factors and components.

Table 4.8.3-1. Common ECS Operator Functions Performed with Production Planning GUIs

Operating Function	GUI	Description	When and Why to Use
Manage Production Strategies	Production Strategies User Interface	The operator can view, create, modify, or delete Production Strategies.	When the priority of a PR needs to change to adjust processing scheduling.

4.8.3.1 Quick Start Using Production Strategies User Interface

To execute the Production Strategies User Interface from the command line prompt, enter:

```
>EcPIProdStratStart <mode>
```

where: <mode> is the ECS mode (e.g., OPS, TS1 or TS2).

Refer to the 910-TDA-022 “Custom Code Configuration Parameters” documentation series, for a listing of EcPIProdStratStart.

4.8.3.2 Production Strategies User Interface Main Screen

Figure 4.8.3-1 is the Production Strategies Main Screen.

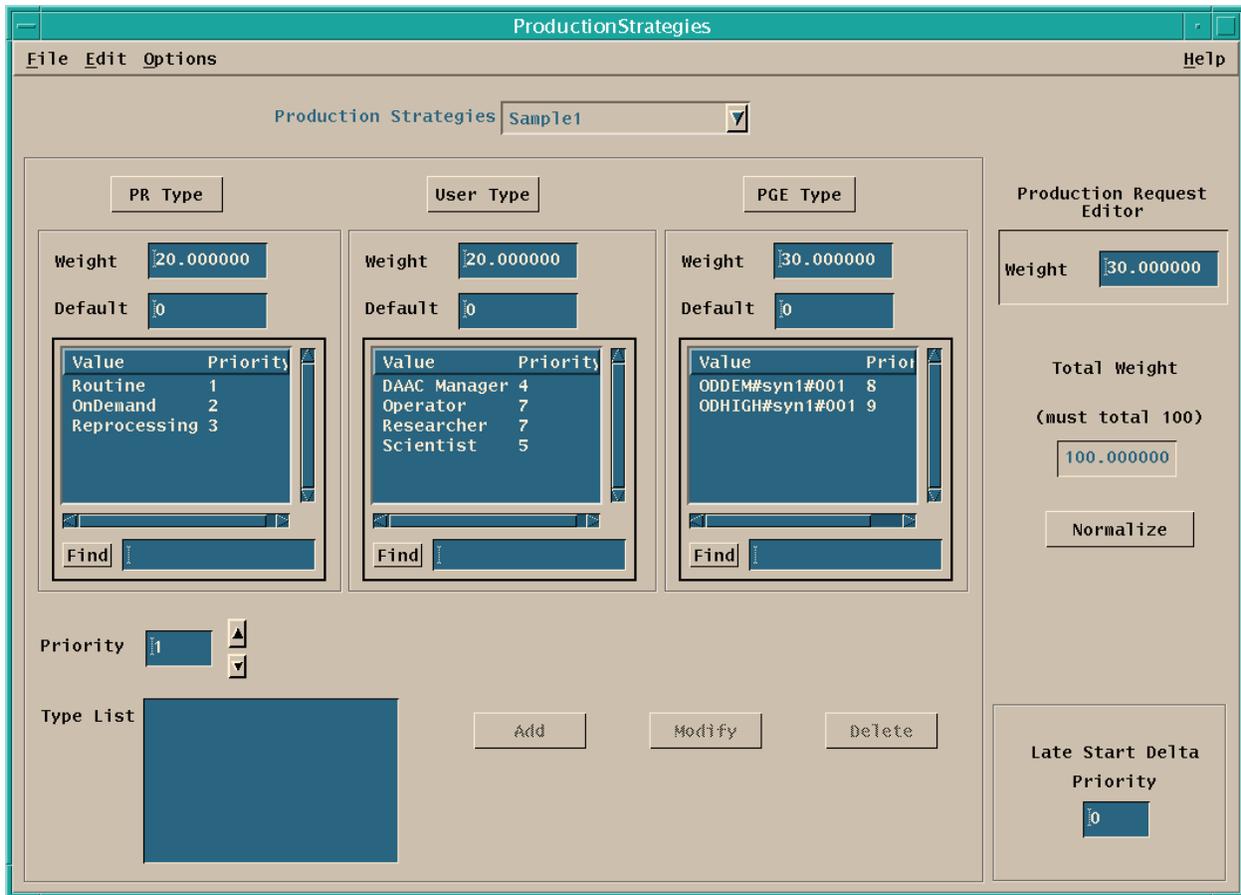


Figure 4.8.3-1. Production Strategies Main Screen

The menu bar for the Production Strategies User Interface GUI contains pulldown menus providing the following capabilities.

File Pulldown:

Open displays the Open Production Strategies pop-up window shown in Figure 4.8.3-2 with the production strategies currently in the PDPS database

New clears all the display areas on the screen for input

Save saves the displayed values to the database for the production strategy displayed

Save As saves the displayed production strategy to the database as a different strategy

Exit to exit the application

Edit Pulldown:

Delete - To delete a production strategy

Options Pulldown:

ActiveStrategy - Displays the **Active Production Strategy** screen shown in Figure 4.8.3-3)

Other options available on the Production Strategies screen:

PR Type selects the PR Type panel as active

User Type selects the User Type panel as active

PGE Type selects the PGE Type panel as active

Normalize adjusts the weighting values to total 100

Add the Type List/Priority pairs displayed on the screen to the selected type

Modify the Type List/Priority pairs displayed on the screen in the selected type

Delete the Type List/Priority pairs displayed on the screen from the selected type

The parameters associated with the Production Strategies User Interface are used as attributes and weights to generate the priorities for use by the overall Production Plan. All the parameters for the Production Strategies User Interface tools are accessible through the operator GUIs. These parameters are listed in Table 4.8.3-2. The parameters correspond to the sub-schema elements in the PDPS database referenced in Section 4.8.3.4 below.

The Production Strategies User Interface creates a two-level scheme for prioritizing the Production Plans. One level divides the plan into components: PR Type, User Type, PGE Type, and the Production Request Editor. Each of these components (except Production Request Editor) is further broken down into elements related to the Type. In the picture of the screen these elements can be observed in the list boxes under the type labels. Each of the elements can be assigned a priority.

The element priority assignment is initiated by selecting one of the type buttons. A type button brings one of the type frames into focus and displays the type elements in the Type List in the lower left corner of the screen. Element priorities are assigned or changed by selecting an element in the Type List, choosing a priority in the scroll box above the Type List, and clicking the appropriate button for Add, Modify, or Delete. The result appears in the type frame list box with the value (element) and priority displayed. The Production Strategies GUI re-computes the overall type priority based on the current element priorities of that type.

The second level of the priority scheme involves giving “weight” to the types. This “weight” is similar in concept to percentage. Note: the sum of the four “weight” values must equal 100. Each of the types can be given a weight. The fourth weight is the one assigned by the user in the Production Request Editor. As the values are entered, the program totals the weights and displays an error dialog if the total is over 100.00. The program does not allow the entry of a weight that causes the total to exceed 100.00. If the total is less than 100.00 the operator can use the “Normalize” button to scale the weight values to base 100.00.

Table 4.8.3-2. Production Strategies Field Descriptions (1 of 2)

Field Name	Data Type	Size)	Entry	Description
Production Strategies	ASCII text-	20	Default = Default	Production Strategy Id.
PR Type	--	--	--	Strategy component name (Production Request).
Weight	Floating Pt number	<100.	Default = 50.00000	Weight for this component in final priority computation.
Default	Integer	<10	Default = 5	Priority for component.
Routine	Integer	<10	Default = 5	(Sub) priority for Routine Requests.
OnDemand	Integer	<10	Default = 7	(Sub) priority for On-Demand Requests.
Reprocessing	Integer	<10	Default = 3	(Sub) priority for Reprocessing Requests.
User Type	--	--	--	Strategy component name (User Type – all users).
Weight	Floating Pt number	<100.	Default = 00.00000	Weight for this component in final priority computation.
Default	Integer	<10	Default = 5	Priority for a component.
DAAC Manager	Integer	<10	Default = 0	(Sub) priority for DAAC Manager Users.
Operator	Integer	<10	Default = 0	(Sub) priority for Operator Users.
Researcher	Integer	<10	Default = 0	(Sub) priority for Researcher Users.
Scientist	Integer	<10	Default = 0	(Sub) priority for Scientist Users.
PGE Type	--	--	--	Strategy component name (Product Generation Executive).
Weight	Floating Pt number	<100.	Default = 00.00000	Weight for this component in final priority computation.
Default	Integer	<10	Default = 5	Priority for component.
<PGEID>	Integer	<10	Default =0	(Sub) priority for PGE.
Production Request Editor	--	--	--	Strategy component name (User that made the PR).
Weight	Floating Pt number	<100.	Default = 50.00000	Weight for this component in final priority computation – the priority is part of the PR.
Total Weight	Floating Pt number	< 100.	No Default	This field is zero unless the user enters a weight in one of the 4 weight fields: PR Type Weight, User Type Weight, PGE Type Weight, or User Selected Weight. If that value then becomes > 100.0, hit the “Normalize” button to normalize.

Table 4.8.3-2. Production Strategies Field Descriptions (2 of 2)

Field Name	Data Type	Size)	Entry	Description
Priority	Integer	< 10	Default = 1	Raise or lower this value by clicking on either the up or the down arrow beside that window.
Type List	ASCII text	20	Default = blank	For an active type (for example, click on the PR Type button), the various valid types appear in this window. Inactive types and their priorities can be added using the “Add” button. Existing active types can be modified using the “Modify” button.
Late Start Delta Priority	Integer	<100	Default = 0	Priority for a component.

4.8.3.2.1 Open Production Strategy Pop-up

The **Open Production Strategies** GUI shown in Figure 4.8.3-2 displays Production Strategies stored in the database. This pop-up is used to select the Production Strategy for display on the Production Strategies Main Screen.

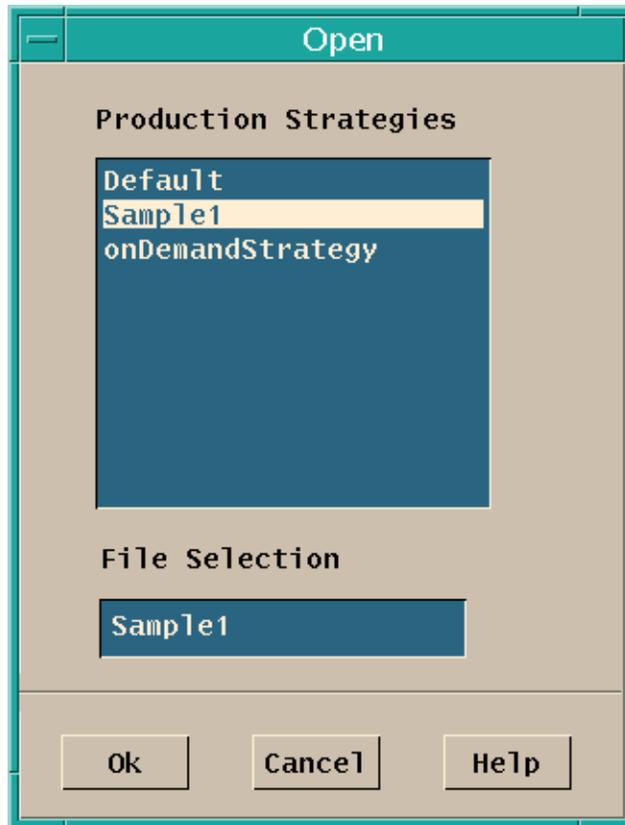


Figure 4.8.3-2. Open Production Strategies GUI Pop-up

The production strategies available in the PDPS database are displayed in the **Production Strategies** window. Selecting one of the displayed strategies causes the selected strategy to be displayed in the File Selection window. Click the “OK” button to accept the selection or the “Cancel” button to reject it to complete the open action.

4.8.3.2.2 Active Production Strategy GUI

The **Active Production Strategy** screen, shown in Figure 4.8.3-3, displays the default values from the PDPS database.

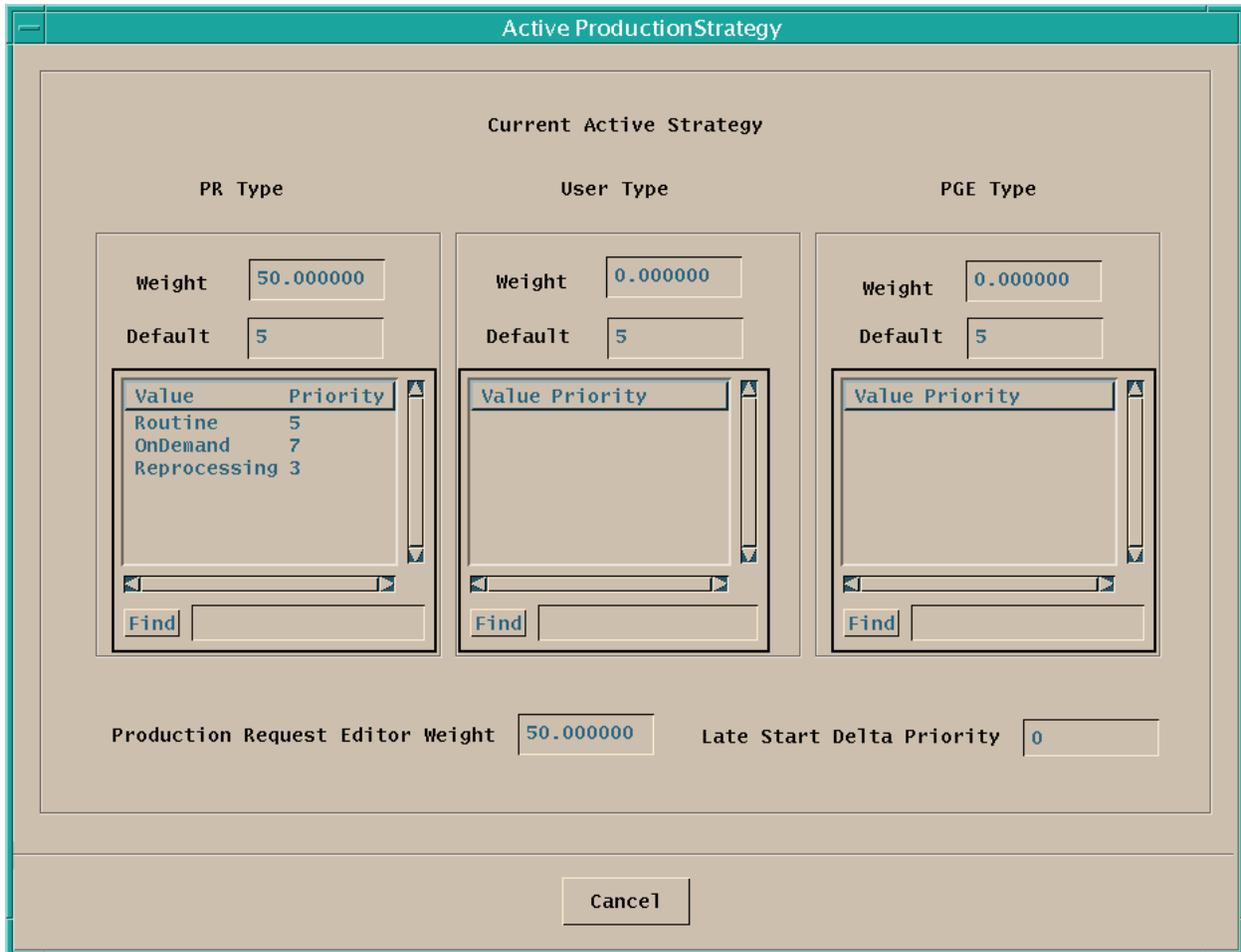


Figure 4.8.3-3. Active Production Strategy GUI Screen

The fields displayed on the **Active Production Strategy** GUI are the same as those on the Production Strategies User Interface, Table 4.8.3-2.

4.8.3.3 Required Operating Environment

These GUIs can be hosted on the Planning Server and Queuing Server or Planning Workstation.

For information on the operating environment, tunable parameters and environment variables of Production Strategies User Interface, refer to the 910-TDA-022 “Custom Code Configuration Parameters” documentation series. Table 4.8.3-3 identifies the supporting products this tool depends upon to function properly.

Table 4.8.3-3. Support Products for Production Strategies User Interface

Product
Sun Solaris 8
MOTIF 1.2.3
PDPS Database

4.8.3.3.1 Interfaces and Data Types

The Planning Subsystem has no interfaces external to ECS. It interfaces with the Data Processing Subsystem through the Planning and Data Processing System (PDPS) database to exchange planning information. The PDPS database is populated with information via the SSI&T components after the integration of a new or updated PGE. An element of the Planning system, which does not require an operator interface, the Subscription Manager interfaces with the ECS Science Data Server Subsystem.

The Planning Subsystem is primarily intended for use by the operations staff of the EOSDIS DAACs. However, the SCFs have the capability to enter Production Requests into the Planning subsystem using the same interface as described above.

4.8.3.4 Databases

The Planning Subsystem includes the PDPS database, which contains information needed to plan the processing at a DAAC. It is also used for resource planning, as it contains information on the configured resources and their allocation. The Data Processing Subsystem, the other major element of the PDPS, uses the database. The following subsections provide a summary description of the PDPS database table, and tools used with the database.

The SSI&T capability, which is a design component within the Data Processing Subsystem, includes capabilities to support the integration and test of the science software (PGEs) within the production, processing environment. As a part of this activity, the SSI&T GUIs, described in Section 4.5 of this document, provide for the entry into the databases used by SSI&T of PGE-related information needed to plan for and run science software. At the conclusion of the SSI&T process, this database information is transferred to the operational databases from the databases established for SSI&T.

The complete database schema is listed in the DID 311 document, 311-CD-623 for the release, *Planning and Data Processing Subsystem Database Design and Database Schema Specifications for the ECS Project*.

4.8.3.5 Special Constraints

There are no special constraints that are associated with the Production Strategies User Interface tools.

4.8.3.6 Outputs

Outputs of the Production Strategies User Interface are provided in the following ways:

1. Production Strategies User Interface responses as described above
2. Updates to the PDPS database (described in section 4.8.3.3)

4.8.3.7 Event and Error Messages

The Production Strategies User Interface tool provides informational messages or warnings for minor errors. Significant events or errors are logged to the ECS Production Strategies Server ALOG file. The Error messages are listed in Appendix A.

4.8.3.8 Reports

The Production Strategies User Interface does not produce any reports.

4.8.4 PIPRGenerator User Interface

The PIPRGenerator, the command line interface for the Production Request Editor, allows the user to create and activate a number of Routine Production Requests using information contained in an input file. The input file contains the PgeIds and GEOIds for the PGEs and primary input granules, respectively, for the Production Requests to be created.

Table 4.8.4-1 summarizes the operation functionality of this tool.

Table 4.8.4-1. Common ECS Operator Functions Performed with PIPRGenerator

Operating Function	GUI	Description	When and Why to Use
Generate and activate Production Requests.	No GUI Interface	The operator inputs a file containing Pgelds and GEOIds for the Production Requests to be generated.	When the user wants to generate multiple PRs for which the key input granulelds are known.

4.8.4.1 Quick Start Using the PIPRGenerator

To execute the PIPRGenerator from the command line prompt, enter:

>**EcPIPGenerator** <mode> <PRInfoFile>

<mode> is the ECS mode (e.g., OPS, TS1 or TS2).

<PRInfoFile> must be a Unix file, which contains pairs of PgeId and GEOId.

The format of PRInfoFile:

ACT#syn1#001 SC:AST_L1BT.001.19074

BTS#syn1#001 SC:AST_L1BT.001.19075

The path of the PRInfoFile is located in the EcPIPGenerator.CFG file.

Refer to the 910-TDA-022 “Custom Code Configuration Parameters” documentation series, for a listing of the EcPIPGenerator.

4.8.4.2 PIPRGenerator Main Screen

There is no main screen for the PIPRGenerator tool. This is a command line interface only.

4.8.4.3 Required Operating Environment

The required operating environment for the PIPRGenerator is the Sun Solaris 8.

4.8.4.3.1 Interfaces and Data Types

The Planning Subsystem has no interfaces external to ECS. It interfaces with the Data Processing Subsystem through the Planning and Data Processing System (PDPS) database to exchange planning information. The PDPS database is populated with information from the

SSI&T components after the integration of a new or updated PGE. An element of the Planning system, which does not require an operator interface, the Subscription Manager, also interfaces with the ECS Science Data Server Subsystem.

The Planning Subsystem is primarily intended for use by the operations staff of the EOSDIS DAACs. However, the capability has been requested and made available for science personnel from the Science Computing Facilities to enter Production Requests into the Planning subsystem using the same interface as described above. This is to be done with the consent of the DAAC operations.

4.8.4.4 Databases

The Planning Subsystem includes the PDPS database, which contains information needed to plan the processing at a DAAC. It is also used for resource planning, as it contains information on the configured resources and their allocation. The Data Processing Subsystem, the other major element of the PDPS, uses the database. The following subsections provide a summary description of the PDPS database table, and tools used with the database.

The SSI&T capability, which is a design component within the Data Processing Subsystem, includes capabilities to support the integration and test of the science software (PGEs) within the production, processing environment. As a part of this activity, the SSI&T GUIs, described in Section 4.5 of this document, provide for the entry into the PDPS database of the PGE-related information needed to plan and run science software.

The complete database schema is listed in the DID 311 document, 311-CD-623, for the release, *Planning and Data Processing Subsystem Database Design and Database Schema Specifications for the ECS Project*.

4.8.4.5 Special Constraints

There are no special constraints associated with the PIPRGenerator User Interface tools.

4.8.4.6 Outputs

Outputs of the PIPRGenerator are as stated below:

1. Generates the Series of Production Requests and activates them as stated above
2. Updates to the PDPS database (described in Section 4.8.3.4)

4.8.4.7 Event and Error Messages

The PIPRGenerator User Interface tool logs informational messages warnings and errors into the EcPIPRGenerator.ALOG and EcPIPRGeneratorDebug.log files. The Error messages are listed in Appendix A.

4.8.4.8 Reports

The PIPRGenerator User Interface does not produce any reports.

4.9 Production Processing

This section describes the Production Processing tools used by DAAC operators:

1. AutoSys/AutoXpert
2. ECS Quality Assurance (QA) Monitor

This page intentionally left blank.

4.9.1 AutoSys Job Management

This section describes how the AutoSys and the Web Interface packages are used by DAAC operations for production processing. The combination of the two packages is referred to as AutoSys Job Management in the following description.

AutoSys is a job scheduling and management COTS product providing mainframe batch scheduling functionality in a distributed UNIX environment. AutoSys has three primary components: AutoSys database, Event Processor, and Remote Agent. The AutoSys database is the repository for all system events, and for all job, monitor, and report definitions. The Event Processor is AutoSys' scheduling engine. It continuously queries the AutoSys database for events to be processed and performs necessary job, resource, and fault management actions. The Remote Agent is a transient process initiated by the Event Processor to run a program on the client machine. It sends messages directly to the Data Server to indicate changes in states for the job (e.g., STARTING, SUCCESS, FAILURE).

AutoSys is embedded in the ECS PDPS software specifically to manage the execution of jobs on science processing computers. Jobs necessary for the execution of DPRs are created and organized into "job boxes," using a custom API named "Job Management" in the ECS PDPS software.

Operator interaction is limited to "force starting" failed jobs from the Ops Console or from the JobScape GUI and monitoring the progress of jobs from JobScape. Entire job boxes should be cancelled using the Planning Subsystem's Production Request Editor. Jobs should not be deleted using the AutoSys Job Definition GUI. This does not communicate with the PDPS database.

For uninterrupted processing, AutoSys supports automatic fault tolerance management via the High Availability option. In this configuration, multiple Event Servers (Primary and Secondary) and Event Processors (Primary and Shadow) can be setup for complete redundancy in case of anomalous situations such as hardware, software, or network problems.

AutoXpert formerly bundled with Autosys is no longer shipped. In place of this component the Autosys Job Management Web Interface is used and provides all the functionality of AutoExpert with the exception of Hostscape. Its components are, JobScape, and TimeScape. HostScape still exists and provides a view of the resources, showing active states of machines, Event Server(s), Event Processors(s), and jobs. JobScape presents a PERT-like view of job processing from a logical point of view. TimeScape presents a Gantt-like view of job processing from a temporal point of view. All three GUIs provide user configurable color codes via the X resources file for correlating unique colors to job and resource states.

AutoSys Job Management is used to perform the operating functions listed in Table 4.9.1-1.

**Table 4.9.1-1. ECS Operator Functions Performed using
AutoSys Job Management (1 of 3)**

Operating Function	Command/Script or GUI	Description	When and Why to Use
Start the Event Processor(s)	Eventor	Starts the Event Processor (and optionally, the Shadow Event Processor).	To start the Event Processor(s). Normally, this would be done when the system is booted.*
Startup GUI Console Panel	/usr/ecs/<mode>/CU STOM/bin/DPS/EcD pPrStartAutosys<mode><AutoSys Instance Id>	Launches AutoSys GUI Control Panel.	To launch Ops Console, HostScape, JobScape, and TimeScape GUIs.
Access database	Xql	Invokes direct Sybase database access utility to issue SQL commands.*	Used for troubleshooting purposes.*
Check overall system	Chk_auto_up	Check the overall health of the system including the environment, configuration files, Event Server(s), and Event Processor(s).	To determine the overall health of the system for troubleshooting purposes.
Control job execution	JobScape	The GUI provides a display of job's progress.	To monitor jobs.
Determine active and completed jobs	Ops Console	View the list of successfully completed jobs by selecting the Jobs Completed button.	To view the successfully completed jobs, which have been removed from AutoSys.
Generate report	Autorep	Generates detailed or summary format report of jobs	To report on the history of job processing.
Manage security	Autosecure	Maintain Edit and Exec Superusers (Edit Superuser can edit any job definition, and Exec Superuser can execute any job and stop the Event Processor(s)).	To manage system security.*
Monitor and manage alarms	Alarm button from Ops Console, HostScape, JobScape, and TimeScape GUIs	View alarms, acknowledge them, and change the status.	To monitor and manage alarms.

**Table 4.9.1-1. ECS Operator Functions Performed using
AutoSys Job Management (2 of 3)**

Operating Function	Command/Script or GUI	Description	When and Why to Use
Monitor and manage jobs	Ops Console button from GUI Control Panel	Compact "Control Center" for monitoring jobs and sending events to jobs: (Kill Job, Force Start Job, On Hold, Off Hold). Can also Launch Alarm Manager and view reports on all events sent to a job.	To monitor and manage jobs. See Controlling Job Execution in this section.
Monitor job history	TimeScope button from GUI Control Panel	To see an overall view of processing for jobs running in AutoSys. However, jobs finishing successfully are removed, and jobs not ready to run are waiting outside of AutoSys. See: Determine active and completed jobs.	To monitor current jobs running.
Monitor jobs by Job Box	JobScope button from GUI Control Panel	Viewing the progress of jobs in the job box. Select a job and click the right mouse button to bring up an Event GUI.	To monitor status of a job.
Monitor resources	HostScope button from GUI Control Panel	Monitor machines, Event Server(s), Event Processor(s), and jobs.	To monitor resources and jobs. This GUI can also be used to check resource utilization to see if the processing load is being balanced across machines.
Perform synchronization between database and active/inactive jobs	Chase	Verifies what AutoSys thinks is running, is actually running.	This command is run as part of the Event Processor(s) startup, and should not have to be run manually.*
Print AutoSys configuration information	Autoflags	Prints AutoSys configuration information such as version and release number, database used, operating system, hostname, and hostid.	To retrieve necessary information to be supplied to the vendor for product license key generation.*

**Table 4.9.1-1. ECS Operator Functions Performed using
AutoSys Job Management (3 of 3)**

Operating Function	Command/Script or GUI	Description	When and Why to Use
Remove events	Archive_events	Removes events and alarms or job_runs information from the database.	This command is run automatically during the daily DB Maintenance cycle. *
Remove Remote Agent log files	Clean_files	*Deletes old Remote Agent log files from client machines, which have had jobs started on them.	This command is run automatically during the daily DB Maintenance cycle, and should not have to be run manually.*
Retrieve status of a job	Autostatus	Retrieves the status of a job.	To monitor job status.*
Send event	Send event from command line Send Event button from Ops Console	Sends an event to manage job execution, stop Event Processor(s), and cancel a scheduled event.	To stop the Event Processor(s), since everything else can be done from the GUIs.*
Verify machine	Autoping	Verifies both client and server machines are configured properly, and the client is functioning properly.	To verify a configured machine for troubleshooting purposes.
View job dependencies and conditions	Job_depends	Generates a detailed report about the dependencies and conditions of a job.	To view job dependencies and conditions.*
View job queue	Ops Console - Jobs Waiting Button	Used to view jobs in CQ_RELEASED, CQ_HOLD and JIL_FAILUR (either the job is already in AutoSys, or AutoSys environment variables are invalid) states by selecting the Jobs Waiting button.	Verify jobs have been queued or determine why a queued job was not released.
View successfully-completed jobs	Ops Console - Jobs Completed Button	Used to view jobs that have successfully completed and have been removed by Job Management from AutoSys.	Used to view list of jobs that have finished but are not in AutoSys.
View processing log files	Autolog	Viewing the Event Processor log file is an alternative to monitoring all jobs and events using the GUIs.	To monitor jobs, events, and alarms for troubleshooting purposes.*
* Not generally used in normal operations			

4.9.1.1 Quick Start Using Autosys Job Management

To invoke AutoSys Job Management, type from the command line prompt:

```
>$ECS_HOME/CUSTOM/utilities/EcDpPrAutosysStart <mode> <AutoSys Instance Id>
```

Where:

<mode> is the ECS mode, which this AutoSys instance is associated with and

<AutoSys Instance Id> is the instance name of this AutoSys

This script sets the appropriate AutoSys environment variables and issues the **autosvc** command to start the GUI Control Panel.

Refer to the 920-TDA-022 “Custom Code Configuration Parameters” documentation series, for a listing of the EcDpPrAutosysStart script.

AutoSys provides various scripts and executables for defining objects, checking system status, accessing the database, monitoring and reporting job status, defining custom calendars, recording sounds, generating license keys, and controlling system security (see *AutoSys User Manual*, Chapter 11 “Using AutoSys Commands”).

4.9.1.2 AutoSys Main Screen

The AutoSys GUI Control Panel provides buttons for launching (bringing up) Ops Console, Job Definition, Calendars (not used), Monitor/Browser (not used), HostScape, JobScape, and TimeScape GUIs. Figure 4.9.1-1 provides a snapshot of the Control Panel GUI.

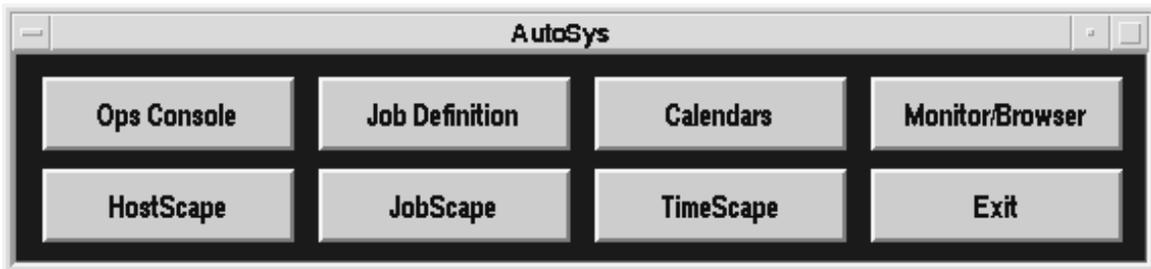


Figure 4.9.1-1. AutoSys GUI Control Panel

For more information about the AutoSys GUI Control Panel see the *AutoSys Users Manual*, Chapter 5 Defining AutoSys Jobs using the GUI.

4.9.1.2.1 AutoSys Ops Console

The AutoSys Ops Console GUI provides an interface to monitor jobs and alarms in real-time. The GUI provides a Job Selection Dialog for filtering jobs based on various parameters such as type, name, state, and machine. Detailed information about the selected job including starting conditions, dependent jobs, and reports can be viewed. In addition, there are buttons for launching Alarm Manager, Job Definition, and Send Event GUIs.

For more information about the AutoSys Ops Console please see the *AutoSys Users Manual*, Chapter 9 the Operator Console.

4.9.1.2.1.1 ECS-Added Functions to the AutoSys Ops Console

ECS has added the following three buttons to the AutoSys Ops Console:

Jobs Completed - When this button is selected a display similar to the following appears on the screen:

Jobs Completed

<u>DPR ID</u>	<u>COMPLETION STATE</u>	<u>PRIORITY</u>	<u>TYPE</u>
MODPGE01#s02280100DEV04	SUCCESS	250	Routine
MODPGE02#s28010500DEV04	SUCCESS	250	OnDemand
MODPGE02#s28011000DEV04	FAILED	250	Reprocessing
MODPGE02#s28011500DEV04	SUCCESS	250	Routine
MODPGE02#s28012000DEV04	SUCCESS	250	Reprocessing

Jobs Completed: 5
Jobs Successful: 4
Jobs Failed: 1

These jobs have completed and have been removed from AutoSys so the database is optimally maintained to increase job throughput. Their completion state is indicated, as well as their Priority and Type. Totals are listed at the bottom as to how many jobs have been completed successfully or failed. Note: the Completion State, Priority and Type are highlighted and underlined: this means you can click these to get the list reordered by any of these fields.

Jobs Waiting - Select this button to display a list of jobs scheduled, but which are waiting for data. When all of the data is available for a job, jobs are released from this holding

queue into AutoSys where they immediately start to execute. The display looks as follows:

Jobs Waiting

<u>DPR ID</u>	<u>PGE ID</u>	<u>PRIORITY</u>	<u>TYPE</u>	<u>REASON</u>
MODPGE01#s02280100DEV04	MODPGE01#syn1#001	250	Routine	Data
MODPGE02#s28010500DEV04	MODPGE02#syn1#001	250	OnDemand	Resources
MODPGE02#s28011000DEV04	MODPGE02#syn1#001	250	Reprocess	Resources
MODPGE02#s28011500DEV04	MODPGE02#syn1#002	250	Routine	Data
MODPGE02#s28012000DEV04	MODPGE02#syn1#001	250	Reprocess	Resources

Jobs Waiting: 5
Routine Jobs Waiting: 2
Reprocessing Jobs Waiting: 2
On Demand Jobs Waiting: 1

This display shows all jobs waiting to be released into AutoSys because they either do not have all their data available (Reason = Data) or there are not enough resources available for them to run (Reason = Resources). The DPR ID, PGE ID and Priority are all displayed, as well as the type of job (On Demand, Routine or Reprocessing). Note: the PGE ID, Priority, Type and Reason are highlighted and underlined: this means you can click these to get the list reordered by any of these fields.

Job Management Client Tool - The Job Management Client Program can be used to manually override the normal PDPS operations of creating jobs from information in the PDPS database and removing them after they have been successfully completed. *Except for creating and deleting ground event jobs (which must be invoked via this tool) and updating the Max Jobs In AutoSys for the various types of requests (On Demand, Routine and Reprocessing), this capability is almost always used only during system testing. You would only need to use this tool if a problem was identified and you were waiting for it to be resolved through normal channels.* You must first select a job box (not a job contained within a job box) to identify a DPR to work with. The following menu is displayed after selecting the **JM Client** button from the AutoSys Ops Console:

```
*** Current DPRId:MODPGE08#s28015500DEV04 Current Mode:DEV04 ***  
  
0) Exit  
1) Create DPR Job
```

- 2) Release DPR Job
- 3) Cancel DPR Job
- 4) Change DPR Id
- 5) View Job Management DPR Queue
- 6) Create Ground Event Job
- 7) Cancel Ground Event Job
- 8) Change Max Concurrent Jobs for PGE Limits table
- 9) Change Max/Min DPRs for Job Class
- a) Trigger release of un-released ready-to-run DPRs

Enter an option:

Once into the program, you can change DPR IDs by selecting option 4). This DPR ID stays in effect until another DPR ID is entered. Options 1) and 2), taken together, create and put a job into AutoSys, or on the queue, if AutoSys is full (the job is in a CQ_RELEASE state). Option 3 can be used to cleanly remove a job from AutoSys (as opposed to manually deleting the job from AutoSys).

Option 8 allows users to modify the DpPrPgeLimits database table. Option 9 allows users to modify the DpPrClassSchedulingLimits database table. Option A should be used whenever options 8 or 9 are used. This causes Job Management Server to re-evaluate ready DPRs for possible entry into AutoSys and execution.

Further details about how to use options 8 and 9 are explained in detail in:

/usr/ecs/<MODE>/CUSTOM/data/DPS/EcDpPrLoadTable.README

4.9.1.2.1.2 How Job Scheduling Works in PDPS/DPS

Chaining

A chain is a tree of DPRs with one root. It can be defined recursively, by picking a leaf DPR and then finding the parent DPR of that node, which contributes the most input granules used by the child. The recursion stops when a DPR only has external inputs. Ties between parents contributing an equal number of granules are resolved in no particular order.

The work of finding chains and assigning a chain a perl script named EcPIDetermineChain.pl, which is called by the Planning Workbench when a plan is activated, does Id to DPRs.

The concept of chaining is important, because all DPRs in a chain, by default are executed on a single computer. This is done to minimize communication of data on network-mounted file systems, by having most of the data for a chain produced and consumed locally.

In order to control chain processing, or to circumvent it altogether, it is necessary to declare PGEs to be chain heads in the Production Request Editor. For example, to completely deactivate PDPS chain processing, one would declare every pgeId to be a chain head. To enable total chain processing, declare no PGE to be a chain head, and EcPIDetermineChain finds chains starting with the DPREP PGEs.

Use of the DpPrPgeLimits table to control where DPRs run

All jobs in a chain are scheduled to run on machines specified for the PGE in the DpPrPgeLimits table. If no machines are listed for a PGE, another mechanism is found, which is discussed later. An easy way to balance the load on two or more computers is to specify an equal number of pgeIds to run on each computer. If this number is large, like 10,000, as many DPRs using this PGE ready to run, can potentially run, and the number is balanced on the valid computers. If the number is small, say 2 per machine, the number of DPRs using this PGE can be throttled, and the DPRs are queued.

Use of the DpPrClassSchedulingLimits table to control how many DPRs run

Using the example mentioned in the previous section, one could balance the number of PGEs on different computers by declaring an equal large number of them can run on different computers. But suppose it is desired to control the total number of DPRs that can run at any one time. This is done using the DpPrClassSchedulingLimits table. This table controls the total number of concurrent DPRs scheduled for Routine, Reprocessing and On-demand processing. When a slot is free, all DPRs ready to run and have empty slots in DpPrPgeLimits are considered, and the DPR with the oldest time stamp in PIDataProcessingRequest is selected to test.

Bypassing DpPrPgeLimits

A machine can be specified for a PGE to run on in the Production Request Editor. If no machine is specified in this way and the pgeId is not entered in DpPrPgeLimits, the computer is found from what was entered during PGE registration in the PIResourceRequirement table.

How to modify the DpPrPgeLimits and DpPrClassSchedulingLimits tables

Two options in the Job Management Client tool, which is accessed through the AutoSys Job Activity Console, allow one to modify these tables by specifying a file of table modification commands:

8) Change Max Concurrent Jobs for PGE Limits table

9) Change Max/Min Dprs for Job Class

Use of these options is explained in detail in:

`/usr/ecs/<MODE>/CUSTOM/data/DPS/EcDpPrLoadTable.README`

Alternatively, one can write one's own load script with SQL statements. You can always add pgeIds for a machine, and new machines to the DpPrPgeLimits table and change the maximum number of DPRs that can concurrently execute in DpPrClassSchedulingLimits, but you must never modify the number currently scheduled or running DPRs in these tables.

There can be no entries in DpPrPgeLimits, but DpPrClassSchedulingLimits must be fully populated. If this table is empty when the Job Management server starts, it loads it by reading the following configuration parameters:

```
DpPrMaxConcurrentDPRs    100 100 100
DpPrMinConcurrentDPRs    0 0 0
```

Job Management puts DPRs into AutoSys

When a DPR completionState is set to PENDING by the PLS Subscription Manager, or is released by the Planning Workbench, the Job Management server checks information in the limits tables and make a determination as to which is the next job to be placed into AutoSys. When jobs finish in AutoSys, or are cancelled by the Production Request Editor, this too triggers Job Management to check its queue to see which jobs can be waiting to get into AutoSys.

In very rare occasions, it is possible for DPS processing to freeze up if no triggering events can occur. In this case, you can use option a) from the Job Management Client to "wake up" the Job Management Server :

a) Trigger release of unreleased ready-to-run Dprs

4.9.1.2.2 AutoSys Job Definition

The AutoSys Job Definition GUI provides an interface to add new jobs, delete existing jobs, and modify attributes for existing jobs. Both Date/Time Options and Adv Features Dialogs can be used to set/modify all job attributes.

For more information about the AutoSys Job Definition please see the *AutoSys Users Manual*, Chapter 5 Defining AutoSys Jobs using the GUI.

Note: This capability must never be used without first checking with someone who knows something about the system internals. The ECS Job Management software handles the creating, running and removing of successfully completed jobs. Manually performing these functions bypasses Job Management housekeeping and can cause the processing system to get out of sync, possibly with severe consequences.

4.9.1.2.3 AutoSys Calendars

Not used by ECS.

4.9.1.2.4 AutoSys Monitor/Browser

Not used by ECS.

4.9.1.2.5 AutoSys Alarm Manager

The AutoSys Alarm Manager GUI provides an interface to monitor and manage alarms. Alarms can be filtered based on type, state, and time; a response can be registered for an alarm; and the state of the alarm can be changed. The Alarm Manager is invoked by clicking on the "Alarm" buttons on either the JobScope or AutoSys Operator's Console GUI.

