4.3.5 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 4.3.5-1 summarizes the operating functions that FLEXlm supports.

Table 4.3.5-1. 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 4.3.5-1. 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 FLEX*admin* product. FLEX*admin* 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.

4.3.5.1 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.

4.3.5.1.1 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 4.3.5-2 summarizes commands available with FLEXlm. See the *FLEXlm End User's Manual* for the complete description of each command and its arguments.

Table 4.3.5-2. Command Line Interfaces (1 of 2)

Command Line Interface	Description and Format	When and Why Used
lmcksum	lmcksum [-c *license_file*]	To verify license file data.
lmdiag	lmdiag [-c *license_file*] \ [-n] [*feature*]	To diagnose problems when a license cannot be checked out.

Table 4.3.5-2. Command Line Interfaces (2 of 2)

Command Line Interface	Description and Format	When and Why Used
lmdown	lmdown [-c *license_file*] [-q]	To shutdown all license daemons (both lmgrd and all vendor daemons) on all nodes.
lmgrd	lmgrd [-app] [-c *license_file*] \\ [-t *timeout_interval*] [-l *logfile*] \\ [-s *timestamp_interval*] [-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 {*infile* | -}] [-o *outfile*] \\ [-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 *file*] *feature user host* \\ *display* or lmremove [-c *file*] -h *feature host* \\ *port handle*	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 checks out the license again after it is freed by lmremove.)
lmreread	lmreread [-c *license_file*] \\ [–vendor *name*]	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 *license file*] *feature* \\ *new-file* or lmswitchr [-c license file] *vendor* \\ *new-file (v5.0+ onl)*	To start recording license events in a new or different log file for the FLEX*admin* tool.
lmstat	lmstat [-a] [-A] [-c *license_file*] \\ [-f *feature*] [-i [*feature*]] \\ [-S *vendor*] [-s *hostname*] \\ [-t *value*]	To report the status of all network licensing activities.
lmver	lmver *filename*	To identify the FLEXlm version of a library or binary file.

4.3.5.2 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.

4.3.5.3 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 installation and release notes for FLEXlm, refer to the ECS Baseline Information System web page, URL http://cmdm.east.hitc.com/.

4.3.5.4 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 the *FLEXlm End User's Manual* for information about the format of a license file, and 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 the *FLEXlm End User's Manual* for details.

4.3.5.5 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's 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 the FLEXlm *End User's* Manual.
- Run lmgrd as a non-privileged user (not *root)* to avoid security risks. Refer to the FLEXlm *End User's* Manual

4.3.5.6 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.

4.3.5.7 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.

See the appendices of the *FLEXlm End User's 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.

4.3.5.8 Reports

FLEXlm's *lmstat* utility can generate the status reports listed in Table 4.3.5-3. Each is written to standard output and may be redirected to a named file or a printer using standard Unix conventions. Reports are generated on demand as required to meet operational needs.

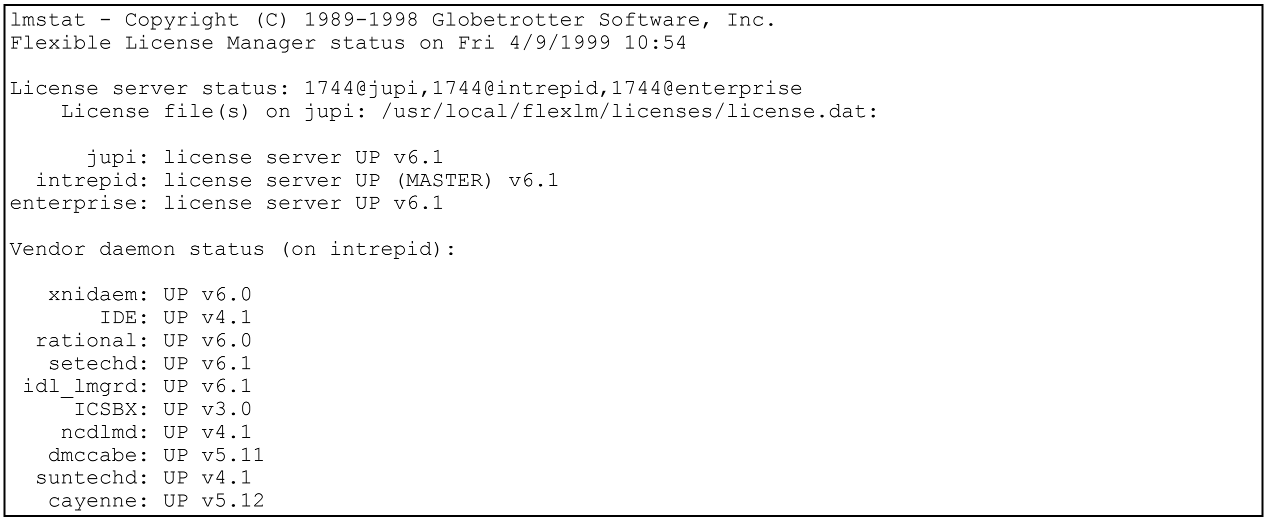
Table 4.3.5-3. Reports

Report Type	Report Description	Example
lmstat -s	Lists status of clients running on a named host.	Figure 4.3.5-1
lmstat -i	Lists license information about all or a named feature.	Figure 4.3.5-2
lmstat -a	Lists all information about current network licensing activities.	Figure 4.3.5-3
lmstat -A	Lists all currently active licenses.	Figure 4.3.5-4
lmstat -f	Lists users of all or a named feature.	Figure 4.3.5-5
lmstat -S	Lists users of all or a named vendor's features.	Figure 4.3.5-6

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

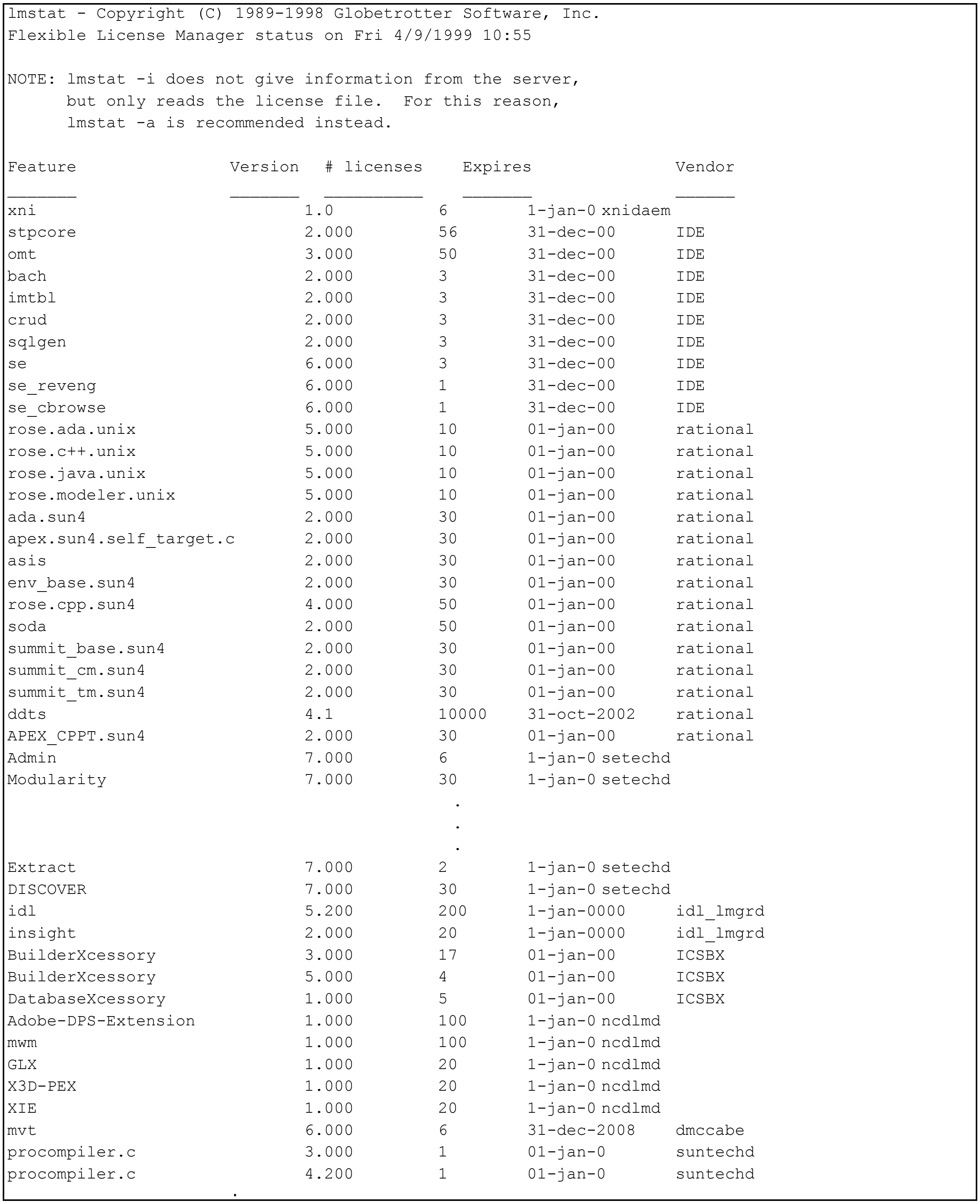
4.3.5.8.1 Sample Reports

The figures (Figure 4.3.5-1 through 4.3.5-6) that follow contain sample FLEXlm status reports. One sample is provided for each report listed in Table 4.3.5-3.



