

### 4.3.5 Tivoli/Courier

Tivoli/Courier (T/Courier), a COTS product, is based on the Tivoli Management Platform (TMP), an architecture and a set of tools for managing client/server systems. More information can be found on Tivoli in Section 4.2.2 and Tivoli Enterprise Console and Tivoli Administration in Section 4.4.6. T/Courier adds software distribution capability to the Tivoli management environment and thus enables the operations staffs at the SMC and the DAAC sites to distribute ECS software, database information, and software documentation as well as commercial software across a multi-platform ECS network. T/Courier provides a centralized software distribution capability to add new software, update existing software with newer versions, and synchronize software on distributed systems.

T/Courier must be installed on all host platforms that will be involved in the distribution. Once installed, T/Courier enables the creation and distribution of profiles (file packages) from one UNIX host to another host or hosts. The file package defines the source path of the files and/or directories to be distributed, defines the destination path of the files and/or directories being distributed, and contains specific instructions on how to set up the files on the receiving platform. T/Courier can distribute a file package to a single host platform or to multiple host platforms concurrently. The distribution can be set up to occur immediately or at some scheduled time.

T/Courier has a graphical user interface (GUI) and a command line interface (CLI). The GUI provides basic software distribution capability through a set of screens. The CLI provides basic distribution capability and other capability (for more advanced users) through a set of T/Courier commands. The GUI capability is described in this section. Refer to the T/Courier User's Guide and Reference manuals for information on the CLI capability.

The most frequently used software distribution functions are listed in Table 4.3.5-1.

**Table 4.3.5-1. Common ECS Operator Functions Performed with T/Courier**

<b>Operating Function</b>	<b>Command/Script or GUI</b>	<b>Description</b>	<b>When and Why to Use</b>
Create file package	Series of GUIs	Create a definition of a package that points to specific files for distribution	Use when there is a new set of files to be distributed or when updates needs to be distributed.
Preview file package	GUI	Preview the contents of a file package	Use to check the listing of files and/or directories that are about to be distributed.
Distribute file package	GUI	Distribute software files.	Use to distribute software to specified hosts.
Schedule distribution	GUI	Schedule distribution of files.	Use to initiate distribution activity and cause it to occur at some predetermined date and time.
Remove file package	GUI	Remove file package from hosts.	Use to remove distributed file packages from the hosts that received them.

#### **4.3.5.1 Quick Start Using Tivoli/Courier**

This section presents an orientation of T/Courier. The underlying assumptions are that:

- Tivoli and T/Courier has been installed on all of the platforms that will be involved in the distribution
- The user has been granted administrator privileges for T/Courier activity.
- Location of the files to be distributed should be known in advance of executing Courier.
- The files' location will be entered into one of the Courier screens.
- Tivoli/Courier can only be accessed through Tivoli.

The documentation of Tivoli Courier used as a basis and referenced in this section is for version/release 3.0.1, contained in ECS Release 4.

##### **4.3.5.1.1 Invoking Tivoli/Courier From the Command Line Interface**

To initiate the execution of T/Courier via Tivoli, log into a Unix platform that has T/Courier installed and enter the following commands:

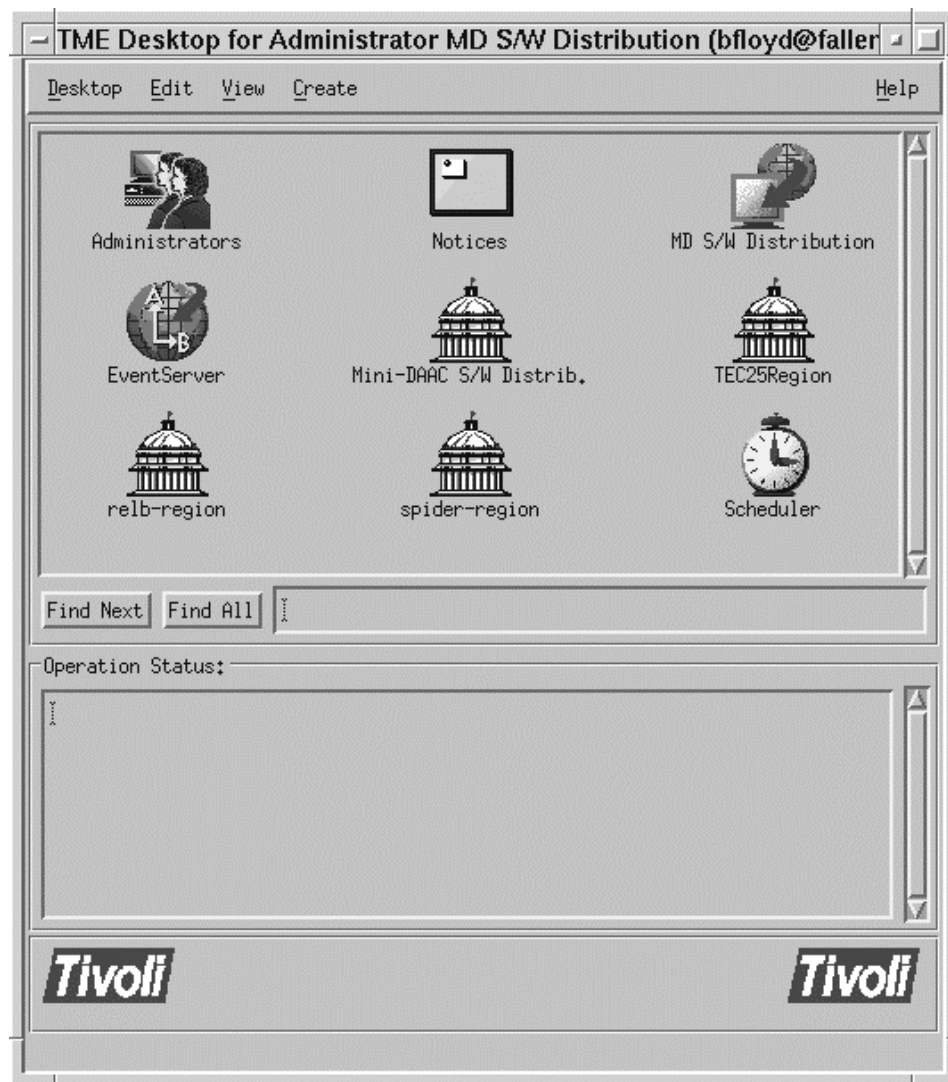
```
source /etc/setup_env.csh (in c shell)

./etc/Tivoli/setup_env.sh (in Bourne or bash shell)

tivoli -font fixed
```

##### **4.3.5.2 Tivoli/Courier Main Screen**

Upon execution, Tivoli will splash its logo screen and then display its Main Screen as shown in Figure 4.3.5-1. This screen provides access to software distribution functions through a series of icons and dialogs. The top part of the screen holds the icons needed to get things started and the bottom part displays the type of process and results of the process that is initiated.

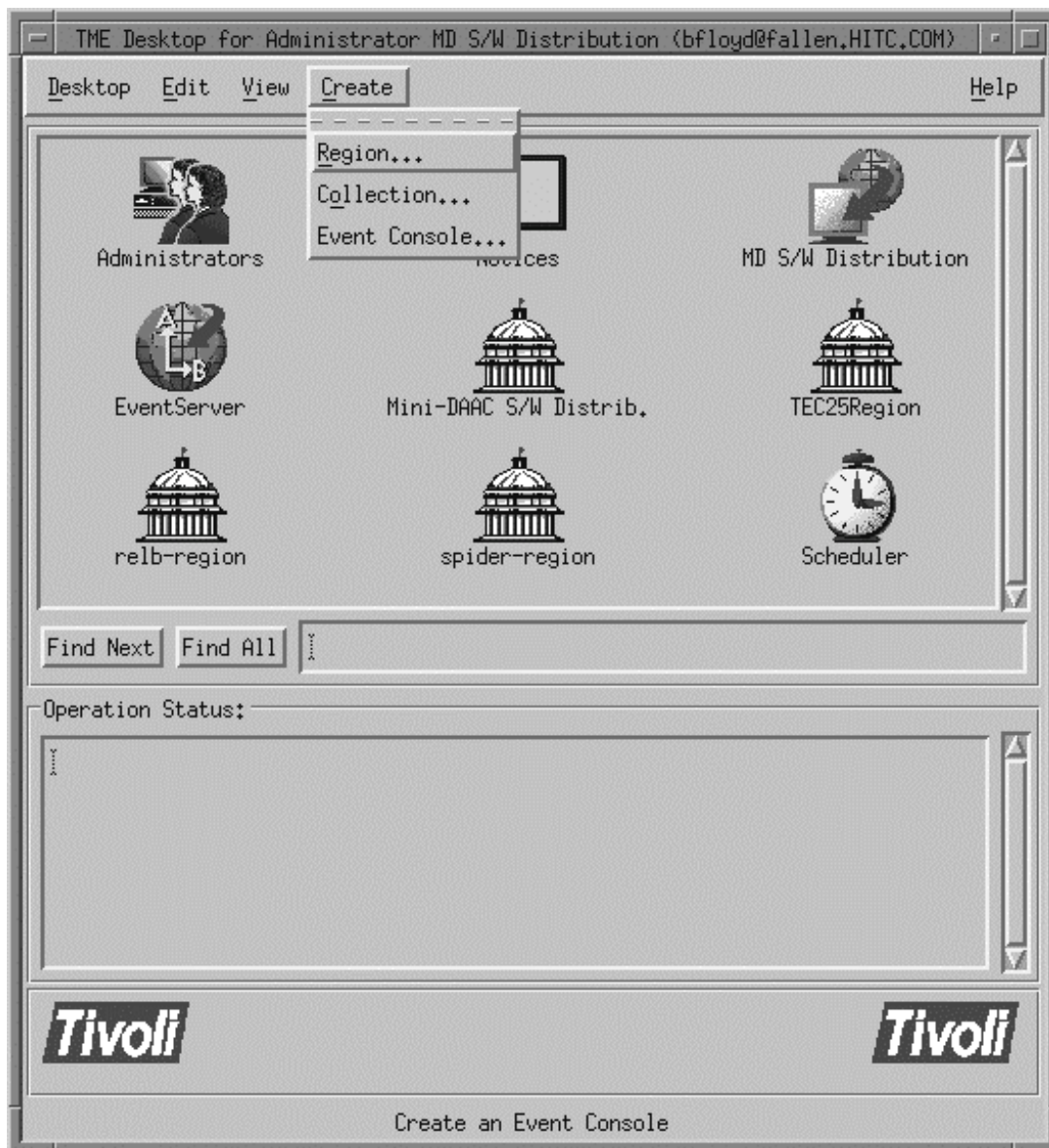


**Figure 4.3.5-1. T/Courier Main Screen**

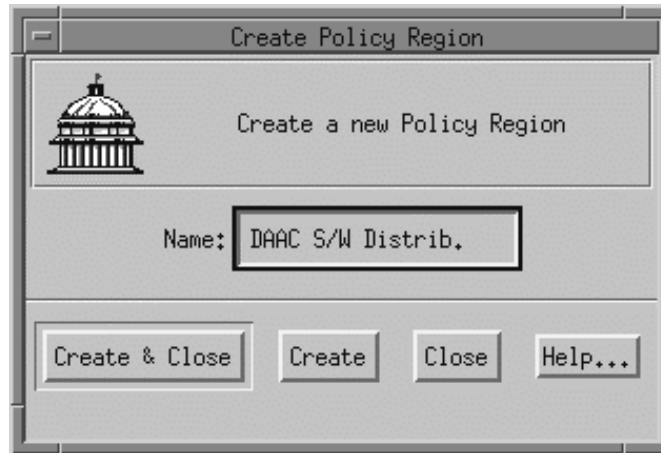
#### **4.3.5.2.1 Establish Policy Region**

Before using T/Courier to distribute software, system resources must be identified and organized for distribution activity. T/Courier uses policy regions to enable organization of software distribution activity. A policy region is a collection of resources that share one or more common policies and it must be created before any distribution can occur.

Create a policy region by clicking the "Create " menu on the T/Courier Create Events Pop-up (see Figure 4.3.5-2) and select "Region." Upon selection of the "Region" option, a Create Policy Region pop-up will appear as shown in Figure 4.3.5-3. Enter a descriptive name for the region that you're about to create and then click the **"Create and Close"** button.

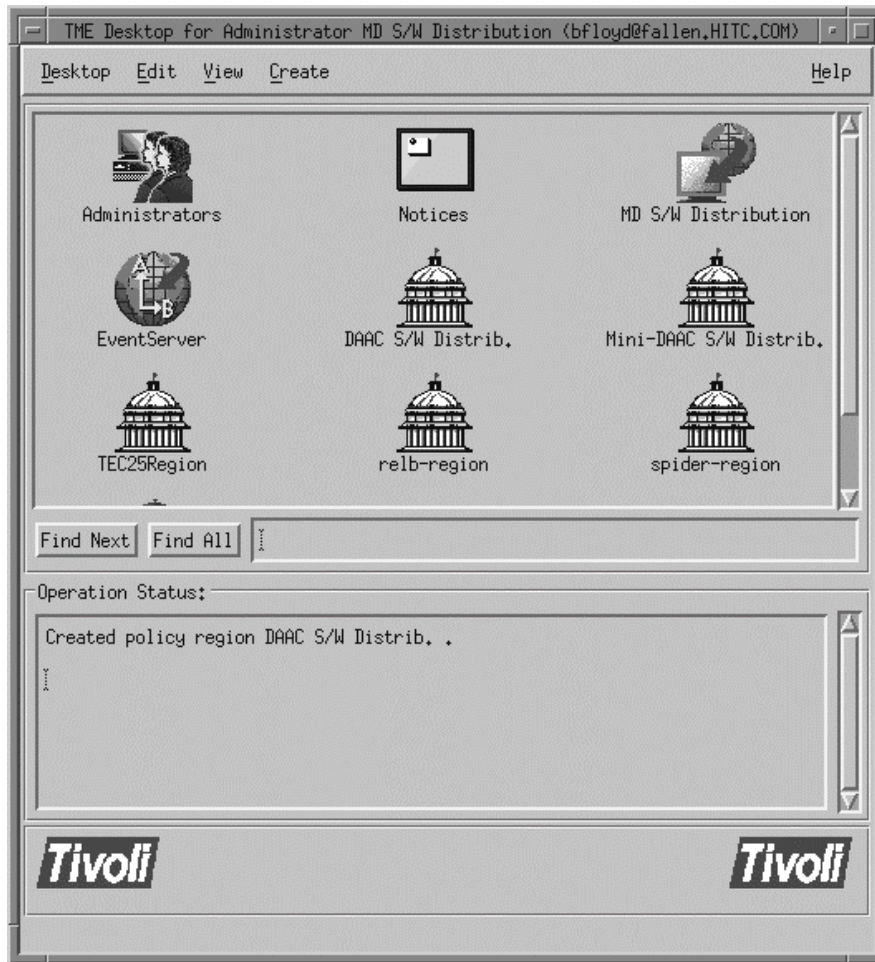


**Figure 4.3.5-2. T/Courier Create Events Pop-up**



***Figure 4.3.5-3. T/Courier Create Policy Region Pop-up***

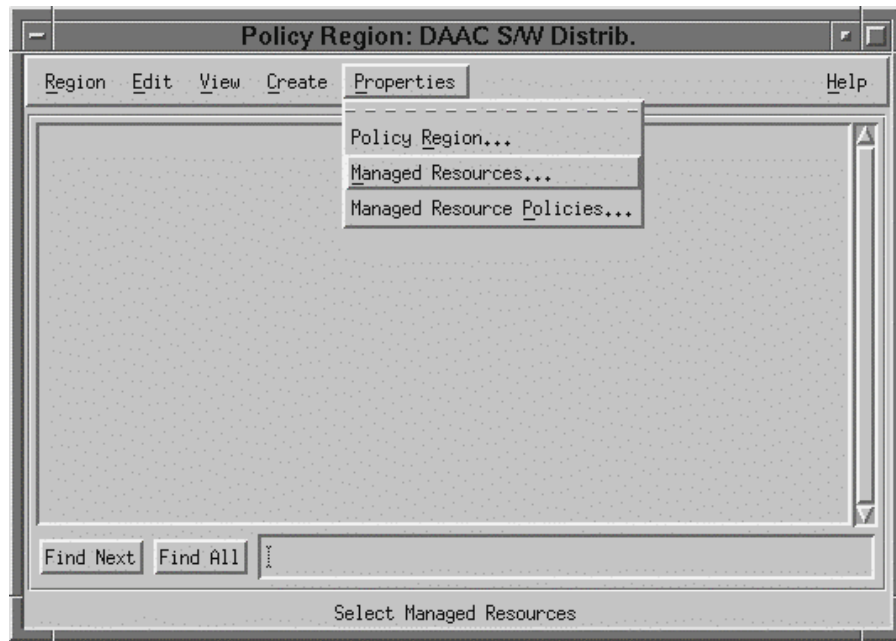
At this point a new policy region (for example, DAAC S/W Distrib.) has been created. This new icon appears on the T/Courier Main Screen, and the status part of the T/Courier main screen reflects the status of the “Create Policy Region” action as shown in Figure 4.3.5-4



**Figure 4.3.5-4. T/Courier Operation Status display, Main Screen**

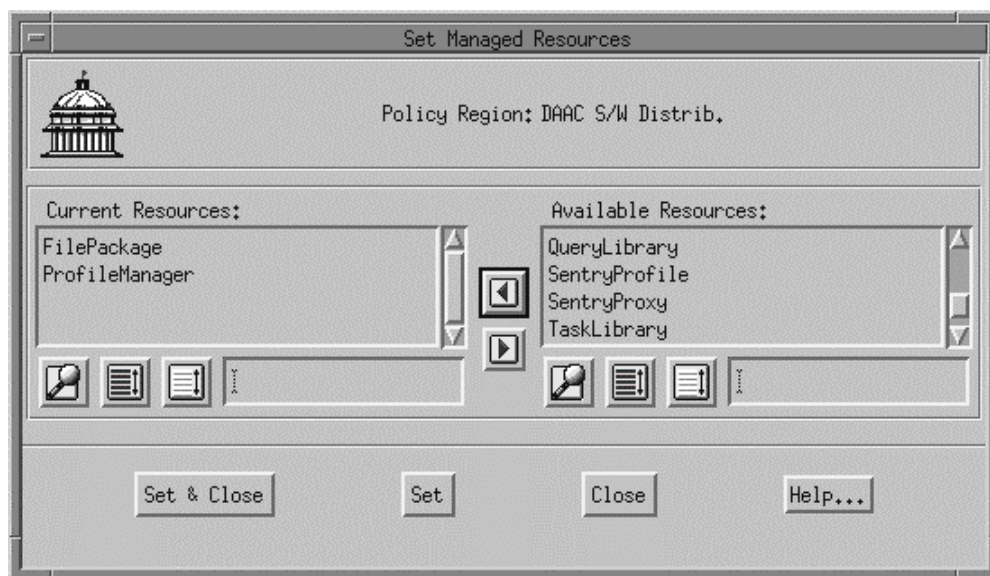
#### **4.3.5.2.2 Select Managed Resources**

Once established, a policy region has to be defined in terms of the resources that it will managed. Double click the new policy region's icon to bring up the first of several screens that has to be completed . The Policy Region pop-up is displayed as shown in Figure 4.3.5-5. Click the "Properties" menu and then select the "Managed Resources " option. This causes



**Figure 4.3.5-5. T/Courier Properties Menu of Policy Region Pop-up**

the “Set Managed Resources” Pop-up to appear as shown in Figure 4.3.5-6. The Set Managed Resources” pop-up enables the selection of resources to be managed. For software distribution activity, FilePackage and ProfileManager are the resources to select and move from the Available Resources side to the Current Resources side of the “Set Managed Resources” Pop-up. Click the “Set and Close” button to save the selections.

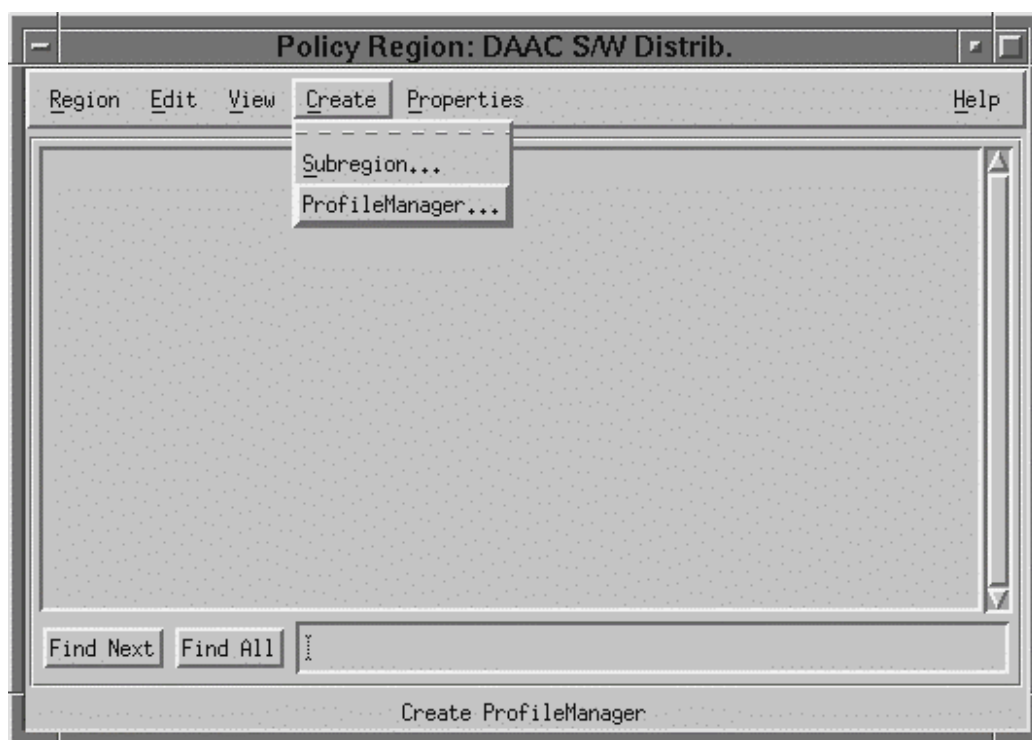


**Figure 4.3.5-6. T/Courier Set Managed Resources Pop-up**



#### 4.3.5.2.3 Create Profile Manager

A policy region is made up of profile managers. A profile manager contains one or more profiles. A profile is also called a file package and it describe the location, path and other characteristics of the files to be distributed. To set up a profile manager, click the “Create” menu on the Policy Region pop-up as shown in Figure 4.3.5-7. Then select the ProfileManager option.



**Figure 4.3.5-7. T/Courier Create Menu of Policy Region Pop-up**

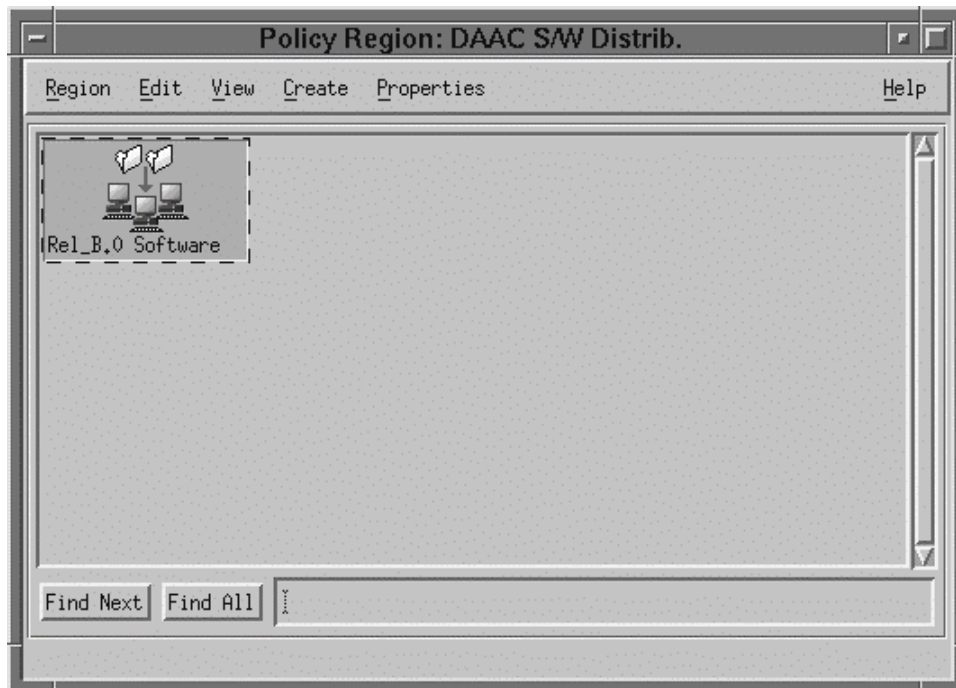
Selection of the ProfileManager option causes the “Create Profile Manager” Pop-up to appear. See Figure 4.3.5-8. Enter a descriptive name for the profile manager being created. Suggest that the profile manager’s name be indicative of the type of profiles that it will contain. A discernible profile manager name will enable one to ascertain the type of profiles the manager holds. This will be particularly useful when other profile managers are added to the region as it will preclude your having to go into a profile manager to determine what profiles (file packages) it controls. Click the “**Create & Close**” button to save your entry.





**Figure 4.3.5-8. T/Courier Create Profile Manager Pop-up**

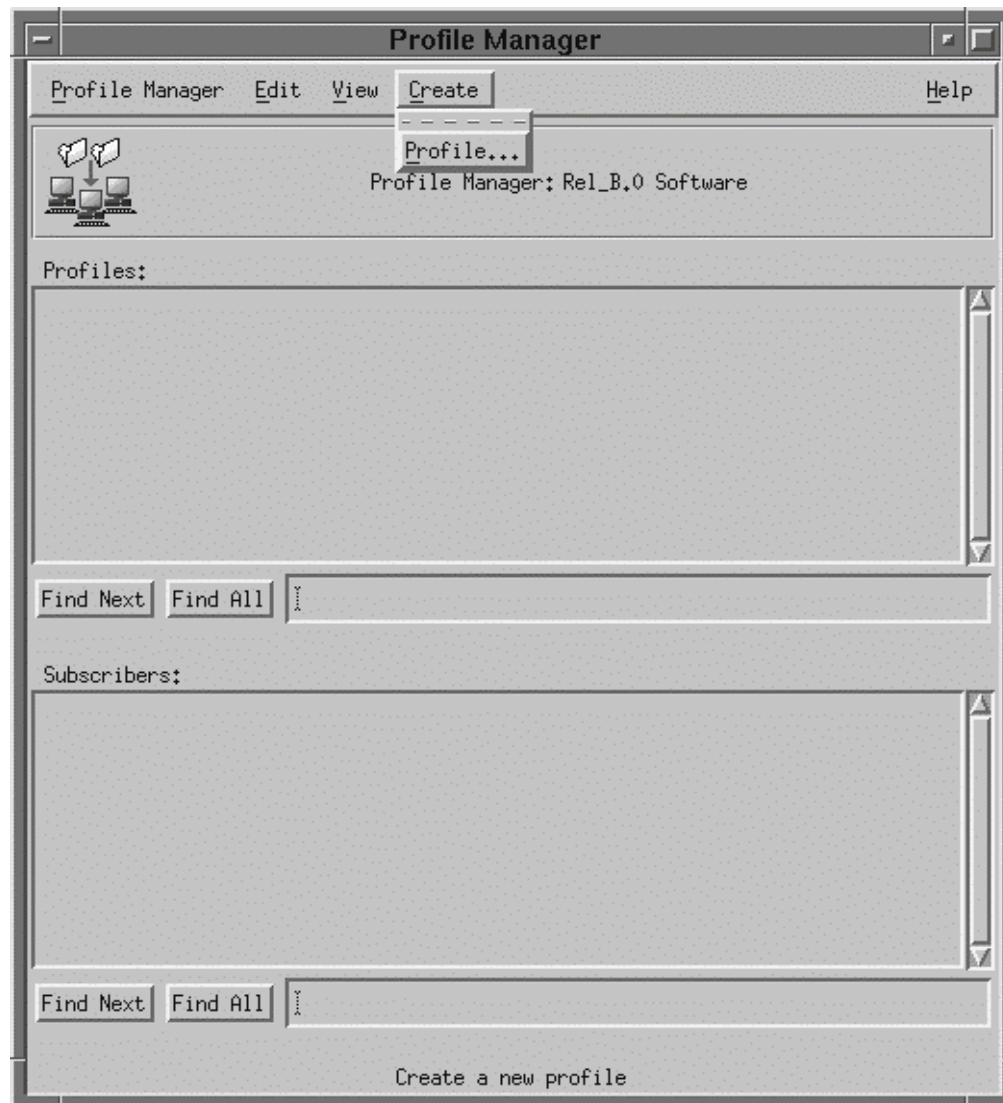
The Policy Region pop-up will now display an icon for the newly created profile manager as shown in Figure 4.3.5-9 below.



**Figure 4.3.5-9. T/Courier Create Profile Manager Pop-up (showing new icon)**

#### 4.3.5.2.4 Create Profile

A profile or file package must be created for the Profile Manager to managed. A profile is a description of a set of files that will be distributed to the subscribing platforms. Double click the newly created profile manager's icon to bring up the Profile Manager (Rel\_B0\_Software) pop-up. Click the "Create" menu on the Profile Manager Screen and then select the "Profile" option as shown in Figure 4.3.5-10.



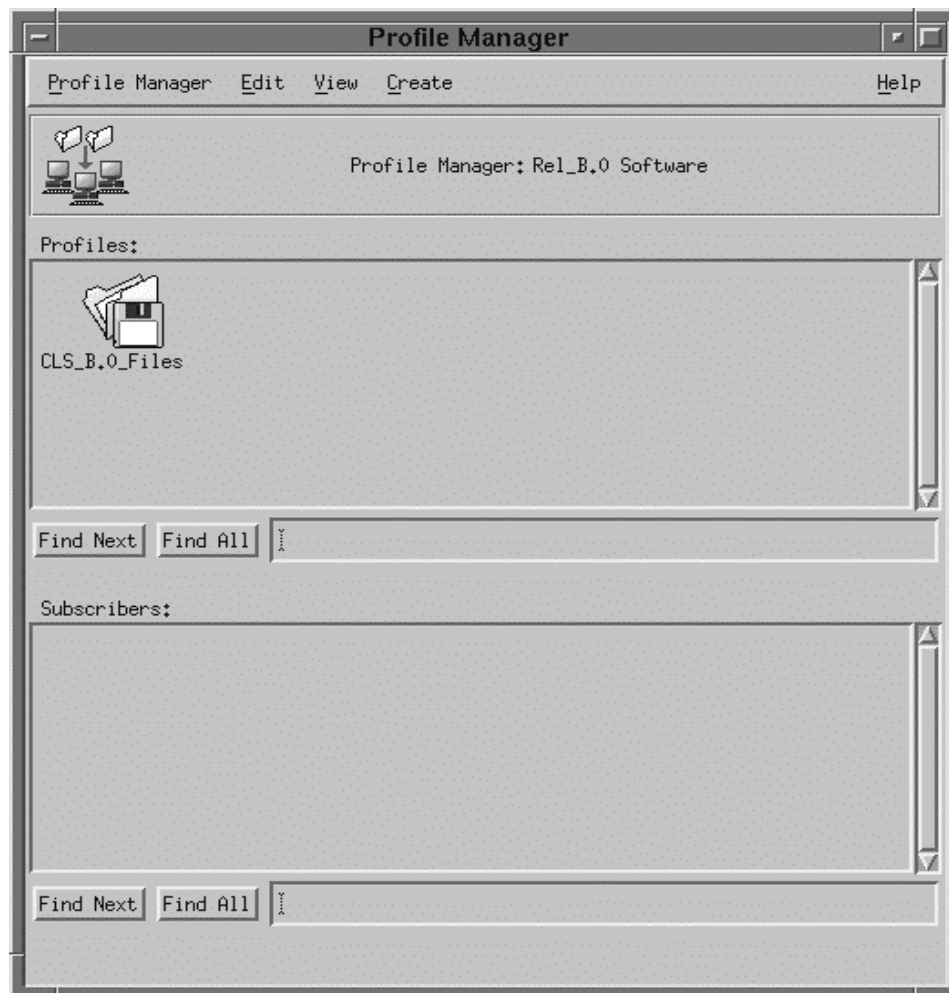
**Figure 4.3.5-10. T/Courier Create Menu of Profile Manager Pop-up**

A "Create Profile" Pop-up is displayed as shown in Figure 4.3.5-11. Enter a descriptive name in the "Name/Icon Label" text box.



**Figure 4.3.5-11. T/Courier Create Profile Pop-up**

Recommend that the chosen name be indicative of the type or category of software files that will be in the profile (file package). Click the “**Create & Close**” button to store your entry and then go back to the Profile Manager pop-up. The Profile Manager now has one profile as shown in Figure 4.3.5-12.



**Figure 4.3.5-12. T/Courier Profile display of Profile Manager Pop-up**

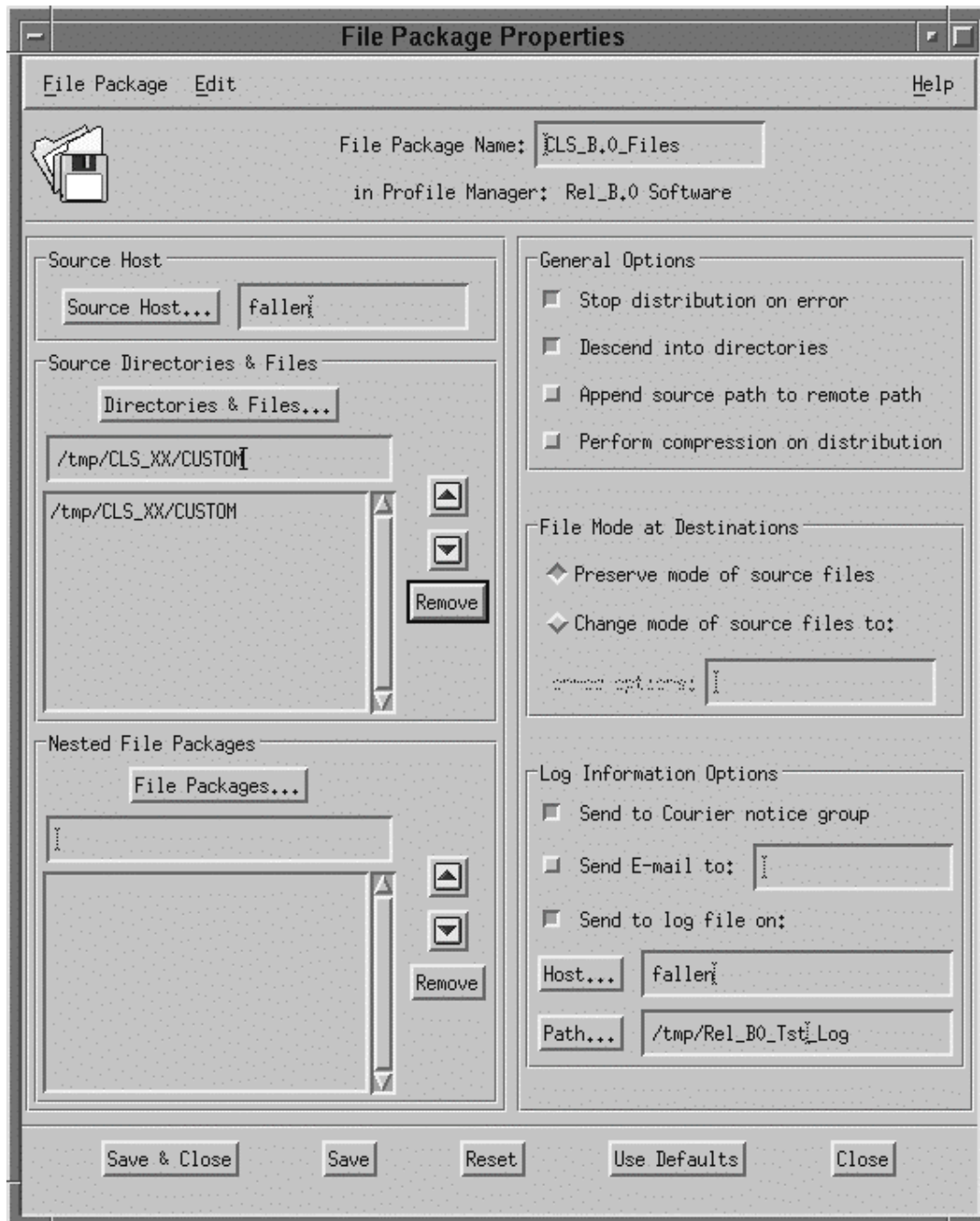
#### **4.3.5.2.5 Define Profile**

The newly created profile (file package) has to be defined. Double click the newly created profile's icon to initiate the definition process. The File Package Properties pop-up is then displayed as shown in Figure 4.3.5-13. Enter the name of the source host, host where a copy of the files to be distributed will reside, in the Source Host box. Enter the path (s) to the source files in the Directories & Files box.

For General Options, select "Stop distribution on error" option and the "Descend into directories" option (Note, selection of the "Descend into directories" option will caused the sub-directories of the listed directories to be distributed as well. If the "Descend" option is not selected, only the top level of the specified directory will be distributed; its sub-directories (if any) will not be distributed).

For File Mode at Destinations Option, select Preserve mode of source files. However, if you need to change the mode of the files at the destination, select the “Change mode of” line and then enter the new mode.

For Log Information Options, select “Send to Courier notice group” and “Send to log file on” option. Enter the name of the host that will hold the log file and the path to the log file.



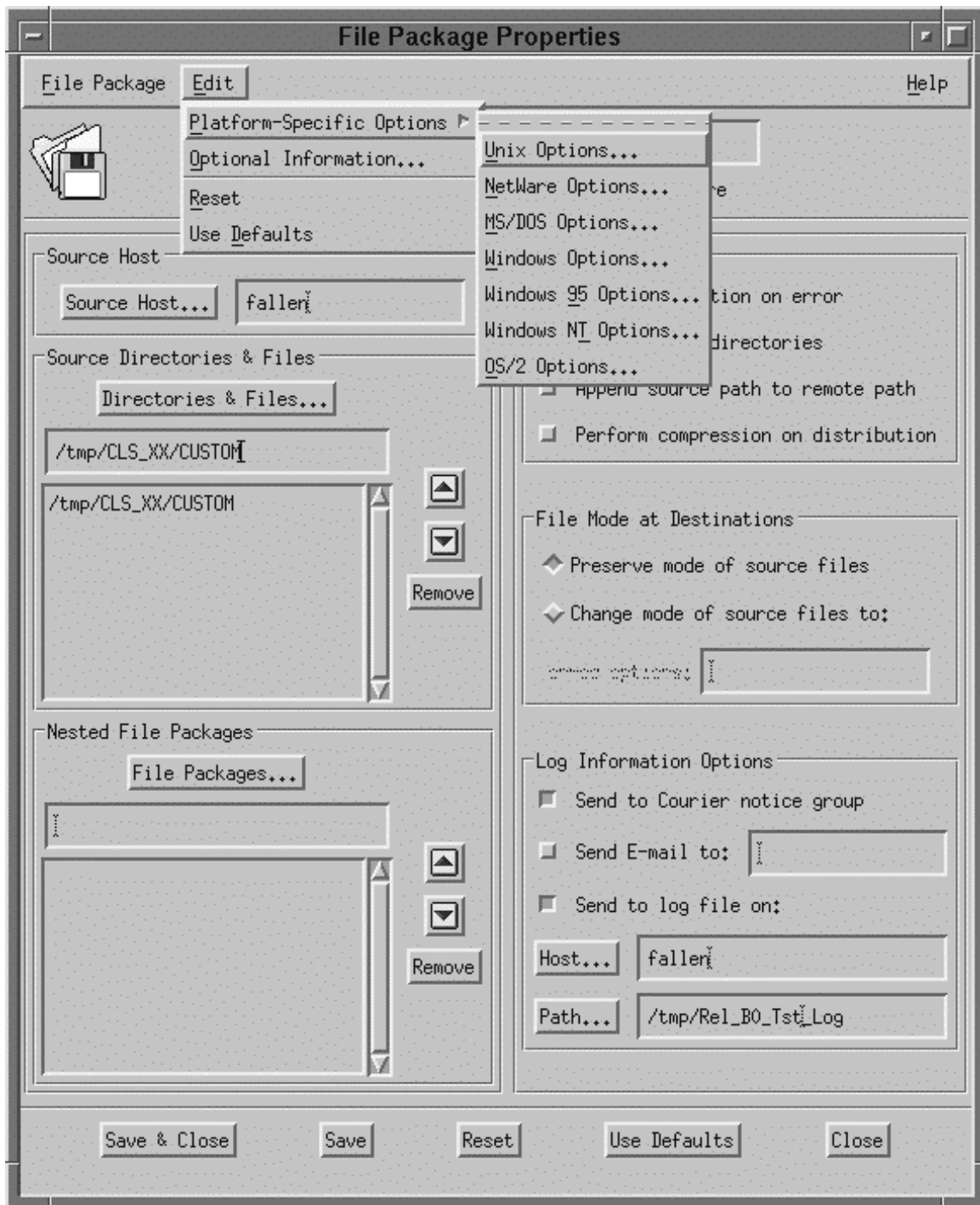
**Figure 4.3.5-13. T/Courier File Package Properties Pop-up**

[Note, always establish a log file. T/Courier will send a detailed status message to an identified log file. Otherwise, only a brief error message and the brief message sent to the Operation Status box on the main screen are provided and there is usually insufficient information to determine the cause of a problem in the Operations Status Box. Note also that the T/Courier default for writing messages to a log file is to overwrite the contents of the log file each time an entry is made to the log. This default can be changed so that messages are appended to the log file each time an entry is made. The Tivoli Courier User's Manual explains how to change the log file default and also provides a detailed description of the other settings and options.]

This completes the necessary entries for the File Package Properties pop-up.

#### **4.3.5.2.6 Establish UNIX Options**

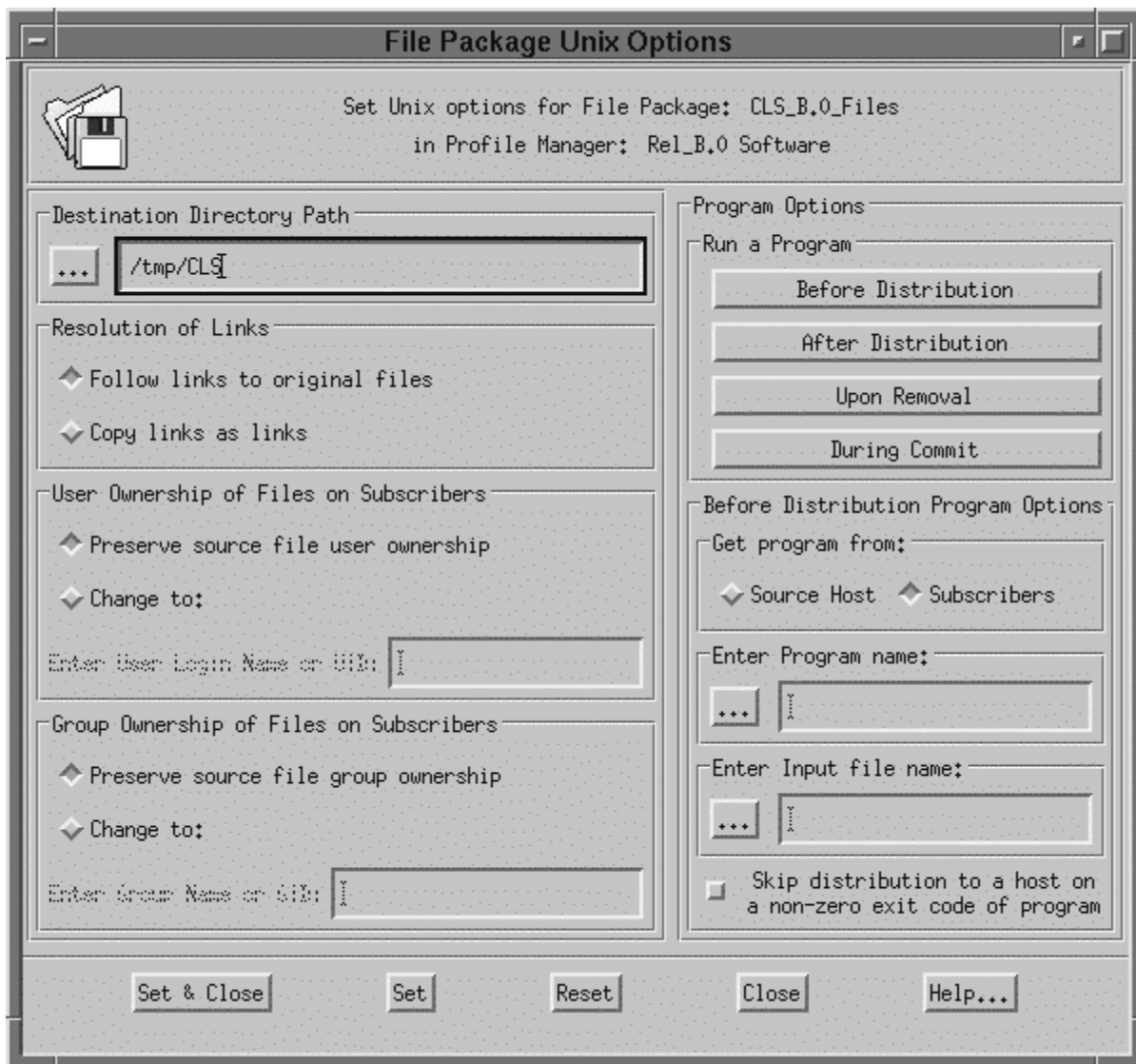
T/Courier can distribute software files to platforms having different operating systems. Therefore, certain T/Courier options are operating system specific. Since ECS platforms are Unix platforms, this section will cover the use of the File Package Unix Option Screen. Refer to the T/Courier User's Manual for the use of other operating systems options screens. To set up the File Package Unix Options Screen, first click the Edit menu on the File Package Properties pop-up. Select the Platform-Specific Options and then the UNIX Options as shown in Figure 4.3.5-14. The File Package UNIX Options pop-up is then displayed.



**Figure 4.3.5-14. T/Courier Edit Menu of File Package Properties Pop-up**

as shown in Figure 4.3.5-15. On the File Package UNIX Options pop-up, enter the path in the





**Figure 4.3.5-15. T/Courier File Package UNIX Options Pop-up**

Destination Directory Path box where the files are to be placed on the destination (receiving) host.

Note of caution: Make sure that the Destination Directory Path being entered is correct. T/Courier operates at the root level. If the specified directory/files names already exist on the destination platform, T/Courier will overwrite them. If the destination directory path does not exist on the destination platform, T/Courier will create the path and then install the files.

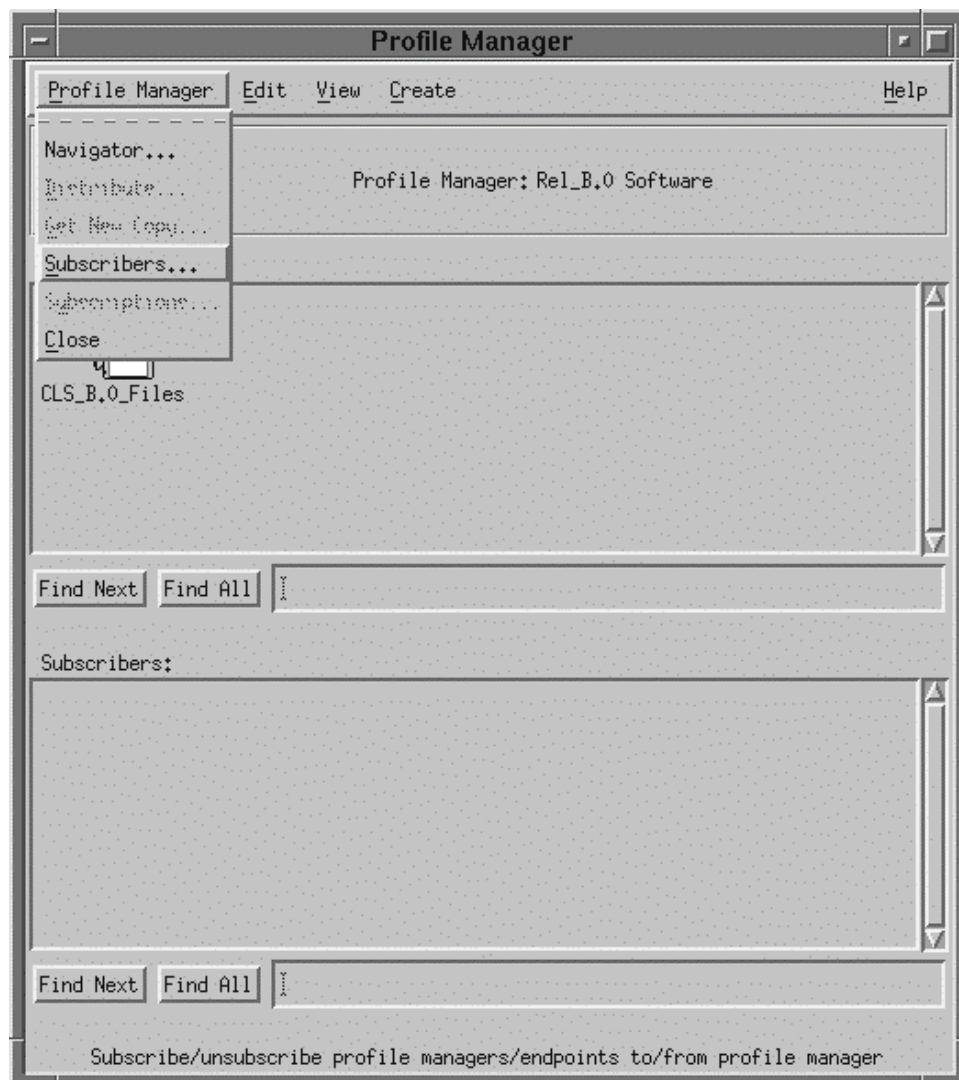
Allow the default selections (Follow links to original files; Preserve source file user ownership; and "Preserve source file group ownership") to stand or select options desired.