For more information about the AutoSys Alarm Manager please see the *AutoSys Users Manual*, Chapter 9, The Operator Console.

4.9.1.2.6 AutoSys HostScope

The HostScope GUI provides an interface to monitor the states of resources, Event Server(s), Event Processor(s), and jobs. This GUI also provides buttons for launching Alarm and Job

Console GUIs. For more information about HostScope see *AutoSyst User Guide for UNIX Version 4.5, Product Overview*.

4.9.1.2.7 Autosys JobScope

The JobScope GUI provides an interface to monitor states of jobs. This GUI presents a Pert-like (network) view of job processing from a job dependency point of view. Alarm and Job Console GUIs can be launched from this GUI. For more information about JobScope see *AutoSys User Guide for UNIX Version 4.5, Product Overview*.

4.9.1.2.8 Autosys TimeScope

The TimeScope GUI provides an interface to monitor states of jobs. This GUI presents a Gantt-like (chart) view of job processing from a time-related point of view. This GUI also provides buttons for launching Alarm and Job Console GUIs. For more information about TimeScope see *AutoSys User Guide for UNIX Version 4.5, Product Overview*.

4.9.1.3 Required Operating Environment

For all COTS packages, appropriate information on operating environments, tunable parameters, environment variables, and a list of vendor documentation can be found in a CM controlled document for each product. To find the documentation for AutoSys, refer to the ECS Baseline Information System web page,

<http://pete.hitc.com/baseline/index.html>.

No tuning or configuration of the product is required, beyond that required by a normal installation.

4.9.1.3.1 Interfaces and Data Types

AutoSys exchanges data of various types through interfaces within ECS. Table 4.9.1-2 lists AutoSys system interfaces for Release 6A.

Table 4.9.1-2. Interfaces Between AutoSys and Other ECS PDPS Components

Interface	Type of Primary Interface Protocols	Type of Backup Interface Protocols	Comments
Job Management, Job Management Client, View Job States	Job Interface Language, Library	None	The Job Management processes use the AutoSys COTS product to create and initiate execution of Data Processing Subsystem administrative jobs managing the Science Processor hardware (SPRHW) assets and for PGE execution.
PGE Execution Manager, Resource Usage	Library	None	The PGE Execution Manager controls and monitors PGE executions including Process Control File and output product storage growth. The Resource Usage process measures the actual resources used by the PGE and reports to AutoSys unexpected resource usage.
Execution Management	Library	None	The Execution Management process initiates the execution of PGEs via AutoSys. It supports the preparation activities prior to PGE execution and subsequent activities after PGE execution. It also provides status of on-demand processing requests and sends out e-mail to originators in case of a failure.

The information in the above table is informational only. It is not needed for normal operations.

4.9.1.4 Databases

All AutoSys information is stored in one of the three Sybase database types. The three types of databases are: Event Server, Monitor Server, and Alarm Server. The Event Server contains all the information about a particular instance of AutoSys such as job definitions, events, monitor and browser definitions, calendar information, and machine definitions. Monitor and Alarm Servers, not used within the ECS, are specialized databases for monitoring events and alarms.

See *AutoSys User Manual*, Chapter 13 “AutoSys Databases” for database schema definitions.

4.9.1.5 Special Constraints

None.

4.9.1.6 Outputs

AutoSys Event Processor(s), Event Server(s), and Remote Agent maintain log files. Table 4.9.1-3 lists AutoSys output log files.

Table 4.9.1-3. Outputs

Output	Description and Format
\$AUTOUSER/out/event_demon.\$AUTOSERV	Event Processor log file.
\$SYBASE/install/errorlog	Event Server error log file.
<i>AutoRemoteDir/auto_rem.joid.run_num.ntry</i>	Remote Agent log file.

4.9.1.7 Event and Error Messages

AutoSys issues both status and error messages to the Event Processor log file (\$AUTOUSER/out/event_demon.\$AUTOSERV).

4.9.1.8 Reports

AutoSys autorep utility provides a method of generating reports containing information about jobs, machines, and global variables currently defined in the database. Table 4.9.1-4 lists AutoSys reports that can be generated. See the *AutoSys User Manual* for further information.

Table 4.9.1-4. Reports

Report Type	Report Description	When and Why Used
Job	The report provides information about jobs and their processing status Summary, detail, and query formats can be generated.	It should be used primarily for generating production history reports, but also can be used for real-time monitoring of jobs.
Machine	The report provides status of machines and attributes such as maximum load, current load, and factor. Summary, detail, and query formats can be generated.	It should be used for real-time monitoring of machines.

4.9.1.8.1 Sample Report

Figure 4.9.1-2 shows a sample AutoSys autorep Job Report obtained by using the command:

```
autorep - d -J % > report
```

Job Name Pri/Kit	Last Start	Last End	Status	Run	
BTS#syn1#020704981301	12/11 17:19	12/11 21:08	SUCCESS	9837/1	
Status/[Event]	Time	Ntry	EventState	ProcessTime	Machine
RUNNING	12/11 17:19:01	1	Processed	12/11 17:19:19	
SUCCESS	12/11 21:08:54	1	Processed	12/11 21:08:59	
BTS#syn1#020704981301A	12/11 17:56	12/11 18:00	SUCCESS	9837/2	
Status/[Event]	Time	Ntry	EventState	ProcessTime	Machine
-	-	-	-	-	-
STARTING	12/11 17:19:15	1	Processed	12/11 17:19:19	pdps1
RUNNING	12/11 17:19:19	1	Processed	12/11 17:19:24	pdps1
FAILURE	12/11 17:19:48	1	Processed	12/11 17:19:50	
[*** ALARM ***]					
JOBFAILURE	12/11 17:19:50	1	Processed	12/11 17:19:56	pdps1
[FORCE_STARTJOB]	12/11 17:56:19	0	Processed	12/11 17:56:27	
STARTING	12/11 17:56:24	2	Processed	12/11 17:56:27	pdps1
RUNNING	12/11 17:56:28	2	Processed	12/11 17:56:32	pdps1
SUCCESS	12/11 18:00:51	2	Processed	12/11 18:00:58	
BTS#syn1#020704981301S	12/11 18:00	12/11 18:11	SUCCESS	9837/1	
Status/[Event]	Time	Ntry	EventState	ProcessTime	Machine
-	-	-	-	-	-
STARTING	12/11 18:00:55	1	Processed	12/11 18:00:58	pdps1
RUNNING	12/11 18:00:59	1	Processed	12/11 18:01:10	pdps1
SUCCESS	12/11 18:11:25	1	Processed	12/11 18:11:32	
BTS#syn1#020704981301P	12/11 19:02	12/11 19:02	SUCCESS	9837/2	
Status/[Event]	Time	Ntry	EventState	ProcessTime	Machine
-	-	-	-	-	-
STARTING	12/11 18:11:29	1	Processed	12/11 18:11:33	pdps1
RUNNING	12/11 18:11:33	1	Processed	12/11 18:11:38	pdps1
SUCCESS	12/11 18:13:20	1	Processed	12/11 18:13:27	
[FORCE_STARTJOB]	12/11 19:01:56	0	Processed	12/11 19:02:02	
STARTING	12/11 19:02:02	2	Processed	12/11 19:02:07	pdps1
RUNNING	12/11 19:02:04	2	Processed	12/11 19:02:07	pdps1
SUCCESS	12/11 19:02:22	2	Processed	12/11 19:02:28	

Figure 4.9.1-2. AutoSys autorep Job Report

Figure 4.9.1-3 shows an AutoSys autorep Job Report (All) obtained by using the command:

autorep -d -m ALL >report

Machine Name	Max Load	Current Load	Factor	O/S
chimera	---	---	1.00	Unix
deepsea	---	---	1.00	Unix
guinness	---	---	1.00	Unix
lasher	---	---	1.00	Unix
monet	---	---	1.00	Unix
nessie	---	---	1.00	Unix
pdps1	---	---	1.00	Unix
string1.pdps1	1	0	----	Unix
pdps2	---	---	1.00	Unix
string1.pdps2	2	0	----	Unix
relbsgi	---	---	1.00	Unix
sdps2	---	---	1.00	Unix
seagull	---	---	1.00	Unix
Current Jobs:				
Job Name	Machine		Status	Load
Priority				
<hr/>				
ETS#syn1#020704971301			RUNNING	1 250

Figure 4.9.1-3. AutoSys autorep Job Report - All

This page intentionally left blank.

4.9.2 ECS Quality Assurance (QA) Monitor

ECS QA Monitor processing capabilities enable DAAC operators to perform duties associated with DAAC QA activities. The ECS Quality Assurance (QA) Monitor GUI is the user-interface for entering data requests and displaying data, status, and error messages. The QA Monitor does not produce data products, but communicates with the science data server to retrieve data that have been previously archived.

The QA Monitor GUI is used to perform the operator functions listed in Table 4.9.2-1.

Table 4.9.2-1. Common ECS Operator Functions Performed with QA Monitor

Operating Function	Description	When and Why to Use
Query data granules	Initiates a request to search the science archive for data granules within a date ¹ interval.	When there is a need to know all archived data granules with the same data type and inserted in the archive at a certain time (date interval).
Retrieve data granules	Initiates a request to get data granules from the science archive.	When data granule(s) needs to be transferred from archive to local disk for visualization.
Visualize data (HDF files)	Display Visualize screen.	When graphical images of data granules need to be viewed to assess quality.
Update metadata	Initiates a request to archive QA information about data granules.	When QA information about data granules needs to be updated in the archive based on DAAC QA activities encompassing use of the Visualize Data function.

4.9.2.1 Quick Start Using QA Monitor

The QA Monitor is used to search for granules in the science archive, to retrieve granules and/or processing history files from the science archive and to update metadata of granules retrieved from the science archive. The QA Monitor GUI's main screen has three windows: a window that displays the ESDTs of granules in the science archive, a data granule display window to select data to browse, and a status window to indicate progress or display error messages. Several dialog screens may be activated when functions are performed. Some windows are display-only for informational text, while others require user-interaction.

4.9.2.1.1 Invoking QA Monitor Tool from the Command Line Interface

To execute QA Monitor from the command line prompt use the startup script provided during installation and located in utilities subdirectory of the operational directory structure:

```
EcDpPrQaMonitorGUIStart <mode>
```

¹ This is the data acquisition beginning and ending date.

<mode> is the ECS Mode for the execution (e.g., OPS, TS1 or TS2).

Refer to the 920-TDA-022 “Custom Code Configuration Parameters” documentation series, for a listing of EcDpPrQaMonitorGUIStart.

4.9.2.2 QA Monitor Main Screen

The QA Monitor Tool Main Screen shown in Figure 4.9.2-1 provides access to all QA Monitor primary functions. It consists of the QRU (Query/Retrieval/Update) data Tab and the Visualize data Tab. The QRU data Tab is activated when the QA Monitor Main Screen is initialized. A group of related objects is displayed by default to help the operator get started. The Data Types, Data Granules, Date Granule Insert (acquisition date of granules) windows, and the Query pushbutton are displayed, plus four desensitized (grayed-out) pushbuttons. These objects and the QRU data object which controls display of this screen are needed to perform a query. Other available data types are displayed in the Data Types window so the operator can choose which data type to query on. This enforces the sequence of events: query, retrieve and update. The grayed-out pushbuttons are sensitized after a Query, which permits selection of the indicated functions. The Visualize data Tab displays a list of science granules available for browsing.

From this screen, the following functions can be performed:

- Request the Science Data Server to search for specific types of Data Granules (Section 4.9.2.2.1)
- Request the Science Data Server to transfer Data Granules to the operator’s computer
- Request the Science Data Server to transfer Production History to the operator’s computer
- Update Data Granules’ Metadata (Section 4.9.2.2.2)
- View graphical images of Data Granules (Section 4.9.2.2.3)
- Print Display Lists of Data Granules and Data Types (Section 4.9.2.2.4)

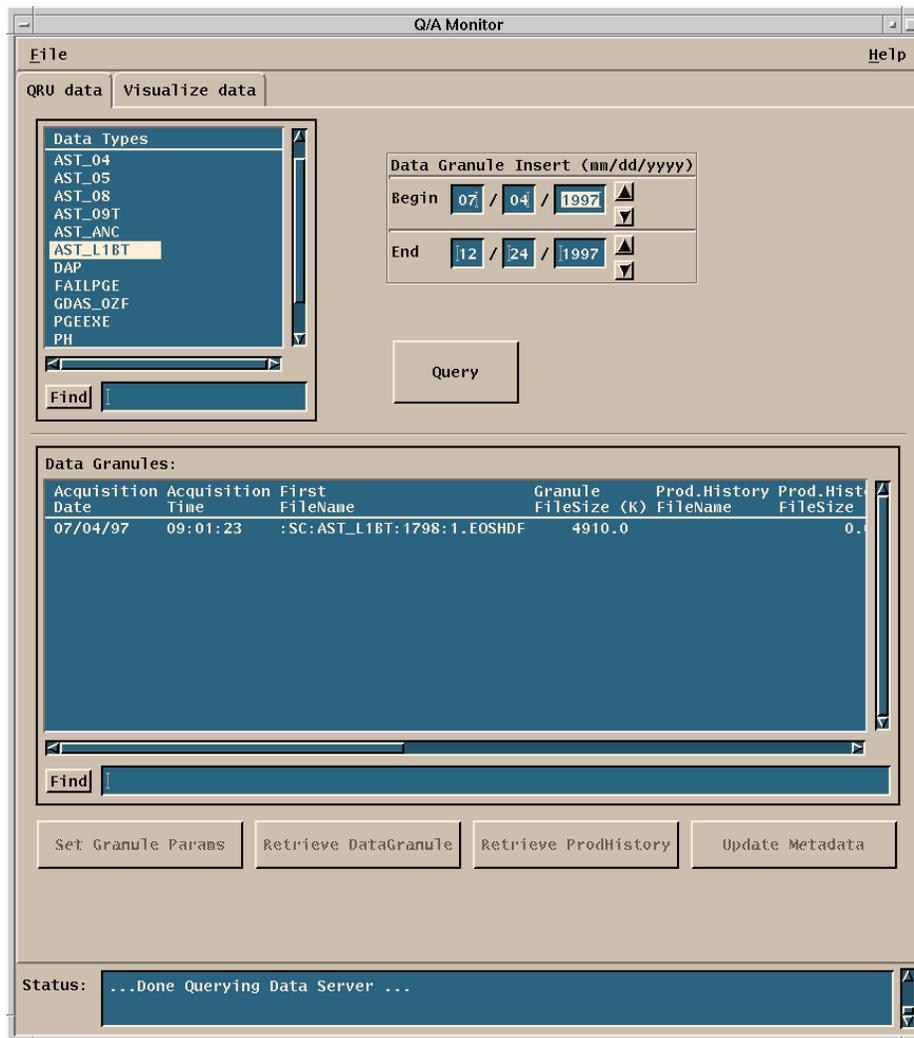


Figure 4.9.2-1. QA Monitor Tool Main Screen

The operator can select the following menu items from the QA Monitor Main Screen

- **File** This menu contains the following items:
 - Print** allows the operator to print the Data Types list and the Data Granules list (Section 4.9.2.2.4)
 - Exit** allows the operator to exit the QA Monitor
- **Help** The "Help" menu offers the following help choices: (Note: Presently not functional)
 - On Help** displays help about using help
 - On Context** displays help about individual screen elements. Context sensitive help pops up a scrolled window that displays text about a specific user interface component. There are two ways to get context sensitive help:

The first way to get context sensitive help is to move focus to an active user interface component (e.g., text entry field, scroll list) and press the Help key. Help is displayed about that element. If the keyboard does not have a key labeled Help, the F1 key usually functions as the help key.

The second way is to select "On Context" from the Help menu. The cursor changes to a question mark. Clicking on any user interface component displays help about that element.

On Window displays high-level help about the window (see "Tutorial" below)

On Keys displays help about using the keyboard and general help about standard user interface components that support complex behaviors. Topics include:

1. How to get context sensitive help
2. How to use keyboard mnemonics and accelerators to access menu functions without using the mouse
3. How to navigate between active fields on windows using the keyboard

Index is currently not supported.

Tutorial displays an extended discussion of the application and its role within the ECS system.

On Version identifies what version of ECS software is currently being used.

On Mode identifies the operating mode of the ECS software currently being used.

Additionally, the following functions may be selected on the QRU data Tab:

- **Query:** refer to Section 4.9.2.2.1
- **Set Granule Params:** refer to Section 4.9.2.2.2
- **Retrieve DataGranule:** refer to Section 4.9.2.2.1
- **Retrieve ProdHistory:** refer to Section 4.9.2.2.1
- **Update Metadata:** refer to Section 4.9.2.2.2

The following functions may be selected on the Visualize data Tab (refer to Section 4.9.2.2.2):

- **Visualize:** refer to Section 4.9.2.2.3
- **Help:** refer to Section 4.9.2.2.3

4.9.2.2.1 Query and Retrieve Data

The QA Monitor allows the operator to search the data archive for data granules by data type. The available data types are displayed when the main screen comes up and allows the operator to choose which data type to query on. The operator may accept the default dates or enter new dates to further constrain the search. Clicking on the Query pushbutton initiates the search

request. If a search is successful a list of data granules are displayed. Then the operator can have them transferred from the data archive to his computer. Where the data is located on the operator's computer is a configurable item. To request transfer of a data granule, highlight the row with information about the granule desired by clicking on it then click on the Retrieve Data pushbutton. A message appears in the Status window indicating the processing status. To get the production history file associated with the granule just click on the Retrieve ProdHist pushbutton.

For a description of the "Query and Retrieve Data" fields, see Table 4.9.2-2 below.

Table 4.9.2-2. QA Monitor Field Descriptions

Field Name	Data Type	Size	Entry	Description
Data Granule Insert Begin End	Date Min Max	8 8	Initial default value (1/2/1901-6/1/2036) can be changed by the operator	The date interval window that data granules are inserted into the data server.
Data Types list	Single Selection	N/A	The Operator selects a data type from the list displayed at startup	The list of all available data types at a specific DAAC.
Data Granules list	Multiple Selection	N/A	The Operator clicks data granule row(s) then clicks retrieve pushbutton	The list of all data granules (in the date interval above) for a particular selected data type is available for retrieval.
Status	Text	N/A	Display status messages only	Display status messages.

In addition, the following pushbuttons are provided:

- **Query** populates the list of data granules on the bottom half of the screen for a particular selected data type within a date interval.
- **Find** (below the data types and data granules list) allows the operator to perform a keyword search for information stored in those 2 lists.
- **Retrieve DataGranule** allows the operator to retrieve data granule(s) from the DAAC's data archive to local disk.
- **Retrieve ProdHistory** allows the operator to retrieve a production history tar file from the DAAC's data archive to local disk.
- **Update Metadata** pops up a Granule Parameters screen (see Section 4.9.2.2.2 below).

4.9.2.2.2 Update Metadata

Clicking on one data granule displayed in the Data Granules Field and then the Update Metadata pushbutton brings up the Granule Parameters screen shown in Figure 4.9.2-2. The list of parameter names is attributes describing the selected data granule. The settings for the parameters are determined by the DAAC, SCF or PGE. Those settings are displayed along with explanations, if any. After DAAC QA of a data granule the DAAC operator may update the operational QA flag providing an explanation. To update a parameter, click the parameter name then click OK. The Update Metadata screen is displayed as shown in Figure 4.9.2-3.

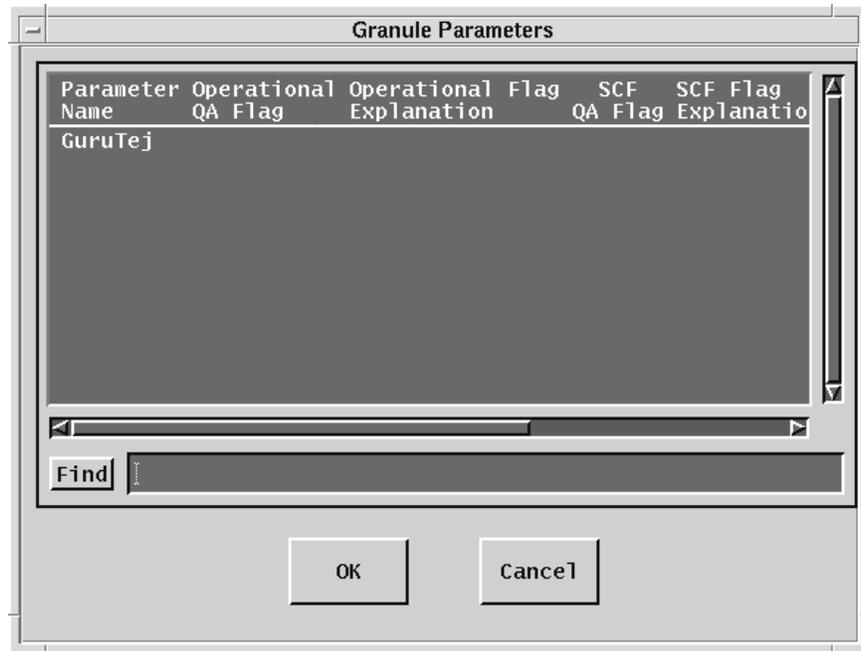


Figure 4.9.2-2. Granule Parameters Dialog

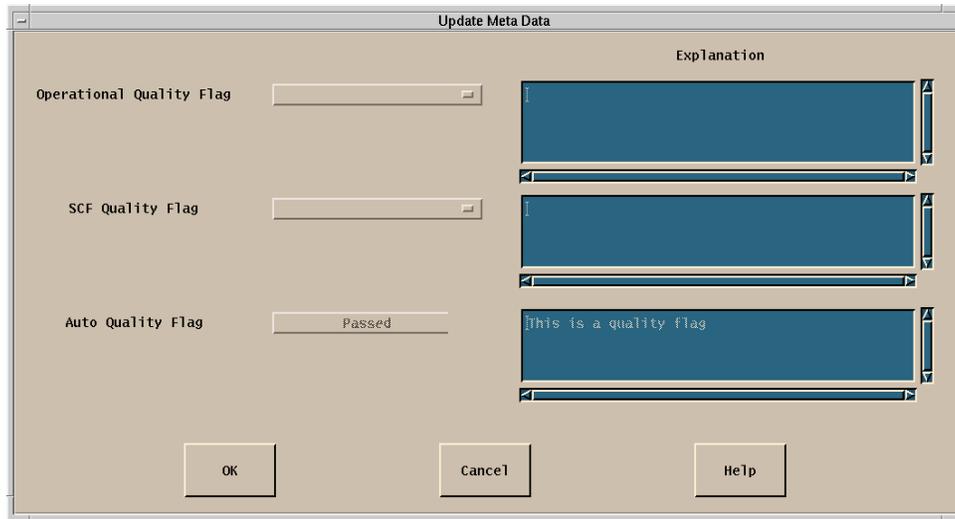


Figure 4.9.2-3. Update Metadata Dialog

For a description of the “Update Metadata” dialog fields, see Table 4.9.2-3 below.

Table 4.9.2-3. Update Metadata Field Descriptions

Field Name	Data Type	Size	Entry	Description
Operational Quality Flag, SCF Quality Flag	Character	N/A	Operator selects value	DAAC and SCF quality status setting of a data granule parameter. The valid values are: <ul style="list-style-type: none"> • Passed • Failed • Being investigated • Not being investigated
Operational Quality Explanation, SCF Quality Explanation	Character	Less than 150 chars	Operator inputs text	Text describing quality status.
Auto Quality Flag	Character	N/A	Operator input not allowed	Quality status of a data granule parameter set during data processing. The valid values are: <ul style="list-style-type: none"> • Passed • Failed • Being investigated • Not being investigated
Automatic Quality Explanation	Character	N/A	Operator input not allowed	Quality status of a data granule parameter set during data processing.

In addition, the following pushbuttons are provided:

- **OK** sends an update request to the data server to update data granule parameter values
- **Cancel** cancels the update request
- **Help** displays the help for Update Metadata

4.9.2.2.3 Visualize Data

Clicking on Visualize data from the QA Monitor GUI main screen brings up the **Visualize data** Tab shown in Figure 4.9.2-4. The operator clicks a row with the name of a science file to browse. When the Visualize pushbutton is activated EOSView is invoked. The EOSView GUI requires operator input to produce a graphical image of the science data file (see Section 4.12.5 “EOSView”). Note that the EOSView GUI can only read data products that are in HDF format.

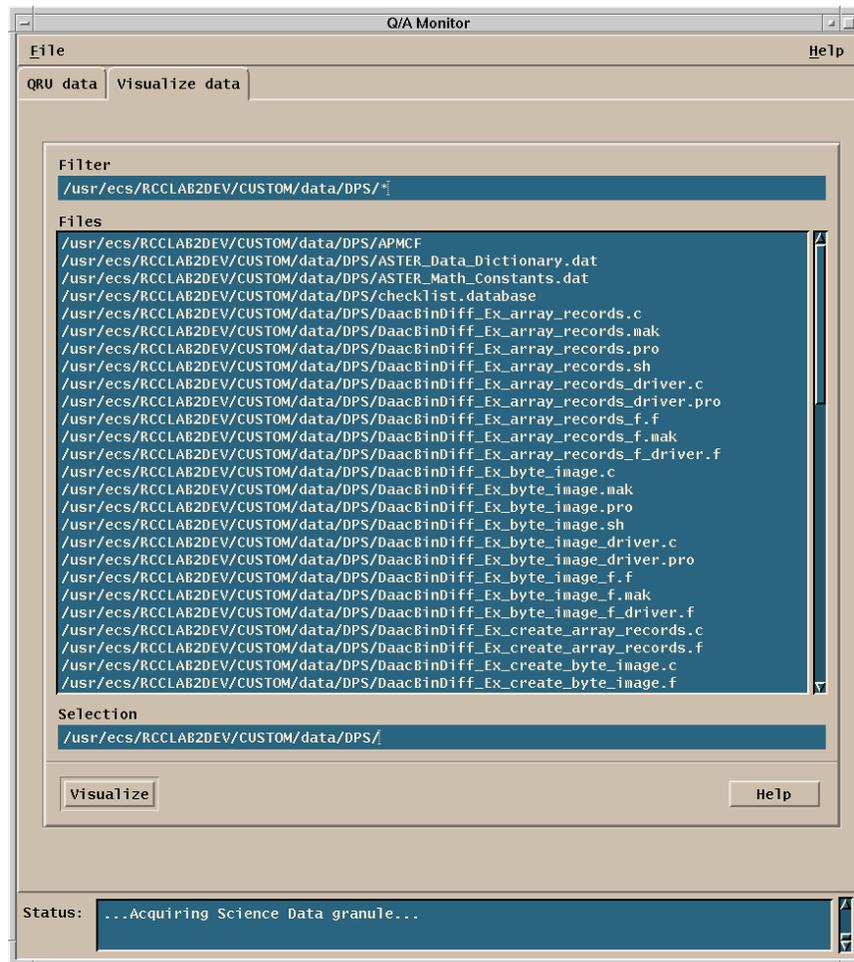


Figure 4.9.2-4. Visualize Data Tab Stack

For a description of the “Visualize data” fields, see Table 4.9.2-4 below.

Table 4.9.2-4. Visualize Data Field Descriptions

Field Name	Data Type	Size	Entry	Description
Filter	Character	N/A	Unedited	Indicates path and filenames to be displayed as established in the configuration file.
Files	Character	N/A	Click on a row	Highlights row information.
Selection	Character	N/A	Unedited	Full pathname of the file selected for viewing is displayed.

In addition, the following pushbuttons are provided:

- **Visualize** invokes EOSView for visualizing data products
- **Help** displays help for the file selection box

4.9.2.2.4 Print Options

Clicking on “File” pull-down menu brings up the **Print Options** dialog shown in Figure 4.9.2-5. This dialog allows the operator to select which list, data types list or data granules list, or both, to be sent to the printer.

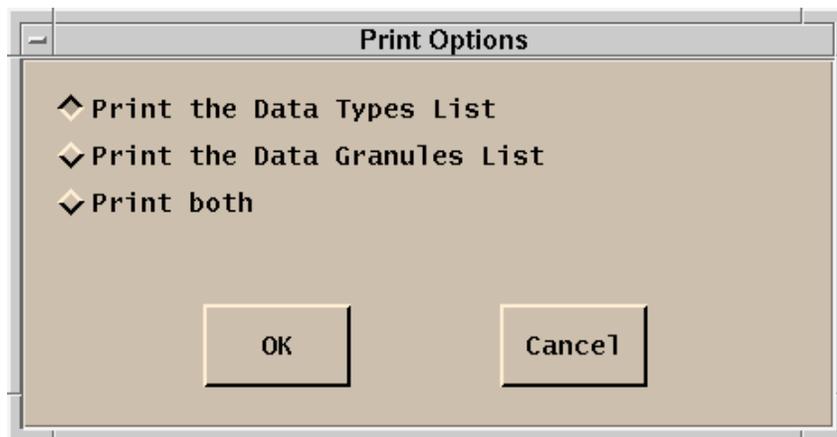


Figure 4.9.2-5. Print Options Dialog

For a description of the “Print Options” fields, see Table 4.9.2-5 below.

Table 4.9.2-5. QA Monitor Tool Field Descriptions

Field Name	Data Type	Size	Entry	Description
Print the Data Types List	N/A	N/A	Select 1 at a time	This toggle button is selected if the operator wants the data types list to be sent to the printer.
Print the Data Granules List	N/A	N/A	Select 1 at a time	This toggle button is selected if the operator wants the data granules list to be sent to the printer.
Print both	N/A	N/A	Select 1 at a time	This toggle button is selected if the operator wants both lists to be sent to the printer.

In addition, the following pushbuttons are provided:

- **OK** sends a print request to the printer for option selected
- **Cancel** cancels the print request

4.9.2.3 Required Operating Environment

For information on the operating environment, tunable parameters and environment variables of the QA Monitor Tool refer to the 920-TDA-022 “Custom Code Configuration Parameters” documentation series.

4.9.2.3.1 Interfaces and Data Types

The QA Monitor Tool does not have any external interfaces to ECS. However, there is an internal interface to the Science Data Server to query and retrieve data granules, and update metadata.

4.9.2.4 Databases

None.

4.9.2.5 Special Constraints

None.

4.9.2.6 Outputs

Other than the output displayed to the operator the QA Monitor is responsible for file (science data files in the HDF format) transfers from the science archive to the local disk of the QA Monitor host computer. It also sends data to the science archive to be stored.

4.9.2.7 Event and Error Messages

The QA Monitor issues status messages to the Status Messages text area at the bottom of the QA Monitor GUI and error messages on pop-up error dialogs. Both event and error messages are listed in Appendix A.

4.9.2.8 Reports

None.

This page intentionally left blank.

4.10 Science Data Archive and Distribution

This section describes the Science Data Archive and Distribution tools used by DAAC operators.

The Data Server Subsystem (DSS) has the responsibility for storing earth science and related data in a persistent fashion, providing search and retrieval access to this data, and supporting the administration of the data, hardware devices, and software products. As part of its retrieval function, the subsystem also provides for distribution of data electronically or on physical media.

The ECS Data Server Operator GUIs provide normal operational control and insight into science data server, storage management, and data distribution subsystem operations. These views into the system are managed by means of five operational tools described in the following sections:

4.10.1 Science Data Server GUI

4.10.2 Storage Management Control GUI

4.10.3 Data Distribution Requests GUI

4.10.4 Granule Deletion Administration Tool

The Product Distribution System (PDS) Stand Alone (PDSSA) User Interface has been removed for Release 7.11 and is not addressed in this document.

This page intentionally left blank.

4.10.1 Science Data Server GUI

The Science Data Server GUI provides the operator two major functions, the management of Earth Science Data Types and the management of all types of requests that the Science Data Server operator is involved with. Further details on these two functions are given in Table 4.10.1-1.

Table 4.10.1-1. Common ECS Operator Functions Performed with the Science Data Server GUI

Operating Function	GUI	Description	When and Why to Use
Manage Science Data Server Earth Science Data Types (ESDTs)	Data Types Tab	Allows operators to manage the ESDTs offered by the Science Data Server	As needed, to manage data type descriptor information and add and update ESDTs
Manage Data Server System Requests	System Requests Tab	Allows operators to manage all the requests within each data server component	As required, to manage requests in each data server component

4.10.1.1 Quick Start Using the Science Data Server

To invoke the ECS Science Data Server GUI, enter the following:

EcDsSdSrvGuiStart <mode>

Where:

<mode> is the ECS mode for the execution (e.g., OPS, TS1 or TS2).

4.10.1.2 Science Data Server Main Screen

The ECS Science Data Server GUI, shown in Figure 4.10.1-1, has two tabs that provide access to each one of the component's screens.

- The Earth Science Data Type Manager is accessed through the **Data Types** tab
- The System Request Manager is accessed through the **System Requests** tab.

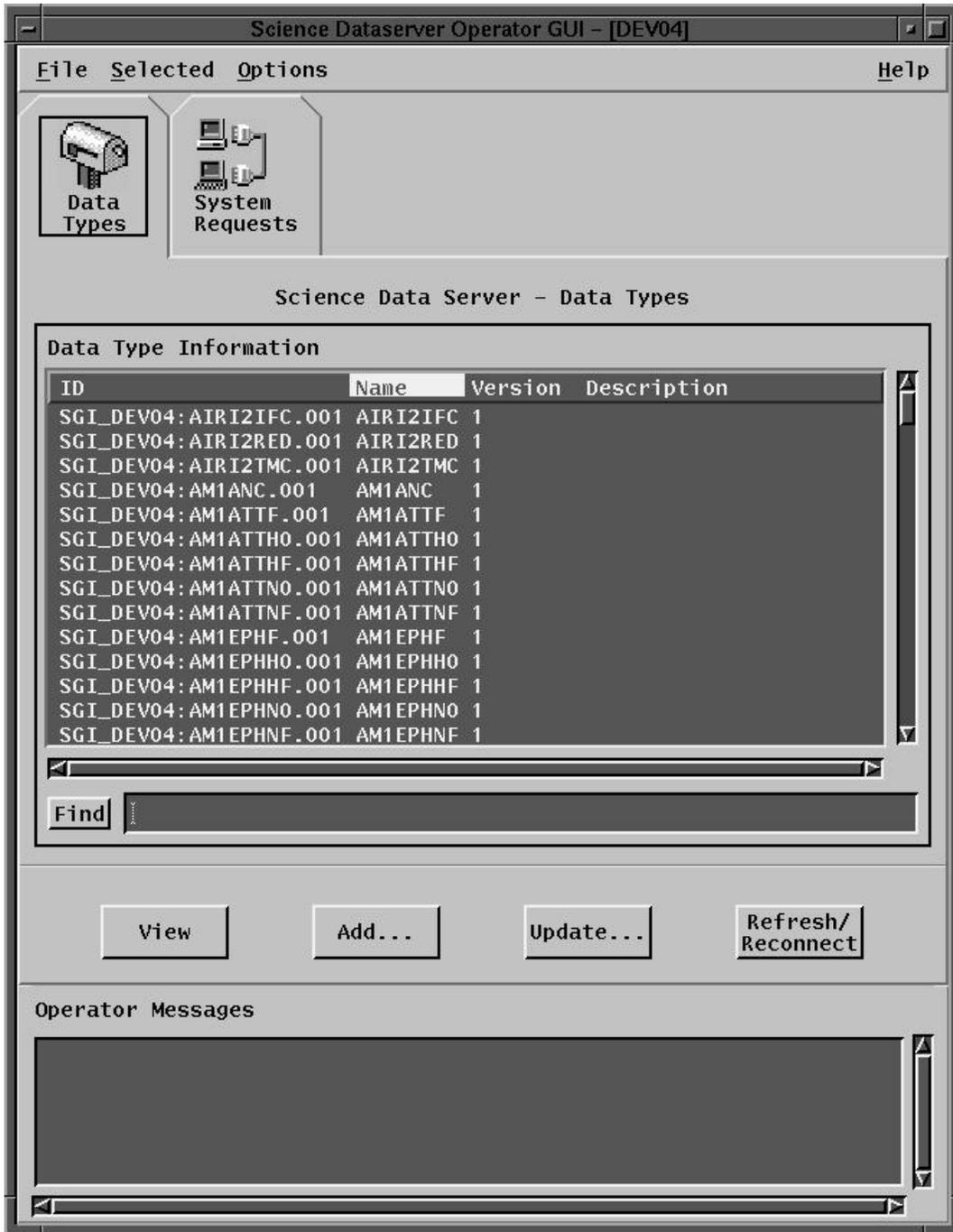


Figure 4.10.1-1. Science Data Server GUI Shown with Default Data Types Tab

The operator can select from the menu bar items at the top of the Science Data Server GUI window for getting help and activating special functions. The menu bar capability is available on all Science Data Server GUI screens. The following options are available through the toolbar:

File - which includes the following item:

Exit (Ctrl-Q) - Exit application (graceful exit)

Select - which includes the following items:

Duplicate - Not implemented

Select All - Not implemented

Deselect All - Deselects any items that were previously selected

Change Permissions - Not implemented

Options - This menu includes the *System Settings* item that opens the Server Polling Options window shown in Figure 4.10.1-2. Polling of the data server can be switched On/Off and the polling rate can be adjusted.

Help - which provides context sensitive help.

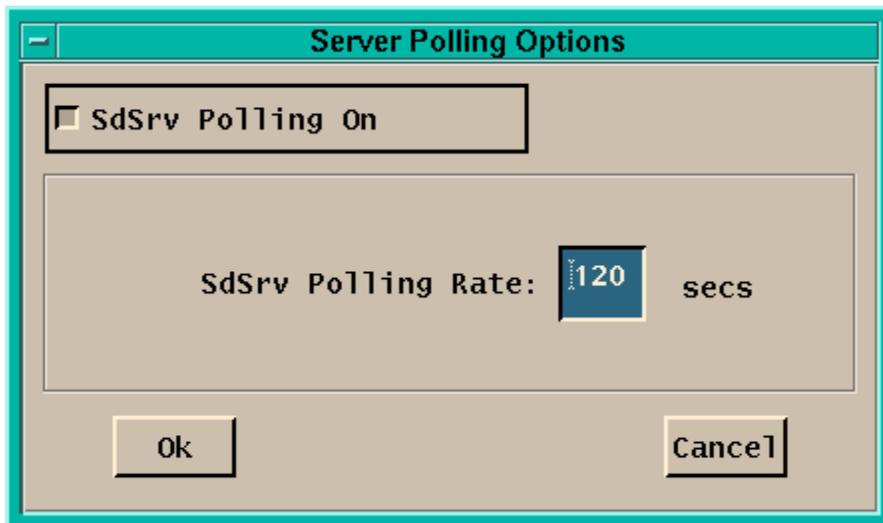


Figure 4.10.1-2. Science Data Server - Server Polling Options

Table 4.10.1-2 describes the fields used on the Server Polling Options window.

Table 4.10.1-2. Science Data Server - Server Polling Field Description

Field Name	Data Type	Size	Entry	Description
SdSrv Polling On	Button	N/A	Optional	Pressing this button on invokes automatic polling by the SDSRV at the time interval specified in the <i>SdSrv Polling Rate: ssss secs</i> field.
SdSrv Polling Rate	Integer	4 digits	Optional	Specify the rate at which the Science Data Server GUI is updated with data coming from the Data Server. The polling rate interval default is 120 seconds.
OK	Button	N/A	Optional	Pressing this button submits the polling rate setup for activation.
Cancel	Button	N/A	Optional	Cancels the submission of a new polling rate setup.

4.10.1.2.1 Data Types Tab

The Data Types Tab is the default screen of the Science Data Server GUI shown in Figure 4.10.1-1. This window provides operations personnel at the DAAC the capability to view descriptor information, add new ESDTs and update ESDTs. A list of currently installed ESDTs is shown containing the ESDT ID, name, version number, and a brief description of the structure and services available for the ESDT. Select the data type and click on the *View* button for additional information describing the structure, contents and services for each existing ESDT.

Table 4.10.1-3 describes the Science Data Server - Data Types fields.

Table 4.10.1-3. Science Data Server - Data Types Field Description

Field Name	Data Type	Size	Entry	Description
Data Type ID	Character	8	System generated	Uniquely identifies the specific type of ESDT.
Name	Character	25	System generated	Name of ESDT.
Version	Integer	3	System generated	Version number of the ESDT (assigned starting at 1).
Description	Character	255	System generated	Includes structure and services available for an ESDT.
Find	Character	255	Optional	This functionality is provided in order to help the user browsing very long ESDT lists.