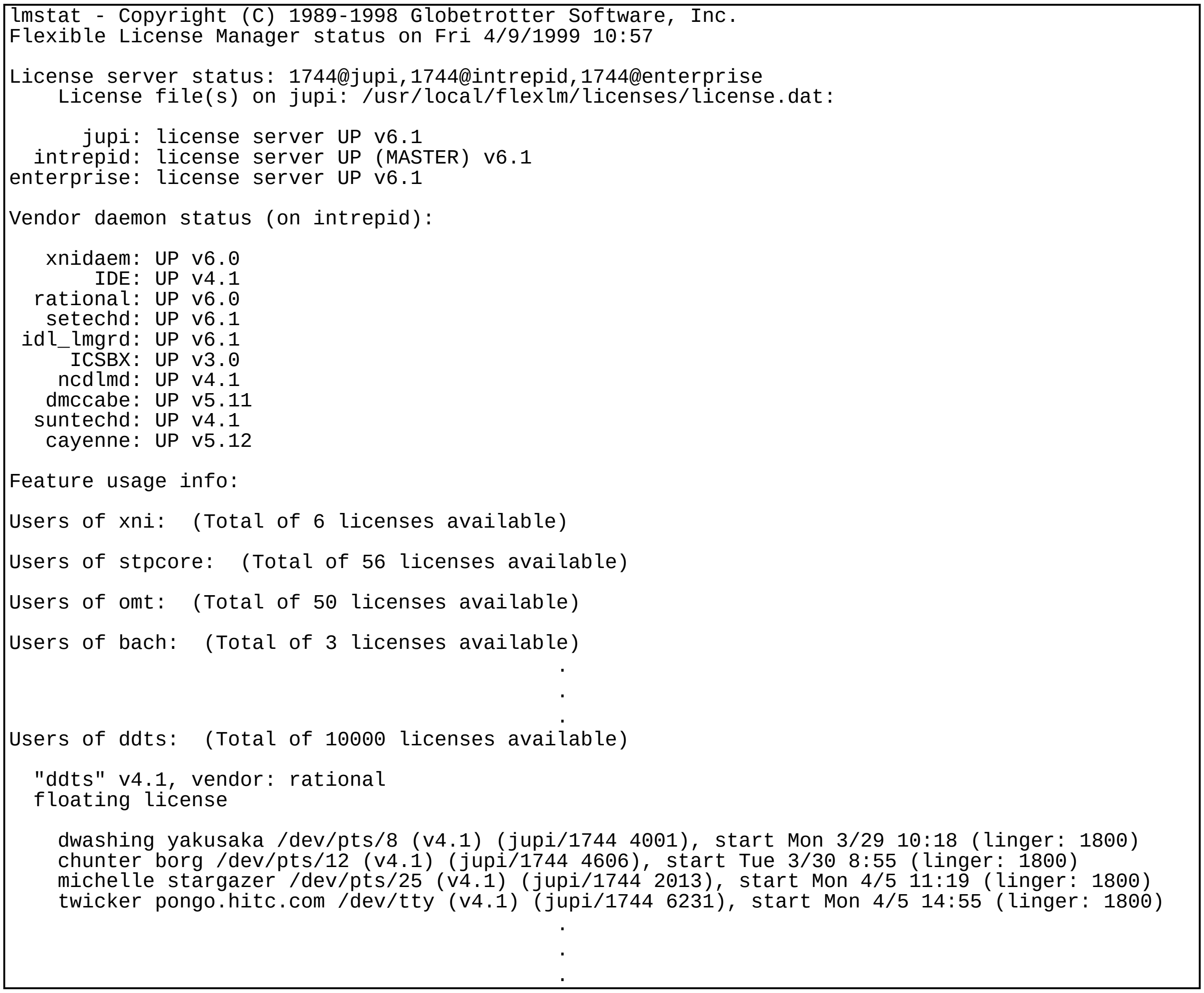
```
lmstat - Copyright (C) 1989-1998 Globetrotter Software, Inc.
Flexible License Manager status on Fri 4/9/1999 10:54

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 4.3.5-1. All Clients (lmstat -s) Report



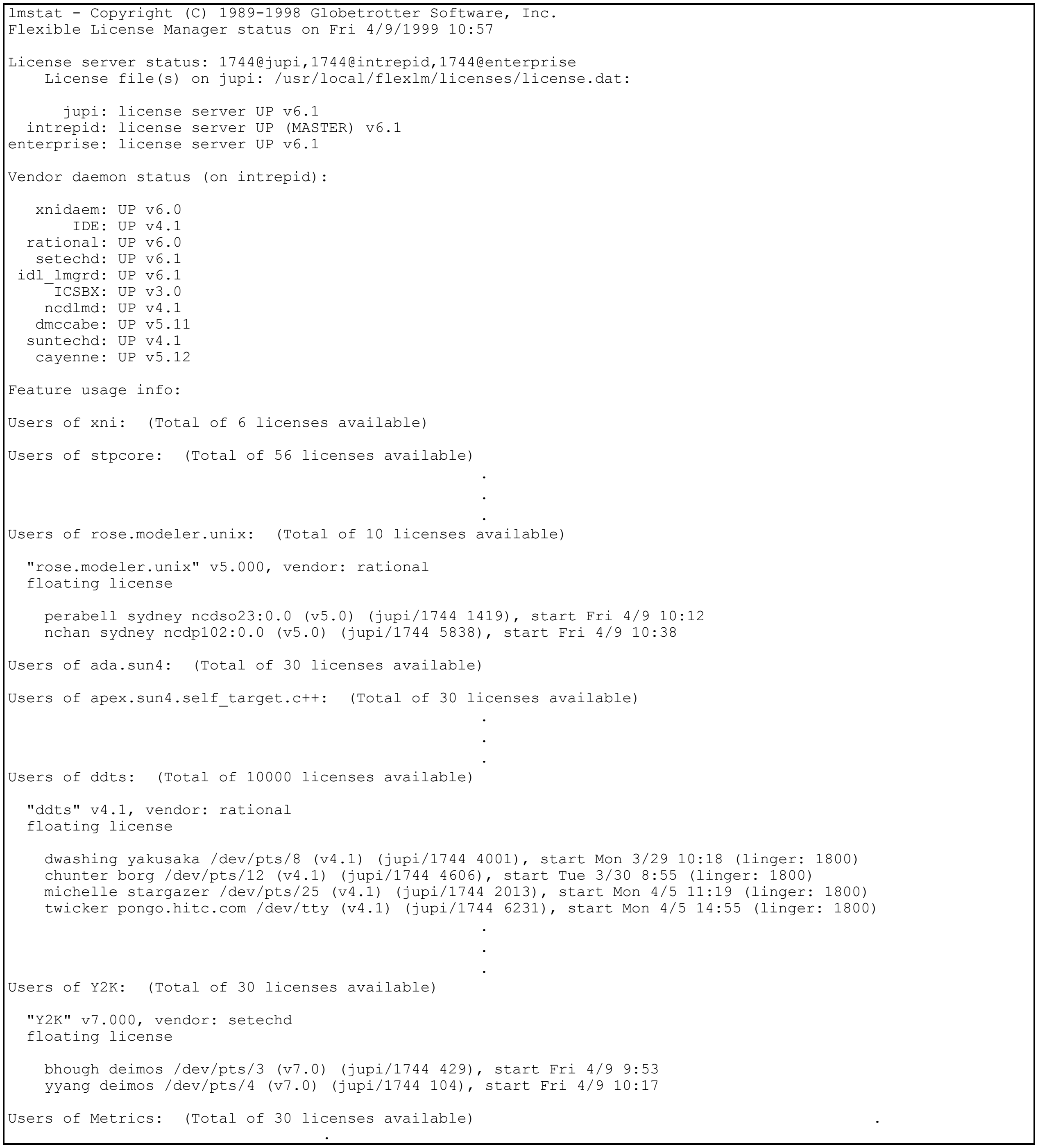
```
lmstat - Copyright (C) 1989-1998 Globetrotter Software, Inc.
Flexible License Manager status on Fri 4/9/1999 10:55

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-0 xnidaem
stpcore                       2.000         56       31-dec-00      IDE
omt                           3.000         50       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         10       01-jan-00      rational
rose.c++.unix                 5.000         10       01-jan-00      rational
rose.java.unix                5.000         10       01-jan-00      rational
rose.modeler.unix             5.000         10       01-jan-00      rational
ada.sun4                      2.000         30       01-jan-00      rational
apex.sun4.self_target.c       2.000         30       01-jan-00      rational
asis                          2.000         30       01-jan-00      rational
env_base.sun4                 2.000         30       01-jan-00      rational
rose.cpp.sun4                 4.000         50       01-jan-00      rational
soda                          2.000         50       01-jan-00      rational
summit_base.sun4              2.000         30       01-jan-00      rational
summit_cm.sun4                2.000         30       01-jan-00      rational
summit_tm.sun4                2.000         30       01-jan-00      rational
ddts                          4.1           10000    31-oct-2002    rational
APEX_CPPT.sun4                2.000         30       01-jan-00      rational
Admin                         7.000         6        1-jan-0 setechd
Modularity                    7.000         30       1-jan-0 setechd
                                             .
                                             .
                                             .
Extract                       7.000         2        1-jan-0 setechd
DISCOVER                      7.000         30       1-jan-0 setechd
idl                           5.200         200      1-jan-0000     idl_lmgrd
insight                       2.000         20       1-jan-0000     idl_lmgrd
BuilderXcessory               3.000         17       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-0 ncdlmd
mwm                           1.000         100      1-jan-0 ncdlmd
GLX                           1.000         20       1-jan-0 ncdlmd
X3D-PEX                       1.000         20       1-jan-0 ncdlmd
XIE                           1.000         20       1-jan-0 ncdlmd
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 4.3.5-2. 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 4.3.5-3. All Licensing 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 4.3.5-4. 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 4.3.5-5. Users of All or Named Features (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 4.3.5-6. Users of All or Named Vendor's Features (lmstat-S) Report

This page intentionally left blank.

4.3.6 Remedy's Action Request System

The Remedy Action Request System provides a distributed Trouble Ticketing Service that furnishes DAACs a common environment and the means of classifying, tracking, and reporting problem occurrences and resolutions to both ECS users and operations personnel. The Trouble Ticketing Service:

- Provides a GUI for operations personnel to access all Trouble Ticket services
- Provide a common Trouble Ticket entry format
- Stores Trouble Tickets
- Retrieves Trouble Tickets via ad-hoc queries
- Allows operations personnel to forward problems from one DAAC to another
- Generates reports and statistics
- Interfaces with user's and operator's e-mail to provide automatic notification
- Offers an application programming interface through which applications can submit Trouble Tickets
- Provides summary information to the SMC from each DAAC to allow trend reports regarding Trouble Tickets
- Defines a consistent "life-cycle" for Trouble Tickets
- Allows each DAAC a degree of customization through definition of further escalation and action rules.

Escalation rules are time-activated events executed on Trouble Tickets meeting a set of specified criteria. Actions taken can include notification (either a user or support staff member), writing to a log file, setting a field value on the Trouble Ticket, or even running a custom designed process. Qualifications can be expressed on any Trouble Ticket data tracks. Active links are similar to escalation rules with the exception that they are defined to take place on a specified action rather than at a given time.

In addition to the functionality provided by Remedy's Action Request System, the Trouble Ticketing Service utilizes a set of custom HTML documents to provide registered users with the capability to submit new Trouble Tickets and query the current status of any of their previous entries. Access to the Trouble Ticketing system through this technique provides users an easy method for reporting problems in an environment already familiar to them. Additionally, as another means of Trouble Ticket entry, the Trouble Ticket services provide a text e-mail template through which automated entry of Trouble Tickets is possible. Support staff members enter Trouble Tickets through the Remedy's Action Request System provided interface for problems received via other methods (e.g., phone calls).

In addition to tracking Trouble Tickets, the Remedy Action Request System also functions as the User Contact Log. Remedy's Action Request System is configured to have a separate form containing the entries User Services personnel enter for each contact they have with a user. A user contact log allows a Trouble Ticket to be initiated from a log entry with the push of a button. The Trouble Ticket is populated with information from the contact log.

User Services and other operations personnel use Remedy's Action Request System to perform the functions listed in Table 4.3.6-1.

Table 4.3.6-1. Common ECS Operating Functions Performed with Remedy's Action Request System (1 of 4)

Operating Function	GUI (Section)	Description	When and Why to Use
Access Trouble Ticket services	User Tool (4.3.6.2)	• Accessed by executing the User Tool and opening a new RelB-Trouble Tickets form • Main Remedy Trouble Ticket screen used for submitting, modifying, or displaying a Trouble Ticket.	When there is a need to submit, query, or revise a Trouble Ticket
Submit a Trouble Ticket	User Tool (RelB-Trouble Tickets form) (4.3.6.2)	• Submitted by executing the User Tool and opening a new RelB-Trouble Tickets form • Trouble Ticket form is used to enter information about the problem	When a problem is either found by or reported to User Services
Retrieve a Trouble Ticket	User Tool (RelB-Trouble Tickets form) (4.3.6.2)	• Retrieved by executing the User Tool and opening a RelB-Trouble Tickets form in Search mode • Allows entry of new information about the problem	When information needs to be added to a trouble ticket or when a trouble ticket needs to be viewed

Table 4.3.6-1. Common ECS Operating Functions Performed with Remedy's Action Request System (2 of 4)

Operating Function	GUI (Section)	Description	When and Why to Use
Forward a Trouble Ticket to another DAAC	User Tool (RelB-Trouble Ticket, RelB-TT-ForwardToSite and RelB-TT-Sites form) (4.3.6.2)	• Forwarded by executing the User Tool and opening RelB-Trouble Ticket form • The Forward-To field contains a drop-down menu for the selection of a forwarding destination. • Forwarded information is supported and held by the RelB-TT-ForwardToSite form • The RelB-TT-Sites form supplies the site name and email address to be used in the forwarding	When a Trouble Ticket is deemed relevant to another site
Generate reports	User Tool (RelB-Trouble Tickets form) (4.3.6.2)	• Generated by executing the User Tool and opening RelB-Trouble Tickets form • Reports can be created and run by either selecting Report button in the tool bar or by selecting Tools -> Reporting from the main menu.	When information is needed about one or more Trouble Tickets
Add, delete, or modify user accounts	User Tool (RelB-User form) (4.3.6.2.1)	• Added, Deleted, or Modified by executing the User Tool and opening a RelB-User form in new or search mode. • RelB-User form contains key information about a user account	When there is a need to add new Remedy users, delete old users, or modify old users
Create/Update User Contact Log entry and submit a Trouble Ticket from a log entry	User Tool (Contact Log form) (4.3.6.2.2)	• Done by executing the User Tool and opening a RelB-Contact Log form • Used to classify, track, and report contacts of ECS users and operators.	When there is a need to record user contacts and generate Trouble Tickets

Table 4.3.6-1. Common ECS Operating Functions Performed with Remedy's Action Request System (3 of 4)