The “Program Options” enables an advanced T/Courier user to execute programs or UNIX scripts at some point during the distribution process. Refer to the T/Courier User’s Manual for use of the “Program Options.” Click the “Set & Close” button to store your entries and selections. On the File Package Properties pop-up, click the “Save and Close” button.

This concludes the definition of the file package.

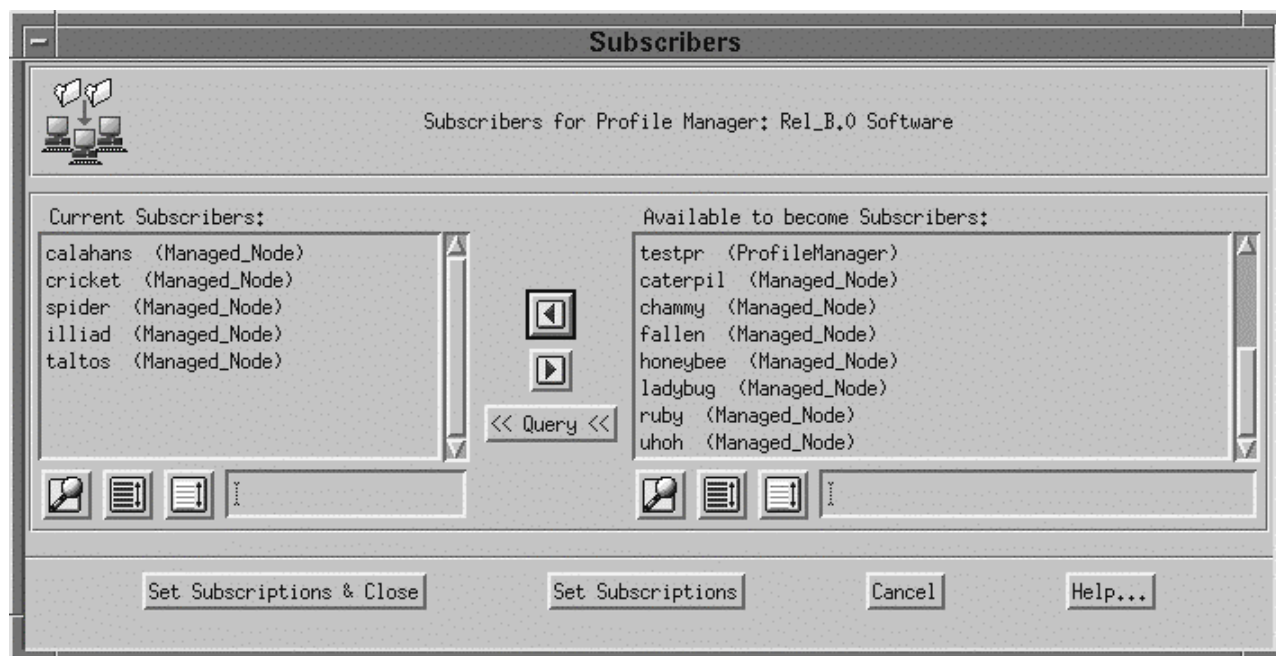
#### 4.3.5.2.7 Select Subscribers

Subscribers (hosts) that will receive the profile, must be identified for the profile manager. On the Profile Manager pop-up, click the Profile Manager menu and select the “Subscriber” option (as shown in Figure 4.3.5-16) to identify subscribers for the profile manager, “Rel\_B.0 Software.”



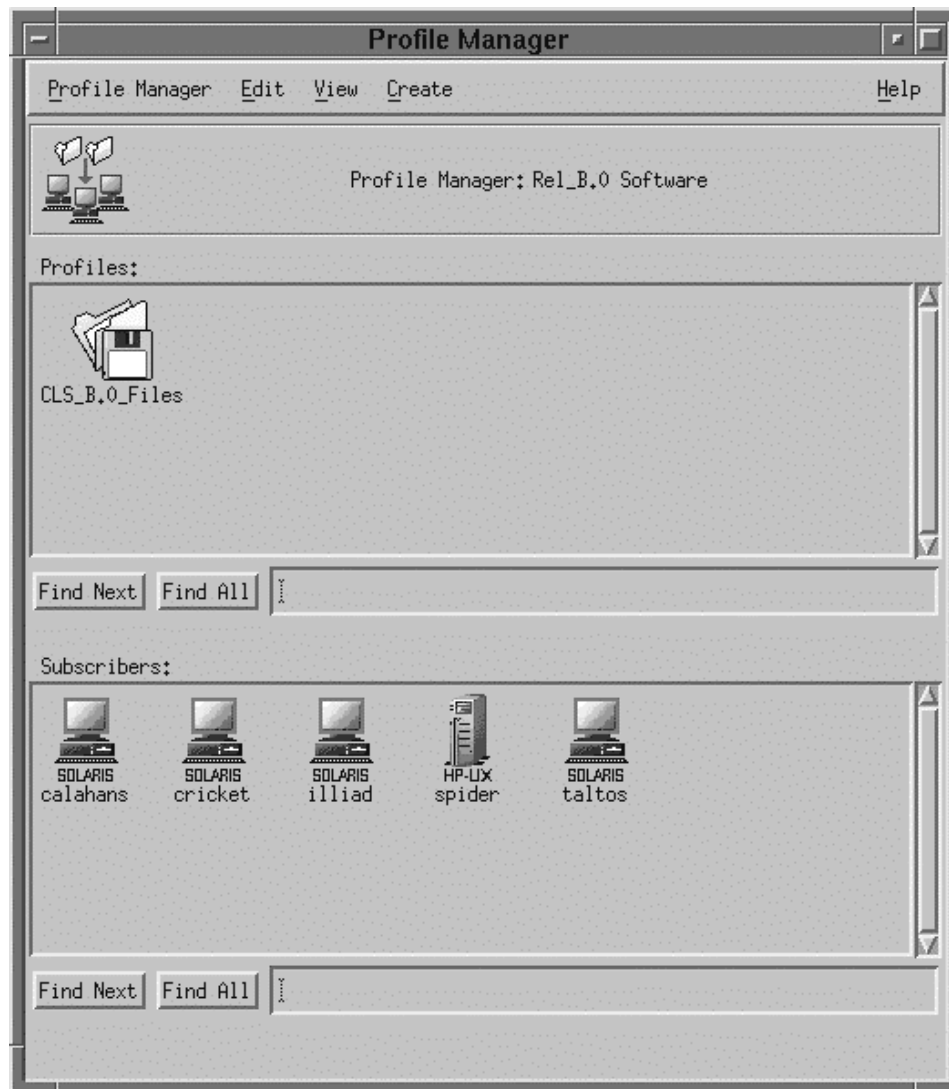
**Figure 4.3.5-16. T/Courier Profile Manager Menu of Profile Manager Pop-up**

Upon selection of the Subscribers option, the Subscribers pop-up is displayed. Select and move appropriate subscribers from the “Available to become Subscribers” side to the “Current Subscribers” side of the Subscribers pop-up. For software distribution purposes, the appropriate “Managed\_Node” should be moved to the Current Subscribers side. [Note, the “Current Subscribers” side is always initially blank. Figure 4.3.5-17 shows subscribers that have already been moved to the “Current Subscribers” side.] Click the “**Set Subscription & Close**” button to save your selections.



**Figure 4.3.5-17. T/Courier Set Subscribers Pop-up**

The Profile Manager, Rel\_B.0 Software, now consists of a profile and a set of subscribers as shown in Figure 4.3.5-18.



**Figure 4.3.5-18. T/Courier Complete Profile Manager Pop-up**

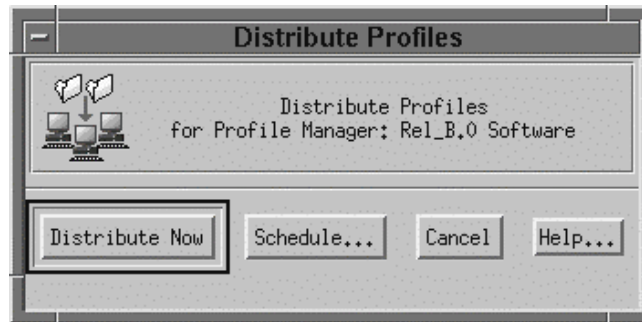
#### **4.3.5.2.8 Distribute Software**

Once a profile has been defined, software distribution can take place. On the Profile Manager Screen, First click the profile's icon and then click the appropriate subscriber(s) icon. See Figure 4.3.5-19. Next, click Profile Manager menu and select the Distribute option. In Figure 4.3.5-19 the profile, CLS\_B.0\_Files, and a subscriber, calahans, have been selected.



**Figure 4.3.5-19. T/Courier Distribution Option of Profile Manager Pop-up**

The “Distribute Profiles” pop-up is displayed as shown in Figure 4.3.5-20. Click the “**Distribute Now**” button to initiate distribution activity immediately. Note, that there is also a “**Schedule**” button which enables the scheduling of distribution to occur at some other time. Refer to the T/Courier User’s Manual for information on the use of this feature.

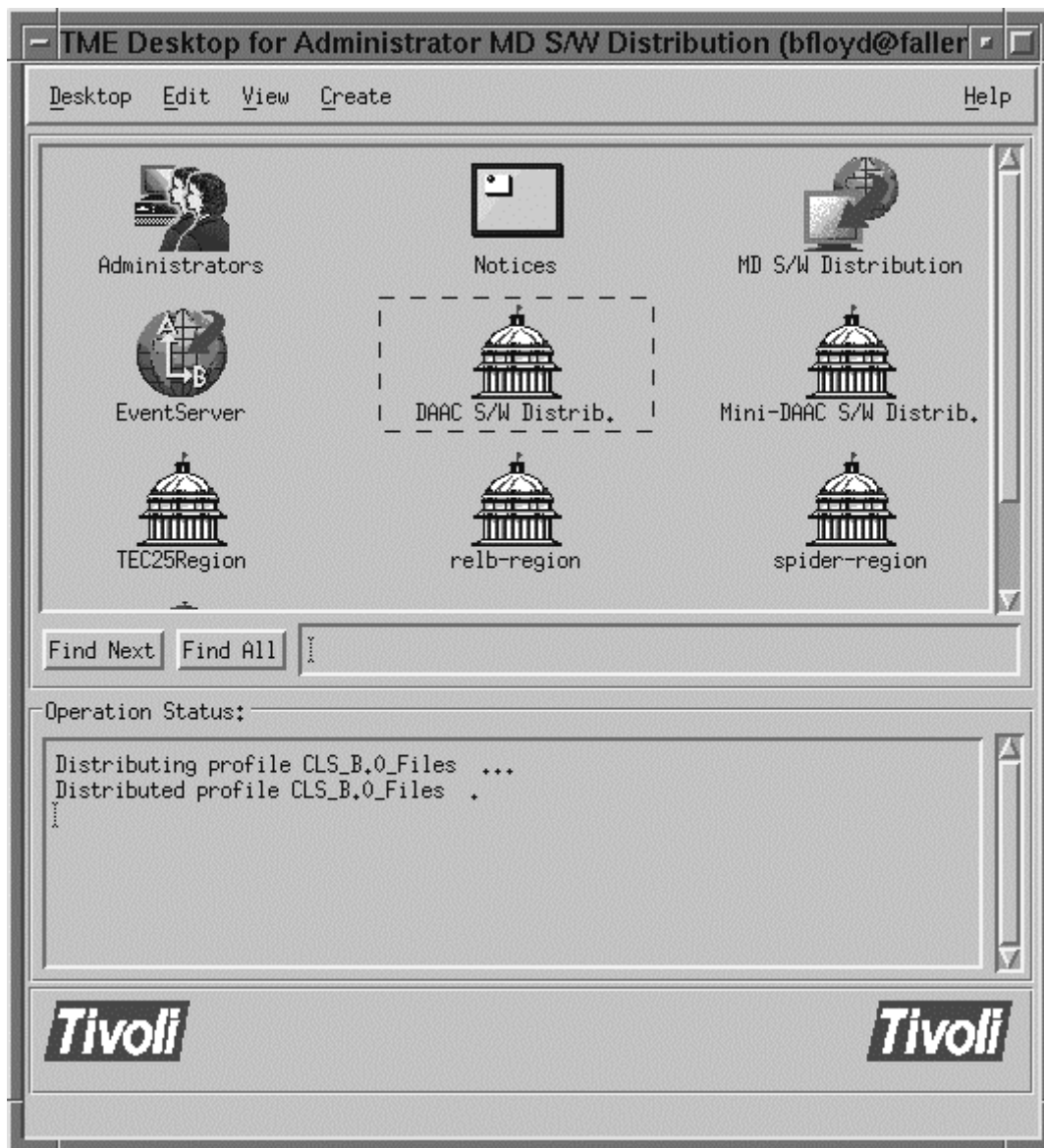


**Figure 4.3.5-20. T/Courier Distribution Profiles Pop-up**

When the “Distribute Profiles” pop-up disappears, the distribution activity is completed.

#### **4.3.5.2.9 Check Status of Software Distribution**

Go back to the T/Courier main pop-up and check the Operation Status box for results of the distribution. See Figure 4.3.5-21 for an example of the Operation Status display. For a detailed description of the distribution process, check the log file.



**Figure 4.3.5-21. T/Courier Distribution Status of Main Screen**

The Log file (if set up on the File Package Option pop-up as in Figure 4.3.5-13) will always provide a more definitive description of the distribution process. For example, contents of the logfile, Rel\_B0\_Tst\_Log, for the above distribution are as follows:



```
File Package:  "CLS_B.0_Files  "
Operation:      install  (m=5)
Finished:       Fri Jun 13 10:39:28 1997
-----
Source messages:
<none>
-----
calahans:SUCCESS
/tmp/CLS      : creating path
=====
```

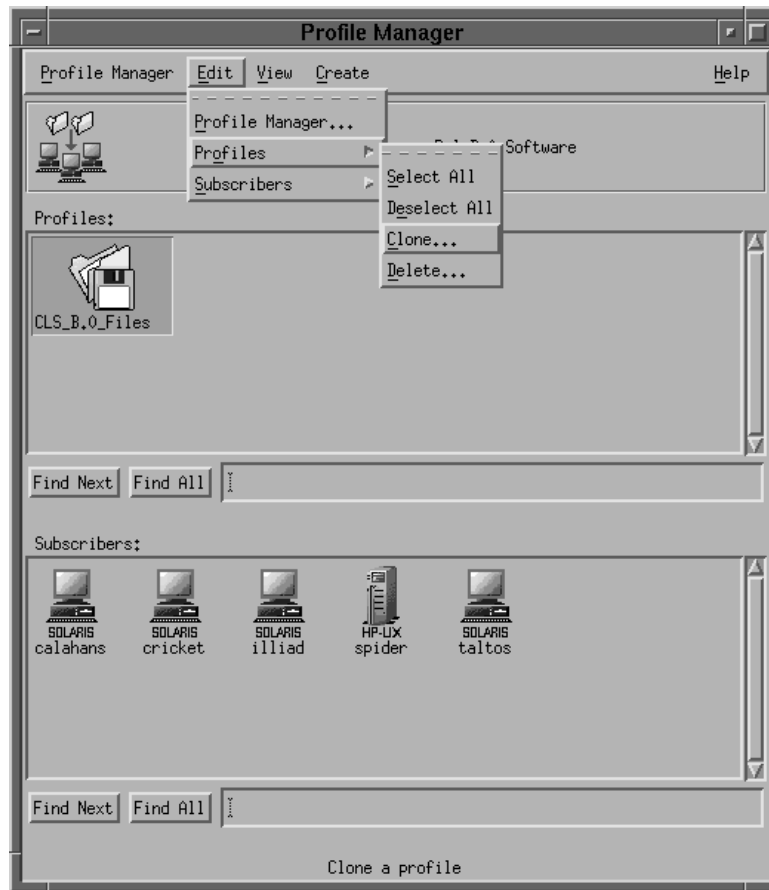
**Figure 4.3.5-22. Contents of the logfile, Rel\_B0\_Tst\_Log**

This log file's content indicates that the software distribution to the subscribing host, calahans, was successful. If an error occurs during distribution and/or the distribution is not successful, the log file will contain a detailed message concerning the problem and it will indicate that the software distribution to the listed platform failed.

#### **4.3.5.2.10 Creating Additional Profiles**

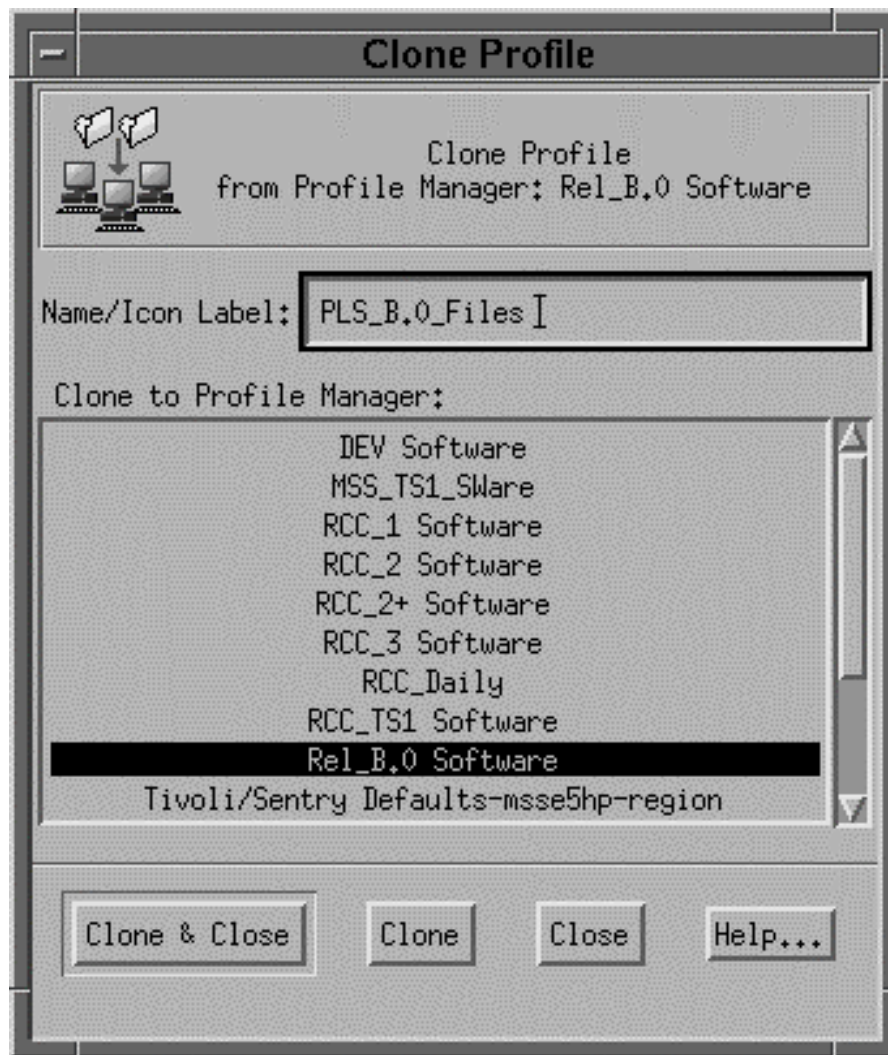
Additional profiles for a profile manager can be created by repeating the aforementioned procedure or by making clones of the profile created previously. Cloning a profile produces a duplicate of the profile being cloned in terms of its definition. If there are a large number of profiles that have to be created with the same settings but different source directory and files and/or destination directory, cloning will save time a lot of profile preparation time and insure settings consistency among the profiles.

To clone a previously created profile, click and highlight the profile to be cloned as shown on the Profile Manager's pop-up (Figure 4.3.5-23). Then click the "Edit" menu on that pop-up.



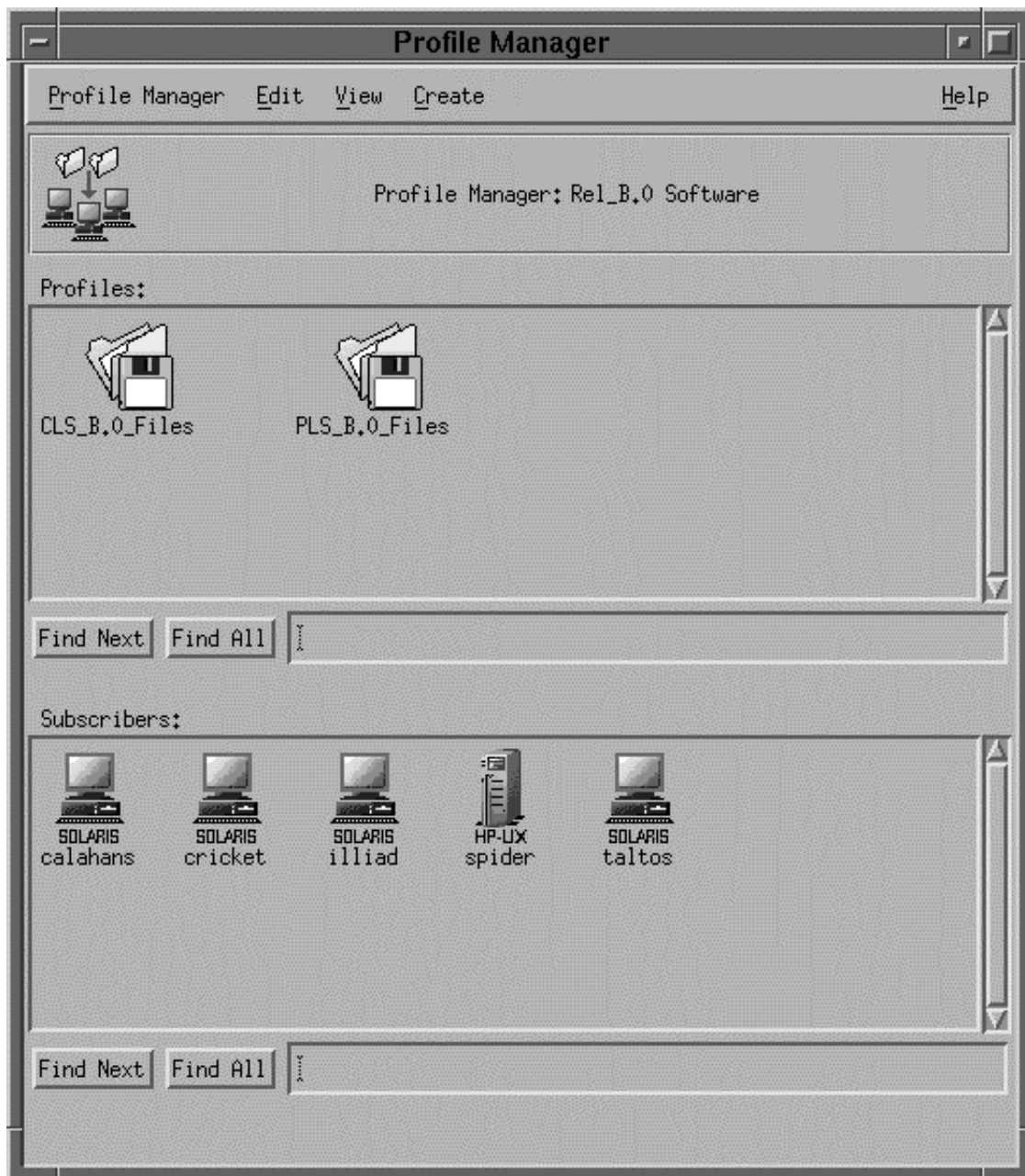
**Figure 4.3.5-23. T/Courier Clone Option of Profile Manager Pop-up**

On the drop down menu, click the “Profiles” option and then click the “Clone” option on the next drop down menu. The “Clone Profile” pop-up will appear as shown in Figure 4.3.5-24. The “Clone Profile” pop-up enables one to create one clone and then close the screen or one can create many clones before closing the screen. If there are more than one clone to be created, enter the name of the new profile into the “Name” box and click the clone button. The newly created profile will appear on the “Profile Manager” pop-up (see Figure 4.3.5-25) and the “Clone Profile” pop-up’s Name/Icon Label box will be cleared so that another new profile’s name can be entered. Repeat this process until all of the profiles have been



**Figure 4.3.5-24. T/Courier Clone Profile Pop-up**

created. When the name of the last profile has been entered, click the **“Clone and Close”**



**Figure 4.3.5-25. T/Courier Profile Clone display of the Main Screen**

button. If there is only one clone to be created, enter the name of the new profile into the “Name/Icon Label” box and then click the **“Clone and Close”** button. The newly created profiles will be displayed on the Profile Manager pop-up and this completes the creation process.

Now, the definition of the clone profile have to be revised before it can be used in software distribution activity. For each clone profile, click the profile’s icon and bring up its File Package

Properties pop-up (as described in Section 4.3.5.2.5) revise the source directory/files section. Then bring up the File Package Unix Options pop-up (as described in Section 4.3.5.2.6) and revise the destination directory path if necessary. Once the source directory/files and the destination path has been appropriately revised, the profile is ready to be used in software distribution activity.

#### **4.3.5.3 Required Operating Environment**

T/Courier requires a UNIX platform as the source files' host. The target or destination platforms can be UNIX platforms or PC's. ECS T/Courier is used on Sun hosts (Solaris 2.5 OS), Hewlett Packard hosts (HP 10.0.1 OS), and SGI (SGI Irix 6.2) hosts.

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 Tivoli/Courier, refer to the ECS Baseline Information System web page, URL <http://cmdm.east.hitc.com/.4.3.5.4> Databases

Tivoli maintains a proprietary data store. Tivoli data is only accessible through the GUI described in Section 4.5.3.2..

#### **4.3.5.4 Databases**

None

#### **4.3.5.5 Special Constraints**

The operator must have Tivoli Administrator privileges to create the regions and associated components.

#### **4.3.5.6 Outputs**

Outputs consist of T/Courier information displayed on the GUIs discussed in Section 4.3.5.2, updates the Tivoli data store, and event and error messages discussed in Section 4.3.5.7.

#### **4.3.5.7 Event and Error Messages**

All event and summary error messages are displayed in the Operation Status box, main screen. Detailed messages would be placed in the log file, if one is established as described in Section 4.3.5.2.5.

#### **4.3.5.8 Reports**

None.

This page intentionally left blank.

### 4.3.6 FLEXlm

FLEXlm is a commercially available network license management product that helps the ECS M&O staffs at the DAACs, EOC, and SMC administer licenses and enforce licensing provisions for FLEXlm-enabled COTS software at the site. It enforces licensing provisions based on information from vendor-provided license keys and lets license administrators allow, deny, or reserve check out of licenses based on user, host, or display. FLEXlm handles floating (concurrent use) licenses, node locked licenses, and combinations of the two.

FLEXlm processing elements include license manager daemons, vendor daemons, license files, and FLEXlm-enabled applications. One or more license manager daemons control vendor daemon operations and enables client applications to contact them. Vendor daemons grant or deny concurrent use licenses requested by applications, tracking how many are checked out and by which users. License files are text files that contain the provisions for one or more licenses from one or more vendors, including the name of the vendor daemon needed to serve the license and the host(s) to use as license server(s). The applications communicate with the license and vendor daemons using embedded FLEXlm client software to request licenses in order to run.

FLEXlm permits use of single, multiple, or redundant server hosts, and can operate more than one license manager daemon on a given node. A license manager daemon serves all the licenses in the license file it uses, and different license files use separate license manager daemons (distinguished by the port number they use to communicate). In a redundant license server configuration, license manager daemons for a license file are executed on three server nodes such that all licenses in the file are available if any two out of the three server nodes is running. In a multiple license server configuration, licenses are allocated among multiple license files and a separate license manager daemon is run for each file.

Table 4.3.6-1 summarizes the operating functions that FLEXlm supports.

**Table 4.3.6-1. Common ECS Operating Functions Performed with FLEXlm (1 of 2)**

Operating Function	Function Name	Description	When and Why to Use
Start license manager	lmgrd	Starts FLEXlm's main daemon program which reads the license file and manages vendor daemons and the connections between them and their client applications.	Used to initiate license management server processes
Stop license manager	lmdown	Shuts down all license daemons (both lmgrd and all vendor daemons) on all nodes.	Used anytime to stop network license activities, such as when the license manager host is to be rebooted.



**Table 4.3.6-1. Common ECS Operating Functions Performed with FLEXlm (2 of 2)**

Operating Function	Function Name	Description	When and Why to Use
Install decimal format licenses	lminstall	Converts licenses between decimal and readable formats and between different versions of FLEXlm license formats.	Used anytime primarily to install decimal format licenses in readable format.
Read new licenses	lmreread	Causes the license servers to reread the license file they are using and start any new vendor daemons.	Used anytime to put the provisions of an updated license file into effect.
Monitor the status of network licensing activities	lmstat	Generates lists containing such information as active licenses, users of licensed product features, users of individual license management daemons, and status of server nodes.	Used anytime to check on the health and functioning of license server daemons, identify licenses installed, determine licenses in use, or review logged licensing events.
Switch to new report log	lmswitchr	Causes the license servers to use a new or different file as the report log.	Not used. Report logs can be read only by the FLEXadmin product. FLEXadmin is not provided in ECS due to security constraints (i.e., use of remote shell utilities).
Verify accuracy of license file	lmcksum	Performs a checksum of a license file	Used anytime to verify data entry errors in a license file
Troubleshoot problems serving licenses	lmdiag	Performs problem diagnosis	Used anytime to help determine why a license cannot be checked out
Obtain license key from vendor	lmhostid	Reports the hostid of a system	Used anytime to determine the host code that must be provided to vendors when obtaining a software license
Recover inaccessible licenses	lmremove	Removes a single user's license for a specified feature	Used when a client node crashes in order to recover a checked out license not automatically freed.
Determine version compatibility between the license server and an application	lmver	Reports the FLEXlm version of a library of binary files.	Used anytime to determine what version of FLEXlm a FLEXlm-enabled product uses

### 4.3.6.1 Quick Start Using FLEXlm

Operators interact with FLEXlm via the license manager daemons and license files. FLEXlm's user interface is a set of Unix-like commands for starting, stopping, and requesting services from a license manager daemon. Command arguments specify input parameters, most notably the name of the license file whose contents determine the servers, daemons, and license provisions affected by the command. Operators install and maintain license files using any preferred editor.

#### 4.3.6.1.1 Command Line Interface

To start FLEXlm license server daemons in a consistent, predictable manner, execute the following startup script:

```
/etc/init.d/lmgrd start
```

Before it invokes FLEXlm's "lmgrd" program, the script adds the extension ".old" to the current FLEXlm log file (if any) so the new daemon will create its own. It then runs "lmgrd" as user "flexlm" to avoid running as "root", and it specifies the license and log file paths the daemons are to use (i.e., "/usr/local/flexlm/licenses/license.dat" and "/tmp/license\_log", respectively).

If license manager daemons are needed to serve licenses in additional license files, they can be started by running the "lmgrd" program as follow:

```
su flexlm -c /etc/opt/licenses/lmgrd.ste -c license_file -l logfile -2 -p & (SUNs only)
```

```
su flexlm -c /etc/opt/licenses/lmgrd -c license_file -l logfile -2 -p & (SGIs only)
```

To stop the FLEXlm license daemons that are running on all machines in the network, execute the FLEXlm command:

```
lmdown
```

However, to shut down the license manager daemons on a single machine only, log on to the machine and type the following command instead:

```
/etc/init.d/lmgrd stop
```

Table 4.3.6-2 summarizes commands available with FLEXlm. See Chapter 6 of the *FLEXlm End User Manual* for the complete description of each command and its arguments.

**Table 4.3.6-2. Command Line Interfaces (1 of 2)**

Command Line Interface	Description and Format	When and Why Used
lmcksum	lmcksum [-c <i>license_file</i> ]	To verify license file data
lmdiag	lmdiag [-c <i>license_file</i> ] \ [-n] [ <i>feature</i> ]	To diagnose problems when a license cannot be checked out
lmdown	lmdown [-c <i>license_file</i> ] [-q]	To shutdown all license daemons (both lmgrd and all vendor daemons) on all nodes.
lmgrd	lmgrd [ -app ] [ -c <i>license_file</i> ] \ [ -t <i>timeout_interval</i> ] [ -l <i>logfile</i> ] \ [ -s <i>timestamp_interval</i> ] [ -2 -p ] [ -v ] \ [ -x lmdown ] [ -x lmremove ]	To run the main daemon program for FLEXlm
lmhostid	lmhostid [-n]	To determine the hostid of a system
lminstall	lminstall [ -i { <i>infile</i>   -} ] [ -o <i>outfile</i> ] \ [ -overfmt {2   3   4   5   5.1   6} ] \ [ -odecimal ]	To convert licenses between decimal and readable formats and between different versions of FLEXlm formats.
lmremove	lmremove [ -c <i>file</i> ] <i>feature user host</i> \ <i>display</i>  <u>or</u>  lmremove [ -c <i>file</i> ] -h <i>feature host</i> \ <i>port handle</i>	To remove a single user's license for a specified feature. (This is only needed when a client node crashes, since that's the only condition where a license is not automatically freed. If the application is active, it checkouts the license again after it is freed by lmremove.)
lmreread	lmreread [-c <i>license_file</i> ] \ [-vendor <i>name</i> ]	To cause the license daemon to reread the license file and start any new vendor daemons that have been added. In addition, one or all pre-existing daemons are signaled to reread the license file for changes in feature licensing information.
lmswitchr	lmswitchr [ -c <i>license file</i> ] <i>feature</i> \ <i>new-file</i>  <u>or</u>  lmswitchr [ -c <i>license file</i> ] <i>vendor</i> \ <i>new-file</i> (v5.0+ onl)	To start recording license events in a new or different log file for the FLEXadmin tool

**Table 4.3.6-2. Command Line Interfaces (2 of 2)**

Command Line Interface	Description and Format	When and Why Used
lmstat	lmstat [-a] [-A] [-c <i>license_file</i> ] \ [-f <i>feature</i> ] [-i [ <i>feature</i> ]] \ [-S <i>vendor</i> ] [-s <i>hostname</i> ] \ [-t <i>value</i> ]	To report the status of all network licensing activities
lmver	lmver <i>filename</i>	To identify the FLEXlm version of a library or binary file.

#### 4.3.6.2 FLEXlm Main Screen

FLEXlm does not provide for operator interaction via a GUI. All interactions are through the Unix command line or a Unix script.

#### 4.3.6.3 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 FLEXlm, refer to the ECS Baseline Information System web page, URL <http://cmdm.east.hitc.com/>.

##### 4.3.6.3.1 Interfaces and Data Types

The ECS fault management tool, Tivoli, monitors FLEXlm's event logs, error logs, and debug file in order to notify operators when activity of interest has occurred. Tivoli interrogates the log file frequently, searching new messages for text strings that match pre-determined criteria. When one is found, the operator is notified via Tivoli's operator event console. Table 4.3.6-3 lists FLEXlm's interfaces for Version 2.0. (Tivoli is discussed in sections 4.2.2, 4.3.5, and 4.4.6.)

**Table 4.3.6-3. Interface Protocols**

Interface	Type of Primary Interface Protocols	Comments
Tivoli	ASCII Log file	Tivoli's log file adapter monitors the FLEXlm debug file in order to notify operators when interesting licensing events occur.

#### 4.3.6.4 Databases

FLEXlm uses license and options files in lieu of a database. License files are independent text files, each of which contains all the site-specific information FLEXlm needs to serve the licenses specified in the file. Every license manager daemon requires a license file, and different license files require separate license manager daemons. To simplify operations, operators may combine

license files obtained from multiple vendors if they are compatible. Refer to chapter 2 of the *FLEXlm End User Manual* for information about the format of a license file, and refer to chapter 3 about when and how to combine them.

Options files are text files associated with specific vendor daemons named in license files. These files allow the operator to specify criteria for granting licenses to users, wait time before reclaiming inactive licenses, and how much license usage information is to be logged. FLEXlm does not require an options file. When specified however, there can only be one options file per vendor daemon, and each vendor needs a separate options file. See chapter 5 of the *FLEXlm End User Manual* for details.

#### **4.3.6.5 Special Constraints**

FLEXlm cannot be run without one or more license files, and most FLEXlm commands require the name of a license file in order to execute. License files identify the host and port number a client is to use to communicate with the license server. If the license file parameter is missing from the command, FLEXlm tries using the file(s) named in the environment variable `LM_LICENSE_FILE`. If `LM_LICENSE_FILE` is not set, the default license file name `/usr/local/flexlm/licenses/license.dat` is assumed.

The *FLEXlm End User Manual* recommends the following operating constraints :

- Keep a copy or link of the license file in the vendor's "default" location; some vendors expect to find their license files at pre-determined locations. Refer to Section 1.1.1 of the *FLEXlm End User Manual*.
- Run `lmgrd` as a non-privileged user (not *root*) to avoid security risks. Refer to Section 6.4 of the *FLEXlm End User Manual*

Also, re-use the old pathname when creating a new debug file, if possible. Tivoli monitors the file in order to notify operators when significant licensing events occur. Changing the pathname requires a corresponding change in the configuration of Tivoli's log file adapter.

#### **4.3.6.6 Outputs**

FLEXlm's principal outputs are inter-process communications with COTS applications attempting to check out and check in FLEXlm licenses, but these are generally transparent to the operator. Outputs visible to the operator include an ASCII log of network licensing events and errors, and messages constituting responses to operator-entered commands.

#### **4.3.6.7 Event and Error Messages**

FLEXlm writes both status and error messages to standard output. Typically, operators redirect all output from the startup command "`lmgrd`" to a file, known as the debug file, to create a FLEXlm log at the site.

Appendix D of the *FLEXlm End User Manual* lists what causes the more common messages an operator may encounter, but primarily those written by the FLEXlm programs. Event and error

messages logged by FLEXlm-enabled COTS applications are sometimes found in the application's manuals. Messages are typically self-explanatory and identify the date/time of the event, the license server host, the product or feature involved, and the name of the user.

#### 4.3.6.8 Reports

FLEXlm's *lmstat* utility can generate the status reports listed in Table 4.3.6-4. Each is written to standard output and may be redirected to a named file or a printer using standard Unix conventions.

**Table 4.3.6-4. Reports**

Report Type	Report Description	When and Why Used
lmstat -s	Lists status of clients running on a named host	As required
lmstat -i	Lists license information about all or a named feature	As required
lmstat -a	Lists all information about current network licensing activities	As required
lmstat -A	Lists all currently active licenses	As required
lmstat -f	Lists users of all or a named feature	As required
lmstat -S	Lists users of all or a named vendor's features	As required

**Note:** FLEXlm documentation refers to a report log and a set of license administration reports associated with a companion product, FLEXadmin. FLEXadmin is not part of the ECS deployment. *lmstat* does not use the “report log” and does not produce FLEXadmin reports.

##### 4.3.6.8.1 Sample Reports

The figures that follow contain sample FLEXlm status reports. One sample is provided for each report listed in Table 4.3.6-4.

```
lmstat - Copyright (C) 1989-1998 Globetrotter Software, Inc.
Flexible License Manager status on Fri 4/9/1999 10:54

License server status: 1744@jupi,1744@intrepid,1744@enterprise
License file(s) on jupi: /usr/local/flexlm/licenses/license.dat:

jupi: license server UP v6.1
intrepid: license server UP (MASTER) v6.1
enterprise: license server UP v6.1

Vendor daemon status (on intrepid):

xnidaem: UP v6.0
IDE: UP v4.1
rational: UP v6.0
setechd: UP v6.1
idl_lmgrd: UP v6.1
ICSBX: UP v3.0
ncdlmd: UP v4.1
dmccabe: UP v5.11
suntechd: UP v4.1
cayenne: UP v5.12
```

**Figure 4.3.6-1. All Clients (lmstat -s) Report**



lmstat - Copyright (C) 1989-1998 Globetrotter Software, Inc.  
Flexible License Manager status on Fri 4/9/1999 10:55

NOTE: lmstat -i does not give information from the server,  
but only reads the license file. For this reason,  
lmstat -a is recommended instead.

Feature	Version	#	licenses	Expires	Vendor
xni	1.0	6		1-jan-0xnidaem	
stpcore	2.000	5	6	31-dec-00	IDE
omt	3.000	5	0	31-dec-00	IDE
bach	2.000	3		31-dec-00	IDE
imtbl	2.000	3		31-dec-00	IDE
crud	2.000	3		31-dec-00	IDE
sqlgen	2.000	3		31-dec-00	IDE
se	6.000	3		31-dec-00	IDE
se_reveng	6.000	1		31-dec-00	IDE
se_cbrowse	6.000	1		31-dec-00	IDE
rose.ada.unix	5.000	1	0	01-jan-00	rational
rose.c++.unix	5.000	1	0	01-jan-00	rational
rose.java.unix	5.000	1	0	01-jan-00	rational
rose.modeler.unix	5.000	1	0	01-jan-00	rational
ada.sun4	2.000	3	0	01-jan-00	rational
apex.sun4.self_target.c	2.000	3	0	01-jan-00	rational
asis	2.000	3	0	01-jan-00	rational
env_base.sun4	2.000	3	0	01-jan-00	rational
rose.cpp.sun4	4.000	5	0	01-jan-00	rational
soda	2.000	5	0	01-jan-00	rational
summit_base.sun4	2.000	3	0	01-jan-00	rational
summit_cm.sun4	2.000	3	0	01-jan-00	rational
summit_tm.sun4	2.000	3	0	01-jan-00	rational
ddts	4.1		10000	31-oct-2002	rational
APEX_CPPT.sun4	2.000	3	0	01-jan-00	rational
Admin	7.000	6		1-jan-0setechd	
Modularity	7.000	3	0	1-jan-0setechd	
			.		
			.		
			.		
Extract	7.000	2		1-jan-0setechd	
DISCOVER	7.000	3	0	1-jan-0setechd	
idl	5.200		200	1-jan-0000	idl_lmgrd
insight	2.000	2	0	1-jan-0000	idl_lmgrd
BuilderXcessory	3.000	1	7	01-jan-00	ICSBX
BuilderXcessory	5.000	4		01-jan-00	ICSBX
DatabaseXcessory	1.000	5		01-jan-00	ICSBX
Adobe-DPS-Extension	1.000		100	1-jan-0ncdlmd	
mwm	1.000		100	1-jan-0ncdlmd	
GLX	1.000	2	0	1-jan-0ncdlmd	
X3D-PEX	1.000	2	0	1-jan-0ncdlmd	
XIE	1.000	2	0	1-jan-0ncdlmd	
mvt	6.000	6		31-dec-2008	dmccabe
procompiler.c	3.000	1		01-jan-0	suntechd
procompiler.c	4.200	1		01-jan-0	suntechd
			.		
			.		
			.		

**Figure 4.3.6-2. License Information (lmstat -i) Report**

```

lmstat - Copyright (C) 1989-1998 Globetrotter Software, Inc.
Flexible License Manager status on Fri 4/9/1999 10:57

License server status: 1744@jupi,1744@intrepid,1744@enterprise
  License file(s) on jupi: /usr/local/flexlm/licenses/license.dat:

    jupi: license server UP v6.1
    intrepid: license server UP (MASTER) v6.1
    enterprise: license server UP v6.1

Vendor daemon status (on intrepid):

  xnidaem: UP v6.0
  IDE: UP v4.1
  rational: UP v6.0
  setechd: UP v6.1
  idl_lmgrd: UP v6.1
  ICSBX: UP v3.0
  ncdlmd: UP v4.1
  dmccabe: UP v5.11
  suntechd: UP v4.1
  cayenne: UP v5.12

Feature usage info:

Users of xni:  (Total of 6 licenses available)

Users of stpcore:  (Total of 56 licenses available)

Users of omt:  (Total of 50 licenses available)

Users of bach:  (Total of 3 licenses available)
.
.
.
Users of ddts:  (Total of 10000 licenses available)

"ddts" v4.1, vendor: rational
floating license

  dwashing yakusaka /dev/pts/8 (v4.1) (jupi/1744 4001), start Mon 3/29 10:18 (linger: 1800)
  chunter borg /dev/pts/12 (v4.1) (jupi/1744 4606), start Tue 3/30 8:55 (linger: 1800)
  michelle stargazer /dev/pts/25 (v4.1) (jupi/1744 2013), start Mon 4/5 11:19 (linger:
1800)
  twicker pongo.hitc.com /dev/tty (v4.1) (jupi/1744 6231), start Mon 4/5 14:55 (linger:
1800)
.
.
.

```

**Figure 4.3.6-3. All Activities (lmstat -a) Report**

lmstat - Copyright (C) 1989-1998 Globetrotter Software, Inc.□

```
Flexible License Manager status on Fri 4/9/1999 10:57

License server status: 1744@jupi,1744@intrepid,1744@enterprise
  License file(s) on jupi: /usr/local/flexlm/licenses/license.dat:

    jupi: license server UP v6.1
    intrepid: license server UP (MASTER) v6.1
    enterprise: license server UP v6.1

Vendor daemon status (on intrepid):

  xnidaem: UP v6.0
    IDE: UP v4.1
  rational: UP v6.0
  setechd: UP v6.1
  idl_lmgrd: UP v6.1
    ICSBX: UP v3.0
  ncdlmd: UP v4.1
  dmccabe: UP v5.11
  suntechd: UP v4.1
  cayenne: UP v5.12

Feature usage info:

Users of xni:  (Total of 6 licenses available)

Users of stpcore:  (Total of 56 licenses available)
.
.
.
Users of rose.modeler.unix:  (Total of 10 licenses available)

"rose.modeler.unix" v5.000, vendor: rational
floating license

  perabell sydney ncdso23:0.0 (v5.0) (jupi/1744 1419), start Fri 4/9 10:12
  nchan sydney ncdp102:0.0 (v5.0) (jupi/1744 5838), start Fri 4/9 10:38

Users of ada.sun4:  (Total of 30 licenses available)

Users of apex.sun4.self_target.c++:  (Total of 30 licenses available)
.
.
.
Users of ddts:  (Total of 10000 licenses available)

"ddts" v4.1, vendor: rational
floating license

  dwashing yakusaka /dev/pts/8 (v4.1) (jupi/1744 4001), start Mon 3/29 10:18 (linger: 1800)
  chunter borg /dev/pts/12 (v4.1) (jupi/1744 4606), start Tue 3/30 8:55 (linger: 1800)
  michelle stargazer /dev/pts/25 (v4.1) (jupi/1744 2013), start Mon 4/5 11:19 (linger:
1800)
  twicker pongo.hitc.com /dev/tty (v4.1) (jupi/1744 6231), start Mon 4/5 14:55 (linger:
1800)
.
.
.
Users of Y2K:  (Total of 30 licenses available)

"Y2K" v7.000, vendor: setechd
floating license

  bhough deimos /dev/pts/3 (v7.0) (jupi/1744 429), start Fri 4/9 9:53
  yyang deimos /dev/pts/4 (v7.0) (jupi/1744 104), start Fri 4/9 10:17

Users of Metrics:  (Total of 30 licenses available)
.
.
.
```

**Figure 4.3.6-4. All Active Licenses (lmstat -A) Report**

```

lmstat - Copyright (C) 1989-1998 Globetrotter Software, Inc.
Flexible License Manager status on Fri 4/9/1999 11:02

Users of ddts:  (Total of 10000 licenses available)

"ddts" v4.1, vendor: rational
floating license

  dwashing yakusaka /dev/pts/8 (v4.1) (jupi/1744 4001), start Mon 3/29 10:18 (linger: 1800)
  chunter borg /dev/pts/12 (v4.1) (jupi/1744 4606), start Tue 3/30 8:55 (linger: 1800)
  michelle stargazer /dev/pts/25 (v4.1) (jupi/1744 2013), start Mon 4/5 11:19 (linger:
1800)
  twicker pongo.hitc.com /dev/tty (v4.1) (jupi/1744 6231), start Mon 4/5 14:55 (linger:
1800)
.
.
.