In addition, the following buttons are provided:

- The **View** button on the Data Types sub-tab displays ESDT descriptor information (read-only) and it is associated dynamic Data Link Library (DLL) filename. Descriptor information consists of groups, objects, and keywords defining a ESDTs metadata, advertised services, subscribable events, data dictionary information, validation criteria, and science parameters. Descriptor information is necessary for the Science Data Server to properly configure itself to perform services related to an ESDT. A DLL is an executable library that is loaded dynamically when needed to fulfill ESDT services. The

Science Data Server - Descriptor Information Dialog (see Figure 4.10.1-3 below) provides the following buttons:

- **Close** exits the dialog
- **Help** displays on-line help information

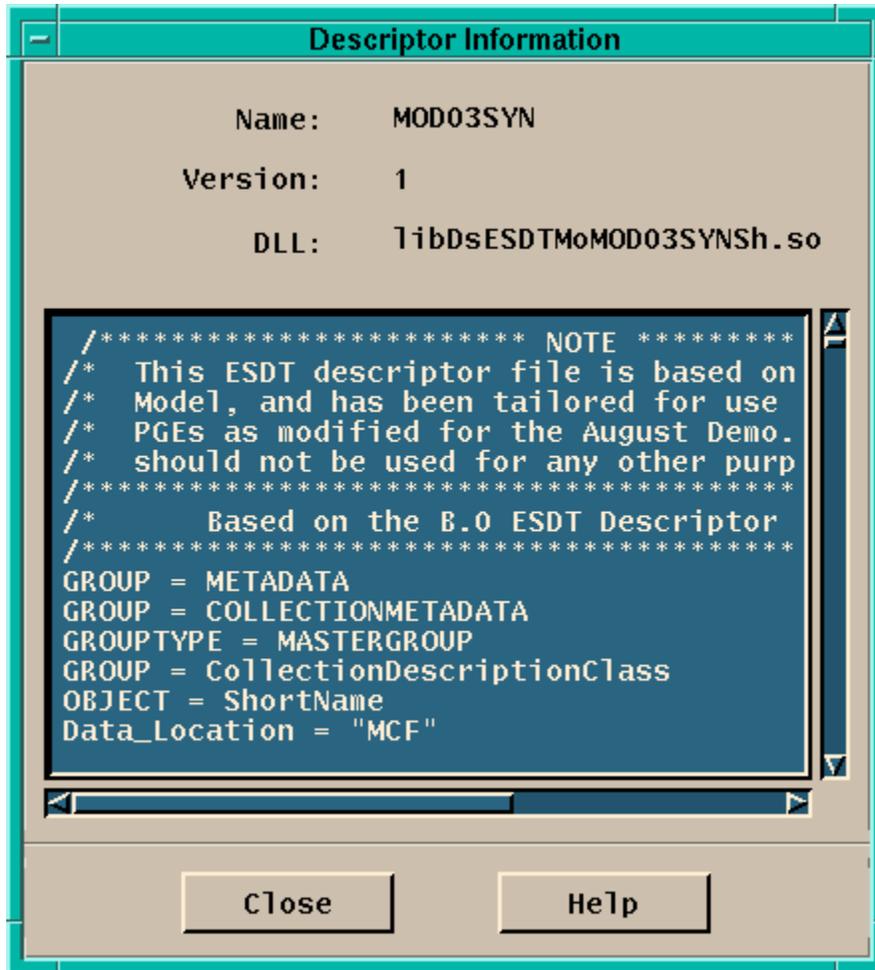


Figure 4.10.1-3. Science Data Server - Descriptor Information Dialog

- The **Add...** button on the Data Types tab shown in Figure 4.10.1-1 opens the Data Type Dialog shown in Figure 4.10.1-4. This window is used to add a new ESDT to the existing installed list of data types based upon input information. The SDSRV GUI has the capability to install multiple ESDTs. Click on the **File...** button to display a list of descriptor filenames to choose from instead of typing them in. Multiple descriptor files can be selected. Click the **OK** button to add the data type. If no error messages appear, the operation has been successfully completed. Click the **Cancel** button to close the dialog without performing an operation. Click the **Clear** button to start all over again the

process of filling in new information. Click the **Help** button to display on-line help information.

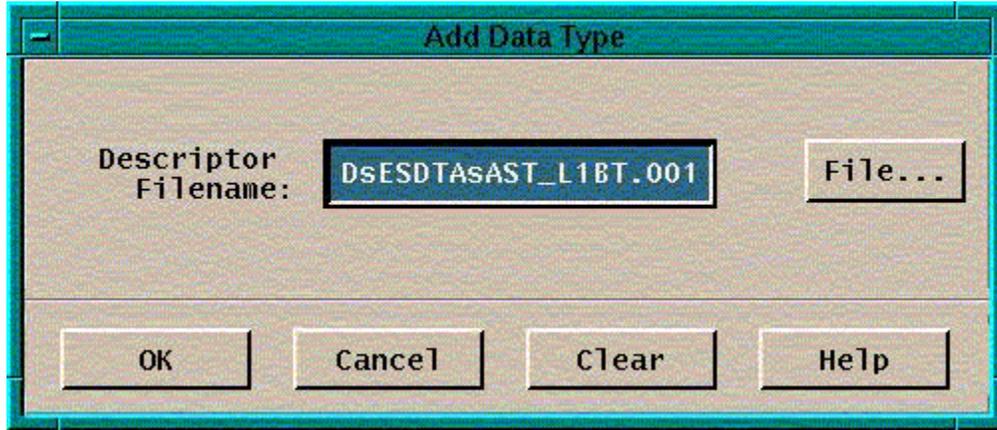


Figure 4.10.1-4. Science Data Server - Add Data Type Dialog

Table 4.10.1-4 describes the Science Data Server - Add Data Type Dialog fields.

Table 4.10.1-4. Science Data Server - Add Data Type Field Description

Field Name	Data Type	Size	Entry	Description
Descriptor Filename	Character string	25	Required	Name of an ASCII file containing the ESDT descriptor file.

- The **Update...** button on the Data Type tab in Figure 4.10.1-1 opens the Update ESDT Dialog shown in Figure 4.10.1-5. This window is used to update an ESDT to the installed list of data types based upon input information. The SDSRV GUI provides the capability to update multiple ESDTs at one time. The Science Data Server needs to be running in Maintenance mode to accept this operation. Click on the **File...** button to display a list of descriptor filenames to choose from instead of typing them in. Multiple descriptor files can be selected. Click the **OK** button to update the data type. If no error messages appear, then the operation has been successfully completed. Click the **Cancel** button to close the dialog without performing an operation. Click the **Clear** button to start all over again the process of filling in new information. Click the **Help** button to display on-line help information.



Figure 4.10.1-5. Science Data Server - Update Data Type Dialog

Table 4.10.1-5 describes the Science Data Server - Update Data Type fields.

Table 4.10.1-5. Science Data Server - Update Data Type Field Description

Field Name	Data Type	Size	Entry	Description
Descriptor Filename	Character string	255	Required	Name of an ASCII file containing the ESDT descriptor file.

- The **Refresh/Reconnect** button on the Data Types sub-tab of Figure 4.10.1-1 updates the data type information screen with current information.
- The **Operator Messages** field on the Data Types sub-tab of Figure 4.10.1-1 displays informational and error messages.

4.10.1.2.2 System Requests Tab

Clicking the **System Requests** tab brings up the System Management Requests window (see Figure 4.10.1-6). This window provides operations personnel at the DAAC the capability to monitor requests the Science Data Server is working with. All requests within the Science Data Server are displayed. Positioning the cursor and clicking on the appropriate column of interest sorts the columns of the list. Positioning the cursor and clicking on the Filter button and entering the attributes on which to filter can filter the requests.

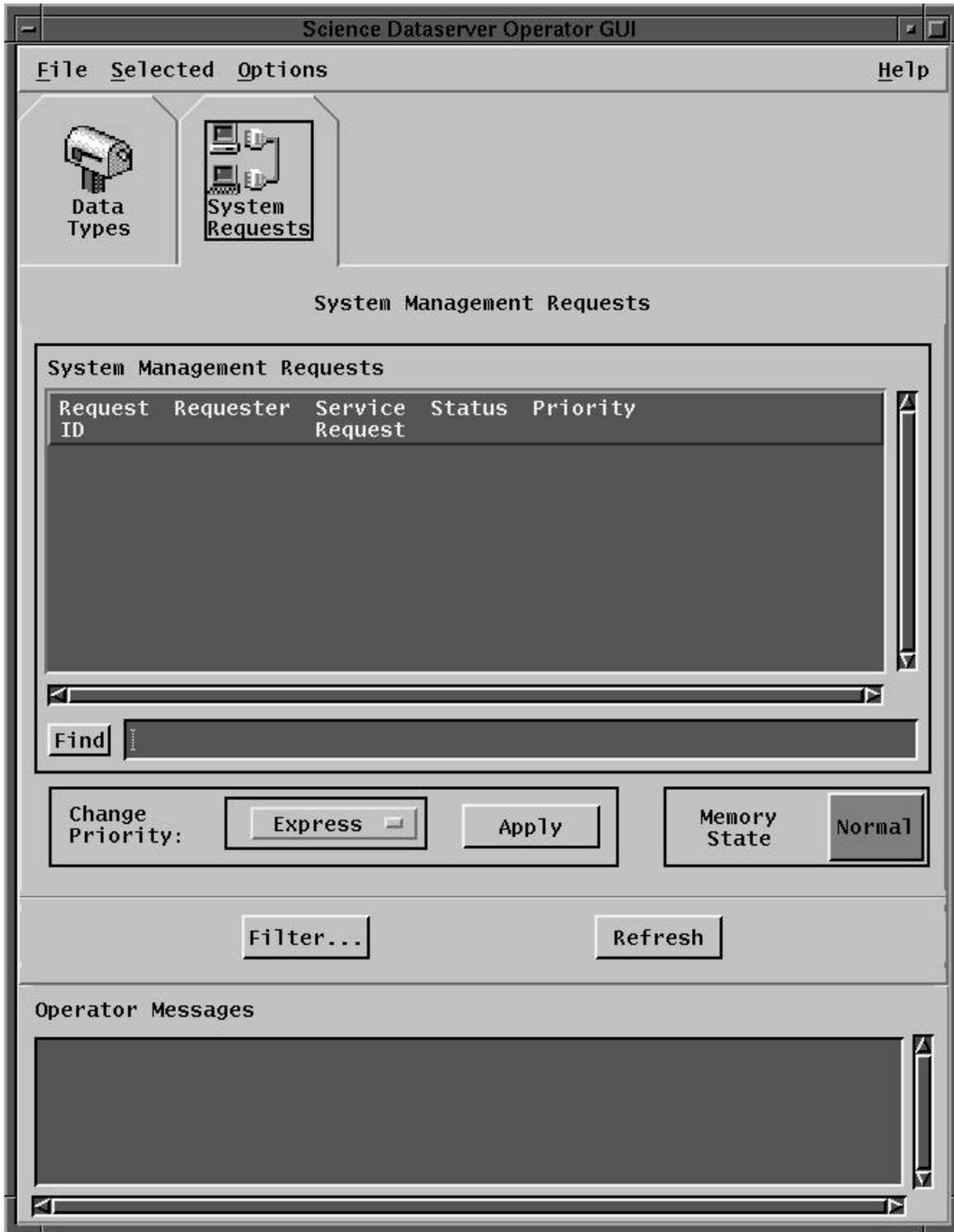


Figure 4.10.1-6. System Management Requests Window

Table 4.10.1-6 describes the System Management Requests Window fields.

Table 4.10.1-6. System Management Requests Field Description

Field Name	Data Type	Size	Entry	Description
Request ID	Character	255	System generated	Unique identifier for the request.
Requester	Variable character	100	System generated	Identifies the user that submitted the request.
Service Request	Character	25	System generated	Types of requests handled are Insert, Acquire, and Delete.
Status	Character	20	System generated	Possible states are Submitted, Queued, Executing, Failed_Retryable, Failed_Fatal, Failed_Unknown and Done.
Priority	Variable character	20	System generated	Priority of the data server system requests, i.e., Express, Very High, High, Normal (default), Low.
Find	Character	255	Optional	If the list is too long, this field can be used to search for an entry.

In addition, the following buttons are provided:

- **Change Priority:** allows the operator to change the priority of each selected request through a pull down menu. Possible values are: Express, Very High, High, Normal (default) and Low.
- **Apply** allows the operator to commit to the priority change selected through the change priority button.
- **Filter...** (See Figure 4.10.1-7) brings up the System Management Filter Requests dialog, which provides a selection of attributes on which to filter for the list of System-wide requests. Filter on system management requests by entering the desired information, then clicking on the Request ID or Requester radio button for the desired attribute. Return to the original list of requests by clicking on the All Requests radio button. Click on other filters associated with State and Priority by clicking on the toggle button. Filter on every attribute associated with a category by clicking the **All** button or clear a category of filters by clicking on the **None** button.
- **Memory State** monitors the current memory state of the data server in regards to values that are set on the server side through configuration parameters. Possible values are: Normal (green), Low (yellow), Very Low (red) and Disabled (gray). The Memory State shows a Disabled state when this functionality has been turned off (by setting the DSSMEMORYMONITORDISABLEFLAG in the data server's configuration file).
- **Refresh** button updates the System Management Requests screen with current information.
- **Operator Messages** is an area where operators can input text related to requests.

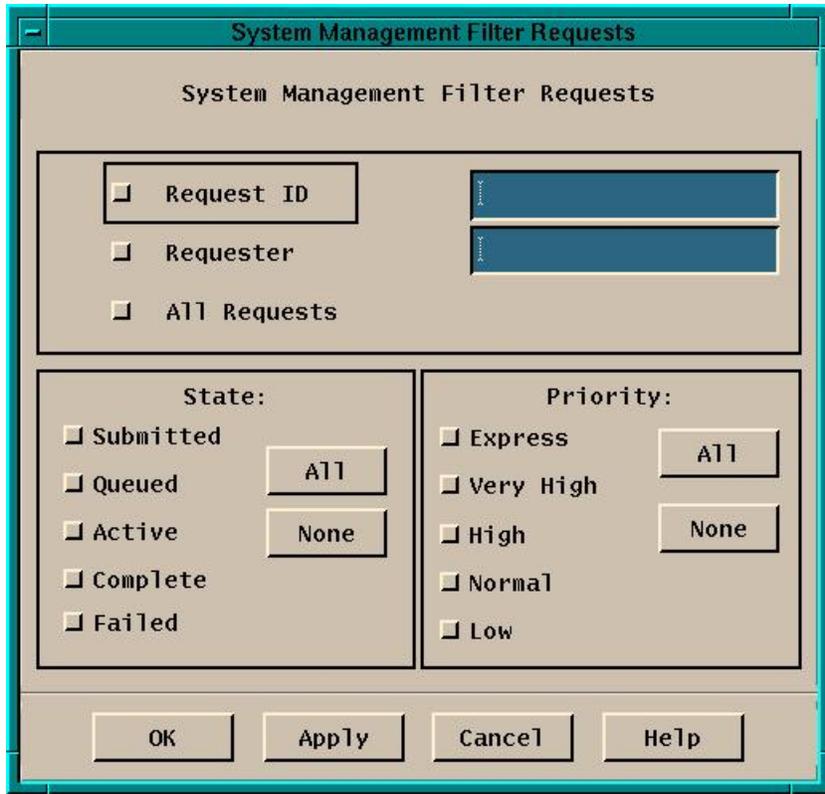


Figure 4.10.1-7. System Management Filter Requests Dialog

Table 4.10.1-7 describes the System Management Filter Requests Dialog fields.

Table 4.10.1-7. System Management Filter Requests Field Description

Field Name	Data Type	Size	Entry	Description
Request ID	Character	255	System generated	Unique identifier for the request.
Requester	Variable character	100	System generated	Identifies the user submitting the request.

In addition, the following buttons are provided:

- **OK** implements filter criteria, and the dialog closes
- **Apply** implements filter criteria, and the dialog remains open for additional filtering
- **Cancel** closes the dialog without saving
- **Help** displays on-line help information
- Back to the System Requests tab description (Figure 4.10.1-6), **Operator Messages** provides informational and error messages to the DAAC Operator

- **Refresh** causes the Data Server to be polled for an update on Requests

4.10.1.3 Required Operating Environment

For information on the operating environment, tunable parameters, and environment variables refer to the 920-TDA-022 “Custom Code Configuration Parameters” documentation series.

4.10.1.3.1 Interfaces and Data Types

Table 4.10.1-8 lists the supporting products that this tool depends upon in order to function properly.

Table 4.10.1-8. Interface Protocols

Product Dependency	Protocols Used	Comments
SDSRV and all clients	Socket	Via client libraries
SDSRV GUIs	X-11	Via client libraries

4.10.1.4 Databases

The Science Data Server GUI does not include the direct managing of any database. It has an interface with the Science Data Server Data Base: however this interface is based on a simple parameter passing function. For further information of the Science Data Server Data Base refer to 311-CD-624, *Science Data Server Database Design and Schema Specifications for the ECS Project*.

4.10.1.5 Special Constraints

The Science Data Server GUI runs only if the Science Data Server is running in the background. Note also that at the moment the Science Data Server GUI is started through a command line that specifies the configuration file that is used to initialize the GUI Application.

4.10.1.5.1 Deleting an ESDT

The Science Data Server GUI does not provide a means of deleting an ESDT. A script command is provided for this purpose and is invoked as follows:

```
>EcDsSrRemesdt <mode> <descriptor file name> where:
<mode> is the operating mode affected (e.g., OPS, TS1 or TS2)
<descriptor file name> is the name of the ESDT's descriptor file
```

4.10.1.6 Outputs

There is no processing associated with the operation of this GUI. The information provided to the operator are retrieved from the Data Server Database described in Section 4.10.1.4 and displayed through the screens discussed in Section 4.10.1.2 and the related sub-sections.

4.10.1.7 Event and Error Messages

Both event and error messages are listed in Appendix A.

4.10.1.8 Reports

This tool produces no reports.

4.10.2 Storage Management Control GUI

This tool enables operations to manage various data storage functions. These functions include the capability to set and modify configurations of various server types (e.g., 8mm tape, Archive Server and DTF tape), manage data location within the archive and on disk, configure stacker slots, display storage events which possibly require operator actions, and view backup failures with the capability to restart a backup operation.

The tool is used to perform the operator functions listed in Table 4.10.2-1.

Table 4.10.2-1. Common ECS Operator Functions Performed with STMGT Control

Operating Function	Command/Script or GUI	Description	When and Why to Use
Configure Servers and Devices	Storage Config Tab	Allows the operator to organize and configure various Server and any associated disk and hardware resources.	As needed to add, delete, or modify the configuration of a Server or hardware resource.
Configure Volume Groups	Vol. Grp. Config. Tab	Allows the operator to organize and configure the data flow in and out of various archives.	As needed to configure archive repositories for ESDTs.
Monitor Media Resources	Resource Mngmnt. Tab	Allows the operator to monitor and set resource availability for media distribution.	As needed to make available or take off-line specific resources.
View Cache Statistics	Cache Stats. Tab	Allows the operator to view the contents and usage statistics for a Read-Only Cache or the Pull Area. Operator can force expiration and cleanup of liens against cached files.	Used when warning is displayed in the message area informing the operator the cache is getting too full.
Search Event Log	Storage Events Tab	Allows the operator to find events from selected parameters.	As needed to locate events.
Monitor Server Operations	Request Status Tab	Allows the operator to monitor the progress of requests throughout the STMGT CI.	As needed to monitor processing (e.g., to detect stalled requests or servers, which appear to be idle).

4.10.2.1 Quick Start Using the Storage Management Control GUI

The Storage Management Control GUI provides the user with the capability to control the following STMGT components:

1. **Storage Configuration (Storage Config.)**
2. **Volume Group Configuration (Vol Grp Config.)**
3. **Resource Management (Resource Mngmnt.)**
4. **Cache Monitoring (Cache Stats.)**
5. **Storage Events (Storage Events)**
6. **Server Monitoring (Request Status)**

To invoke the Storage Management Control GUI, enter the following command line:

```
>EcDsStmgtGuiStart <mode>....
```

where:

<mode> corresponds to the system mode in which to operate (e.g., OPS, TS1 or TS2). Note that the Unix “**DISPLAY**” environment variable must be properly set for the GUI to display on your screen.

4.10.2.2 The Storage Management Control Operator Main Screen

The Storage Management (STMGT) Control Main Screen, shown in Figure 4.10.2-1, consists of 6 tabs, which provide user control of the components listed above. The operator can select from the following toolbar menu items located at the top of the window for each component:

File contains the exit command to close the application

Options allows the operator to set polling rates for various displays within the GUI

Backup allows the operator to recover from failures to store backup copies of ingested data

Delete allows the operator to remove data

Help provides context sensitive help to the operator

The STMGT Control Main Screen also contains an **Operator Messages** list box to display error conditions.

4.10.2.2.1 Storage Configuration Tab

The Storage Configuration Tab, shown in Figure 4.10.2-1, displays server identification information, and allows the operator to view and control the server configuration including adding, deleting and modifying server instances and their associated disk and hardware resources.

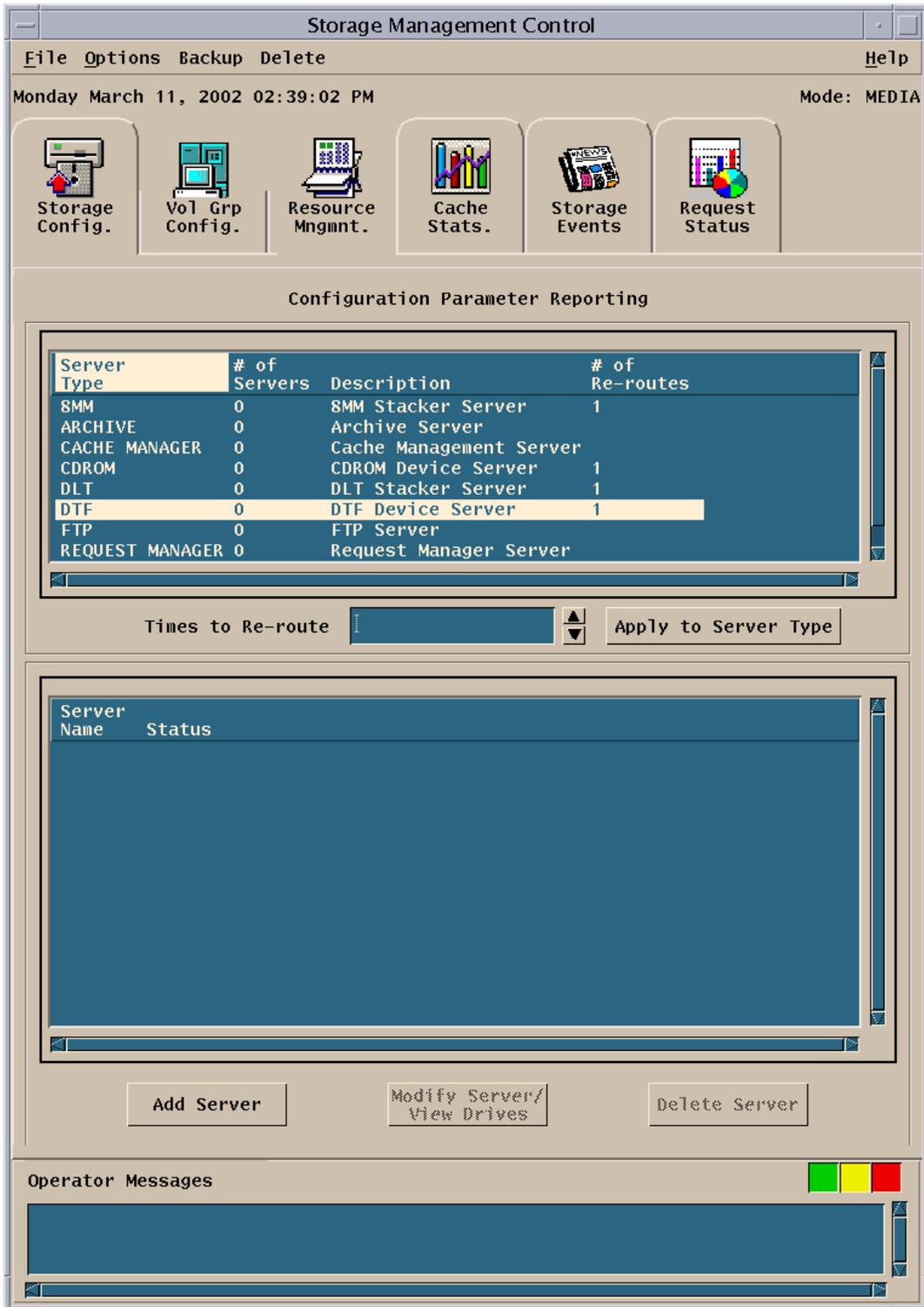


Figure 4.10.2-1. Storage Management Server Main Screen with the Default Storage Configuration Tab Displayed (Storage Config.)

This includes servers to both distribute data and archive data. Activating a separate window and inserting a set of parameters for that type of server can configure each specific server.

Selecting a Server Type brings up a list of configured servers in the list box below. Table 4.10.2-2 describes the fields shown on the Storage Configuration tab.

Table 4.10.2-2. Storage Management Server Field Description

Field Name	Data Type	Size	Entry	Description
Server Type	Character	20	Preloaded	Type of server to configure.
# of Servers	Integer	N/A	Calculated	Number of Server instances configured for this Server Type.
Description	Character	100	Preloaded	Brief description of Server Type.
# of Re-routes	Integer	3	Required	For media server types, the maximum number of times a request can be re-routed to an alternate instance of a media server before the request is failed.
Times to Re-route	Character	3	Required	Number of times to re-route (see above). The user can change the value by modifying this field and pressing the Apply to Server Type button.
Server Name	Character	20	System generated	Name, which uniquely identifies the server instance within the mode.
Status	Character	N/A	System generated	Status of the server, either up or down.
Operator Messages	Character	N/A	System generated	System will generate any necessary messages for the operator to read

The Storage Configuration Tab includes several buttons. These buttons are: **Apply to Server Type**, **Add Server**, **Modify Server/View Drives** and **Delete Server**. (Note: When a media server is selected, the **Modify Server/View Drives** button is automatically re-labeled as **Modify Server/View Stacks** or **Modify Server/View Devices**.) The functionality associated with these buttons is discussed below along with the pop-up windows that are displayed when the buttons are depressed.

Apply to Server Type updates the number of re-routes configured for that media type.

Add Server allows the operator to add a server instance of the selected Server Type to the configuration. A screen appears, which allows the operator to configure the parameters and resources for the new server instance. Different Server Types have different screens to address the particular configuration of that Server Type.

Modify Server allows the operator to change the parameter settings or resources associated with the selected server. For media servers, this includes stacker and device resources. The screen appearing is identical to the screen used for **Add Server** for the same Server Type. These screens are discussed in the following sections.

Delete Server allows the operator to remove configuration of a selected server instance. This removes all stored configuration information associated with parameters and configured

resources for the deleted server. Operators are not permitted to delete servers that are running; the executing server instance must be brought down before it can be deleted.

All server configuration windows provide **OK** and **Cancel** buttons. Pressing the **OK** button saves the server configuration information to the database and closes the server configuration window. **Cancel** closes the server configuration window without saving the configuration information to the database.

All server configuration windows also provide a **Service Threads** field with an associated **Allocate by Priority** button. The **Service Threads** field specifies the number of worker threads, which is allocated within the server instance to process requests. By clicking on the **Allocate by Priority** button, these threads can be allocated by priority in order to reserve certain resources for higher priority requests. Note that lower-priority threads can be used to service higher priority requests, but never vice versa. Figure 4.10.2-2 illustrates the window that is presented when clicking on any **Allocate by Priority** button. All fields refer to the number of threads allocated to service requests of the indicated priority (or higher). By default, all service threads are created as low priority service threads, since these can be pre-empted by any priority request. So that the total of the numbers in each of the 5 different thread type fields equals the number in the Total field, the number of low threads is automatically re-calculated whenever the number of any of the other thread types is changed.

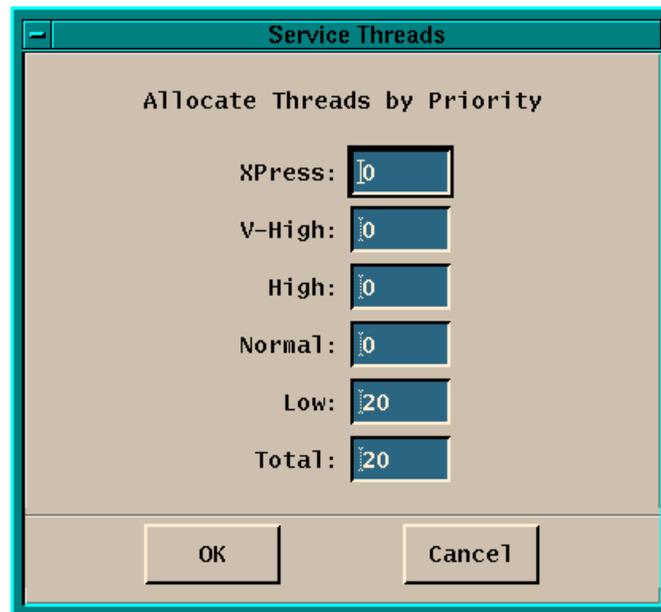


Figure 4.10.2-2. Service Thread Configuration Window

4.10.2.2.1.1 Request Manager Server Configuration

Figure 4.10.2-3 displays the server configuration window for configuring instances of the Request Manager Server.

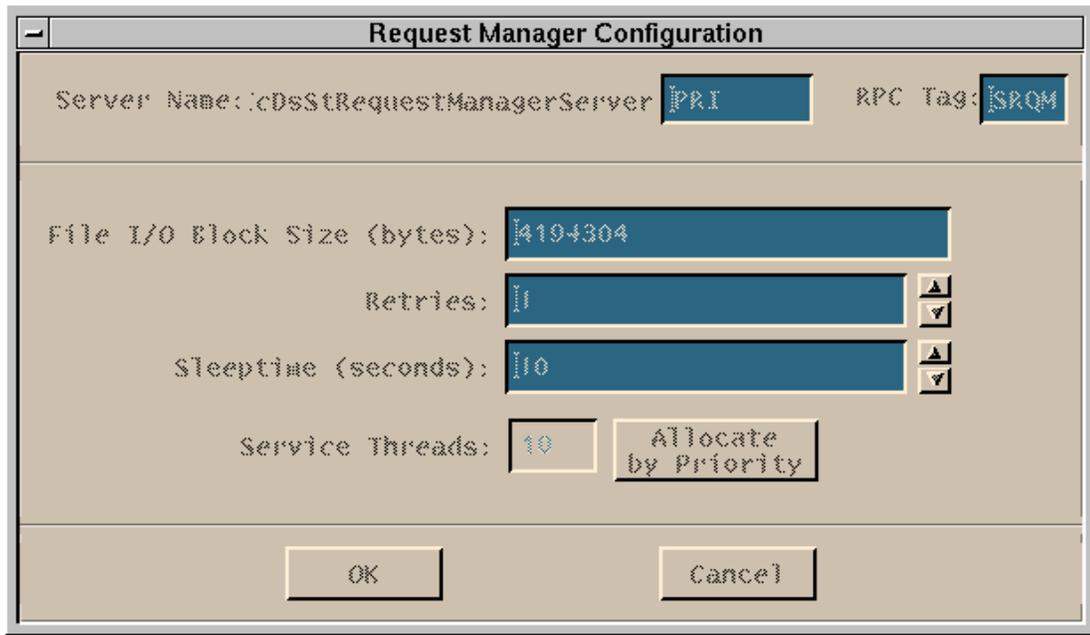


Figure 4.10.2-3. Request Manager Configuration Window

Table 4.10.2-3 describes the fields shown on the Request Manager Configuration pop-up window.

Table 4.10.2-3. Request Manager Configuration Window Field Descriptions

Field Name	Data Type	Size	Entry	Description
Server Name	Character	6	Required	Name of this instance of the Request Manager Server. The only names which are recognized by the STMGT client interface are PRI and SEC.
RPC Tag	Character	4	Required	The four-character tag, which appears in RPC Ids submitted by the Request Manager Server. Recommended values are RQMA and RQMB.
File I/O Block Size (bytes)	Integer	16	Required	The block size to be used for file I/O.
Retries	Integer	3	Required	The number of times to retry failed operations.
Sleeptime (seconds)	Integer	5	Required	The time to sleep, in seconds, between retries.
Service Threads	Integer	5	Required	The number of Service Threads available to process requests submitted to the Request Manager Server.

By clicking on the **Allocate by Priority** button, these threads can be allocated by priority in order to reserve certain resources for higher priority requests as shown in Figure 4.10.2-2. Clicking the **OK** button allows the operator to accept the entries specified for the Request Manager Configuration. Clicking the **Cancel** button allows the operator to terminate their request.

4.10.2.2.1.2 Staging Disk Server Configuration

Figure 4.10.2-4 displays the server configuration window for configuring instances of the Staging Disk Server.

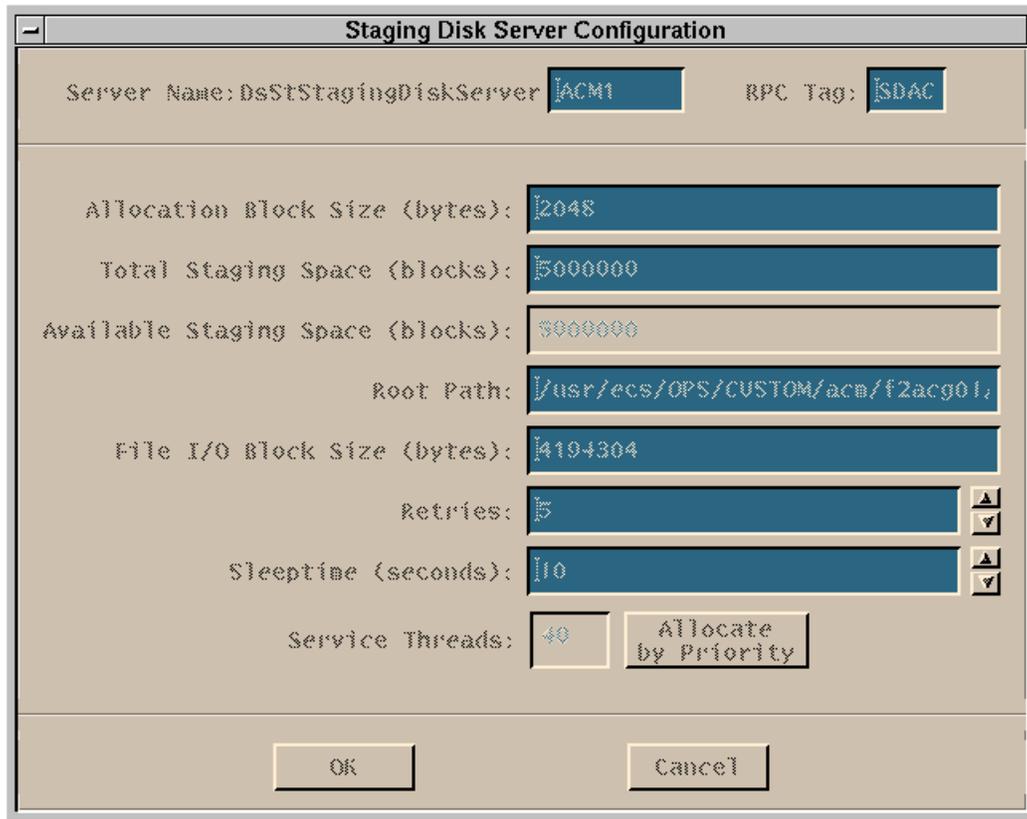


Figure 4.10.2-4. Staging Disk Server Configuration Window

Table 4.10.2-4 describes the fields shown on the Staging Disk Server Configuration pop-up.

Table 4.10.2-4. Staging Disk Server Configuration Window Field Descriptions (1 of 2)

Field Name	Data Type	Size	Entry	Description
Server Name	Character	6	Required	Name of this instance of the Staging Disk Server.
RPC Tag	Character	4	Required	The four-character tag, which appears in RPC Ids submitted by the Staging Disk Server. Recommended values are STDA, STDB, etc.
Allocation Block Size (bytes)	Integer	16	Required	Block size in bytes to be used for allocation purposes. Clients, which must accommodate links to other files, allocate one block per link to be created.

**Table 4.10.2-4. Staging Disk Server Configuration Window Field Descriptions
(2 of 2)**

Field Name	Data Type	Size	Entry	Description
Total Staging Space (blocks)	Integer	16	Required	Total space available on disk for allocation. The units used are allocation blocks.
Available Staging Space (blocks)	Integer	16	System generated	Space remaining, which is available for allocation. The units used are allocation blocks. This is calculated based on the Total Staging Space and the current set of allocated staging disks as stored in the database.
Root Path	Character	250	Required	Path on disk where staging disks are created. Each staging disk is named <i>disknnn</i> , where <i>nnn</i> is a unique number.
File I/O Block Size (bytes)	Integer	16	Required	The block size to be used for file input or output (I/O).
Retries	Integer	3	Required	The number of times to retry failed operations.
Sleeptime (seconds)	Integer	5	Required	The time to sleep, in seconds, between retries.
Service Threads	Integer	5	Required	The number of Service Threads available to process requests submitted to the Request Manager Server.

By clicking on the **Allocate by Priority** button, these threads can be allocated by priority in order to reserve certain resources for higher priority requests. Clicking the **OK** button allows the operator to accept the entries specified for the Staging Disk Server Configuration. Clicking the **Cancel** button allows the operator to terminate their request.

4.10.2.2.1.3 Cache Manager Configuration

Cache Manager Server instances can be configured to manage Read-Only Caches or to manage the Pull Area. The Pull Area Manager toggle controls the type of cache being managed. If **Pull Area Manager** is toggled to **No**, the configured instance is defined to manage a Read-Only Cache. Otherwise, the configuration defines the cache information to be used by the Pull Monitor Server. Figure 4.10.2-5 displays the server configuration window for configuring instances of the Cache Manager Server. Note that the last three fields are grayed out. These fields are enabled only when configuring the Pull Area.

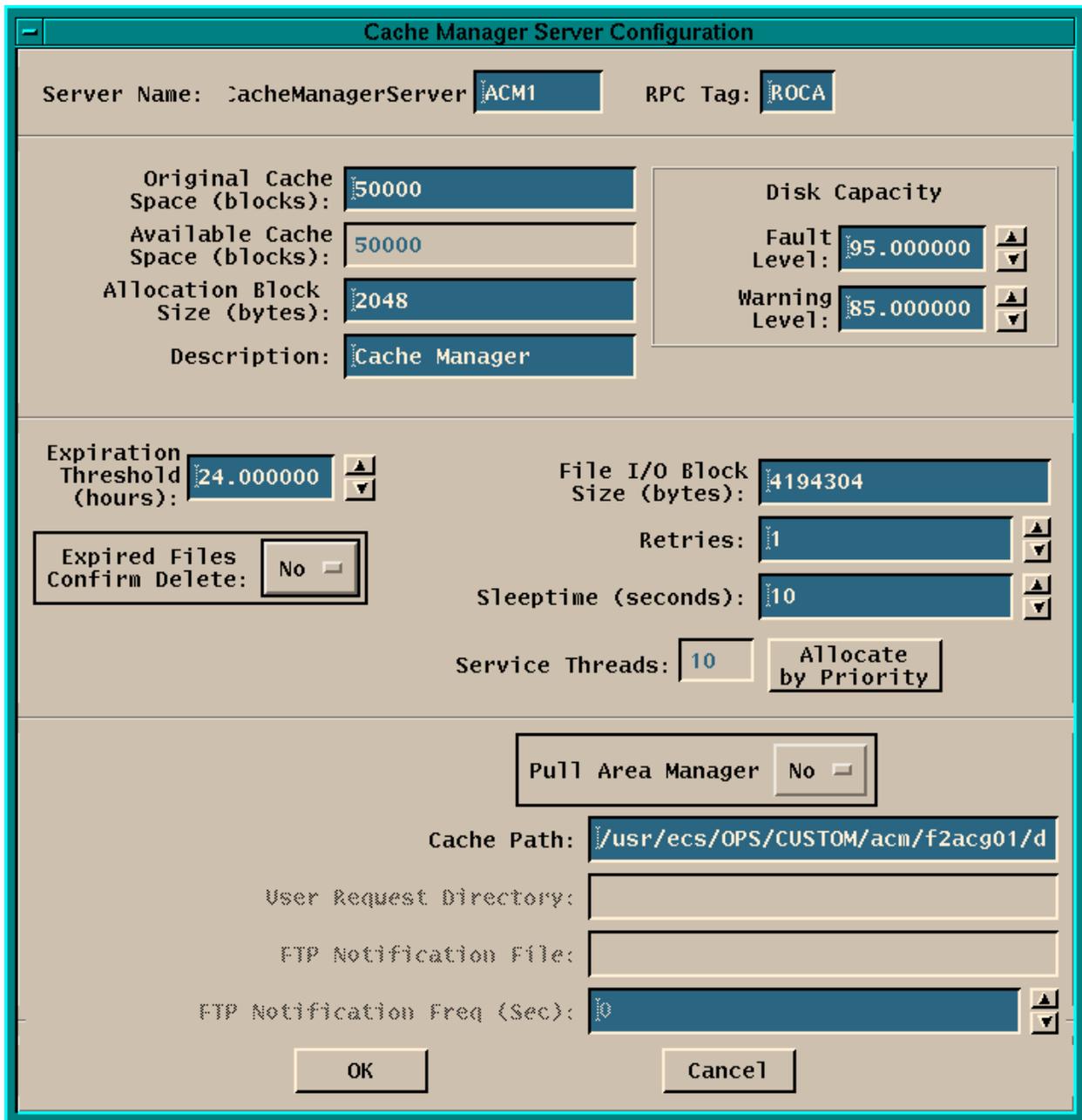


Figure 4.10.2-5. Cache Manager Server Configuration Window

Table 4.10.2-5 describes the fields shown on the Cache Manager Server Configuration pop-up window.