Operating Function	GUI (Section)	Description	When and Why to Use
Provide a description of a hardware problem that corresponds to a Trouble Ticket	User Tool (Hardware Information form) (4.3.6.2.3)	• Provided by executing the User Tool and opening a RelB-Hardware Information form, or by selecting the Hardware Information link from the Trouble Tickets form • Screen used to enter detailed information about failed hardware components (e.g., part and serial numbers) and the actions taken to correct the problem	When detailed hardware information needs to be provided beyond what can be entered on the Trouble Tickets form
Customize pulldown menus on RelB-Trouble Tickets form	User Tool (RelB-Menu-Closing Codes, RelB-Menu-Hardware Resources, RelB-Menu-Software Resources, RelB-Menu-Key Words, RelB-Menu-Problem Type, form) (4.3.6.2.4-4.3.6.2.8)	• Customized by executing the User Tool and opening: RelB-Menu-Closing Codes, RelB-Menu-Hardware Resources, RelB-Menu-Software Resources, RelB-Menu-Key Words, RelB-Menu-Problem Type, and Sites forms • Pick list items can be added, deleted, or modified from these forms	When current menu is inadequate
Add, delete, modify a site name and email address	User Tool (RelB-TT-Sites form) (4.3.6.2.9)	• Added, deleted, or modified by executing the User Tool and opening RelB-TT-Sites form • Provides a picklist of sites (DAACs), SMC, NISN, and EMSn	When there is a need to add, delete, or modify the site name and email address information that is used in forwarding
Notification and/or customization at different states of a Trouble Ticket	Admin Tool and User Tool (RelB-TT-Times form) (4.3.6.2.10 & 4.3.6.2.11)	• Accessed by executing the AdminTool to open the correct filter, escalation, or active link • Accessed by clicking on User Tool icon and opening RelB-TT-Times form to review/modify a Trouble Ticket	To notify someone or set fields as soon as a Trouble Ticket reaches a particular state or if a Trouble Ticket is in a particular state too long

Table 4.3.6-1. Common ECS Operating Functions Performed with Remedy's Action Request System (4 of 4)

Operating Function	GUI (Section)	Description	When and Why to Use
Notify the user of a Remedy event	Notification Tool (4.3.6.2.12)	• Notify by executing the Remedy Notification Tool • Allows properties and options to be modified via pull-down menus	When there is a need for an alternative to email notification
Import entries into a particular form	Import Tool (4.3.6.2.13)	• Imported by executing the Remedy Import Tool • Enables the user to import entries into a form from a file generated by the Admin tool	When there is a need to import existing entries rather than retyping information manually
Submit a Trouble Ticket via HTML	Trouble Ticket HTML (4.3.6.2.14)	• Submitted by opening the Trouble Ticket form in a Web browser • Submit, obtain a list and view details of Trouble Tickets	When there is a need to submit Trouble Tickets without going through Remedy

4.3.6.1 Quick Start Using Remedy's Action Request System

This section describes how to invoke Remedy and provides a description of customized Remedy GUIs. Standard Remedy features (e.g., pull-down menus) are not discussed in this document. For more information on Remedy's Action Request System, refer to the following:

- *Remedy's Action Request Concepts Guide version 4.5*, Chapter 1 "Overview," page 1-1
- *Remedy's Action Request System Quick Start Guide version 4.0*, Chapter 2 "Remedy Administrator," page 2-1
- *Remedy's Action Request System Quick Start Guide version 4.0*, Chapter 3 "Remedy User," page 3-1
- For information on the fields of the GUIs shown in this section, use the Context Sensitive Help available for that GUI.

The documentation of AR System used as a basis and referenced in this section is for version/release 4.0 and 4.5.

4.3.6.1.1 Invoking Remedy's Action Request System from the Command Line Interface

The current ECS Remedy User Tool configuration is set for a UNIX environment. In UNIX manual page syntax, the command to run the User Tool is:

```
% aruser
```

The other Remedy version 4.5 tools (the Admin tool, Notifier, and Import) are Windows based. By default, the Home folder is in the Remedy User installation folder. To execute the AR Admin tool from the command line interface use:

```
$ c:\progra~1\remedy\aradmin.exe
```

To execute the AR Notification tool from the command line interface use:

```
        $ c:\progra~1\remedy\notifier.exe
```

To execute the AR Import tool from the command line interface use:

```
        $ c:\progra~1\remedy\arimport.exe
```

For more information on the names and locations of configuration and option files in the AR System installation for Windows, contact your administrator or see the *AR System Installation and Getting Started Guide*.

4.3.6.2 Remedy's User Tool Main Screen (RelB-Trouble Tickets Form)

Remedy's Action Request System User Tool Main screen is the RelB-Trouble Tickets Form GUI shown in Figure 4.3.6-1. From here Trouble Tickets can be submitted, queried, modified, or forwarded.

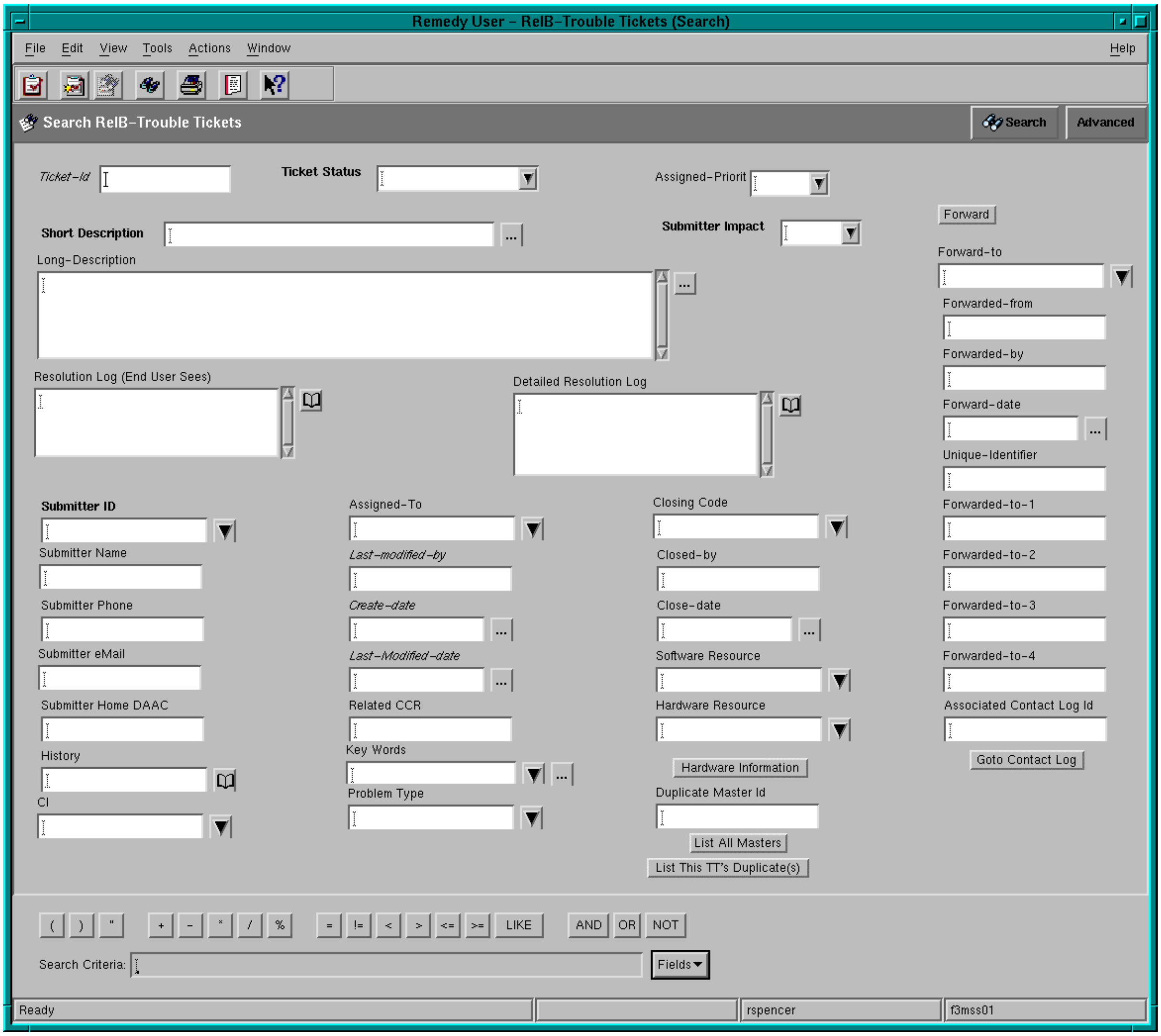


Figure 4.3.6-1. RelB-Trouble Tickets Form GUI

Table 4.3.6-2 provides a description of the RelB-Trouble Tickets Form fields.

Table 4.3.6-2. RelB-Trouble Tickets Field Descriptions (1 of 3)

Field Name	Data Type	Size	Entry	Description
Ticket-Id	Character	15	System generated	Ticket number, which is set and maintained by the system.
Ticket Status	Selection	*	Required	Status of the Trouble Ticket.
Assigned-Priorit	Selection	*	Optional	Priority of Trouble Ticket assigned at the site.
Short Description	Character	128	Required	Short Description of the problem.
Submitter Impact	Selection	*	Required	Impact of the problem to the submitter.
Long-Description	Character	4060	Optional	Long Description of the problem.
Resolution Log (End User Sees)	Diary	Unlim	Optional	General steps in the resolution of the problem.
Detailed Resolution Log	Diary	Unlim	Optional	Detailed steps in problem resolution.
Submitter ID	Character	30	Required	User Id of the Submitter.
Submitter Name	Character	30	Optional	Full Name of the Submitter.
Submitter Phone	Character	30	Optional	Phone number of the Submitter.
Submitter eMail	Character	64	Optional	E-mail address of the Submitter.
Submitter Home DAAC	Character	60	Optional	Home DAAC of the Submitter.
History	Diary	Unlim	Optional	• Upon submission or modification, the person assigned to the ticket and the ticket status are indicated in the History field • Due to a limitation in Remedy, this information is only written when the Assigned-to and Status fields are modified
CI	Character	30	Optional	Name of the configuration item to which the problem is associated.
Assigned-To	Character	30	Optional	Person who Trouble Ticket has been assigned to.
Last-modified-by	Character	30	System generated	Person who last modified the Trouble Ticket.
Create-date	Date/Time	17	System generated	Date and time Trouble Ticket was created at the present site (mm/dd/yy hh:mm:ss).
Last-Modified-date	Date/Time	17	System generated	Date and time the Trouble Ticket was last modified (mm/dd/yy hh:mm:ss).

* Note, the size of a field with a "selection" data type can vary and the size is automatically adjusted to the size of the item selected from the selection list.

Table 4.3.6-2. RelB-Trouble Tickets Field Descriptions (2 of 3)

Field Name	Data Type	Size	Entry	Description
Related CCR	Character	60	Optional	ID of a related CCR or CCRs. [if more than one CCR, separate CCRs by a space for readability].
Key Words	Character	255	Optional	Key words to help identify this Trouble Ticket (e.g., Hardware, Software, Configuration).
Problem Type	Character	30	Optional	Type of problem addressed by this Trouble Ticket (e.g., Configuration Error, Hardware Problem, Software Problem).
Closing Code	Character	60	Optional	Code assigned to the type of problem necessitating the writing of this Trouble Ticket.
Closed-by	Character	60	Optional	Person that closed this Trouble Ticket.
Close-date	Date/Time	17	Optional	Date and time this Trouble Ticket was closed. (mm/dd/yy hh:mm:ss)
Software Resource	Character	60	Optional	Software Resource the problem came from.
Hardware Resource	Character	60	Optional	Hardware Resource the problem came from.
Duplicate Master Id	Character	25	Optional	The Ticket-ID of the primary Trouble Ticket for the problem reported in this trouble ticket and its associated duplicate trouble tickets (other tickets reporting the same problem).
Forward-to	Character	60	Optional	Site this Trouble Ticket was last forwarded to.
Forwarded-from	Character	60	Optional	Site that forwarded this Trouble Ticket.
Forwarded-by	Character	60	Optional	Contact person at the forwarding site.
Forward-date	Date/Time	17	Optional	Date and time Trouble Ticket was forwarded. (mm/dd/yy hh:mm:ss)
Unique-Identifier	Character	20	Optional	• Unique identifier which is established at the origination site • This identifier should NEVER be changed once set
Forwarded-to-1	Character	60	Optional	First site to have been forwarded this Trouble Ticket.

Table 4.3.6-2. RelB-Trouble Tickets Field Descriptions (3 of 3)

Field Name	Data Type	Size	Entry	Description
Forwarded-to-2	Character	60	Optional	Second site to have been forwarded this Trouble Ticket.
Forwarded-to-3	Character	60	Optional	Third site to have been forwarded this Trouble Ticket.
Forwarded-to-4	Character	60	Optional	Fourth site to have been forwarded this Trouble Ticket.
Associated Contact Log Id	Character	30	Optional	ID number of the Associated Contact Log entry for this trouble ticket.