```

**Figure 4.3.6-5. Users of Named Feature (lmstat -f) Report**

```

lmstat - Copyright (C) 1989-1997 Globetrotter Software, Inc.
Flexible License Manager status on Fri 4/9/1999 11:49

DAEMONS in configuration file: xnidaem IDE rational setechd idl_lmgrd ICSBX ncdlmd dmccabe
suntechd cayenne
Users of features served by setechd:
Users of Admin:  (Total of 6 licenses available)

Users of Modularity:  (Total of 30 licenses available)

Users of Delta:  (Total of 30 licenses available)

Users of Package:  (Total of 30 licenses available)

Users of AutoDoc:  (Total of 2 licenses available)

Users of Dormant:  (Total of 1 licenses available)

Users of Y2K:  (Total of 30 licenses available)

"Y2K" v7.000, vendor: setechd
floating license

bhough deimos /dev/pts/3 (v7.0) (jupi/1744 429), start Fri 4/9 9:53
yyang deimos /dev/pts/4 (v7.0) (jupi/1744 104), start Fri 4/9 10:17

Users of Metrics:  (Total of 30 licenses available)

Users of C_CPP:  (Total of 30 licenses available)
.
.
.
Users of DISCOVER:  (Total of 30 licenses available)

"DISCOVER" v7.000, vendor: setechd
floating license

bhough deimos /dev/pts/3 (v7.0) (jupi/1744 1349), start Fri 4/9 9:53
yyang deimos /dev/pts/4 (v7.0) (jupi/1744 1236), start Fri 4/9 10:17
.
.
.

```

**Figure 4.3.6-6. Users of Named Vendor's Features (lmstat-S) Report**

### 4.3.7 iFOR/LS

iFOR/LS is a commercially available network license management product that helps the ECS M&O staff at the DAACs, EOC, and SMC administer licenses and enforce licensing provisions for iFOR/LS-enabled COTS software at the site. It enforces licensing provisions based on information from vendor-provided license keys and lets license administrators control and distribute licenses and permissions across multiple license servers. iFOR/LS handles numerous types of licenses, including floating (concurrent use), nodelocked, site, and metered licenses.

iFOR/LS consists of one or more license servers and the client software embedded in iFOR/LS-enabled applications. Each license server is a daemon (netlsd) that manages data about licenses belonging to that server. It grants licenses to applications that request them, if one is available and the requester is authorized. It also maintains records of network licensing events, some of which are copied, by an ECS-developed script, to a log monitored by the COTS product Tivoli so that operators can be notified when interesting events occur.

The license servers manage only “server-based” licenses. Server-based licenses allow products to be run on multiple nodes. Licenses for products run solely on a single node are called “nodelocked” and are maintained manually in an iFOR/LS file on the node. iFOR/LS license servers do not maintain or report information about nodelocked licenses or license activity.

iFOR/LS uses location broker services of the Network Computing System (NCS), which must be installed for iFOR/LS to work. Location brokers employ two types of daemons, global (glbd) and local (llbd), to manage replicatable databases about distributed services available on the network. iFOR/LS clients use the location broker to identify supporting license server(s), then request from the servers the licenses they need in order to run.

iFOR/LS license servers can be configured to run on multiple hosts, but licenses must be divided amongst the servers. While this achieves a measure of redundancy, licenses registered with a failed server are not available for use.

Table 4.3.7-1 summarizes the operator functions that iFOR/LS supports.

**Table 4.3.7-1. Common ECS Operator Functions Performed with iFOR/LS (1 of 2)**

Operating Function	User Interface	Description	When and Why to Use
Start license manager (section 4.3.7.1.1.1)	i4lmd	Starts the iFOR/LS license server daemon program.	Used to initiate license management server processes
Stop license manager (section 4.3.7.1.1.1)	n/a <sup>1</sup>		
Installing new licenses (section 4.3.7..2)	i4admin	Displays and edits the license server database	Used anytime to put add or alter provisions for a license.
Deleting installed licenses (section 4.3.7..2)	i4admin	Displays and edits the license server database	Used anytime to remove a license from the system.
Audit installed licenses (section 4.3.7..2)	i4stat	Displays currently installed licenses	Used anytime to check details about iFOR/LS licenses installed on network servers.
Monitor network licensing activities (section 4.3.7..2)	i4stat	Provides various status information about licenses (other than nodelocked) and servers	Used anytime to check on licenses installed, licenses in use, and current users of licenses.
Review history of network licensing activities (section 4.3.7.1.1.1)	i4rpt	Generates various reports about license server events, and includes a facility for scrubbing the event log	Used anytime to review logged licensing events.
Troubleshoot problems serving licenses (section 4.3.7.1.1.1)	i4tv	Verifies that network license servers are functioning correctly	Used anytime on nodes containing a licensed product or license server to verify what iFOR/LS servers are active and that each can serve a license

<sup>1</sup> iFOR/LS does not provide operators a mechanism for shutting down a license server directly. However, HP OpenView provides a command, OVLNetLSDown, which stops the local iFOR/LS license server user to manage HP OpenView licenses. See the HP OpenView, Network Node Manager Products, Installation Guide (1995) for details.

**Table 4.3.7-1. Common ECS Operator Functions Performed with iFOR/LS (2 of 2)**

Operating Function	User Interface	Description	When and Why to Use
Obtain license password from vendor	i4target	Reports the unique identifier of a system	Used anytime to determine the target identifier that must be provided to vendors when obtaining a software license
Keep location broker databases synchronized	lb_admin  drm_admin	Registers NCS-based servers in location broker databases  Checks and maintains the network-wide location broker database.	Used anytime to inspect the contents of location broker databases and to correct database errors  Used anytime to administer the replication of NCS location broker databases
Maintain "user" files	Unix edit program "vi"	Edits "user" files on license server hosts	Used anytime to update the "user" file to record individual user privileges and priorities for licensed product

#### 4.3.7.1 Quick Start Using iFOR/LS

iFor/LS consists of several independent programs and a collection of daemons. The tool is not integrated and each of the programs must be started independently, as required. The collection of daemons is normally started at system boot-time, but the **i4ldm** program provides the ability to initiate the daemons, if necessary, during normal operations.

Certain activities such as removing or recovering an iFOR/LS database, changing the server license, or replacing a LAN card require shutting down the license manager. iFOR/LS does not provide a special command for doing this, but certain iFOR/LS applications such as HP OpenView do<sup>2</sup>. In the absence or failure of a shutdown command, send SIGTERM (kill -15) to the i4lmd process that is executing.

<sup>2</sup> The HP OpenView command to stop the iFOR/LS license server is *OVLNetLSDown*. See the HP OpenView Network Node Manager Products Installation Guide.

iFOR/LS provides a command line interface for all functions and includes two screens, **i4admin** and **i4stat**<sup>3</sup>, for real-time and historical reporting, license creation, and administration. The **i4admin** screen facilitates adding, editing, and removing vendor products and their license provisions from a license server database. The **i4stat** screen presents status information on product licenses. Table 4.3.7-2 in section 4.3.7-3 below summarize iFOR/LS commands available to an operator.

The following documents describe how to use iFOR/LS:

- *iFOR/LS Quick Start Guide*<sup>4</sup>
- *iFOR/LS Quick Start Guide, Hewlett-Packard Version*<sup>5</sup>
- *iFOR/LS Administrator's Guide*<sup>6</sup>
- *iFOR/LS Installation Notes*
- *HP OpenView, Network Node Manager Products, Installation Guide, 1995*
- *Network License System Administration Guide (Silicon Graphics Computer Systems)*
- *AIX Version 4.1 iFOR/LS System Management Guide*
- *AIX Version 4.1 iFOR/LS Tips and Techniques*

Consult the iFOR/LS Administrator's Guide in particular. Its command reference section describes the format and use of iFOR/LS commands and configuration files as well as relevant NCS commands and files.

#### 4.3.7.1.1 Invoking iFOR/LS From the Command Line Interface

**Note:** The license server for a node<sup>7</sup> starts automatically at boot time if the startup file contains the line, **START\_I4LMD=1**. On HP-UX 10.x systems, the startup file is at /etc/rc.config.d/i4lmd. If it is necessary to restart the daemons, execute the iFOR/LS program from the command line prompt using:

**i4lmd**

The i4lmd program has no user interface. The operator may verify the server has started either by running the **i4tv** utility (see below) or by typing at a Unix prompt:

---

<sup>3</sup> Names for iFOR/LS programs for the HP-UX 10.x operating system have an "i4" prefix. Functionally equivalent versions of the same programs built for other operating systems have an "ls\_" prefix. Most iFOR/LS documents use the "ls\_" names, but ECS runs iFOR/LS under HP-UX 10.x.

<sup>4</sup> Version 2 and 3 of the Guide is available via the Gradient Technologies home page on the World Wide Web.

<sup>5</sup> This version is distributed with the iFOR/LS software and is usually found at /opt/ifor/ls/conf/i4qsguide on nodes on which iFOR/LS software has been installed.

<sup>6</sup> Version 2 and 3 of the Guide is available via the Gradient Technologies home page on the World Wide Web.

<sup>7</sup> Throughout their documentation, Gradient uses the term "node" to refer to a network-attached computer or machine.



**ps -ed | grep i4lmd**

Refer to the Command Reference section of the iFOR/LS Administrator's Guide for details on the **i4lmd** utility.

The iFOR/LS command line interface performs all license management functions supported by the **i4admin** and **i4stat** screens plus other functions that support iFOR/LS operation. Tables 4.3.7-2 and 4.3.7-3 present the iFOR/LS and NCS commands operators are likely to use, summarizing information contained in the iFOR/LS Quick Start Guide and the command reference section of the iFOR/LS Administrator's Guide. Certain NCS commands that operate interactively (i.e., process subcommands entered at the command line in response to a prompt) are noted in the appropriate table. See the Guide for the complete description of each command and its arguments.

Operator interaction with iFOR/LS to install or delete licenses is carried out with the **i4admin** screen. To execute this program from the command line use:

**i4admin**

Note: Vendors often distribute license passwords electronically and in a form ready for use as an argument with the i4admin command. Use the command line interface in such cases. The license can be installed with a single command, and electronically copying information reduces the chances of error.

Operator interaction with iFOR/LS to check detailson licenses is carried out with the i4stat screen. To execute this program from the command line use:

**i4stat**

Operator interaction with iFOR/LS to report on licenses is carried out with the i4rpt screen. To execute this program from the command line use:

**i4rpt**

Operator interaction with iFOR/LS to verify network license servers are functioning is carried out with the i4tv screen. To execute this program from the command line use:

**i4tv**

**Table 4.3.7-2. Command Line Interfaces – iFOR/LS**

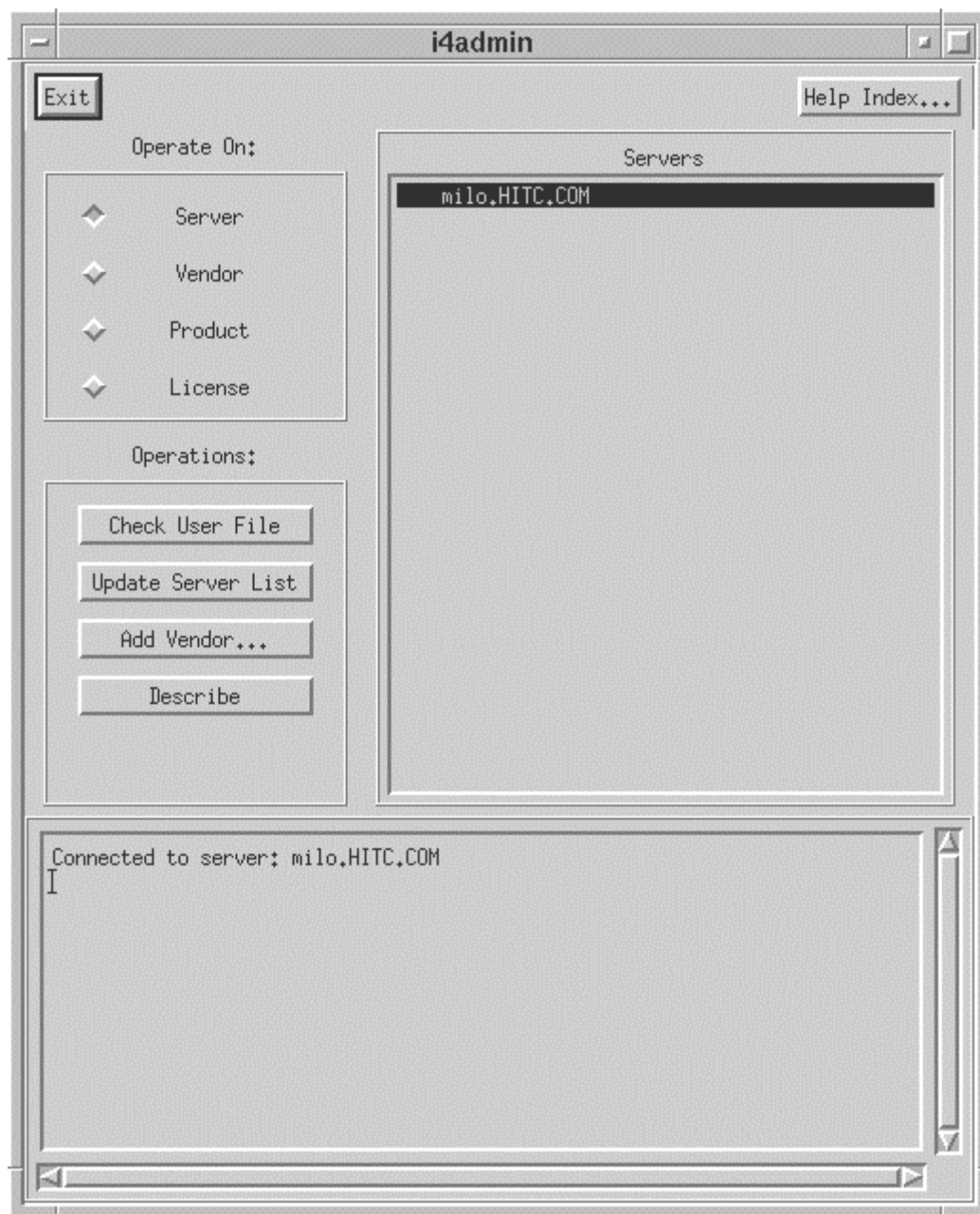
Command Line Interface	Description and Format	When and Why Used
i4admin	i4admin [-n <i>node_name</i> ] [-r] [-l   -z   -h   -usage   -version ] {-a   -s   -d   -f } {-v   -p} <i>argument1 argument2...</i>	To verify license file data
i4stat	i4stat {-t   -i   -a   -u <i>user_name</i> } [[-n <i>server</i> ] [-v <i>vendor</i> ][ -p <i>product</i> [ -r <i>version</i> ] ] [ -z ]   [ -h   -usage   -version]	To diagnose problems when a license cannot be checked out
i4report	i4report [[ -n <i>node_name</i> ] [ -c ] [ -z ] [ <i>event_type_list</i> ] [ <i>information_filter_list</i> ]   [ -h   -usage   -version ]]	To shutdown all license daemons (both lmgrd and all vendor daemons) on all nodes.
i4lmd	i4lmd [[ -no <i>event_list</i>   -o   -v   -z ]]	To run the main daemon program for iFOR/LS
i4target	i4target [-O   -v   -h]	To determine the hostid of a system
i4tv	i4tv [ -n <i>hostname</i> ] [ -v ] [ -z ] [ -h   -usage   -version ]	To verify the license servers are running properly
i4config	i4config	To verify that NCS is properly set up and ensure iFOR/LS runs
i4first	i4first	To start the location brokers and license server. This script, created by <i>i4config</i> , is run at most once each time <i>i4config</i> is run

**Table 4.3.7-3. Command Line Interfaces – NCS**

Command Line Interface	Description and Format	When and Why Used
drm_admin	drm_admin [ -version ]	Used xxxx to manage replication of NCS location broker databases. Subcommands issued via standard input specify actions such as add, delete, merge, and list replicas of the global location broker
lb_admin	lb_admin [ -nq ] [ -version ]	Used any time to register NCS-based servers with location brokers. Subcommands issued via standard input specify actions associated with adding, correcting, and removing information in local and global location broker databases
lb_find	lb_find [ -q ] [ -v ] [ -dl ]	Used to identify the location, port, and type of global location brokers running on a network
glbd	glbd [ -create { -first <i>family_name</i>   -from <i>host_name</i> } ] glbd [ -change_family <i>family_name</i> ] [ -listen <i>family_list</i> ] [ -version ]	Used to start a replicatable global location broker daemon on a node
llbd	llbd [ -listen <i>family_list</i> ] [ -version ]	Used to start a local location broker daemon on a node
nrglbd	nrglbd [ -version ]	Used to start a non-replicatable version of a global location broker daemon on a node

#### 4.3.7.2 iFOR/LS Main Screen

The iFOR/LS GUI consists of two independent screens. There isn't a single "Main screen" to select other functions on. These two screens, **i4admin** and **i4stat**, are described below.



**Figure 4.3.7-1. i4admin screen**

Controls on the **i4admin** screen are:

**Exit** to quit **i4admin**

**Help Index** provides a list of Help subjects for the screen.

Other controls on the screen are grouped by purpose:

Operate On: allows the operator to select from a list of the selected objects located to the right of the group.

**Server** generates a list of the registered servers.

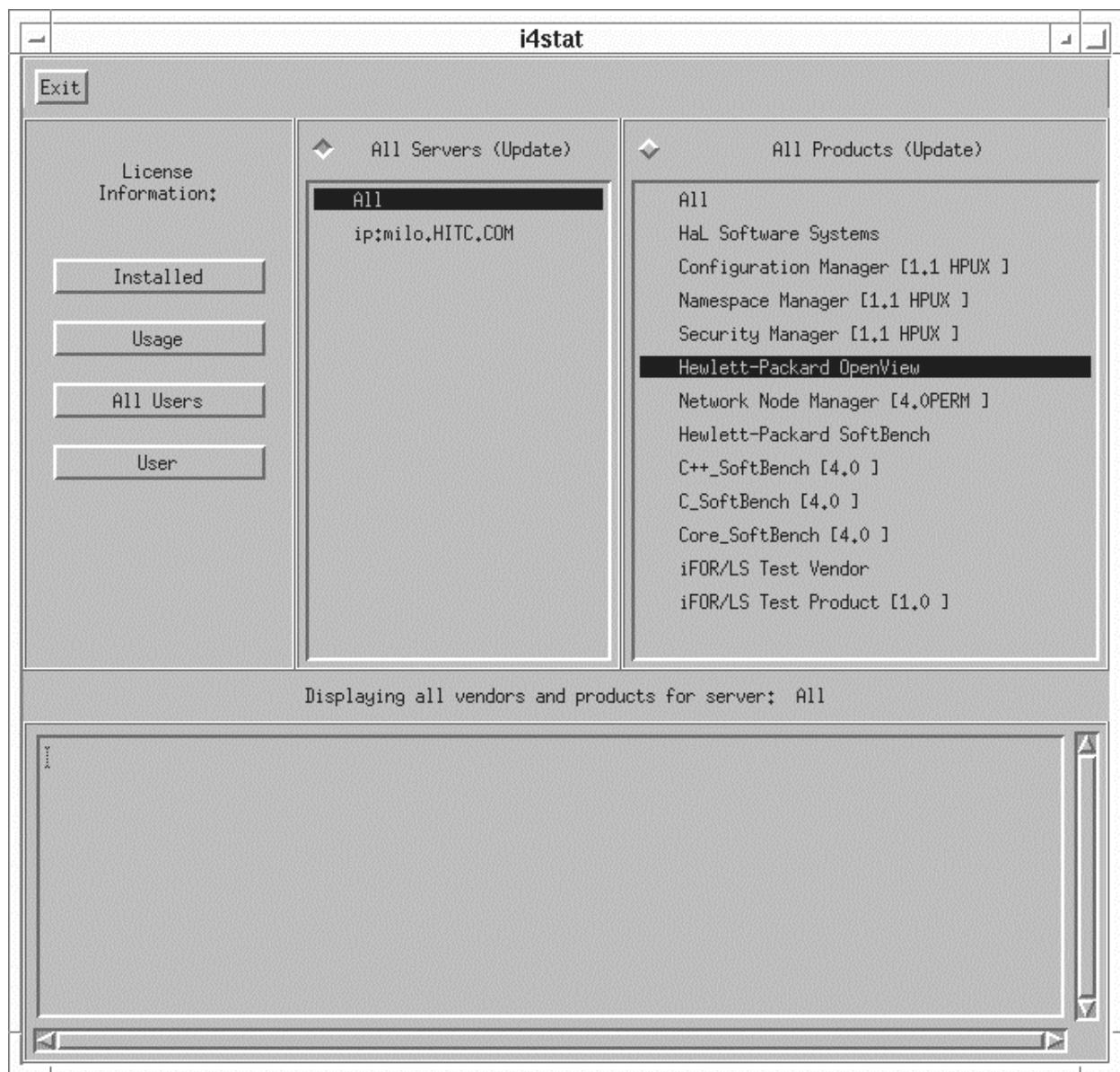
**Vendor** generates a list of registered vendors.

**Product** generates a list of registered products.

**License** generates a list of registered licenses.

Operations: allows the operator to choose the action to perform with the selection from the Operate On group.

The choices available in this group depend on the Operate On selection. If more data than is currently displayed on the screen is required a pop-up will be invoked to solicit the additional data or provide for input.



**Figure 4.3.7-2. i4stat screen**

Controls on the **i4stat** screen are:

**Exit** to quit **i4stat**

License Information: allows the operator to filter from the lists of the selected objects located to the right of the group.

**Installed** generates a list of the installed licenses for the listed servers and products.

**Usage** generates a list of the license usage for the listed servers and products

**All Users** generates a list of the users for the listed servers and products.

**User** generates a list of licenses used by the selected user.

The output available in this group depend on the License Information selection. If more data than is currently displayed on the screen is required a pop-up will be invoked to solicit the additional data or provide for input. The list generated appears at the bottom of the screen.

**All Servers:** displays all the registered servers - used to select servers for the lists to be generated.

**All Products:** displays all the registered products - used to select products for the lists to be generated.

See Chapter 3 (“*Displaying Information and Generating Reports*”) and the Command Reference sections of the iFOR/LS Administrator’s Guide for more details about the **i4stat** display, where it is discussed under its alternate name “**ls\_stat**”.

### 4.3.7.3 Required Operating Environment

iFOR/LS is a UNIX-based application that operates under many environments, including HP-UX, Sun Solaris, and IRIX. It relies on the naming or locating services of the Network Computing System in order for clients to find and use the license servers. At least one node in the network must run a global location broker, and any node that runs either a license server daemon or local location broker must run a local location broker as well. The DCE daemon, **rpcd**, serves as the local location broker for ECS.

The target ECS operating environment for iFOR/LS license servers is HP-UX 10.x. Table 4.3.7-4 lists the system directories in which key iFOR/LS-related programs and data files can be found on HP-UX 10.x platforms.

**Table 4.3.7-4. iFOR/LS Directories**

Directory	Contents
/opt/ifor/ls/bin	iFOR/LS executables
/opt/ifor/ls/conf	iFOR/LS configuration files
/var/opt/ifor	iFOR/LS database, log, user, and nodelock files
/opt/ifor/conf	configuration and startup/shutdown scripts
/usr/sbin/ncs	NCS executables
/etc/ncs	NCS configuration files
/var/ncs	NCS global location broker databases and log file

iFOR/LS-enabled applications using server-based licenses cannot run unless a license manager is active on some network node accessible to the application. A license manager can be started only if iFOR/LS software has been installed on the node and a location broker configured for iFOR/LS exists and is running. (The NCS **/b\_admin** and **lb\_find** commands can be used to determine if local and global location broker daemons are running, respectively. See the iFOR/LS Administrator's Guide for details).

#### 4.3.7.3.1 Interfaces and Data Types

The ECS fault management tool, Tivoli, monitors iFOR/LS' licensing and error events in order to notify operators when activity of interest has occurred. The ECS script **MsLiiFORLSMkDayLog** periodically extracts information about the day's events from the iFOR/LS database and stores it in a log file. Tivoli interrogates the log file frequently, searching for text strings that match pre-determined criteria. When one is found, the operator is notified via Tivoli's operator event console.

Table 4.3.7-6 lists the iFOR/LS messages that trigger operator notifications by the initially deployed Tivoli configuration. See section 4.2.3 for details about Tivoli's configuration.

**Table 4.3.7-5. Interface Protocols**

Interface	Type of Primary Interface Protocols	Comments
<b>MsLiiFORLSMkDayLog</b>	iFOR/LS command line utility, <i>i4report</i>	MsLiiFORLSMkDayLog is an ECS customization script that extracts data about licensing events from the iFOR/LS event database for storage in an ASCII event log. Tivoli's log file adapter monitors the log in order to notify operators when interesting licensing events occur.

**Table 4.3.7-6. iFOR/LS Messages That Trigger Operator Notifications via Tivoli (1 of 2)**

iFOR/LS Message
License not found in database (network license server/server)
iFOR/LS license not found (network license server/server)
Past expiration date on license (network license server/server)
Before start date on iFOR/LS license (network license server/server)
Version not found in database (network license server/server)
iFOR/LS license version not found in database (network license server/server)
Not enough licenses (network license server/server)
Not enough iFOR/LS licenses (network license server/server)
Bad I/O (network license server/server)
No licenses available (network license server/server)



**Table 4.3.7-6. iFOR/LS Messages That Trigger Operator Notifications via Tivoli (2 of 2)**

iFOR/LS Message
No iFOR/LS licenses available (network license server/server)
Time disparity too large (network license server/server)
Database not valid for this server (network license server/server)
Database corrupt (network license server/server)
Fatal error-check error log (network license server/server)
Vendor key is incorrect (network license server)
iFOR/LS server not found (network license server/server)
Wrong version of iFOR/LS Server (network license server/server)
iFOR/LS log file contains invalid data (network license server/server)
No matching socket families found (network license server/library)
No valid server handle exists, cannot log message (network license server/library)
Internal error (network license server/tools)
license will expire
communications failure (network computing system/RPC runtime)

#### 4.3.7.4 Databases

iFOR/LS maintains an internal database containing all information about vendors, products, and licenses. The database is stored in a file named `lic_db` in the `/var/opt/ifor` directory. Operators have no interaction with its schema, parameters, or macros.

#### 4.3.7.5 Special Constraints

iFOR/LS manuals recommend the following operating constraints :

- The Network Computing System must be installed in order to run iFOR/LS.
- System clocks on license server and location broker nodes must be synchronized to within two minutes of each other.
- The interactive iFOR/LS shell script *i4config* and the *i4first* companion it creates must be run to configure and start the location brokers and license manager whenever setting up a license server node. They should be rerun when an iFOR/LS server is not working and NCS communications errors are received. See the iFOR/LS Quick Start Guide and the iFOR/LS Administrator's Guide for details.

### 4.3.7.6 Outputs

iFOR/LS' principal outputs are inter-process communications with COTS applications attempting to check out and check in licenses, but these are generally transparent to the operator. Other outputs visible to the operator include reports discussed in section 4.3.7.7 below and messages constituting responses to operator-entered commands.

### 4.3.7.7 Event and Error Messages

iFOR/LS writes all event and error messages to an internal database stored in file "log\_file" in directory /var/opt/ifor on the license server node. Retrieve information about license management events and errors via the **i4report** utility.

The iFOR/LS System Management Guide lists some of the more common messages an operator may encounter. Messages are typically self-explanatory, identifying the date/time of the event, the license server host, the product or feature involved, and the name of the user.

### 4.3.7.8 Reports

iFOR/LS' **i4report** utility can generate the log reports listed in Table 4.3.7-8. Each is written to standard output and may be redirected to a named file or a printer using standard Unix conventions. Most reports have the same format, but vary in the events they include.

#### 4.3.7.8.1 Sample Reports

The figures that follow show sample iFOR/LS reports. One sample is provided for each report listed in Table 4.3.7-7.

**Table 4.3.7-7. Reports**

Report Type	Report Description	When and Why Used
i4report -a	Lists all logged events and messages	As required
i4report -d	Lists all license database modifications	As required
i4report -e	Lists all error events	As required
i4report -f	Lists any fatal error events	As required
i4report -l	Lists all events related to handling or processing of licenses	As required
i4report -m	Lists all messages that were logged by a software product or a license server	As required
i4report -r1	Lists the number of requests for licenses, the number of licenses granted, and the percent of rejected requests for each vendor product	As required
i4report -r2	Lists, by user, the number of requests for licenses, the number of licenses granted, and the percent of rejected license requests	As required
i4report -s	Lists all server start/stop events	As required

```

i4report Version 3.0.0 HP-UX
(c) Copyright 1991,1992,1993, Hewlett-Packard Company, All Rights Reserved
(c) Copyright 1991,1992,1993,1994,1995 Gradient Technologies Inc., All Rights Reserved

Network License Server Log Report (3.0.0 HP-UX)
For: milo.HITC.COM

Index      Vendor      Product              Vrsn      User      Node              Group      Amt Time
Event
-----
1          Hewlett-Packard OpenView (700) Network No 4.0PERM root      et3sv3.HITC.COM root
1 11/26/96 18:43:20 C H E C K E D
2          Hewlett-Packard OpenView (700) Network No 4.0PERM root      relbhpms.HITC.COM root
1 11/26/96 18:44:29 C H E C K E D
11483      Hewlett-Packard OpenView (700) Network No 4.0PERM root      relbhpms.HITC.COM
root 1 12/16/96 16:16:58 C H E C K E D
11484      Hewlett-Packard OpenView (700) Network No 4.0PERM root      et3sv3.HITC.COM root
1 12/16/96 16:19:25 C H E C K E D
11485      iFOR/LS Test Vendor      *** message: 12/16/96 16:21:19 user file not found
(/var/opt/ifor/user_file) - anyone can use product: iFOR/LS Test Product
11486      iFOR/LS Test Vendor (4) iFOR/LS Test 1.0      root      relbhpms.HITC.COM sys
1 12/16/96 16:21:19 G R A N T E D on mach type HP
11487->11486 iFOR/LS Test Vendor (4) iFOR/LS Test 1.0      root      relbhpms.HITC.COM sys
1 12/16/96 16:21:19 R E L E A S E D. time of use = 00:00
11488      Hewlett-Packard OpenView (700) Network No 4.0PERM root      relbhpms.HITC.COM
root 1 12/16/96 16:21:58 C H E C K E D
33879      Hewlett-Packard OpenView (700) Network No 4.0PERM root      relbhpms.HITC.COM
root 1 01/20/97 08:13:35 C H E C K E D
33904      Hewlett-Packard OpenView (700) Network No 4.0PERM root      et3sv3.HITC.COM root
1 01/20/97 08:54:32 C H E C K E D
33905      Hewlett-Packard OpenView (700) Network No 4.0PERM root      slimer.HITC.COM root
1 01/20/97 08:57:13 C H E C K E D
33906      Hewlett-Packard OpenView (700) Network No 4.0PERM root      relbhpms.HITC.COM
root 1 01/20/97 08:58:35 C H E C K E D
33907      Hewlett-Packard OpenView      *** message: 01/20/97 08:59:11 user file not found
(/var/opt/ifor/user_file) - anyone can use product: Network Node Manager
33908->31636 Hewlett-Packard OpenView (700) Network No 4.0PERM root      rush.gsfc.nasa.gov
root 1 01/20/97 08:59:11 license timed out. time of use = 2 days, 14:04:28
33909      Hewlett-Packard OpenView (700) Network No 4.0PERM root      rush.gsfc.nasa.gov
root 1 01/20/97 08:59:11 G R A N T E D on mach type HP
33910      Hewlett-Packard OpenView (700) Network No 4.0PERM root      et3sv3.HITC.COM root
1 01/20/97 08:59:32 C H E C K E D
33911      Hewlett-Packard OpenView (700) Network No 4.0PERM root      slimer.HITC.COM root
1 01/20/97 09:02:13 C H E C K E D
34105      Hewlett-Packard OpenView      *** message: 01/20/97 13:11:12 NetLS: License not
found in database (network license server/server) [ld010001]
34106      Hewlett-Packard OpenView (1400)      V4.60TEMP root      rush.gsfc.nasa.gov
root 1 01/20/97 13:11:12 License not found in database (network license
server/server).
98929      Hewlett-Packard OpenView (700) Network No 4.0PERM root      ****
1 04/19/97 07:48:56 C H E C K E D
98930      Hewlett-Packard OpenView (700) Network No 4.0PERM root      ****
1 04/19/97 07:48:57 C H E C K E D
100435     Hewlett-Packard OpenView (700) Network No 4.0PERM root      ****
**** 1 04/21/97 01:39:27 C H E C K E D
100842     Hewlett-Packard OpenView (700) Network No 4.0PERM root      ****
**** 1 04/21/97 12:57:16 C H E C K E D

finished.

```

**Figure 4.3.7-3. Events and Messages (i4report -a) Report**

```

i4report  Version 3.0.0 HP-UX
(c) Copyright 1991,1992,1993, Hewlett-Packard Company, All Rights Reserved
(c) Copyright 1991,1992,1993,1994,1995 Gradient Technologies Inc., All Rights Reserved

Network License Server Log Report (3.0.0 HP-UX)
For: milo.HITC.COM

Index      Vendor      Product      Vrsn      User      Node      Group      Amt Time
Event
-----
-----
1          Hewlett-Packard SoftBench
12/03/97 10:08:47 vendor added to license data base
2          Hewlett-Packard SoftBench (2600) Core_Soft 4.0
1 12/03/97 10:08:48 P R O D U C T A D D E D ConcurrentAccess (start date: 11/19/96, exp
date 12/31/93)
3          Hewlett-Packard SoftBench (3560) C_SoftBen 4.0
1 12/03/97 10:08:48 P R O D U C T A D D E D ConcurrentAccess (start date: 11/19/96, exp
date 12/31/93)
4          Hewlett-Packard OpenView
12/03/97 10:24:58 vendor added to license data base
5          Hewlett-Packard OpenView (700) Network No 4.0TEMP
64000 12/03/97 10:25:16 P R O D U C T A D D E D ConcurrentAccess (start date: 06/13/97, exp
date 10/11/97)
6          Hewlett-Packard SoftBench (2600) Core_Soft 4.0
1 12/03/97 10:30:40 P R O D U C T A D D E D ConcurrentAccess (start date: 11/19/96, exp
date 12/31/93)
7          Hewlett-Packard SoftBench (3560) C_SoftBen 4.0
1 12/03/97 10:31:00 P R O D U C T A D D E D ConcurrentAccess (start date: 11/19/96, exp
date 12/31/93)
8          Hewlett-Packard SoftBench (2617) C++_SoftB 4.0
1 12/03/97 10:31:10 P R O D U C T A D D E D ConcurrentAccess (start date: 11/19/96, exp
date 12/31/93)
9          Hewlett-Packard OpenView (700) Network No 4.0TEMP
64000 12/03/97 16:00:00 P R O D U C T A D D E D ConcurrentAccess (start date: 11/18/97, exp
date 03/18/98)

finished.

```

**Figure 4.3.7-4. License Database Modifications (i4report -d) Report**

```

i4report  Version 3.0.0 HP-UX
(c) Copyright 1991,1992,1993, Hewlett-Packard Company, All Rights Reserved
(c) Copyright 1991,1992,1993,1994,1995 Gradient Technologies Inc., All Rights Reserved

Network License Server Log Report (3.0.0 HP-UX)
For: milo.HITC.COM

Index      Vendor      Product      Vrsn      User      Node      Group      Amt Time
Event
-----
-----
1          Hewlett-Packard OpenView (700)
1 12/04/97 18:09:46 Version not found in database (network license server/server). sys
2          Hewlett-Packard OpenView (700)
1 12/04/97 18:09:47 Version not found in database (network license server/server). sys
3          Hewlett-Packard OpenView (700)
1 12/04/97 18:09:47 License not found in database (network license server/server). sys
4          Hewlett-Packard OpenView (700)
1 12/05/97 10:06:42 Version not found in database (network license server/server). sys
5          Hewlett-Packard OpenView (700)
1 12/05/97 10:06:42 Version not found in database (network license server/server). sys
6          Hewlett-Packard OpenView (700)
1 12/05/97 10:06:42 License not found in database (network license server/server). sys
7          Hewlett-Packard OpenView (700)
1 12/05/97 10:51:16 Version not found in database (network license server/server). sys
8          Hewlett-Packard OpenView (700)
1 12/05/97 10:51:16 Version not found in database (network license server/server). sys
9          Hewlett-Packard OpenView (700)
1 12/05/97 10:51:16 License not found in database (network license server/server). sys

finished.

```

**Figure 4.3.7-5. Error Events (i4report -e) Report**

```
i4report Version 3.0.0 HP-UX
(c) Copyright 1991,1992,1993, Hewlett-Packard Company, All Rights Reserved
(c) Copyright 1991,1992,1993,1994,1995 Gradient Technologies Inc., All Rights Reserved

N e t w o r k   L i c e n s e   S e r v e r   L o g   R e p o r t   (3.0.0 HP-UX)
For: milo.HITC.COM

Index      Vendor      Product          Vrsn      User      Node          Group      Amt Time
Event
-----
-----

finished.
```

**Figure 4.3.7-6. Fatal Error Events (i4report -f) Report**

```

i4report  Version 3.0.0 HP-UX
(c) Copyright 1991,1992,1993, Hewlett-Packard Company, All Rights Reserved
(c) Copyright 1991,1992,1993,1994,1995 Gradient Technologies Inc., All Rights Reserved

Network License Server Log Report (3.0.0 HP-UX)
For: milo.HITC.COM

Index      Vendor      Product              Vrsn      User      Node              Group      Amt Time
Event
-----
-----
1          iFOR/LS Test Vendor (4) iFOR/LS Test 1.0      root      milo.HITC.COM      sys        1
12/03/97 10:05:30 G R A N T E D on mach type HP
2          iFOR/LS Test Vendor (4) iFOR/LS Test 1.0      root      milo.HITC.COM      sys        1
12/03/97 10:05:30 R E L E A S E D. time of use = 00:00
3          Hewlett-Packard OpenView (700) Network No 4.0TEMP root      ****          ****
1 12/04/97 18:09:47 G R A N T E D on mach type HP
4          Hewlett-Packard OpenView (700) Network No 4.0TEMP root      ****          ****
1 12/04/97 18:14:47 C H E C K E D
(700) Network No 4.0TEMP root      ****          ****      1 12/05/97 08:04:57 C H E C
K E D

194        Hewlett-Packard OpenView (700) Network No 4.0TEMP root      ****          ****
1 12/05/97 10:04:59 C H E C K E D
195 ->3      Hewlett-Packard OpenView (700) Network No 4.0TEMP root      ****          ****
1 12/05/97 10:06:17 R E L E A S E D. time of use = 15:56:30
196        Hewlett-Packard OpenView (700) Network No 4.0TEMP root      ****          ****
1 12/05/97 10:06:42 G R A N T E D on mach type HP
197        Hewlett-Packard OpenView (700) Network No 4.0TEMP root      ****          ****
1 12/05/97 10:11:42 C H E C K E D

204        Hewlett-Packard OpenView (700) Network No 4.0TEMP root      ****          ****
1 12/05/97 10:46:42 C H E C K E D
205        Hewlett-Packard OpenView (700) Network No 4.0TEMP root      ****          ****
1 12/05/97 10:51:16 M U L T I P L E G R A N T on mach type HP
206        Hewlett-Packard OpenView (700) Network No 4.0TEMP root      ****          ****
1 12/05/97 10:51:42 C H E C K E D
207        Hewlett-Packard OpenView (700) Network No 4.0TEMP root      ****          ****
1 12/05/97 10:56:17 C H E C K E D
208        Hewlett-Packard OpenView (700) Network No 4.0TEMP root      ****          ****
1 12/05/97 10:56:42 C H E C K E D
209        Hewlett-Packard SoftBench (2600) Core_Soft 4.0      rwagle      ****          ****
1 12/05/97 10:56:46 G R A N T E D on mach type HP
210        Hewlett-Packard SoftBench (3560) C_SoftBen 4.0      rwagle      ****          ****
1 12/05/97 10:56:46 G R A N T E D on mach type HP
211 ->209    Hewlett-Packard SoftBench (2600) Core_Soft 4.0      rwagle      ****          ****
1 12/05/97 10:58:18 R E L E A S E D. time of use = 01:32
212 ->210    Hewlett-Packard SoftBench (3560) C_SoftBen 4.0      rwagle      ****          ****
1 12/05/97 10:58:18 R E L E A S E D. time of use = 01:32

finished.

```

**Figure 4.3.7-7. License-related Events (i4report-l) Report**

```

i4report  Version 3.0.0 HP-UX
(c) Copyright 1991,1992,1993, Hewlett-Packard Company, All Rights Reserved
(c) Copyright 1991,1992,1993,1994,1995 Gradient Technologies Inc., All Rights Reserved

Network License Server Log Report (3.0.0 HP-UX)
For: milo.HITC.COM

Index      Vendor      Product          Vrsn      User      Node          Group      Amt Time
Event
-----
1          iFOR/LS Test Vendor  *** message: 12/03/97 10:05:30 user file not found
(/var/opt/ifor/user_file) - anyone can use product: iFOR/LS Test Product
2          Hewlett-Packard OpenView *** message: 12/04/97 18:09:47 user file not found
(/var/opt/ifor/user_file) - anyone can use product: Network Node Manager
3          Hewlett-Packard OpenView *** message: 12/04/97 18:09:47 NetLS: Version not
found in database (network license server/server) [ld010007]
4          Hewlett-Packard OpenView *** message: 12/05/97 10:06:42 user file not found
(/var/opt/ifor/user_file) - anyone can use product: Network Node Manager
5          Hewlett-Packard OpenView *** message: 12/05/97 10:06:42 NetLS: Version not
found in database (network license server/server) [ld010007]
6          Hewlett-Packard OpenView *** message: 12/05/97 10:51:16 user file not found
(/var/opt/ifor/user_file) - anyone can use product: Network Node Manager
7          Hewlett-Packard OpenView *** message: 12/05/97 10:51:16 NetLS: Version not
found in database (network license server/server) [ld010007]
8          Hewlett-Packard SoftBench *** message: 12/05/97 10:56:46 user file not found
(/var/opt/ifor/user_file) - anyone can use product: Core_SoftBench
9          Hewlett-Packard SoftBench *** message: 12/05/97 10:59:02 user file not found
(/var/opt/ifor/user_file) - anyone can use product: Core_SoftBench
10         Hewlett-Packard SoftBench *** message: 12/05/97 11:05:37 user file not found
(/var/opt/ifor/user_file) - anyone can use product: Core_SoftBench
11         Hewlett-Packard SoftBench *** message: 12/05/97 11:09:28 user file not found
(/var/opt/ifor/user_file) - anyone can use product: Core_SoftBench
12         Hewlett-Packard SoftBench *** message: 12/05/97 11:09:54          ...adding
reference
13         Hewlett-Packard SoftBench *** message: 12/05/97 11:10:57          ...adding
reference
14         Hewlett-Packard SoftBench *** message: 12/05/97 11:11:31
...removing reference
15         Hewlett-Packard SoftBench *** message: 12/05/97 11:12:00          ...adding
reference
16         Hewlett-Packard SoftBench *** message: 12/05/97 11:12:00 user file not found
(/var/opt/ifor/user_file) - anyone can use product: C_SoftBench
17         Hewlett-Packard SoftBench *** message: 12/05/97 11:12:47
...removing reference
18         Hewlett-Packard SoftBench *** message: 12/05/97 11:13:36          ...adding
reference
19         Hewlett-Packard SoftBench *** message: 12/05/97 11:13:36 user file not found
(/var/opt/ifor/user_file) - anyone can use product: C_SoftBench
20         Hewlett-Packard SoftBench *** message: 12/05/97 11:14:07
...removing reference
21         Hewlett-Packard SoftBench *** message: 12/05/97 11:32:35          ...adding
reference
22         Hewlett-Packard SoftBench *** message: 12/05/97 11:32:44
...removing reference
23         Hewlett-Packard SoftBench *** message: 12/05/97 11:33:22          ...adding
reference

finished.

```

**Figure 4.3.7-8. Messages (i4report -m) Report**



```
i4report Version 3.0.0 HP-UX
(c) Copyright 1991,1992,1993, Hewlett-Packard Company, All Rights Reserved
(c) Copyright 1991,1992,1993,1994,1995 Gradient Technologies Inc., All Rights Reserved
```

```
Network License Server Usage Report (3.0.0 HP-UX)
For: milo.HITC.COM
```

Vendor	Product	Vrsn	Num of Req	Num Granted	% Rejections
Hewlett-Pack	C++_SoftBenc	4.0	1	1	0%
Hewlett-Pack	C_SoftBench	4.0	7	7	0%
Hewlett-Pack	Core_SoftBen	4.0	5	5	0%
Hewlett-Pack	Network Node	4.0TEMP	2	2	0%
iFOR/LS Test	iFOR/LS Test	1.0	1	1	0%

finished.

**Figure 4.3.7-9. Product Requests (i4report -r1) Report**

```
i4report Version 3.0.0 HP-UX
(c) Copyright 1991,1992,1993, Hewlett-Packard Company, All Rights Reserved
(c) Copyright 1991,1992,1993,1994,1995 Gradient Technologies Inc., All Rights Reserved
```

```
Network License Server User Usage Report (3.0.0 HP-UX)
For: milo.HITC.COM
```

User	Vendor	Product	Vrsn	Num of Req	Num Granted	% Rejections
rwagle	Hewlett-Pack	C++_SoftBenc	4.0	1	1	0%
rwagle	Hewlett-Pack	C_SoftBench	4.0	7	7	0%
rwagle	Hewlett-Pack	Core_SoftBen	4.0	5	5	0%
root	Hewlett-Pack	Network Node	4.0TEMP	2	2	0%
root	iFOR/LS Test	iFOR/LS Test	1.0	1	1	0%

Total Users : 2

finished.

**Figure 4.3.7-10. User Requests (i4report -r2) Report**

```

i4report  Version 3.0.0 HP-UX
(c) Copyright 1991,1992,1993, Hewlett-Packard Company, All Rights Reserved
(c) Copyright 1991,1992,1993,1994,1995 Gradient Technologies Inc., All Rights Reserved

N e t w o r k   L i c e n s e   S e r v e r   L o g   R e p o r t   (3.0.0 HP-UX)
For: milo.HITC.COM

Index      Vendor   Product          Vrsn    User    Node          Group    Amt Time
Event
-----
1          *** svr started                                     12/03/97
10:04:49

finished.

```

**Figure 4.3.7-11. Server Stop/Start (i4report -s) Report**

## **4.4 Security and Accountability**

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

1. DCE Cell Manager
2. TCP Wrapper
3. Crack
4. SATAN
5. TripWire
6. Tivoli (Enterprise Console and Admin)
7. Cryptographic Management Interface (CMI)

This page intentionally left blank.

#### 4.4.1 DCE Cell Manager

The DCE Cell Manager (also referred to as “Chisholm DCE Cell Manager 1.6.2”) toolset can administer various components and interactions of a DCE cell including:

- **Namespace Manager** - manages a DCE cell’s directory namespace
- **Configuration Manager** - manages a DCE cell’s Distributed Time Service (DTS) processes and configuration
- **Security Manager** - manages a DCE cell’s security registry

DCE Cell Manager is used to perform the operator functions listed in Table 4.4.1-1.