**Table 4.10.2-5. Cache Manager Server Configuration Window Field Descriptions
(1 of 3)**

Field Name	Data Type	Size	Entry	Description
Server Name	Character	6	Required	Name of this instance of the Cache Manager Server. Recommended value for the Cache Manager instance, which manages the Pull Area is PULL.
RPC Tag	Character	4	Required	The four-character tag, which appears in RPC Ids submitted by the Cache Manager Server. Recommended values are ROCA, ROCB, etc. for Cache Manager instances, which manage Read-Only Caches and PULL for the Cache Manager instance which manages the Pull Area.
Original Cache Space (blocks)	Integer	16	Required	Total size of the cache on disk. The units used are allocation blocks.
Available Cache Space (blocks)	Integer	16	System generated	Free space remaining in the cache. The units used are allocation blocks. This is calculated based on the Original Cache Space less the combined size of the files currently cached as stored in the database.
Allocation Block Size (bytes)	Integer	16	Required	Block size in bytes to be used for cache allocation purposes.
Description	Character	100	Optional	A mnemonic description of the cache for use by the operator.
Disk Capacity – Fault Level	Real	16	Required	The level at which a message is sent to the operator via the STMGT GUI to indicate that the cache is critically full. Unless Expired Files Confirm Delete is set to Yes, this should be set to 99.5%, since the Cache Manager automatically cleans up files as needed to make space.
Disk Capacity – Warning Level	Real	16	Required	The level at which a message is sent to the operator via the STMGT GUI to indicate the cache is becoming full. Unless Expired Files Confirm Delete is set to Yes, this should be set to 99.5%, since the Cache Manager automatically cleans up files as needed to make space.

**Table 4.10.2-5. Cache Manager Server Configuration Window Field Descriptions
(2 of 3)**

Field Name	Data Type	Size	Entry	Description
Expiration Threshold (hours)	Real	16	Required	The number of hours a lien is held against a cached file. A lien can be either an expressed intent to copy or link the file from the cache or a physical link to the file from another area (e.g., a staging disk). If the lien expires and space is required in the cache, the lien is removed automatically unless the Expired Files Confirm Delete flag is set to Yes. For normal operations, this field should be set to at least 24 hours.
Expired Files Confirm Delete	Boolean	N/A	Required	If set to No, the Cache Manager automatically makes space for new files in the cache by removing files whose liens have expired. If this flag is set to Yes, the operator must confirm the removal of expired liens before the liens can be deleted. (See the Cache Statistics tab later in this section for information on deleting expired liens.)
File I/O Block Size (bytes)	Integer	16	Required	The block size used for file I/O.
Retries	Integer	16	Required	The number of times to retry failed operations.
Sleeptime (seconds)	Integer	16	Required	The time to sleep, in seconds, between retries.
Service Threads	Integer	16	Required	The number of Service Threads available to process requests submitted to the Staging Disk Server.
Pull Area Manager	Boolean	N/A	Required	Indicates whether this instance of the Cache Manager manages a Read-Only Cache (ROC) or the Pull Area. Only one instance of the Cache Manager should be configured as a Pull Area Manager, though multiple instances can be configured to manage ROCs.
Cache Path	Character	250	Required	Path on disk where the cache physically resides.

**Table 4.10.2-5. Cache Manager Server Configuration Window Field Descriptions
(3 of 3)**

Field Name	Data Type	Size	Entry	Description
User Request Directory	Character	250	Required if Pull Area Manager is Yes	If this instance has been flagged as a Pull Area Manager, the User Request Directory identifies the path on disk where user directories are created. These directories are the visible paths, which are used for staging FTP Pull distributions. The path specified here must match the login path specified in the configuration for anonymous FTP, if anonymous FTP is used for client acquisition of FTP Pull orders.
FTP Notification File	Character	250	Required if Pull Area Manager is Yes	A fully qualified path and filename where a list of acquired files are placed to permit the Cache Manager to automatically clean up files, which have been successfully pulled. The Pull Area Manager only uses this field.
FTP Notification Freq (Sec)	Integer	16	Required if Pull Area Manager is Yes	The frequency with which the Pull Area Manager reviews the list of acquired files and performs the corresponding clean up.

4.10.2.2.1.4 FTP Server Configuration

Figure 4.10.2-6 is the FTP server configuration window for configuring instances of the FTP Server.

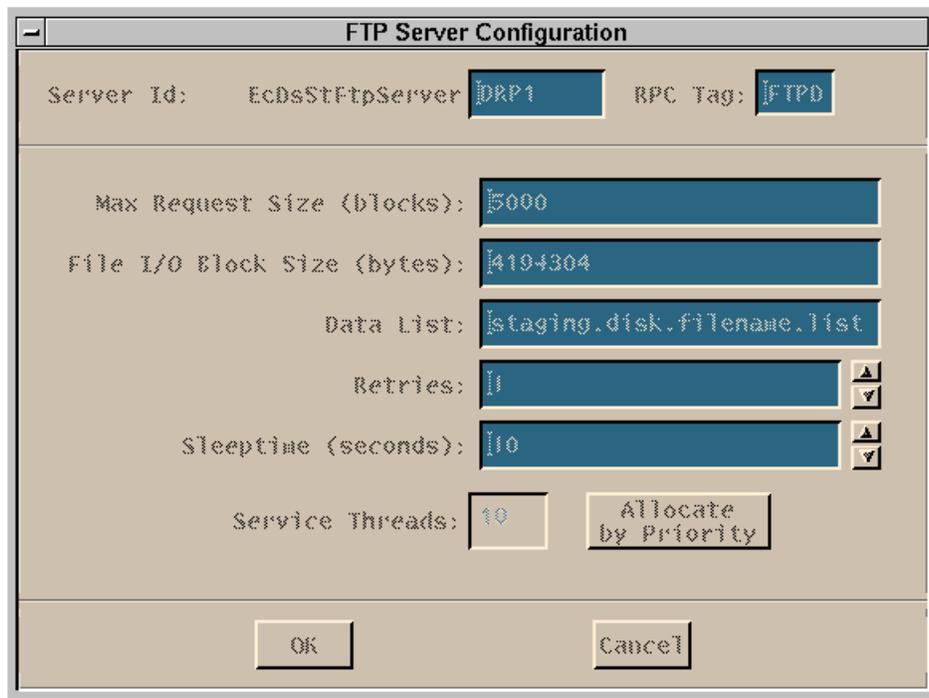


Figure 4.10.2-6. FTP Server Configuration Window

Table 4.10.2-6 describes the fields shown on the FTP Server Configuration pop-up.

Table 4.10.2-6. FTP Server Configuration Window Field Descriptions (1 of 2)

Field Name	Data Type	Size	Entry	Description
Server Id	Character	6	Required	Name of this instance of the FTP Server. The ID of the FTP server for external FTP distributions should be configured as NONE.
RPC Tag	Character	4	Required	The four-character tag, which appears in RPC Ids submitted by the FTP Server. Recommended values are FTPA, FTPB, etc.
Max Request Size (blocks)	Integer	16	Required	The maximum aggregate size of files, which can be electronically transferred as a single request. The size is specified in terms of file I/O blocks.
File I/O Block Size (blocks)	Integer	16	Required	The block size to be used for file input/output (I/O).

Table 4.10.2-6. FTP Server Configuration Field Description (2 of 2)

Field Name	Data Type	Size	Entry	Description
Data List	Character	100	Required	The name of the file, which is used by DDIST to specify the list of files to be electronically distributed. This must match the corresponding parameter in the DDIST configuration.
Retries	Integer	16	Required	The number of times to retry failed operations.
Sleeptime (seconds)	Integer	16	Required	The time to sleep, in seconds, between retries.
Service Threads	Integer	16	Required	The number of Service Threads available to process requests submitted to the FTP Server.

4.10.2.2.1.5 8MM and DLT Server Configuration

Figure 4.10.2-7 is the configuration screen for media servers that support stackers. This screen is used to configure instances of the 8MM Server and the Digital Linear Tape (DLT) Server. Table 4.10.2-7 describes the field information related to this pop-up window.



Figure 4.10.2-7. Media Server Configuration Window (Stacker-based)

**Table 4.10.2-7. Media Server Configuration Window Field Descriptions
(Stacker-based) (1 of 2)**

Field Name	Data Type	Size	Entry	Description
Server Name	Character	6	Required	Name of this instance of the Media Server.
RPC Tag	Character	4	Required	The four-character tag, which appears in RPC Ids submitted by the Media Server. Recommended values for 8MM servers are ATEA, ATEB, etc., and for the DLT server are DLTA, DLTB, etc.
Media Server Block Size (bytes)	Integer	16	Required	The block size used for tape input/output (I/O). This block size is also the basis unit for specifying media capacity.
Capacity (blocks)	Integer	16	Required	The maximum aggregate size of files, which can be transferred on a single piece of media. The size is specified in terms of Media Server Block Size.
Default Block Factor	Integer	16	Required	The number of blocks, which are read/written at once when performing I/O.
File I/O Block Size (bytes)	Integer	16	Required	The block size used for file I/O.
Retries	Integer	16	Required	The number of times to retry failed operations.
Sleeptime (seconds)	Integer	16	Required	The time to sleep, in seconds, between retries.
Number of Columns	Integer	16	Required	The column width to which packing slips should be formatted.
Number of Rows	Integer	16	Required	The page length to which packing slips should be formatted.
Print Queue	Character	64	Required	The Unix named print queue to which packing slips should be sent for requests serviced by this Media Server instance.
Allow Distribution Across Network	Boolean	N/A	Required	Indicates whether or not this Media Server instance permits media distribution from a staging disk, which is NFS-mounted in the event the data cannot be locally staged.

**Table 4.10.2-7. Media Server Configuration Window Field Description
(Stacker-based) (2 of 2)**

Field Name	Data Type	Size	Entry	Description
Staging Threads	Integer	16	Required	The number of Service Threads available to locally stage data for media distribution by this Media Server instance. It also does not correspond to the number of devices available for media distribution. A separate thread pool is automatically created for actual media I/O, based on the number of media devices configured for use by this Media Server instance.
Stacker ID	Character	64	Required	A unique identifier that identifies a stacker associated with this Media Server instance.
Description	Character	100	Optional	Additional mnemonic information provided by the operator during configuration to assist in identifying the listed stacker.

The Media Server Configuration window includes two buttons not found on other server configuration windows: **Allow Distribution Across Network** and **Save New Server**.

Allow Distribution Across Network Toggles whether or not the server instance permits distribution to media from an NFS-mounted directory. Some media types (e.g. CD-ROM) required a sustained input/output (I/O) level, which cannot be guaranteed when I/O is being done across a network. For such media types, this flag should be toggled to **No**.

Save New Server This button forces the displayed configuration information to be added to the database without closing the server configuration window. It must be pressed prior to adding new stackers.

The Media Server Configuration window also includes a pull-down menu – Stacker Operations – with the following options:

- **Add Stacker...** Allows the operator to add and configure a new stacker resource. A list of pre-configured stacker models is shown when this menu option is selected. The operator can select from the list of pre-configured stacker models, or select the **Unknown** option. In either case, the window shown below in Figure 4.10.2-8 is presented. If a pre-configured model is selected, the fields in the Stacker Configuration window are pre-populated with the appropriate values for the selected model.
- **Modify Stacker/View Drives** If a stacker is selected, this option permits the operator to modify the configuration of the selected stacker resource. This also brings up the window shown in Figure 4.10.2-8.
- **Delete Stacker** If a stacker is selected, this option permits the operator to remove the selected stacker as a configured resource for this server instance. Note that

removing a stacker automatically removes all configured devices associated with that stacker.

Whether adding or modifying configuration for a stacker resource, the window shown below in Figure 4.10.2-8 is presented. (Note: When adding a stacker, the **Save Modified Stacker** button is re-labeled **Save New Stacker**).

BMM Tape Stacker Configuration

Server Name:

Stacker ID:

Stacker Model:

Description:

Stacker Path:

Element Number: ▲ ▼

Stacker Number: Barcode Enabled

Number Of Slots: ▲ Fixed Slot 0 automatically
▼ created - Do not include in count.

Drive Operations

Drive Name	Description
CRW31-2	Tape Drive

Find

OK Save Modified Stacker Cancel

Figure 4.10.2-8. Stacker Configuration Window

Table 4.10.2-8 provides a detailed description of the Stacker Configuration window fields.

Table 4.10.2-8. Stacker Configuration Window Field Descriptions

Field Name	Data Type	Size	Entry	Description
Server Name	Character	64	Required	The name of the Media Server instance, which manages this stacker.
Stacker ID	Character	64	Required	A unique identifier, which identifies this particular stacker.
Stacker Model	Character	64	Optional	The model of stacker, if known.
Description	Character	100	Optional	A mnemonic description of the stacker used by the operator to clarify which stacker is intended.
Stacker Path	Character	250	Required	The Unix fully qualified path to the device path for the stacker robotic arm.
Element Number	Integer	16	Required	The element number of the stacker.
Stacker Number	Integer	16	Required	The stacker number for this Media Server instance (e.g., 1, 2, or 3).
Number of Slots	Integer	16	Required	The number of usable slots provided by the stacker. Do not count the cleaning tape slot in this count – it is automatically accounted for.
Drive Name	Character	n/a	System generated	Name of drive located inside this stacker
Description	Character	n/a	System generated	A mnemonic description of the drive used by the operator to clarify which type is intended.

Note that by pushing down the **Barcode Enabled** button, the server has the capability to initiate an automated resynchronization with the database.

The Stacker Configuration window also includes a pull-down menu – Drive Operations – with the following options:

- **Add Drive...** Allows the operator to add and configure a new drive, which is located inside this stacker. A list of pre-configured drive models is shown when this menu option is selected. The operator can select from the list of pre-configured drive models, or select the **Unknown** option. In either case, the window shown below in Figure 4.10.2-9 is presented. If a pre-configured model is selected, the fields in the Device Configuration window are pre-populated with the appropriate values for the selected model.
- **Modify Drive** If a drive is selected, this option permits the operator to modify the configuration of the selected device. This also brings up the window shown in Figure 4.10.2-9.
- **Delete Drive** If a drive is selected, this option permits the operator to remove the selected device as a configured resource for this stacker.

Whether adding or modifying configuration for a device, the window shown in Figure 4.10.2-9 (Stacker Device Configuration Window) is presented.

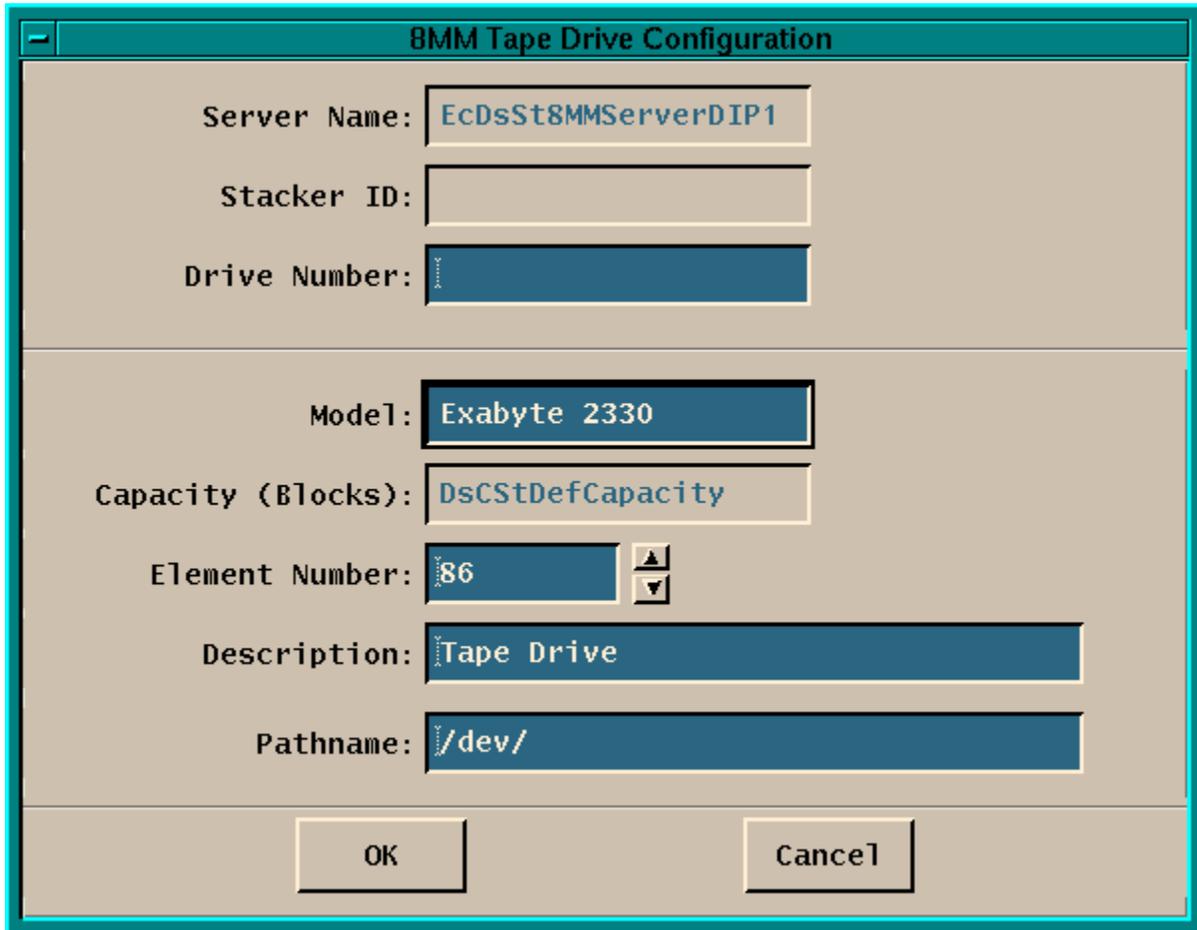


Figure 4.10.2-9. Stacker Device Configuration Window

Table 4.10.2-9 provides a detailed description of the Stacker Device Configuration window fields.

Table 4.10.2-9. Stacker Device Configuration Window Field Descriptions (1 of 2)

Field Name	Data Type	Size	Entry	Description
Server Name	Character	64	Required	The name of the Media Server instance, which manages this device.
Stacker ID	Character	64	Required	A unique identifier, which identifies the stacker in which this device resides.

Table 4.10.2-9. Stacker Device Configuration Window Field Descriptions (2 of 2)

Field Name	Data Type	Size	Entry	Description
Drive Number	Integer	16	Required	The ordinal index of the device within the stacker (e.g., 1, 2, 3...).
Model	Character	64	Optional	The model of device, if known.
Capacity (Blocks)	Integer	16	System generated	Tape capacity, as expressed in Media Server blocks. This cannot be changed in this release, and is presented as a courtesy to the operator.
Element Number	Integer	16	Required	The element number of the device.
Description	Character	100	Optional	A mnemonic description of the device used by the operator to clarify which device is meant (e.g., Topmost).
Pathname	Character	250	Required	The Unix fully qualified path to the device path for the drive.

4.10.2.2.1.6 DTF and CD-ROM Server Configuration

Figure 4.10.2-10 is the configuration screen for media servers, which supports standalone devices. This screen is used to configure instances of the DTF Server and the CD-ROM Server. Table 4.10.2-10 describes the field information related to this pop-up window.

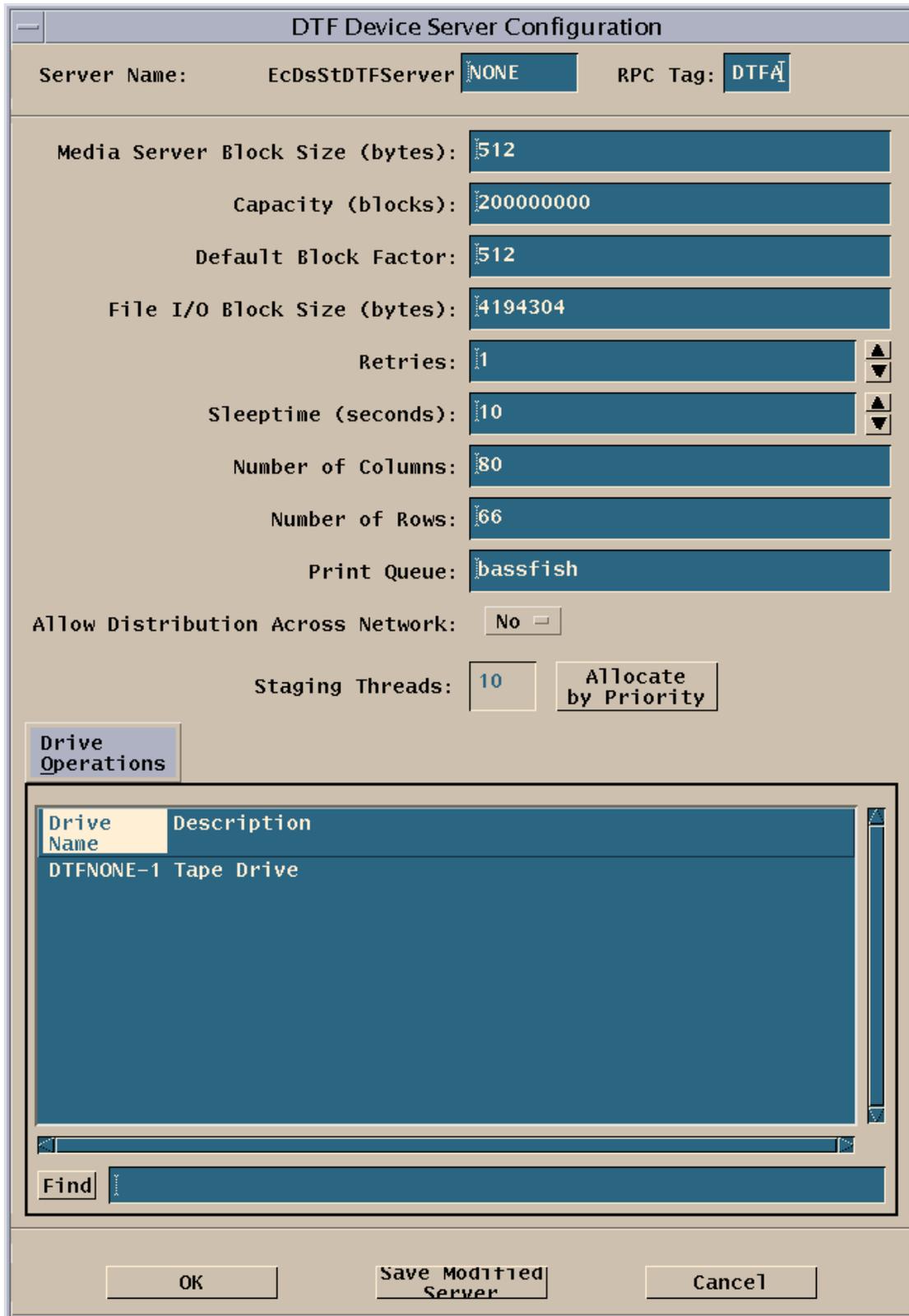


Figure 4.10.2-10. Media Server Configuration Window (Standalone-based)

The Media Server Configuration window includes two buttons not found on other server configuration windows: **Allow Distribution Across Network** and **Save New Server**.

Allow Distribution Across Network Toggles whether or not the server instance permits distribution to media from an NFS-mounted directory. Some media types (e.g. CD-ROM) required a sustained I/O level, which cannot be guaranteed when I/O is being done across a network. For such media types, this flag should be toggled to **No**.

Save New Server This button forces the displayed configuration information to be added to the database without closing the server configuration window. It must be pressed prior to adding new devices.

**Table 4.10.2-10. Media Server Configuration Window Field Descriptions
(Standalone-based) (1 of 2)**

Field Name	Data Type	Size	Entry	Description
Server Name	Character	6	Required	Name of this instance of the Media Server.
RPC Tag	Character	4	Required	The four-character tag appearing in RPC Ids submitted by the Media Server. Recommended values for DTF servers are DEEA, DEEB, etc., and for the CDROM server are CDRA, CDRB, etc.
Media Server Block Size (bytes)	Integer	16	Required	The block size used for media I/O. This block size is also the basis unit for specifying media capacity.
Capacity (blocks)	Integer	16	Required	The maximum aggregate size of files, which can be transferred on a single piece of media. The size is specified in terms of Media Server Block Size.
Default Block Factor	Integer	16	Required	The number of blocks, which are read/written at once when performing I/O.
File I/O Block Size (bytes)	Integer	16	Required	The block size to be used for file I/O.
Retries	Integer	16	Required	The number of times to retry failed operations.
Sleeptime (seconds)	Integer	16	Required	The time to sleep, in seconds, between retries.
Number of Columns	Integer	16	Required	The column width to which packing slips should be formatted.
Number of Rows	Integer	16	Required	The page length to which packing slips should be formatted.
Print Queue	Character	64	Required	The Unix named print queue to which packing slips should be sent for requests serviced by this Media Server instance.

**Table 4.10.2-10. Media Server Configuration Window Field Descriptions
(Standalone-based) (2 of 2)**

Field Name	Data Type	Size	Entry	Description
Allow Distribution Across Network	Boolean	N/A	Required	Indicates whether or not this Media Server instance permits media distribution from a staging disk, which is NFS-mounted in the event that the data cannot be locally staged. The CDROM server is not reliable for distribution from NFS-mounted disks, as I/O to the CD-ROM device must be sustained.
Staging Threads	Integer	16	Required	The number of Service Threads available to locally stage data for media distribution by this Media Server instance. It also does not correspond to the number of devices available for media distribution. A separate thread pool is automatically created for actual media I/O, based on the number of media devices configured for use by this Media Server instance.
Drive Name	Character	64	Required	A unique identifier, which identifies a drive associated with this Media Server instance.
Description	Character	100	Optional	Additional mnemonic information provided by the operator during configuration to assist in identifying the listed drive.

By clicking on the **Allocate by Priority** button, these threads can be allocated by priority in order to reserve certain resources for higher priority requests. Clicking the **Save Modified Server** button allows the operator to save any changes that were made to the server configuration.

The Media Server Configuration window also includes a pull-down menu – Drive Operations – with the following options:

- **Add Drive...** Allows the operator to add and configure a new drive to be managed by this media server instance. A list of pre-configured drive models is shown when this menu option is selected. The operator can select from the list of pre-configured drive models, or select the **Unknown** option. In either case, the window shown below in Figure 4.10.2-11 is presented. If a pre-configured model is selected, the fields in the Device Configuration window are pre-populated with the appropriate values for the selected model.
- **Modify Drive** If a drive is selected, this option permits the operator to modify the configuration of the selected device. This also brings up the window shown in Figure 4.10.2-11.

- **Delete Drive** If a drive is selected, this option permits the operator to remove the selected device as a configured resource for this server instance.

Whether adding or modifying configuration for a device, the window shown below in Figure 4.10.2-11 is presented.

The screenshot shows a window titled "DTF Drive Configuration" with a light beige background. The window contains several input fields with blue text and a blue highlight. The fields are: "Server Name" with the value "EcDsStDTFServerNONE", "Drive Number" with the value "1", "Model" with the value "Sony DTF Tape", "Capacity (Blocks)" with the value "200000000", "Element Number" with the value "86" and up/down arrow buttons, "Description" with the value "Tape Drive", and "Pathname" with the value "/dev/rmt/3n". At the bottom of the window are two buttons: "OK" and "Cancel".

Figure 4.10.2-11. Standalone Device Configuration Window

Table 4.10.2-11 provides a detailed description of the Standalone Device Configuration window fields.

Table 4.10.2-11. Standalone Device Configuration Window Field Descriptions

Field Name	Data Type	Size	Entry	Description
Server Name	Character	64	Required	The name of the Media Server instance, which manages this device.
Drive Number	Integer	16	Required	The ordinal index of the device for the Media Server instance (e.g., 1, 2, 3).
Model	Character	64	Optional	The model of device, if known.
Capacity (Blocks)	Integer	16	System generated	Media capacity, as expressed in Media Server blocks. This cannot be changed in this release, and is presented as a courtesy to the operator.
Element Number	Integer	16	Required	The element number of the device.
Description	Character	100	Optional	A mnemonic description of the device used by the operator to clarify which device is meant (e.g., Left side of table).
Pathname	Character	250	Required	The Unix fully qualified path to the device path for the drive.

4.10.2.2.1.7 Archive Server Configuration

Figure 4.10.2-12 is the configuration screen for configuring instances of the Archive Server.

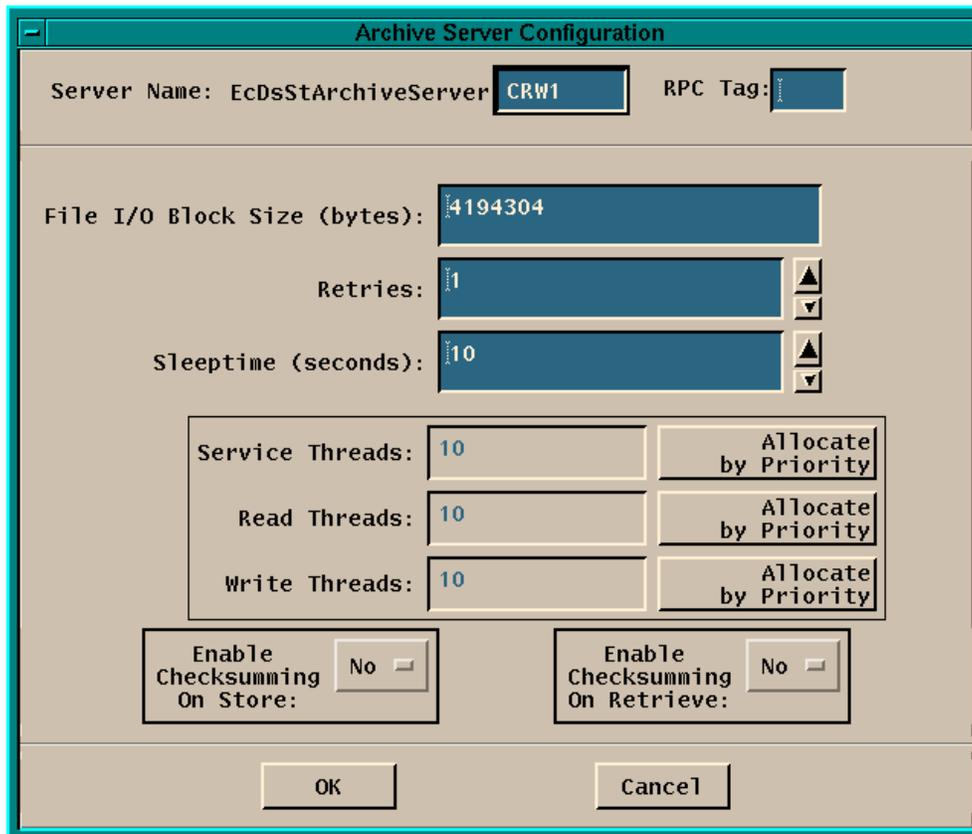


Figure 4.10.2-12. Archive Server Configuration Window

Table 4.10.2-12 describes the fields shown on the Archive Server Configuration pop-up window.

Table 4.10.2-12. Archive Server Configuration Window Field Descriptions

Field Name	Data Type	Size	Entry	Description
Server Name	Character	6	Required	Name of this instance of the Archive Server.
RPC Tag	Character	4	Required	The four-character tag, which appears in RPC Ids submitted by the Archive Server. Recommended values are ARCA, ARCB, etc.
File I/O Block Size (bytes)	Integer	16	Required	The block size to be used for file I/O.
Retries	Integer	3	Required	The number of times to retry failed operations.
Sleeptime (seconds)	Integer	5	Required	The time to sleep, in seconds, between retries.
Enable Checksumming on Store	Boolean	N/A	Required	If set to Yes, the Archive Server computes a CRC-32 checksum for each file stored in the archive.
Enable Checksumming on Retrieve	Boolean	N/A	Required	If set to Yes, the Archive Server directs the Cache Manager to compute a CRC-32 checksum for each file retrieved from the archive and validate it against the checksum previously computed.
Service Threads	Integer	5	Required	The numbers of Service Threads available to process requests, which do not directly involve AMASS devices. High-level Store and Retrieve requests are split into subordinate StoreFile and RetrieveFile requests, which are serviced by read and write threads, respectively.
Read Threads	Integer	5	Required	The number of Service Threads available to process requests, which are sent to a Cache Manager to read from an AMASS device. Rather, it should reflect the number of devices available in the AMASS silo for reading, adjusted to allow for caching of files.
Write Threads	Integer	5	Required	The number of Service Threads available to process requests, which write to an AMASS device. Rather, it should reflect the number of devices available in the AMASS silo for writing, adjusted to allow for caching of files.

4.10.2.2.2 Volume Group Configuration (Vol Grp Config.) Tab

This tab, shown in Figure 4.10.2-13, displays information about the volume groups configured to store archive data. It also allows the operator to create new volume groups, add alternative volume group history set to separate forward processing and reprocessing data, browse the history of volume group configuration, and redirect volume groups from one physical location to another.

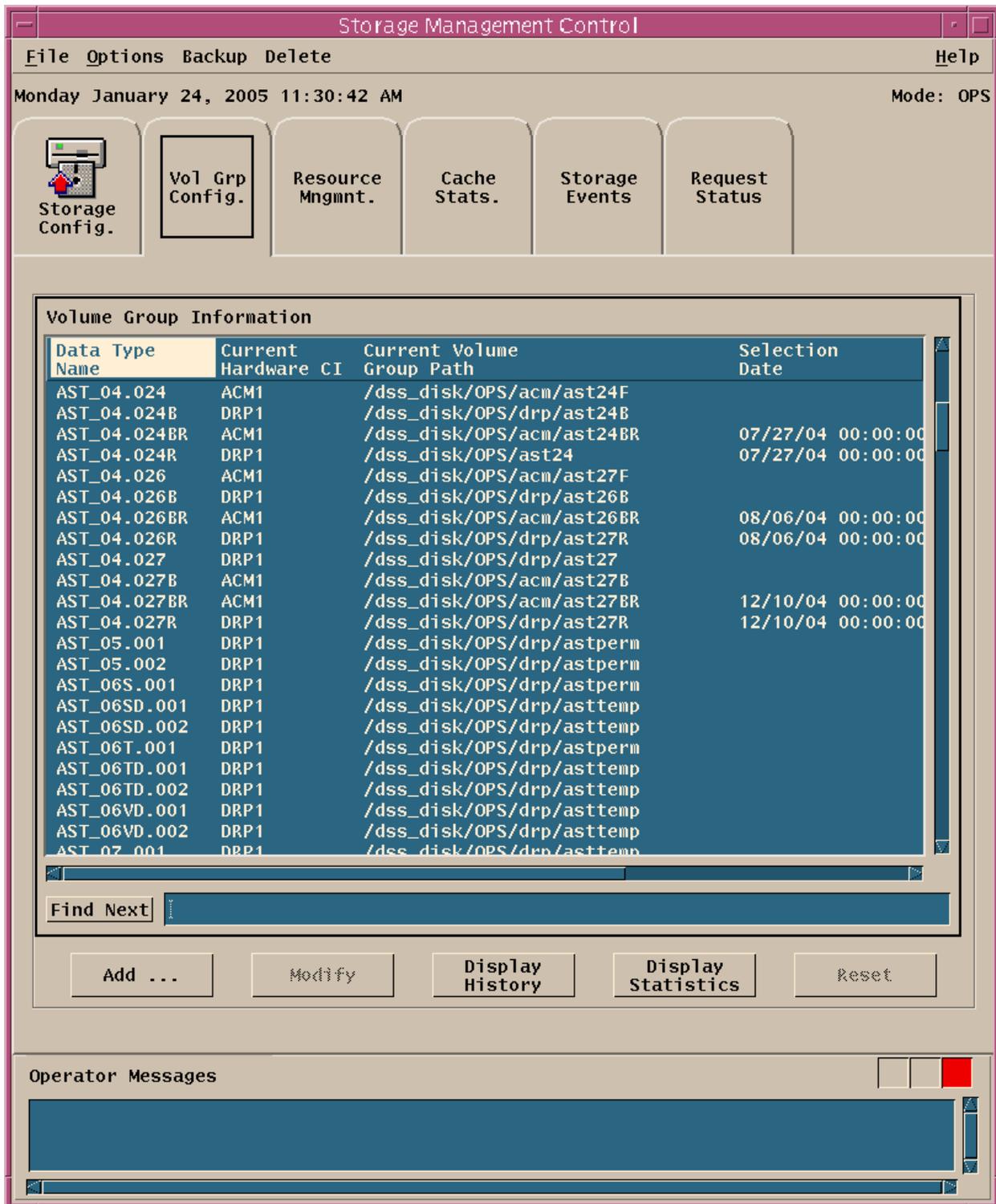


Figure 4.10.2-13. Volume Group Configuration (Vol Grp Config.) Tab

Table 4.10.2-13 provides a detailed description of the Volume Group Configuration Tab fields.

Table 4.10.2-13. Volume Group Configuration Window Field Descriptions

Field Name	Data Type	Size	Entry	Description
Data Type Name	Character	16	Required	An ESDT shortname with version identifier.
Current Hardware CI	Character	16	Required	The Hardware CI of the Archive Server instance currently responsible for storing data of the specified data type.
Current Volume Group Path	Character	255	Required	The fully qualified Unix path to where data is stored for the specified data type.
Selection Date	Character	32	Required	Non-NULL selection date defined for the ESDT version of which there are two volume group history sets defined for forward processing and reprocessing data respectively.
UNIX Compression Factor	Character	3	Required	The extent to which a file of the specified data type is expected to be compressed using the Unix compress algorithm. For example, if the Unix compression factor were 80%, a 10MB file would have an expected compressed size of 2MB using the Unix compress algorithm.
GZIP Compression Factor	Character	3	Required	The extent to which a file of the specified data type is expected to be compressed using the GNU gzip compress algorithm. For example, if the GNU gzip compression factor were 80%, a 10MB file would have an expected compressed size of 2MB using the GNU gzip compress algorithm.
Find Next	Character	N/A	Optional	Allows the operator to perform a keyword search for items in the Volume Group Information box.
Operator Messages	Character	N/A	System generated	System will generate any necessary messages for the operator to read

The following buttons provide the operator with several configuration operations with regard to the Volume Groups:

- **Add...** Allows the operator to add Volume Groups
- **Modify** Allows the operator to modify the configuration of one or more Volume Groups. Also used to version a set of ESDTs
- **Display History** Allows the operator to review the set of physical file storage locations, which have been associated with a Volume Group over time

- **Display Statistics** Allows the operator to review historical compression statistics, which can be used to guide the selection of a configured compression factor
- **Reset** Allows the operator to un-select the selections in the Volume Group Information box

Each of the windows associated with the above buttons is described in the following sections.

4.10.2.2.2.1 Add Volume Group(s)

Add Volume Groups enable user to add all volume groups, including alternative volume group sets for reprocessing in one transaction. Figure 4.10.2-14 is the window presented when the **Add...** button is pressed.

Add Volume Groups

Data Type.Version:

Add Options: Primary Backup Offsite Alternative

Selection Date (mm/dd/yyyy): / /

New Alternative VG For: REPROCESSING FORWARD PROCESSING

UNIX Compression Factor (%): GZIP Compression Factor (%):

Primary	Primary Alternative
HWCI: <input type="text" value="ACM1"/> ▼	HWCI: <input type="text" value="DRP1"/> ▼
VG Path: <input type="text" value="/archive/mod/acm/mod399"/>	VG Path: <input type="text"/>
Backup	Backup Alternative
HWCI: <input type="text" value="DRP1"/> ▼	HWCI: <input type="text" value="ACM1"/> ▼
VG Path: <input type="text" value="/archive/mod/drp/mod399"/>	VG Path: <input type="text"/>
Offsite	Offsite Alternative
HWCI: <input type="text"/>	HWCI: <input type="text"/>
VG Path: <input type="text"/>	VG Path: <input type="text"/>

Figure 4.10.2-14. Add Volume Groups Pop-up Window

Table 4.10.2-14 provides the descriptions for all common fields in the Add Volume Groups pop-up window.

Table 4.10.2-14. Add Volume Groups Window Field Descriptions (1 of 2)

Field Name	Data Type	Size	Entry	Description
Data Type. Version	Character	16	Required	An ESDT shortname with version identifier.
Add Options	Check box	N/A	Required	Determine what volume groups are going to be configured. The primary check box can only be selected if the Data Type.version is brand new, i.e. no volume group configured for it. Otherwise, the check box is disabled. The backup check box can only be selected when primary volume group has been configured or the primary check box is selected. The offsite check box follows same rule as the Backup check box. Alternative checkbox is enabled only when the primary volume group for the given ESDT has been configured or the check box for the primary has been selected. For a new ESDT version, all volume groups, including alternative volume groups can be configured in one transaction by selecting appropriate check boxes.
Selection Date	Character	8	Required if adding Alternative Volume Group History Set	When the alternative check box is selected, the Selection Date section is enabled and is required to be filled out by the user. Selection Date is a separate date to guide Archive Server to select a appropriate Volume Group History set for storing / retrieving data. When acquisition date is not null and less than the Selection Date, Reprocessing Volume Group history set will be used, otherwise, forward processing Volume Group history set will be used.
New Alternative VG for	Radio Button	N/A	Required if adding Alternative Volume Group History Set	Alternative volume groups can be configured either for reprocessing or even for forward processing. Default is for reprocessing. Although the flexibility to add a new alternative for forward processing is supported, it should be used with a great caution.

Table 4.10.2-14. Add Volume Groups Window Field Descriptions (2 of 2)

Field Name	Data Type	Size	Entry	Description
Unix Compression Factor (%)	Integer	3	Required if adding Primary Volume Group	The extent to which a file of the specified data type is expected to be compressed using the Unix compress algorithm. For example, if the Unix compression factor were 80%, a 10MB file would have an expected compressed size of 2MB using the Unix compress algorithm.
GZIP Compression Factor (%)	Integer	3	Required if adding Primary Volume Group	The extent to which a file of the specified data type is expected to be compressed using the GNU gzip compress algorithm. For example, if the GNU gzip compression factor were 80%, a 10MB file would have an expected compressed size of 2MB using the GNU gzip compress algorithm.