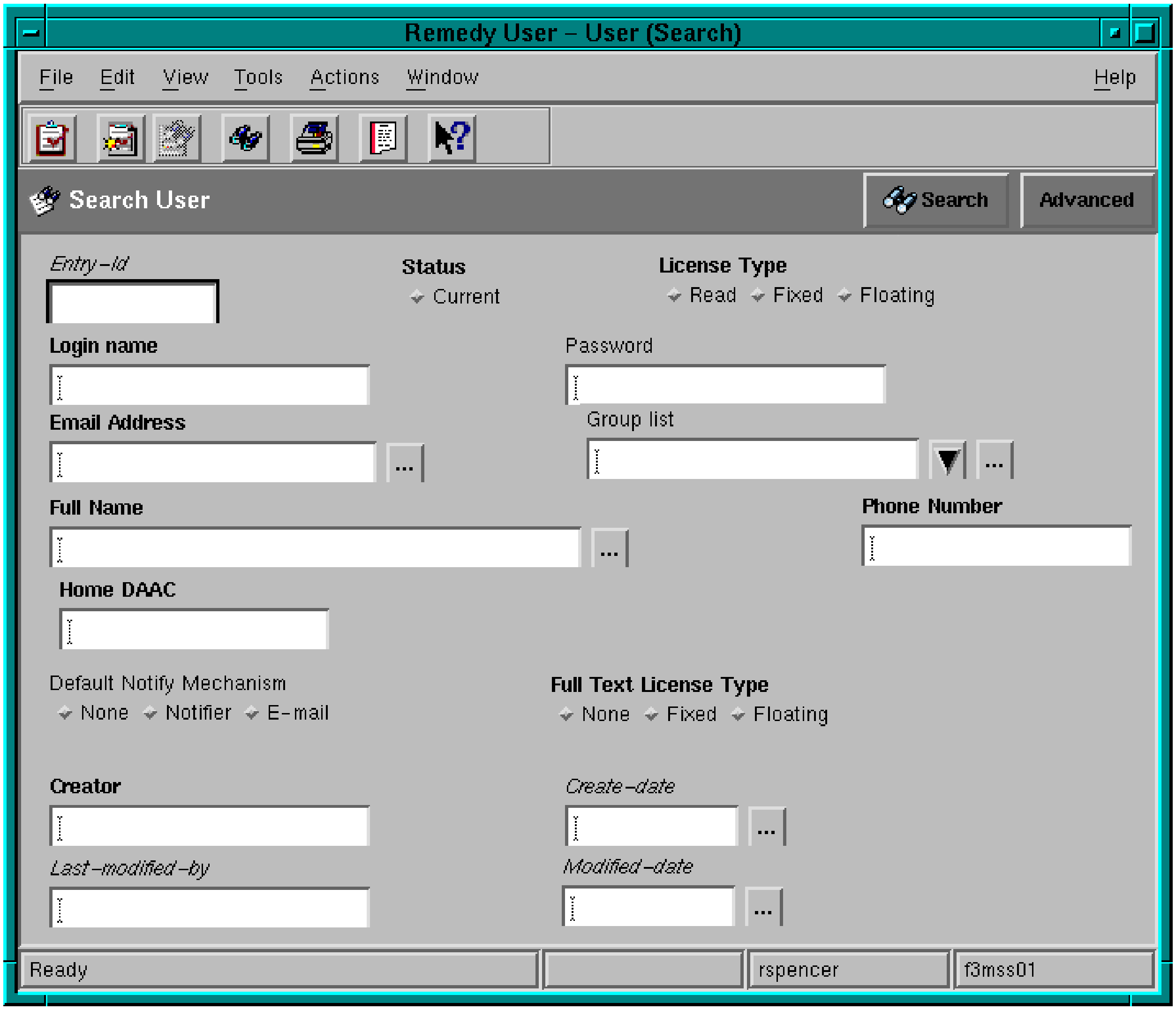
In addition to the fields described in the above table, the RelB-Trouble Tickets form provides the following buttons (active links):

- **Forward** – Forwards this Trouble Ticket to the site specified in the "Forward-to" field.
- **Hardware Information** – Opens a window associated with this Trouble Ticket to hold hardware information.
- **List All Masters** -- All Trouble Tickets that are duplicates of each other have one master. This button lists all master Trouble Tickets.
- **List This TT's Duplicate(s)** – List all Trouble Tickets having duplicates associated with this Trouble Ticket.
- **Goto Contact Log** – If the displayed Trouble Ticket was created from a Contact Log entry then this button opens a window to that Contact Log entry. The Trouble Ticket was created from a Contact Log entry if the "Associated Contact Log Id" field has a value in it.

The RelB-Trouble Tickets form also provides the Search Criteria field, which is part of Remedy's Advanced Search Bar -- an aid for querying the database – and is available on any form. Users can query the database in two ways. One can enter a value into any of the fields displayed on the form and click on the Search button in the tool bar or at the top of any form; this will open a display containing the database records having these values. Alternatively, one can specify search criteria in the Advanced Search Bar at the bottom of the GUI. Clicking on the "Fields" button in the search bar obtains a pull down menu (or pick list) of fields, selection values, and keywords. These can be chosen in combination with any of the operations listed in the search bar before clicking on the form's Search button to display qualifying records.

4.3.6.2.1 Remedy's User Tool Screen (User Form)

The administrator to add, modify or remove users of the Action Request (AR) System uses the "User" form, shown in Figure 4.3.6-2. The "User" form is used in conjunction with the "Group" form to provide users with permissions ultimately determining which operations individual users can perform and which forms and fields they can access. For more information on the "User" form and the AR System access control, refer to the Action Request System Server Administrator's Guide.



Ready | rspencer | f3mss01

Figure 4.3.6-2. User Form GUI

Table 4.3.6.3 provides a description of the User Form fields.

Table 4.3.6-3. User Form Field Descriptions

Field Name	Data Type	Size	Entry	Description
Entry-Id	Character	15	System generated	Entry ID of user.
Login name	Character	30	Required	Login name of user.
Password	Character	30	Optional	Password of User.
Email Address	Character	255	Required	E-mail address of User.
Group list	Character	255	Optional	Groups to which the user belongs.
Full Name	Character	128	Required	Full Name of User.
Phone Number	Character	55	Required	Phone Number of User.
Home DAAC	Character	55	Required	Home DAAC of User.
Creator	Character	30	Required	Person who created the user account.
Create-date	Date/Time	17	System generated	Date and time the entry was created at the present site. (mm/dd/yy hh:mm:ss)
Last-modified-by	Character	30	System generated	User ID of person that last modified the user entry.
Modified-date	Date/Time	17	System generated	Date and time of last modification to user entry (mm/dd/yy hh:mm:ss).

*Note, the size of a field with a "selection" data type can vary and the size is automatically adjusted to the size of the item selected from the selection list.

In addition to the fields described in the above table, the RelB-Trouble Tickets form provides the following buttons (active links):

- Status - Is user current or not as shown by the “current” button.
- License Type - What type of license does this user have (e.g., read, fixed, floating)?
- Default Notify Mechanism - Notification method (e.g., None, Notifier, and Email buttons.)
- Full Text License Type - Not applicable

4.3.6.2.2 Remedy’s User Tool Screen (Contact Log Form)

The Contact Log Form GUI, shown in Figure 4.3.6-3, is used to enter information about a contact to User Services.

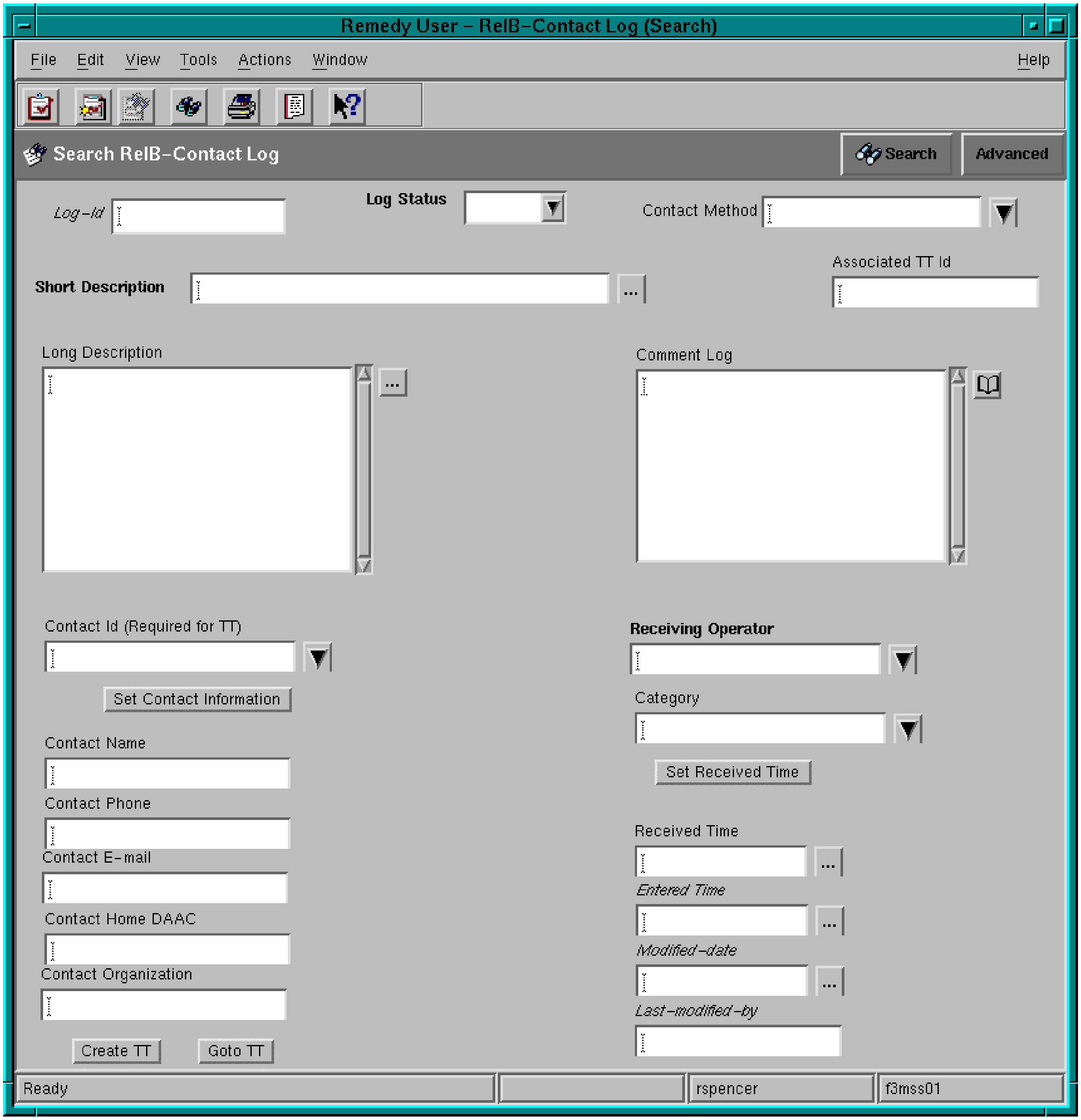


Figure 4.3.6-3. Contact Log Form GUI

Table 4.3.6-4 provides a description of the Contact Log Form fields.

Table 4.3.6-4. Contact Log Form Field Descriptions

Field Name	Data Type	Size	Entry	Description
Log-Id	Character	15	System generated	ID of Contact Log.
Log Status	Selection	*	Required	Status of Contact Log.
Contact Method	Character	50	Optional	Method used to contact the person entering the Contact Information.
Short Description	Character	128	Required	Short Description of the contact.
Associated TT Id	Character	60	Optional	If a Trouble Ticket is created from this Contact Log then this is the related Trouble Ticket ID.
Long Description	Character	255	Optional	Long Description of the contact.
Comment Log	Diary	Unlim	Optional	Any comments pertaining to the contact.
Contact Id (Required for TT)	Character	30	Optional	User ID of person calling in. (Required to create a Trouble Ticket)
Receiving Operator	Character	30	Required	Person receiving and entering the call.
Category	Character	60	Optional	Category of the contact.
Contact Name	Character	30	Optional	Name of person calling in.
Contact Phone	Character	20	Optional	Phone number of person calling in.
Contact E-mail	Character	64	Optional	E-mail of person calling in.
Contact Home DAAC	Character	60	Optional	Home DAAC of person calling in.
Contact Organization	Character	60	Optional	Organization of person calling in.
Received Time	Date/Time	17	Optional	Date and Time the contact was first made. (mm/dd/yy hh:mm:ss)
Entered Time	Date/Time	17	System generated	Date and Time initial information is entered. (mm/dd/yy hh:mm:ss)
Modified-date	Date/Time	17	System generated	Date and time of last modification to this Contact Log. (mm/dd/yy hh:mm:ss)
Last-modified-by	Character	30	System generated	User ID of person that last modified this Contact Log. (mm/dd/yy hh:mm:ss)

* Note, the size of a field with a "selection" data type can vary and the size is automatically adjusted to the size of the item selected from the selection list.