**Table 4.4.1-1. Common ECS Operator Functions Performed with DCE Cell Manager**

Operating Function	GUI	Description	When and Why to Use
Organizing Cell Directory Service (CDS) Namespace Directories	Namespace Manager	Manages a DCE cell’s directory namespace	When the need arises to manage, maintain, and monitor the cell namespace, keeping entries and access to them up to date
Configuring the Basic Cell	Configuration Manager	Manages a DCE cell’s configuration	When a DCE cell’s configuration needs managing
Synchronizing system clocks	Configuration Manager	Manages a DCE cell’s Distributed Service tasks	When system clocks need synchronizing
Populating the Security Registry	Security Manager	Manages a DCE cell’s security registry	When the need arises, to: <ul style="list-style-type: none"><li>• allow addition, modification, and deletion of principal accounts for new users</li><li>• create groups of principals to control access to registry and CDS objects</li><li>• set policies on passwords and accounts</li><li>• maintain replicas of the database that contains the registry</li></ul>

##### 4.4.1.1 Quick Start Using DCE Cell Manager

For more information on using DCE Cell Manager, see Section Starting and Using DCE Cell Manager, page 32 of the *Cell Manager 1.6. Overview and User’s Guide*.

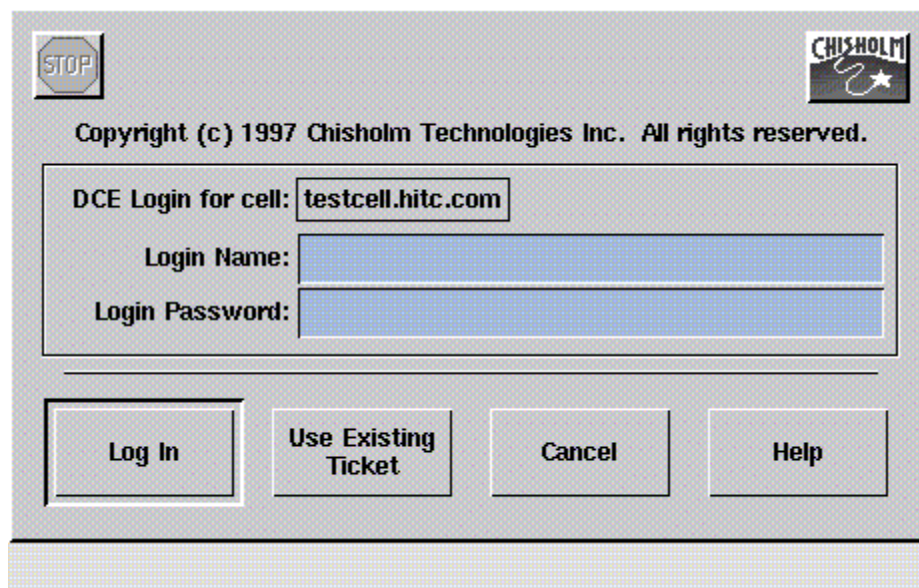
#### 4.4.1.1.1 Command Line Interface

To execute the DCE Cell Manager from the command line prompt type:

```
/opt/cellmgr/bin/cellManager
```

#### 4.4.1.2 DCE Cell Manager Main Screen GUIs

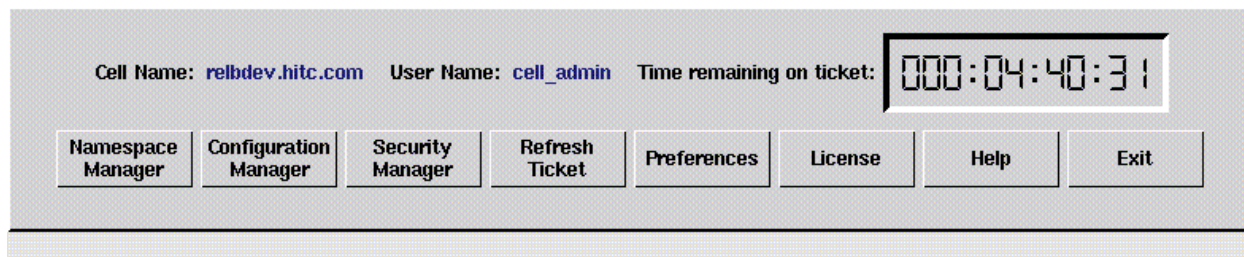
In order to use the DCE Cell Manager, the operator must first login to a DCE Cell (See Figure 4.4.1-1). For more information on starting the DCE Cell Manager , see section on Starting and Using DCE Cell Manager and Logging In, pages 32 through 34 of the *Cell Manager 1.6 Overview and User's Guide*.



**Figure 4.4.1-1. DCE Login Dialog**

#### 4.4.1.2.1 DCE Cell Manager Launcher

If the login is accepted, the DCE Cell Manager Launcher will appear as shown in Figure 4.4.1-2. If the login name or password is not accepted, a window will pop up with an appropriate error message and the operator may try again.



**Figure 4.4.1-2. DCE Cell Manager Launcher**

The DCE Cell Manager provides the following options:

- The **Namespace Manager** button brings up the Namespace Manager GUI. The initial display of this window is a tree structure with the cell root entries and a first level of directory entries. For more information, Cell Manager 1.6 Namespace Manager User's Guide.
- The **Configuration Manager** button brings up the Configuration Manager GUI which lists the hosts currently configured in the cell. For more information, Cell Manager 1.6 Configuration Managers User's Guide.
- The **Security Manager** button brings up the Security Manager GUI showing the organizations in the registry and represents the organization domain. For more information, Cell Manager 1.6 Security Manager User's Guide.
- The **Preferences** button brings up the DCE Launcher Preferences GUI. From this GUI additional buttons can be added to the Launcher window for applications commonly used in a DCE management session. Buttons on the DCE Cell Manager Launcher window can also be replaced or deleted. For more information, pages 36 through 39 of the Cell Manager 1.6 Overview and User's Guide.
- The **Refresh Ticket** button brings up a GUI where the DCE password must be entered. It is use to refresh the time remaining on the Authentication Ticket used by the DCE Cell Manager and which is indicated on the upper right box. For more information, see pages 35-36 of the Cell Manager 1.6 OverView and User's Guide.

- The **License** button brings up the Cell Manager License GUI. It is used to view, edit or load the License Agreement purchased. Not, the License can only be viewed or modified by the Super User or Systems Administrator. For more information, see pages 39 through 42 of the Cell Manager 1.6 Overview and User's Guide.
- The **Help** buttons brings up a display with concise information about specific windows, fields, buttons, and other application components.
- The **Exit** button provides options for exiting, including closing the DCE Cell Manager Launcher window.

#### **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 DCE Cell Manager, refer to the ECS Baseline Information System web page, URL

<http://cmdm.east.hitc.com/>.

**The document is 914-TDA-058, "DCE Cell Manager Installation and Configuration Report"**

#### **4.4.1.4 Databases**

None.

#### **4.4.1.5 Special Constraints**

None.

#### **4.4.1.6 Outputs**

The DCE Cell Manager provides the option of recording output from any DCE commands and commands executed during a current session of the tool and saving them to a history file. For more information, see Appendix B: Common Features of DCE Cell Manager Tools, *Cell Manager 1.6 Overview and User's Guide*.

#### **4.4.1.7 Event and Error Messages**

None.

#### **4.4.1.8 Reports**

None.



### 4.4.2 TCP Wrappers

TCP Wrappers allow the operator to control access to various network services through the use of access control list. They also provide logging information of wrapped network services which may 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 performs 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 intended to be 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 4.4.2-1.

**Table 4.4.2-1. Common ECS Operator Functions Performed with TCP Wrappers**

<b>Operating Function</b>	<b>Command/Action</b>	<b>Description</b>	<b>When and Why to Use</b>
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

#### **4.4.2.1 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.

##### **4.4.2.1.1 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.

##### **4.4.2.2 TCP Wrapper Main Screen**

TCP Wrapper does not have a graphical user interface.

##### **4.4.2.3 Required Operating Environment**

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

##### **4.4.2.4 Databases**

None.

##### **4.4.2.5 Special Constraints**

None.

#### 4.4.2.6 Outputs

Table 4.4.2-2 describes TCP Wrappers output.

**Table 4.4.2-2. 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 <i>local3.info</i> and the log file is <i>/etc/syslog.conf</i> .	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: <ul style="list-style-type: none"><li>• Date</li><li>• Time</li><li>• host</li><li>• Service</li><li>• Response to request</li><li>• Event</li></ul>

Figure 4.4.2.1 shows an example of log file created by TCP Wrappers. The log file may be examined with available tools like the **vi** editor. The contents of the log file may be used to generate reports on Service Request activity for the host.

```
Text Editor V3.5.1 [vulcan] - wrappers, dir; /var/log
File View Edit Find
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 4.4.2-1. 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.

#### **4.4.2.7 Event and Error Messages**

None.

#### **4.4.2.8 Reports**

None.

### 4.4.3 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 guessable passwords, and it should not be used in place of getting a really good, secure passwd program replacement.

Crack works by making many individual passes over the password entries that are 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 whitespace 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 that 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 4.4.3-1.

**Table 4.4.3-1. Common ECS Operator Functions Performed with Crack**

Operating Function	Command	Description	When and Why to Use
Ensure passwords used aren't easy to guess	<code>crack filename</code>	scans the password file and makes guesses of passwords	regularly review systems for weak passwords

#### 4.4.3.1 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 may be found in the ReadMe file described in Section 4.4.3.3 Required Operating Environment.

The documentation of Crack used as a basis and referenced in this section is for version 4.1.

#### **4.4.3.1.1 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.

#### **4.4.3.2 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 4.4.3-1 below.

```
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 4.4.3-1. Xterm window of Crack startup message and initialization**

#### 4.4.3.3 Required Operating Environment

Crack can run on any ECS 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 XRP Baseline Manager to determine where in ClearCase the ReadMe file resides

#### 4.4.3.4 Databases

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

#### 4.4.3.5 Special Constraints

None.

#### 4.4.3.6 Outputs

Table 4.4.3-2 describes the output for Crack.

**Table 4.4.3-2. Outputs**

Output	Description and Format
Report 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

#### 4.4.3.7 Event and Error Messages

None.

#### 4.4.3.8 Reports

Crack 4.1 does not support reports.



#### 4.4.4 SATAN

The Security Administrator Tool for Analyzing Networks (SATAN) tool is a COTS freeware product used for security compliance management. The tool is used to evaluate DAAC components for common security deficiencies and provide recommendations to enhance security.

SATAN gathers information about specified hosts and networks by examining network services (for example, finger, NFS, NIS, ftp, and rexd). It can then report this data in a summary format or, with a simple rule-based system, investigate potential security problems. Problems are described briefly and pointers provided to patches or workarounds. In addition to reporting vulnerabilities, SATAN gathers general network information (network topology, network services run, types of hardware and software being used on the network). SATAN has an exploratory mode that allows it to probe hosts that have not been explicitly specified. Thus, SATAN can probe not only targeted hosts, but also hosts outside your administrative domain.

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

**Table 4.4.4-1. Common ECS Operator Functions Performed with SATAN**

Operating Function	Command	Description	When and Why to Use
Check systems for security leaks/holes	<code>satan machinename</code>	Checks NFS export to unprivileged programs Checks NFS export via portmapper Checks Unrestricted NFS export Checks NIS password file access Checks rexd access Checks Sendmail vulnerabilities Checks TFTP file access Checks Remote shell access Checks Unrestricted X server access Checks Writable FTP home directory Checks wu-ftpd vulnerability	Whenever a security event has been detected or on a scheduled basis

##### 4.4.4.1 Quick Start Using SATAN

This section presents an orientation of SATAN. Additional information on SATAN can be obtained at the following URL:

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

The SATAN home page is shown in Figure 4.4.4-1 below.

The SATAN home page is also the GUI used by SATAN to communicate with the operator. When SATAN initializes, it starts its own local web server. Operators can then use Netscape (See

Netscape Navigator, Section 4.12.5) to interact with the program. The URL for SATAN is provided in the ReadMe file (See Required Operating Environment, Section 4.4.4.3 below).

Additional information regarding SATAN is available from the link on the home page and the SATAN help files.

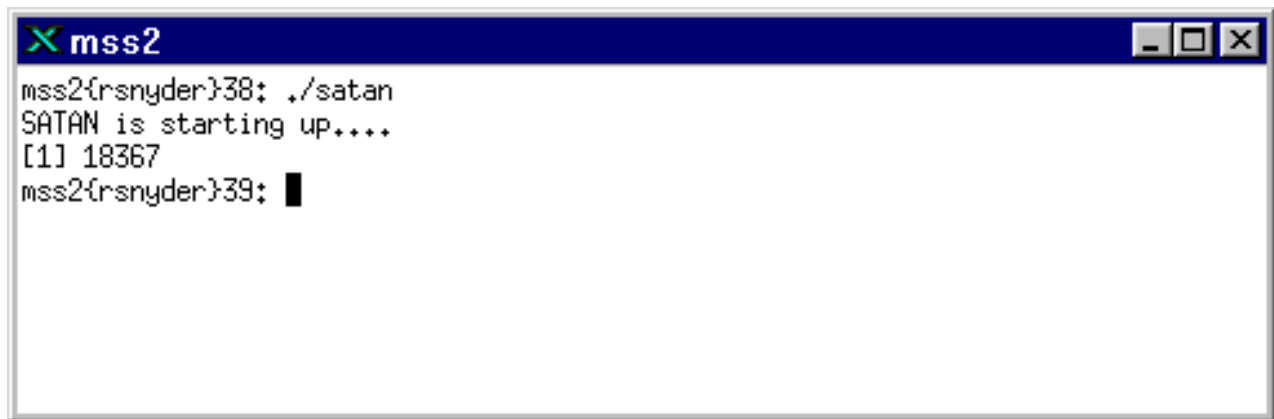
The documentation of SATAN used as a basis and referenced in this section is for version 1.1.1.

#### **4.4.4.1.1 Invoking SATAN From the Command Line Interface**

To execute SATAN from the command line prompt use:

```
satan
```

This command launches an Xterm window with the SATAN startup message shown in Figure 4.4.4-1 below.

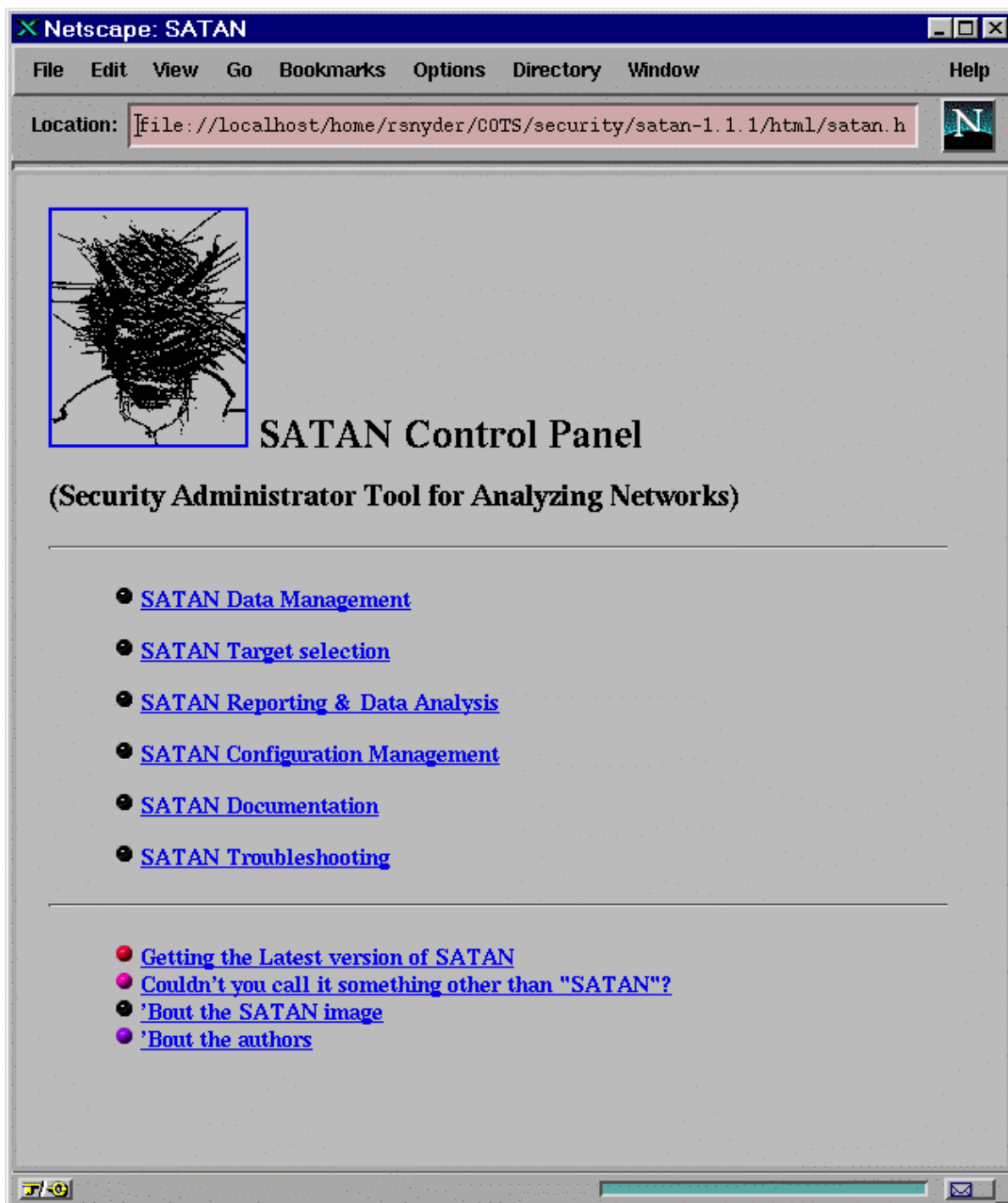


***Figure 4.4.4-1. Xterm window of SATAN showing startup message***

After SATAN starts, the operator may use Netscape Navigator to communicate with the program through ECS SATAN home page (see Figure 4.4.4-2 below).

#### **4.4.4.2 SATAN Main Screen**

After the SATAN startup message shown in Figure 4.4.4-1 is displayed the SATAN home page (Figure 4.4.4-2) is available from the local web server. The operator may access on-line help for all SATAN functions using the SATAN Documentation link.



*Figure 4.4.4-2. SATAN web home page*

#### **4.4.4.3 Required Operating Environment**

SATAN requires UNIX. SATAN can run on the following platforms: HP J210, Sun Sparc 20/50, and Sun Sparc 20/71. However, due to the potential for security breaches with uncontrolled use, it is implemented on a few select machines.

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 SATAN, refer to the ECS Baseline Information System web page, URL <http://cmdm.east.hitc.com/>.

#### **4.4.4.4 Databases**

The table of common deficiencies used by SATAN is proprietary. This table provides the specific tests SATAN uses to determine “deficiencies”. Modifications to this table are discussed in the on-line documentation available through the SATAN home page (Figure 4.4.4-2).

#### **4.4.4.5 Special Constraints**

The amount of memory that SATAN uses can be large, depending on how many hosts are being scanned in or are in the database, so it should be run at off-peak hours.

#### **4.4.4.6 Outputs**

SATAN outputs are available using the “SATAN Reporting & Data Analysis” selection on the SATAN home page (Figure 4.4.4-2).

#### **4.4.4.7 Event and Error Messages**

None.

#### **4.4.4.8 Reports**

See Section 4.4.4.6 above.

#### 4.4.5 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 very little 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, then the signature will also change. 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 that have 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 will be able to detect 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 4.4.5-1.

**Table 4.4.5-1. Common ECS Operator Functions Performed with SATAN**

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	<ul style="list-style-type: none"><li>• when another file needs to be monitored</li><li>• checks the integrity of the file system specified when the daemon is started</li></ul>
Compare file signatures with database	Done by Tripwire “chron” run periodically	Compares files current signatures against the data base and emails the operator a notification for changed files.	<ul style="list-style-type: none"><li>• This activity is a continuous, periodic performed on a configured interval by the “chron” run</li></ul>
Update the signatures datastore	Done manually in response to “Interactive Update” prompts	Updates the signature datastore when the email notification discloses legitimate changes	<ul style="list-style-type: none"><li>• As necessary to maintain a valid datastore of signatures.</li></ul>

#### 4.4.5.1 Quick Start Using Tripwire

This section presents an orientation of Tripwire. 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 1.2.

Tripwire is automatically invoked on all machines by a “chron” 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 in the Xterm.

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

##### 4.4.5.1.1 Invoking Tripwire From the Command Line Interface

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 #|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.

#### **4.4.5.2 Tripwire Main Screen**

Tripwire does not have a GUI. The Tripwire startup message is shown in Figure 4.4.5-1 below.

```

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 4.4.5-1. Xterm window with Tripwire showing Tripwire startup message.**

### 4.4.5.3 Required Operating Environment

Tripwire runs on all Sun, SGI, and HP 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 ECS Baseline Information System web page, URL <http://cmdm.east.hitc.com/>.



#### 4.4.5.4 Databases

Tripwire uses a datastore of captured information. Tripwire must be started from the command line interface to manually update this datastore with the information emailed to the operator.

#### 4.4.5.5 Special Constraints

None.

#### 4.4.5.6 Outputs

Tripwire generates the outputs presented in Table 4.4.5-2 below in the filename specified on the command line invocation. A sample of the generated report is shown in section 4.4.5.8.

**Table 4.4.5-2. 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 4.4.5.8, 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 that the Tripwire signature datastore is out-of-date. If the determination is that the datastore 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.

#### 4.4.5.7 Event and Error Messages

None.

#### 4.4.5.8 Reports

Tripwire must be started from the command line interface to request the Tripwire report. A sample of Tripwire output is listed below.

```

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.CPdcB    0cPX1H.DYS.s1vZdKD.ELMDR
snfru (sig2): 0PcgK/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 4.4.5-2. Tripwire Report**

#### 4.4.6 Tivoli Enterprise Console and Administration

This section describes how Tivoli Enterprise Console (TEC) and Tivoli/Admin are used by DAAC Operators to perform security functions. For additional information about Tivoli, see Section 4.2.2, “Tivoli” and Section 4.3.5 Tivoli/Courier.

The TEC is used to collect and monitor security events. Through the TEC the operator is able to monitor a users successful/unsuccessful login attempt, a successful/unsuccessful DCE login attempt, file access denial, number of unsuccessful login attempts, etc. TEC refers to the collection of capabilities that can be performed on managed objects. There is no specific graphic or visualization of “The Tivoli Enterprise Console”. For more information, see the *Tivoli/ Enterprise Console Event Adapter Guide* and the *Tivoli/Enterprise Console User’s Guide*.

The Tivoli Desktop is used to provide access and permissions to various applications. Tivoli provides the ability to create a unique desktop for each user. Thus enabling each user to have a unique set of available applications.

The Tivoli/Admin is used for the management/creation of UNIX accounts, UNIX Groups, and host mappings. The Tivoli/Admin GUI (the Tivoli Main Screen, Figure 4.4.6-2) provides the ability to create a DCE account and map the user to their assigned roles. For more information, see the *Tivoli User and Group Management Guide* and the *Tivoli Host Management Guide*.

The most frequently used Enterprise Console and Tivoli/Admin functions are listed in Table 4.4.6-1.

**Table 4.4.6-1. Common ECS Operator Functions Performed with Tivoli/Admin and TEC**

Operating Function	Command/Script or GUI	Description	When and Why to Use
Create Event Console	Series of GUIs	Create an event console	Use to allow the operator to view selected events that are received by the Event Server
Create user account	Series of GUIs	Create a definition of a new user	Use when there is a new user account to be created
Manage NIS maps	Series of GUIs	Distribute software files.	Use to distribute software to specified hosts.
Manage Unix host	Series of GUIs	Allows centralized administration of a Tivoli managed host	Use to check the status of a host or to modify certain parameters.
View Event	GUI	View an event displayed in the Event Console	To obtain more detailed information concerning an event in the Event Console

#### **4.4.6.1 Quick Start Using Tivoli/Admin**

This section presents an orientation of Tivoli/Admin. The underlying assumptions are that:

- Tivoli and Tivoli/Admin has been installed on all of the platforms that is involved in the administration process
- The user has been granted administrator privileges for Tivoli/Admin activity.

The documentation of Tivoli used as a basis and referenced in this section is for version/release 3.0.1, contained in the ECS Release 4.

##### **4.4.6.1.1 Invoking Tivoli/Admin and Tivoli Enterprise Console from the Command Line Interface**

Both Tivoli/Admin and Tivoli Event Console are started from the main Tivoli GUI. The icons for each specific application is described in detail in later sections of this document.

To execute Tivoli from the command line prompt use:

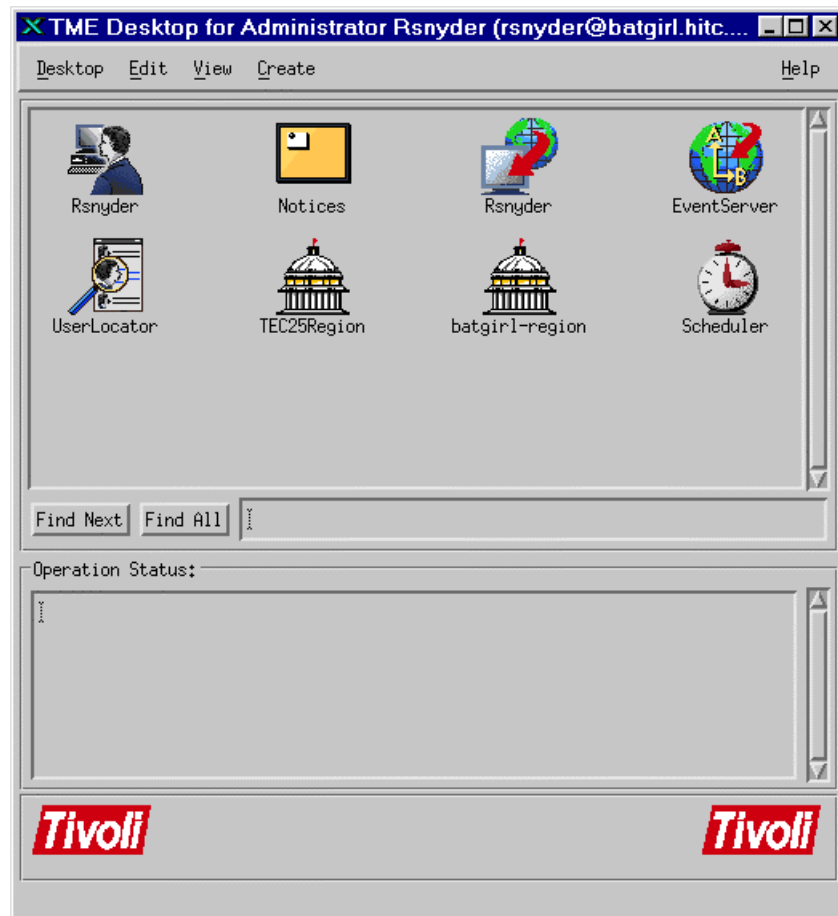
```
source /etc/setup_env.csh (in c shell)

./etc/Tivoli/setup_env.sh (in Bourne or bash shell)

tivoli -font fixed
```

#### **4.4.6.2 Tivoli Main Screen**

The Main Screen as shown in Figure 4.4.6-2 facilitates software distribution activities through a series of icons and dialogs. The top part of the screen holds the icons needed to initiate the activities, and the bottom part will display the type of process and the results of the process that is initiated.



**Figure 4.4.6-2. Tivoli Main Screen**

#### **4.4.6.3 Required Operating Environment**

Tivoli/Admin and Tivoli Event Console require a UNIX platform that serves as the Tivoli Managed Region Server and as the Tivoli Event Server. The Tivoli Managed Nodes can be UNIX platforms or PC's. ECS Tivoli managed nodes consists of Sun hosts (Solaris 2.5.1 OS), Hewlett Packard hosts (HP 10.0.1 OS), and SGI (SGI Irix 6.2) hosts.

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 ReadMe file for each product. To find the ReadMe file for Tivoli/Admin and Tivoli/Event Console, use the XRP Baseline Manager to determine where in ClearCase the ReadMe file resides.

#### **4.4.6.4 Databases**

Tivoli uses two databases, a Sybase database and a Tivoli proprietary database. Operators are unable to access any of the information that either of them contain.

#### **4.4.6.5 Special Constraints**

The operator must be a Tivoli Administrator to create the regions and associated components.

#### **4.4.6.6 Outputs**

Output from Tivoli is displayed on the screen in the form of its GUIs or a log entry (discussed in Section 4.4.6.7).

#### **4.4.6.7 Event and Error Messages**

All event and summary error messages are displayed in the Operation Status box within the main screen (Figure 4.4.6-3). Certain error messages may also appear on the Xterm console that was used to start Tivoli.

#### **4.4.6.8 Reports**

None.

#### 4.4.7 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 ECS 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 4.4.7-1.

**Table 4.4.7-1. Common ECS Operator Functions Performed with CMI**

Operating Function	Command / GUI	Description	When and Why to Use
Start CMI program	<i>EcSeAuthnProg</i>	This will bring 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 will cause the program to generate a randomized username and password	This is only needed when a new user account is required by an ECS server.

##### 4.4.7.1 Quick Start Using CMI

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

##### 4.4.7.1.1 Command Line Interface

To execute CMI from the command line prompt use:

> **EcSeAuthnProg**

Refer to the 920-TDx-013 "Custom Code Configuration Parameters" documentation series for a listing of **EcSeAuthnProg**. **There are no configuration parameters for EcSeAuthnProg!**

##### 4.4.7.2 CMI Main Screen

Figure 4.4.7-1 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 will return a user name and password which are displayed in the associated fields.

The image shows a window titled "ConnectAuth". Inside the window, there are three input fields. The first field is labeled "Enter Application Key :" and is a single-line text box. Below it, there are two more fields: "User Id" and "Password", each with a corresponding single-line text box. The background of the window has a light beige color with a subtle pattern of small blue dots.

**Figure 4.4.7-1. CMI Main Screen**

The Table 4.4.7-2 below describes all the fields found in the CMI Screen in Figure 4.4.7-1.

**Table 4.4.7-2. CMI Field Descriptions**

Field Name	Data Type	Size	Entry	Description
Application Key	Integer	1 to 10 digits	Required	Key that identifies 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.

#### 4.4.7.3 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.



For further information on CMI 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 GSFC DAAC. Note, there are no configurable or tunable parameters or environment variables of CMI.

#### **4.4.7.3.1 Interfaces and Data Types**

CMI utilizes no special data types or interfaces.

#### **4.4.7.4 Databases**

None.

#### **4.4.7.5 Special Constraints**

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

#### **4.4.7.6 Outputs**

All information is displayed on the CMI screen.

#### **4.4.7.7 Event and Error Messages**

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

#### **4.4.7.8 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 that are available to the specialist cover a broad spectrum. Some are COTS tools that fall 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 will have to 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 SSIT Manager

The SSIT 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 SCFs. The SSIT Manager application provides access to all COTS tools and custom applications that are part of the SSI&T environment. The SSIT 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 SSIT 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 SSIT Manager; all other programs run from the Manager are also accessible 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 that tells an executable where to find its various inputs and outputs, as well as the values for any specific runtime parameters. Different various of these files are used both by the SSIT 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 SSIT Manager GUI.

**Table 4.5.1-1. Common ECS Operator Functions Performed through the SSIT Manager GUI (1 of 3)**

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

**Table 4.5.1-1. Common ECS Operator Functions Performed through the SSIT 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"> <li>• open xterm window via Tools pull-down menu</li> <li>• also can be opened via Unix command <i>xterm</i></li> </ul>	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"> <li>• FORCHECK</li> <li>• Prohibited Function Checker</li> <li>• Process Control File (PCF) Checker</li> <li>• Prolog extractor</li> </ul>	<ul style="list-style-type: none"> <li>• check FORTRAN 77 science software</li> <li>• check if certain functions are used in the science software which conflict with the production environment</li> <li>• check the syntax of the data in the Process Control File</li> <li>• Extract prologs from science software</li> </ul>	<ul style="list-style-type: none"> <li>• to ensure that science code conforms to ECS standards</li> <li>• to ensure that the delivered PCF is of the proper syntax</li> <li>• To extract prologs from science software</li> </ul>
Product Examination	select the following via Tools:Product Examination pull-down menu: <ul style="list-style-type: none"> <li>• IDL</li> <li>• EOSView</li> </ul>	opened via Tools pull-down menu	<ul style="list-style-type: none"> <li>• Ad-hoc graphical analysis</li> <li>• For viewing an arbitrary file (e.g., standard product) in HDF format</li> </ul>
File Comparison	select the following via Tools: Product Examination File Comparison <ul style="list-style-type: none"> <li>• ASCII files</li> <li>• Binary files</li> <li>• HDF files (GUI)</li> <li>• HDF files (hdiff)</li> </ul>	compares the outputs of the science software between the DAAC and SCF	ensures that the output that was generated at the SCF when running the science software is the same output that is generated at the DAAC

**Table 4.5.1-1. Common ECS Operator Functions Performed through the SSIT 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"> <li>• Emacs</li> <li>• Xedit</li> </ul>	text editors	edit arbitrary text file
PDPS Database	select the following via Tools:PDPS Database pull-down menu: <ul style="list-style-type: none"> <li>• PCF ODL Template</li> <li>• Check ODL Files</li> <li>• SSIT Science Metadata Update</li> <li>• SSIT Operational Metadata Update GUI</li> </ul>	<ul style="list-style-type: none"> <li>• creates an ODL file template from the science software PCF</li> <li>• Check the ODL file updates PGE and ESDT SCIENCE metadata in the PDPS /SSIT database</li> <li>• updates PGE OPERATIONAL metadata via GUI in the PDPS /SSIT database</li> </ul>	To initialize and update the Planning/Production (PDPS) databases: <ul style="list-style-type: none"> <li>• SSIT version</li> <li>• Production version</li> </ul>
Data Server	select the following via Tools: Data Server pull-down menu: 1. Acquire DAP 2. Insert Static 3. Insert Test Dynamic 4. Insert EXE TAR 5. Edit SSAP 6. Get MCF	1. Acquires 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 from the Data Server.	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 SSIT Manager

The SSIT Manager provides a common interface to the SSI&T tools. An overview of the SSIT Manager GUI is provided in Section 4.5.1.2. A more detailed discussion of the tools accessed via this GUI is provided in subsequent sections.

The following assumptions are made with regard to the use of the SSIT Manager application.

- The operator is located at a workstation or server to which the SSIT Manager has been configured
- The operator has proper authorization to access the PDPS/SSIT database and the Data Server
- To access files in ClearCase, the operator has a ClearCase view already set
- The operator's environment has been configured as documented in the pertinent sections of the Help menu, available from the main window of the SSIT Manager (see Figure 4.5.1-1) 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 SSIT Manager at the Unix command line:

*DpAtMgr ConfigFile* <config\_filename> ecs\_mode <mode> &

where:

config\_filename is the name of the user's personal Process Framework configuration file for this program, as customized from **\$ECS\_HOME/CUSTOM/cfg/**.

ecs\_mode is the ECS mode of operation, e.g., OPS, TS2

#### **4.5.1.1.1 Invoking SSIT Manager From the Command Line Interface**

To start SSIT Manager at the Unix command line:

*DpAtMgr ConfigFile* <config\_filename> ecs\_mode <MODE> &

where config\_filename is the name of the user's personal Process Framework configuration file for this program, as customized from **\$ECS\_HOME/CUSTOM/cfg/**.

##### **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. Command Line Interfaces (Sun) (1 of 3)**

<b>Command Line Interface</b>	<b>Description and Format</b>	<b>When and Why Used</b>
EcDpAtMgr	startup of SSIT 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



**Table 4.5.1-2. Command Line Interfaces (Sun) (2 of 3)**

Command Line Interface	Description and Format	When and Why Used
sparcworks	Ad hoc code analysis	As needed
ghostview	Postscript file viewer	As needed
netscape	WWW browser Netscape	As needed
acoread	PDF file viewer Adobe Acrobat	As needed
DpAtMgrForcheck	FORTRAN 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
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
EcDpAt CreateODLtmplate	Create PGE metadata ODL template file	Before running EcDpAtDefinePGE
EcDpAtVerifyODL	Verify PGE metadata ODL template file	Before Running EcDpAtDefinePGE
EcDpAtDefinePGE	Update PDPS/SSIT database with SCIENCE metadata	Before executing PGE in SSIT environment
DpAtOpDbGui	Update PDPS/SSIT database with OPERATIONAL metadata	Before executing PGE in SSIT 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 SSIT or Production environment
DpAtInsertTest	Inserts test dynamic input file into the Data Server	Before executing PGE in SSIT environment

**Table 4.5.1-2. Command Line Interfaces (Sun) (3 of 3)**

Command Line Interface	Description and Format	When and Why Used
DpAtInsertExeTar	Inserts tar file of executables, etc. needed to run PGE file into the Data Server	Before executing PGE in SSIT 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 SSIT Manager. A small number of SSIT tools run only on the SGI platform. Because of security considerations, these tools cannot be run from the SSIT Manager on the Sun. They may only be run from the Unix command line on the SGI platform.

**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/SSIT Database Update GUI

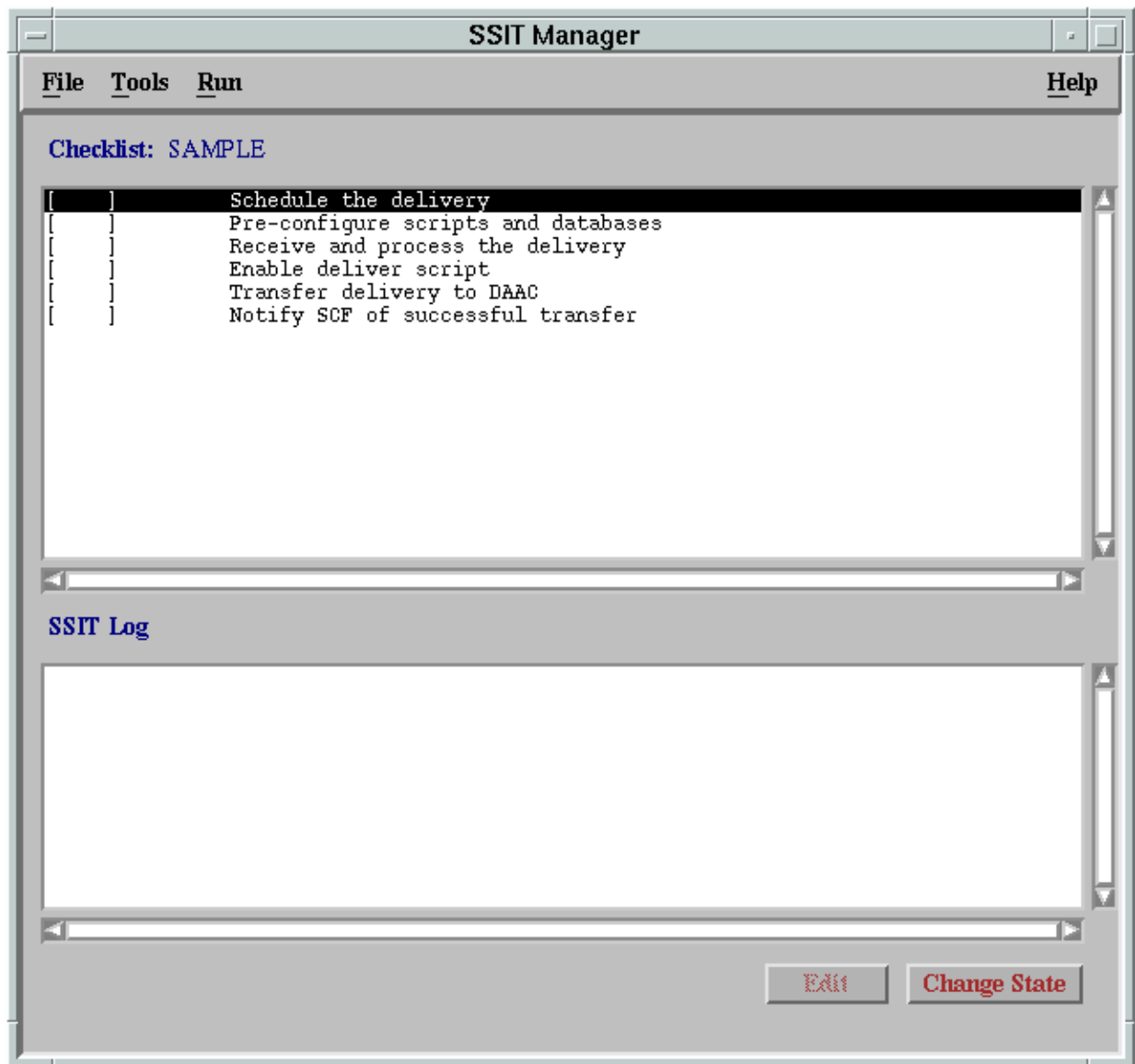
Table 4.5.1-4 lists SGI platform tools associated with the SSIT process.

**Table 4.5.1-4. SGI Tools Description**

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

#### **4.5.1.2 SSIT Manager**

Figure 4.5.1-1 illustrates the SSIT Manager. This screen is the starting point for SSI&T activities. It provides access to a collection of tools that are used in the SSIT 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. SSIT Manager Main Screen**

The SSIT 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 may 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 SSIT 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:

- **File** - Offers the option to exit from the SSIT Manager application.
- **Tools** - Provides access to a collection of tools that are of use in the SSI&T process. The tools that are available are described in the following section.
- **Run** - User-customizable menu, intended for programs such as scripts to run science software.
- **Help** - Provides context-sensitive help.

## SSIT Manager Buttons

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

- **Edit** - The checklist items may be edited by the SSI&T specialist to include additional detail on the SSI&T process. 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.

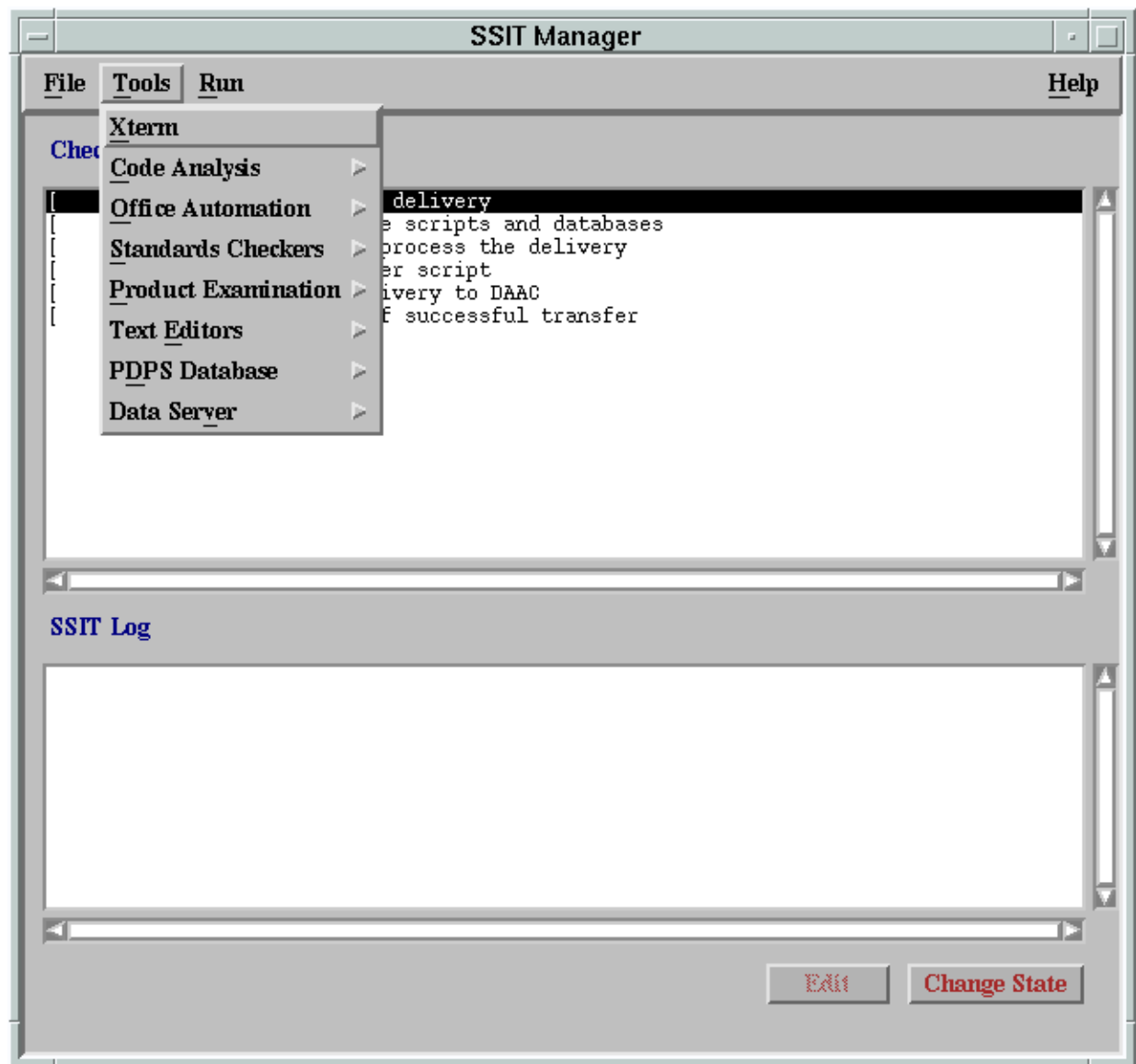
### 4.5.1.2.1 SSIT Manager Tools

There are several tools that are accessible through the SSIT Manager. After selecting the Tools menu option of the menu bar, a set of options is available (see Figure 4.5.1-2 which shows the items available from the menu item).

The first item on the list, the Xterm option, will start an Xterm window session. Selecting any of the other options displays a secondary list of particular tools within that category. The tools categories and tools are described in the following table.

Many of the items accessed through the Tool Menu are COTS products for which documentation exists and requires no further discussion. 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.