4.10.2.2.2 Add Alternative Volume Group History Set

Alternative Volume Group History Set is designed for holding reprocessing data. When “Alternative” check box is selected, “Selection Date”, “New Alternative VG for” sections on the window will be enabled. However, which HWCI/VG Path section will be enabled will dependent upon what volume group(s) have been configured (or selected). For instance, if both primary and backup volume groups have been configured for a given ESDT version, both “Primary Alternative” and “Backup Alternative” entry sections will be enabled and required to be entered.

The Figure 4.10.2-15 displays the window with which only Primary Alternative is selected.

Add Volume Groups

Data Type.Version:

Add Options: Primary Backup Offsite Alternative

Selection Date (mm/dd/yyyy): / /

New Alternative VG For: REPROCESSING FORWARD PROCESSING

UNIX Compression Factor (%): GZIP Compression Factor (%):

Primary	Primary Alternative
HWCI: <input type="text" value=""/>	HWCI: <input type="text" value="DRP1"/>
VG Path: <input type="text" value=""/>	VG Path: <input type="text" value="/archive/ast/drp/ast009R"/>
Backup	Backup Alternative
HWCI: <input type="text" value=""/>	HWCI: <input type="text" value=""/>
VG Path: <input type="text" value=""/>	VG Path: <input type="text" value=""/>
Offsite	Offsite Alternative
HWCI: <input type="text" value=""/>	HWCI: <input type="text" value=""/>
VG Path: <input type="text" value=""/>	VG Path: <input type="text" value=""/>

Figure 4.10.2-15. Add Volume Groups Pop-Up Window

Additional to the table 4.10.2-14, Table 4.10.2-15 provides other field description in the Add Volume Groups pop-up window

Table 4.10.2-15. Add Volume Groups Window Field Descriptions (1 of 2)

Field Name	Data Type	Size	Entry	Description
HWCI (for Primary)	Selection List	N/A	Required if Primary enabled	The Hardware CI of the Archive Server instance that is currently responsible for storing data of the specified data type to the Primary Archive.
Volume Group Path (For Primary)	Character	Unlimited	Required if Primary enabled	The fully-qualified Unix path to where data is currently being stored for the specified data type to the Primary Archive.
New HWCI (For Backup)	Selection List	N/A	Required if Backup enabled	The Hardware CI of the Archive Server instance that is currently responsible for storing data of the specified data type to the Backup Archive.
Volume Group Path (For Backup)	Character	Unlimited	Required if Backup enabled	The fully-qualified Unix path to where data is currently being stored for the specified data type to the Backup Archive.
New HWCI (For Offsite)	Selection List	N/A	Required if Offsite enabled	The Hardware CI of the Archive Server instance that is currently responsible for storing data of the specified data type to the Offsite Archive.
Volume Group Path (For Offsite)	Character	Unlimited	Required if Offsite enabled	The fully-qualified Unix path to where data is currently being stored for the specified data type to the Offsite Archive.
HWCI (for Primary Alternative)	Selection List	N/A	Required if Primary Alternative enabled	The Hardware CI of the Archive Server instance that is currently responsible for storing reprocessing data of the specified data type to the Primary Alternative Archive.
Volume Group Path (For Primary Alternative)	Character	Unlimited	Required if Primary Alternative enabled	The fully-qualified Unix path to where reprocessing data is currently being stored for the specified data type to the Primary Alternative Archive.
New HWCI (For Backup Alternative)	Selection List	N/A	Required if Backup Alternative enabled	The Hardware CI of the Archive Server instance that is currently responsible for storing reprocessing data of the specified data type to the Backup Alternative Archive.
Volume Group Path (For Backup Alternative)	Character	Unlimited	Required if Backup Alternative enabled	The fully-qualified Unix path to where data is currently being stored for the specified data type to the Backup Alternative Archive.

Table 4.10.2-15. Add Volume Groups Window Field Descriptions (2 of 2)

Field Name	Data Type	Size	Entry	Description
New HWCI (For Offsite Alternative)	Selection List	N/A	Required if Offsite Alternative enabled	The Hardware CI of the Archive Server instance that is currently responsible for storing reprocessing data of the specified data type to the Offsite Alternative Archive.
Volume Group Path (For Offsite Alternative)	Character	Unlimited	Required if Offsite Alternative enabled	The fully-qualified Unix path to where data is currently being stored for the specified data type to the Offsite Alternative Archive.

4.10.2.2.2.3 Modify Volume Group

Figure 4.10.2-16a is the window, which can be used to modify Volume Group configuration information. This window is displayed when any ESDT version, for which no alternative Volume Group History Set configured, is selected and the **Modify** button is pressed.

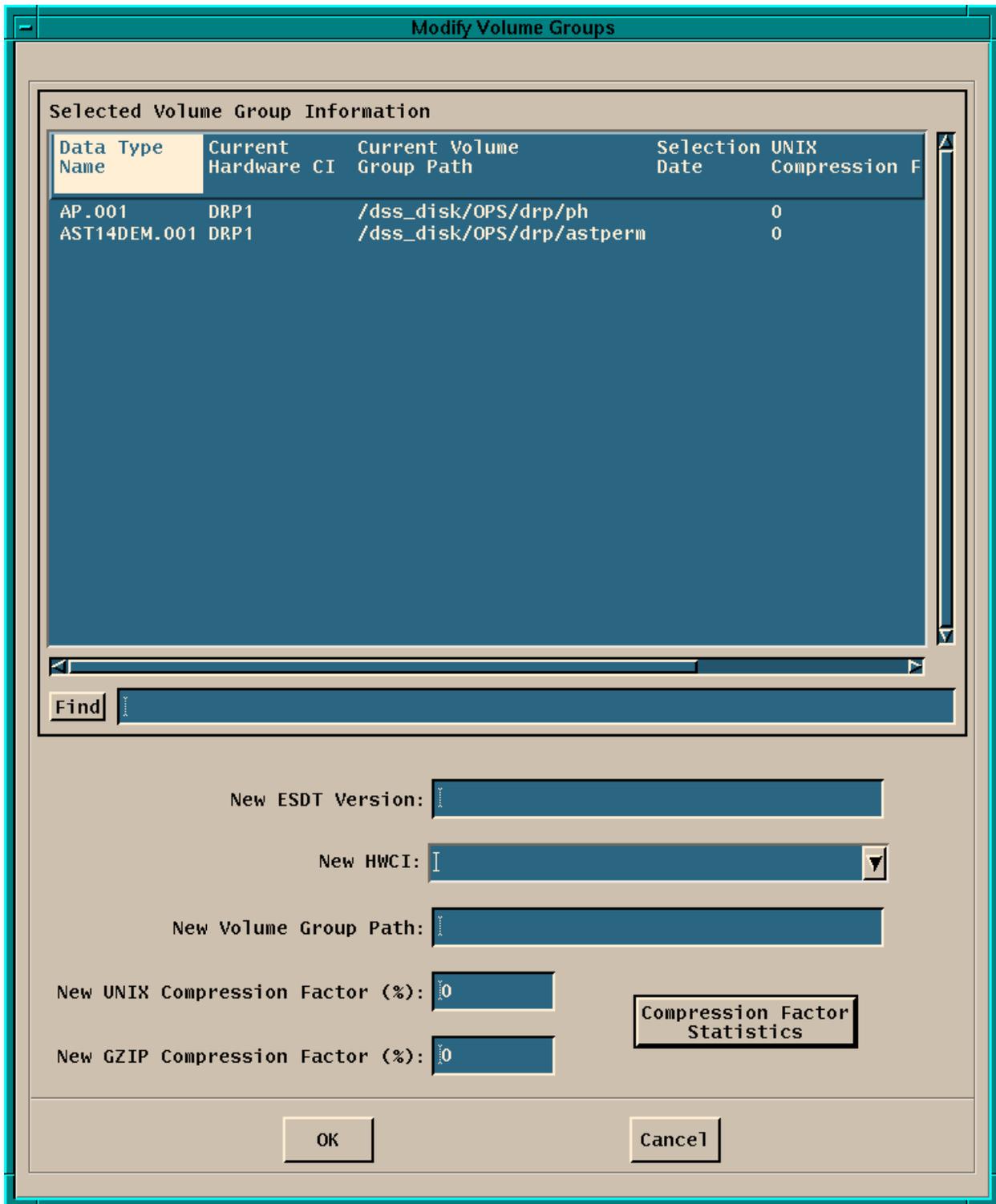


Figure 4.10.2-16a. Modify Volume Group Configuration Pop-up Window

Figure 4.10.2-16b is the window, which can be used to modify Volume Group configuration information for alternative Volume Group History Set. This window is displayed when any ESDT version, for which alternative Volume Group History Set configured, is selected and the **Modify** button is pressed. Under this circumstance, only HWCI and Volume Group Path can be modified.

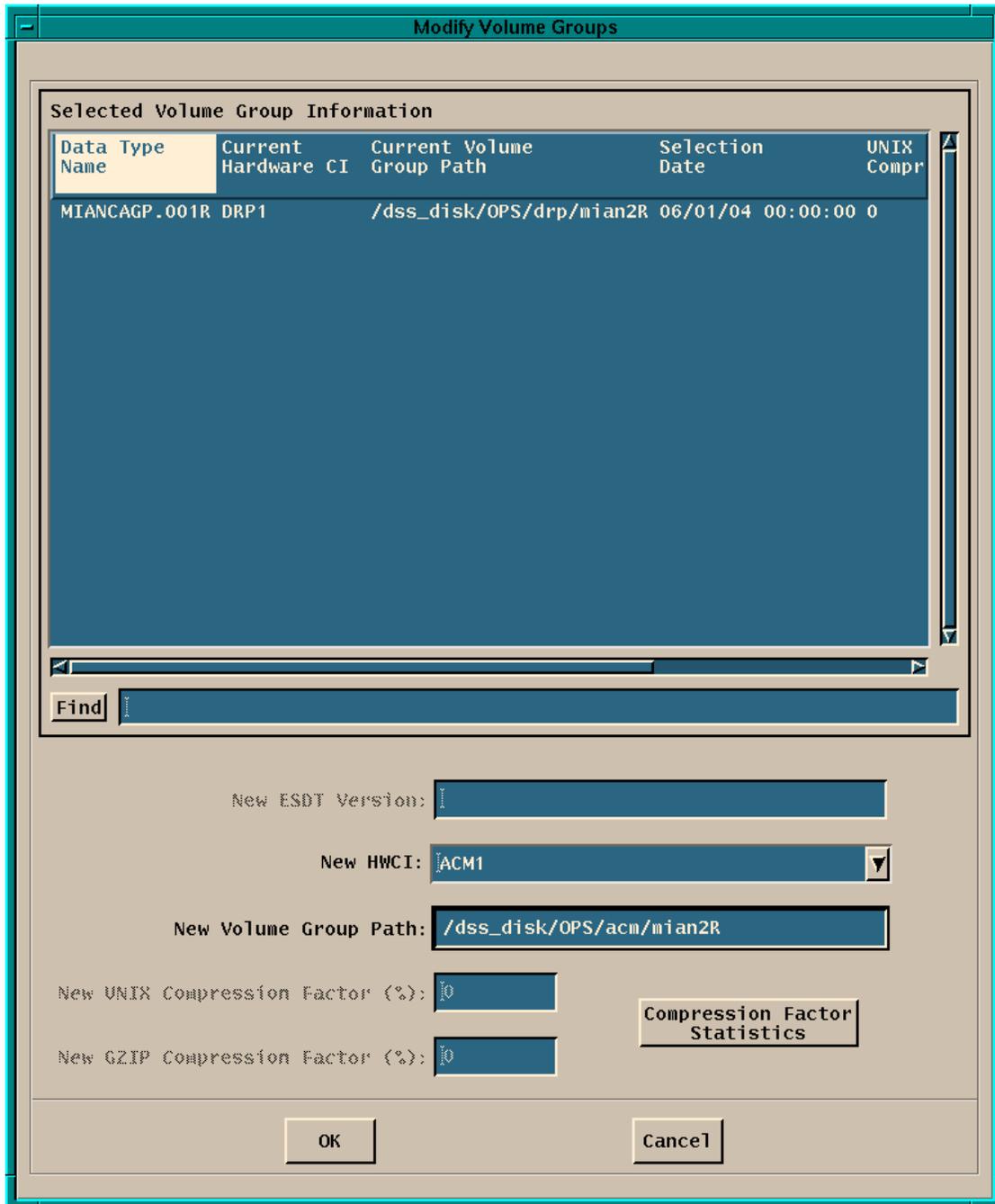


Figure 4.10.2-16b. Modify Volume Group Configuration Pop-up Window

Table 4.10.2-16 provides a detailed description of the Modify Volume Group Pop-up window fields. Note that multiple Volume Groups can be selected for simultaneous modification. Those fields for which new values are specified are updated; fields, which are left blank retain their existing values for each affected Volume Group.

This also provides a simple mechanism for versioning ESDTs as a group. By specifying a new ESDT Version, new entries can be created for Volume Groups for each of the selected Volume Groups, updating the ESDT version of the original Volume Group with the new ESDT version. All other configuration information for the new version Volume Groups are retained, unless explicitly altered by placing an entry in one or more of the other fields on the window. This functionality will be disabled when any ESDT version selected as one group and some of them hold more than one Volume Group History Set.

Pressing the **Compression Factor Statistics** button brings up the Compression Statistics window for each of the selected Volume Groups. This permits the operator to review historical statistics regarding compression rates for each selected Volume Group. See Section 4.10.2.2.2.4 for more information regarding the Compression Statistics window.

Table 4.10.2-16. Modify Volume Group Pop-up Window Field Descriptions (1 of 2)

Field Name	Data Type	Size	Entry	Description
Data Type Name	Character	16	Required	An ESDT shortname with version identifier.
Current Hardware CI	Character	16	Required	The Hardware CI of the Archive Server instance currently responsible for storing data of the specified data type.
Current Volume Group Path	Character	Unlimited	Required	The fully qualified Unix path to where data is currently being stored for the specified data type.
Selection Date	Character	32	Required	Non-NULL selection date defined for the ESDT version of which there are two volume group history sets defined for forward processing and reprocessing data respectively.
Unix Compression Factor	Character	3	Required	The extent to which a file of the specified data type is currently expected to be compressed using the Unix compress algorithm. For example, if the Unix compression factor were 80%, a 10MB file would have an expected compressed size of 2MB using the Unix compress algorithm.

Table 4.10.2-16. Modify Volume Group Pop-up Window Field Descriptions (2 of 2)

Field Name	Data Type	Size	Entry	Description
GZIP Compression Factor	Character	3	Required	The extent to which a file of the specified data type is currently expected to be compressed using the GNU gzip compress algorithm. For example, if the GNU gzip compression factor were 80%, a 10MB file would have an expected compressed size of 2MB using the GNU gzip compress algorithm.
New ESDT Version	Character	3	Optional	The new version identifier to be applied to all displayed ESDTs. If specified, the set of selected volume groups are re-added as new volume groups for the new version identifier.
New HWCI	Character	16	Optional	The Hardware CI of the Archive Server instance made responsible for the volume group. If the operator does not specify this, the currently configured value is retained.
New Volume Group Path	Character	255	Optional	This entry identifies the new path in which data is stored. If the operator does not specify this, the currently configured value is retained.
New Unix Compression Factor (%)	Integer	3	Optional	The new expected compression rate for files compressed using the Unix compress method. If the operator does not specify this, the currently configured value is retained.
New GZIP Compression Factor (%)	Integer	3	Optional	The new expected compression rate for files compressed using the GNU gzip compress method. If the operator does not specify this, the currently configured value is retained.

4.10.2.2.4 Volume Group History

Figure 4.10.2-17 is the window used to view historical Volume Group configuration information. This window is displayed when the **Display History** button is pressed.

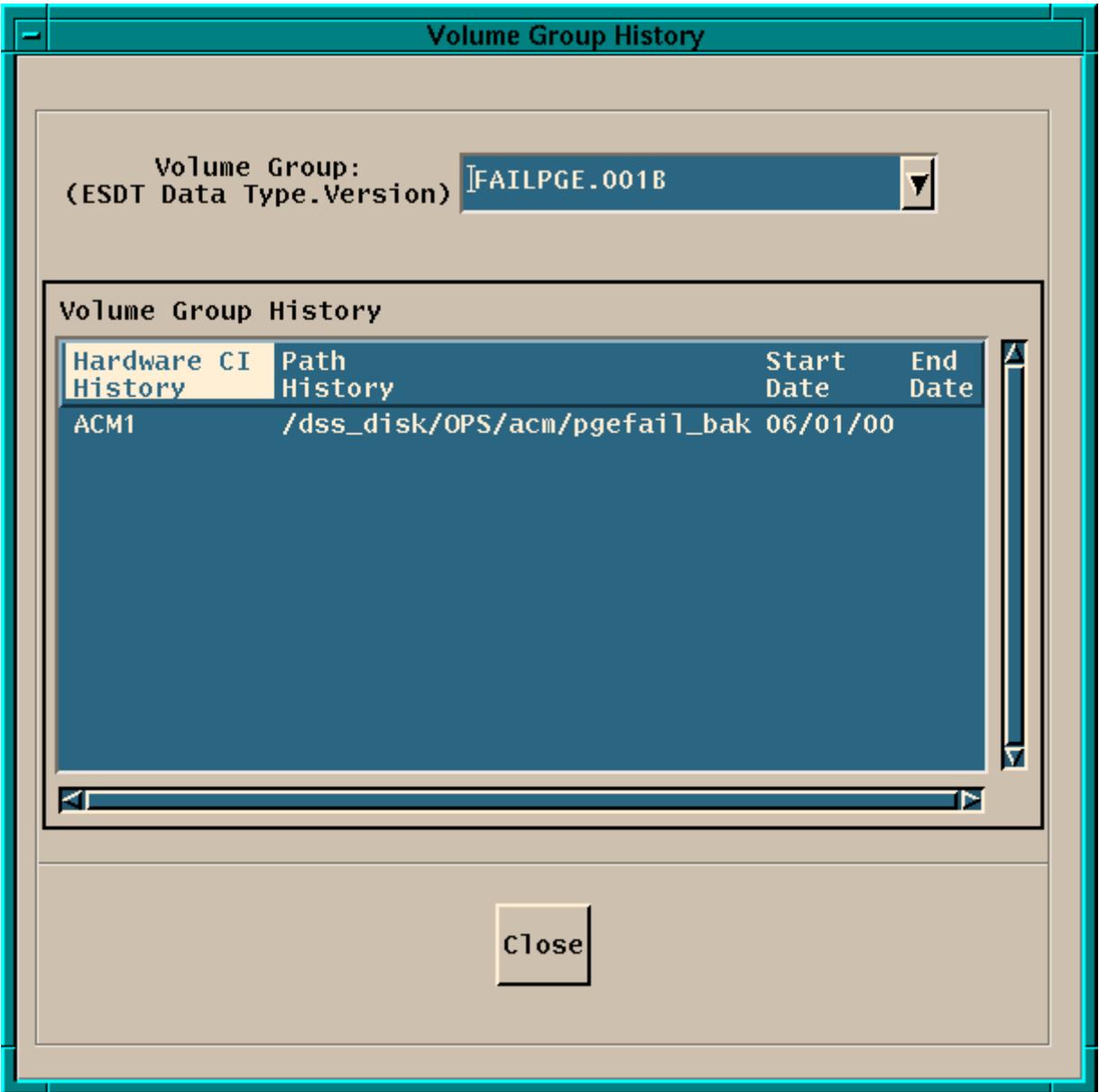


Figure 4.10.2-17. Volume Group History Pop-up Window

Table 4.10.2-17 provides a detailed description of the Volume Group History Pop-up window fields.

Table 4.10.2-17. Volume Group History Field Description

Field Name	Data Type	Size	Entry	Description
Volume Group (ESDT Data Type. Version)	Character	16	Required	An ESDT shortname with version identifier.
Hardware CI History	Character	16	Required	In reverse chronological order, the Hardware CI of the Archive Server instances that have been responsible for storing data of the specified data type. The current instance is listed first.
Path History	Character	Unlimited	Required	In reverse chronological order, the fully qualified Unix paths to where data has been stored for the specified data type. The current path is listed first.
Start Date	Date	16	Required	The date on which this configuration became active for the listed data type.
End Date	Date	16	Required	The date on which this configuration was superseded by new configuration information. If blank, this row reflects the current configuration for the volume group. If any row has a blank end date, the volume group is closed, and no further data is accepted for that volume group.

4.10.2.2.2.5 Compression Statistics

Figure 4.10.2-18 is the window used to view historical compression statistics information. This window is displayed when the **Display Statistics** button is pressed from the Volume Group Configuration tab, or when the **Compression Factor Statistics** button is pressed from the Modify Volume Groups window.

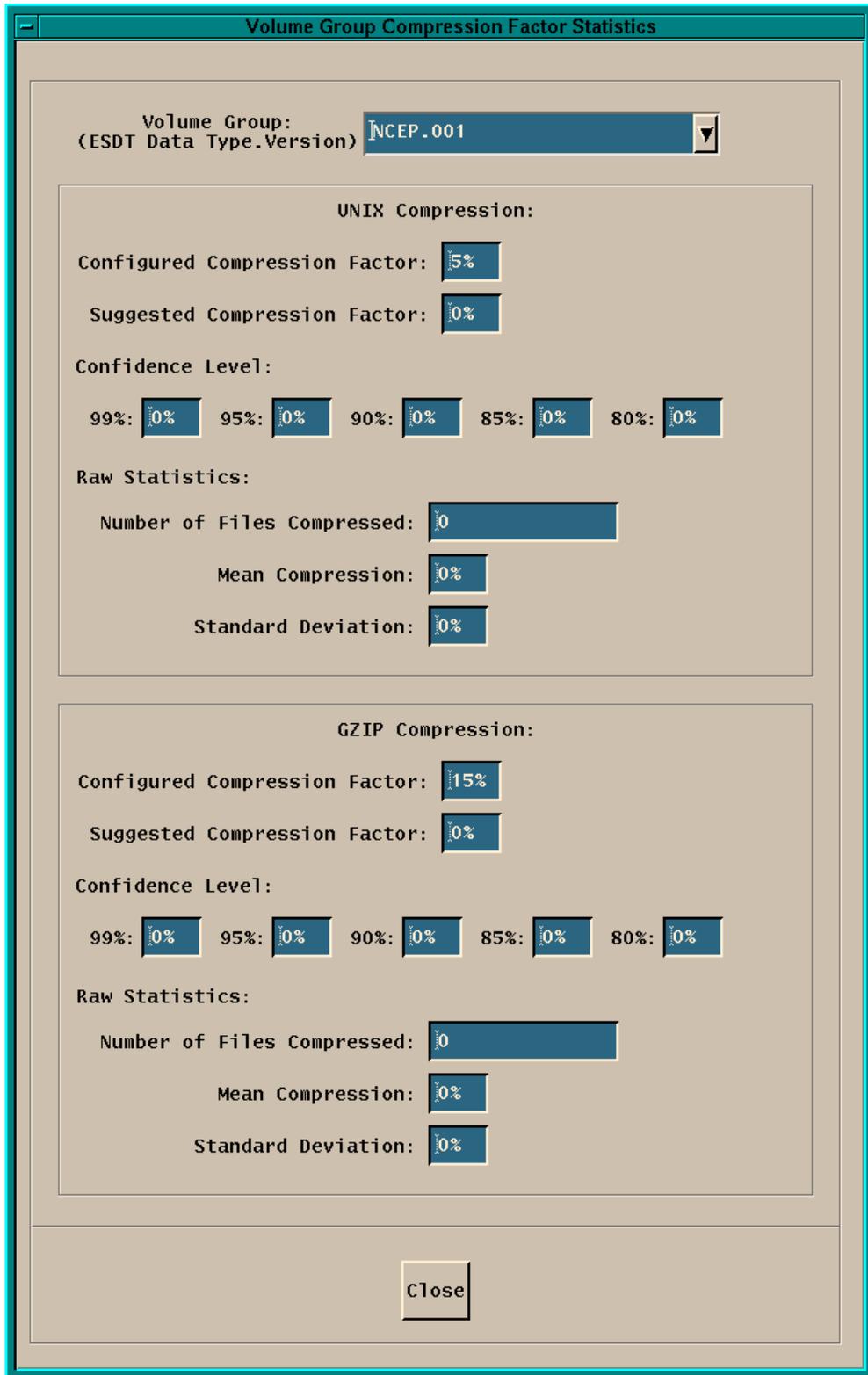


Figure 4.10.2-18. Volume Group Compression Factor Statistics Pop-up Window

Table 4.10.2-18 provides a detailed description of the Volume Group Compression Factor Statistics Pop-up window fields.

Table 4.10.2-18. Volume Group Compression Factor Statistics Pop-up Window Field Descriptions

Field Name	Data Type	Size	Entry	Description
Volume Group (ESDT Data Type. Version)	Character	16	Required	An ESDT shortname with version identifier.
Configured Compression Factor	Character	4	Required	The extent to which a file of the specified data type is currently expected to be compressed. For example, if the Unix compression factor were 80%, a 10MB file would have an expected compressed size of 2MB using the Unix compress algorithm.
Suggested Compression Factor	Character	4	Calculated	The recommended compression factor based on a statistical analysis of historical compression data for the data type.
Confidence Level (99%, 95%, 90%, 85%, 80%)	Character	4	Calculated	Each of these fields specifies the minimum compression that can be reasonably expected at each of the five confidence levels, based on statistical analysis. As the confidence level increases, the expected minimum compression factor drops.
Number of Files Compressed	Integer	16	Calculated	The number of files of this data type which have been compressed to date.
Mean Compression	Character	4	Calculated	The average compression for the set of files of this data type which have been compressed to date.
Standard Deviation	Character	4	Calculated	A statistical factor, which is computed based on the variation from the average compression experienced for files of this data type. The greater the standard deviation, the less reliable and predictable the expected compression factor is.

Note: The GZIP Compression Factors work identically to the UNIX Compression Factors.

4.10.2.2.3 Resource Management Tab (Resource Mngmnt)

The Resource Management tab allows the operator to monitor and adjust the availability of given storage devices as shown in Figure 4.10.2-19.

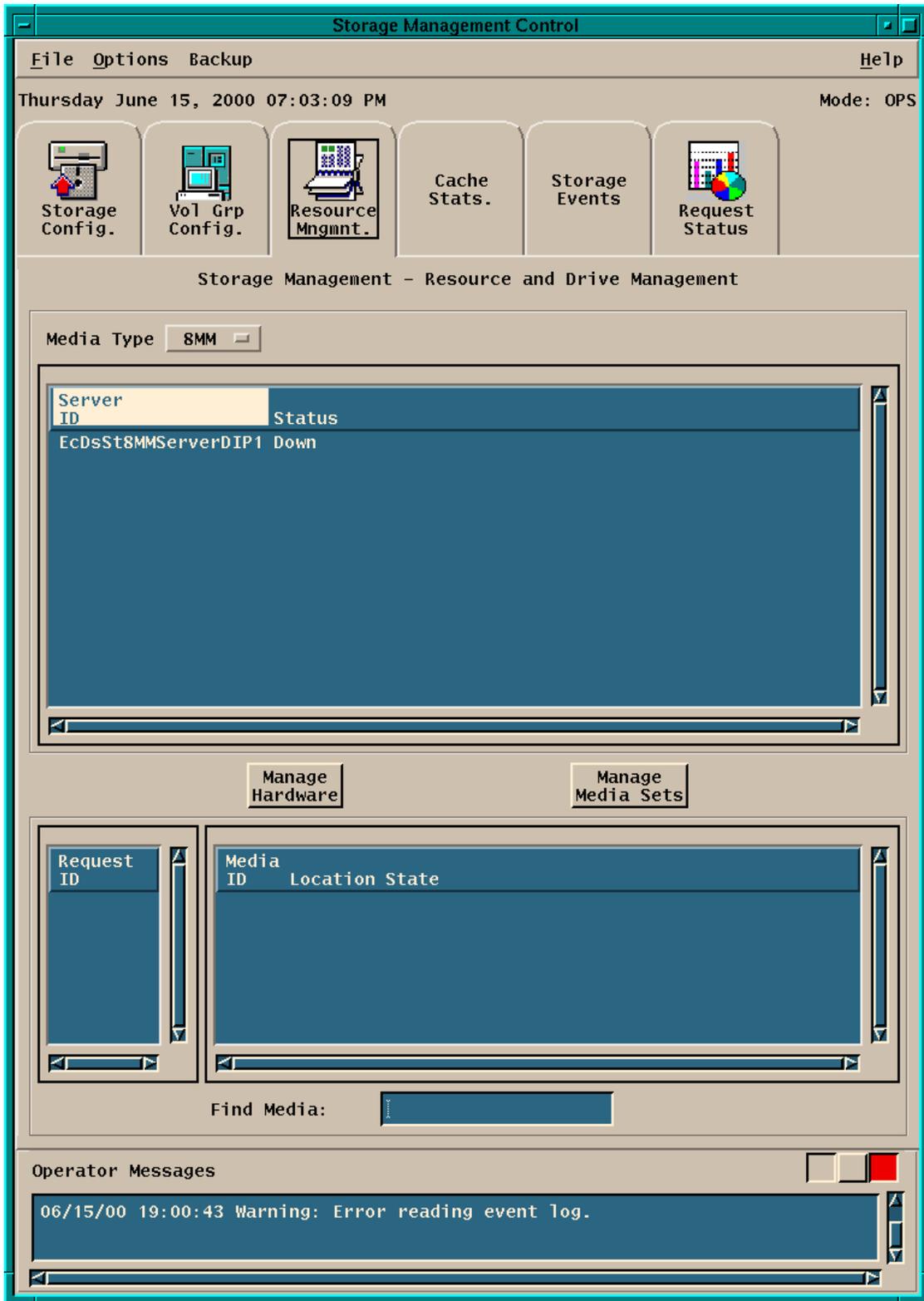


Figure 4.10.2-19. Resource Management (Resource Mngmnt) Tab

When the operator highlights a server ID, the related Request IDs appear. Clicking on one of the Request IDs, the corresponding media IDs are displayed on the screen, and the current known location and state are displayed for each media.

The following functionality is provided through the three buttons present on the Resource Management Tab.

- **Media Type** allows the operator to select different media type
- **Manage Hardware** allows the operator to manage Stackers, Drivers and Slots by means of the pop-up window shown in Figure 4.10.2-19. Table 4.10.2-19 provides details on the data fields of the Manage Stackers popup window
- **Manage Media Sets** allows the operator to define and manage groups of logically related media by means of the pop-up window shown in Figure 4.10.2-21. Table 4.10.2-21 provides details on the data fields of the Manage Media Sets popup

Table 4.10.2-19 describes the data fields shown in the Resource Management Tab.

Table 4.10.2-19. Resource Management Tab Field Description

Field Name	Data Type	Size	Entry	Description
Server ID	Character	N/A	System provided	Server ID for this media type.
Status	Character	N/A	System provided	Status of the server.
Request ID	Character	N/A	System provided	Identifier of the resource request.
Media ID	Character	N/A	System provided	Media related to this request tape.
Location	Character	N/A	System provided	Location of the media listed in the media ID field (drive, slot and stacker).
State	Character	N/A	System provided	State of the media (e.g., Tape writing, distribution finished, drive loaded, etc.).
Find Media	Character	50	Optional	The operator can use the Find Media capability to locate a specific piece of media, as well as identifying which request the media is associated with.
Operator Messages	Character	N/A	System provided	Any informational, warning or error messages are displayed here.

4.10.2.2.3.1 Manage Hardware

Figure 4.10.2-20 (Manage Stackers) is the window used to manage the availability and status of physical hardware. This window is displayed when the **Manage Hardware** button is pressed.

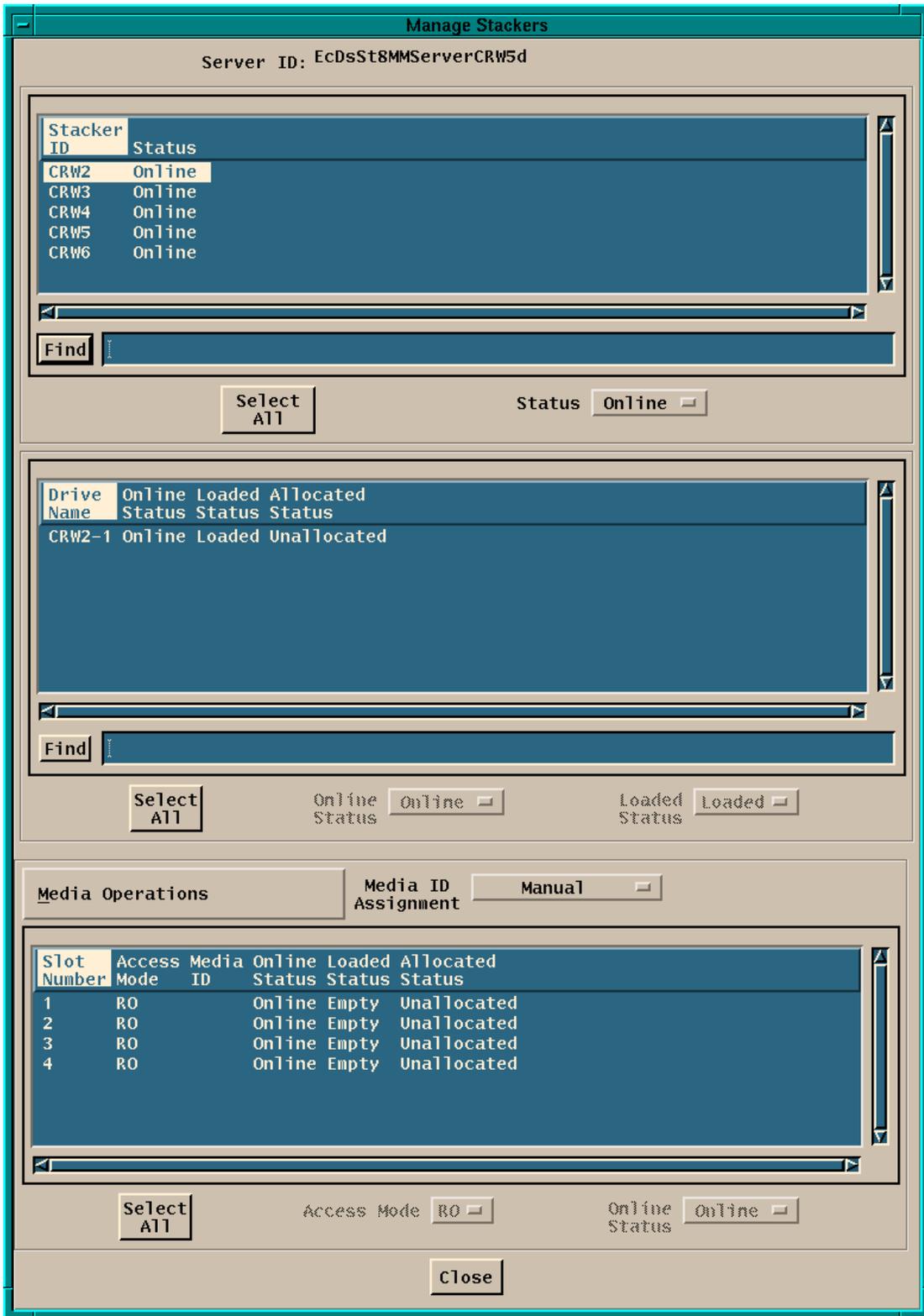


Figure 4.10.2-20. Manage Stackers Pop-up Window

The following functionality is provided through the several buttons present on the Manage Stackers window:

- **Find.** The operator can enter information into the field area to search the list for a specific stacker, device, or piece of media
- **Online Status.** The scheduling of Stackers, Slots and Drives is allowed through **Online Status** buttons. Once a Stacker, Slot or Device has been selected, it can be either put **Online** or taken **Offline** by clicking on the provided button
- **Loaded Status** button allows the operator change the state of a drive to loaded or unloaded. Once a Device has been selected, it can be either marked **Loaded** or marked **Unloaded** by clicking on the provided button
- **Media ID Assignment** If set to **Manual**, the operator must manually enter the media ID for each piece of media loaded. If set to **Auto Increment**, the operator can enter the first media ID, and any additional slots to be filled are assigned media Ids, which sequentially follow the media ID entered by the operator. When using a handheld bar code reader, the **Media ID Assignment** button should be set to **Manual**
- **Access Mode** button allows the operator to change the access mode of a piece of media to read only (RO) or read/write (RW)

Table 4.10.2-20 provides a detailed description of the Manage Stackers window fields.

Table 4.10.2-20. Manage Stackers Pop-up Window Field Descriptions (1 of 2)

Field Name	Data Type	Size	Entry	Description
Stacker ID	Character	N/A	System provided	Identifier of the stacker that is being scheduled
Status	Character	N/A	System provided	Current availability of the stacker (online or offline).
Find	Character	N/A	User Input	Allows the operator to enter search criteria to find a stacker or drive.
Drive Name	Character	N/A	System provided	Identifier of the drive that is being scheduled. The Drive Names are displayed corresponding to the highlighted Stacker ID.
Online Status	Character	N/A	System provided	Current availability of the device or slot (online or offline).
Loaded Status	Character	N/A	System provided	Indicates whether the drive or slot is Occupied or Empty.
Allocated Status	Character	N/A	System provided	Indicates whether or not the drive or slot has been allocated for use by a request.
Slot Number	Integer	N/A	System provided	The slot numbers corresponding to the highlighted Stacker ID.

Table 4.10.2-20. Manage Stackers Pop-up Window Field Descriptions (2 of 2)

Field Name	Data Type	Size	Entry	Description
Access Mode	Character	N/A	System provided	Read-only or read/write.
Media ID	Character	N/A	System provided	Identifier for the piece of media. The barcode printed on the media is typically used as the media identifier.

The Media Operations pull-down menu offers several facilities to the operator for loading and unloading media:

- **Load Media** allows the operator to load one or more pieces of media
- **Unload Media** allows the operator to unload one or more pieces of media
- **Replace Media** allows the operator to both load and unload pieces of media as a single operation
- **Load Media Set** allows the operator to load a group of associated media, which have been identified as a media set. Selecting the **Load Media Set** menu option brings up the popup shown in Figure 4.10.2-21



Figure 4.10.2-21. Load Media Set Pop-up Window

Table 4.10.2-21 provides a detailed description of the Load Media Set window fields.

Table 4.10.2-21. Load Media Set Pop-up Window Field Descriptions

Field Name	Data Type	Size	Entry	Description
Select Media Set	Character	N/A	System provided	Allows the operator to select the media set to be loaded from a pull-down list of available media sets.
Slot Number	Integer	N/A	System provided	The slot numbers corresponding to the highlighted Stacker ID.
Media ID	Character	N/A	System provided	Indicates which media from the media set should be loaded into the associated slot ID.
Access Mode	Character	N/A	Optional	Indicates whether the media is loaded with an Access Mode of read-only or read/write.

4.10.2.2.3.2 Manage Media Sets Screen

Figure 4.10.2-22 is the Manage Media Sets screen.

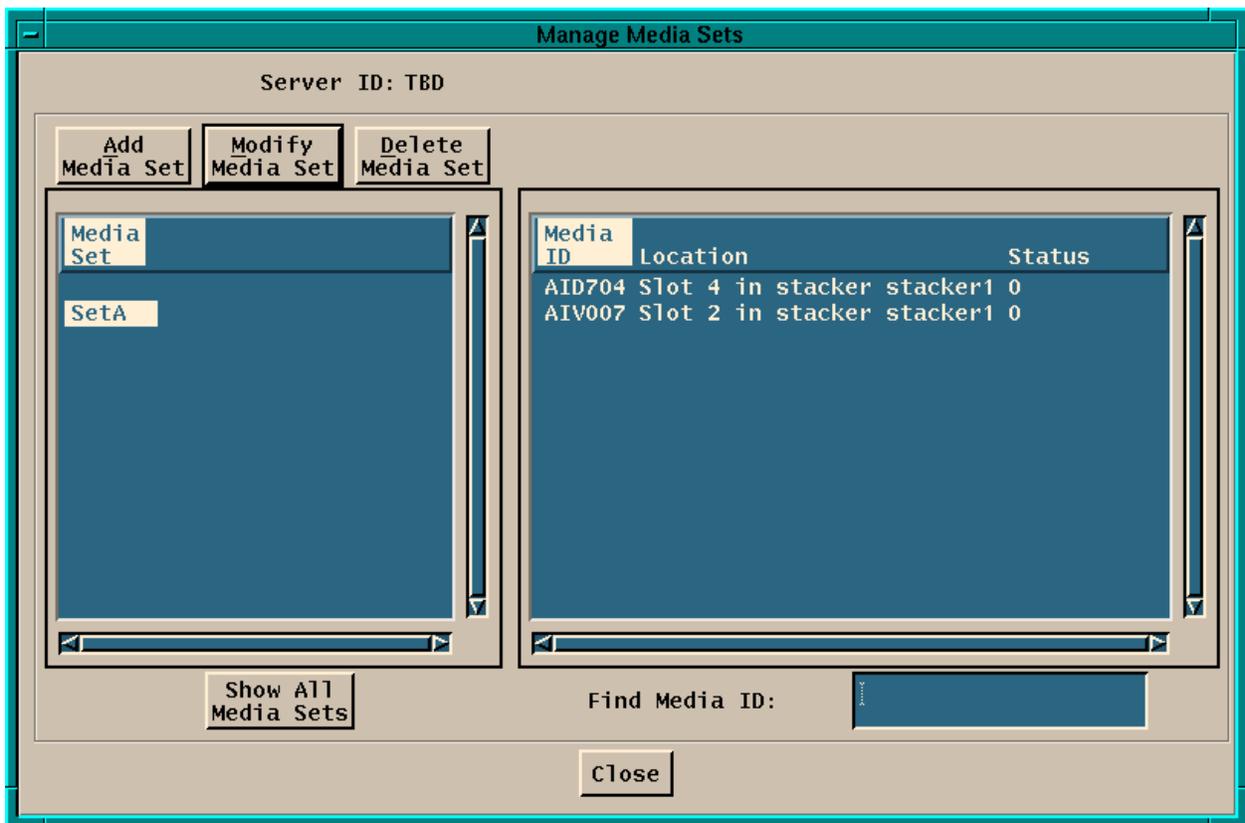


Figure 4.10.2-22. Manage Media Sets Pop-up Window

Table 4.10.2-22 describes the Manage Media Sets window fields.