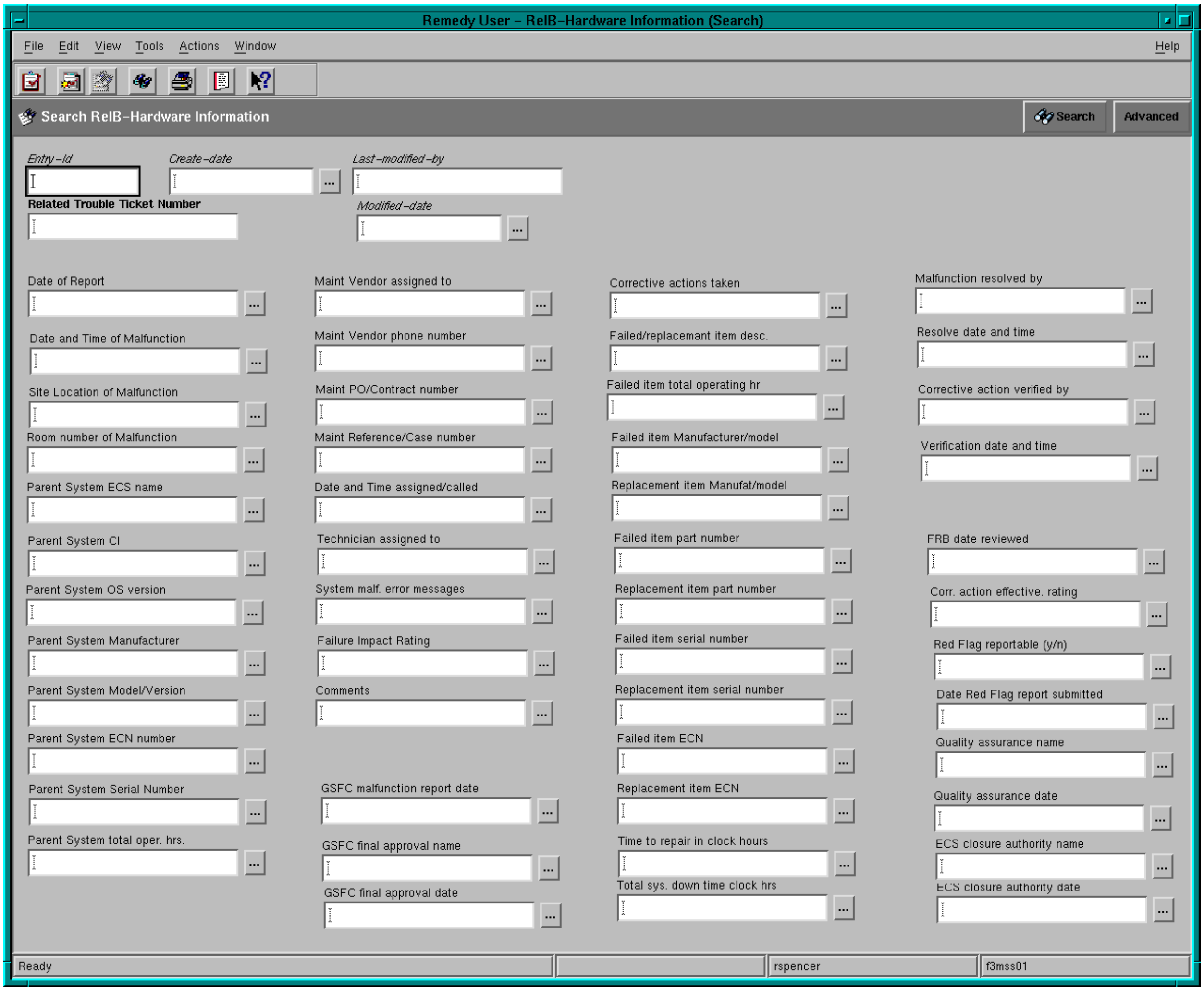
In addition to the fields described in the above table, the Contact Log Form provides the following buttons:

- **Set Contact Information**--inserts data for fields: Contact Name, Contact Phone, Contact Email, and Contact Home DAAC provided the user's login name and associated data is registered in the Remedy User Form.

- **Set Received Time**--inserts the current time in the Received Time field.

4.3.6.2.3 Remedy's User Tool (Hardware Information Form)

The Hardware Information Form GUI shown in Figure 4.3.6-4 is used to enter information about a particular piece of hardware that corresponds to a Trouble Ticket.



Remedy User – RelB–Hardware Information (Search)
File
Edit
View
Tools
Actions
Window
Help
Search RelB–Hardware Information
Search
Advanced
Entry–Id
Create–date
Last–modified–by
Related Trouble Ticket Number
Modified–date
Date of Report
Date and Time of Malfunction
Site Location of Malfunction
Room number of Malfunction
Parent System ECS name
Parent System CI
Parent System OS version
Parent System Manufacturer
Parent System Model/Version
Parent System ECN number
Parent System Serial Number
Parent System total oper. hrs.
Maint Vendor assigned to
Maint Vendor phone number
Maint PO/Contract number
Maint Reference/Case number
Date and Time assigned/called
Technician assigned to
System malf. error messages
Failure Impact Rating
Comments
GSFC malfunction report date
GSFC final approval name
GSFC final approval date
Corrective actions taken
Failed/replacemant item desc.
Failed item total operating hr
Failed item Manufacturer/model
Replacement item Manufat/model
Failed item part number
Replacement item part number
Failed item serial number
Replacement item serial number
Failed item ECN
Replacement item ECN
Time to repair in clock hours
Total sys. down time clock hrs
Malfunction resolved by
Resolve date and time
Corrective action verified by
Verification date and time
FRB date reviewed
Corr. action effective. rating
Red Flag reportable (y/n)
Date Red Flag report submitted
Quality assurance name
Quality assurance date
ECS closure authority name
ECS closure authority date
Ready
rspencer
f3mss01

Figure 4.3.6-4. Hardware Information Form GUI

Table 4.3.6-5 provides a description of the Hardware Information Form fields.

Table 4.3.6-5. Hardware Information Form Field Descriptions (1 of 3)

Field Name	Data Type	Size	Entry	Description
Entry-Id	Character	15	System generated	Entry ID of user.
Create-date	Date/Time	17	System generated	Date and time the entry was created at the present site. (format: mm/dd/yy hh:mm:ss)
Last-modified-by	Character	30	System generated	User ID of person that last modified the Hardware Information screen.
Related Trouble Ticket Number	Character	30	Required	ID of a related Trouble Ticket.
Modified-date	Date/Time	17	System generated	Date and time of last modification to Hardware Information screen. (format: mm/dd/yy hh:mm:ss)
Date of Report	Character	255	Optional	Date the problem was reported. (format: mm/dd/yy)
Date and Time of Malfunction	Character	255	Optional	Date and time the problem was noticed or approximate time of failure. (format: mm/dd/yy hh:mm:ss)
Site Location of Malfunction	Character	255	Optional	DAAC site where the problem occurred.
Room number of Malfunction	Character	255	Optional	DAAC room number where the problem occurred.
Parent System ECS name	Character	255	Optional	Site-specific ECS name of the parent system.
Parent System CI	Character	255	Optional	Associated Baseline control item ID of parent system.
Parent System OS version	Character	255	Optional	Operating system version of the parent system.
Parent System Manufacturer	Character	255	Optional	Name of parent system vendor.
Parent System Model/Version	Character	255	Optional	Model name and version numbers of the parent system.
Parent System ECN number	Character	255	Optional	Equipment control number of parent system.
Parent System Serial Number	Character	255	Optional	Serial number of parent system.
Parent System total oper. hrs.	Character	255	Optional	Number of cumulative hours the parent system has been in operation (since the last failure).
Maint Vendor assigned to	Character	255	Optional	Name of the vendor contracted to maintain the hardware.
Maint Vendor phone number	Character	255	Optional	Maintenance vendor point of contact phone number.
Maint PO/Contract number	Character	255	Optional	Purchase order and contract number for maintenance vendor.

Table 4.3.6-5. Hardware Information Form Field Descriptions (2 of 3)

Field Name	Data Type	Size	Entry	Description
Maint Reference/Case number	Character	255	Optional	Reference/case number assigned by the vendor for the hardware problem.
Date and Time assigned/called	Character	255	Optional	Date and time the maintenance vendor was called and notified of the problem. (format: mm/dd/yy hh:mm:ss)
Technician assigned to	Character	255	Optional	Name of vendor maintenance technician.
System malf. error messages	Character	255	Optional	Error messages provided by the system.
Failure Impact Rating	Character	255	Optional	Failure criticality or severity rating.
Comments	Character	255	Optional	Field to provide any additional comments.
Corrective actions taken	Character	255	Optional	Actions taken to resolve the problem.
Failed/replacement item desc.	Character	255	Optional	Description of the failed component and its replacement.
Failed item total operating hr	Character	255	Optional	Number of hours the failed component was used in operation.
Failed item Manufacturer/model	Character	255	Optional	Manufacturer name and model number of failed component.
Replacement item Manufat/model	Character	255	Optional	Manufacturer name and model # of replacement component.
Failed item part number	Character	255	Optional	Vendor part number of failed component.
Replacement item part number	Character	255	Optional	Vendor part number of replacement component.
Failed item serial number	Character	255	Optional	Serial number of failed component.
Replacement item serial number	Character	255	Optional	Serial number of replacement component.
Failed item ECN	Character	255	Optional	Equipment control number of failed component.
Replacement item ECN	Character	255	Optional	Equipment control number of replacement component.
Time to repair in clock hours	Character	255	Optional	Elapsed time (not including delays, e.g., technician waiting for part, vendor shipping time, etc.) in clock hours taken to troubleshoot and isolate the problem, to replace the component, and test and verify the fix.
Total sys. down time clock hrs	Character	255	Optional	Elapsed time in clock hours the system was down for repair (includes administrative and logistical delays).
Malfunction resolved by	Character	255	Optional	Name of person who resolved the problem.

Table 4.3.6-5. Hardware Information Form Field Descriptions (3 of 3)

Field Name	Data Type	Size	Entry	Description
Resolve date and time	Character	17	Optional	Date and time the problem was resolved. (format: mm/dd/yy hh:mm:ss)
Corrective action verified by	Character	255	Optional	Person who verified the problem has been resolved.
Verification date and time	Character	17	Optional	Date and time problem resolution was verified. (format: mm/dd/yy hh:mm:ss)
FRB date reviewed	Character	11	Optional	Date the Failure Review Board reviewed the problem and corrective action. (format: mm/dd/yy)
Corr. action effective rating	Character	255	Optional	Effectiveness rating assigned by Failure Review Board (FRB).
Red Flag reportable (y/n)	Character	1	Optional	Record of FRB determining if the problem meets Red Flag reporting criteria.
Date Red Flag report submitted	Character	8	Optional	If the problem meets Red Flag criteria, the date it was reported. (format: mm/dd/yy)
Quality assurance name	Character	255	Optional	Name of quality assurance person reviewing the problem and its resolution.
Quality assurance date	Character	8	Optional	Date the quality assurance person reviewed the problem and its resolution.
ECS closure authority name	Character	255	Optional	Name of person from the ECS review board who can close the Trouble Ticket.
ECS closure authority date	Character	8	Optional	Date the ECS review board closed the Trouble Ticket. (format: mm/dd/yy)
GSFC malfunction report date	Character	8	Optional	Date the malfunction was reported to the GSFC review board. (format: mm/dd/yy)
GSFC final approval name	Character	255	Optional	Name of person from GSFC review board who can approve problem and its resolution.
GSFC final approval date	Character	8	Optional	Date approved by GSFC review board. (format: mm/dd/yy)

4.3.6.2.4 Remedy's User Tool (RelB-Menu-Closing Codes Form)

The RelB-Menu-Closing Codes Form GUI, shown in Figure 4.3.6-5, is used to add, delete, or modify the list of closing code choices for the field, Closing Code.