**Figure 4.5.1-2. SSIT Manager Main Screen - Tools Menu**

**Table 4.5.1-5. SSIT Manager Tools Description (1 of 5)**

Category	Tools	Tool Description & Use	Further Information
UNIX Access	Xterm	<p>This tools option allows the operator to open an Xterm window and enter Unix commands, as needed</p> <p>The other SSIT Manager windows remain available while an Xterm window is open</p>	Unix documentation is available online via 'man' command
Code Analysis	SPARCworks	<ul style="list-style-type: none"> <li>• SPARCworks is a COTS package developed by Sun Micro Systems</li> <li>• Selecting this option from the Tools list will start the SPARCworks tool which opens a separate window</li> <li>• Among the tools that are available in SPARCworks is the code analysis tool which allows for static code analysis to aid in detecting memory leaks</li> <li>• For the SGI systems the ProDev Workshop product provides similar functionality.</li> </ul>	<ul style="list-style-type: none"> <li>• SPARCworks includes online documentation describing its features</li> <li>• Other SPARCworks documentation is delivered with ECS. This documentation includes: <ul style="list-style-type: none"> <li>– SPARCworks Browsing Source Code</li> <li>– SPARCworks Debugging a Program 3.0.1</li> <li>– SPARCworks Tutorial 3.0.1</li> <li>– SPARCworks Performance Tuning an Application 3.0.1</li> </ul> </li> </ul>
Office Automation	GhostView	<ul style="list-style-type: none"> <li>• Selecting Ghostview from the Tools menu item starts the Ghostview application</li> <li>• Ghostview is a freeware tool</li> <li>• This is used for displaying Postscript format files</li> </ul>	<ul style="list-style-type: none"> <li>• Ghostview includes online documentation on its use</li> <li>• Since this is a freeware product, there is no other documentation provided by ECS for Ghostview</li> </ul>
	Netscape	<ul style="list-style-type: none"> <li>• Netscape is a COTS package developed by Netscape Corp.</li> <li>• This tool is a 'Browser' which is used to access Internet WWW web sites</li> <li>• EOSDIS has adopted the WWW and Netscape as an internal standard for the publication and exchange of various types of documentation, schedules, etc.</li> <li>• ECS documentation is available through the WWW, including documentation related to SSIT</li> </ul>	<ul style="list-style-type: none"> <li>• Documentation on Netscape is available online</li> <li>• Other documentation on Netscape is available in this document in Section 4.12.5, "Netscape Navigator."</li> </ul>

**Table 4.5.1-5. SSIT Manager Tools Description (2 of 5)**

Category	Tools	Tool Description & Use	Further Information
	Adobe Acrobat	<ul style="list-style-type: none"> <li>• Adobe Acrobat Reader is a COTS package developed by Adobe Systems Inc.</li> <li>• This product is used to view electronic documents that have been created in the Portable Document Format (PDF) file format</li> <li>• Many of the ECS documents have been created or converted to this format</li> </ul>	<ul style="list-style-type: none"> <li>• Documentation on Adobe Acrobat products is available online</li> <li>• Other documentation on Adobe Acrobat is available in this document in Section 4.12.5, "Netscape Navigator"</li> <li>• Other documentation for this application is delivered with ECS</li> </ul>
Standards Checkers	FORCHECK	<ul style="list-style-type: none"> <li>• FORCHECK is a COTS tool used to verify that software follows certain coding standards</li> <li>• FORCHECK will analyze source code that is written in Fortran 77 with ANSI extensions</li> <li>• When selected from the SSIT Manager window, and xterm window is created in which the program runs</li> <li>• There is no GUI associated with FORCHECK.</li> <li>• 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."</li> </ul>	<ul style="list-style-type: none"> <li>• Documentation on FORCHECK is available online</li> <li>• Other documentation for this applications is delivered with ECS. This includes:</li> <li>• FORCHECK for Sun/SunOS, A Fortran Verifier and Programming Aid, User Guide</li> </ul>
	Prohibited Function Checker	<ul style="list-style-type: none"> <li>• 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</li> <li>• These coding standards are limited to prohibitions against the use of certain types of functions that may cause problems in the production environment</li> <li>• The Prohibited Function Checker checks for functions defined in a science software source code.</li> </ul>	<ul style="list-style-type: none"> <li>• On-line help is provided with the Prohibited Function Checker</li> <li>• A description of the use of the tool is provided in the following sections.</li> </ul> <p>This tool is described in detail in the following 4.5.1.2.4.2 section.</p>



**Table 4.5.1-5. SSIT Manager Tools Description (3 of 5)**

Category	Tools	Tool Description & Use	Further Information
Standards Checkers	Process Control File Checker	<ul style="list-style-type: none"> <li>• The Process Control File (PCF) checker is a custom application used to verify that the format of the PCF is correct</li> <li>• A PCF is a part of the delivered science software delivery package</li> <li>• The PCF specifies the names and locations of files used by the science software executables, and defines the correspondence between (1) the file specifications and (2) the logical identifiers used by the science software to reference the specified files.</li> </ul>	<ul style="list-style-type: none"> <li>• On-line help is provided with the PCF Checker</li> <li>• A description of the use of the tool is provided in the following sections</li> <li>• Note that this tool consists of a GUI front end added to an ECS SDP Toolkit utility</li> <li>• Documentation of the utility is found in the SDP Toolkit Users Guide for the ECS Project.</li> </ul> <p>This tool is described in detail in the following 4.5.1.2.4.3.1</p>
	Prolog extractor	Custom application which extracts code prologs from science software	Online help is provided
Product Examination	IDL	<ul style="list-style-type: none"> <li>• IDL (Interactive Data Language) is a COTS product, developed by Research Systems Inc.</li> <li>• It is used to interactively visualize and analyze scientific and engineering data products</li> <li>• IDL can be used with data files of an arbitrary format.</li> </ul>	<ul style="list-style-type: none"> <li>• Documentation on IDL is available online</li> <li>• Other documentation for this applications is delivered with ECS</li> </ul>
	EOSView	<ul style="list-style-type: none"> <li>• EOSView is a custom application used to display HDF format files and the metadata associated with those files</li> <li>• EOSView is expected to be primarily by the EOS scientific community, but will also be used by the operations community to support the SSIT process (for file inspection, etc.) and by the DAAC quality assurance groups to view and analyze problems arising from production operations</li> </ul>	<ul style="list-style-type: none"> <li>• Documentation on EOSView is available online</li> <li>• EOSView is integral with the ECS Desktop; for more information, see Section 4.12.7, "EOSView"</li> </ul>

**Table 4.5.1-5. SSIT Manager Tools Description (4 of 5)**

Category	Tools	Tool Description & Use	Further Information
Product Examination	File Comparison	<ul style="list-style-type: none"> <li>• File comparison tools, which are custom products of ECS, are provided with the SSIT Manager</li> <li>• Tools are provided to support file comparison in three different formats: ASCII, Binary, and HDF</li> <li>• These tools would be used by the SSIT Specialists to compare the benchmark data products generated at the SCFs with the data products generated as a part of the SSIT process at the DAAC</li> <li>• All standard data products will be produced in HDF format, for which the HDF file comparison utility will be used</li> <li>• For intermediate or other data products that may not adhere to the standards, the binary or ASCII file comparison tools may be used</li> <li>• In the case of binary files, the DAAC operator must write code to do the comparison, for which a support environment is provided</li> </ul>	<ul style="list-style-type: none"> <li>• Documentation on the file comparison tools is available online</li> <li>• These tools are described more fully in following subsections of this document.</li> </ul>
<b>Text Editors</b>	Emacs	Emacs is a freeware text editing tool most frequently used by the SSIT specialists in developing scripts or ad hoc programs to be run in the SSIT environment	<ul style="list-style-type: none"> <li>• Documentation on Emacs is available online</li> <li>• Since this is a freeware product, there is no other documentation provided by ECS</li> </ul>
	Xedit	Xedit is a freeware text editing tool most frequently used by the SSIT specialists in developing scripts or ad hoc programs to be run in the SSIT environment	<ul style="list-style-type: none"> <li>• Xedit includes online documentation on its use</li> <li>• Since this is a freeware product, there is no other documentation provided by ECS</li> </ul>
<b>PDPS Database</b>	PCF ODL Template	Creates a PDPS PGE SCIENCE metadata ODL file from a science software PCF	This tool is described in detail in the following 4.5.1.2.8.1 section.
	Check ODL	Parses ODL file and records errors. This tool accesses the PDPS DB for information on PGE/ESDT whose ODL is already available	This tool is described in detail in the following 4.5.1.2.8.2 section.
	SSIT Science Metadata Update	Updates PDPS/SSIT database with PGE and ESDT SCIENCE metadata read from ODL files	This tool is described in detail in the following 4.5.1.2.8.3 section.

**Table 4.5.1-5. SSIT Manager Tools Description (5 of 5)**

Category	Tools	Tool Description & Use	Further Information
	SSIT Operational Metadata Update GUI	Updates PDPS/SSIT 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 the following 4.5.1.2.9.1 section.
	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 the following 4.5.1.2.9.2 section.
	Insert Static	Inserts a static input file to the Data Server	This tool is described in detail in the following 4.5.1.2.9.3 section.
	Insert Test Dynamic	Inserts a test dynamic input file to the Data Server	This tool is described in detail in the following 4.5.1.2.9.4 section.
	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 the following 4.5.1.2.9.5 section.
	SSAP Editor	Read/Write Metadata and record errors.	This tool is described in detail in the following 4.5.1.2.9.6 section.

#### **4.5.1.2.2 Xterm**

Xterm is a COTS product. This “tool” activates an Xterm window. This allows the operator to use the Unix environment and resources from a PC. The Xterm window has all the properties of a PC window while the “screen” portion is essentially a Unix terminal. The user will need 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 provides the SPARCworks tool to assist in analyzing C code.



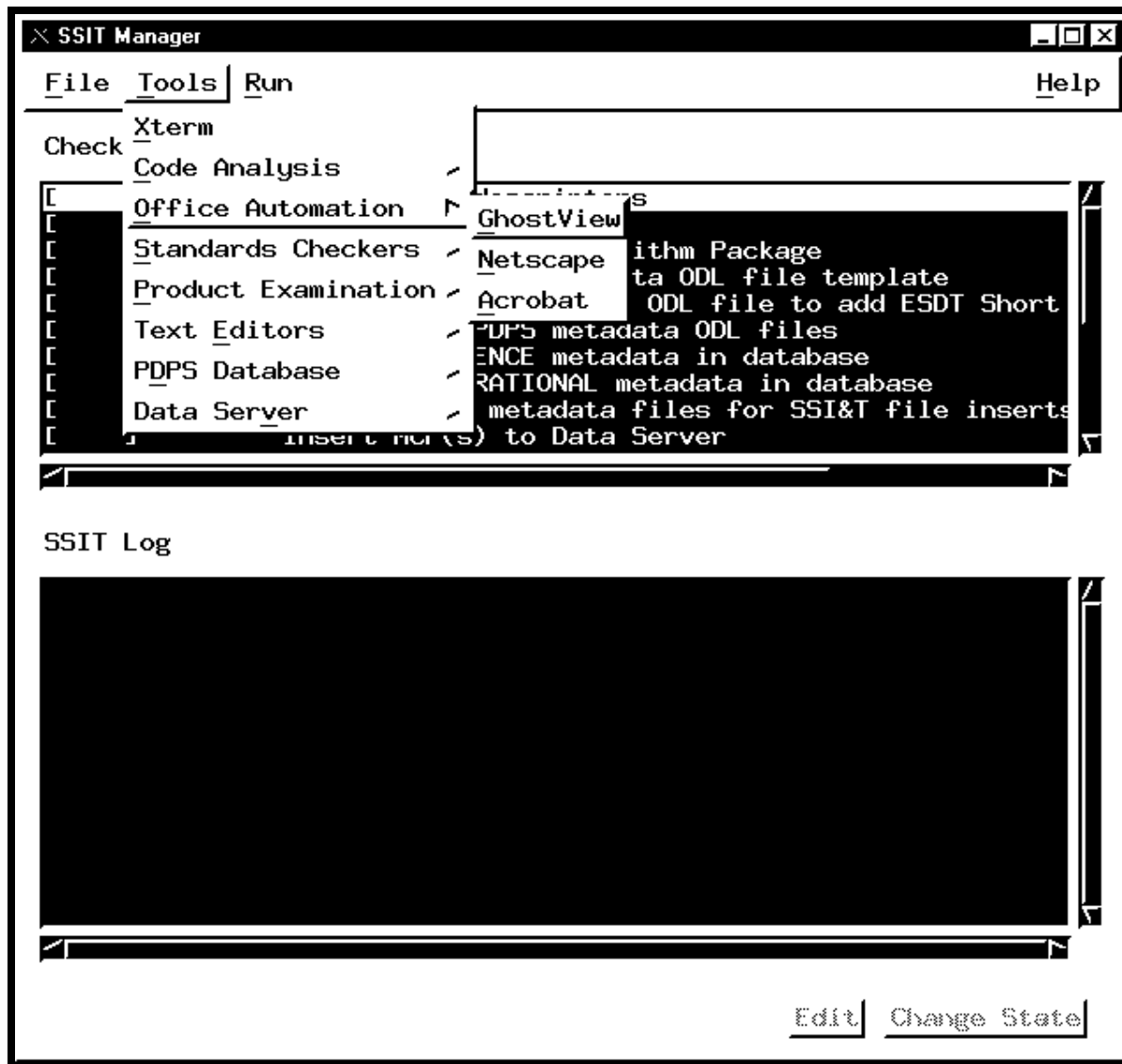
**Figure 4.5.1-4. SSIT Manager Main Screen with Code Analysis tool menu displayed.**

#### 4.5.1.2.3.1 SPARCworks

SPARCworks is a COTS product. This function is not documented for operators. Analysis of the program code in the operational environment will result in significant performance reduction.

#### 4.5.1.2.4 Office Automation

The Office Automation menu provides tools to assist operators in documentation and administrative tasks.

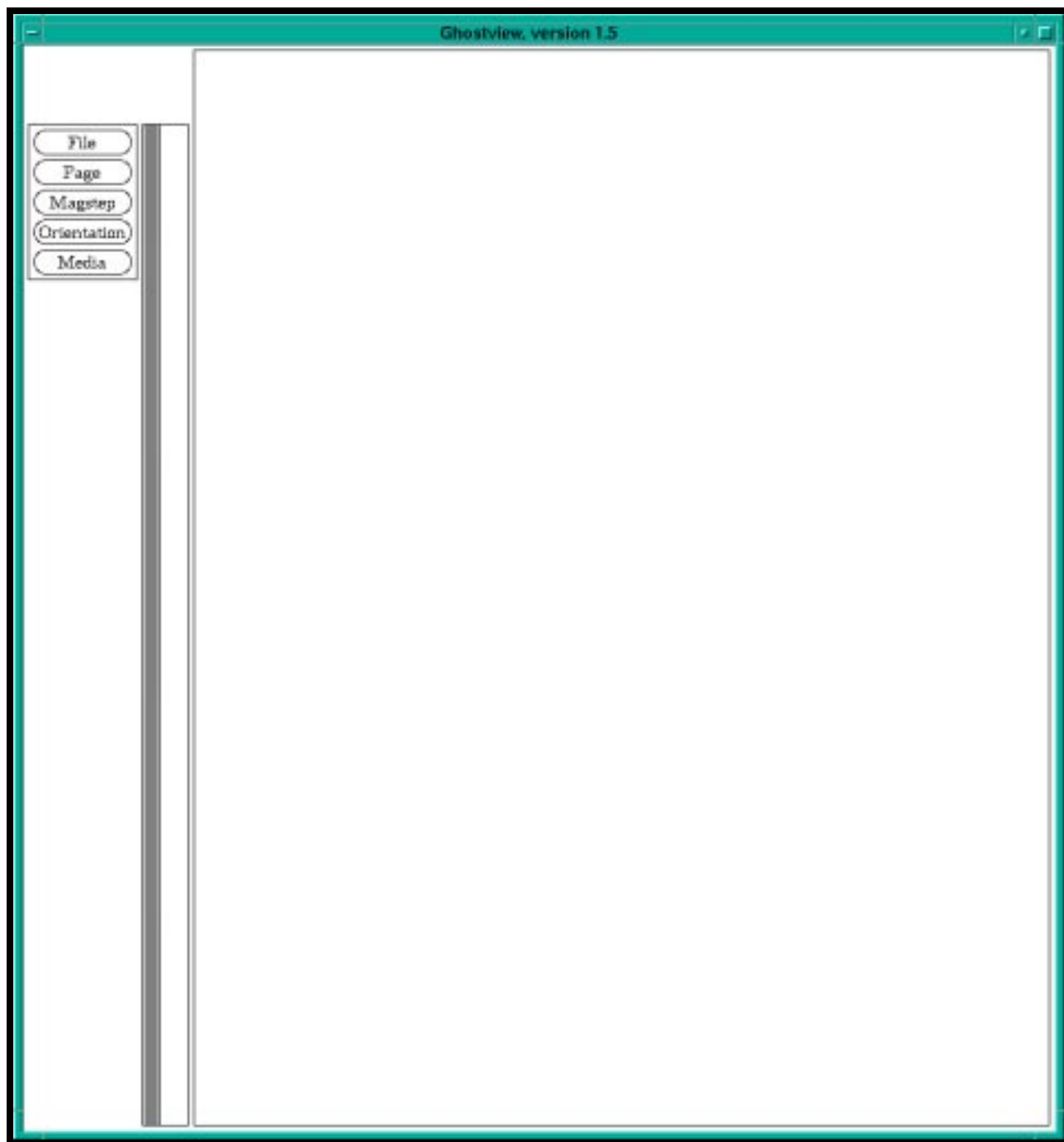


**Figure 4.5.1-5. SSIT Manager Main Screen with Office Automation tool menu displayed.**

#### **4.5.1.2.4.1 GhostView**

GhostView is a COTS product. GhostView 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 may be cut & pasted into MS Word® documents.

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



***Figure 4.5.1-6. GhostView Main Screen***



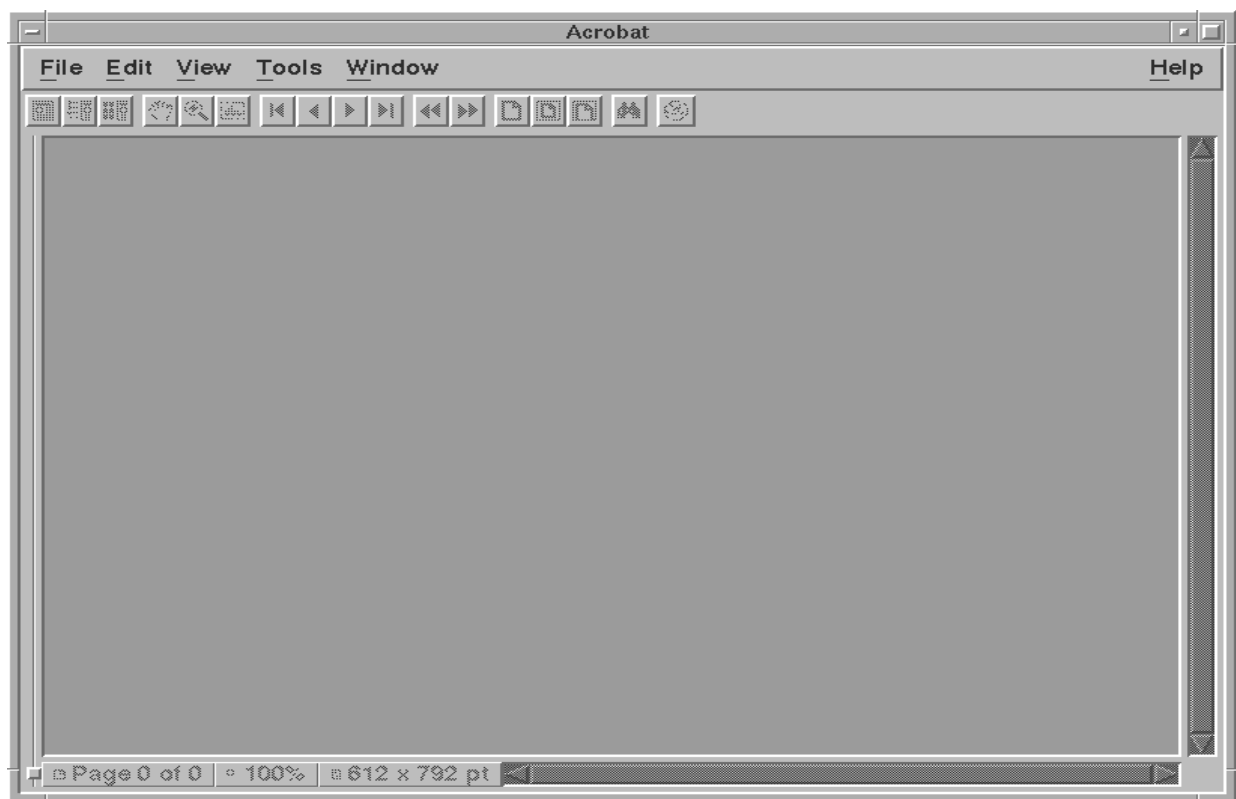
#### **4.5.1.2.4.2 Netscape**

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 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 may be cut & pasted into Word® documents.

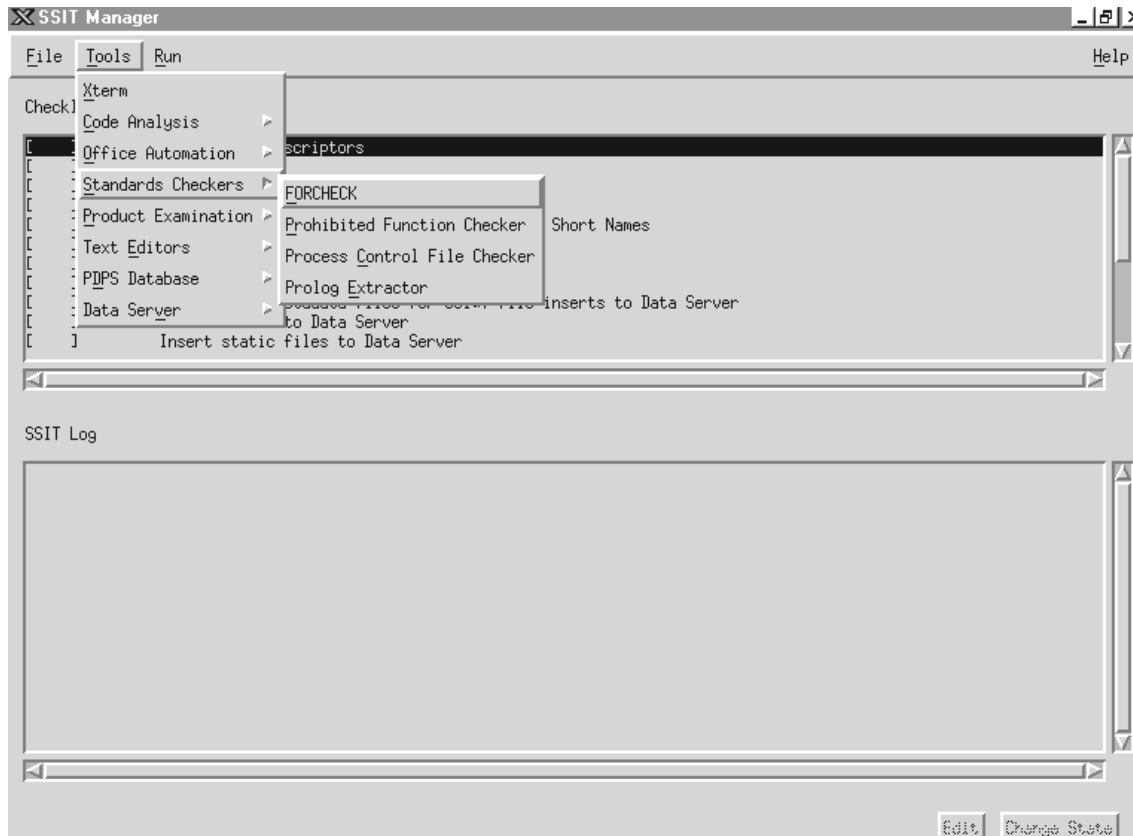
More information about Acrobat is available on the Web free from <http://www.adobe.com/>.



***Figure 4.5.1-7. Acrobat Main Screen***

#### **4.5.1.2.5 Standards Checkers**

The Standards Checkers menu provides tools to assist operators in validation of DAPs.



**Figure 4.5.1-8. SSIT Manager, Tools Menu, Standards Checker Submenu Choices**

#### 4.5.1.2.5.1 FORCHECK

FORCHECK is a COTS tool. FORCHECK checks science software written in FORTRAN 77. FORCHECK may 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. Note: FORCHECK does not process FORTRAN 90 code.

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



**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 User's Guide* for more information.

#### **4.5.1.2.5.2 Prohibited Function Checker**

The Prohibited Function Checker is an SSI&T tool that uses a GUI interface to check specified source code for prohibited function calls. 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 may cause errors in the production environment.

Prohibited Function Checker (PFC) GUI serves the following functionality:

- Toolkit to process program and text files for prohibited functions.
- GUI interface that allows the user to select file(s) to be checked.
- Writes output to a log.
- User can scan through any prohibited functions found from the output.

The following table provides the language and the file extensions that can be checked for prohibited function calls in 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

**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 can 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.

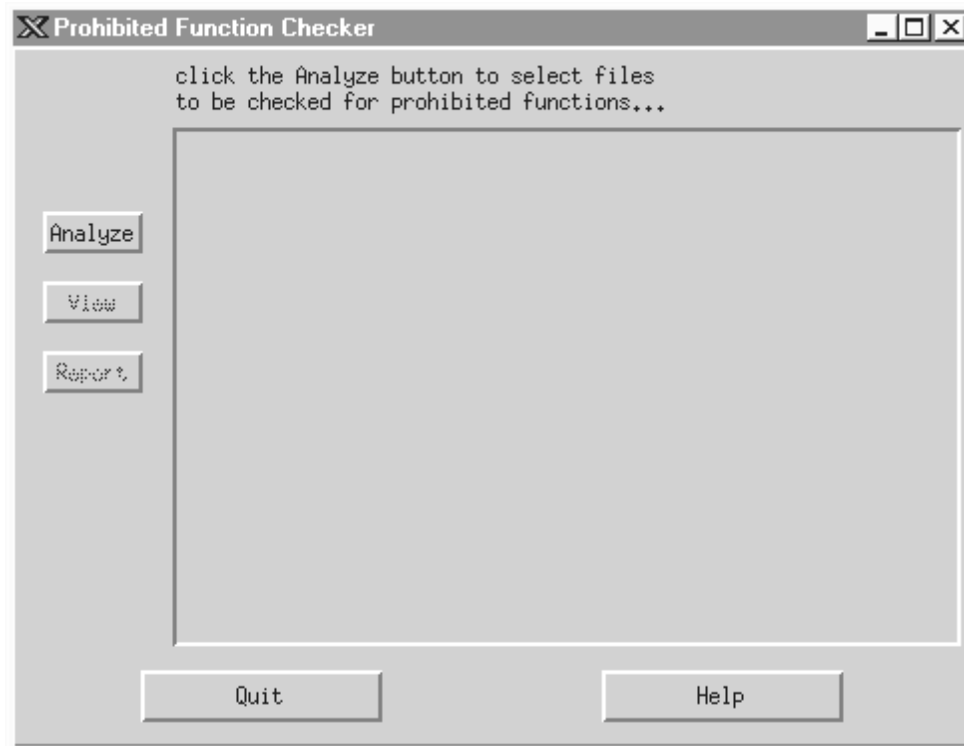
#### **4.5.1.2.5.2.1 Checking for Prohibited Functions: GUI Version**

Assumptions:

The SSI&T Manager is running.

Steps to Check for Prohibited Functions:

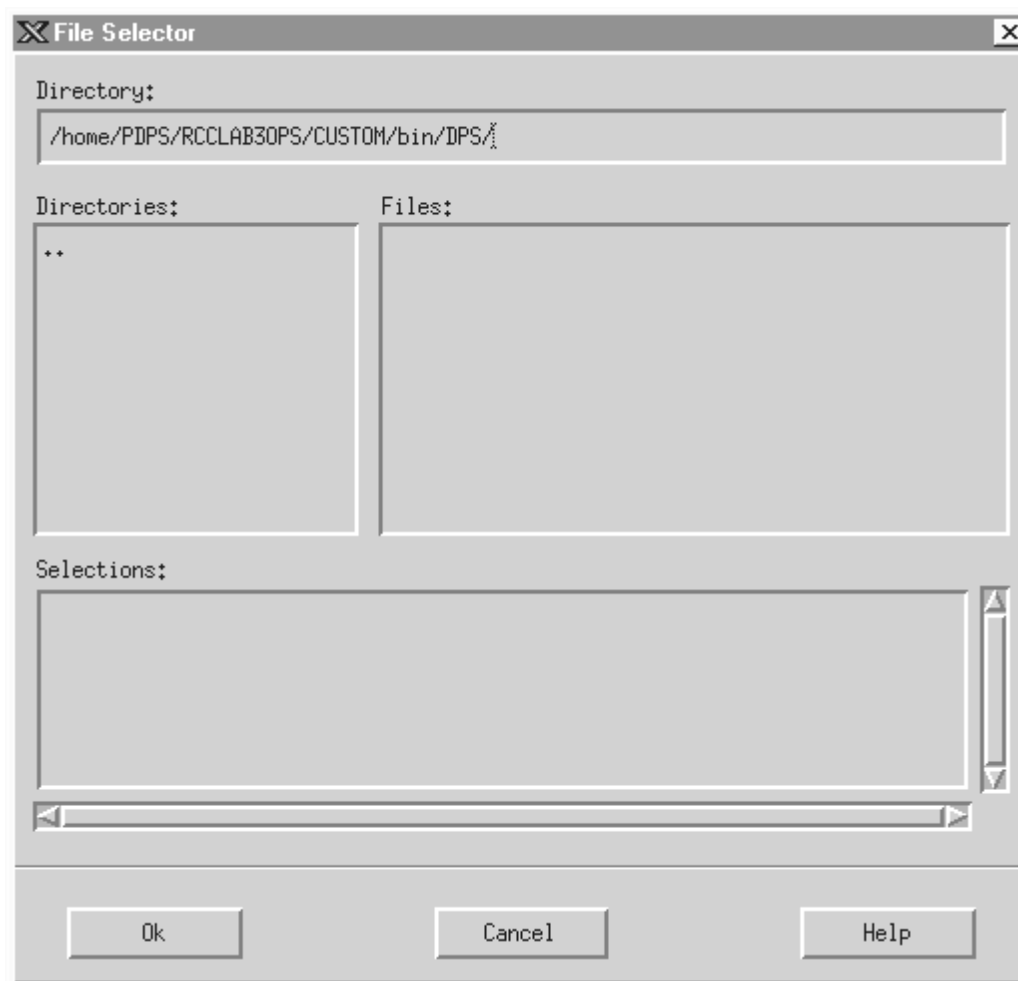
1. From the *SSI&T Manager Main Screen*, click on the **Tools** menu, then choose **Standards Checkers**, and then choose **Prohibited Function Checker** (Figure 4.5.1-8).
  - This displays the *Prohibited Function Checker Pop-up*
  - This Pop-up has five buttons (Analyze, View, Report, Quit and Help) which are described in this section.



**Figure 4.5.1-10. Prohibited Function Checker Pop-up**

2. In the *Prohibited Function Checker POP-UP*, click on the **Analyze** button.
  - ◆ The *File Selector POP-UP* is displayed.
3. In the *File Selector POP-UP* (Figure 4.5.1-11 below), choose the **Directories** sub-window, and double click on the desired directory.
 

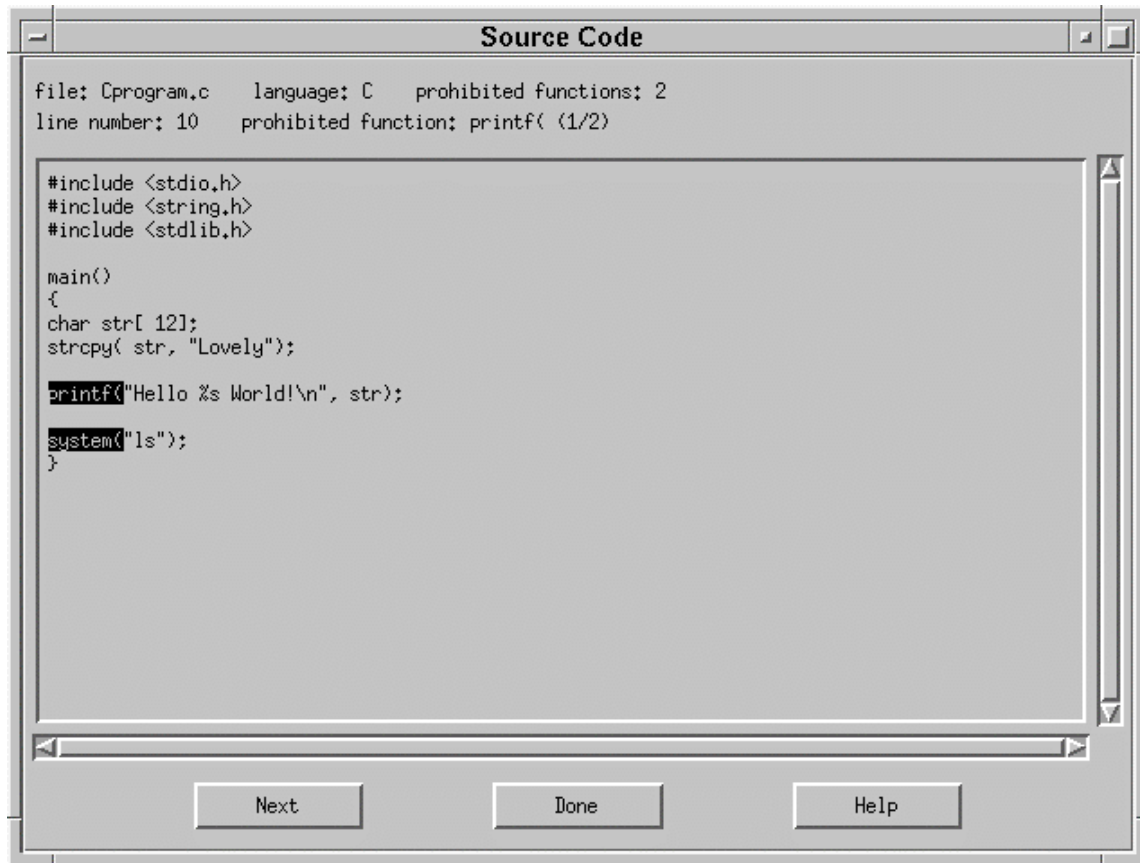
Repeat this step until the directory with the source files to be checked are displayed in the **Files** sub-window.
4. Within the **Files** sub-window, click on the source files to be analyzed. Highlight each source file by clicking on it.
  - ◆ To choose groups of contiguous files, hold down the left mouse button and drag the mouse down.
  - ◆ To choose non-contiguous files, hold down the Control key while clicking on file names.
5. The files selected will show in the **Selections** sub-window.



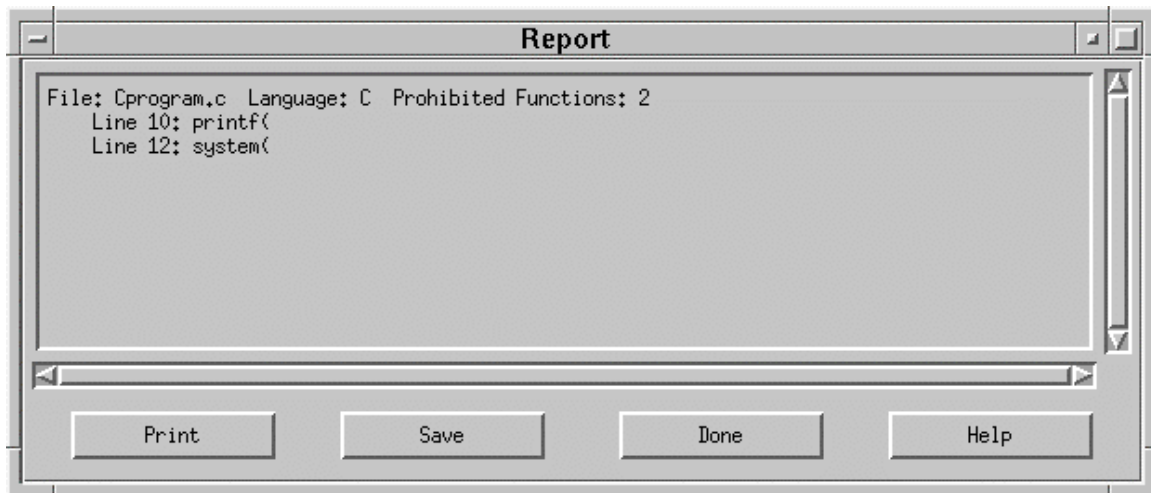
**Figure 4.5.1-11. File Selector POP-UP**

6. In the *File Selector POP-UP*, click on the **OK** button. The *Prohibited Function Checker* will then start analyzing for prohibited functions within the selected files.
7. The *File Selector POP-UP* disappears, and the *Prohibited Function Checker* window displays all of the selected files containing prohibited functions. The results of the *Prohibited Function Checker* can be examined using the View button (only one source file can be selected at a time to view the code) or Report button.
8. Highlight one of the source files in the *Prohibited Function Checker* window, and then click on the **View** button.
  - ◆ The *Source Code POP-UP* is displayed (Figure 4.5.1-12).
  - ◆ 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*.
- ◆ All prohibited functions can be viewed similarly, by examining one source file at a time.



**Figure 4.5.1-12. Source Code POP-UP**



**Figure 4.5.1-13. Report POP-UP**

9. Use the Report button on the *Prohibited Function Checker POP-UP* to:
  - ◆ Display the *Report POP-UP* (Figure 4.5.1-13).
  - ◆ Display for each source file, a list of prohibited functions found.
10. The results of the source file, 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*. Specify the directory and file name in which to save the results.
  - ◆ Clicking on the **Print** button prints the results on the default printer.
11. The **Help** button within any WINDOW brings up the HELP POP-UP.

#### **4.5.1.2.5.3 The Process Control File Checker**

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

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



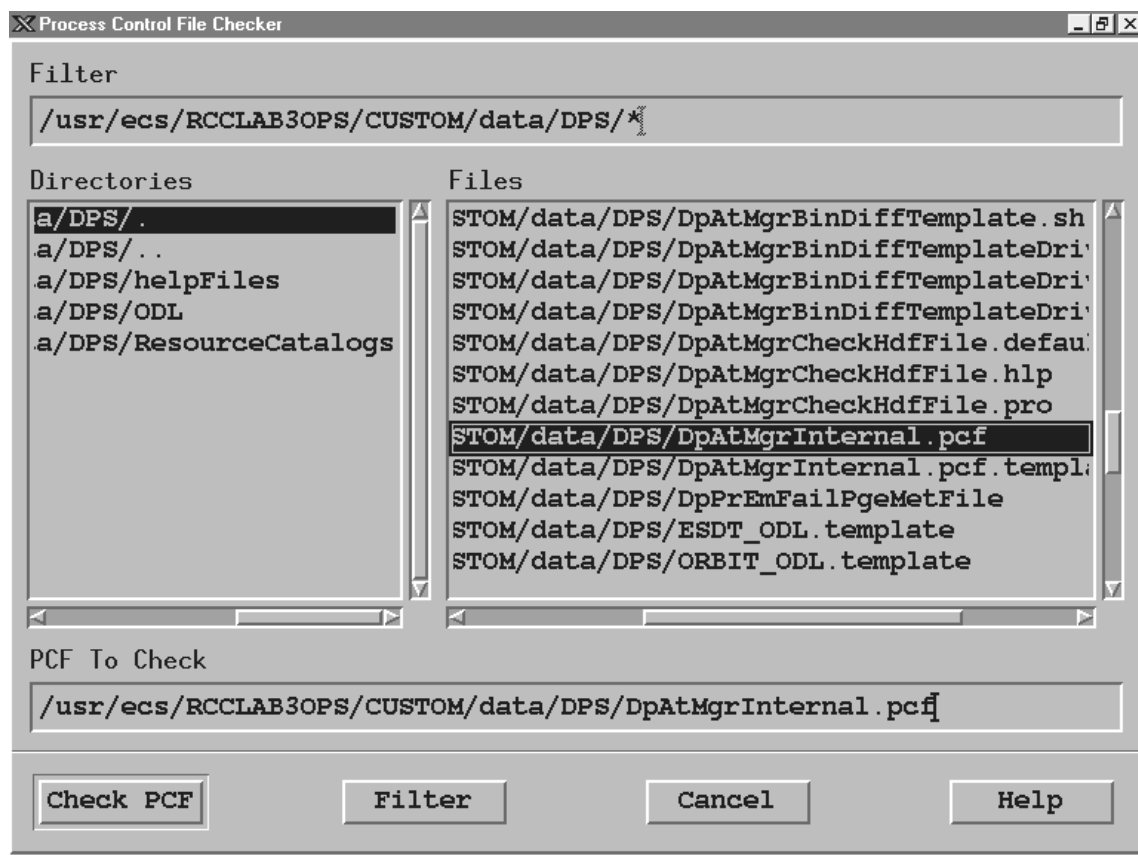
#### 4.5.1.2.5.3.1 Process Control File Checker

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 PCF Checker tool is started from the TOOLS item on the menu bar, on the *STANDARDS CHECKER* menu from the SSIT Manager Pop-up. Figure 4.5.1 -8 shows the submenu choices.

When the PCF Checker application is started from the TOOLS/STANDARDS CHECKER menu item, the Pop-up shown in Figure 4.5.1-14 is displayed.



**Figure 4.5.1-14. PCF Checker Pop-up**

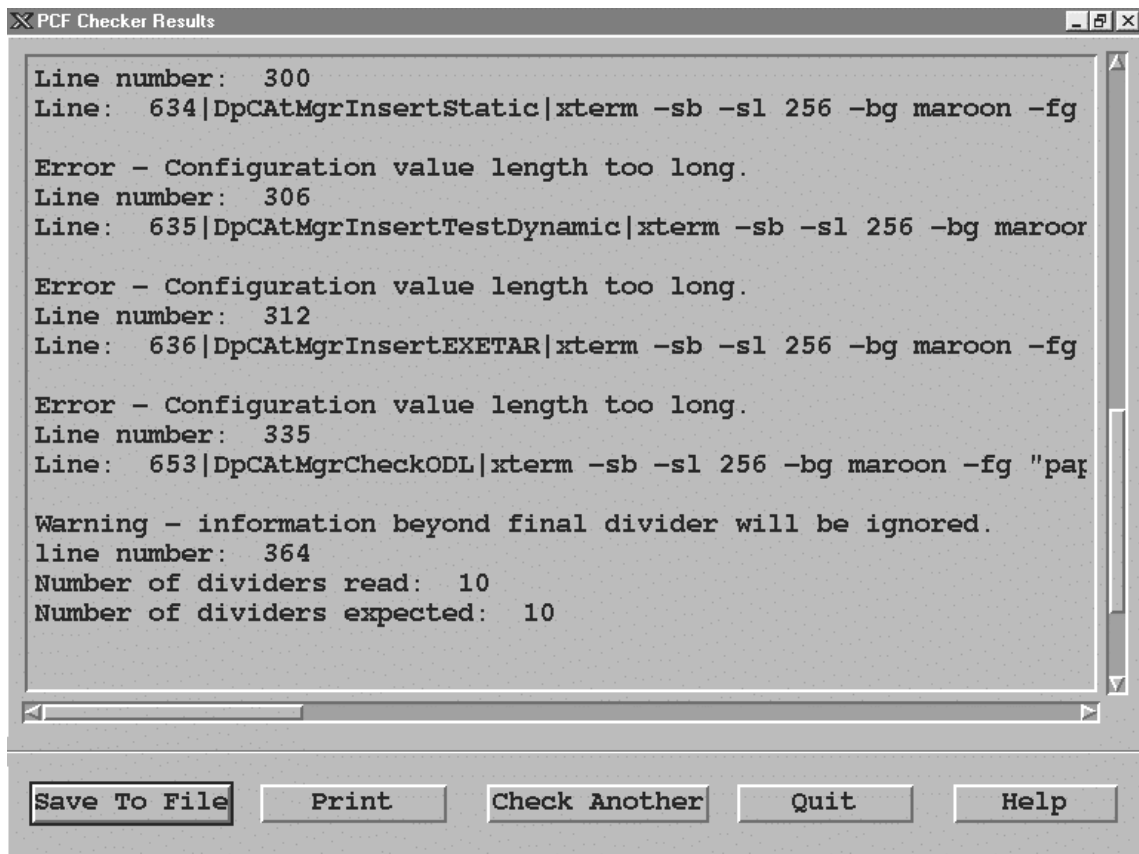
Table 4.5.1-7 describes the PCF Checker's fields.

**Table 4.5.1-7. 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

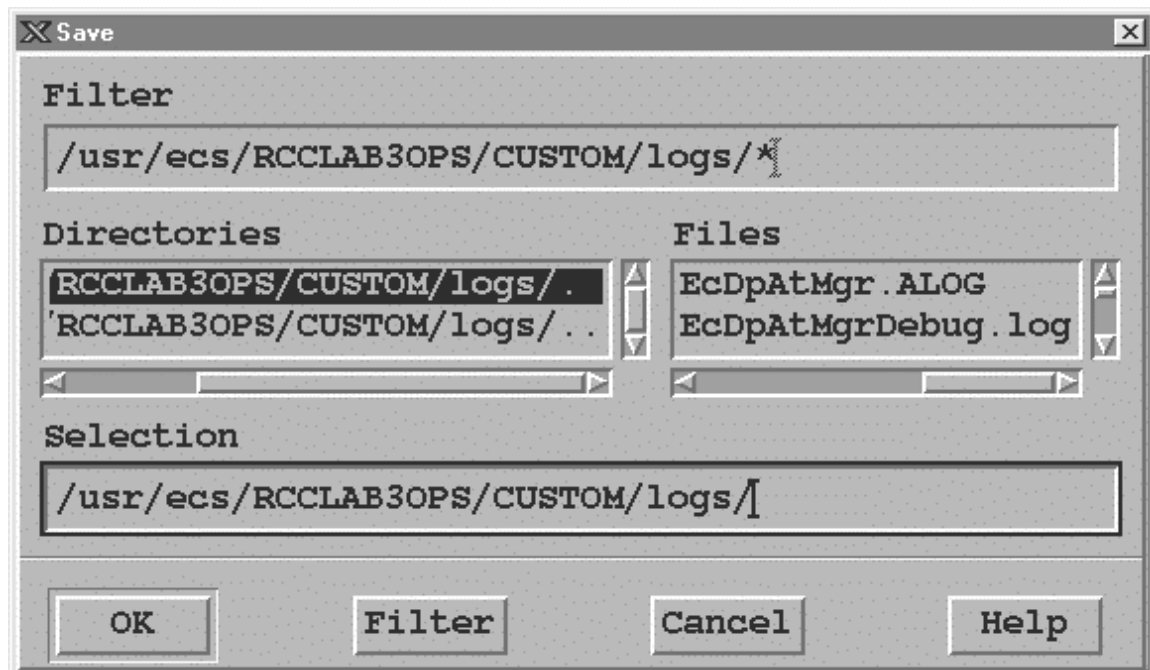
The PCF Pop-up allows the user to work through the directory structure on the local machine in order to select PCFs to be checked, by selecting the desired directory in the Directories window. The Filter window allows users to limit the files that are displayed. From the File window, the user can select 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.

If the selected file does not exist, an error dialog is displayed to inform the user. Similarly, if the specified PCF exists but can not 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.

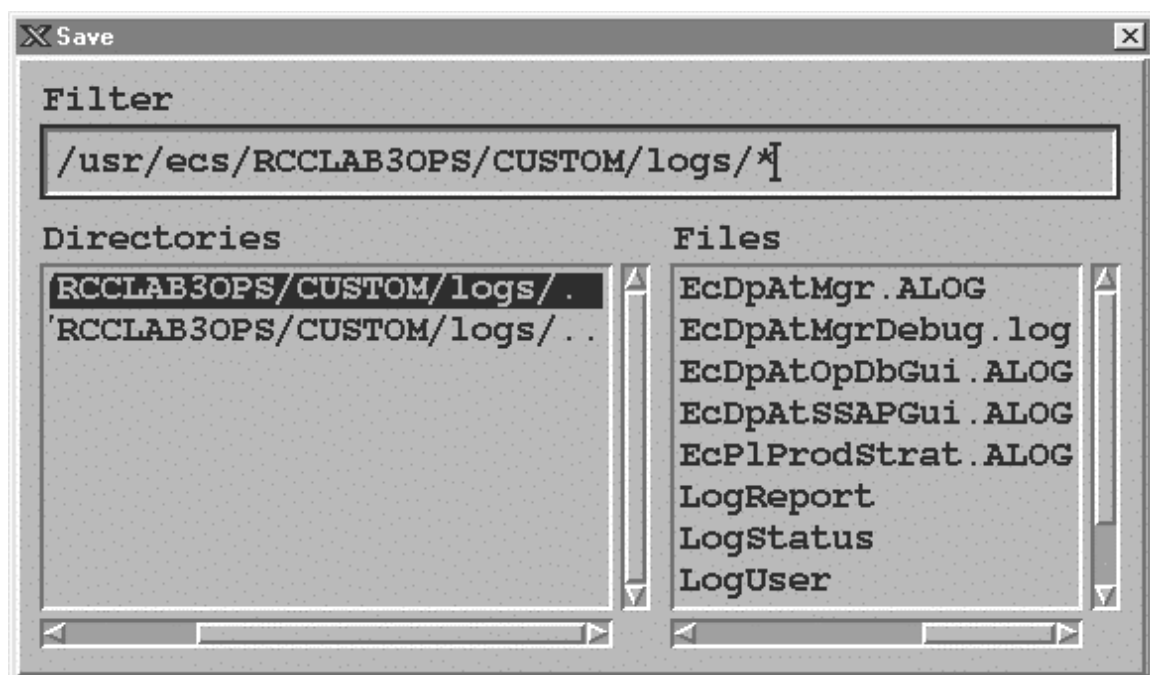


**Figure 4.5.1-15. PCF Checker Results Pop-up**

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 will bring 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.**



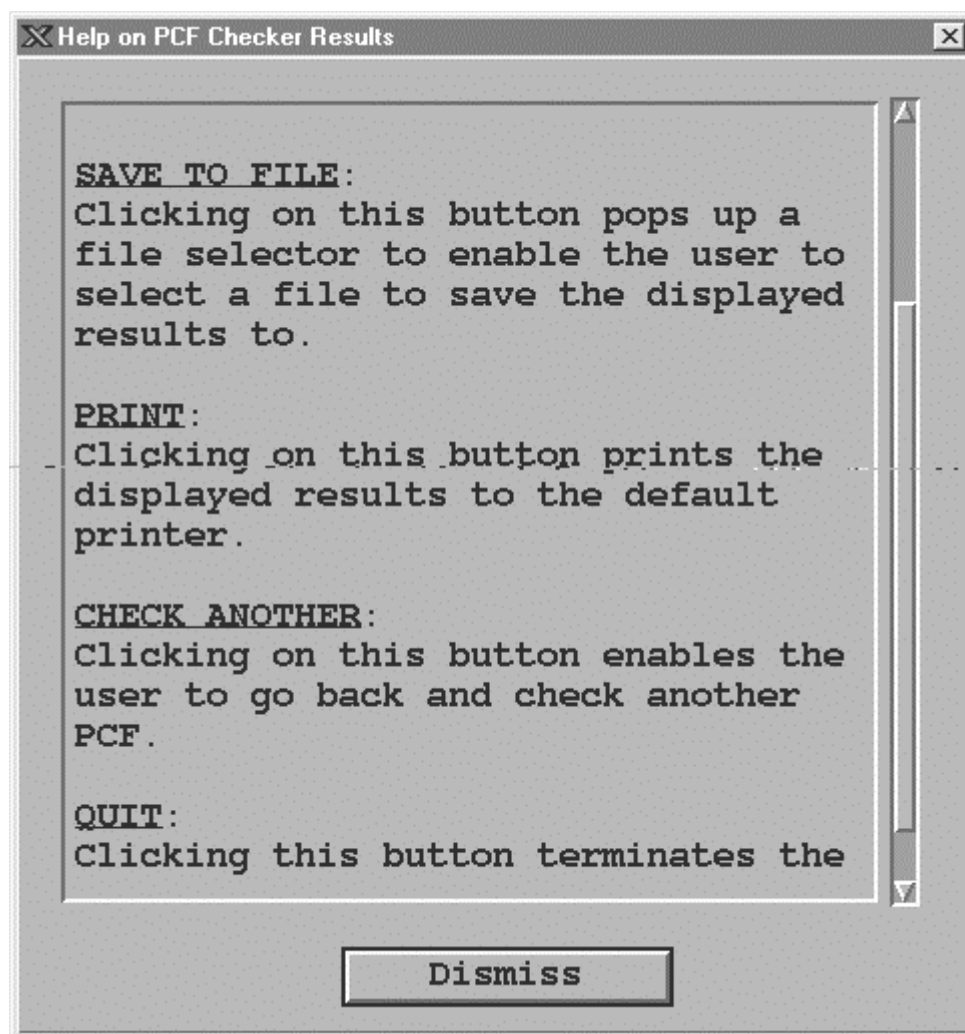
**Figure 4.5.1-17. The Pop up display for Filtering files from the Save Pop-up**

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 may select a listed file or may enter his/her own selection. Selecting the Filter button brings up another screen that allows the user to limit the number of files listed. Figure 4.5.1-17 shows the filter screen.