Table 4.10.2-22. Manage Media Sets Pop-up Window Field Descriptions

Field Name	Data Type	Size	Entry	Description
Media Set	Character	N/A	System provided	The mnemonic identifier used to refer to a logical aggregation of media.
Media ID	Character	N/A	System provided	Identifier for a piece of media in the highlighted media set. The barcode printed on the media is typically used as the media identifier.
Location	Character	N/A	System provided	Location of the media.
Status	Character	N/A	System provided	Status of the media.
Find Media ID	Character	50	Optional	Used to locate a specific piece of media, as well as identifying which request the media is associated with.

The Manage Media Sets window provides three buttons to manage media sets:

- **Add Media Set** - Allows the operator to define a new media set. This brings up the window shown in Figure 4.10.2-23
- **Modify Media Set** - Allows the operator to add to or remove from the media in the selected media set. This also brings up the window shown in Figure 4.10.2-23
- **Delete Media Set** - If a media set is selected, this option permits the operator to remove the definition for the selected media set

Whether adding or modifying a defined media set, the window shown below in Figure 4.10.2-23 is displayed.

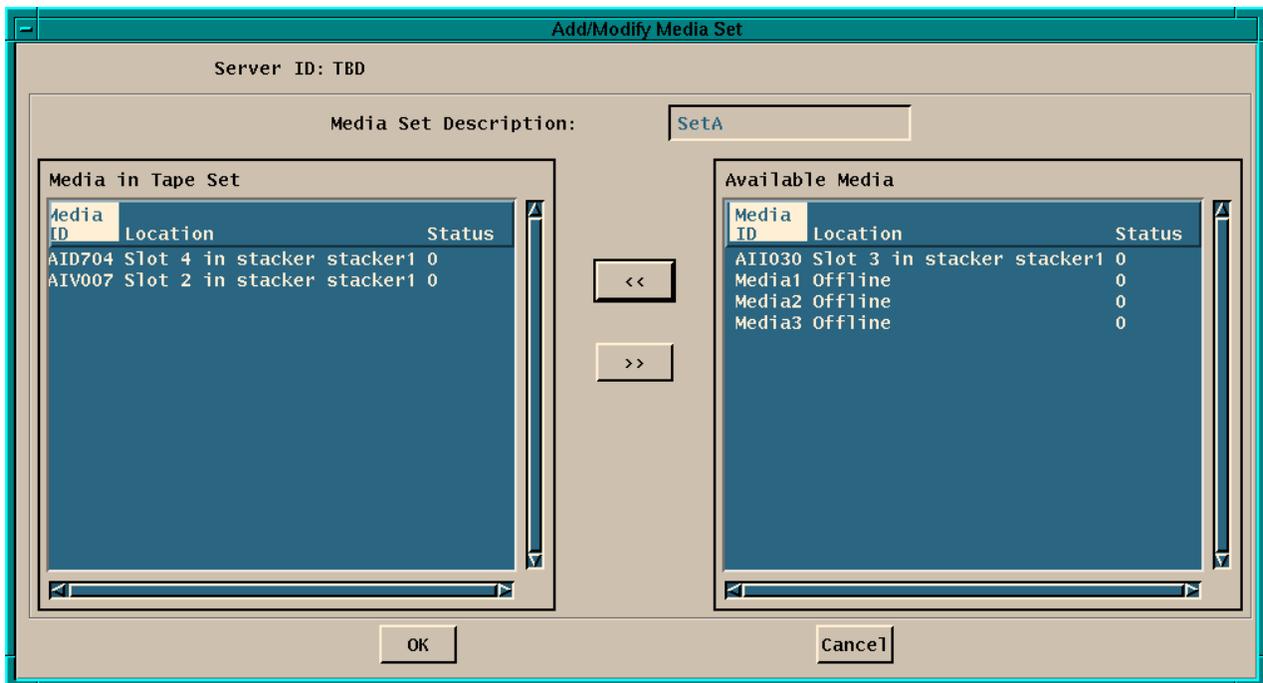


Figure 4.10.2-23. Add/Modify Media Set Pop-up Window

Table 4.10.2-23 describes the Add/Modify Media Set window fields.

Table 4.10.2-23. Add/Modify Media Set Pop-up Window Field Descriptions

Field Name	Data Type	Size	Entry	Description
Media Set Description	Character	N/A	System provided	The mnemonic identifier used to refer to a logical aggregation of media.
Media ID	Character	N/A	System provided	Identifier for a piece of media in the media set. The barcode printed on the media is typically used as the media identifier.
Location	Character	N/A	System provided	Location of the media.
Status	Character	N/A	System provided	Status of the media.

The << and >> buttons permit media to be moved between the list of media in the named media set and the list of available media which are not in the media set.

4.10.2.2.4 Cache Monitoring (Cache Stats.) Tab

This tab displays all the files in the selected cache, which can be a Read-Only Cache or the Pull Area (see Figure 4.10.2-24).

It reports general statistics on the entire cache and allows the operator to delete expired liens against files in the cache area. If the cache reaches an operator-configured threshold, the operator is warned with a message in the operator messages area. If the operator does not delete expired liens and allows the cache to fill, the server is not able to copy new files to the cache area.

When the **Mark Delete** button is pressed, all liens against the selected item in the list are marked for deletion. Multiple items can be selected. When the **Unmark Delete** button is pressed the delete flag is removed for all items selected in the list.

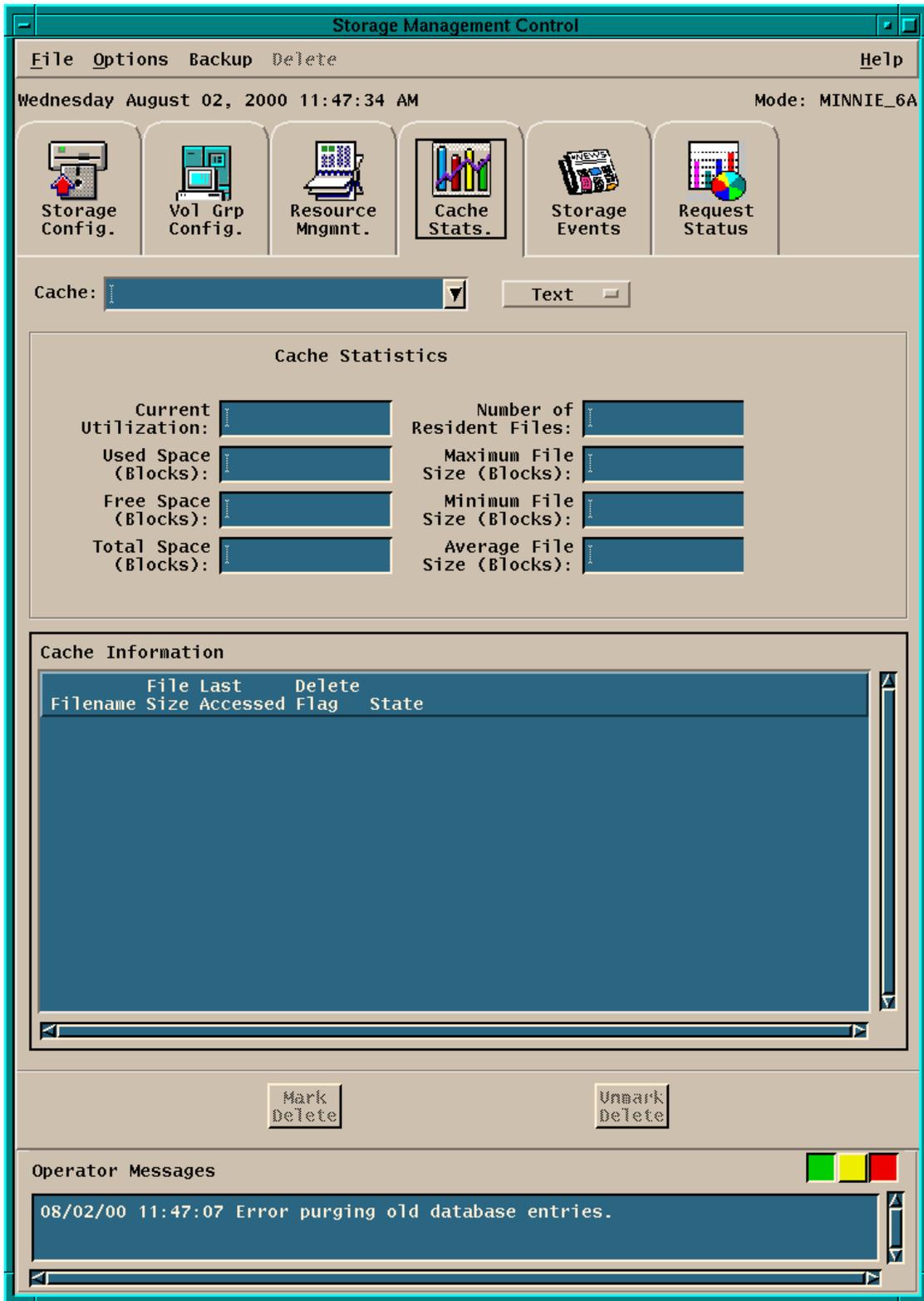


Figure 4.10.2-24. Cache Monitoring Tab (Cache Stats.)

Table 4.10.2-24 provides the detailed descriptions of the fields shown in the Cache Monitoring Tab Storage Management Control window.

Table 4.10.2-24. Cache Monitoring Tab Field Descriptions

Field Name	Data Type	Size	Entry	Description
Cache	Character	N/A	System generated	Cache Manager Server Names.
Current Utilization	Float	N/A	System generated	Percent of cache space that is full.
Used Space (Blocks)	Integer	N/A	System generated	Amount of space in cache that is being used.
Free Space (Blocks)	Integer	N/A	System generated	Amount of space in cache that is free.
Total Space (Blocks)	Integer	N/A	System generated	The total space in the cache.
Number of Resident Files	Integer	N/A	System generated	The number of files in the cache.
Maximum File Size (Blocks)	Integer	N/A	System generated	The size of the largest file in the cache.
Minimum File Size (Blocks)	Integer	N/A	System generated	The size of the smallest file in the cache.
Average File Size (Blocks)	Integer	N/A	System generated	The average size of the files in the cache.
Cache Information	--	--	--	Information about files kept in the Cache Server.
Filename	Character	N/A	System generated	File name, which is cached.
File Size	Integer	N/A	System generated	Size of cached file.
Last Accessed	Date/time	N/A	System generated	Latest date and time in which cached file has been accessed.
Delete Flag	Character	N/A	System generated	Flag indicating if cached file has been deleted (Y/N).
State	Character	N/A	System generated	State of the cached file (Present/standby).
Operator Messages	Character	N/A	System generated	Location where informational, warning or error messages are displayed.

4.10.2.2.5 Storage Events Tab

This tab, shown in Figure 4.10.2-25, allows the operator to search for events in the Event Log. Various search parameters are shown in the Event Log Search Parameter box. When the **Search** button is selected, the results are shown in the Event log box.

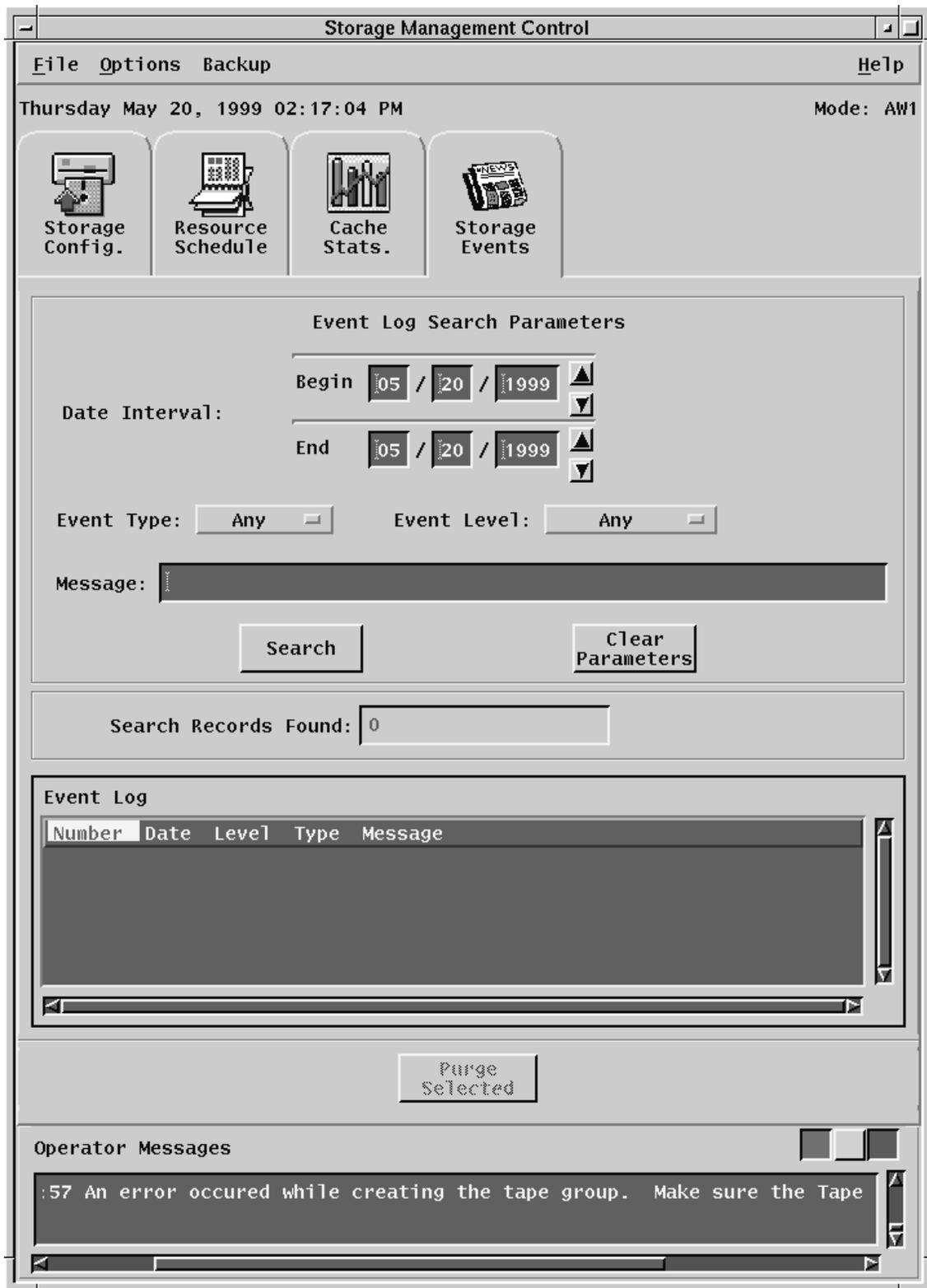


Figure 4.10.2-25. Storage Events Tab

The **Clear Parameters** button deletes entries made in the Event Log Search Parameters data fields. The **Purge Selected** button, when selected, deletes the entries in the Event log, which are highlighted (“Selected”) by the operator. A context sensitive **Help** button provides the operator with information on the listed events,

Table 4.10.2-25 provides the detailed descriptions of the Event Log fields.

Table 4.10.2-25. Event Log Field Descriptions

Field Name	Data Type	Size	Entry	Description
Date Interval	Date/time	21	System generated	Date and time event was entered into the event log.
Event Type	Character	10	System generated	Classification of event into various defined (TBS) types.
Event Level	Character	11	System generated	Classification of event into various defined (TBS) levels.
Message	Character	255	System generated	Message as entered in the Event Log.
Event Log Number	Integer	4	System generated	Sequence number of the Event Log.
Event Log Date	Date/time	8	System generated	The date and time the event was generated.
Event Log Level	Integer	11	System generated	Identify the category level of the event (information, warning, error, severe, fatal, unknown, or any).
Event Log Type	Character	10	System generated	Where the Event Log is generated (device, cache, software, COTS, Sybase, Pulldisk, any, or unknown).
Event Log Message	Character	255	System generated	Message as entered in the Event Log.
Operator Messages	Character	n/a	System generated	System will generate any necessary messages for the operator to read

Note: the same fields are reported in the “Operator Messages” box in Figure 4.10.2-23. The data type, size, and the description are the same while the entry is “optional” based on the search criteria that the operator uses.

4.10.2.2.6 Server Monitoring (Request Status) Tab

This tab, shown in Figure 4.10.2-26, allows the operator to monitor processing activity in all of the Storage Management servers for a given mode. The main table lists the requests, which are currently being serviced by Storage Management servers, and those, which completed in the last 24 hours. Four filters are provided via the pull-down menu to control how much information is displayed, so that the operator can focus in on particular areas.

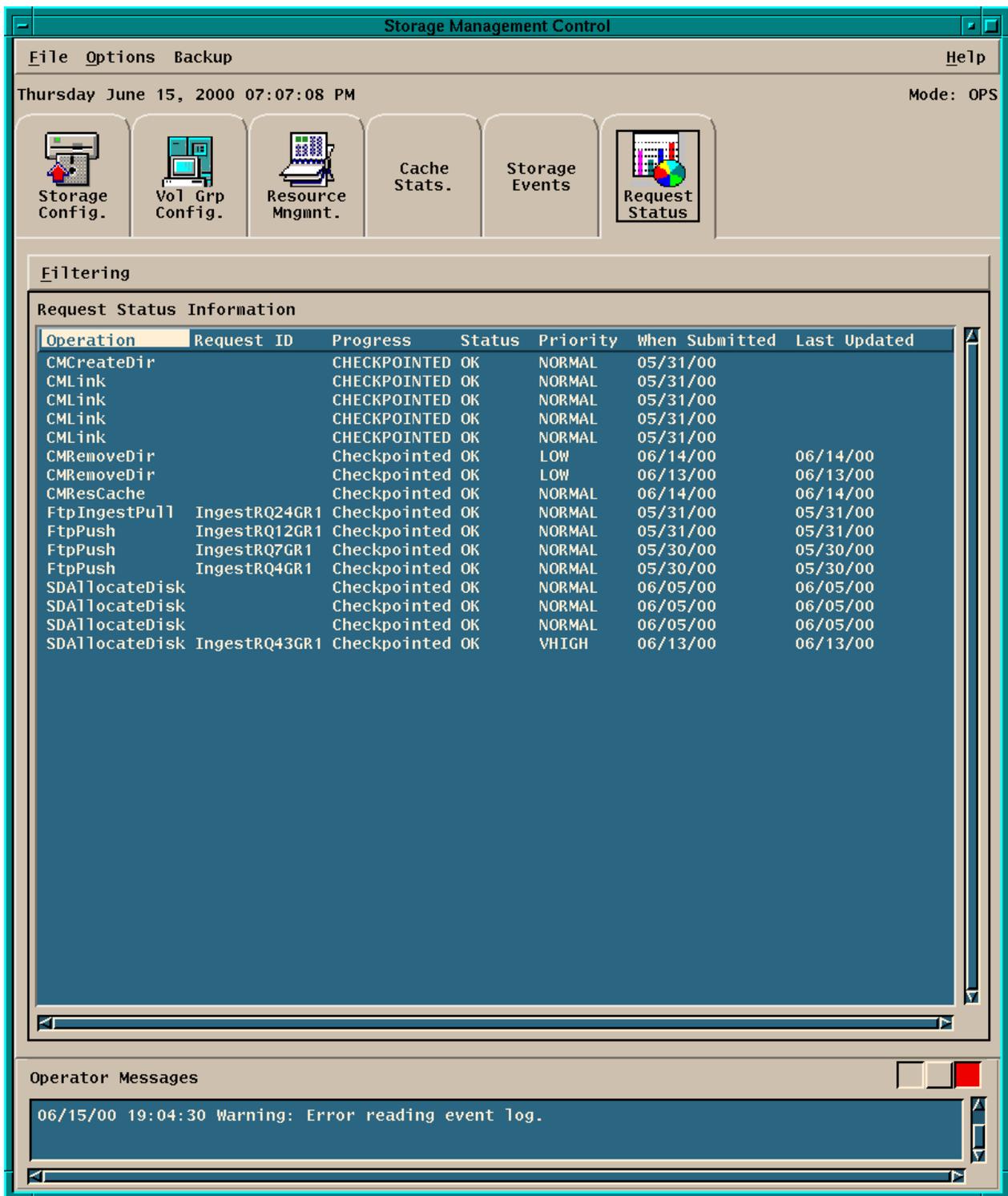


Figure 4.10.2-26. Server Monitoring (Request Status) Tab

The filtering menu provides four filtering options:

- **Server** The Server filter controls what activity is displayed by limiting the list to the requests being/having been serviced by a specific server. Selecting “All” displays all requests throughout the Storage Management CI
- **Operation** The Operation filter allows the operator to focus on a specific type of operation. The list of operations is dynamically generated to reflect those operations for which requests are currently in queue
- **Processing State** The Processing State filter allows the operator to differentiate between requests which are being actively processed; which have been completed, either to successful completion or a retryable error state; or which are suspended, awaiting the outcome of another event
- **Submitter** The Submitter filter allows the operator to see the status of requests submitted by a specific client process. The list of possible clients is dynamically generated to reflect the list of clients with outstanding requests

The main area of the Server Activity screen provides a tabular listing of the requests in the system, as constrained by the various pull-down filters. The columns shown are described in Table 4.10.2-26. Clicking on any of the column headers sorts the listed requests in order by the column selected. For example, clicking on the Last Updated column header lists the requests in order from the least recently updated to the most recently updated.

Table 4.10.2-26. Server Monitoring Tab Column Description

Column Heading	Description
Operation	The type of operation represented by the request. For example, an ArStore operation refers to a Store operation being serviced by an Archive Server.
Request ID	The identifier by which a request can be recognized by operations staff. Right clicking on a request ID displays the full RPC ID associated with the request.
Progress	The stage of processing which the request is currently working. For operations, which provide incremental progress (e.g., file copies), this can include a numeric progress indication (e.g., 100 of 750 KB).
Status	The status of the request. OK indicates a request which has completed successfully.
Priority	The priority of the request. Requests can have any of the following priorities: LOW, MEDIUM, HIGH, VHIGH, and XPRESS.
When Submitted	The time and date the Storage Management server received the request, which is responsible for the request.
Last Updated	The time and date when the status was last updated for the request.
Operator Messages	System will generate any necessary messages for the operator to read.

4.10.2.2.7 Storage Management Control Menu Screens Available

The **Restart Backup** pop-up window, shown in Figure 4.10.2-27, appears by selecting the **Restart** option on the **Backup** menu. This pop-up allows for restarting a backup.

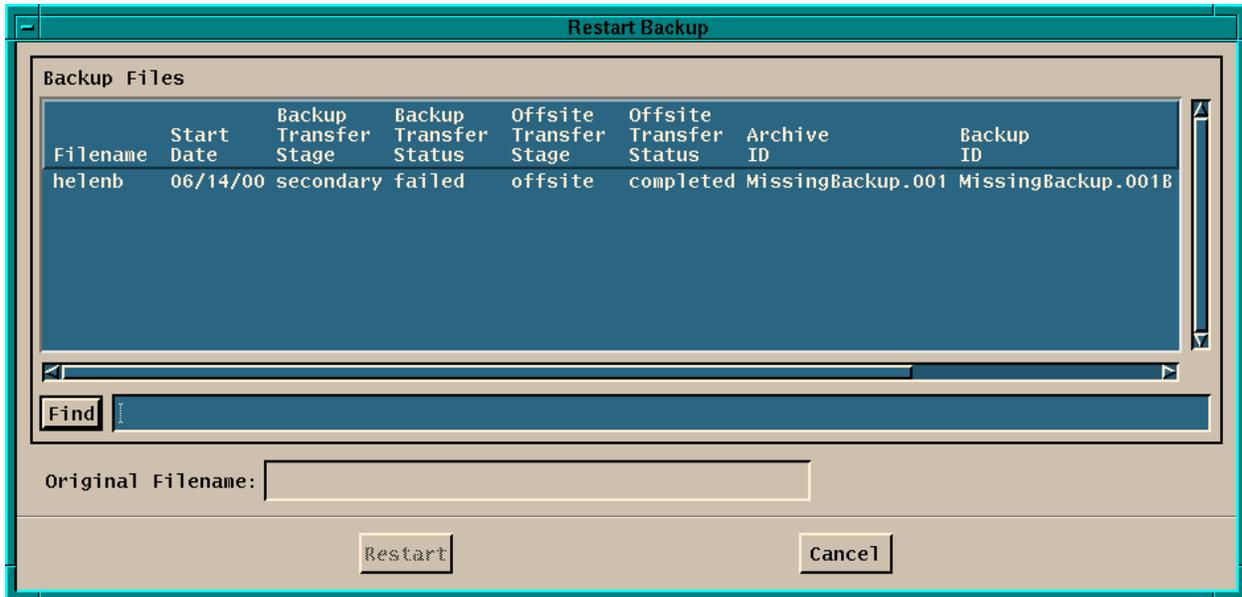


Figure 4.10.2-27. Restart Backup Pop-up Window

Table 4.10.2-27 provides the detailed descriptions for the Restart Backup window fields.

Table 4.10.2-27. Restart Backup Pop-up Window Field Descriptions

Field Name	Data Type	Size	Entry	Description
Filename	Character	100	System generated	File name, which uniquely identifies file.
Start Date	Date/time	21	System generated	Date and time at the start of the file. GMT in the format: mm/dd/YY hh:mm:ss.sss
Backup Transfer Stage	Character	9	System generated	Stage of file transfer with respect to the backup volume group.
Backup Transfer Status	Character	9	System generated	Status of file transfer with respect to the backup volume group.
Offsite Transfer Stage	Character	9	System generated	Stage of file transfer with respect to the offsite volume group.
Offsite Transfer Status	Character	9	System generated	Status of file transfer with respect to the offsite volume group.
Archive ID	Character	30	System generated	Identification of data in Archive files.
Backup ID	Character	30	System generated	Identification of data in Backup files.
Find	Character	255	Optional	Allows the operator to perform a keyword search for items in the Device Name field
Original Filename	Character	200	System generated	Real file name (including full path to files).

The screen, shown in Figure 4.10.2-28, is displayed when the **Restart** button is pressed from the Restart Backup window.

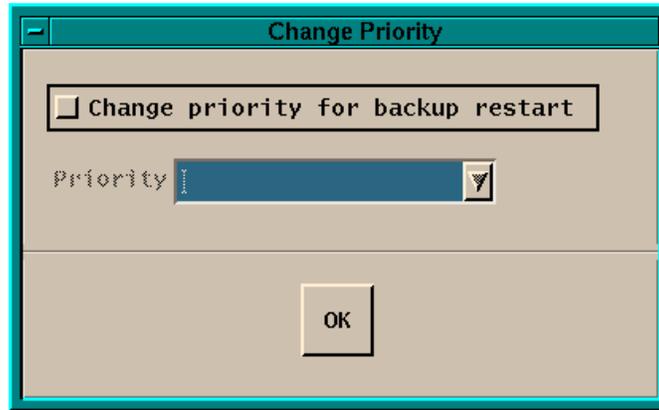


Figure 4.10.2-28. Change Priority Pop-up Window

Table 4.10.2-28 describes the Change Priority window fields.

Table 4.10.2-28. Change Priority Pop-up Window Field Descriptions

Field Name	Data Type	Size	Entry	Description
Change priority for backup restart	Toggle button	N/A	Optional	Allows the operator to change the priority before restarting backup.
Priority	Character	6	Optional	Backup can have any of the following priorities: LOW, MEDIUM, HIGH, VHIGH and XPRESS.

Selecting the **System Settings** menu function from the **Options** menu shows the pop up window allowing the operator to select the polling rates for various GUI displays – see Figure 4.10.2-29 (Polling Rate Selection).

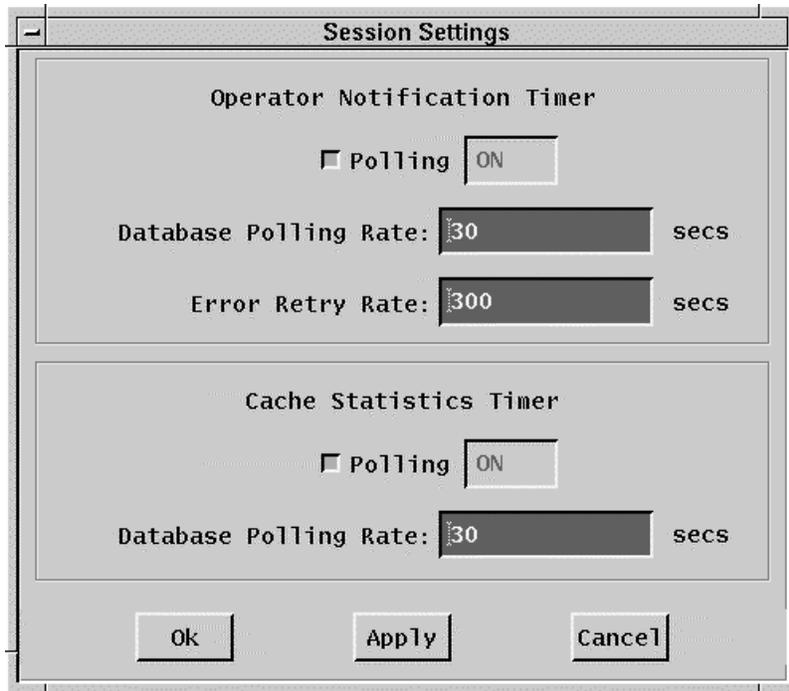


Figure 4.10.2-29. Polling Rate Selection Pop-up Window

Table 4.10.2-29 provides the descriptions of the Polling Rate Selection Pop-up window fields.

Table 4.10.2-29. Polling Rate Selection Pop-up Window Field Descriptions

Field Name	Data Type	Size	Entry	Description
Polling	Character	3	System provided	The current status of polling.
Database Polling Rate	Integer	Unlimited	Optional	Rate at which the event log is updated.
Error Retry Rate	Integer	Unlimited	Optional	Rate at which an update of the event log is attempted after an error condition is detected.
Database Polling Rate	Integer	Unlimited	Optional	Rate at which the cache statistics are updated.

The **Ok** button implements the new selections and closes the pop-up window.

The **Apply** button applies the new selections entered by the user to the window.

The **Cancel** button ignores the changes and closes the window.

4.10.2.3 Required Operating Environment

For information on the operating environment, tunable parameters, and environment variables of the Storage Management Control Tool refer to the 920-TDA-022 “Custom Code Configuration Parameters” documentation series.

Table 4.10.2-30 identifies the supporting products this tool depends upon to function properly.

Table 4.10.2-30. Support Products for Storage Management Control

Product Dependence	Protocol Used	Comments
Sybase	SQL	Checkpoint, configuration and request management for the server
Clients/GUIs	Socket	To transfer data and information

4.10.2.4 Databases

The Storage Management Tool uses data that is provided by the Storage Management Database. For details about the Storage Management Database, please refer to DID 311-CD-625, *Storage Management Database Design and Schema Specifications*.

4.10.2.5 Special Constraints

AMASS has to be up in order for the Storage management Tool to archive and retrieve files. Moreover, the FTP server on the ACP host has to have debug logging enabled to write filenames of the pulled files to syslog for pull notification to the pull monitor.

4.10.2.6 Outputs

None

4.10.2.7 Event and Error Messages

See Appendix A, for event and error messages.

4.10.2.8 Reports

None

This page intentionally left blank.

4.10.3 Data Distribution Requests GUI

The Data Distribution software is capable of delivering ordered ECS product data electronically using FTP push/pull and SecureCopy. With the introduction of the Product Distribution System (PDS) software in ECS Release 6A, the Data Distribution software is configured to handle only electronic FTP delivery requests. All hard media delivery requests are handled by PDS, which is described in Section 4.10.5.

The Data Distribution Requests GUI displays detailed information on individual data distribution requests and provides the capability to filter requests, change the priority of requests, and mark requests as shipped. The GUI's main window provides operations personnel at the DAACs the capability for managing data distribution requests. A summary of the functions performed by the Data Distribution Requests GUI is given in Table 4.10.3-1.

Table 4.10.3-1. Common ECS Operator Functions Performed with the Data Distribution GUI

Operating Function	Tab	Description	When and Why to Use
Manage Data Distribution Request Activities	Distribution Requests Tab	Allows the operators to view and track data distribution requests	As required to monitor detailed information on data distribution request activities, change priority of requests, and mark requests as shipped, suspend/resume selected requests, suspend/resume all requests, and filter requests.

4.10.3.1 Quick Start Using Data Distribution

To start the Data Distribution Requests GUI, enter the following command line:

>**EcDsDdistGuiStart** <mode> where:

<mode> is the ECS mode for the execution (e.g., **OPS, TS1 or TS2**).

4.10.3.2 Data Distribution Main Screen

The Data Distribution Requests GUI Main Screen has five tabs:

- The **Data Distribution (Distrib'n) Requests** tab provides the functionality needed to track the activity related to product distribution requests
- The **System Requests** tab has not yet been defined as of ECS Release 7 delivery. Request level information and action for requests not in the DDIST CSCI in the DSS can now be obtained from or performed through the Synergy Order Manager
- The **Tape Ids** tab allows hard media such as tapes and CD ROMs to be searched from the Distribution list based on the ID or Distribution Request Number
- The **Preamble Editor** tab allows the email and packing list headers to be edited
- The **Event Logging** tab has not yet been defined as of ECS Release 7 delivery

The following sub-sections describe the use of these tabs.

4.10.3.2.1 The Data Distribution Requests Tab

The Data Distribution Requests Tab, shown in Figures 4.10.3-1, is the default tab appearing when the Data Distribution Requests GUI is invoked.

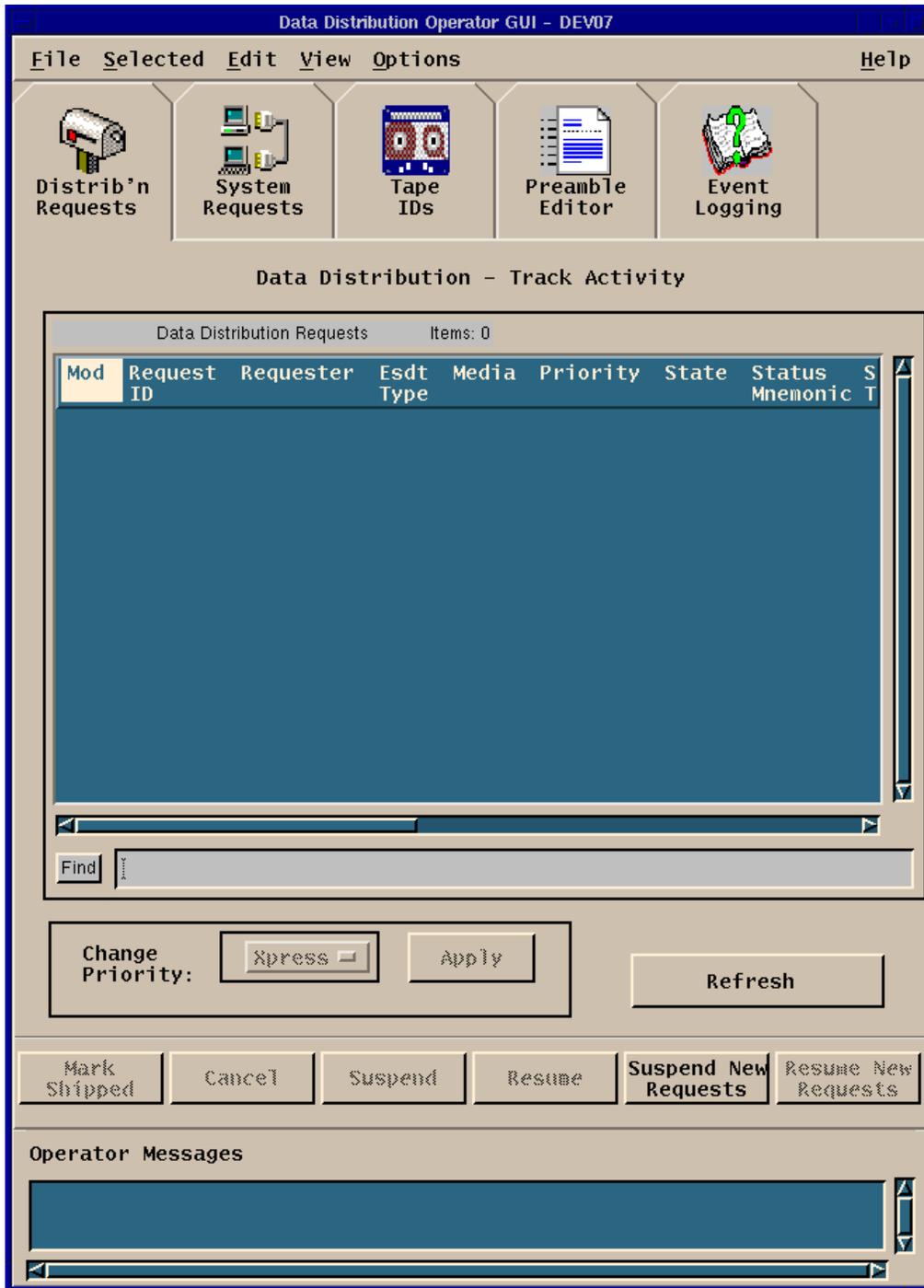


Figure 4.10.3-1. Data Distribution Main Screen showing Data Distribution Request Tab (1 of 3)

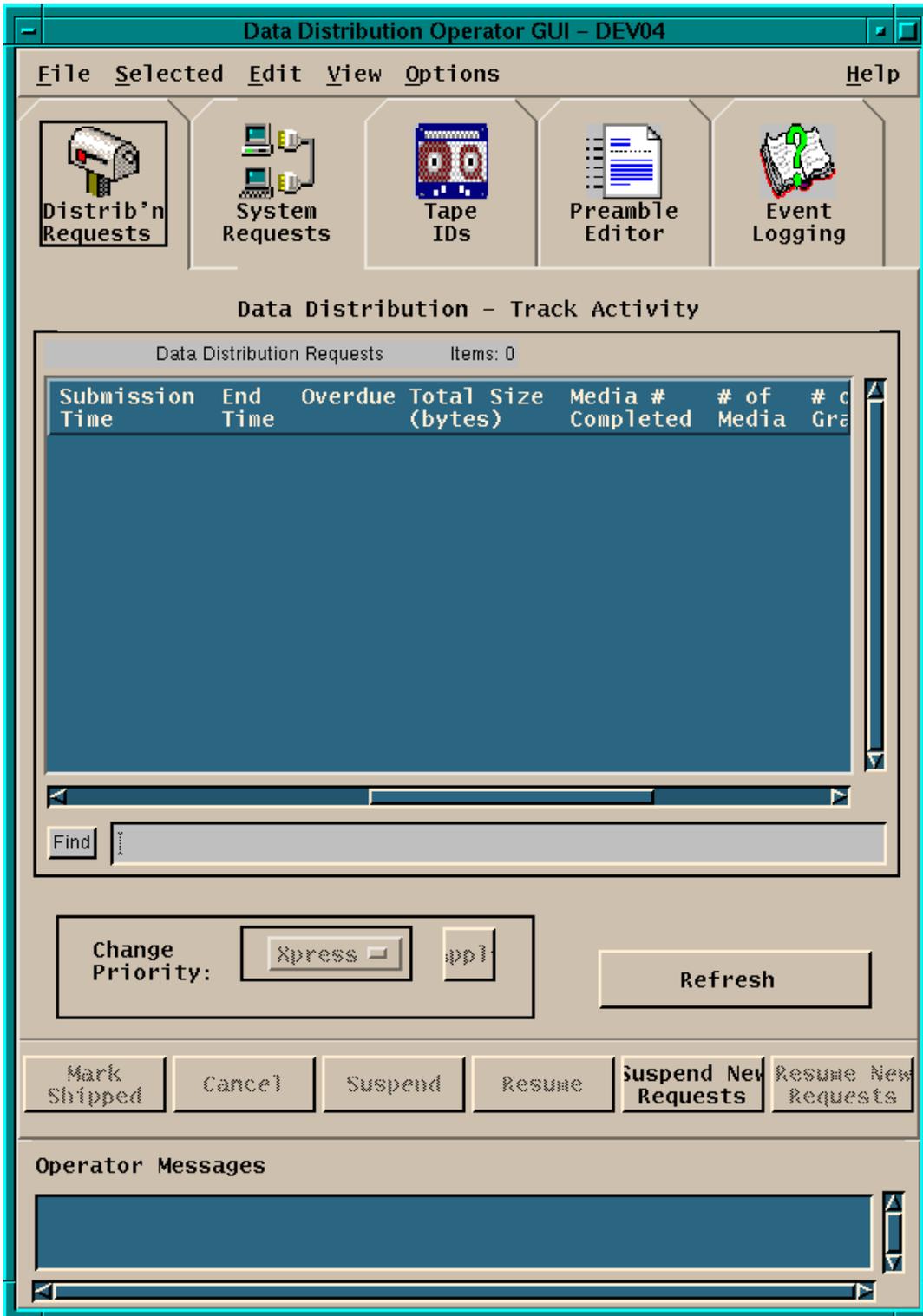


Figure 4.10.3-1. Data Distribution Main Screen showing Data Distribution Request Tab (2 of 3)

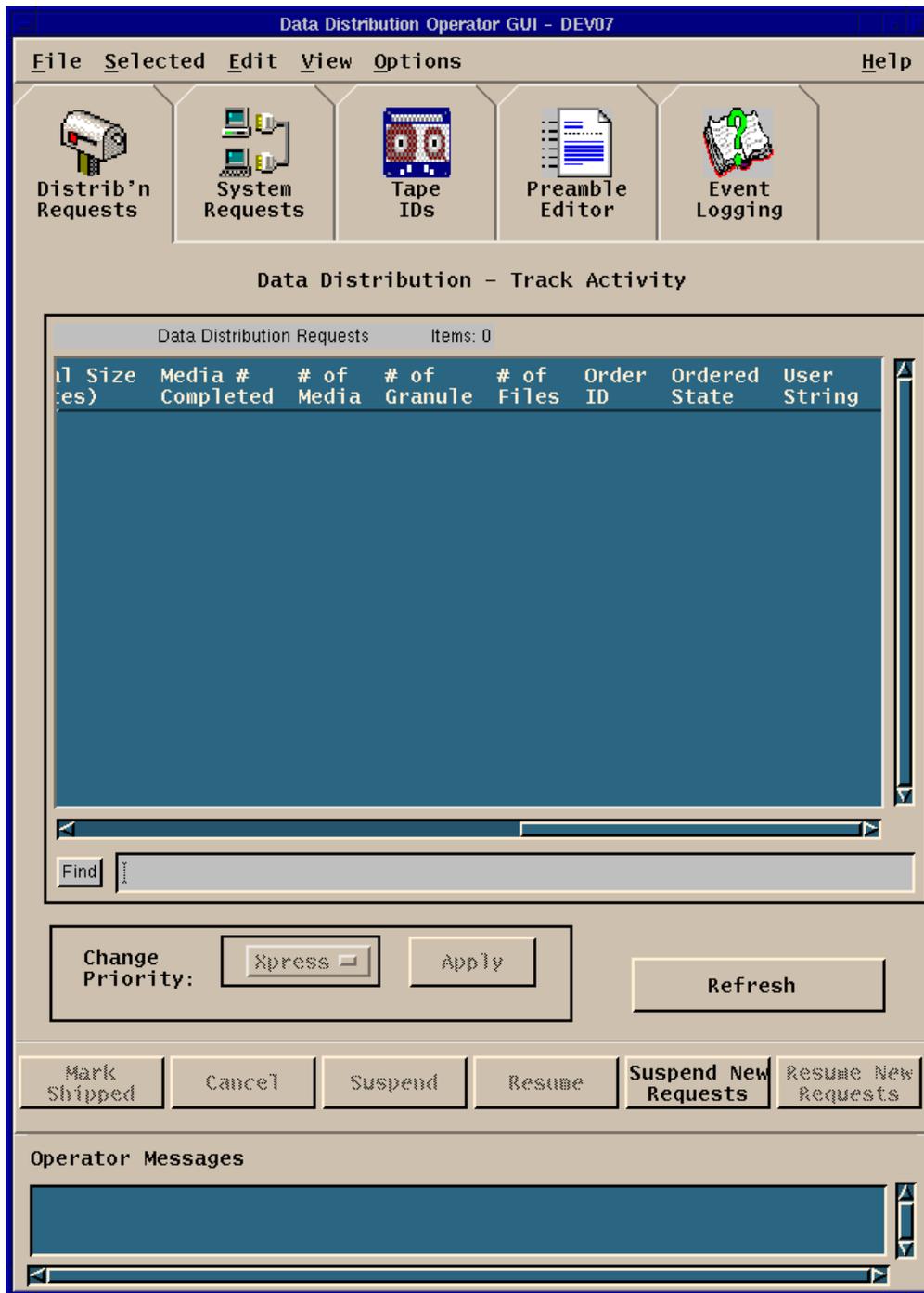


Figure 4.10.3-1. Data Distribution Main Screen showing Data Distribution Request Tab (3 of 3)

The Data Distribution Request tab displays data distribution requests. The major component is the Track Activity panel, which lists the data distribution requests currently being handled by the Data Distribution server. The total number of requests is displayed at the top of the panel in the "items" field. Several parameters associated with each individual request are displayed. The list can be sorted by column. All of the parameters included for each request are identified and described in Table 4.10.3-2.