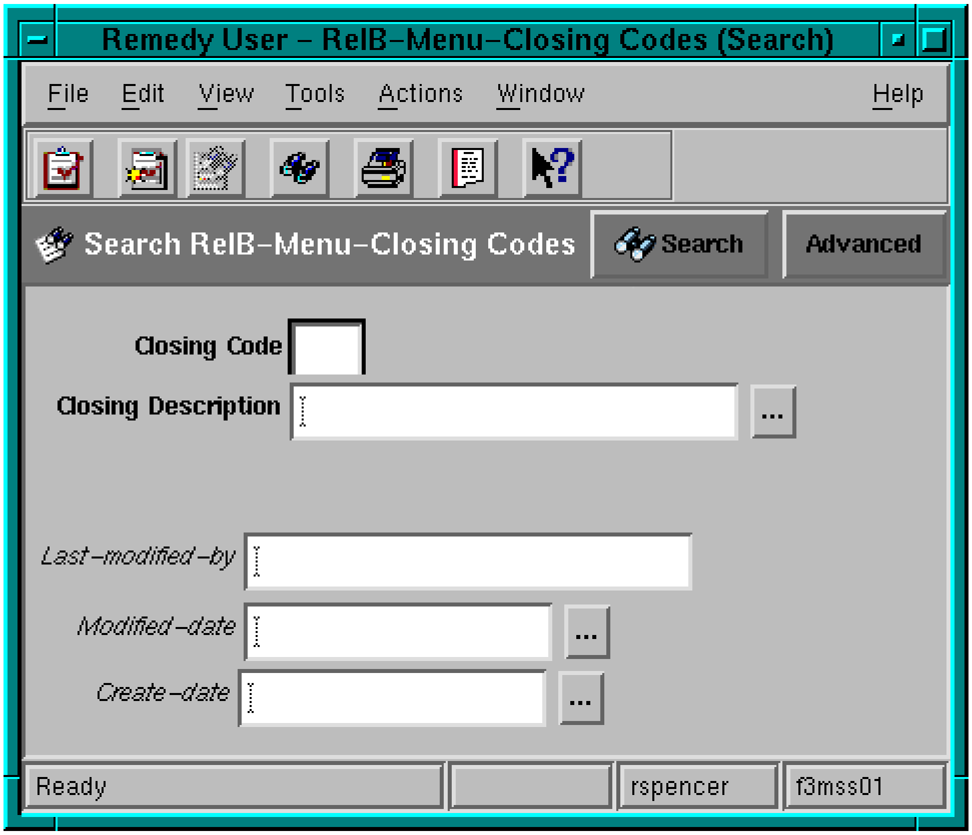


Figure 4.3.6-5. RelB-Menu-Closing Codes Form GUI

Table 4.3.6-6 provides a description of the RelB-Menu-Closing Code Form fields.

Table 4.3.6-6. RelB-Menu-Closing Codes Field Descriptions

Field Name	Data Type	Size	Entry	Description
Closing Code	Character	2	Required	Two letter code that corresponds with the Closing Description; this is where codes can be added, deleted or changed.
Closing Description	Character	128	Required	Problem summary.
Last-modified-by	Character	30	System generated	User ID of person that last modified the closing codes.
Modified-date	Date/Time	17	System generated	Date and time of last modification to closing codes. (mm/dd/yy hh:mm:ss)
Create-date	Date/Time	17	System generated	Date and time of the closing codes were created at the present site. (mm/dd/yy hh:mm:ss)

4.3.6.2.5 Remedy's User Tool (RelB-Menu-Hardware Resources Form)

The RelB-Menu Hardware Resources Form GUI, shown in Figure 4.3.6-6, is where one adds, deletes, or modifies the hardware resource choices for the field, Hardware Resource.

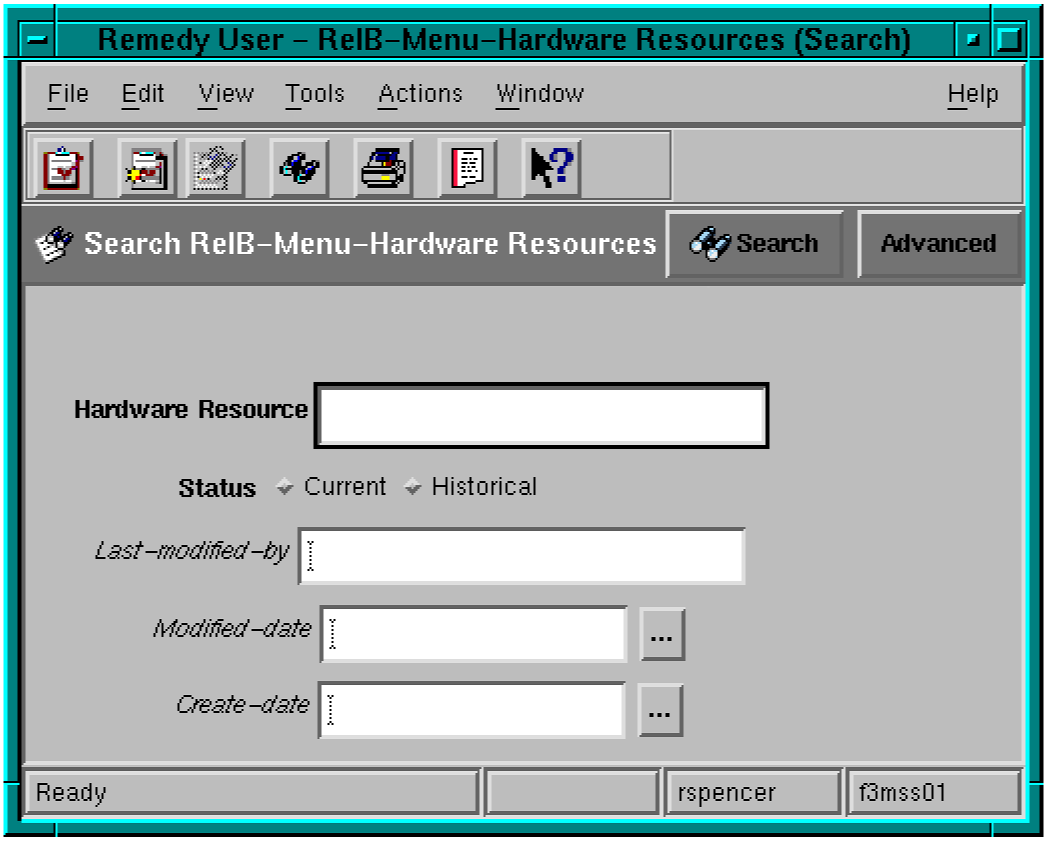


Figure 4.3.6-6. Tool RelB-Menu-Hardware Resources Form GUI

Table 4.3.6-7 provides a description of the RelB-Menu-Hardware Resources Form fields.

Table 4.3.6-7. RelB-Menu-Hardware Resources Form Field Descriptions (1 of 2)

Field Name	Data Type	Size	Entry	Description
Hardware Resource	Character	30	Required	Hardware resource to be added, deleted or modified.
Last-modified-by	Character	30	System generated	User ID of person that last modified the hardware resources.

Table 4.3.6-7. RelB-Menu-Hardware Resources Form Field Descriptions (2 of 2)

Field Name	Data Type	Size	Entry	Description
Modified-date	Date/Time	17	System generated	Date and time of last modification to hardware resources. (mm/dd/yy hh:mm:ss)
Create-date	Date/Time	17	System generated	Date and time the hardware resource was created at the present site. (mm/dd/yy hh:mm:ss)

* Note, the size of a field with a "selection" data type can vary and the size is automatically adjusted to the size of the item selected from the selection list.

In addition to the fields described in the above table, the RelB-Trouble Tickets form provides the following buttons (active links):

- Status – Current or historical status for this hardware resource.

4.3.6.2.6 Remedy's User Tool (RelB-Menu-Key Words Form)

The RelB-Menu-Key Words Form GUI, shown in Figure 4.3.6-7, is used to add, delete, or modify the key word choices for the field, Key Word.

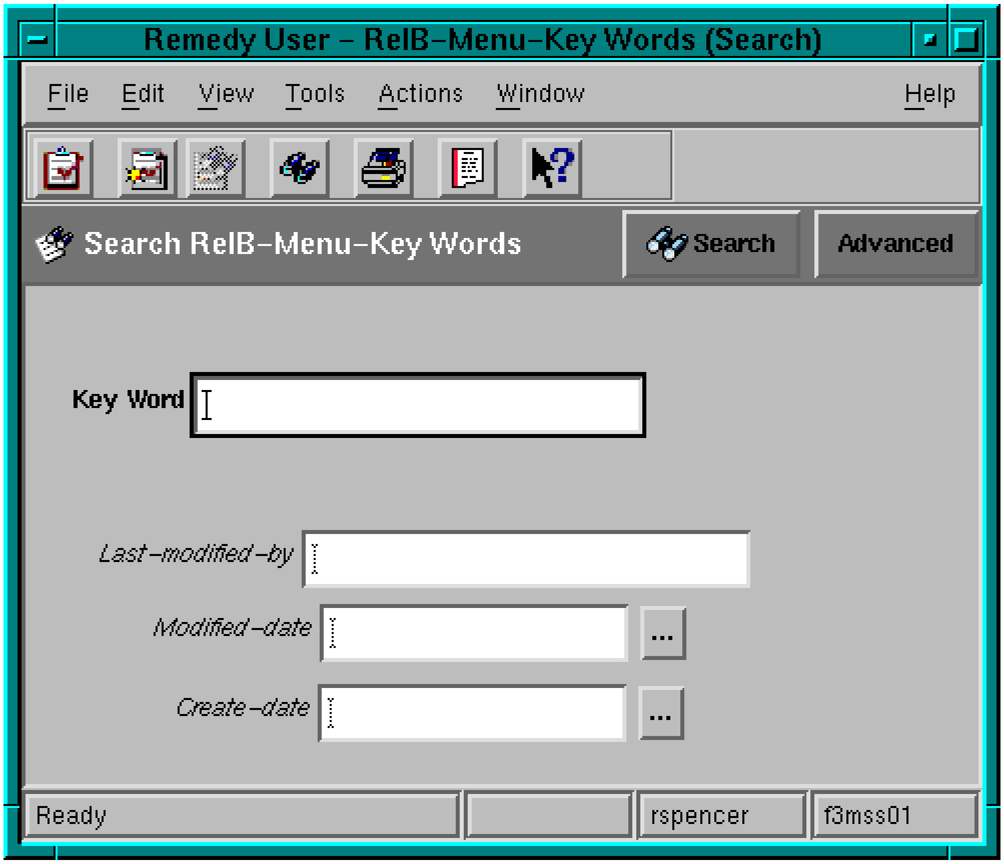


Figure 4.3.6-7. RelB-Menu-Key Words Form GUI

Table 4.3.6-8 provides a description of the RelB-Menu-Key Words Form fields.

Table 4.3.6-8. RelB-Menu-Key Words Form Field Descriptions

Field Name	Data Type	Size	Entry	Description
Key Word	Character	30	Required	Key word for the Trouble Ticket; this is where key words can be added, deleted or modified.
Last-modified-by	Character	30	System generated	User ID of person that last modified the Key Words.
Modified-date	Date/time	17	System generated	Date and time of last modification to Key Words. (mm/dd/yy hh:mm:ss)
Create-date	Date/time	17	System generated	Date and time the Key Words were created at the present site. (mm/dd/yy hh:mm:ss)

4.3.6.2.7 Remedy's User Tool (RelB-Menu-Problem Type Form)

The RelB-Menu-Problem Type Form GUI, shown in Figure 4.3.6-8, is used to add, delete, or modify the problem type choices for the field, Problem Type.