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 the starting delimiters are

**Table 4.5.1-8. Prolog Extractor standard delimiters**

Language	Type	Delimiter
FORTRAN 77	source	!F77
FORTRAN 90	source	!F90
C	sourc	!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 may be specified through use of command line flags (see below).

The Prolog Extractor program will return a warning if the module contains no prologs at all. It does not warn of functions within a particular file that are missing prologs.

In order for this tool to work, the file must have one of the following file extensions:

**Table 4.5.1-9. 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 SSIT Manager under the TOOLS menu item Standards Checkers.

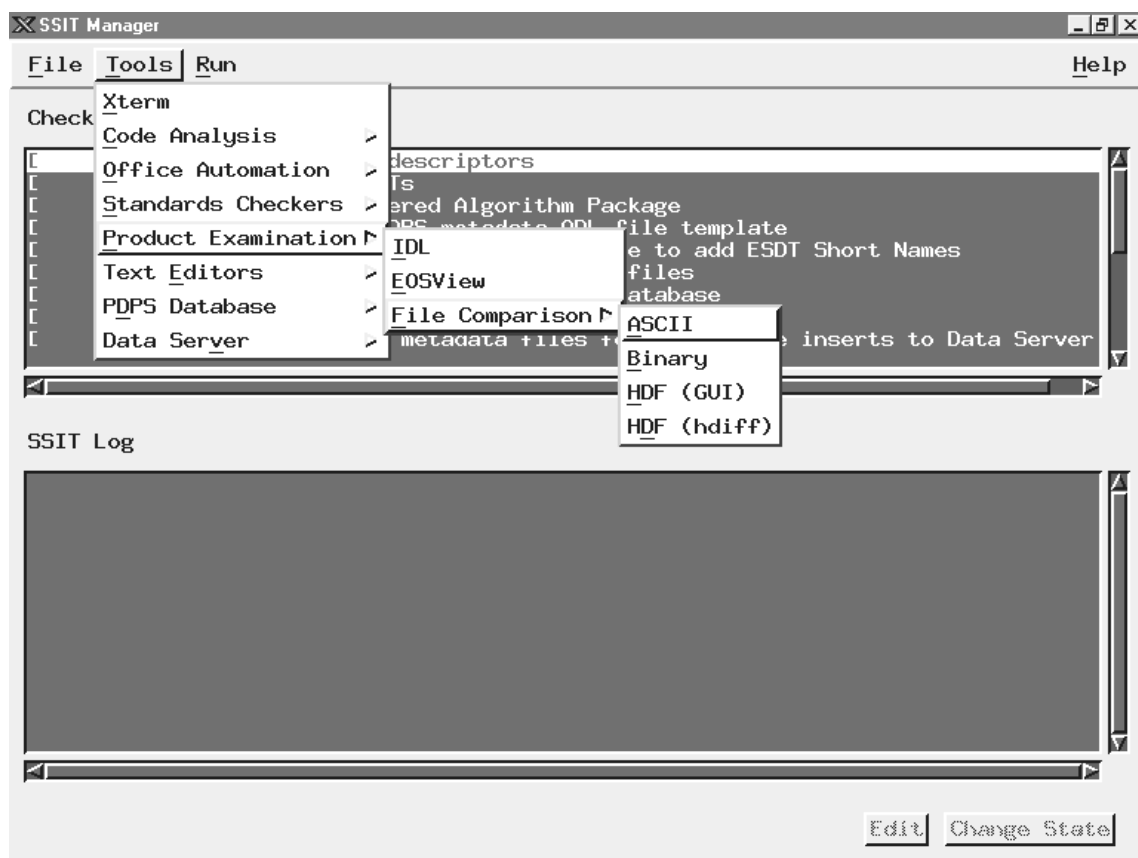
Output from this tool is always written to a file.

By default, the SSIT Manager internal PCF specifies that this output file be named prologs.txt in the local directory. You may change this by editing the internal PCF, item #650.

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. 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. SSIT 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, that provides capabilities for data visualization and comparison of various file formats. IDL is provided so that 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 described in section 4.12.5, “EOSView”.

#### **4.5.1.2.6.3 File Comparison Tools**

This section describes how the File Comparison tools are used by the SSI&T Specialist during the SSI&T process.

The File Comparison tools are selected from the Tools item on the menu bar, and appears 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 would be most frequently used to insure that the data products generated from the PGEs run in the SSI&T environment are identical (within tolerance) to the files that were produced by the PGEs at the SCF.

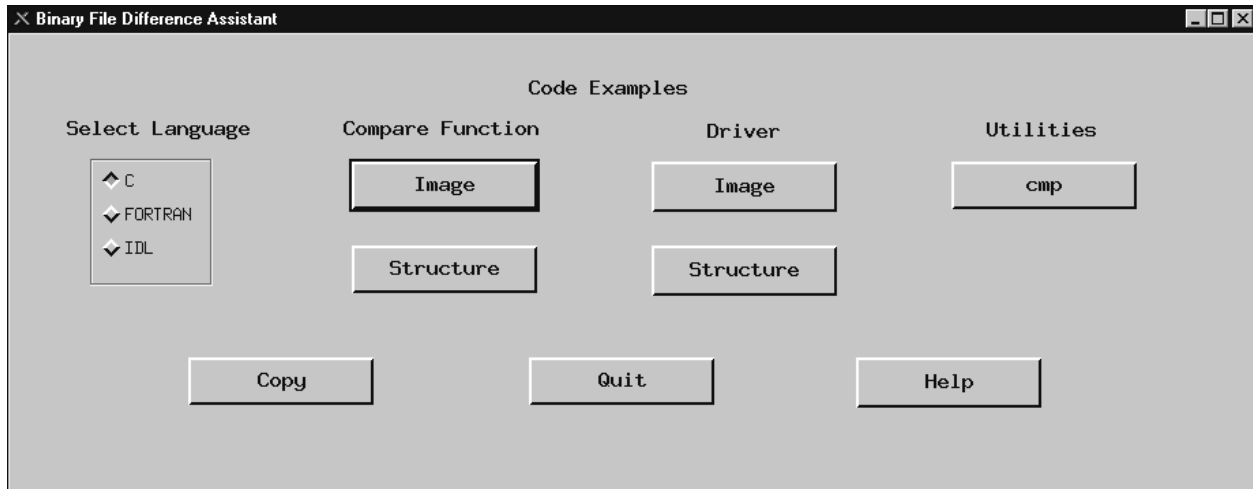
##### **4.5.1.2.6.3.1 ASCII File Comparison Tool**

When the ASCII File Comparison tools is started from the TOOLS/PRODUCT EXAMINATION menu item, an Xterm is spawned, which prompts the user for input. The user enters two ASCII filenames, and the differences between the files is displayed. The Tools is based on the use of the X-Window application `xdiff`.

##### **4.5.1.2.6.3.2 Binary File Difference Environment**

When the Binary File Difference Environment is started from the TOOLS/PRODUCT EXAMINATION menu item, 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 is 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 that was 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 IDL (Interactive Data Language)].
- 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.

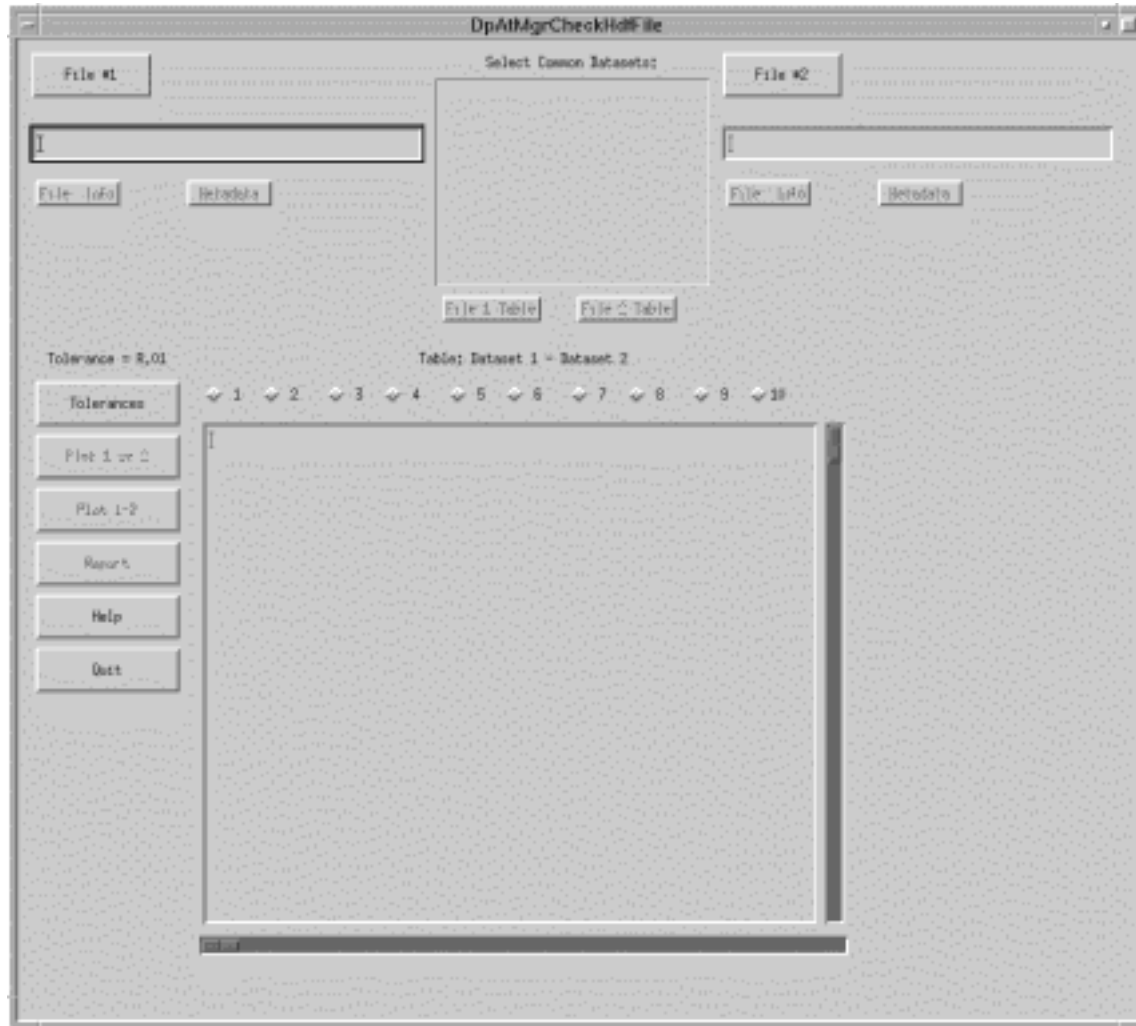
The **Code examples** section of the Pop-up displays simple code examples that do binary file comparison. The operator may copy and paste these in their code if desired. Examples have been compiled and tested. This sample code is provided to help the operator in visualizing a completed comparison module. 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 SSIT Manager under the **Tools** menu item **Product Examination**, sub-menu **File Comparison**, sub-menu **HDF (GUI)**.

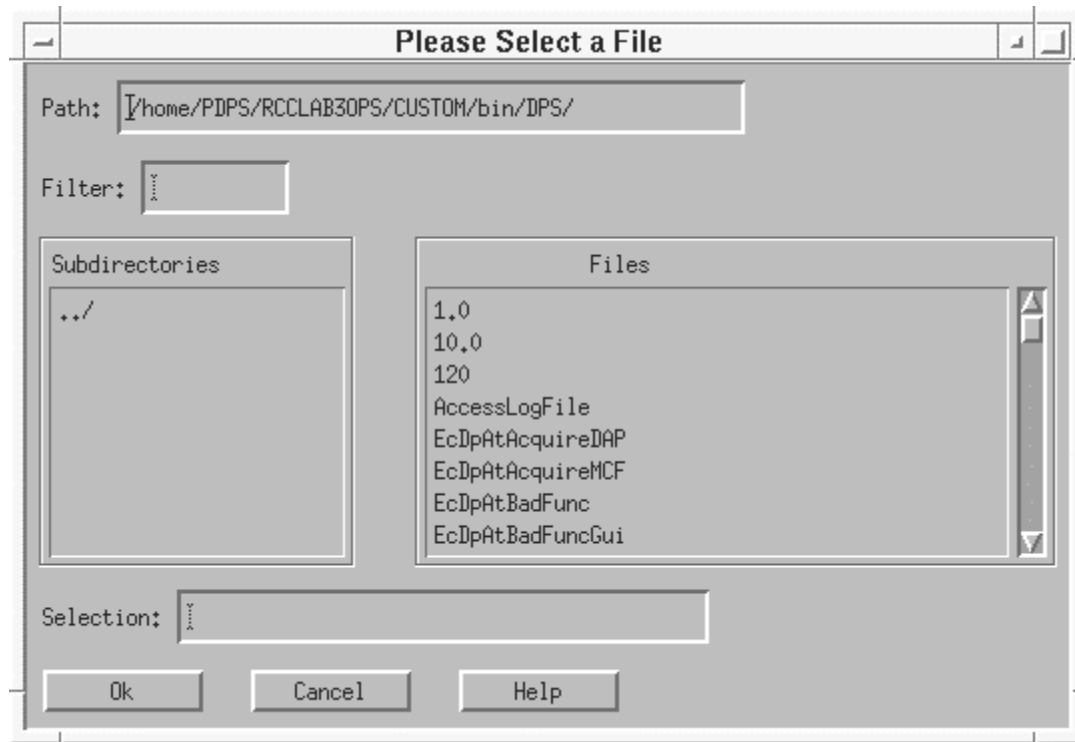
The tool is also available from the UNIX command line by typing *EcDpAtCheckHdfFile*. Both methods will result in the display of the screen in Figure 4.5.1-21 (*note that this program is written in IDL, so it invokes the IDL environment to run.*)



**Figure 4.5.1-21. HDF (GUI) Pop-up**

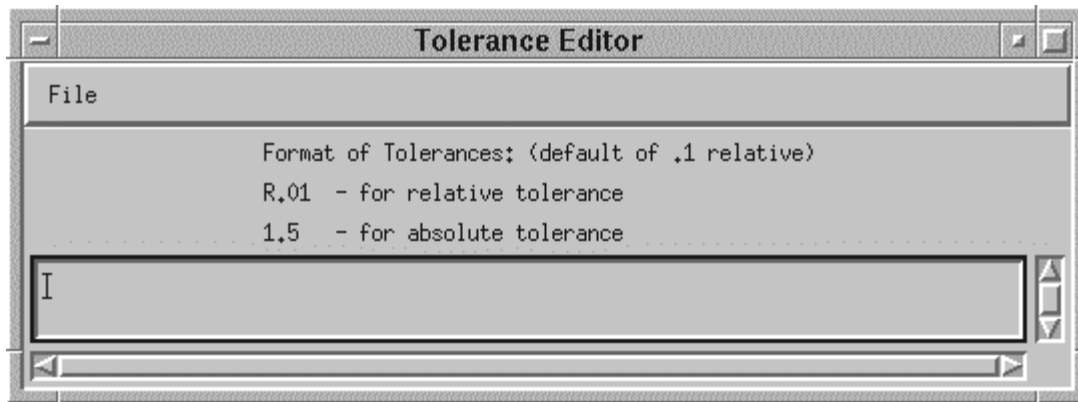
The HDF File Difference Tool Pop-up allows the operator to compare and contrast 2 HDF files via a graphic user interface. The button functions on this Pop-up are discussed

- The selection of the files to be compared is done by clicking the **File #1** or **File #2** buttons. 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 will return to the HDF File Comparison Pop-up and insert 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 provides also the facility to select the **Common Datasets** of both files from the list available in the center upper part of the screen.
- Once the datasets are identified, the operator selects the type of tolerance (relative or absolute) that is used to compare the numerical values of the two datasets. This is accomplished by clicking on the **Tolerances** button (Figure 4.5.1-21) and typing the appropriate value of tolerance as shown in Figure 4.5.1-23.

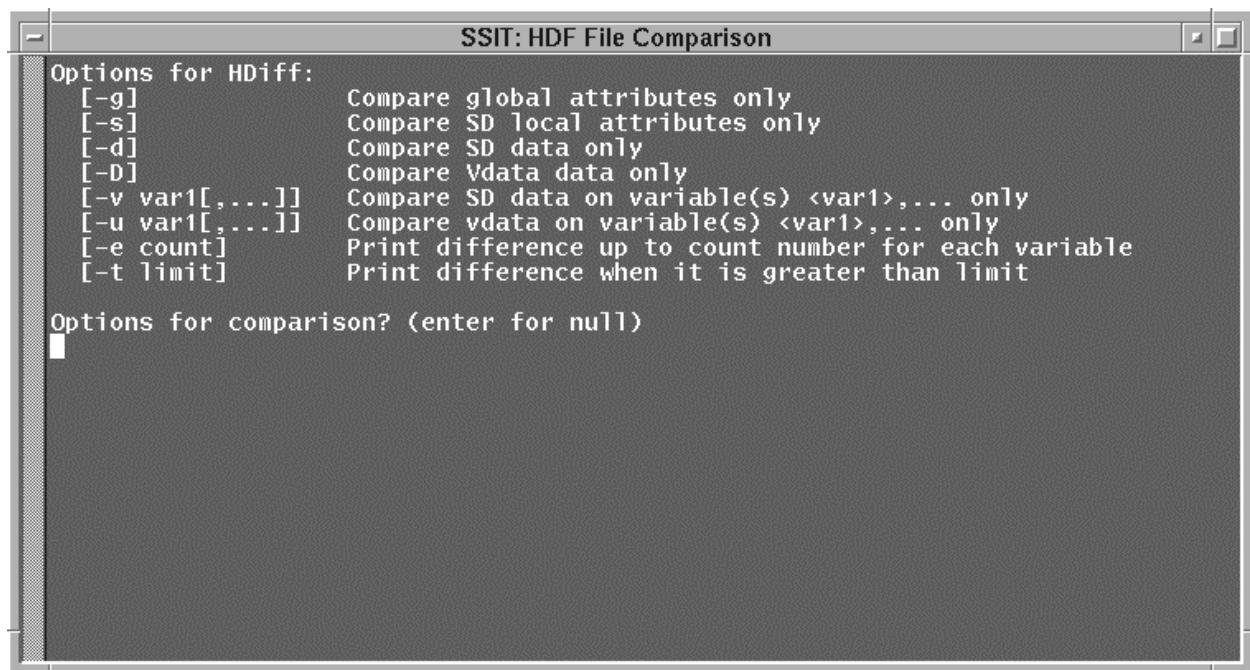


**Figure 4.5.1-23. Tolerance Editor Pop-up**

- This interface allows the operator to input comparison coefficients (shown in Figure 4.5.1-21 as radio buttons **1** through **10** next to the **Tolerance** 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 datasets;
  - **Plot 1-2** shows the difference between the two datasets.
- Documentation for the HDF file difference tool is obtained by clicking on the **Help** button on the main screen.
- Clicking on the **Quit** button will close this Tool.

#### **4.5.1.2.6.4 HDF File Comparison - hdiff**

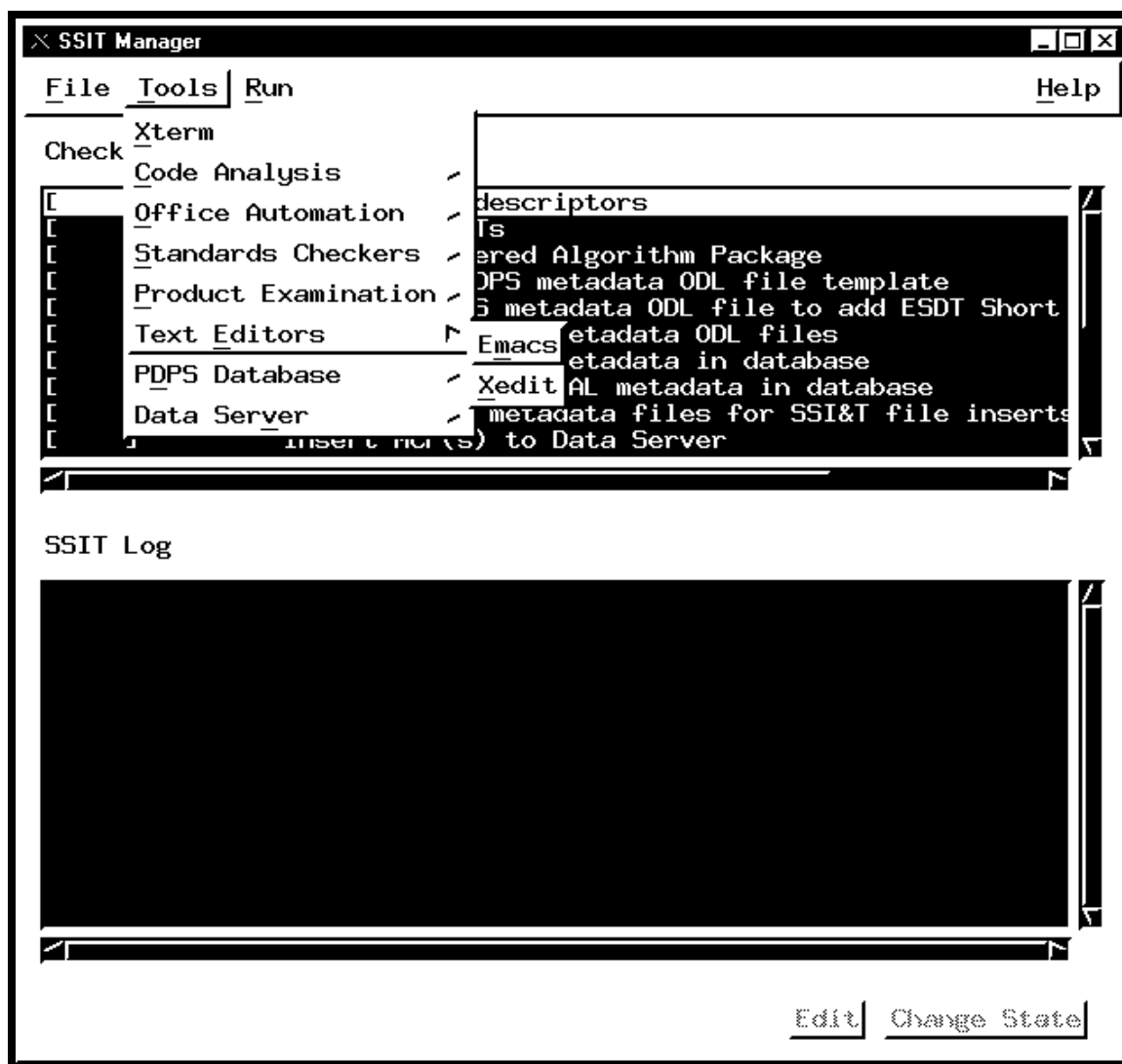
The HDF File Comparison hdiff tool spawns a Xterm when it is started from the TOOLS/PRODUCT EXAMINATION menu item which prompts the user for input. There is no graphics screen for this function. It is run through the command line interface. The operator is also provided with a list of options for different kind of comparisons that 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 is displayed.



**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).

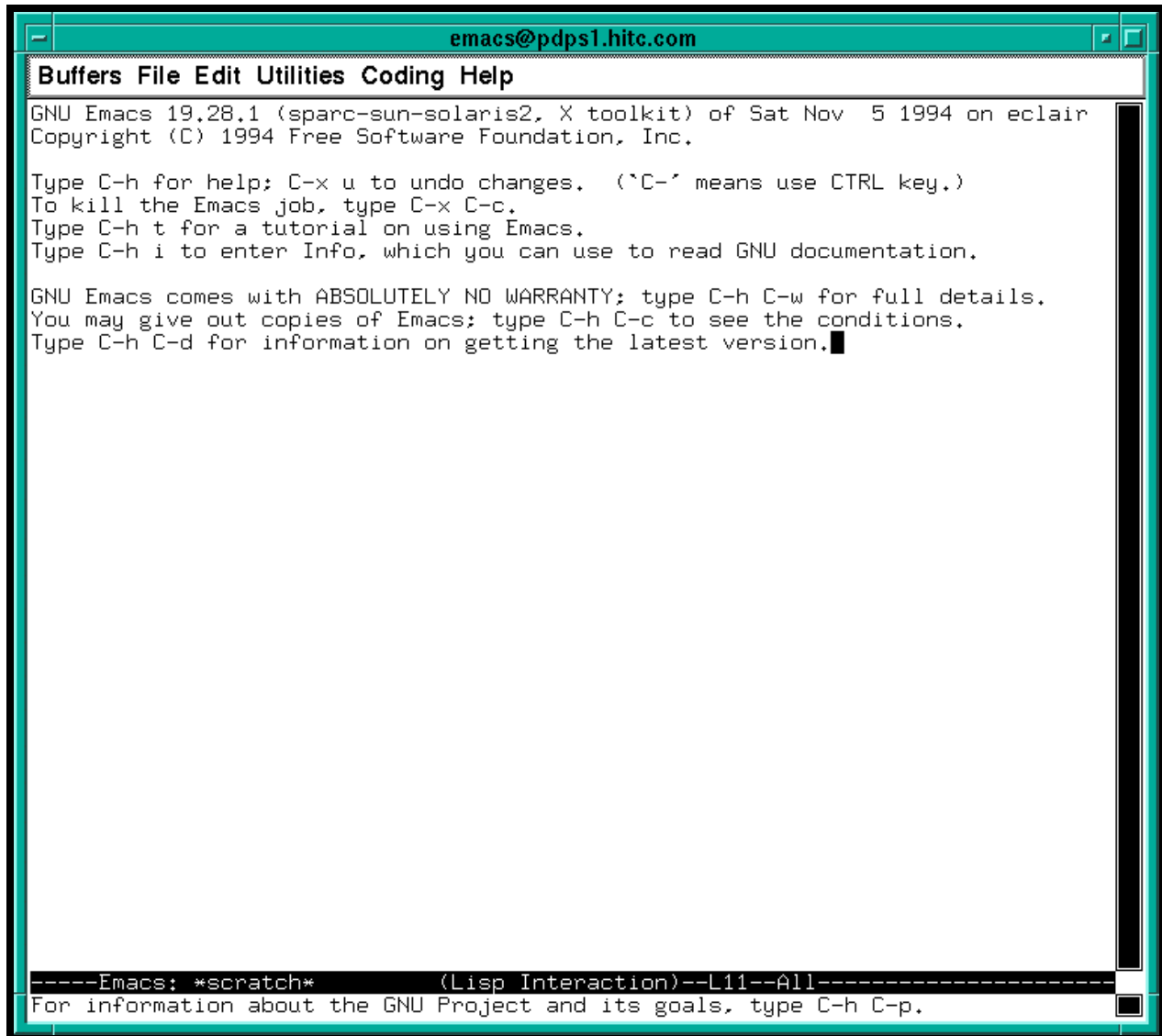


**Figure 4.5.1-25. SSIT Manager Main Screen with Text Editors tool menu displayed.**

#### 4.5.1.2.7.1 Emacs

Emacs is a COTS product. Emacs provides SSI&T users with Editor capabilities.

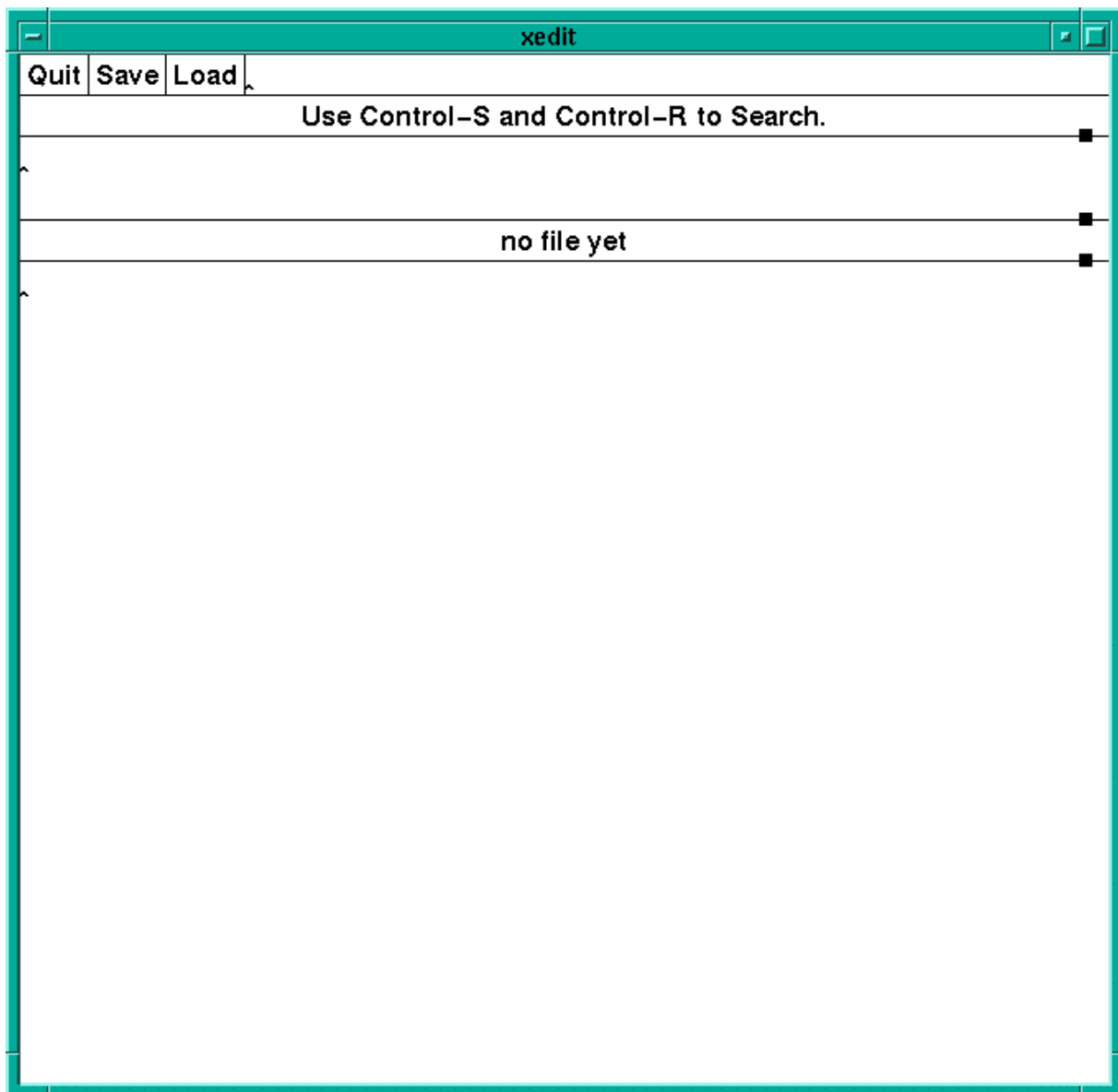
More information about Emacs is available on the Web free from <ftp://prep.ai.mit.edu/pub/gnu/>



**Figure 4.5.1-26. Emacs Pop-up.**

#### 4.5.1.2.7.2 Xedit

Xedit is a COTS product. Xedit provides SSI&T users with Editor capabilities.



**Figure 4.5.1-27. Xedit Pop-up.**

#### 4.5.1.2.8 PDPS Database

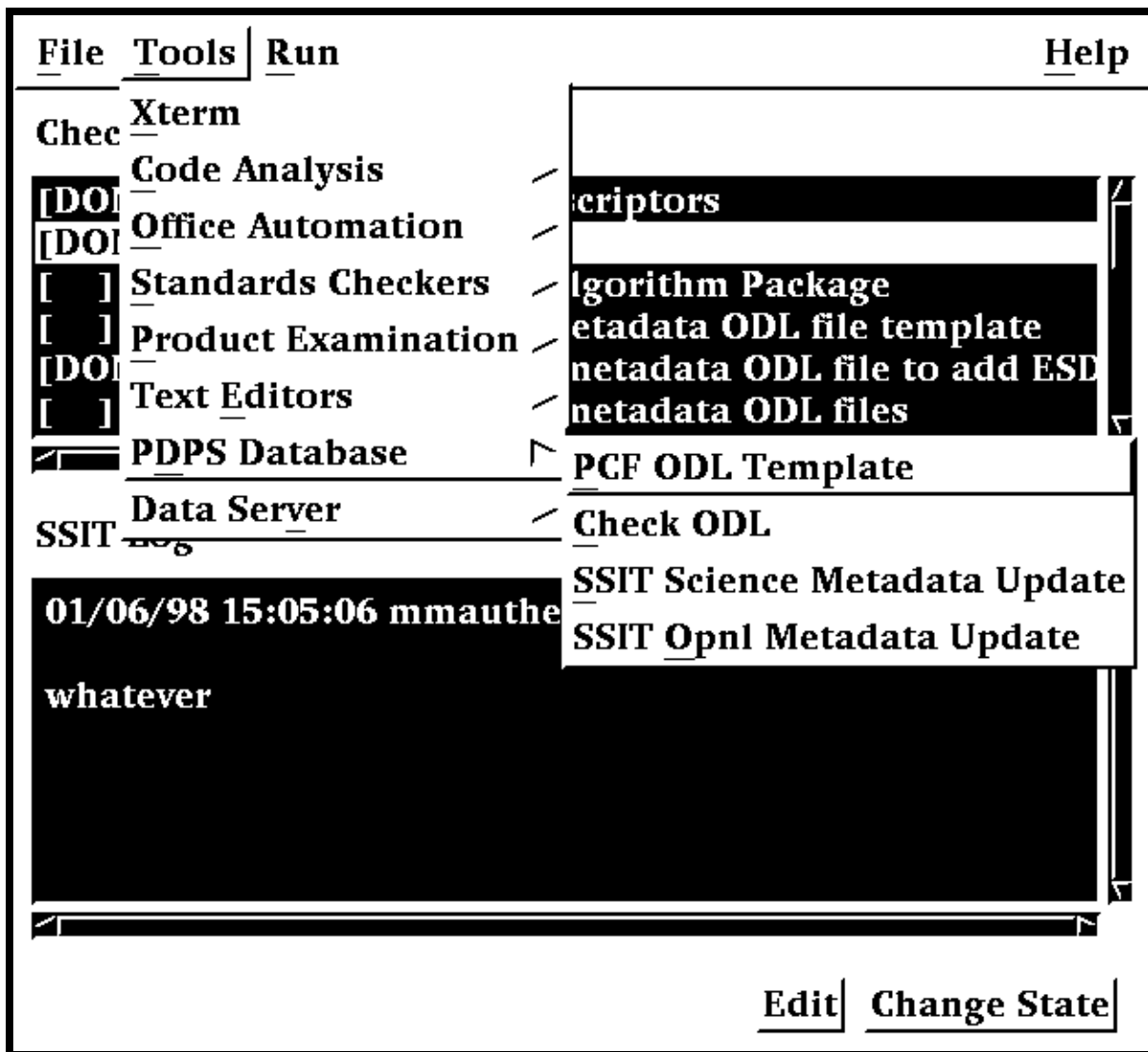
The PDPS Database Update Tools are accessed from the *TOOLS* menu item of the SSIT Manager Pop-up tool bar, submenu *PDPS Database*. These tools are used to register a PGE with the PDPS database, during which critical information about the PGE and under what conditions it should be



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 Operations Metadata Update (a.k.a. PGE Registration Pop-up).



**Figure 4.5.1-28. SSIT Manager with PDPS Database tool menu displayed**

All programs can be started from the *TOOLS/PDPS Database* menu available on the SSI&T Manager Tool as well 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 a 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 which is currently blank, as needed. In particular, each PCF file entry must be associated with one or more ESDTs.

The file edited through the previous step can be checked by the Check ODL Script, available through the SSIT Manager via *Tools/PDPS Database/Check ODL*.

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 aren't any errors) before the SSIT 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 user prompts and explanations follow. Note that if there is a default value for an entry (in most cases this will only occur if you run the program more than once) it will appear at the end of the prompt line.

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

Configuration filename? (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) that the tool will execute in. In most cases this will be 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 for which you want to check the ODL. 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 PGE Metadata ODL File help page for more information about naming and filling out the PGE ODL file.

Any success or error messages will then be displayed followed by:

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

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

Check ODL program output

The Check ODL program will search 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

User must look in the LogStatus file in the /usr/ecs/{MODE}/CUSTOM/logs directory or in the user-installed directory if the user has installed a personal copy of SSIT.

If the required ODL file(s) cannot be found the tool will report:

DpAtCheckOdlDpAtScienceMd::ProcessOdlFile (PGE):

Unable to open PGE Science Metadata ODL file

{FILENAME}

{FILENAME} will be replaced by the filename that the tool was searching for. If this message appears check the configuration settings (see Preconditions to running the Check ODL program) and that the ODL file exists.

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

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

```
NO ERRORS FOUND
```

#### 4.5.1.2.8.3 SSIT Science Metadata Update

The next step is to update the PDPS/SSIT database with science metadata, where the latter is defined as PDPS metadata that comes from the Instrument Teams (ITs), and rarely changes. This is done by selecting *SSIT Science Metadata Update* 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/SSIT 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 *SSIT Science Metadata Update* from the *PDPS Database* submenu to run the program. If a PGE already exists the operator will be prompted to update it. If a PGE does not already exist a character based window opens that 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.

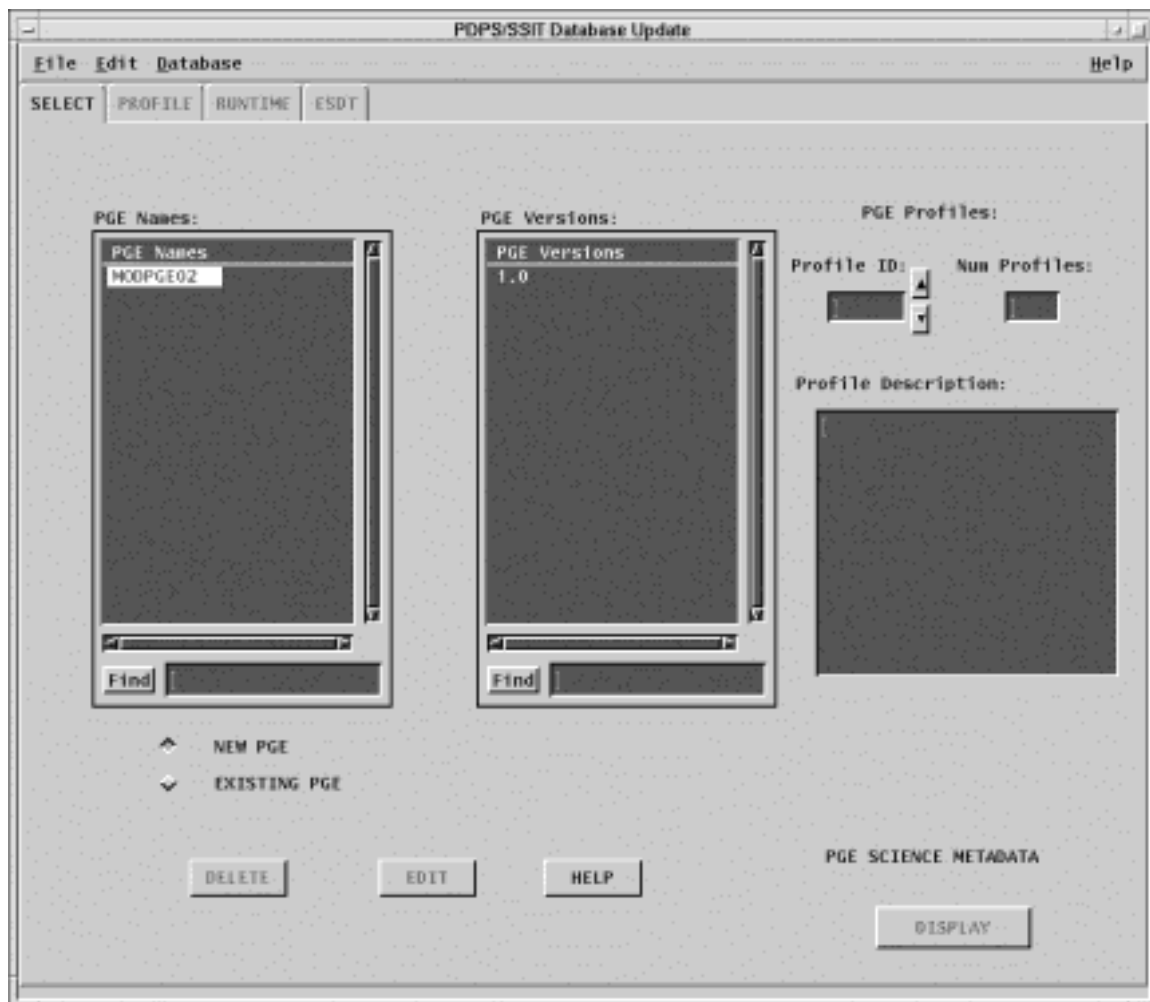
#### **4.5.1.2.8.4 SSIT Operational Metadata Update**

The SSIT 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/SSIT Database Operational Metadata Update Pop-up – showing the SELECT Tab**

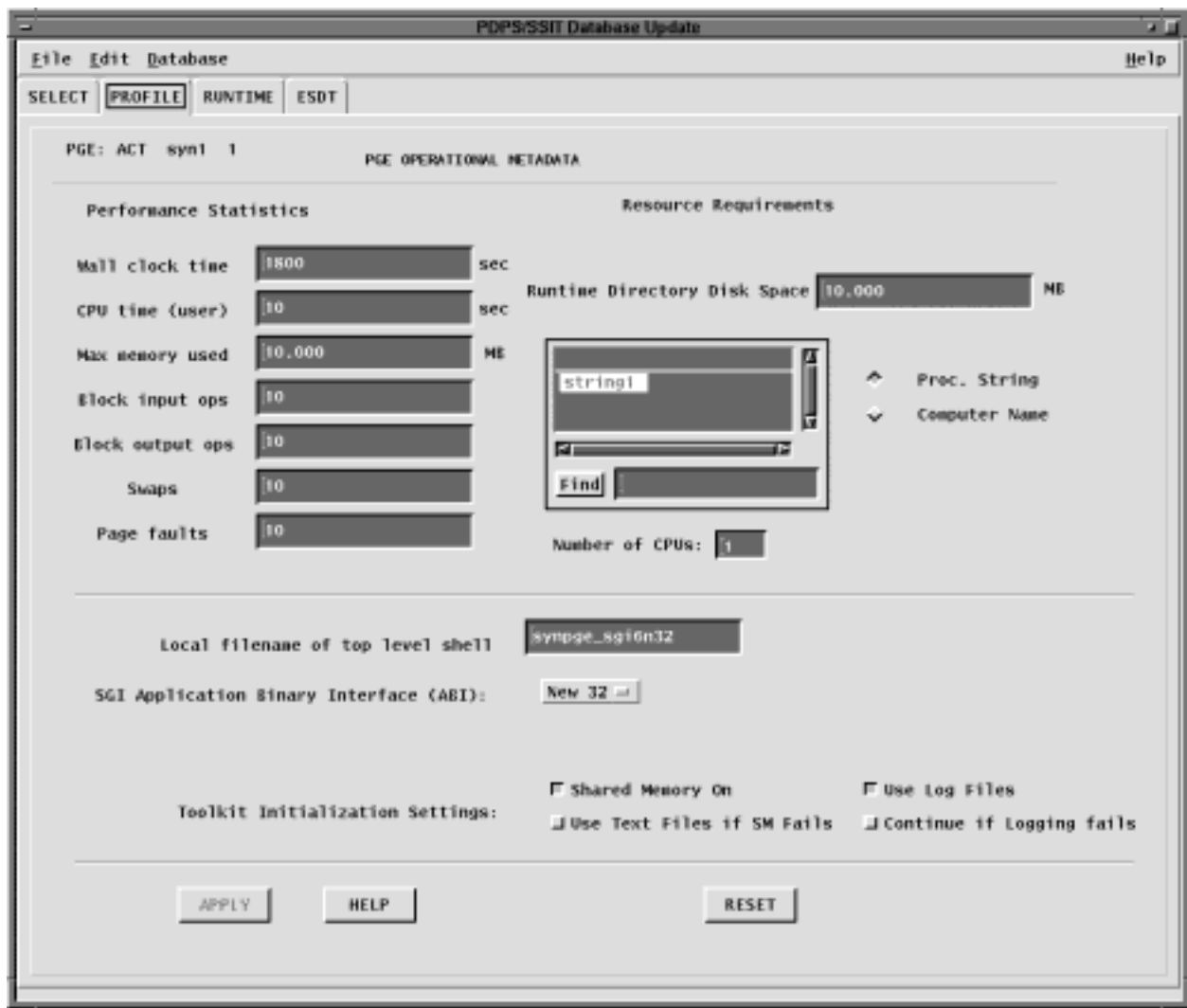
- The user may select to list either *New* or *Existing PGEs*, the difference being that new PGEs have not yet had their Operational metadata placed into the database.
- Selecting a PGE from the list populates the version list with the all the versions of the selected PGE presently defined in the PDPS database.
- Once the PGE has been selected, information on the PGE profile is also made available to the user through a window that shows a *Profile Description* 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.)