Table 4.10.3-2. Data Distribution - Track Activity Panel Field Description (1 of 2)

Field Name	Data Type	Size	Entry	Description
MOD	Boolean	1	System generated	Checkmark that shows which requests have been selected and/or modified by the operator during the current session.
Request ID	Character	Unlimited	System generated	Unique identifier for the request.
Requester	Character	Unlimited	System generated	Identifies the user that submitted the request.
ESDT Type	Character	Unlimited	System generated	Identifies the science instrument/data provider type.
Media	Character	Unlimited	System generated	Type of media to be used for distribution. Values are read from the Registry.
Priority	Character	Unlimited	System generated	Priority at which the distribution request is processed relative to other distribution requests, Normal is its default value. Other Values are: Xpress, Vhigh, High and Low.
State	Character	Unlimited	System generated	Request states are: pending, active, staging, waiting for shipment, shipped, canceled, transferring, suspended, suspended with errors.
Status Mnemonic	Character	Unlimited	System generated	Displays a small message that indicates there is an operator message attached to the request.
Submission Time	Date/time	19	System generated	Time when the submit service was invoked upon the request. The time is standard GMT. The format used is: mm/dd/yyyy hh:mm:ss.
End Time	Date/time	19	System generated	Time when the distribution request has been satisfied. Time is in standard GMT, the format is mm/dd/yyyy hh:mm:ss.
Overdue Total Size (bytes)	Integer	Unlimited	System generated	Total size in bytes of the data to be distributed in the request.
Media # Completed	Integer	Unlimited	System generated	Number of media that have been already filled up by the distribution request that is being processed.
# of Media	Integer	Max # of Media configurable	System generated	Number of Media that need to be used to completely fulfill a (media) distribution request if it were not to be compressed.

Table 4.10.3-2. Data Distribution - Track Activity Panel Field Description (2 of 2)

Field Name	Data Type	Size	Entry	Description
# of Granule	Integer	Unlimited	System generated	Number of granules comprising the distribution request.
# of Files	Integer	Unlimited	System generated	Number of files in the distribution request.
Order ID	Character	Unlimited	System generated	The unique order ID that the entire data server uses for identifying the distribution request.
Ordered State	Character	20	System generated	Request State can be changed directly by the operator by means of the button provided on the Track Activity screen (see below). See "State" field for values.
User String	Character	255	System generated	Contains original user request ID. Used to correlate a DN with original request.

The operator can select from the menu bar items at the top of the Data Distribution GUI window for getting help and activating less-frequently used secondary functions. The menu bar capability is available on all Data Server GUI screens. The following menus are available:

- **File** includes the following items:
 - **View** opens a text viewer window
 - **Open Terminal** opens an Xterm window
 - **Save As...** saves the current contents of the Data Distribution Requests table, Tape Request and Tape ID tables, or the Preamble Text editor to a text file
 - **Print** sends the current contents of the Data Distribution Requests table, Tape Request and Tape ID tables, or the Preamble Text editor to the default printer
 - **Exit** (Ctrl-Q) exits the application (graceful exit)
- **Selecte**d opens a menu comprising the following items:
 - **Select All** selects all the requests on the main screen
 - **Deselect All** unselects all the requests on the main screen
- **Edit** that includes the following items:
 - **Cut** removes the selected text from the Preamble Text editor and puts in the X Windows clipboard
 - **Copy** places a copy of the selected text from the Preamble Text editor into the X Windows clipboard
 - **Paste** inserts the text on the X Windows clipboard into the selected area of the Preamble Text editor
- **View** includes the following items:

- **Refresh** redraws the window
- **Filter** opens the Filter Control window
- **Detailed** sends the detailed information of the selected distribution request to the operator messages text field
- **Options** includes the following items:
 - **System Settings** opens the Refresh Options window where the operator is given the option to toggle the **DDist base polling** *On* and *Off* through the provided toggle button (see Figure 4.10.3-2). In case the operator decided to have the polling of the Data Distribution Database *On*, the polling rate can be edited. Details on the parameters that can be input by the operator in the Refresh Options screen are provided in Table 4.10.3-3.
 - **Verify Connection** checks the connections to the distribution server, and sends the connection status to the operator message text field
 - **Reconnect** attempts to reestablish communications to the Distribution server
 - **Help** provides on-line help to the operator.

Table 4.10.3-3. Refresh Options Field Description

Field Name	Data Type	Size	Entry	Description
DDist Polling Rate	Integer	0-9999	Optional	Allows the operator to specify the polling rate in seconds for updating the Task Activity Window (default is 30 seconds).
Error Retry Rate	Integer	0-9999	Optional	Time in seconds the system is going to wait before trying to poll the Data Server, after a failed attempt.

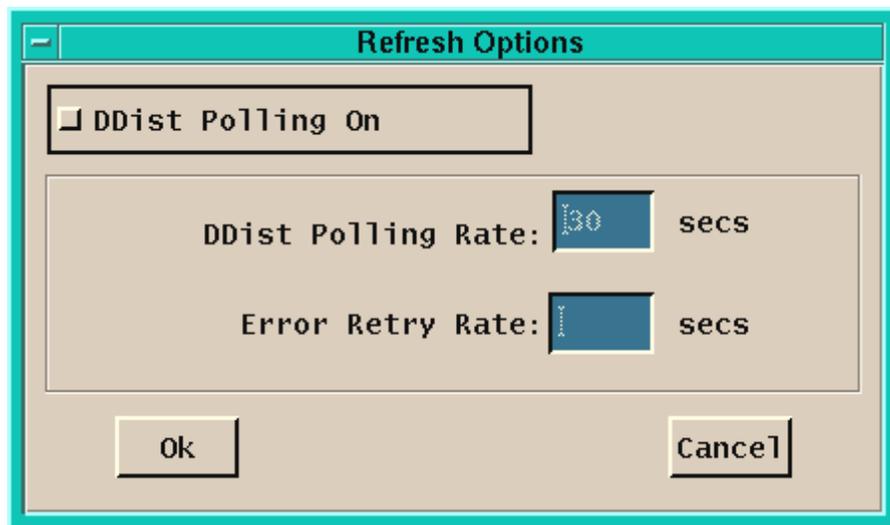


Figure 4.10.3-2. Refresh Options Window

The Data Distribution Tab includes additional functionality associated with the following buttons:

- **Apply** allows the operator to change the priority of the distribution requests selected in the Track Activity panel. Available selections are Xpress, Vhigh, High, Normal (default), and Low. The priority selection is handled through a pull down menu
- **Mark Shipped** allows the operator to change the state of the selected Hard Media distribution request from waiting for shipment to shipped when the request has actually been shipped
- **Filter (from the Options menu)** brings up the Filter Request Dialog (see Figure 4.10.3-3), which provides a selection of attributes from the list of distribution requests on which to filter. Filter on **Request ID** and **Requester** is done by selecting the corresponding toggle button and entering the desired information. Selecting the **All Requests** radio button returns to the original state of the request list. Further request filtering is allowed by selecting one or more media type list entries. The operator filters on all Media types by clicking the **All** button or clears all selected filters by clicking on the **None** button. .Options for the Media Type are determined by entries within the Registry.

Filtering is also allowed based on possible states of the request by selection through the available radio buttons in the **State:** panel. By clicking on **All**, the operator can filter on all possible states. All selected filters can be cleared by clicking on the **None** button. Selectable states include: Pending, Active, Staging, Transferring, Cancelled, Suspended, Suspended with Errors, Waiting for Shipment, and Shipped.

In addition, the following pushbuttons are available:

- **OK** applies all selected filters and closes the filter dialog
- **Apply** implements all filters and keeps the filter dialog open (in case other filtering needs to be done.)
- **Cancel** closes the filter dialog without applying the selected filters
- **Help** displays on-line help information

Table 4.10.3-4 describes the Data Distribution - Filter Requests Dialog.

<input type="checkbox"/> Request ID	<input type="text"/>
<input type="checkbox"/> Requester	<input type="text"/>
<input type="checkbox"/> All Requests	

Media Type:

SMM CDROM D3 DLT FtpPull FtpPush	<input type="button" value="All"/>
	<input type="button" value="None"/>

State:

<input type="checkbox"/> Pending	<input type="checkbox"/> Suspended
<input type="checkbox"/> Active	<input type="checkbox"/> Suspended with Errors
<input type="checkbox"/> Staging	<input type="checkbox"/> Waiting for Shipment
<input type="checkbox"/> Transferring	<input type="checkbox"/> Shipped
<input type="checkbox"/> Cancelled	<input type="checkbox"/> Failed
<input type="button" value="All"/>	<input type="button" value="None"/>

<input type="button" value="OK"/>	<input type="button" value="Apply"/>	<input type="button" value="Cancel"/>	<input type="button" value="Help"/>
-----------------------------------	--------------------------------------	---------------------------------------	-------------------------------------

Figure 4.10.3-3. Data Distribution - Filter Requests Dialog

Table 4.10.3-4. Data Distribution - Filter Requests Field Description

Field Name	Data Type	Size	Entry	Description
Request ID	Character	Unlimited	Operator Selected	Unique identifier for the request.
Requester	Variable character	Unlimited	Operator Selected	Identifies user that submitted the request.
All Requests	Boolean	1	Operator Selected	When toggled ON, all requests are displayed.
Media Type	Character	Unlimited	Operator Selected	Request(s) with media attribute among the selected types are added to filtered list.
State	Boolean	1	Operator Selected	Request(s) with State attribute within the toggled ON states are added to the filtered list.

Back to the Data Distribution Tab, the following additional buttons are also available:

- **Refresh** updates the Data Distribution Request screen with the most recent list of requests
- **Cancel**, **Suspend** and **Resume** allow the operator to, respectively, cancel, suspend or resume the requests selected in the Track Activity list
- Finally the **Suspend New Requests** and **Resume New Requests** buttons suspend all and resume all requests currently present in the Data Distribution server
- **Operator Messages**: any error encountered during an operation to a request in the list is displayed in the operator messages window at the bottom of the screen

4.10.3.2.2 System Requests Tab

The functionality associated with the System Requests tab shown in Figure 4.10.3-4 is not yet defined, as of ECS Release 7 delivery. However, request level information and action for requests not in the DDIST CSCI in the DSS can now be obtained from or performed through the Synergy Order Manager. See Sections 4.11.15 and 4.11.16.

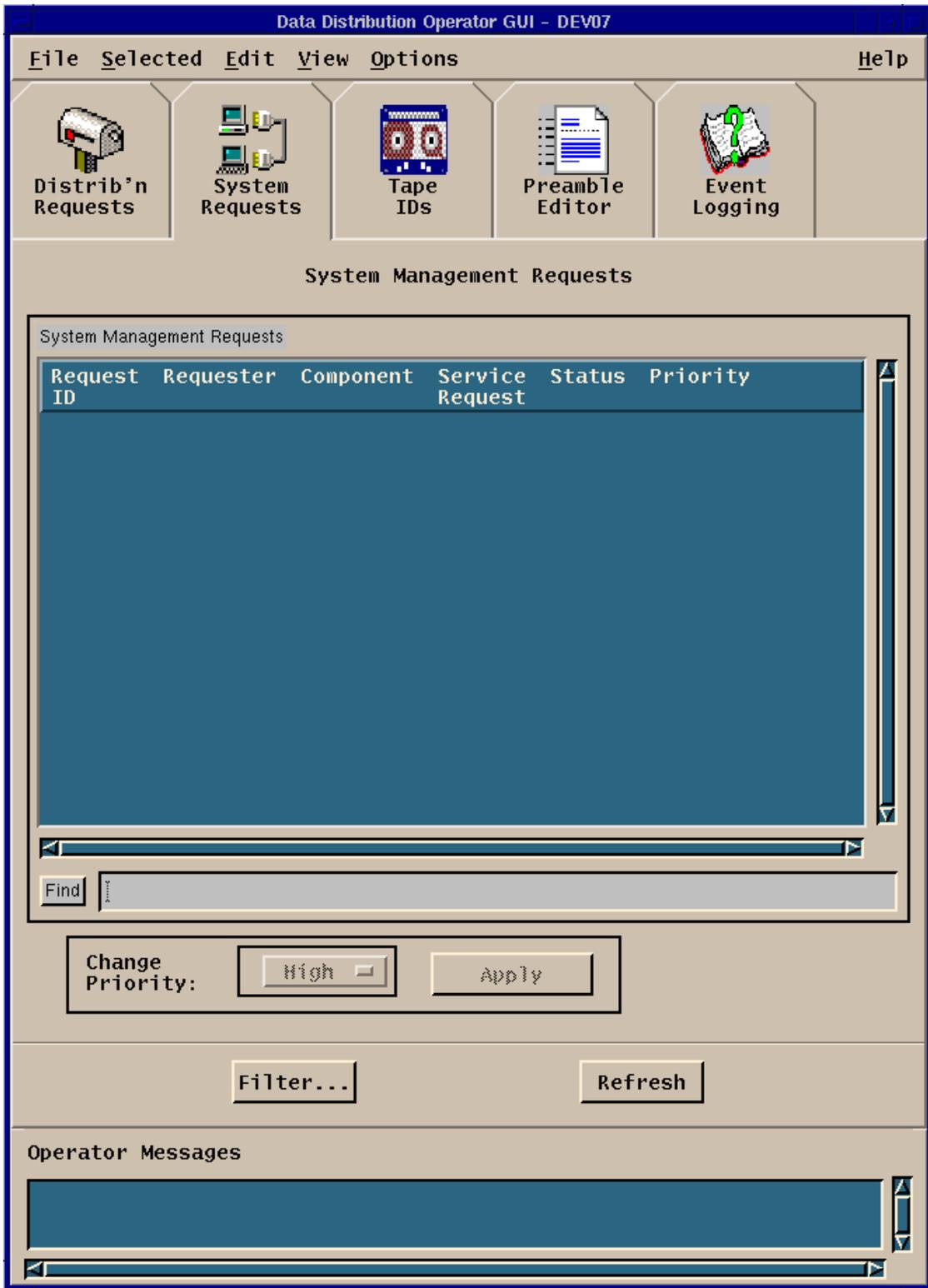


Figure 4.10.3-4. System Requests Tab

4.10.3.2.3 Tape IDs Tab

The purpose of the Tape Id tab shown in Figure 4.10.3-5 is to find and display Distribution Hard Media Request Items and Media IDs associated with these requests.

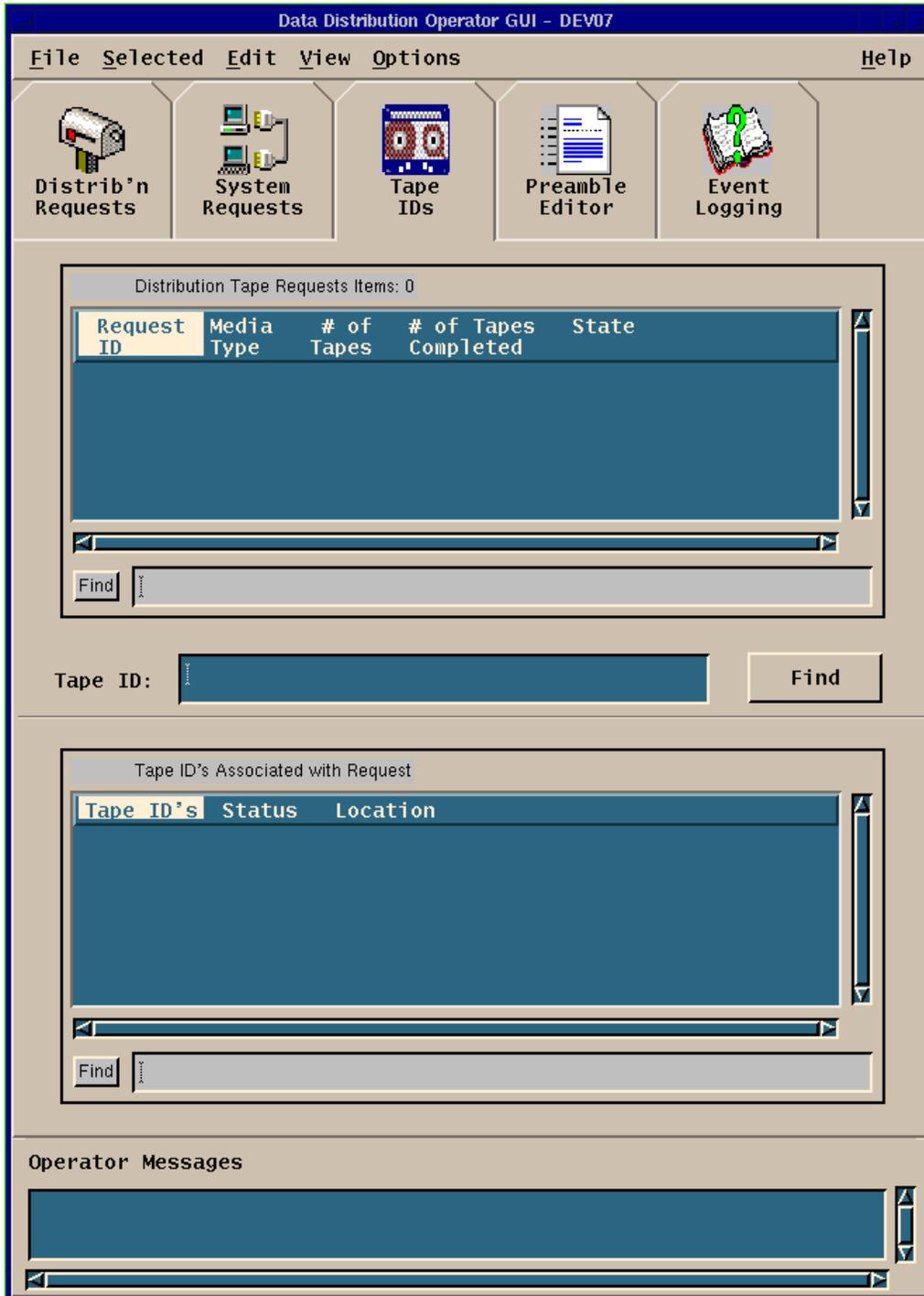


Figure 4.10.3-5. Tape IDs Tab

The tab contains two list panels, one for Distribution Hard Media Requests and the other for Media ID's Associated with Request. The first list displays the total number of media requests at the top of the panel. Several parameters associated with each individual request are displayed to the user through this panel. The list can be sorted by column. All of the parameters included for each request in the Distribution Hard Media Requests panel are listed and described in Table 4.10.3-5.

Table 4.10.3-5. Distribution Hard Media Requests Items Field Description

Field Name	Data Type	Size	Entry	Description
Request ID	Character	Unlimited	System generated	Unique identifier for the request.
Media Type	Character	Unlimited	System generated	Type of tape media to be used for distribution. Values are 8mm, DTF, CD ROM, and DLT.
# of Tapes	Integer	Max # of Media	System generated	Number of Media that need to be used to completely fulfill a media distribution request.
# of Tapes Completed	Integer	Unlimited	System generated	Number of Media that have already been filled up by the distribution request that is being processed
State	Character	Unlimited	System generated	Request states are: pending, active, staging, waiting for shipment, shipped, canceled, transferring, suspended, suspended with errors.

The second list displays the total set of media associated with the request selected in the first list. Several parameters associated with each individual request are displayed to the user through this panel. The list can only be sorted by Tape ID column. All of the parameters included for each request in the Media Ids Associated with Request panel are listed and described in Table 4.10.3-6.

Table 4.10.3-6. Media IDs Field Description

Field Name	Data Type	Size	Entry	Description
Tape ID's	Character	Unlimited	System generated	Unique identifier for the media.
Status	Character	Unlimited	System generated	Storage Management description of the media status.
Location	Character	Unlimited	System generated	Physical location of the media.
Operator Messages	Character	Unlimited	System generated	System will provide any necessary messages for the operator to read in this area.

The Media ID's tab includes additional functionality associated with the following button:

- **Find** allows the operator to search the database for a specified Media ID. If found, the media's associated Request ID is displayed in the Distribution Hard Media Requests list, and all of the media associated with the Request are listed in the Media ID list.

4.10.3.2.4 Preamble Editor Tab

The purpose of the Preamble Editor tab as shown in Figure 4.10.3-6 is to provide editing functions for the preamble files for email and packing lists.

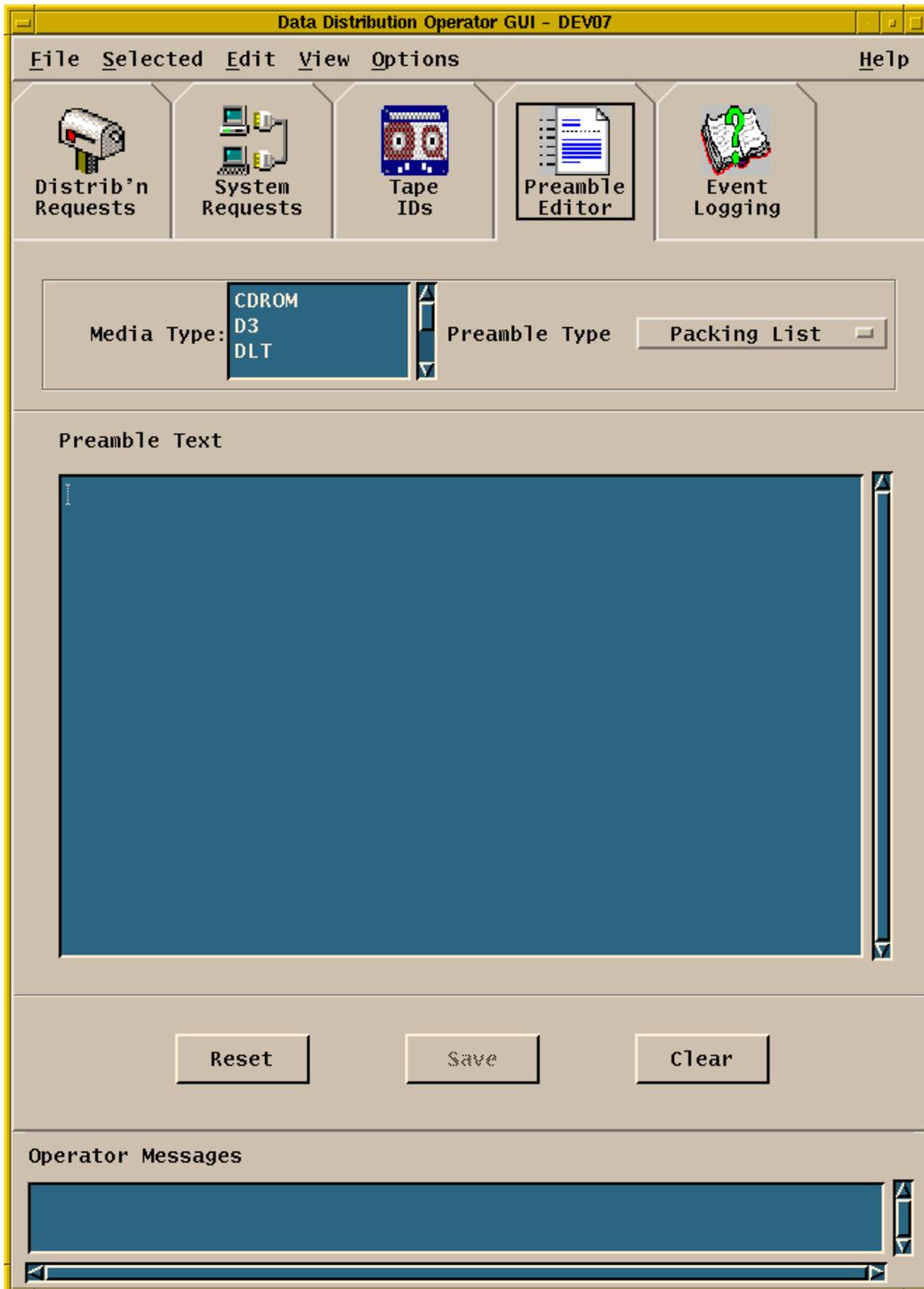


Figure 4.10.3-6. Preamble Editor Tab

The Preamble Editor tab allows the editing of each media type's Packing list, Successful report, and Failure report Preamble messages. The major component of this tab is the text-editing window, in which the Preamble messages are displayed and changed. Above the text-editing window are two gadgets for selecting the Media Type and Report Type. The Media Type gadget is a scrolled list, which is automatically filled from entries in the registry on which types of media are available. The Report Type gadget is a pull down list with three options: Packing List, Successful Email, and Failed Email. By selecting a value from either one of these gadgets, the corresponding media/report Preamble text is loaded from the appropriate file into the text-editing window.

Below the text-editing window are three buttons: Reset, Save, and Clear. Pressing the Reset button discards any changes made and reloads the current media/report Preamble text from the appropriate file. Pressing the Save button writes the current contents of the text edit window to the appropriate file. Pressing the Clear button removes all text from the text-editing window.

If any changes are made to the text in the text-editing window after loading or saving, the GUI displays a reminder notice to save if the Media Type, Report Type, or tab is changed. Also, the editing functions Cut, Copy, Paste in the Edit menu are enabled within the Preamble Editor tab. These editing functions are also available by right clicking within the text-editing window.

4.10.3.2.5 Event Logging Tab

The functionality associated with the Event Logging tab shown in Figure 4.10.3-7 is not yet defined, as of ECS Release 7 delivery.

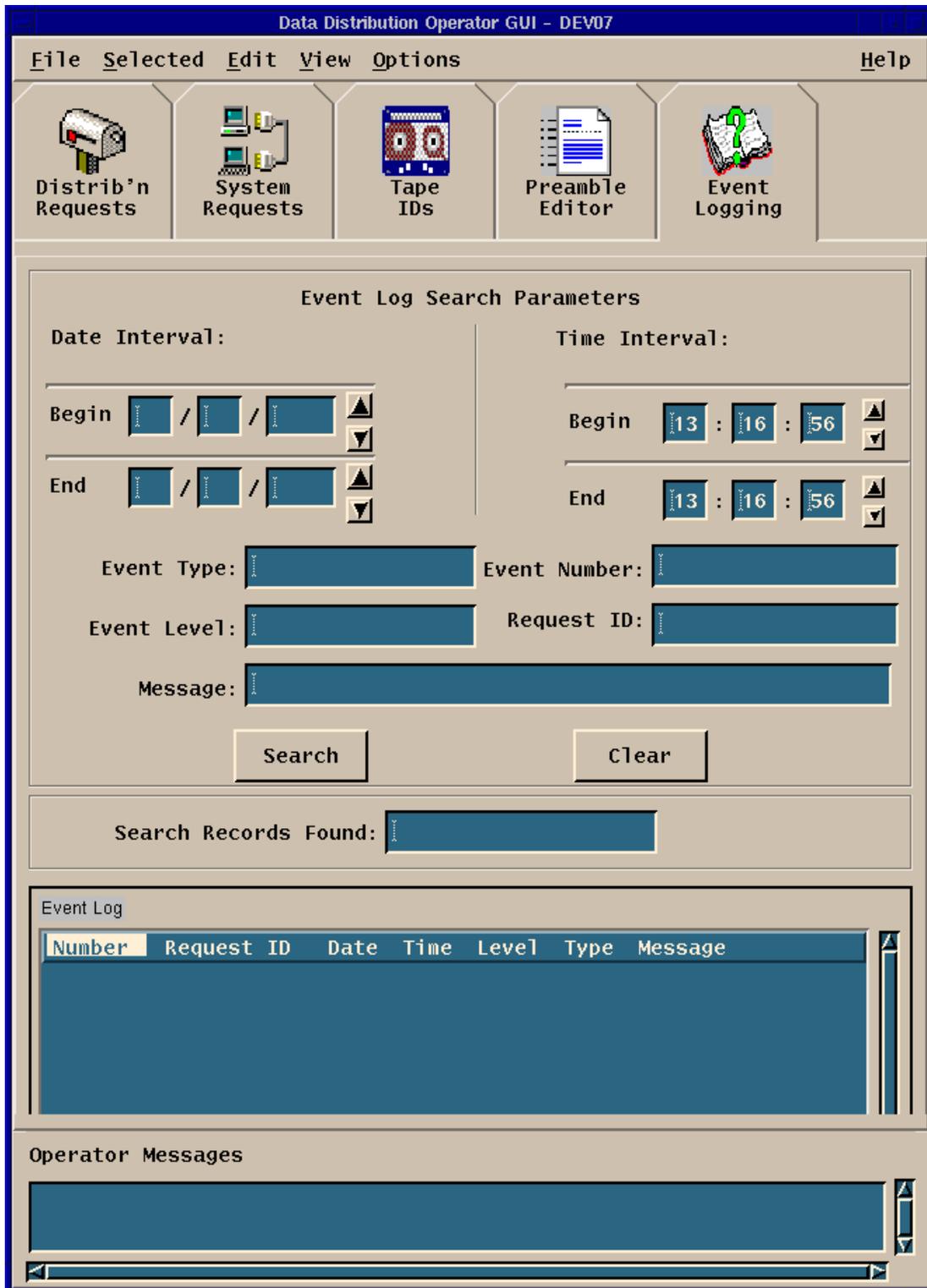


Figure 4.10.3-7. Event Logging Tab

4.10.3.3 Required Operating Environment

For information on the operating environment, tunable parameters, and environment variables refer to the 920-TDA-022 “Custom Code Configuration Parameters” documentation series.

4.10.3.3.1 Interfaces and Data Types

Table 4.10.3-7 identifies the supporting products this tool depends upon in order to function properly.

Table 4.10.3-7. External Interface Protocols

Product Dependency	Protocol Used	Comments
DDIST and all clients	Socket	via DDIST client libraries

4.10.3.4 Databases

The Data Distribution Requests Tool displays and updates the list of distribution requests after retrieving the information from the EcDsDistributionServer database associated with a mode. Mode refers to the system environment (e.g., OPS TS1 or TS2). Details about the architecture of the EcDsDistributionServer database can be found in the applicable section of DID 311-CD-101, *Data Distribution Database Design and Schema Specifications for the ECS Project*.

4.10.3.5 Special Constraints

The Data reported in the Task Activity window has to be retrieved from the Data Distribution database, as specified in the previous paragraph. The Data Distribution database must then be up and running before invoking the Data Distribution Tool.

4.10.3.6 Outputs

The Data Distribution Tool mainly is used to display data produced by other ECS components and does not generate any specific output.

4.10.3.7 Event and Error Messages

Both event and error messages are listed in Appendix A.

4.10.3.8 Reports

None.

This page intentionally left blank.

4.10.4 Granule Deletion Administration Tool

The Granule Deletion Administration Tool provides the ECS Operations Staff with the ability to delete granules using a command line interface. The granules can be deleted from both the inventory and archive or just the archive. Granules are not physically deleted from the Archive. The directory entry is deleted so that the files cannot be accessed. The physical storage occupied by the deleted granules is not reclaimed through this operation.

The deletion process can involve deleting the specified granules along with associated granules, as long as any other granules do not reference the associated granules (e.g., browse, PH, QA). The deletion process can also involve deleting the specified granules even if they are inputs to other granules.

4.10.4.1 Quick Start Using the Granule Delete Administration Tool

Enter the following command to start the Granule Deletion Administration Tool:

```
>EcDsGranuleDelete ConfigFile  
/usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG ecs_mode <MODE> <command  
line parameters>
```

There are various command line parameters and they are used in combination with each other. Table 4.10.4-1 provides a description of the parameters.

Table 4.10.4-1. Command Line Parameters of the Granule Deletion Administration Tool

Parameter Name	Description
name	ESDT Short Name of the granules to delete.
version	ESDT Version ID of the granules to delete.
begindate	Beginning Date of the temporal range of granules to delete.
enddate	Ending Date of the temporal range of granules to delete.
insertbegin	Beginning time when granules to delete were inserted.
insertend	Ending time when granules to delete were inserted.
localgranulefile	Name of file containing ESDT ShortName, Version IDs and Local Granule Ids of the granules to delete.
geoidfile	Name of file containing geoids of the granules to delete.
log	Name of log file to record the deletion operations. This parameter is mandatory.
physical	Delete from inventory and archive.
DFA	Delete from archive only.
noprompt	Do not prompt for confirmation of the delete.
display	Display the candidate granules for deletion, but do not delete.
noassoc	Do not delete associated granules (QA, Browse, PH).
delref	Delete granules that are referenced by other granules.

There are some parameters that are mandatory. The parameter physical, DFA or display must be specified. There are several parameters used to specify the science granules to delete: name, version, begindate and enddate or name, version, insertbegin and insertend or localgranulefile or geoidfile.

4.10.4.2 Granule Deletion Administration Tool Commands

The Granule Deletion Administration Tool provides the following 44 granule deletion options:

1. Confirmed deletion of science and associated granules from the Archive and Inventory. The science granules must meet the ShortName, VersionID and temporal range criteria input by the user.
2. Unconfirmed deletion of science and associated granules from the Archive and Inventory. The science granules must meet the ShortName, VersionID and temporal range criteria input by the user.
3. Confirmed deletion of science granules from the Archive and Inventory. The science granules must meet the ShortName, VersionID and temporal range criteria input by the user.
4. Unconfirmed deletion of science granules from the Archive and Inventory. The science granules must meet the ShortName, VersionID and temporal range criteria input by the user.
5. Confirmed deletion of referenced and un-referenced science granules from the Archive and Inventory. The science granules must meet the ShortName, VersionID, and temporal range criteria input by the user.
6. Unconfirmed deletion of referenced and un-referenced science granules from the Archive and Inventory. The science granules must meet the ShortName, VersionID, and temporal range criteria input by the user.
7. Confirmed deletion of referenced and un-referenced science granules and associated granules from the Archive and Inventory. The science granules must meet the ShortName, VersionID, and temporal range criteria input by the user.
8. Unconfirmed deletion of referenced and un-referenced science granules and associated granules from the Archive and Inventory. The science granules must meet the ShortName, VersionID, and temporal range criteria input by the user.
9. Confirmed deletion of science granules from the Archive. The science granules must meet the ShortName, VersionID and temporal range criteria input by the user.
10. Unconfirmed deletion of science granules from the Archive. The science granules must meet the ShortName, VersionID and temporal range criteria input by the user.
11. Confirmed deletion of science and associated granules from the Archive and Inventory. The science granules must meet the ShortName, VersionID and insert time range input by the user.

12. Unconfirmed deletion of science and associated granules from the Archive and Inventory. The science granules must meet the ShortName, VersionID and insert time range input by the user.
13. Confirmed deletion of science granules from the Archive and Inventory. The science granules must meet the ShortName, VersionID and insert time range input by the user.
14. Confirmed deletion of science granules from the Archive and Inventory. The science granules must meet the ShortName, VersionID and insert time range input by the user.
15. Confirmed deletion of referenced and un-referenced science granules from the Archive and Inventory. The science granules must meet the Shortname, VersionID, and insert time range input by the user.
16. Unconfirmed deletion of referenced and un-referenced science granules from the Archive and Inventory. The science granules must meet the Shortname, VersionID, and insert time range input by the user.
17. Confirmed deletion of referenced and un-referenced science granules and associated granules from the Archive and Inventory. The science granules must meet the Shortname, VersionID, and insert time range input by the user.
18. Unconfirmed deletion of referenced and un-referenced science granules and associated granules from the Archive and Inventory. The science granules must meet the Shortname, VersionID, and insert time range input by the user.
19. Confirmed deletion of science granules from the Archive. The science granules must meet the ShortName, VersionID and insert time range input by the user.
20. Unconfirmed deletion of science granules from the Archive. The science granules must meet the ShortName, VersionID and insert time range input by the user.
21. Confirmed deletion of science and associated granules from the Archive and Inventory. The science granules to delete are defined in a file containing ShortName, VersionID and LocalGranuleId.
22. Confirmed deletion of science and associated granules from the Archive and Inventory. The science granules to delete are defined in a file containing ShortName, VersionID and LocalGranuleId.
23. Confirmed deletion of science granules from the Archive and Inventory. The science granules to delete are defined in a file containing ShortName, VersionID and LocalGranuleId.
24. Unconfirmed deletion of science granules from the Archive and Inventory. The science granules to delete are defined in a file containing ShortName, VersionID and LocalGranuleId.