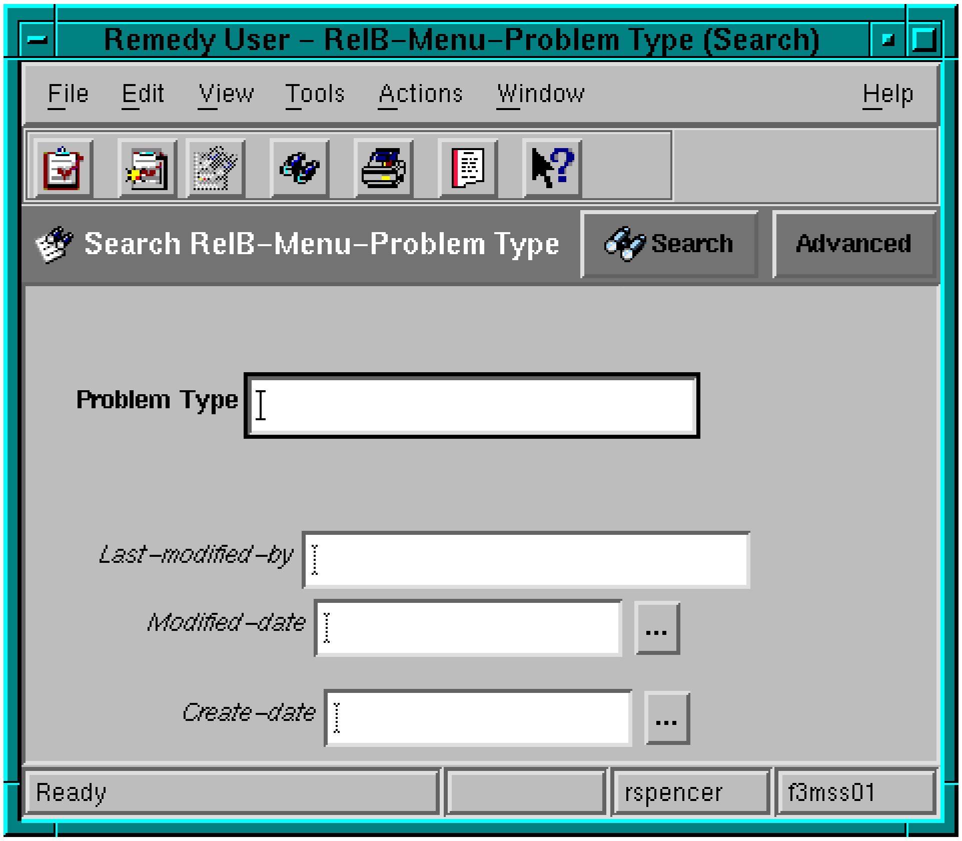


Figure 4.3.6-8. RelB-Menu-Problem Type Form GUI

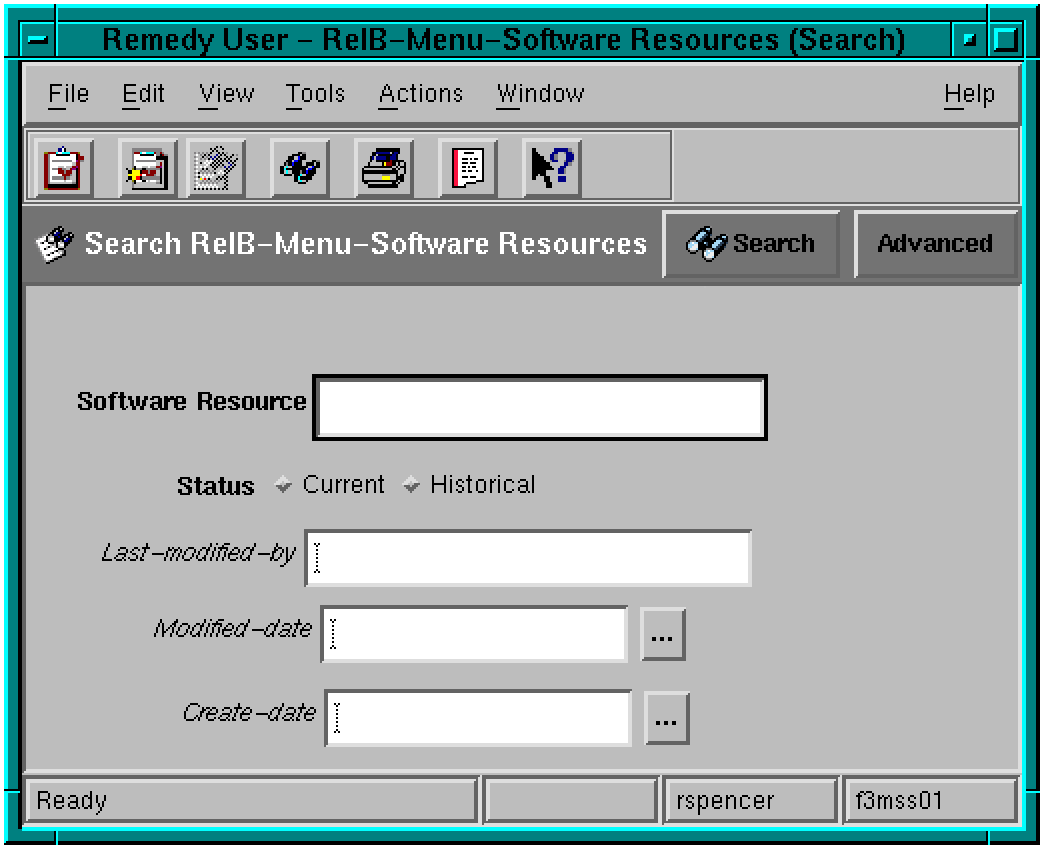
Table 4.3.6-9 provides a description of the RelB-Menu-Problem Type Form fields.

Table 4.3.6-9. RelB-Menu-Problem Type Form Field Descriptions

Field Name	Data Type	Size	Entry	Description
Problem Type	Character	30	Required	Problem type of the Trouble Ticket; this is where problem types can be added, deleted or modified.
Last-modified-by	Character	30	System generated	User ID of person that last modified the Problem Type.
Modified-date	Date/Time	17	System generated	Date and time of last modification to Problem Type. (mm/dd/yy hh:mm:ss)
Create-date	Date/Time	17	System generated	Date and time the Problem Type was created at the present site. (mm/dd/yy hh:mm:ss)

4.3.6.2.8 Remedy's User Tool (RelB-Menu-Software Resources Form)

The RelB-Menu-Software Resources Form GUI, shown in Figure 4.3.6-9, is used to add, delete, or modify the software resource choices for the field, Software Resource.


Remedy User – RelB–Menu–Software Resources (Search)

Ready

rspencer

f3mss01

Figure 4.3.6-9. RelB-Menu-Software Resources Form GUI

Table 4.3.6-10 provides a description of the RelB-Menu-Software Resource Form fields.

Table 4.3.6-10. RelB-Menu-Software Resources Form Field Descriptions

Field Name	Data Type	Size	Entry	Description
Software Resource	Character	30	Required	Software resource to be added, deleted or modified.
Last-modified-by	Character	30	System generated	User ID of person that last modified the software resources.
Modified-date	Date/Time	17	System generated	Date and time of last modification to software resources. (mm/dd/yy hh:mm:ss)
Create-date	Date/Time	17	System generated	Date and time the software resources were created at the present site. (mm/dd/yy hh:mm:ss)

* Note, the size of a field with a "selection" data type can vary and the size is automatically adjusted to the size of the item selected from the selection list.

In addition to the fields described in the above table, the RelB-Trouble Tickets form provides the following buttons (active links):

- Status – Current or historical status of this software resource.

4.3.6.2.9 Remedy's User Tool (RelB-TT-Sites Form)

The RelB-TT-Sites Form GUI, shown in Figure 4.3.6-10, indicates the site name and email address to be used in forwarding.

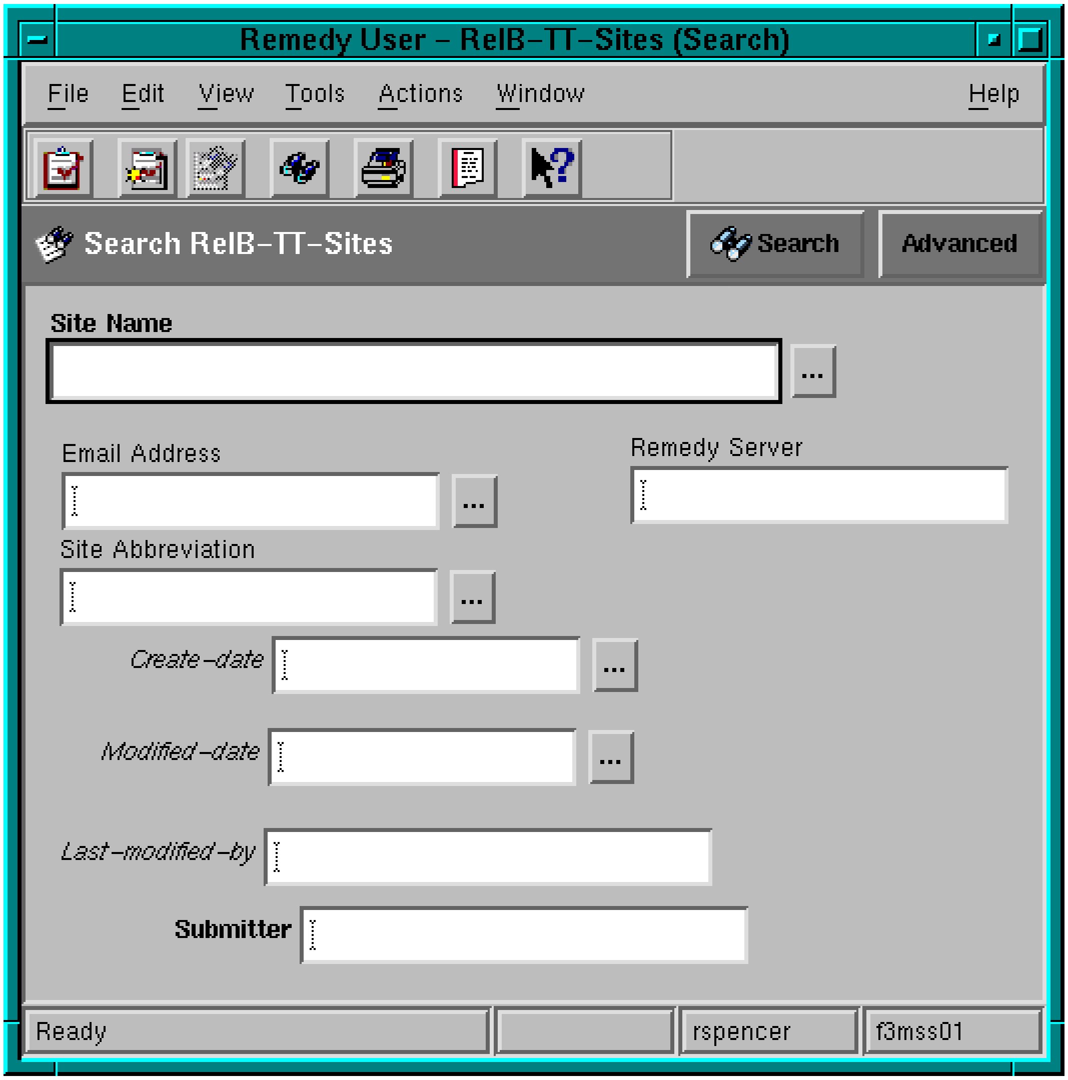


Figure 4.3.6-10. RelB-TT-Sites Form GUI

Table 4.3.6-11 provides a description of the RelB-TT-Sites form fields.

Table 4.3.6-11. RelB-TT-Sites Form Field Descriptions (1 of 2)

Field Name	Data Type	Size	Entry	Description
Site Name	Character	128	Required	Name of EOS Site.
Email Address	Character	255	Optional	E-mail address of EOS Site.
Remedy Server	Character	55	Optional	Name of server at site where Remedy is installed.

Table 4.3.6-11. RelB-TT-Sites Form Field Descriptions (2 of 2)

Field Name	Data Type	Size	Entry	Description
Site Abbreviation	Character	255	Optional	Abbreviation of site name.
Create-date	Date/Time	17	System generated	Date and time the RelB-TT-Sites were created at the present site. (mm/dd/yy hh:mm:ss)
Modified-date	Date/Time	17	System generated	Date and time of last modification to RelB-TT-Sites. (mm/dd/yy hh:mm:ss)
Last-modified-by	Character	30	System generated	User ID of person that last modified the RelB-TT-Sites.
Submitter	Character	30	Required	User ID.

4.3.6.2.10 Remedy's User Tool (RelB-TT-Times Form)

The RelB-TT-Times Form GUI (Figure 4.3.6-11) is used to indicate escalation times. An escalation time is an amount of time given for some action to occur on a trouble ticket. If action is not initiated within the specified time, Remedy is set up to notify the assigned support staff member or maybe a higher level manager of the presence of the trouble ticket so that he/she may initiate immediate action.