### **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 field (i.e. or 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 performing functionality associated with this tool. Clicking on *Reset* sets the fields to the current values in the database ignoring all the changes operated by the user during the current session. To record updated or new values in the database te 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/SSIT Database Operational Metadata Update Pop-up -showing the PROFILE Tab**



**Table 4.5.1-10. PDPS/SSIT Database Operational Update Profile  
Field Descriptions (1 of 2)**

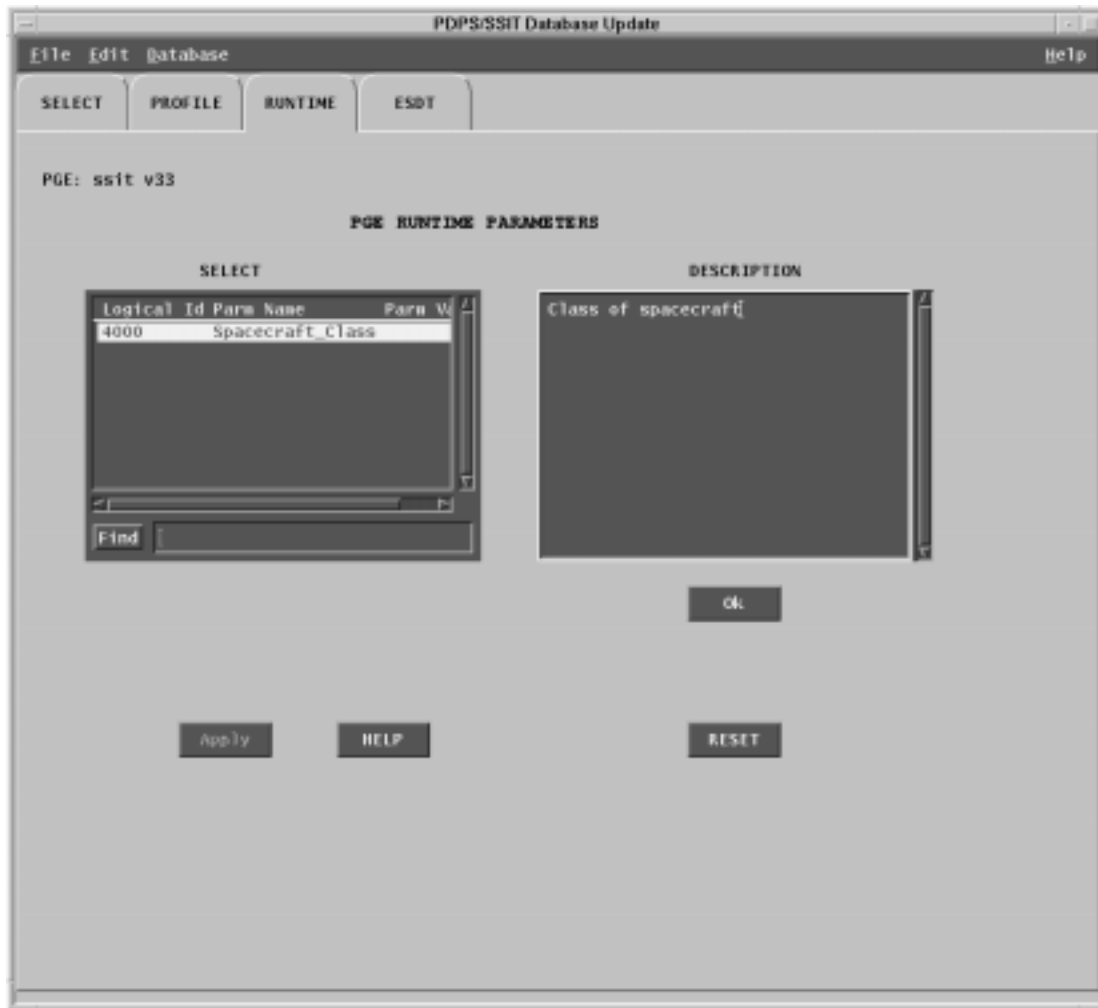
Field Name	Data Type	Size (of characters)	Entry	Description
Wall clock Time	float	N/A	User	real time used by the process (in seconds); filled in both after processing with SCF Toolkit and with DAAC Toolkit (i.e. with Autosys)
CPU Time	float	N/A	User	CPU time used by the process in seconds
Max Memory 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
Disk Space used in PGE run	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 will be 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	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 will be activated during PGE execution

**Table 4.5.1-10. PDPS/SSIT 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 will log information during PGE execution.
Toolkit Initialization Settings: Continue If Logging fails			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 which 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 may 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 parameters fields are shown in Table 4.5.1-11.



**Figure 4.5.1-31. PDPS/SSIT Database Operational Metadata Update Pop-up -showing the Runtime Tab**

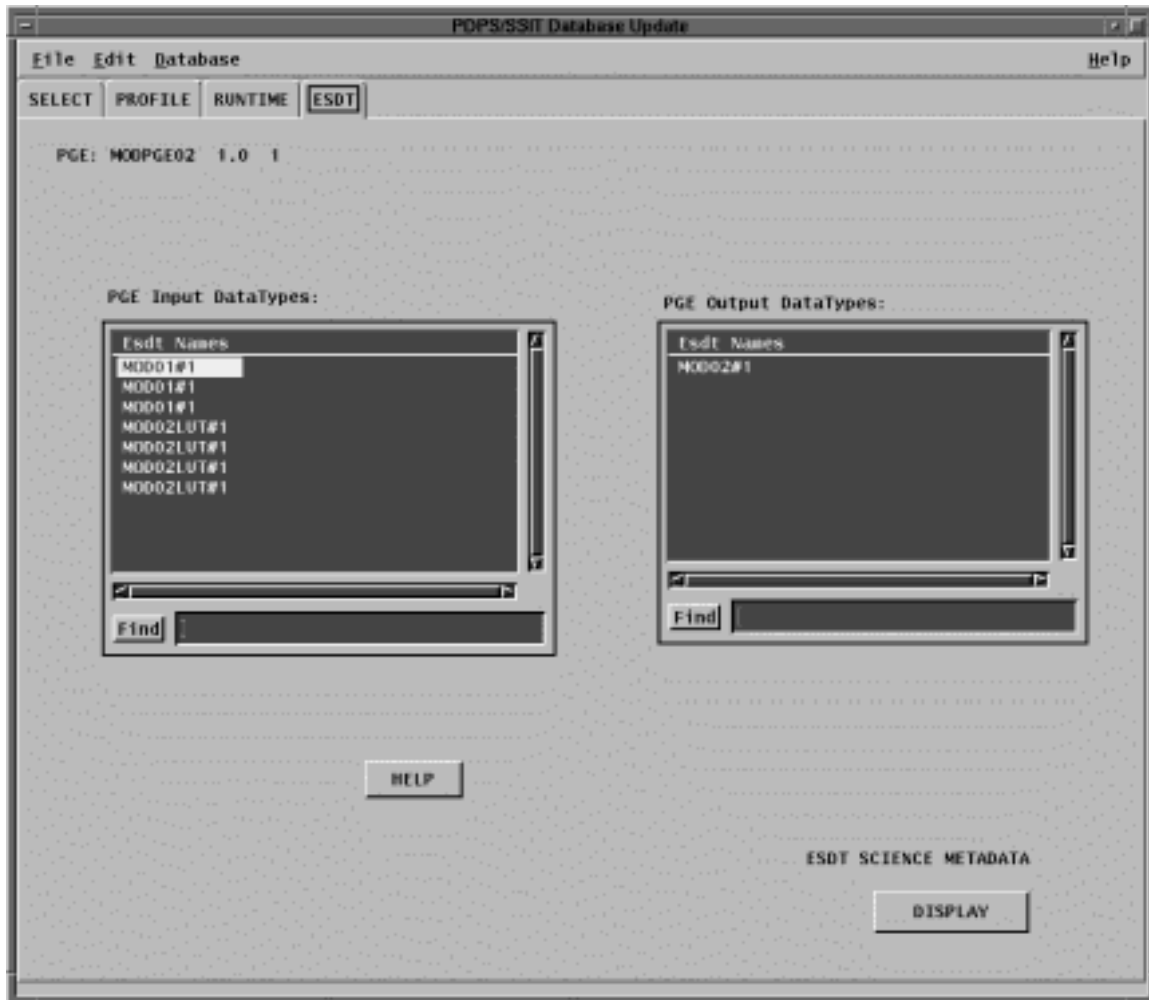
**Table 4.5.1-11. PDPS/SSIT Database Operational Metadata Update  
Runtime View Field Description**

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 Datatypes and the PGE Output Datatypes. On selection of a particular ESDT, the associated PDPS Science ESDT metadata can be viewed by clicking on the *Display* button. This data is read only and may not be modified through this Pop-up.

The program looks for the file it set in the configuration file parameter  
**DPAT\_ESDT\_SCIENCE\_MD.**

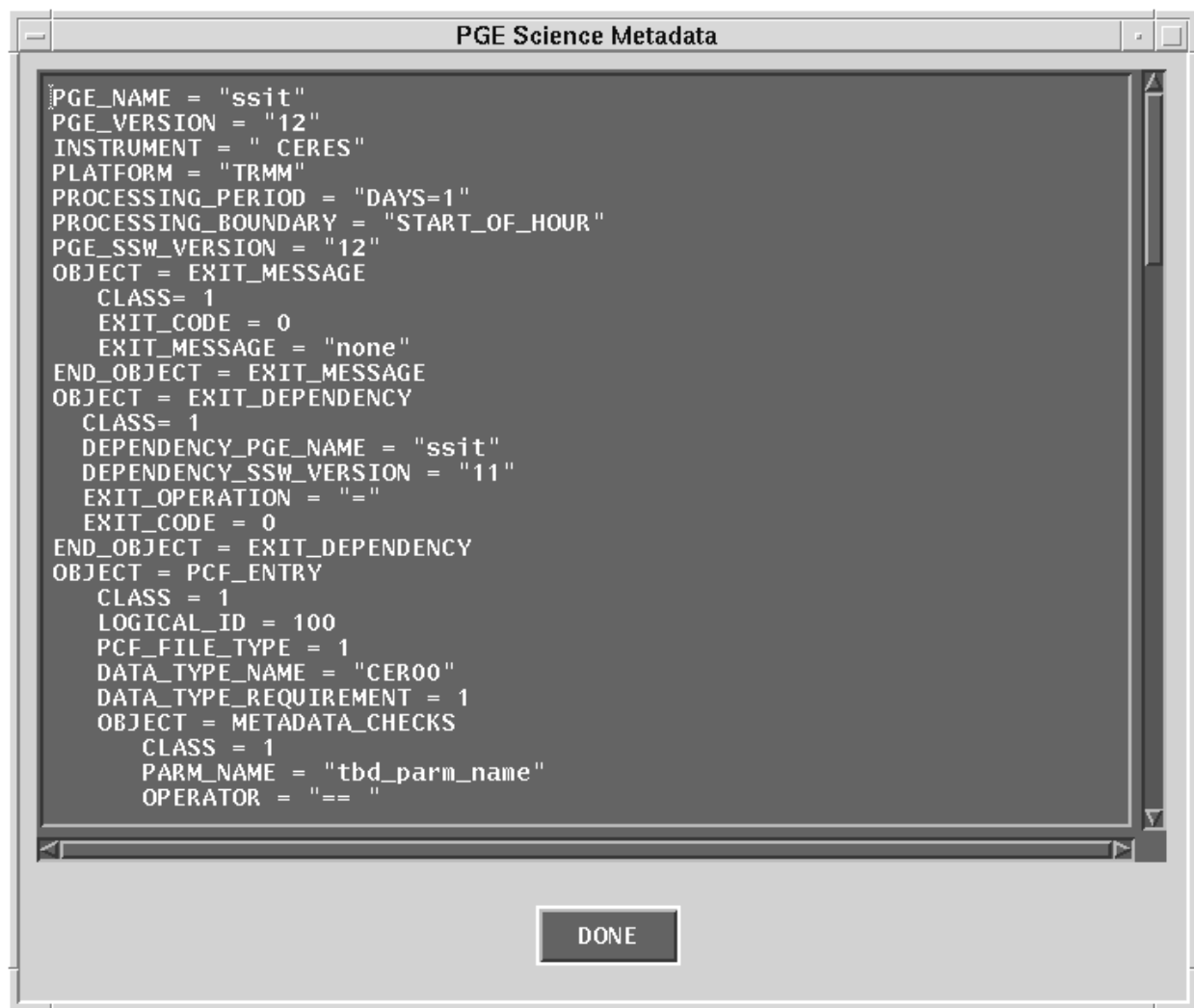


**Figure 4.5.1-32. PDPS/SSIT Database Operational Metadata Update Pop-up - showing the ESDT Tab**

## 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).

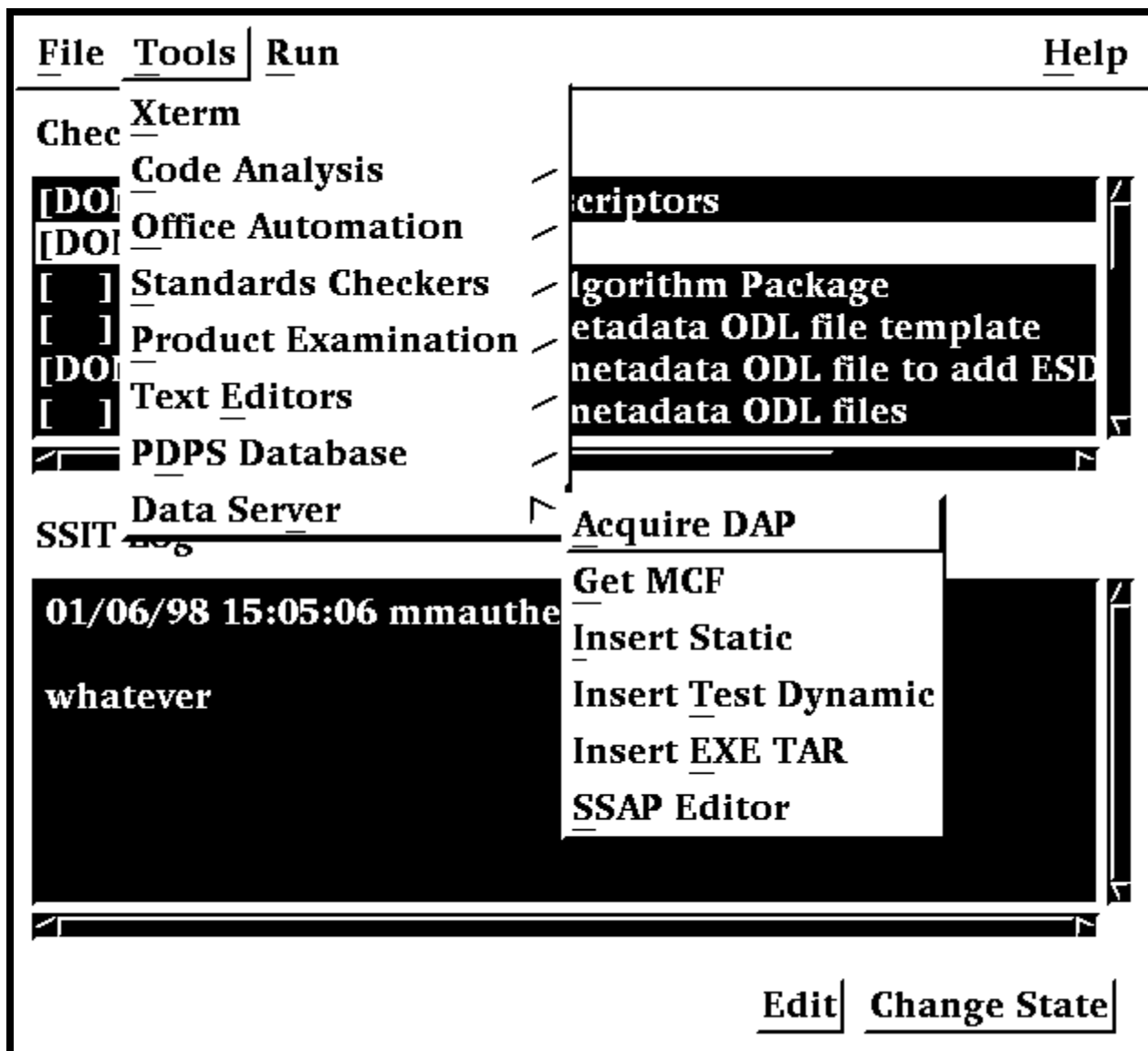
A Display screen pops-up also if the request for modification to the database fails because of some mandatory field not being properly set (see screen of *PROFILE* view). Editing is not allowed in this window. Clicking on "Done" closes this window.



**Figure 4.5.1-33. PDPS/SSIT 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 may be started from the TOOLS:Data Server menu. Figure 4.5.1-34 shows the submenu choices.



**Figure 4.5.1-34. SSIT Manager, Tools Menu, Data Server Submenu Choices**

#### **4.5.1.2.9.1 Acquire DAP**

This program is used to retrieve the Delivered Algorithm Package (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 will be 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 may be processed by Ingest. When this occurs, Ingest inserts the DAP into the Data Server, triggering subscription notification. Second, the DAP may be inserted with the Insert Test File.
3. After the DAAC operator received the email subscription notification, he or she must have saved 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 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 may be used to acquire any type of granule with a known UR.

After the file is staged, the operator may 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 may arrive at the DAAC through other means than Ingest, e.g., simple ftp.

#### **4.5.1.2.9.2 Get MCF**

This tool has a character based user interface. When the tool is selected from the Tools\Data Server\Get MCF menu, the user is prompted to provide pertinent information about the ESDT for the Metadata Configuration File that has to be acquired from the Data Server, including 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 tools 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 SSIT and in production. Static Files are files which rarely change 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. This UR is read by the Planning and Processing systems at runtime. so that 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 SSIT 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 that if there is a default value for an entry (in most cases this will only occur if you run the program more than once) it will appear 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 will be 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 that uses this static file.

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

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

Is there is 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 will 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 that it can be found by the tool.

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 that it can be found by the tool. It can be created by using the Get MCF tool to get a Metadata Control File which specifies the fields and type of data that is required for the ASCII Metadata file.

If it IS a Static Multi-File Granule (more than 1 data file) the following prompts will 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 are 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 will then be 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 that the file resides in the directory specified above. It can be created by using the Get MCF tool to get a Metadata Control File which specifies the fields and type of data that is required for the ASCII Metadata file.

Any success or error messages will then be 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 SSIT and in production. Dynamic Files are files which change at each instance of PGE processing; for example, Level 0 data files. This tool performs a function that is normally done by Ingest. It allows the user to insert dynamic files to the Data Server for testing purposes. Normally, Ingest will take in such files and insert 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) will 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 SSIT 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 that if there is a default value for an entry (in most cases this will only occur if you run the program more than once) it will appear 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 will be 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 will 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 that it can be found by the tool.

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 that it can be found by the tool. It can be created by using the Get MCF tool to get a Metadata Control File which specifies the fields and type of data that is required for the ASCII Metadata file.

If it IS a Dynamic Multi-File Granule (more than 1 data file) the following prompts will 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 are 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 will then be 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 that the file resides in the directory specified above. It can be created by using the Get MCF tool to get a Metadata Control File which specifies the fields and type of data that is required for the ASCII Metadata file.

Any success or error messages will then be 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 SSIT 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 SSIT 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 SSIT 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 that if there is a default value for an entry (in most cases this will only occur if you run the program more than once) it will appear 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 will be 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 will be 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 that it can be found by the tool.

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

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

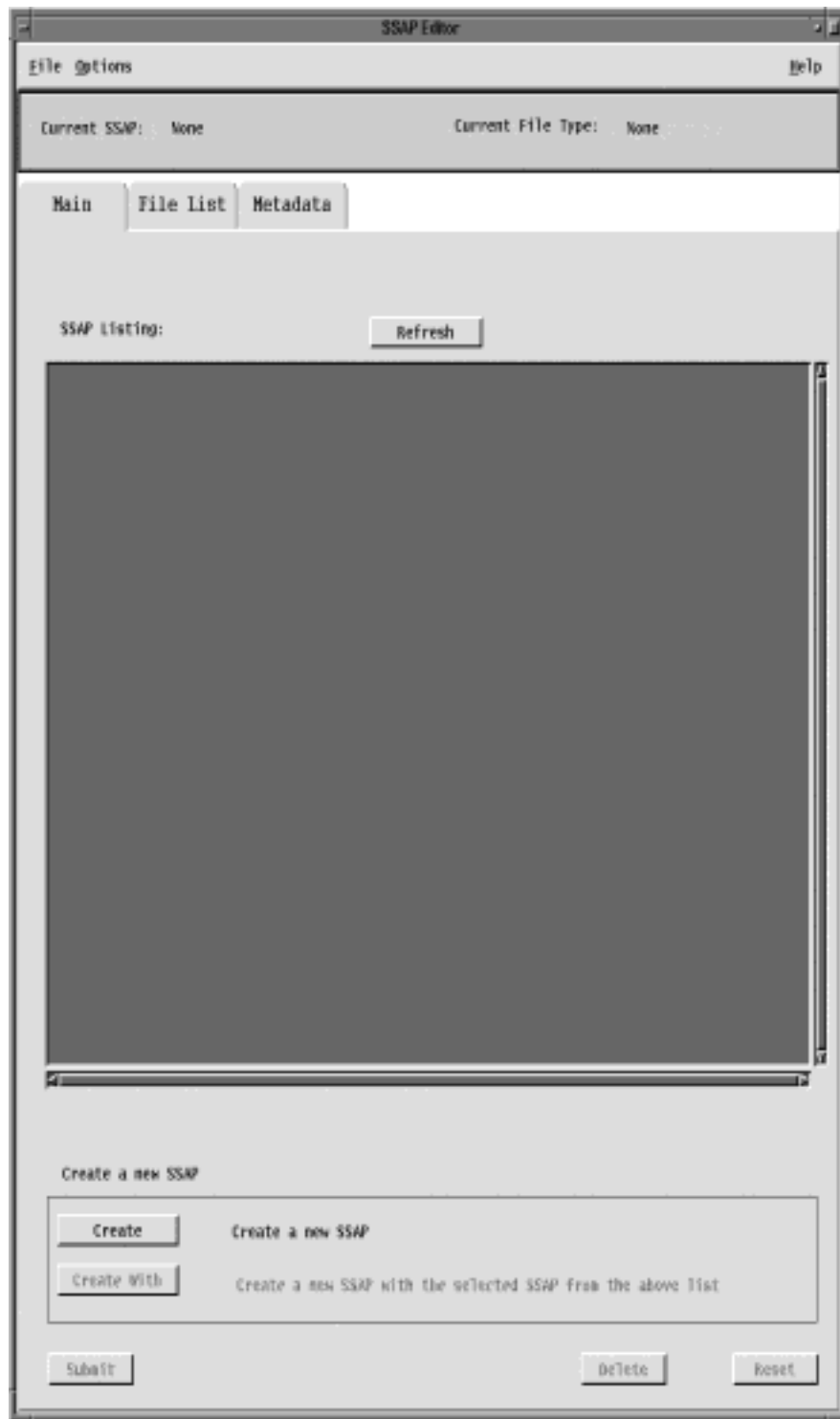
Top level shell filename within tar file?

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

#### **4.5.1.2.9.6 Science Software Archive Packages (SSAP) Editor**

The SSAP Editor 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 **Tools**, **Data Server**, and **SSAP Editor** to start the tool's main window, shown in Figure 4.5.1-35.

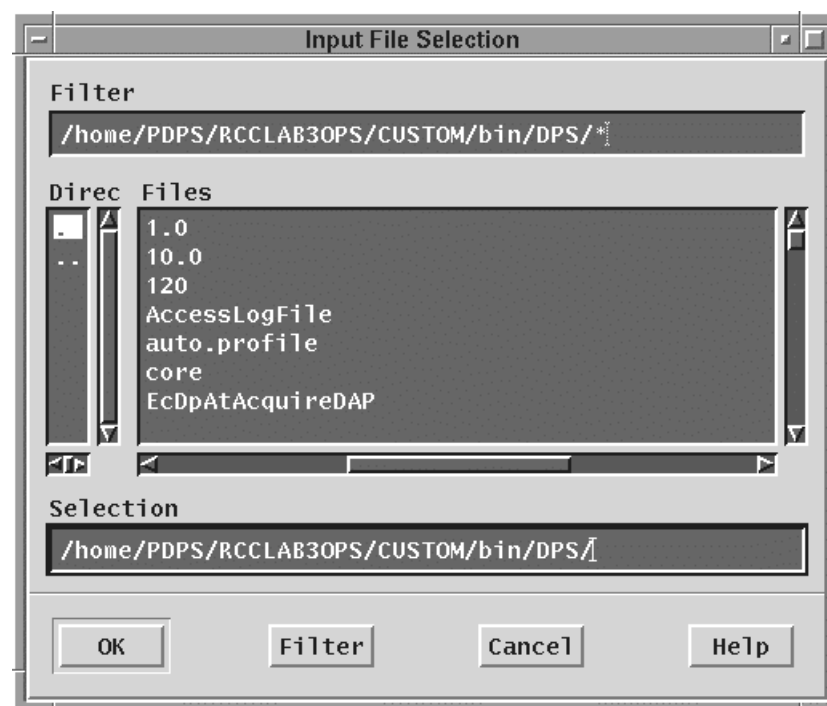


**Figure 4.5.1-35. SSAP Editor Main Pop-up**

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 will not allow the operator to come up in a mode that is not consistent with the one established at startup.

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 that shows 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.
  - **Batch Mode**; allows the operator to run a script including command lines for 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.

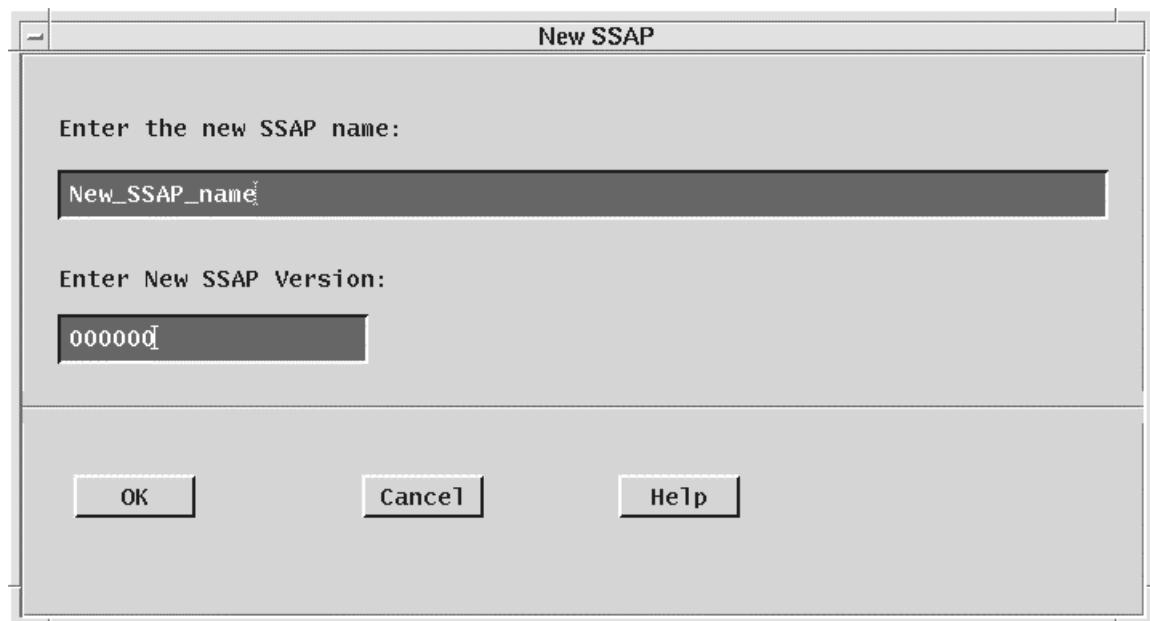
The SSAP Editor Pop-up main screen has three tabs: **Main**, **File List** and **Metadata**.

The following three subsections provide details on the views that are 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 that appears when the SSAP Editor is started. It contains the following controls:

- A free text window that lists all the existing SSAPs that are 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 that allows 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.



**Figure 4.5.1-37. New SSAP Window**

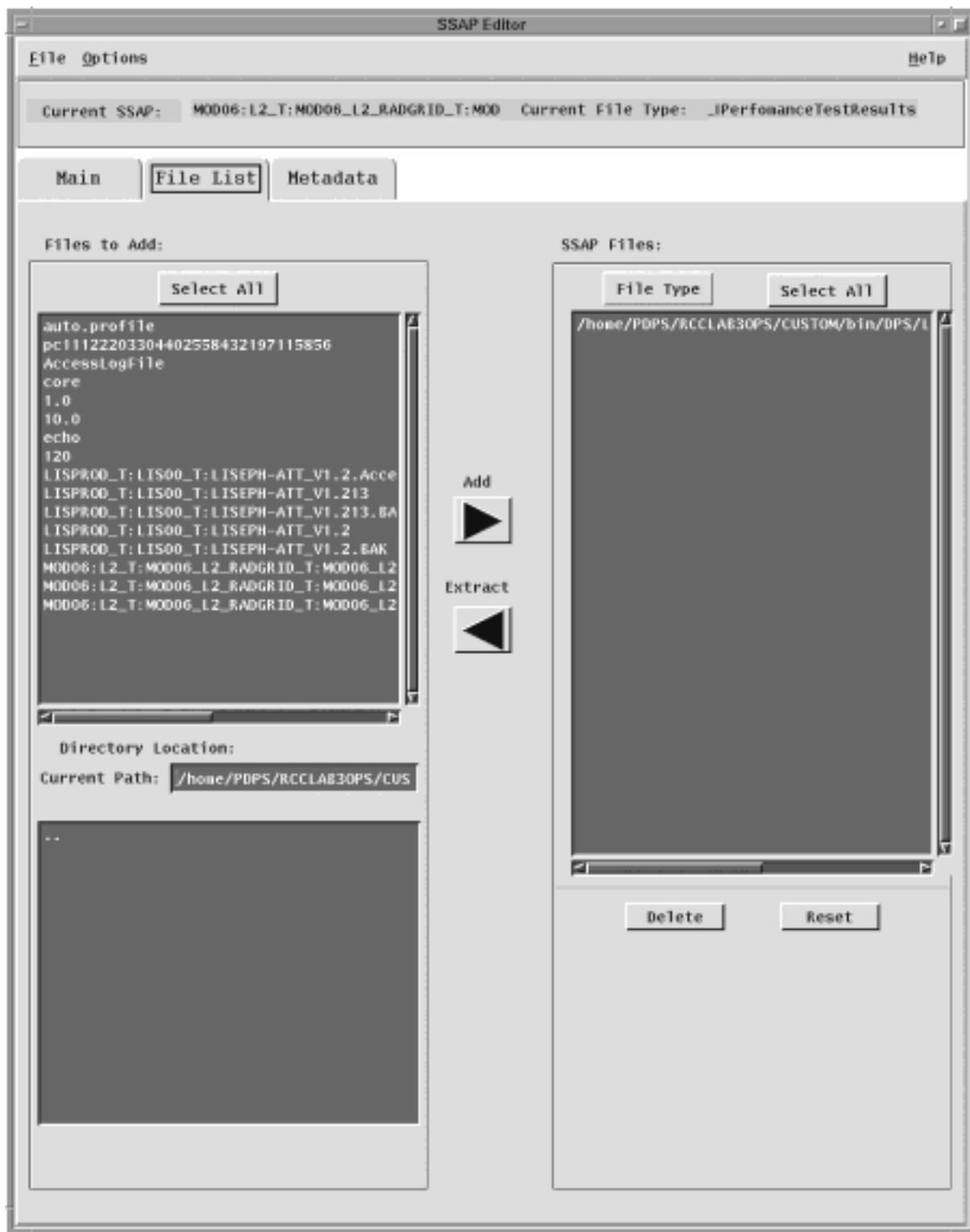
- A **Create With** button is provided as an alternative for the user to create a new SSAP starting with components of an existing SSAP file. To do this, the operator should click on an existing file, and change the parameters of the old SSAP with the name, SSAP version, and other

attributes of the new SSAP. The operator will complete the creation of the new SSAP with saving the changes to the new SSAP. Both the old and new SSAPs will 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.
- A **Delete** button is used by the operator for deleting a selected SSAP from the Data Server.
- 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 a screen, shown in Figure 4.5.1-38.



**Figure 4.5.1-38. File List 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. 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 Metadata files can be edited by the operator clicking 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.

SSAP Editor

File Options Help

Current SSAP: MOD06:I2\_T:MOD06\_I2\_RADGRID\_T:MOD Current File Type: \_IPerfoamanceTestResults

Main File List Metadata

Algorithm Information

Algorithm Package Name: Billy's Algorithm

Algorithm Package Version: 2.2.2

Maturity Code: Operational

Delivery Purpose: Because I want to

Acceptance Date: 01 / 02 / 1997 OK

PGE Information

PGE Name: The best PGE

PGE ID: 666

PGE Function: To cause problems

PGE Version: 1.1.1

SM Version: 1.1.1.1

PGE Date Last Modified: 12 / 03 / 1965 OK

SM Date Last Modified: 01 / 01 / 1997 OK

Edit Assoc Collections

Save Reset

**Figure 4.5.1-39. Metadata 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 that the operator is required to fill in are provided in the following Table 4.5.1-12

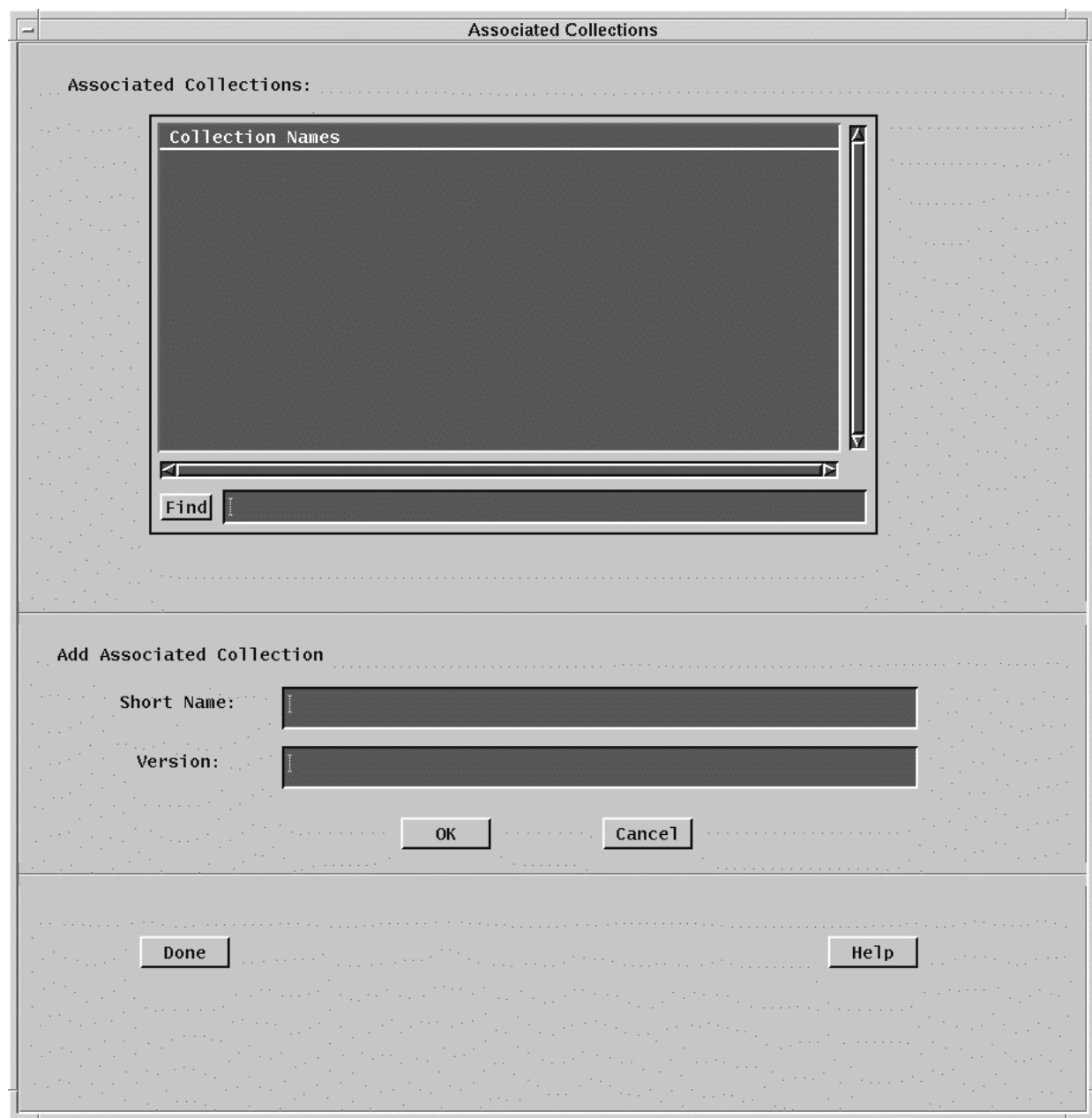
**Table 4.5.1-12. SSAP Editor - Metadata Tab Field Descriptions**

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 that was used to generate the SSAP
PGE ID	string	unlimited	user	ID of the original PGE that was 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 that was 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.

The **Edit Assoc Collection** button allows the operator to edit the ESDT's that 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.



**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 that the operator is requested to provide in this Pop-up are described in Table 4.5.1-13.

**Table 4.5.1-13. 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

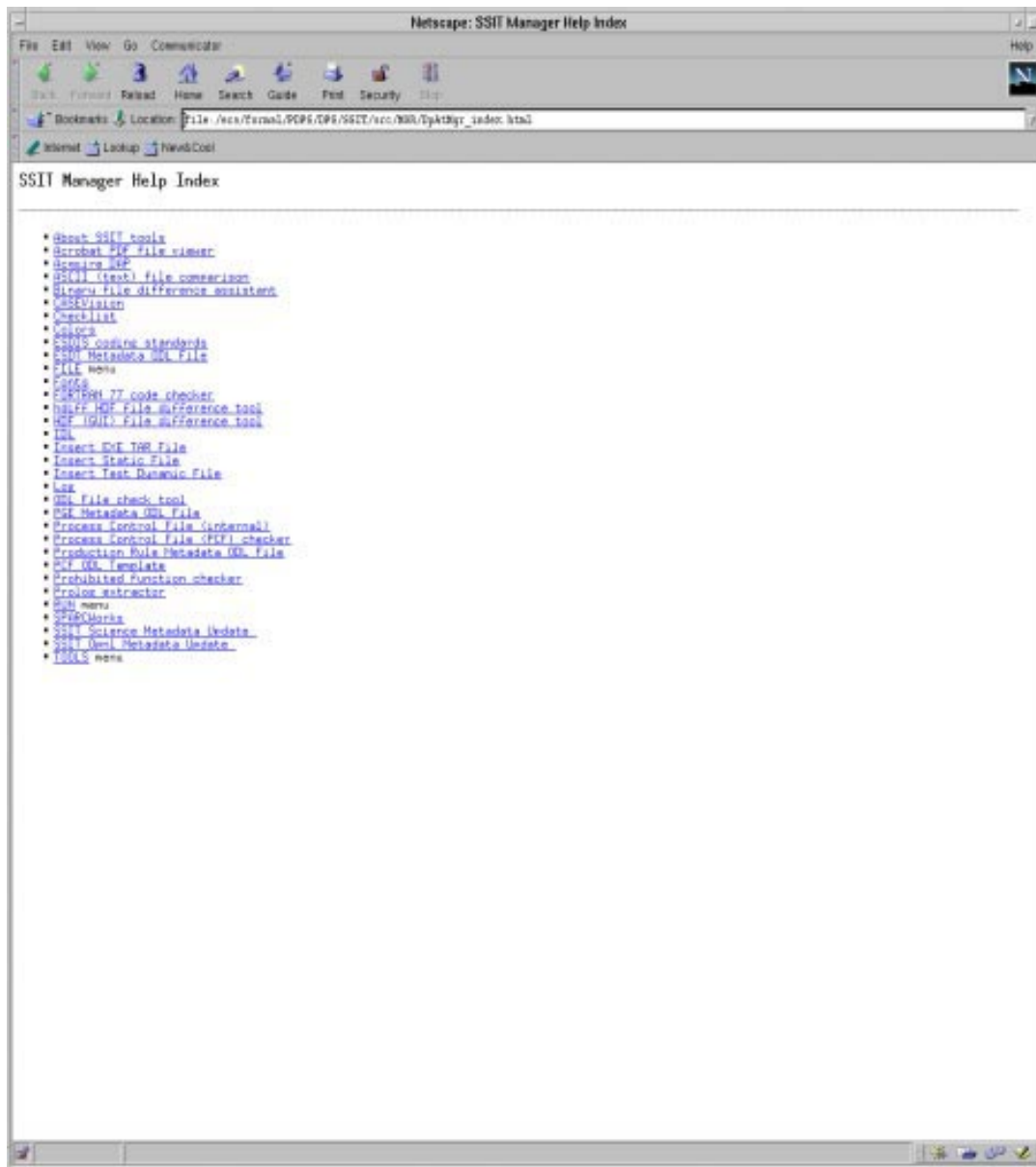
- Once the selection and/or the addition have been completed, the operator can confirm the changes by clicking on the **Done** button. Further context sensitive help is available to the operator by clicking on the **Help** button.
- Once **Done** is clicked, the operator is returned to the Metadata screen, 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 SSIT 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 #, etc.





**Figure 4.5.1-41. SSIT Manager Help Index**

### 4.5.1.3 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 all of the COTS packages used in SSIT, refer to the ECS Baseline Information System web page, URL <http://cmdm.east.hitc.com/>.

For Ecs Custom Software tools used in SSI&T 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 GSFC DAAC.

#### 4.5.1.3.1 Interfaces and Data Types

The SSIT Manager exchanges data of various types through interfaces within and external to ECS. Table 4.5.1-14 lists SSIT Manager system interfaces for Version 2.0.

**Table 4.5.1-14. SSIT 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 Data Bases/Files

The SSIT 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 will 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 that are involved with the SSIT Manager and the associated applications. These files are summarily described in the following table.

**Table 4.5.1-15. 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 SSIT software	
SSI&T Internal Process Control File	\$PGS_PC_INFO_FILE (a parameter in the configuration file)	<ul style="list-style-type: none"> <li>• SSIT Manager uses a <b>Process Control File</b> to determine names and locations of files</li> <li>• This Process Control File defines correspondence between specifications for actual files, and the logical identifiers used by the SSIT Manager to access those files</li> <li>• This SSI&amp;T PCF should not be confused with the PCF used by and included with the science software</li> </ul>	SSIT Manager
Process Framework Configuration	\$ECS_HOME/CUSTOM/cfg/DpAt*.CFG	Enable Process Framework for PDPS database access, ECS event logging	All SSIT programs
SSIT Manager Run Menu	\$ECS_HOME/CUSTOM/data/DPS/ssit_run_menu.txt	Add items to SSIT Manager RUN menu	SSIT Manager
SSIT Manager help files	\$ECS_HOME/CUSTOM/docs/WW/DPS/DpAtMgr*.html	Online help	All tools in SSIT Manager
Sample checklist	\$ECS_HOME/CUSTOM/data/DPS/checklist.sample	SSIT Manager Checklist	SSIT 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 4.5.1-15. SSI&T Files (2 of 2)**

<b>File Title</b>	<b>File Name</b>	<b>Purpose</b>	<b>Associated SSI&amp;T Application/Tool</b>
Xresources files	In \$ECS_HOME/CUSTOM/data/DPS: <ul style="list-style-type: none"><li>• XDpAtMgr</li><li>• Emacs</li><li>• Xdiff</li><li>• Xbadfunc</li><li>• DpAtMgrCheckHdfFile.defaults</li></ul>	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 SSIT 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 may 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 SSIT Manager displays errors in a popup message box (see Appendix A for additional details). The SSIT 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 the table below.

**Table 4.5.1-16. Reports (1 of 2)**

Report Type	Report Description	When and Why Used
Algorithm Integration & Test Management Reports	<ul style="list-style-type: none"> <li>• provide the operations staff information on Algorithm Integration and Test application software events which have occurred</li> <li>• This information is available from the MSS database</li> </ul>	
	<ul style="list-style-type: none"> <li>• Ad-hoc reports include: Production Requests and associated Data Processing Requests, Data Subscriptions, PGE Profiles, etc.</li> <li>• This information is available from the PDPS database</li> </ul>	These reports can be used to track modifications and provide historical information on these data objects
PGE Profile Reports	<ul style="list-style-type: none"> <li>• 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).</li> <li>• A profile will be captured for each type of machine, i.e., Sun, SGI, etc., for which the PGE is to execute</li> <li>• Statistics are collected to establish standard deviations, variances, and averages of resource profile values</li> </ul>	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

**Table 4.5.1-16. Reports (2 of 2)**

<b>Report Type</b>	<b>Report Description</b>	<b>When and Why Used</b>
I & T Activity Report	capture information about the activities which have occurred and activities which are occurring in the Algorithm Integration and Test environment	
PGE Profile Update Report	capture information to track the updates which have occurred in the PDPS Database to PGE Profiles	
PGE I&T Reports	<ul style="list-style-type: none"><li>• capture information on PGEs as they progress through the AI&amp;T process.</li><li>• These reports include:<ul style="list-style-type: none"><li>– Code Analysis Report</li><li>– Standards Checker Report</li><li>– File Comparison Reports</li><li>– AI &amp; T Discrepancy Reports</li><li>– Inspection Reports</li><li>– Integration Reports</li><li>– Acceptance Reports</li></ul></li></ul>	These reports are to trouble shoot problems and will 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 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 also stores and manages request information, and provides for data preprocessing storage and insertion into the appropriate ECS storage location.

This page intentionally left blank



## 4.6.1 Data Ingest GUI

The Data Ingest Tool allows the Operator to 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 functions.

**Table 4.6.1-1. Operator Ingest Functions**

Operating Function	GUI	Description	When and Why to Use
Ingest History Log Viewing	History Log Tab	<ul style="list-style-type: none"><li>• Displays a log that contains the results of past ECS ingest requests.</li><li>• Specifies search criteria (e.g., time range), the provider ID, data set name, and request status of the Ingest History Log for log display.</li></ul>	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"><li>• Monitors the status of all or a subset of the ingest requests that are in progress.</li></ul>	As required to verify nominal system operations
Operator Request Update	Monitor/Control Tab	<ul style="list-style-type: none"><li>• Updates (i.e., change priority, cancel, suspend, and resume) an on-going ingest request. (Future release)</li></ul>	As needed to cancel a problem request; suspend, resume or change priority of an on-going request
Hard Media Ingest	Media Ingest Tab	<ul style="list-style-type: none"><li>• Operator performs hard media ingest (e.g., 8mm tape)</li><li>• Media Ingest Session is configured on the Operator's GUI to accept the request and submit it to the ECS system</li></ul>	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"><li>• Operator updates parameter values for each external data provider</li><li>• Value includes volume threshold and request threshold</li></ul>	As needed to tune the system to current or expected daily activity thresholds on an external data provider basis.

<sup>1</sup>Individual 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

This section presents an orientation of the Ingest GUIs and how to quickly get into the Ingest system.

#### 4.6.1.1.1 Invoking ECS Data Ingest Tool from the Command Line

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

**EcInGUIStart** <mode>

<mode> is the ECS mode for the execution.

#### 4.6.1.2 ECS Data Ingest Main Screen

The Ingest Main screen, with the "Welcome to ECS Ingest GUI Interface" screen, is displayed (see Figure 4.6.1-2). The major functions available through tabs from 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).

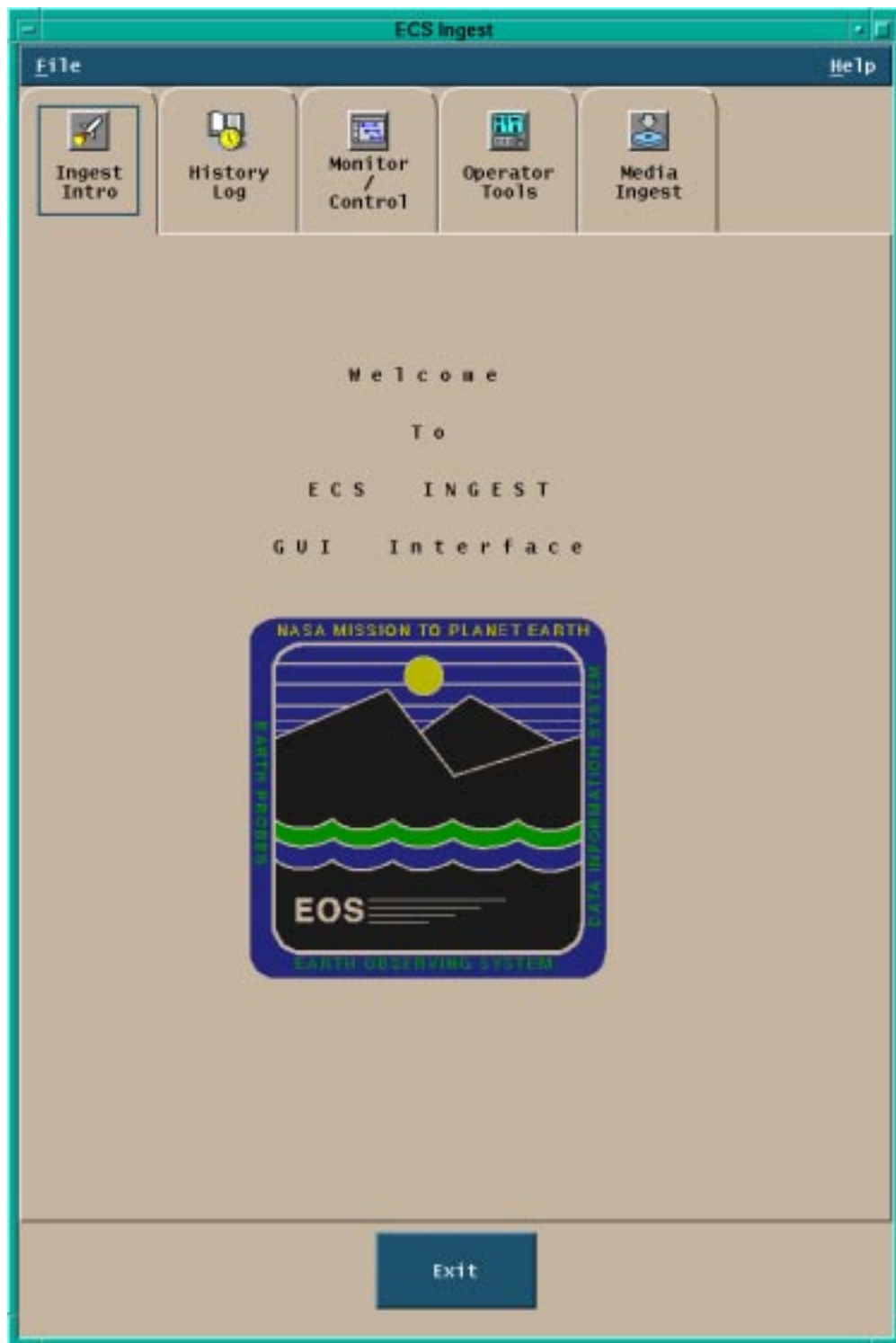
**"Monitor/Control"** provides operations personnel the capability to monitor and update ongoing ingest activities in the system (see Section 4.6.1.2.2).