25. Confirmed deletion of referenced and un-referenced science granules from the Archive and Inventory. The science granules to delete are defined in a file containing ShortName, VersionID and LocalGranuleId.
26. Unconfirmed deletion of referenced and un-referenced science granules from the Archive and Inventory. The science granules to delete are defined in a file containing ShortName, VersionID and LocalGranuleId.
27. Confirmed deletion of referenced and un-referenced science granules and associated granules from the Archive and Inventory. The science granules to delete are defined in a file containing ShortName, VersionID and LocalGranuleId.
28. Unconfirmed deletion of referenced and un-referenced science granules and associated granules from the Archive and Inventory. The science granules to delete are defined in a file containing ShortName, VersionID and LocalGranuleId.
29. Confirmed deletion of science granules from the Archive. The science granules to delete are defined in a file containing ShortName, VersionID and LocalGranuleId.
30. Unconfirmed deletion of science granules from the Archive. The science granules to delete are defined in a file containing ShortName, VersionID and LocalGranuleId.
31. Confirmed deletion of science and associated granules from the Archive and Inventory. The science granules to delete are defined in a file containing SDSRV Granule Ids (basetype, shortname, version ID and db ID).
32. Unconfirmed deletion of science and associated granules from the Archive and Inventory. The science granules to delete are defined in a file containing SDSRV Granule Ids(basetype, shortname, version ID and db ID).
33. Confirmed deletion of science granules from the Archive and Inventory. The science granules to delete are defined in a file containing SDSRV Granule Ids (basetype, shortname, version ID and db ID).
34. Unconfirmed deletion of science granules from the Archive and Inventory. The science granules to delete are defined in a file containing SDSRV Granule Ids (basetype, shortname, version ID and db ID).
35. Confirmed deletion of referenced and un-referenced science granules from the Archive and Inventory. The science granules to delete are defined in a file containing SDSRV Granule Ids (basetype, shortname, version ID and db ID).
36. Unconfirmed deletion of referenced and un-referenced science granules from the Archive and Inventory. The science granules to delete are defined in a file containing SDSRV Granule Ids (basetype, shortname, version ID and db ID).
37. Confirmed deletion of referenced and un-referenced science granules and associated granules from the Archive and Inventory. The science granules to delete are defined in a file containing SDSRV Granule Ids (basetype, shortname, version ID and db ID).

38. Unconfirmed deletion of referenced and un-referenced science granules and associated granules from the Archive and Inventory. The science granules to delete are defined in a file containing SDSRV Granule Ids (basetype, shortname, version ID and db ID).
39. Confirmed deletion of science granules from the Archive. The science granules to delete are defined in a file containing SDSRV Granule Ids (basetype, shortname, version ID and db ID).
40. Unconfirmed deletion of science granules from the Archive. The science granules to delete are defined in a file that contains SDSRV Granule Ids (basetype, shortname, version ID and db ID).
41. Display science granules that are candidates for deletion. The science granules must meet the ShortName, VersionID and temporal range criteria input by the user.
42. Display science granules that are candidates for deletion. The science granules must meet the ShortName, VersionID and insert time range criteria input by the user.
43. Display science granules that are candidates for deletion. The science granules to delete are defined in a file containing ShortName, VersionID and LocalGranuleId.
44. Display science granules that are candidates for deletion. The science granules to delete are defined in a file containing SDSRV Granule Ids (basetype, shortname, version ID and db ID).

4.10.4.2.1 Confirmed Deletion of Science and Associated Granules from the Inventory and Archive by ESDT Short Name, Version ID and Data Temporal Coverage

This command has the form:

EcDsGranuleDelete

ConfigFile usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG

ecs_mode <MODE>

-log <logfilename>

-name <ESDT ShortName>

-version <ESDT VersionID>

-begindate <date>

-enddate <date>

-physical

This command physically deletes science granules specified by <ESDT_Shortname> and <ESDT_VersionID> and within the temporal range specified by parameters <begindate> and <enddate>. Associated granules are also deleted as long as other granules do not reference them.

The <logfilename> parameter specifies the log file the deletion process should use to write deletion activity.

The -physical parameter specifies the granules are deleted from the Inventory and the Archive.

The number of candidate science granules to be deleted is displayed and the user is prompted to confirm the deletion.

4.10.4.2.2 Unconfirmed Deletion of Science and Associated Granules from the Inventory and Archive by ESDT Short Name, Version ID and Data Temporal

This command has the form:

EcDsGranuleDelete

ConfigFile usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG

ecs_mode <MODE>

-log <logfilename>

-name <ESDT ShortName

-version <ESDT VersionID>

-begindate <date>

-enddate <date>

-physical

-noprompt

This command physically deletes science granules specified by <ESDT_Shortname> and <ESDT_VersionID> and within the temporal range specified by parameters <begindate> and <enddate>. Associated granules are also deleted as long as other granules do not reference them.

The <logfilename> parameter specifies the log file the deletion process should use to write deletion activity.

The -physical parameter specifies the granules are deleted from the Inventory and the Archive.

The -noprompt parameter specifies the user does not want to confirm the deletion of the granules.

4.10.4.2.3 Confirmed Deletion of Science Granules from the Inventory and Archive by ESDT Short Name, Version ID and Data Temporal Coverage

This command has the form:

EcDsGranuleDelete

ConfigFile *usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG*

ecs_mode *<MODE>*

-log *<logfilename>*

-name *<ESDT ShortName*

-version *<ESDT VersionID>*

-begindate *<date>*

-enddate *<date>*

-physical

-noassoc

This command physically deletes science granules specified by *<ESDT_Shortname>* and *<ESDT_VersionID>* and within the temporal range specified by parameters *<begindate>* and *<enddate>*.

The *<logfilename>* parameter specifies the log file the deletion process should use to write deletion activity.

The *-physical* parameter specifies the granules are deleted from the Inventory and the Archive.

The *-noassoc* parameter specifies the associated granules are not deleted even if other granules do not reference them.

The number of candidate science granules to be deleted is displayed and the user is prompted to confirm the deletion.

4.10.4.2.4 Unconfirmed Deletion of Science Granules from the Inventory and Archive by ESDT Short Name, Version ID and Data Temporal Coverage

This command has the form:

EcDsGranuleDelete

ConfigFile *usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG*

ecs_mode *<MODE>*

-log *<logfilename>*

-name *<ESDT ShortName*

-version *<ESDT VersionID>*

-begindate *<date>*

-enddate *<date>*

-physical

-noassoc

-noprompt

This command physically deletes science granules specified by <ESDT_Shortname> and <ESDT_VersionID> and within the temporal range specified by parameters <begindate> and <enddate>.

The <logfilename> parameter specifies the log file the deletion process should use to write deletion activity.

The -physical parameter specifies the granules are deleted from the Inventory and the Archive.

The -noassoc parameter specifies the associated granules are not deleted even if other granules do not reference them.

The -noprompt parameter specifies the user is not prompted to confirm the deletion.

4.10.4.2.5 Confirmed Deletion of Referenced and Unreferenced Science Granules from the Inventory and Archive by ESDT Short Name, Version ID and Data Temporal Coverage

This command has the form:

EcDsGranuleDelete

ConfigFile usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG

ecs_mode <MODE>

-log <logfilename>

-name <ESDT ShortName>

-version <ESDT VersionID>

-begindate <date>

-enddate <date>

-physical

-noassoc

-delref

This command physically deletes science granules specified by <ESDT_Shortname> and <ESDT_VersionID> and within the temporal range specified by parameters <begindate> and <enddate>. The science granules should be deleted even if other granules reference them.

The <logfilename> parameter specifies the log file the deletion process should use to write deletion activity.

The `-physical` parameter specifies the granules are deleted from the Inventory and the Archive.

The `-noassoc` parameter specifies the associated granules are not deleted even if other granules do not reference them.

The `-delref` parameter specifies the science granules should be deleted even if other granules reference them.

The number of candidate science granules to be deleted is displayed and the user is prompted to confirm the deletion.

4.10.4.2.6 Unconfirmed Deletion of Referenced and Unreferenced Science Granules from the Inventory and Archive by ESDT Short Name, Version ID and Data Temporal Coverage

This command has the form:

EcDsGranuleDelete

ConfigFile usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG

ecs_mode <MODE>

-log <logfilename>

-name <ESDT ShortName>

-version <ESDT VersionID>

-begindate <date>

-enddate <date>

-physical

-noassoc

-delref

-noprompt

This command physically deletes science granules specified by `<ESDT_Shortname>` and `<ESDT_VersionID>` and within the temporal range specified by parameters `<begindate>` and `<enddate>`. The science granules should be deleted even if other granules reference them.

The `<logfilename>` parameter specifies the log file the deletion process should use to write deletion activity.

The `-physical` parameter specifies the granules are deleted from the Inventory and the Archive.

The `-noassoc` parameter specifies the associated granules are not deleted even if other granules do not reference them.

The `-delref` parameter specifies the science granules should be deleted even if other granules reference them.

The `-noprompt` parameter specifies the granules should be deleted without a confirmation from the user.

4.10.4.2.7 Confirmed Deletion of Referenced and Unreferenced Science Granules and Associated Granules from the Inventory and Archive by ESDT Short Name, Version ID and Data Temporal Coverage

This command has the form:

EcDsGranuleDelete

ConfigFile `usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG`

ecs_mode `<MODE>`

-log `<logfilename>`

-name `<ESDT ShortName>`

-version `<ESDT VersionID>`

-begindate `<date>`

-enddate `<date>`

-physical

-delref

This command physically deletes science granules specified by `<ESDT_Shortname>` and `<ESDT_VersionID>` and within the temporal range specified by parameters `<begindate>` and `<enddate>`. The science granules should be deleted even if other granules reference them. Associated granules are also deleted as long as other granules do not reference them.

The `<logfilename>` parameter specifies the log file the deletion process should use to write deletion activity.

The `-physical` parameter specifies the granules are deleted from the Inventory and the Archive.

The `-delref` parameter specifies the science granules should be deleted even if other granules reference them.

The number of candidate science granules to be deleted is displayed and the user is prompted to confirm the deletion.

4.10.4.2.8 Unconfirmed Deletion of Referenced and Unreferenced Science Granules and Associated Granules from the Inventory and Archive by ESDT Short Name, Version ID and Data Temporal Coverage

This command has the form:

EcDsGranuleDelete

ConfigFile usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG

ecs_mode <MODE>

-log <logfilename>

-name <ESDT ShortName

-version <ESDT VersionID>

-begindate <date>

-enddate <date>

-physical

-delref

-noprompt

This command physically deletes science granules specified by <ESDT_Shortname> and <ESDT_VersionID> and within the temporal range specified by parameters <begindate> and <enddate>. The science granules should be deleted even if other granules reference them. Associated granules are also deleted as long as other granules do not reference them.

The <logfilename> parameter specifies the log file the deletion process should use to write deletion activity.

The -physical parameter specifies the granules are deleted from the Inventory and the Archive.

The -delref parameter specifies the science granules should be deleted even if other granules reference them.

The -noprompt parameter specifies the granules should be deleted without a confirmation from the user.

4.10.4.2.9 Confirmed Deletion of Science Granules from the Archive by ESDT Short Name, Version ID and Data Temporal Coverage

This command has the form:

EcDsGranuleDelete

ConfigFile usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG

ecs_mode <MODE>
-log <logfilename>
-name <ESDT ShortName>
-version <ESDT VersionID>
-begindate <date>
-enddate <date>
-DFA

This command deletes from the Archive science granules specified by <ESDT_Shortname> and <ESDT_VersionID> and within the temporal range specified by parameters <begindate> and <enddate>.

The <logfilename> parameter specifies the log file the deletion process should use to write deletion activity.

The -DFA parameter specifies the granules are deleted from the Archive.

The number of candidate science granules to be deleted is displayed and the user is prompted to confirm the deletion.

4.10.4.2.10 Unconfirmed Deletion of Science Granules from the Archive by ESDT Short Name, Version ID and Data Temporal Coverage

This command has the form:

EcDsGranuleDelete

ConfigFile *usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG*

ecs_mode <MODE>
-log <logfilename>
-name <ESDT ShortName>
-version <ESDT VersionID>
-begindate <date>
-enddate <date>
-DFA
-noprompt

This command deletes from the Archive science granules specified by <ESDT_Shortname> and <ESDT_VersionID> and within the temporal range specified by parameters <begindate> and <enddate>.

The <logfilename> parameter specifies the log file the deletion process should use to write deletion activity.

The -DFA parameter specifies the granules are deleted from the Archive.

The -noprompt parameter specifies the granules should be deleted without a confirmation from the user.

4.10.4.2.11 Confirmed Deletion of Science and Associated Granules from the Archive and Inventory by ESDT Short Name, Version ID and Insert Time Range

This command has the form:

EcDsGranuleDelete

ConfigFile usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG

ecs_mode <MODE>

-log <logfilename>

-name <ESDT ShortName>

-version <ESDT VersionID>

-insertbegin <date>

-insertend <date>

-physical

This command physically delete science granules specified by <ESDT_Shortname> and <ESDT_VersionID> and within the insert time range specified by parameters <insertbegin> and <insertend>. Associated granules are also deleted as long as other granules do not reference them.

The <logfilename> parameter specifies the log file the deletion process should use to write deletion activity.

The -physical parameter specifies the granules are deleted from the Inventory and the Archive.

The number of candidate science granules to be deleted is displayed and the user is prompted to confirm the deletion.

4.10.4.2.12 Unconfirmed Deletion of Science and Associated Granules from the Archive and Inventory by ESDT Short Name, Version ID and Insert Time Range

This command has the form:

EcDsGranuleDelete

ConfigFile usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG

ecs_mode <MODE>
-log <logfilename>
-name <ESDT ShortName>
-version <ESDT VersionID>
-insertbegin <date>
-insertend <date>
-physical
-noprompt

This command physically deletes science granules specified by <ESDT_Shortname> and <ESDT_VersionID> and within the insert time range specified by parameters <insertbegin> and <insertend>. Associated granules are also deleted as long as other granules do not reference them.

The <logfilename> parameter specifies the log file the deletion process should use to write deletion activity.

The *-physical* parameter specifies the granules are deleted from the Inventory and the Archive.

The *-noprompt* parameter specifies the user does not want to confirm the deletion of the granules.

4.10.4.2.13 Confirmed Deletion of Science Granules from the Archive and Inventory by ESDT Short Name, Version ID and Insert Time Range

This command has the form:

EcDsGranuleDelete

ConfigFile *usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG*

ecs_mode <MODE>
-log <logfilename>
-name <ESDT ShortName>
-version <ESDT VersionID>
-insertbegin <date>
-insertend <date>
-physical
-noassoc

This command physically delete science granules specified by <ESDT_Shortname> and <ESDT_VersionID> and within the insert time range specified by parameters <insertbegin> and <insertend>.

The <logfilefilename> parameter specifies the log file the deletion process should use to write deletion activity.

The -physical parameter specifies the granules are deleted from the Inventory and the Archive.

The -noassoc parameter specifies the associated granules are not deleted even if other granules do not reference them.

The number of candidate science granules to be deleted is displayed and the user is prompted to confirm the deletion.

4.10.4.2.14 Unconfirmed Deletion of Science Granules from the Archive and Inventory by ESDT Short Name, Version ID and Insert Time Range

This command has the form:

EcDsGranuleDelete

ConfigFile usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG

ecs_mode <MODE>

-log <logfilefilename>

-name <ESDT ShortName>

-version <ESDT VersionID>

-insertbegin <date>

-insertend <date>

-physical

-noassoc

-noprompt

This command physically delete science granules specified by <ESDT_Shortname> and <ESDT_VersionID> and within the insert time range specified by parameters <insertbegin> and <insertend>.

The <logfilefilename> parameter specifies the log file the deletion process should use to write deletion activity.

The -physical parameter specifies the granules are deleted from the Inventory and the Archive.

The -noassoc parameter specifies the associated granules are not deleted even if other granules do not reference them.

The `-noprompt` parameter specifies the user does not want to confirm the deletion of the granules.

4.10.4.2.15 Confirmed Deletion of Referenced and Unreferenced Science Granules from the Archive and Inventory by ESDT Short Name, Version ID and Insert Time Range

This command has the form:

EcDsGranuleDelete

ConfigFile `usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG`

ecs_mode `<MODE>`

-log `<logfilename>`

-name `<ESDT ShortName>`

-version `<ESDT VersionID>`

-insertbegin `<date>`

-insertend `<date>`

-physical

-noassoc

-delref

This command physically delete science granules specified by `<ESDT_Shortname>` and `<ESDT_VersionID>` and within the insert time range specified by parameters `<insertbegin>` and `<insertend>`.

The `<logfilename>` parameter specifies the log file the deletion process should use to write deletion activity.

The `-physical` parameter specifies the granules are deleted from the Inventory and the Archive.

The `-noassoc` parameter specifies the associated granules are not deleted even if other granules do not reference them.

The `-delref` parameter specifies the science granules should be deleted even if other granules reference them.

The number of candidate science granules to be deleted is displayed and the user is prompted to confirm the deletion.

4.10.4.2.16 Unconfirmed Deletion of Referenced and Unreferenced Science Granules from the Archive and Inventory by ESDT Short Name, Version ID and Insert Time Range

This command has the form:

EcDsGranuleDelete

ConfigFile *usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG*

ecs_mode *<MODE>*

-log *<logfilename>*

-name *<ESDT ShortName>*

-version *<ESDT VersionID>*

-insertbegin *<date>*

-insertend *<date>*

-physical

-noassoc

-delref

-noprompt

This command physically delete science granules specified by *<ESDT_Shortname>* and *<ESDT_VersionID>* and within the insert time range specified by parameters *<insertbegin>* and *<insertend>*.

The *<logfilename>* parameter specifies the log file the deletion process should use to write deletion activity.

The *-physical* parameter specifies the granules are deleted from the Inventory and the Archive.

The *-noassoc* parameter specifies the associated granules are not deleted even if other granules do not reference them.

The *-delref* parameter specifies the science granules should be deleted even if other granules reference them.

The *-noprompt* parameter specifies the user is not prompted to confirm the deletion.

4.10.4.2.17 Confirmed Deletion of Referenced and Unreferenced Science Granules and Associated Granules from the Archive and Inventory by ESDT Short Name, Version ID and Insert Time Range

This command has the form:

EcDsGranuleDelete

ConfigFile usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG

ecs_mode <MODE>

-log <logfilename>

-name <ESDT ShortName>

-version <ESDT VersionID>

-insertbegin <date>

-insertend <date>

-physical

-delref

This command physically delete science granules specified by <ESDT_Shortname> and <ESDT_VersionID> and within the insert time range specified by parameters <insertbegin> and <insertend>. Associated granules are also deleted as long as other granules do not reference them.

The <logfilename> parameter specifies the log file the deletion process should use to write deletion activity.

The -physical parameter specifies the granules are deleted from the Inventory and the Archive.

The -delref parameter specifies the science granules should be deleted even if other granules reference them.

The number of candidate science granules to be deleted is displayed and the user is prompted to confirm the deletion.

4.10.4.2.18 Unconfirmed Deletion of Referenced and Unreferenced Science Granules and Associated Granules from the Archive and Inventory by ESDT Short Name, Version ID and Insert Time Range

This command has the form:

EcDsGranuleDelete

ConfigFile usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG

ecs_mode <MODE>

-log <logfilename>
-name <ESDT ShortName>
-version <ESDT VersionID>
-insertbegin <date>
-insertend <date>
-physical
-delref
-noprompt

This command physically delete science granules specified by <ESDT_Shortname> and <ESDT_VersionID> and within the insert time range specified by parameters <insertbegin> and <insertend>. Associated granules are also deleted as long as other granules do not reference them.

The <logfilename> parameter specifies the log file the deletion process should use to write deletion activity.

The -physical parameter specifies the granules are deleted from the Inventory and the Archive.

The -delref parameter specifies the science granules should be deleted even if other granules reference them.

The -noprompt parameter specifies the user is not prompted to confirm the deletion.

4.10.4.2.19 Confirmed Deletion of Science Granules from the Archive by Short Name, Version ID and Insert Time Range

This command has the form:

EcDsGranuleDelete

ConfigFile usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG

ecs_mode <MODE>

-log <logfilename>

-name <ESDT ShortName>

-version <ESDT VersionID>

-insertbegin <date>

-insertend <date>

-DFA

This command deletes from the Archive science granules specified by <ESDT_Shortname> and <ESDT_VersionID> and within the insert time range specified by parameters <insertbegin> and <insertend>.

The <logfilename> parameter specifies the log file the deletion process should use to write deletion activity.

The -DFA parameter specifies the granules are deleted from the Archive.

The number of candidate science granules to be deleted is displayed and the user is prompted to confirm the deletion.

4.10.4.2.20 Unconfirmed Deletion of Science Granules from the Archive by Short Name, Version ID and Insert Time Range

This command has the form:

EcDsGranuleDelete

ConfigFile usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG

ecs_mode <MODE>

-log <logfilename>

-name <ESDT ShortName>

-version <ESDT VersionID>

-insertbegin <date>

-insertend <date>

-DFA

-noprompt

This command deletes from the Archive science granules specified by <ESDT_Shortname> and <ESDT_VersionID> and within the insert time range specified by parameters <insertbegin> and <insertend>.

The <logfilename> parameter specifies the log file the deletion process should use to write deletion activity.

The -DFA parameter specifies the granules are deleted from the Archive.

The -noprompt parameter specifies the user is not prompted to confirm the deletion.

4.10.4.2.21 Confirmed Deletion of Science and Associated Granules from the Inventory and Archive by ESDT Short Name, Version ID and Local Granule Id

This command has the form:

EcDsGranuleDelete

ConfigFile *usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG*

ecs_mode *<MODE>*

-log *<logfilename>*

-localgranulefile *<filename>*

-physical

This command physically deletes science granules defined in a file containing ESDT ShortName, ESDT Version ID and Local Granule Id. Associated granules are also deleted as long as other granules do not reference them.

The *<logfilename>* parameter specifies the log file the deletion process should use to write deletion activity.

The *-localgranulefile* parameter specifies the file that contains the science granules to delete. The file contains ESDT Short Name, ESDT Version ID and Local Granule Id.

The *-physical* parameter specifies the granules are deleted from the Inventory and the Archive.

The number of candidate science granules to be deleted is displayed and the user is prompted to confirm the deletion.

4.10.4.2.22 Unconfirmed Deletion of Science and Associated Granules from the Inventory and Archive by ESDT Short Name, Version ID and Local Granule Id

This command has the form:

EcDsGranuleDelete

ConfigFile *usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG*

ecs_mode *<MODE>*

-log *<logfilename>*

-localgranulefile *<filename>*

-physical

-noprompt

This command physically deletes science granules defined in a file containing ESDT ShortName, ESDT Version ID and Local Granule Id. Associated granules are also deleted as long as other granules do not reference them.

The <logfilename> parameter specifies the log file the deletion process should use to write deletion activity.

The -localgranulefile parameter specifies the file containing the science granules to delete. The file contains ESDT Short Name, ESDT Version ID and Local Granule Id.

The -physical parameter specifies the granules are deleted from the Inventory and the Archive.

The -noprompt parameter specifies the user is not prompted to confirm the deletion.

4.10.4.2.23 Confirmed Deletion of Science Granules from the Inventory and Archive by ESDT Short Name, Version ID and Local Granule Id

This command has the form:

EcDsGranuleDelete

ConfigFile usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG

ecs_mode <MODE>

-log <logfilename>

-localgranulefile <filename>

-physical

-noassoc

This command physically deletes science granules defined in a file containing ESDT ShortName, ESDT Version ID and Local Granule Id.

The <logfilename> parameter specifies the log file the deletion process should use to write deletion activity.

The -localgranulefile parameter specifies the file containing the science granules to delete. The file contains ESDT Short Name, ESDT Version ID and Local Granule Id.

The -physical parameter specifies the granules are deleted from the Inventory and the Archive.

The -noassoc parameter specifies the associated granules are not deleted even if other granules do not reference them.

The number of candidate science granules to be deleted is displayed and the user is prompted to confirm the deletion.

4.10.4.2.24 Unconfirmed Deletion of Science Granules from the Inventory and Archive by ESDT Short Name, Version ID and Local Granule Id

This command has the form:

EcDsGranuleDelete

ConfigFile usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG

ecs_mode <MODE>

-log <logfilename>

-localgranulefile <filename>

-physical

-noassoc

This command physically deletes science granules defined in a file containing ESDT ShortName, ESDT Version ID and Local Granule Id.

The <logfilename> parameter specifies the log file the deletion process should use to write deletion activity.

The -localgranulefile parameter specifies the file containing the science granules to delete. The file contains ESDT Short Name, ESDT Version ID and Local Granule Id.

The -physical parameter specifies the granules are deleted from the Inventory and the Archive.

The -noassoc parameter specifies the associated granules are not deleted even if other granules do not reference them.

The -noprompt parameter specifies the user does not want to confirm the deletion of the granules.

4.10.4.2.25 Confirmed Deletion of Referenced and Unreferenced Science Granules from the Inventory and Archive by ESDT Short Name, Version ID and Local Granule Id

The command has the form:

EcDsGranuleDelete

ConfigFile usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG

ecs_mode <MODE>

-log <logfilename>

-localgranulefile <filename>

-physical

-noassoc

-delref

This command physically deletes science granules defined in a file containing ESDT ShortName, ESDT Version ID and Local Granule Id.

The <logfile> parameter specifies the log file the deletion process should use to write deletion activity.

The -localgranulefile parameter specifies the file containing the science granules to delete. The file contains ESDT Short Name, ESDT Version ID and Local Granule Id.

The -physical parameter specifies the granules are deleted from the Inventory and the Archive.

The -noassoc parameter specifies the associated granules are not deleted even if other granules do not reference them.

The -delref parameter specifies the science granules should be deleted even if other granules reference them.

The number of candidate science granules to be deleted is displayed and the user is prompted to confirm the deletion.

4.10.4.2.26 Unconfirmed Deletion of Referenced and Unreferenced Science Granules from the Inventory and Archive by ESDT Short Name, Version ID and Local Granule Id

The command has the form:

EcDsGranuleDelete

ConfigFile usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG

ecs_mode <MODE>

-log <logfile>

-localgranulefile <filename>

-physical

-noassoc

-delref

This command physically deletes science granules defined in a file containing ESDT ShortName, ESDT Version ID and Local Granule Id.

The <logfile> parameter specifies the log file the deletion process should use to write deletion activity.

The `-localgranulefile` parameter specifies the file containing the science granules to delete. The file contains ESDT Short Name, ESDT Version ID and Local Granule Id.

The `-physical` parameter specifies the granules are deleted from the Inventory and the Archive.

The `-noassoc` parameter specifies the associated granules are not deleted even if other granules do not reference them.

The `-delref` parameter specifies the science granules should be deleted even if other granules reference them.

The `-noprompt` parameter specifies the user is not prompted to confirm the deletion.

4.10.4.2.27 Confirmed Deletion of Referenced and Unreferenced Science Granules and Associated Granules from the Inventory and Archive by ESDT Short Name, Version ID and Local Granule Id

The command has the form:

EcDsGranuleDelete

ConfigFile *usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG*

ecs_mode *<MODE>*

-log *<logfilename>*

-localgranulefile *<filename>*

-physical

-delref

This command physically deletes science granules defined in a file containing ESDT ShortName, ESDT Version ID and Local Granule Id. Associated granules are also deleted as long as other granules do not reference them.

The *<logfilename>* parameter specifies the log file the deletion process should use to write deletion activity.

The `-localgranulefile` parameter specifies the file containing the science granules to delete. The file contains ESDT Short Name, ESDT Version ID and Local Granule Id.

The `-physical` parameter specifies the granules are deleted from the Inventory and the Archive.

The `-delref` parameter specifies the science granules should be deleted even if other granules reference them.

The number of candidate science granules to be deleted is displayed and the user is prompted to confirm the deletion.

4.10.4.2.28 Unconfirmed Deletion of Referenced and Unreferenced Science Granules and Associated Granules from the Inventory and Archive by ESDT Short Name, Version ID and Local Granule Id

The command has the form:

EcDsGranuleDelete

ConfigFile usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG

ecs_mode <MODE>

-log <logfilename>

-localgranulefile <filename>

-physical

-delref

-noprompt

This command physically deletes science granules defined in a file containing ESDT ShortName, ESDT Version ID and Local Granule Id. Associated granules are also deleted as long as other granules do not reference them.

The <logfilename> parameter specifies the log file the deletion process should use to write deletion activity.

The -localgranulefile parameter specifies the file containing the science granules to delete. The file contains ESDT Short Name, ESDT Version ID and Local Granule Id.

The -physical parameter specifies the granules are deleted from the Inventory and the Archive.

The -delref parameter specifies the science granules should be deleted even if other granules reference them.

The -noprompt parameter specifies the user is not prompted to confirm the deletion.

4.10.4.2.29 Confirmed Deletion of Science Granules from the Archive by ESDT Short Name, Version ID and Local Granule Id

This command has the form:

EcDsGranuleDelete

ConfigFile usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG

ecs_mode <MODE>

-log <logfilename>

-localgranulefile <filename>

-DFA

This command deletes from the Archive science granules defined in a file containing ESDT ShortName, ESDT Version ID and Local Granule Id.

The <logfilename> parameter specifies the log file the deletion process should use to write deletion activity.

The -localgranulefile parameter specifies the file containing the science granules to delete. The file contains ESDT Short Name, ESDT Version ID and Local Granule Id.

The -DFA parameter specifies the granules are deleted from the Archive.

The number of candidate science granules to be deleted is displayed and the user is prompted to confirm the deletion.

4.10.4.2.30 Unconfirmed Deletion of Science Granules from the Archive by ESDT Short Name, Version ID and Local Granule Id

This command has the form:

EcDsGranuleDelete

ConfigFile usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG

ecs_mode <MODE>

-log <logfilename>

-localgranulefile <filename>

-DFA

-noprompt

This command deletes from the Archive science granules defined in a file containing ESDT ShortName, ESDT Version ID and Local Granule Id.

The <logfilename> parameter specifies the log file the deletion process should use to write deletion activity.

The -localgranulefile parameter specifies the file containing the science granules to delete. The file contains ESDT Short Name, ESDT Version ID and Local Granule Id.

The -DFA parameter specifies the granules are deleted from the Archive.

The -noprompt parameter specifies the granules should be deleted without confirmation from the user.

4.10.4.2.31 Confirmed Deletion of Science and Associated Granules from the Inventory and Archive by SDSRV Granule Id

The command has the form:

EcDsGranuleDelete

ConfigFile usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG

ecs_mode <MODE>

-log <logfilename>

-geoidfile <filename>

-physical

This command physically deletes science granules specified by SDSRV Granule Id. Associated granules are also deleted as long as other granules do not reference them.

The <logfilename> parameter specifies the log file the deletion process should use to write deletion activity.

The -geoidfile parameter specifies the file containing the science granules to delete. The file contains SDSRV Granule Ids.

The -physical parameter specifies the granules are deleted from the Inventory and the Archive.

The number of candidate science granules to be deleted is displayed and the user is prompted to confirm the deletion.

4.10.4.2.32 Unconfirmed Deletion of Science and Associated Granules from the Inventory and Archive by SDSRV Granule Id

The command has the form:

EcDsGranuleDelete

ConfigFile usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG

ecs_mode <MODE>

-log <logfilename>

-geoidfile <filename>

-physical

-noprompt

This command physically deletes science granules specified by SDSRV Granule Id. Associated granules are also deleted as long as other granules do not reference them.

The <logfile> parameter specifies the log file the deletion process should use to write deletion activity.

The -geoidfile parameter specifies the file containing the science granules to delete. The file contains SDSRV Granule Ids.

The -physical parameter specifies the granules are deleted from the Inventory and the Archive.

The -noprompt parameter specifies the user does not want to confirm the deletion of the granules.

4.10.4.2.33 Confirmed Deletion of Science Granules from the Inventory and Archive by SDSRV Granule Id

The command has the form:

EcDsGranuleDelete

ConfigFile usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG

ecs_mode <MODE>

-log <logfile>

-geoidfile <filename>

-physical

-noassoc

This command physically deletes science granules specified by SDSRV Granule Id.

The <logfile> parameter specifies the log file the deletion process should use to write deletion activity.

The -geoidfile parameter specifies the file containing the science granules to delete. The file contains SDSRV Granule Ids.

The -physical parameter specifies the granules are deleted from the Inventory and the Archive.

The -noassoc parameter specifies the associated granules are not deleted even if other granules do not reference them.

The number of candidate science granules to be deleted is displayed and the user is prompted to confirm the deletion.

4.10.4.2.34 Unconfirmed Deletion of Science Granules from the Inventory and Archive by SDSRV Granule Id

The command has the form:

EcDsGranuleDelete

ConfigFile usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG

ecs_mode <MODE>

-log <logfilename>

-geoidfile <filename>

-physical

-noassoc

-noprompt

This command physically deletes science granules specified by SDSRV Granule Id.

The <logfilename> parameter specifies the log file the deletion process should use to write deletion activity.

The -geoidfile parameter specifies the file containing the science granules to delete. The file contains SDSRV Granule Ids.

The -physical parameter specifies the granules are deleted from the Inventory and the Archive.

The -noassoc parameter specifies the associated granules are not deleted even if other granules do not reference them.

The -noprompt parameter specifies the user does not want to confirm the deletion of the granules.

4.10.4.2.35 Confirmed Deletion of Referenced and Unreferenced Science Granules from the Inventory and Archive by SDSRV Granule Id

The command has the form:

EcDsGranuleDelete

ConfigFile usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG

ecs_mode <MODE>

-log <logfilename>

-geoidfile <filename>

-physical

-noassoc

-delref

This command physically deletes science granules specified by SDSRV Granule Id.

The <logfilename> parameter specifies the log file the deletion process should use to write deletion activity.

The `-geoidfile` parameter specifies the file containing the science granules to delete. The file contains SDSRV Granule Ids.

The `-physical` parameter specifies the granules are deleted from the Inventory and the Archive.

The `-noassoc` parameter specifies the associated granules are not deleted even if other granules do not reference them.

The `-delref` parameter specifies the science granules should be deleted even if other granules reference them.

The number of candidate science granules to be deleted is displayed and the user is prompted to confirm the deletion.

4.10.4.2.36 Unconfirmed Deletion of Referenced and Unreferenced Science Granules from the Inventory and Archive by SDSRV Granule Id

The command has the form:

EcDsGranuleDelete

***ConfigFile** `usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG`*

***ecs_mode** <MODE>*

***-log** <logfile>*

***-geoidfile** <filename>*

-physical

-noassoc

-noprompt

This command physically deletes science granules specified by SDSRV Granule Id.

The `<logfile>` parameter specifies the log file the deletion process should use to write deletion activity.

The `-geoidfile` parameter specifies the file containing the science granules to delete. The file contains SDSRV Granule Ids.

The `-physical` parameter specifies the granules are deleted from the Inventory and the Archive.

The `-noassoc` parameter specifies the associated granules are not deleted even if other granules do not reference them.

The `-delref` parameter specifies the science granules should be deleted even if other granules reference them.

The `-noprompt` parameter specifies the user does not want to confirm the deletion of the granules.

4.10.4.2.37 Confirmed Deletion of Referenced and Unreferenced Science Granules and Associated Granules from the Inventory and Archive by SDSRV Granule Id

The command has the form:

EcDsGranuleDelete

ConfigFile usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG

ecs_mode <MODE>

-log <logfilename>

-geoidfile <filename>

-physical

-delref

This command physically deletes science granules specified by SDSRV Granule Id. Associated granules are also deleted as long as other granules do not reference them.

The <logfilename> parameter specifies the log file the deletion process should use to write deletion activity.

The -geoidfile parameter specifies the file containing the science granules to delete. The file contains SDSRV Granule Ids.

The -physical parameter specifies the granules are deleted from the Inventory and the Archive.

The -delref parameter specifies the science granules should be deleted even if other granules reference them.

The number of candidate science granules to be deleted is displayed and the user is prompted to confirm the deletion.

4.10.4.2.38 Unconfirmed Deletion of Referenced and Unreferenced Science Granules from the Inventory and Archive by SDSRV Granule Id

The command has the form:

EcDsGranuleDelete

ConfigFile usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG

ecs_mode <MODE>

-log <logfilename>

-geoidfile <filename>

-physical

-noprompt

This command physically deletes science granules specified by SDSRV Granule Id. Associated granules are also deleted as long as other granules do not reference them.

The <logfile> parameter specifies the log file the deletion process should use to write deletion activity.

The -geoidfile parameter specifies the file containing the science granules to delete. The file contains SDSRV Granule Ids.

The -physical parameter specifies the granules are deleted from the Inventory and the Archive.

The -delref parameter specifies the science granules should be deleted even if other granules reference them.

The -noprompt parameter specifies the user does not want to confirm the deletion of the granules.

4.10.4.2.39 Confirmed Deletion of Science Granules from Archive SDSRV Granule Id

This command has the form:

EcDsGranuleDelete

***ConfigFile** usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG*

***ecs_mode** <MODE>*

***-log** <logfile>*

***-geoidfile** <filename>*

-DFA

This command deletes from the Archive science granules defined in a file containing SDSRV Granule Ids.

The <logfile> parameter specifies the log file the deletion process should use to write deletion activity.

The -geoidfile parameter specifies the file containing the science granules to delete. The file contains SDSRV Granule Ids.

The -DFA parameter specifies the granules are deleted from the Archive.

The number of candidate science granules to be deleted is displayed and the user is prompted to confirm the deletion.

4.10.4.2.40 Unconfirmed Deletion of Science Granules from Archive by SDSRV Granule Id

This command has the form:

EcDsGranuleDelete

ConfigFile *usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG*

ecs_mode *<MODE>*

-log *<logfilename>*

-geoidfile *<filename>*

-DFA

-noprompt

This command deletes from the Archive science granules defined in a file containing SDSRV Granule Ids.

The *<logfilename>* parameter specifies the log file the deletion process should use to write deletion activity.

The *-geoidfile* parameter specifies the file containing the science granules to delete. The file contains SDSRV Granule Ids.

The *-DFA* parameter specifies the granules are deleted from the Archive.

The *-noprompt* parameter specifies the granules should be deleted without confirmation from the user.

4.10.4.2.41 Display Science Granules Specified by Short Name, Version Id and Data Temporal Range

The command has the form:

EcDsGranuleDelete

ConfigFile *usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG*

ecs_mode *<MODE>*

-log *<logfilename>*

-name *<ESDT ShortName>*

-version *<ESDT VersionID>*

-begindate *<date>*

-enddate *<date>*

-display

This command displays the science granules that would be deleted if the command were executed without the *-display*. The science granules are specified by parameters <ESDT ShortName> and <ESDT_VersionID> within the temporal range specified by parameters <begindate> and <enddate>.

The <logfilename> parameter specifies the log file where the candidate science granules are written. The SDSRV Granule Id and Local Granule Id is written to the log for each science granule that is a candidate for deletion.

The SDSRV Granule Id and Local Granule Id of each candidate granule is displayed to the user along with the total number of granules.

4.10.4.2.42 Display Science Granules Specified by Short Name, Version Id and Insert Time Range

The command has the form:

EcDsGranuleDelete

ConfigFile usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG

ecs_mode <MODE>

-log <logfilename>

-name <ESDT ShortName>

-version <ESDT VersionID>

-insertbegin <date>

-insertend <date>

-display

This command displays the science granules that would be deleted if the command were executed without the *-display*. The science granules are specified by parameters <ESDT ShortName> and <ESDT_VersionID> within the insert time range specified by the parameters <insertbegin> and <insertend>.

The <logfilename> parameter specifies the log file where the candidate science granules are written. The SDSRV Granule Id and Local Granule Id is written to the log for each science granule that is a candidate for deletion.

The SDSRV Granule Id and Local Granule Id of each candidate granule is displayed to the user along with the total number of granules.

4.10.4.2.43 Display Science Granules Specified by Short Name, Version Id and Local Granule Id

The command has the form:

EcDsGranuleDelete

ConfigFile *usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG*

ecs_mode *<MODE>*

-log *<logfilename>*

-localgranulefile *<filename>*

-display

This command displays the science granules that would be deleted if the command were executed without the *-display*. The science granules are specified by parameters *<ESDT ShortName>* and *<ESDT_VersionID>* and Local Granule Id.

The *<logfilename>* parameter specifies the log file where the candidate science granules are written. The SDSRV Granule Id and Local Granule Id is written to the log for each science granule that is a candidate for deletion.

The *-localgranulefile* parameter specifies the file containing the science granules to delete. The file contains ESDT Short Name, ESDT Version ID and Local Granule Id.

The SDSRV Granule Id and Local Granule Id of each candidate granule is displayed to the user along with the total number of granules.

4.10.4.2.44 Display Science Granules Specified SDSRV Granule Id

The command has the form:

EcDsGranuleDelete

ConfigFile *usr/ecs/CUSTOM/<MODE>/cfg/EcDsGranuleDelete.CFG*

ecs_mode *<MODE>*

-log *<logfilename>*

-geoidfile *<filename>*

-display

This command displays the science granules that would be deleted if the command were executed without the *-display*. SDSRV Granule Id specifies the science granules.

The *<logfilename>* parameter specifies the log file where the candidate science granules are written. The SDSRV Granule Id and Local Granule Id is written to the log for each science granule that is a candidate for deletion.

The -geoidfile parameter specifies the file containing the science granules to delete. The file contains SDSRV Granule Ids.

The SDSRV Granule Id and Local Granule Id of each candidate granule is displayed to the user along with the total number of granules.

4.10.4.3 Required Operating Environment

For information on the operating environment, tunable parameters, and environment variables refer to the 920-TDA-022 “Custom Code Configuration Parameters” documentation series.

4.10.4.3.1 Interfaces and Data Types

Table 4.10.4-2 lists the supporting products this tool depends upon to function properly.

Table 4.10.4-2. Interface Protocols

Product Dependency	Protocols Used	Comments
SDSRV and all clients	Socket	Via client libraries

4.10.4.4 Databases

The Granule Deletion Administration tool does not include the direct managing of any database. It has an interface with the Science Data Server Data Base: however this interface is based on a simple parameter passing function. For further information of the Science Data Server Data Base refer to 311-CD-624, *Science Data Server Database Design and Schema Specifications for the ECS Project*.

4.10.4.5 Special Constraints

The Granule Deletion Administration Tool runs only if the Science Data Server is running in the background. Note also when the Granule Deletion Administration Tool is started through a command line, the command line specifies the configuration file used to initialize the application.

4.10.4.6 Outputs

None.

4.10.4.7 Event and Error Messages

None.

4.10.4.8 Reports

None.

This page intentionally left blank.