**"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 MenuBar 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 4.6.1-2. Ingest Main Screen shown in the Ingest Intro Tab**

#### 4.6.1.2.1 History Log Tab

The History Log provides operations personnel 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 (see Figure 4.6.1-3):

- **Ingest 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 will be 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 4.6.1-3. Ingest History Log Tab**

For a description of the “History Log” tab fields, see Table 4.6.1-2 below.

**Table 4.6.1-2. History Log Field Descriptions**

<b>Field Name</b>	<b>Data Type</b>	<b>Size</b>	<b>Entry</b>	<b>Description</b>
Start Date/Time	Date/Time	4	Operator Selected (Default = Current Date/Time - 24 hrs.)	Entry fields for the Start Date and Time
Stop Date/Time	Date/Time	4	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**”, that allow to toggle between the two possible optional types of report.
- The button “**Clear All**” that clears the current screen.

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.

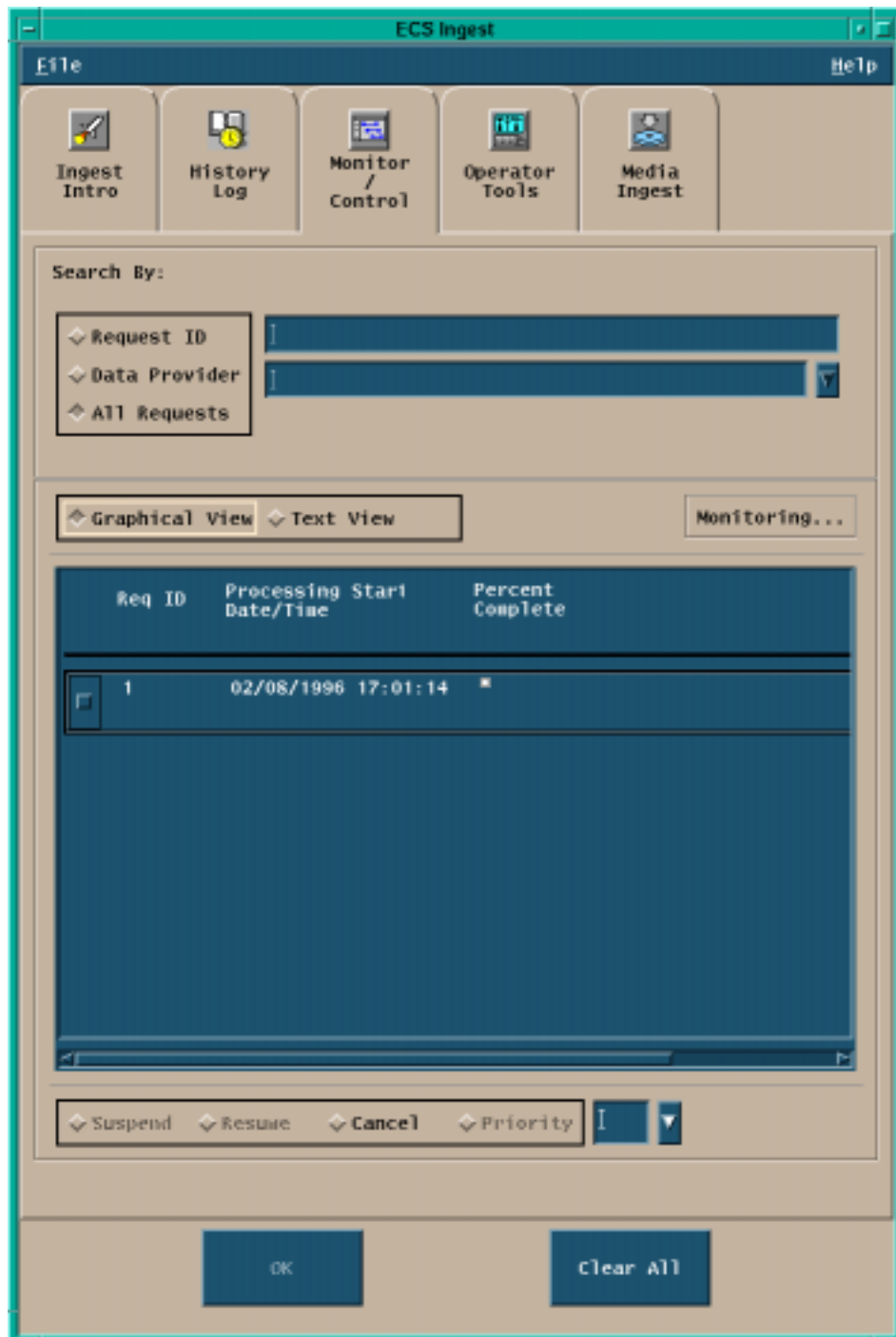
#### **4.6.1.2.2 Monitor/Control Tab**

Clicking on the Monitor/Control tab brings up the Ingest Status Monitoring/Request Control tab (see Figure 4.6.1-4). This tab provides operations personnel the capability to monitor ongoing ingest activities in the system and can alert the operator if a request has passed the requested expiration date.

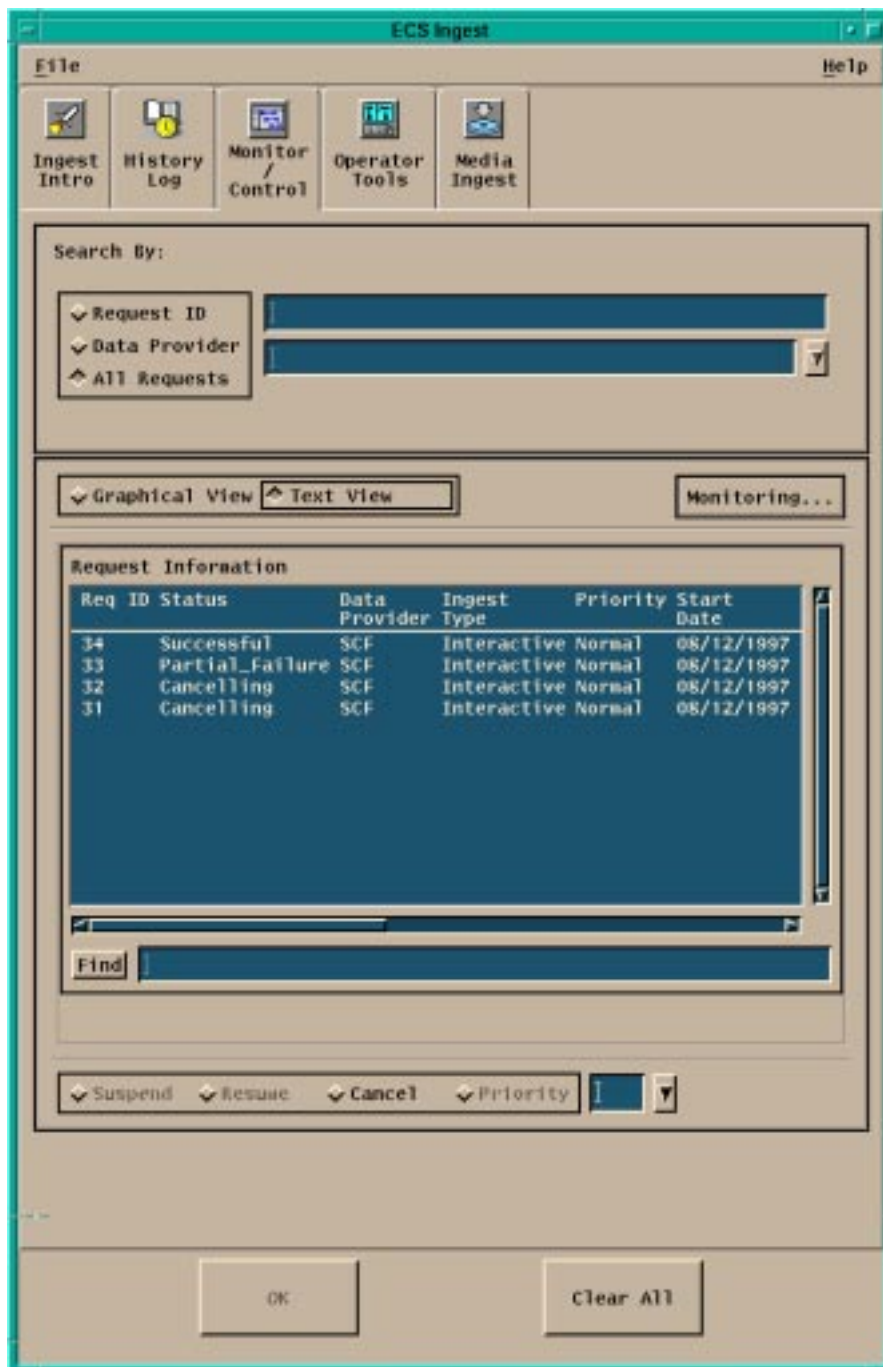
The Ingest Monitor/Control tab is divided into four regions:

- 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 graphical view for the request information. If the test view button is selected, the text view region is shown as in Figure 4.6.1-5.
- 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 the capability to update ongoing ingest activities in the system. Update services include **suspend**, **resume**, **cancel**, and **change priority**. These services are for a future release.



**Figure 4.6.1-4. Ingest Monitor/Control Tab**



**Figure 4.6.1-5. Ingest Monitor/Control Tab (Text View)**



Table 4.6.1-3 describes the fields available in 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...**” switch button that allows to select the 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 delete one or more items from the list of request;
- The pull down list “**Data Provider**”, which allows the operator to select from a list of valid external data providers;
- The “**All Requests**” toggle button that displays all requests currently in the system;
- The “**Suspend**” radio button that suspends the processing of an ongoing ingest request (future release);
- The “**Resume**” radio button that resumes the processing of a previously suspended ingest request (future release);
- The “**Cancel**” radio button that aborts the processing of an ongoing ingest request (future release);
- The “**Priority**” radio button - that is used in conjunction with a pulldown list of allowed priority levels - allows the operator to assign a priority order to an ingest request (future release).

#### **4.6.1.2.3 Operator Tools Tab**

Clicking on the Operator Tools tab brings up the Operator Tools window (see Figure 4.6.1-6). 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 will have initial values at the time of delivery.

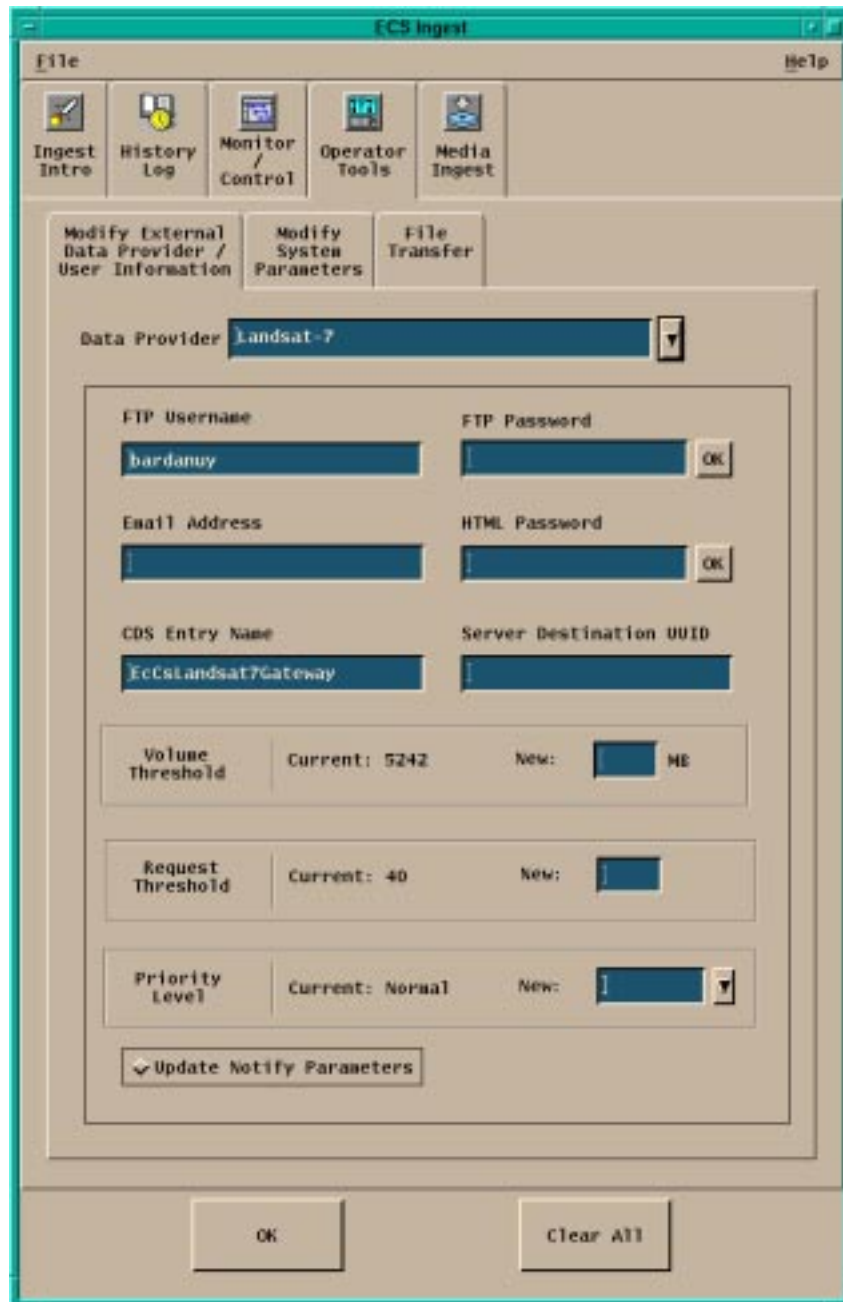
The Operator Tools Tab is divided into three sub-tabs:

- **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 4.6.1-6)
- **Modify System Parameters sub-tab** allows setup of thresholds and system information on a system-wide basis (see Figure 4.6.1-8)
- **File Transfer sub-tab** allows setup of external node information (see Figure 4.6.1-9)

See Tables 4.6.1-4, 4.6.1-6, 4.6.1-7 below for a description of fields that appear, respectively, in the 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**” pull down menus allow, respectively, for exiting capability and displaying of context sensitive help information.



**Figure 4.6.1-6. 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"> <li>• Provides an input entry field for FTP Password</li> <li>• The FTP Password is NOT reflected to the screen when the operator inputs information (and is encrypted prior to storage)</li> </ul>
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"> <li>• HTML password for authorized science users</li> <li>• The HTML password is NOT reflected to the screen when the operator inputs information (and is encrypted prior to storage)</li> </ul>
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 (see Figure 4.6.1-7). 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.

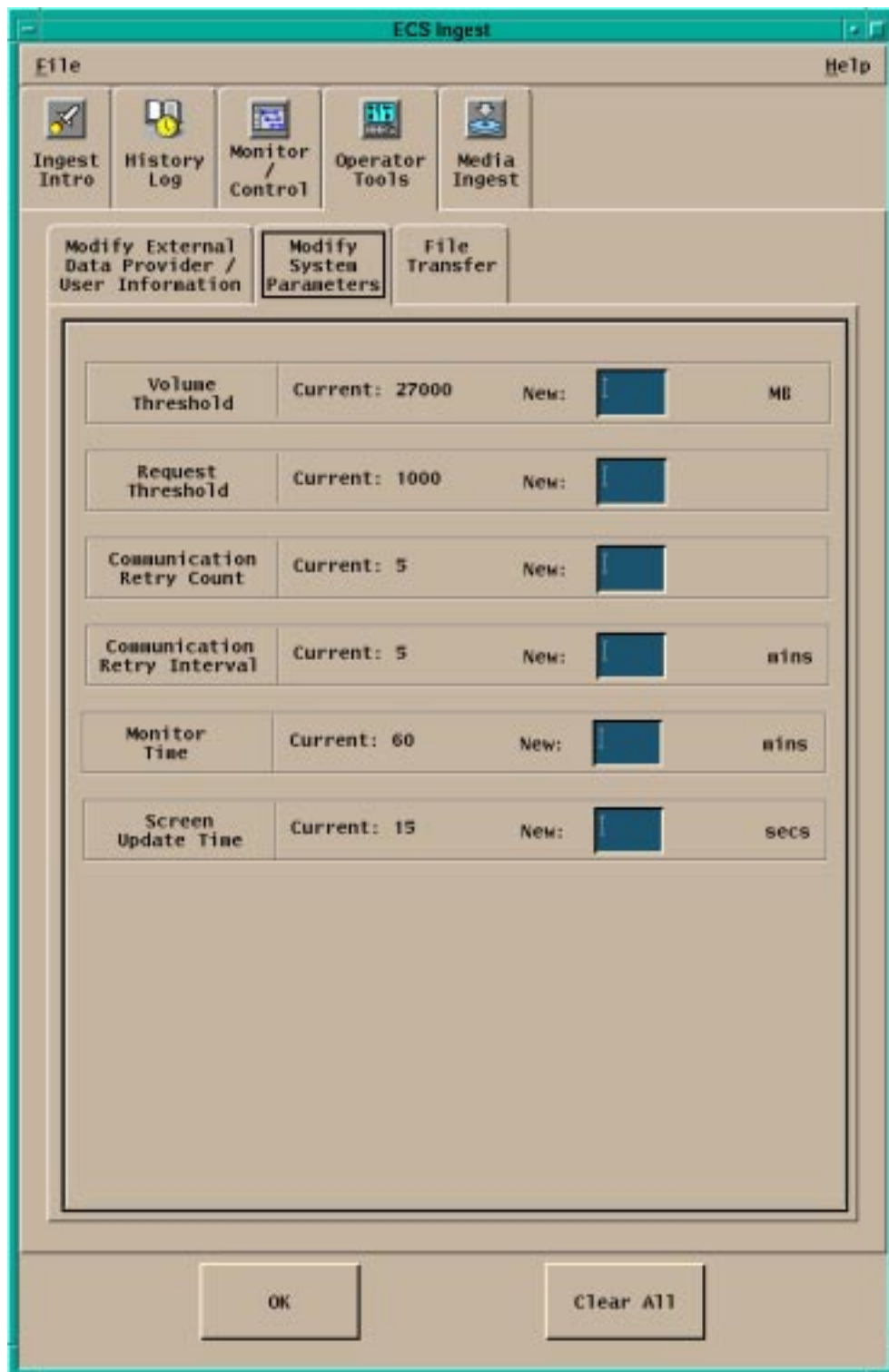
The image shows a 'Notify Parameters' dialog box. It contains the following fields and controls:

- Notify Type:** A text box containing the word 'Buffer'.
- Notify FTP Node:** An empty text box.
- Notify FTP Directory:** An empty text box.
- Notify FTP Username:** An empty text box.
- Notify FTP Password:** An empty text box.
- Buttons:** There is an 'OK' button next to the password field and a larger 'OK' button at the bottom left. A 'Cancel' button is located at the bottom right.

**Figure 4.6.1-7. Update Notify Parameters Pop-up**

**Table 4.6.1-5. Update Notify Parameters Field Descriptions**

Field Name	Data Type	Size	Entry	Description
Notify Type	Text Field	10	Operator Selected (Default = current value or blank, if no current value exists)	Provides Notify Type for the given data provider
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"><li>• Provides an input entry field for Notify FTP Password</li><li>• The Notify FTP Password is NOT reflected to the screen when the operator inputs information (and is encrypted prior to storage)</li></ul>



**Figure 4.6.1-8. Operator Tools - 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"> <li>• Provides the current amount of time (in minutes) that a completed request is retained on the monitoring display prior to removal.</li> <li>• Provides an input area to set a new monitor time</li> </ul>
Screen Update Time	Integer	N/A	Operator Selected (Default = current value)	<ul style="list-style-type: none"> <li>• Provides the current system setting for screen update interval in seconds and an input area to set a new value</li> </ul>

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





**Figure 4.6.1-9. Operator Tools - 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 to specify if the operation objective is to perform a **Generic File Transfer** or to **Build SMC History Files**. With the **OK** button the operator initiate the desired file transfer.

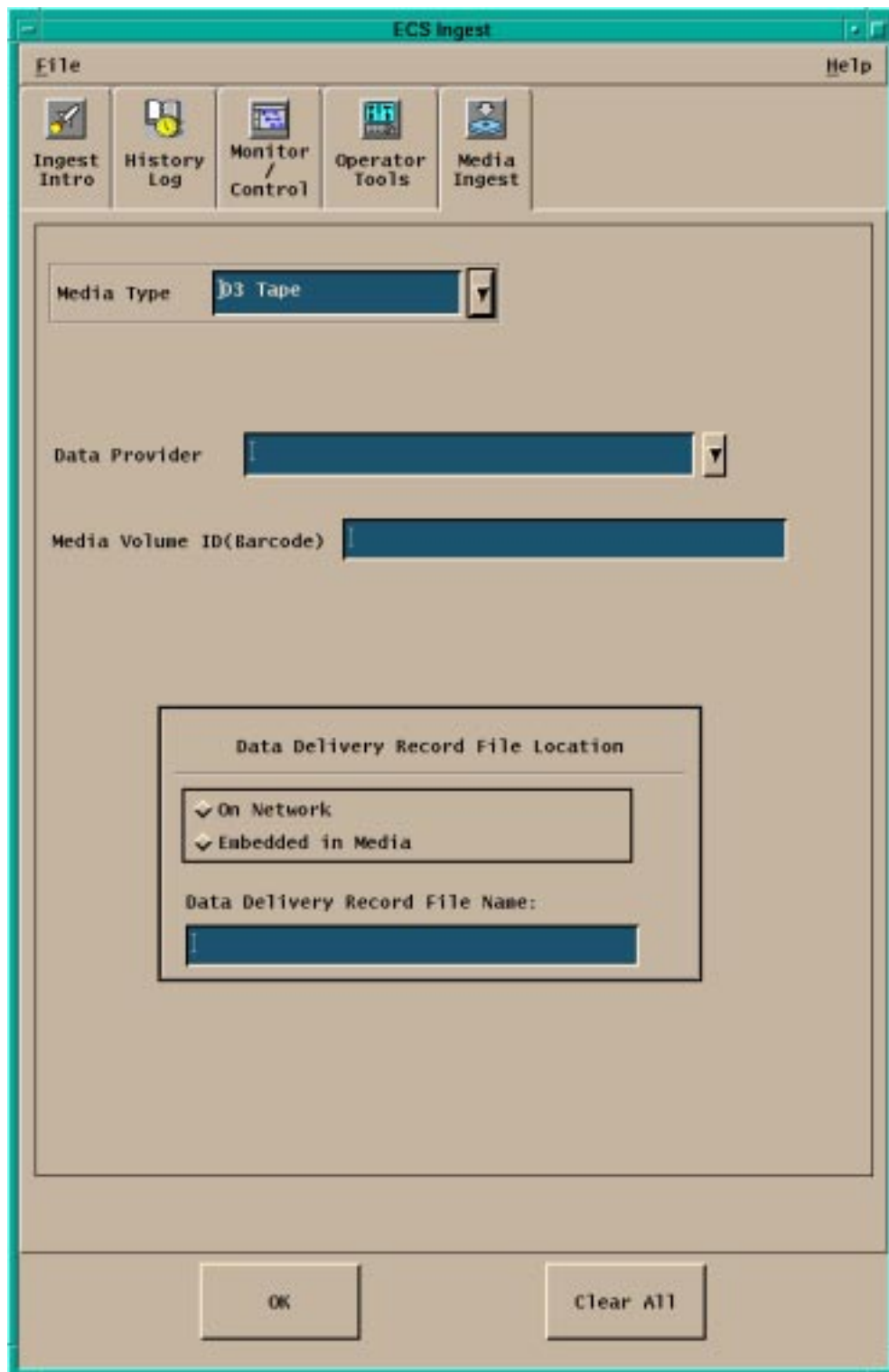
#### 4.6.1.2.4 Media Ingest Tab

Clicking on the Media Ingest tab brings up the Ingest Media window, shown in Figure 4.6.1-10. This tab provides DAAC operations personnel the capability to perform media ingest (e.g., specifying media for ingest, such as 8 mm tape).

The Ingest Media Ingest Tab is divided into three regions:

- Media Type – allows the operator to select three types of media (8mm tape, 4mm tape, and D3 tape)
- Data Provider and Media Volume ID – allows the operator to specify media stacker information (if a stacker is available for the media type), the media provider ID, and media ID.
- 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.

A different GUI is presented if the media type chosen requires a stacker. See Figure 4.6.1-10 for an example of a GUI without a Stacker. See Figure 4.6.1-11 for an example of a GUI with a Stacker ID. Table 4.6.1-8 describes the fields of this tab.



**Figure 4.6.1-10. Ingest Media Tab (Without Stacker ID)**

ECS Ingest

File Help

Ingest Intro History Log Monitor / Control Operator Tools Media Ingest

Media Type

Stacker ID  Stacker Slot ID

Data Provider

Media Volume ID(Barcode)

Data Delivery Record File Location

☐ On Network  
☐ Embedded in Media

Data Delivery Record File Name:

OK Clear All

**Figure 4.6.1-11. Ingest Media Tab Requiring Stackers**

**Table 4.6.1-8. Ingest Media Field Descriptions**

Field Name	Data Type	Size	Entry	Description
Media Volume ID	integer	N/A	Operator Selected	Media Volume ID input
DDR File Name	character	unlimited	Operator Selected	Name of the Data Delivery Record file
Stacker Slot ID	integer	N/A	Operator Selected	Provides the location (slot ID) of the hard media in the stacker (Presented if Media Type chosen requires a stacker)
Stacker ID	integer	N/A	Operator Selected	Identifies the media stacker in which the media is to be inserted(Presented if Media Type chosen requires a stacker)

**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 GSFC DAAC. These documents can be found in the Technical Documents of the ECS BIS.

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	DCE	via RPC call

#### 4.6.1.4 Databases

The ECS Data Ingest Tool interface with the Ingest Database. For details about the design and schema of the Ingest Database refer to 311-CD-103-005, *Ingest Database Design and Schema Specifications*.

#### 4.6.1.5 Special Constraints

None

#### 4.6.1.6 Outputs

Operations staff have access to three 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.
MSS event log	The MSS event log contains critical events of interest to DAAC operations staff. Stored in MSS.
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 media ingest 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.

**Table 4.6.1-11. Standard Ingest Production Reports**

<b>Report Type</b>	<b>Report Description</b>	<b>When and Why Used</b>
Ingest Request History Report	<p>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.).</p> <p>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.</p>	The report may be generated for specified time periods and executed on a regular basis (based on site policy).
Ingest Request Performance Report	<p>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.</p>	The report may be generated for specified time periods and executed on a regular basis (based on site policy).
Ingest Granule Performance Report	<p>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..</p>	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 4.6.1-12.

Req ID	Data Provider	Completion 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 4.6.1-12. Sample Ingest Request History Report**

An example Ingest Request Performance Report is shown in Figure 4.6.1-13.

Data Provider	Total Requests	Total Request Errors	Granules Avg   Max		Files Avg   Max		Size Avg   Max		Transfer Time Avg   Max		Pre - Proc Time Avg   Max		Archive Time Avg   Max	
SDPF	112	2	24	112	62	336	106.5	114.5	51	105	7	23	12	212
TSDIS	.....	.....												

**Figure 4.6.1-13. Sample Ingest Request Performance Report**



An example Ingest Granule Performance Report is shown in Figure 4.6.1-14.

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 4.6.1-14. Sample Ingest Granule Performance Report**

This page intentionally left blank.

## **4.7 Resource Planning**

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

This page intentionally left blank.

### 4.7.1 Resource Planning

The Resource Editor GUI defines the computer resources which will be allocated to run the production data. The Resource Scheduler GUI is used to create resource reservations. It 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 Scheduler GUI is used by the resource planner to create, validate and approve reservations until a conflict-free plan is achieved. At this point the reservations will be "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 which is used for resource planning and to edit/add to these resource items.	<ul style="list-style-type: none"><li>• To define system resources</li><li>• To add and modify resources</li></ul>
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	<ul style="list-style-type: none"> <li>To register the reservation as the ground event</li> </ul>
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

This section presents an orientation of Resource Planning GUIs. 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.

##### 4.7.1.1.1 Command Line Interface

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

**EcPIRpReStart** <mode> <MSGSVR\_ID>

<mode> is the ECS mode for the execution.

The <MSGSVR\_ID> is the TBS Server Identifier.

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

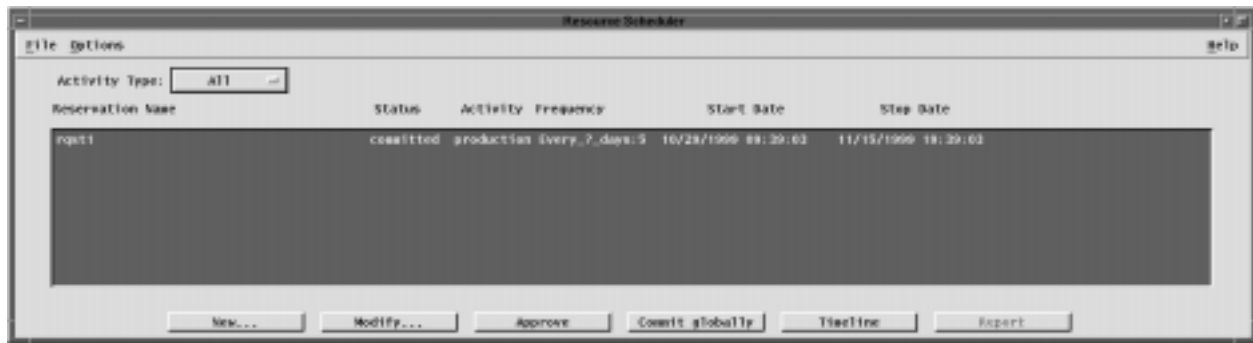
**EcPIRpSiStart** <mode> <MSGSRV\_ID>

<mode> is the ECS mode for the execution.

The <MSGSVR\_ID> is the TBS Server Identifier.

##### 4.7.1.2 Resource Scheduler Main Screen

The Resource Scheduler GUI, depicted in Figure 4.7.1-2, 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 4.7.1-2. Resource Scheduler GUI**

There are several buttons at the bottom of the GUI which affect the status of the reservations. The functions associated to 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, 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** - 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.

#### **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 depicted in Figure 4.7.1-3.



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:

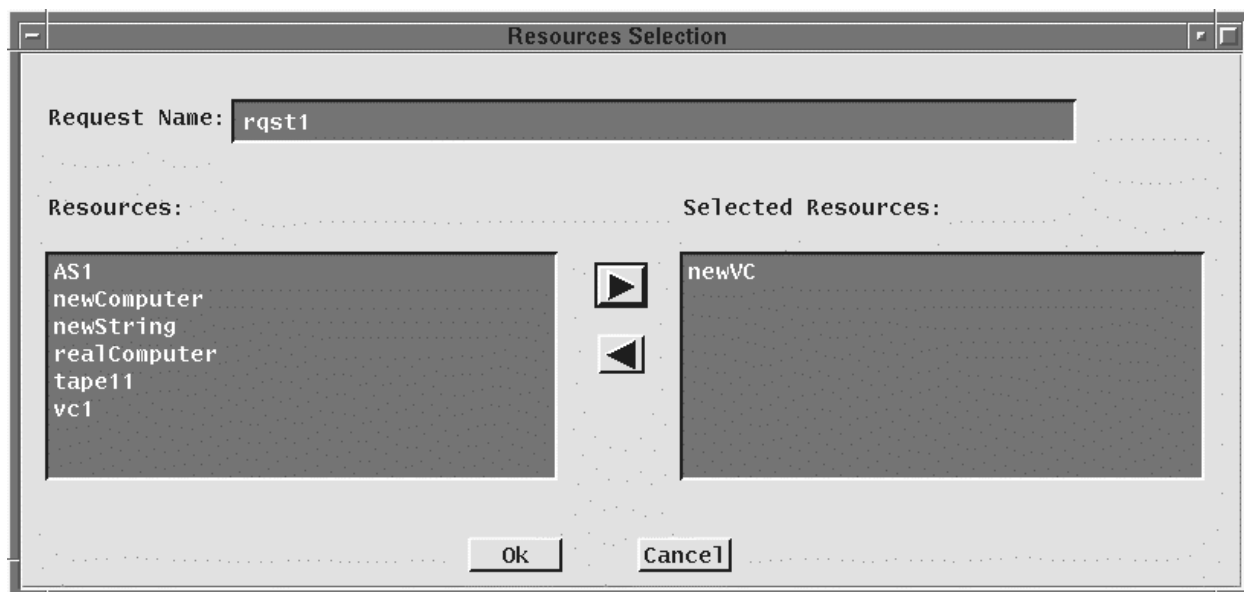
**Figure 4.7.1-3. Resource Reservation Request Edit/Definition GUI**

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.
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 a the reservation is to be repeated at some frequency (see below) 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.
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 will display the Resources Selection GUI (Figure 4.7.1-4) that displays two windows, one containing a list of 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.



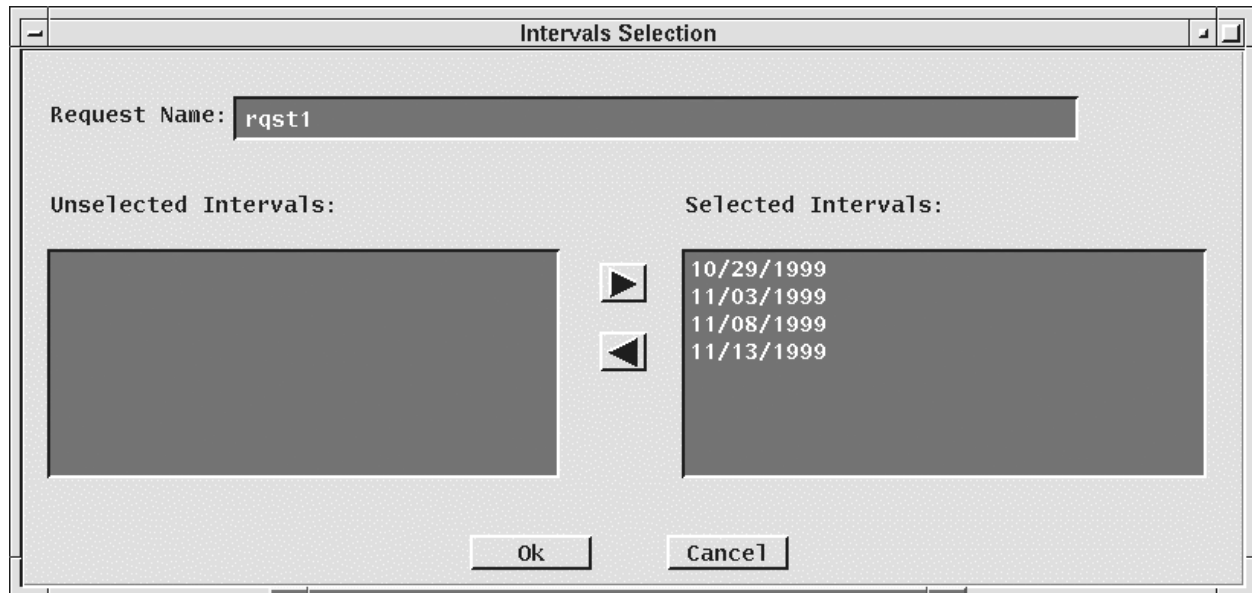
**Figure 4.7.1-4. Resources Selection GUI**

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 below.

**Table 4.7.1-3. Frequency Qualifiers for Resource Reservation Request Edit/Definition GUI**

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.
Weekends	--	Resource reservation for every Saturday & Sunday, between the start date and end date, for the start time and end time specified.
'Every_?_day s'	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 (Figure 4.7.1-5) is displayed, providing a pair of windows . 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 4.7.1-5. 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 4.7.1-6.

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. The use of a resource over a period of time is represented by one or more ‘resource reservation’ bars 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 4.7.1-6. 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

Note, this functionality will be made available in future releases.

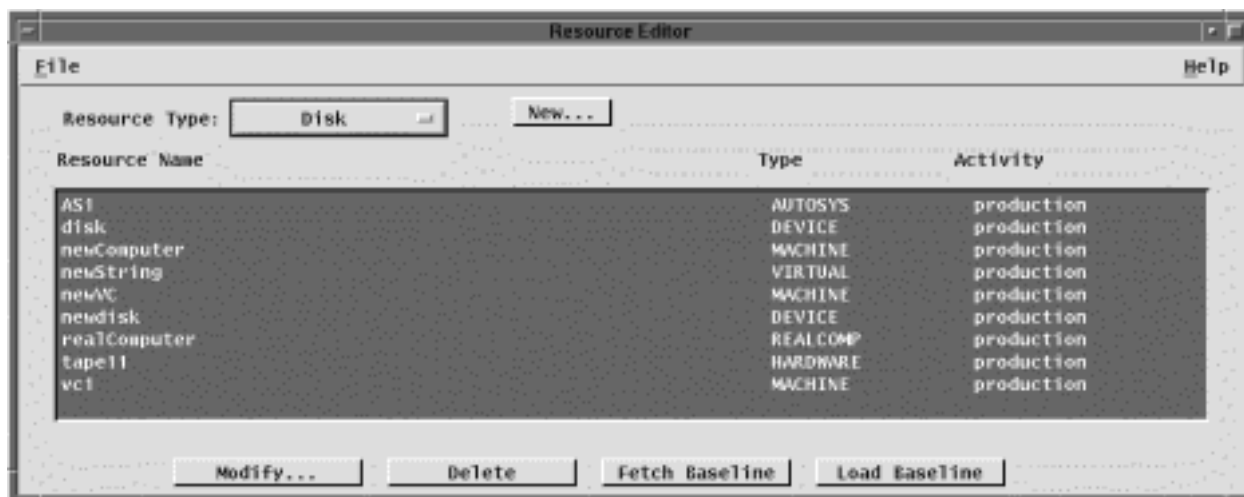
#### *Figure 4.7.1-7. Report Generator GUI (Future Release)*

#### 4.7.1.3 Resource Editor GUI

The resource list may be created from scratch or be initialized from the MSS-managed configuration list, through a 'fetch baseline' operation. Resources may be added to or deleted from the resource list without affecting the MSS configuration list. This is useful, for example, for identifying resources that will be available in the future for the purposes of planning. The Resource Editor GUI, Figure 4.7.1-9, is the tool used for adding resources or modifying the characteristics of existing resources. Upon 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 select 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 a Tivoli job is run (see Section 4.2.3) that generates a file of configuration information used by Resource Planning. This job is run by pressing the '**Fetch Baseline**' button. When the operator determines that the Tivoli job has completed, press the '**Load Baseline**' button to extract the needed information from the Tivoli generated file and load it into the PDPS database.

The GUIs for the definition of specific resource are discussed in the following subsections.

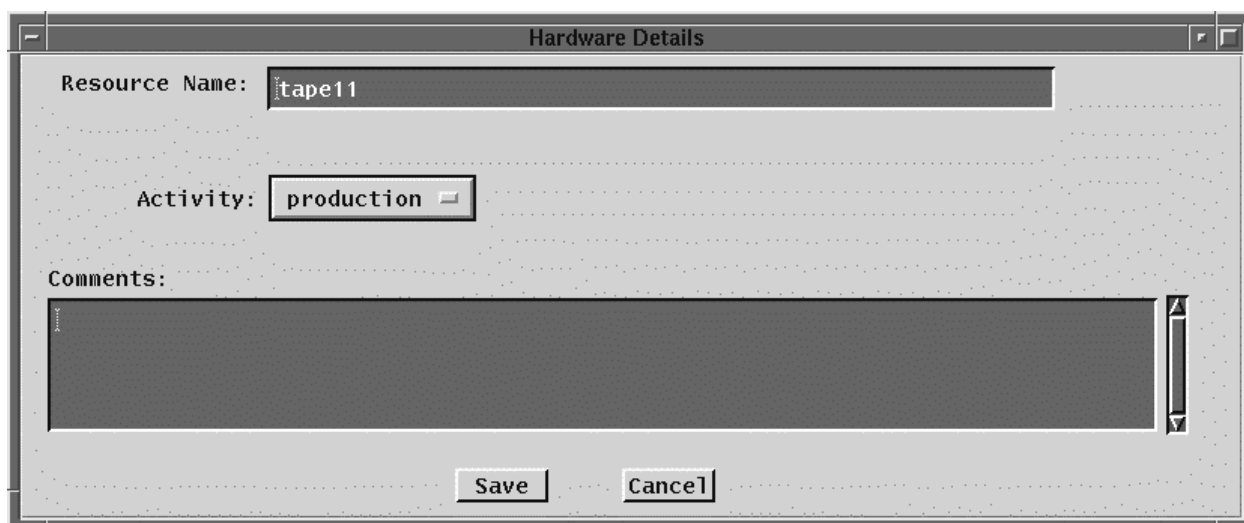


**Figure 4.7.1-8. Resource Editor GUI**

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 4.7.1-9, is used to define the characteristics of generic hardware resources at the DAAC, such as communications equipment, etc. .



**Figure 4.7.1-9. 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 4.7.1-10, is used to define the characteristics of disk partition type resources.

The screenshot shows a window titled "Disk Partition Details". It contains the following fields:

- Resource Name:** A text box containing the value "disk".
- Activity:** A dropdown menu with "production" selected.
- Partition Size:** A text box containing "1000000000" followed by the unit "MBytes".
- Block Size:** A text box containing "1024" followed by the unit "Bytes".
- Comments:** A large, empty text area for user input.
- Buttons:** "Save" and "Cancel" buttons at the bottom right.

**Figure 4.7.1-10. 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.
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 4.7.1-11, is used to define the characteristics of virtual computer resources .

**Figure 4.7.1-11. Virtual Computer Details GUIs**

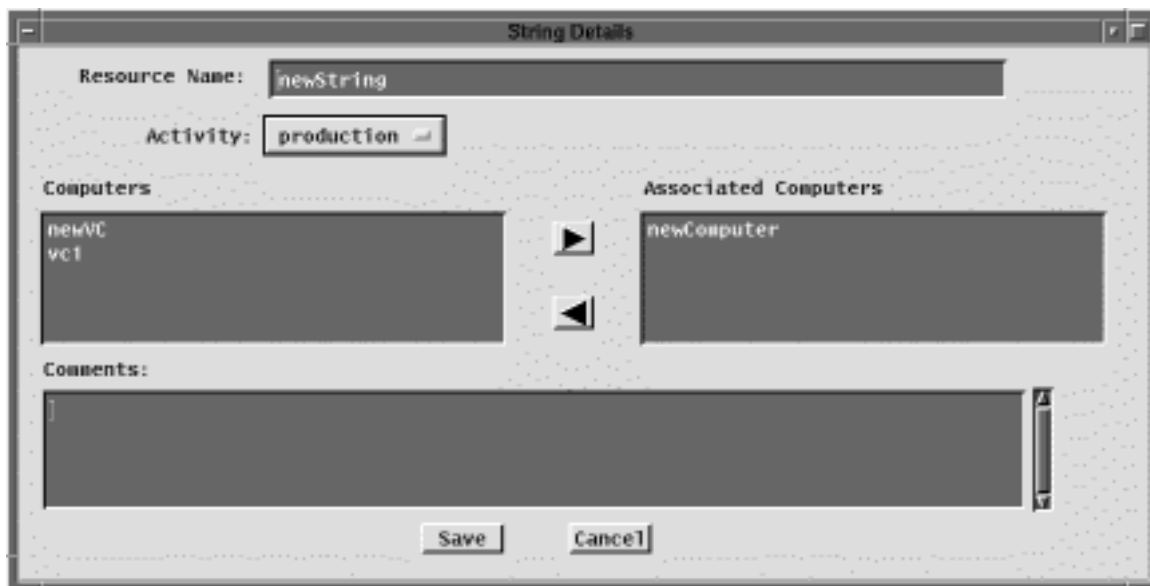
The characteristics of the other fields are described in Table 4.7.1-6. .

**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.
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"> <li>• A list of the disks previously defined for that site.</li> <li>• This list of disks is used to select from when a disk is associated (or disassociated) with the computer using the arrow buttons.</li> </ul>
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 4.7.1-12, is used to define the characteristics of a string, or a collection of virtual computer resources.



**Figure 4.7.1-12. String Details GUIs**

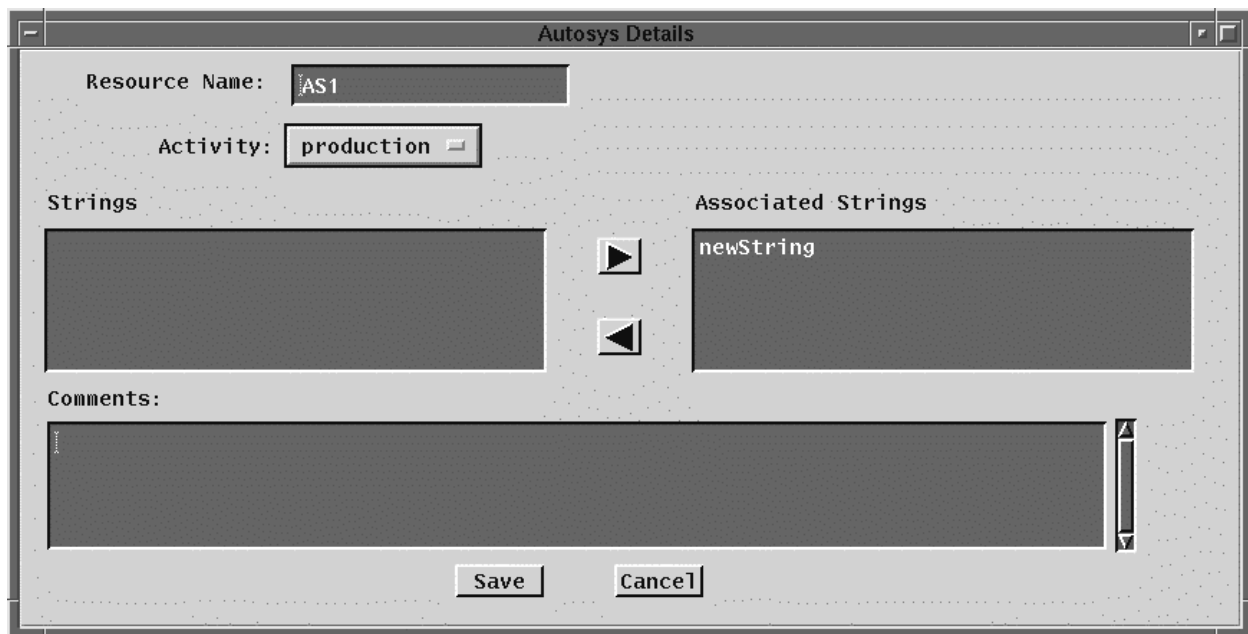
The characteristics of the other fields are described in Table 4.7.1-7.

**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.
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 4.7.1-13 and 4.7.1-14.

**Figure 4.7.1-13. Real Computer Details GUIs**



**Figure 4.7.1-14. 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-TDx-013 “Custom Code Configuration Parameters” documentation series. The “x” refers to the installed location, e.g. 920-TDG-013 is for GSFC DAAC.

#### **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 3 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 provides 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 D.

#### **4.7.1.9 Reports**

Note, this functionality will be made available in the future releases.

This page intentionally left blank.