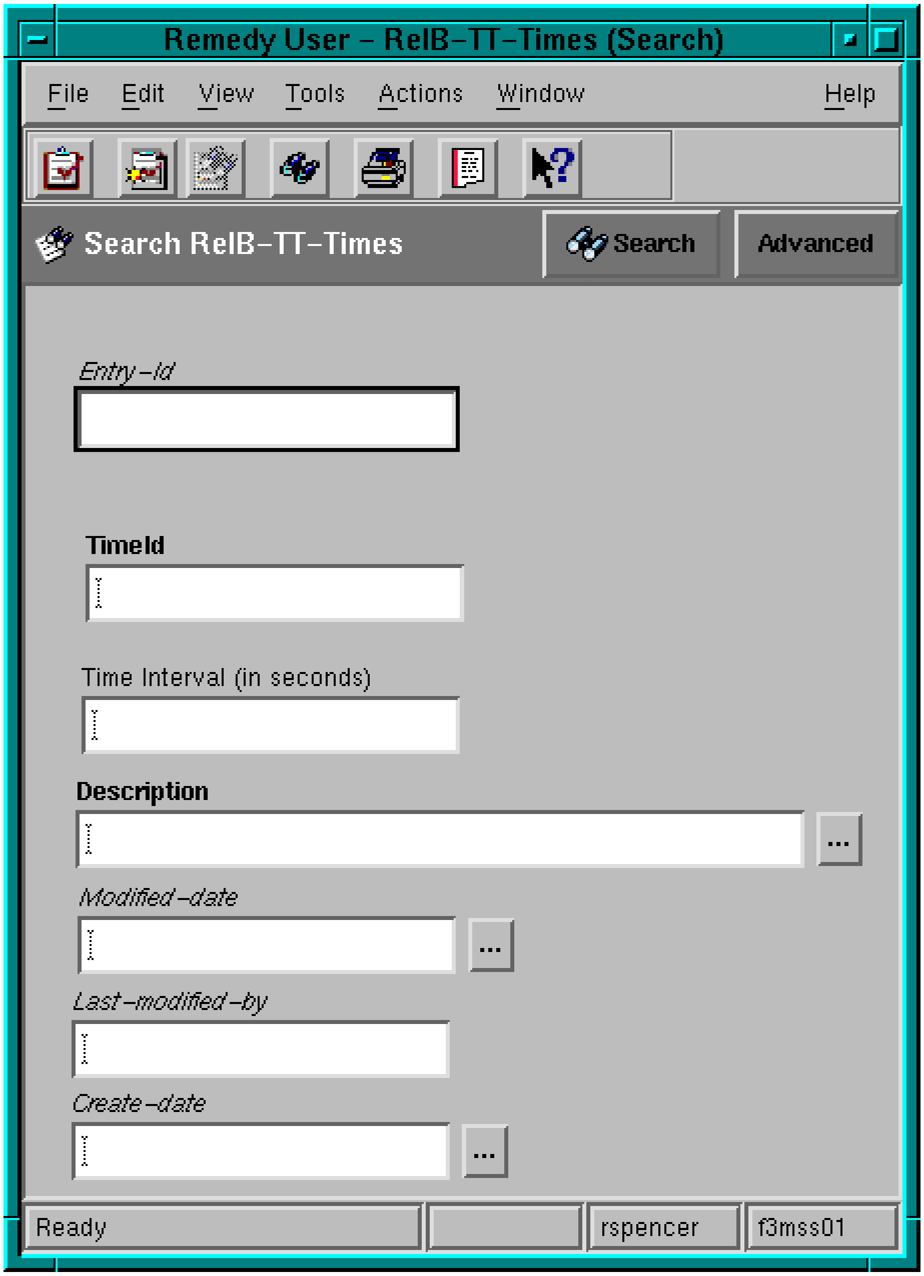


Figure 4.3.6-11. RelB-TT-Times Form GUI

Table 4.3.6-12 provides a description of the RelB-TT-Times Form fields.

Table 4.3.6-12. RelB-TT-Times Form Field Descriptions

Field Name	Data Type	Size	Entry	Description
Entry-Id	Character	15	System generated	Entry ID of Time entry.
TimeId	Character	30	Required	Escalation ID of Time entry.
Time Interval (in seconds)	Integer	4	Optional	Time interval (in seconds) for escalation to take place.
Description	Character	128	Required	What escalation this time corresponds to.
Modified-date	Date/Time	17	System generated	Date and time of last modification to RelB-TT-Sites. (mm/dd/yy hh:mm:ss)
Last-modified-by	Character	30	System generated	User ID of person that last modified the RelB-TT-Sites.
Create-date	Date/Time	17	System generated	Date and time the RelB-TT-Sites were created at the present site. (mm/dd/yy hh:mm:ss)

4.3.6.2.11 Remedy's Admin Tool GUI

The Remedy Administrator Tool is the tool one uses to create, modify, and delete Remedy objects (e.g., forms and menus). Figure 4.3.6-12 shows the main Administrator Tool GUI and its starting screen, the server window, and the workflow objects categories.

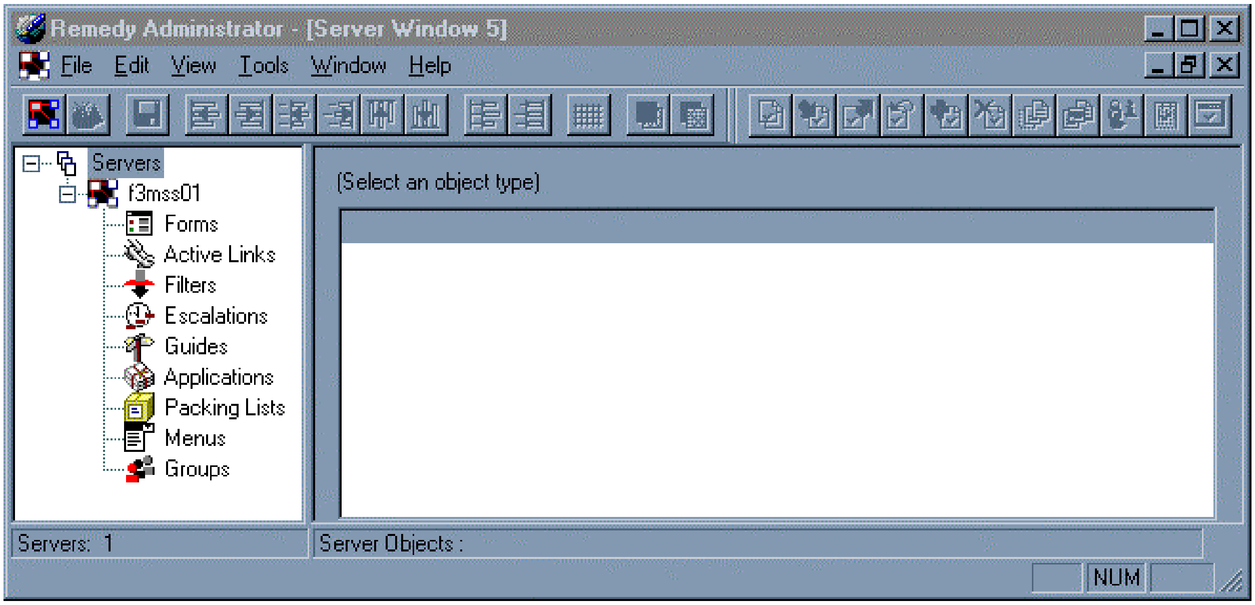


Figure 4.3.6-12. Admin Tool GUI

Table 4.3.6-13 provides a description of the Admin Tool GUI Workflow objects. For more information on these options, see the Remedy Administrator Tutorial using the Administrator

Tool's Help menu or see its standard online help that is accessed by clicking the Start button and then selecting Programs → Action Request System → Remedy Administrator Help.

Table 4.3.6-13. Admin Tool GUI, Workflow Object Descriptions

Workflow Object	Size	Entry	Description
Forms	Variable	System Generated	List of available forms.
Active links	Variable	System Generated	List of available active links.
Filters	Variable	System Generated	List of available filters.
Escalations	Variable	System Generated	List of available escalations.
Guides	Variable	System Generated	List of available guides.
Applications	Variable	System Generated	List of available applications.
Packing lists	Variable	System Generated	List of available packing lists.
Menus	Variable	System Generated	List of available menus.
Groups	Variable	System Generated	List of available groups.

4.3.6.2.12 Remedy's Notification Tool

The Notification Tool can be used to alert users when a trouble ticket has been submitted or when a trouble ticket progresses from one status (e.g., new, assigned, etc.) to another status. Figure 4.3.6-13 shows the Notification Tool screen.

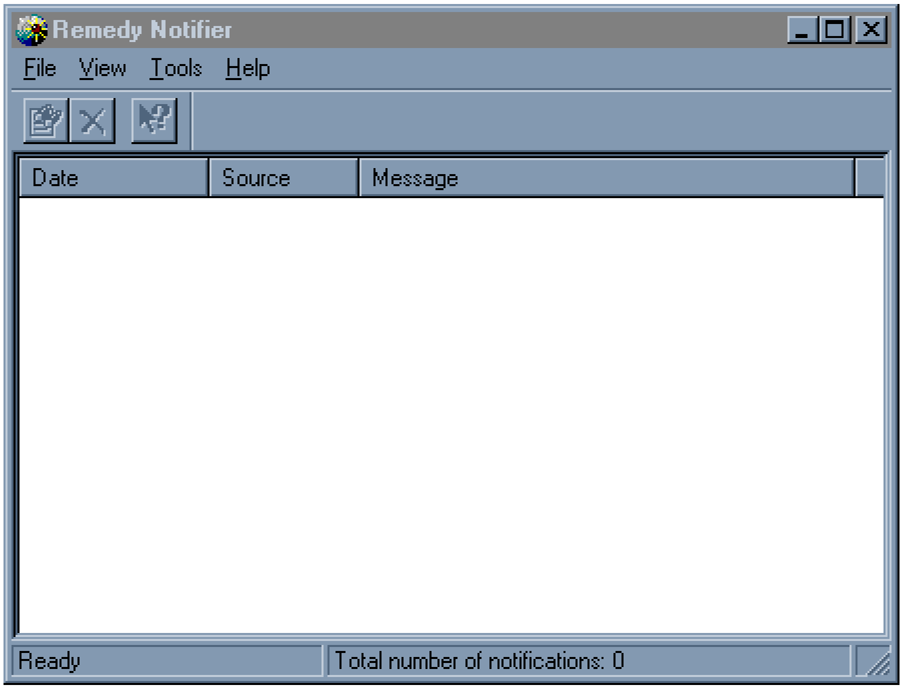


Figure 4.3.6-13. Notification Tool GUI

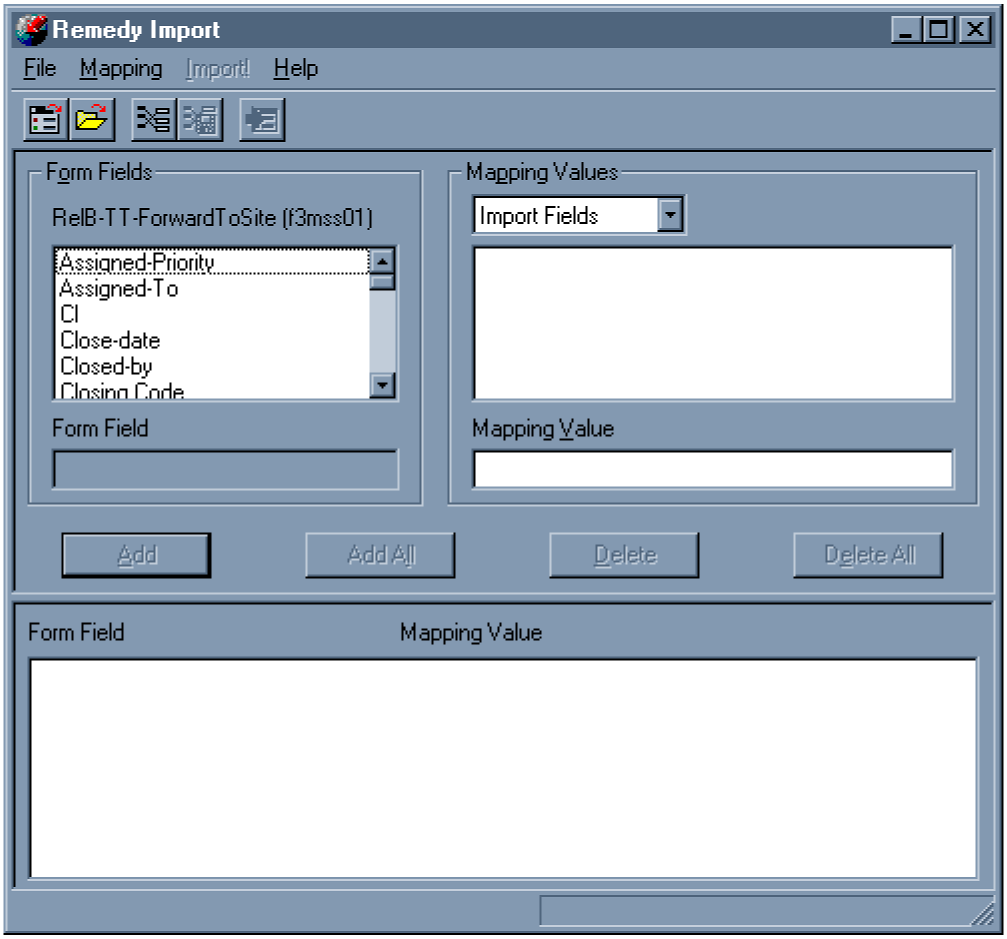
Table 4.3.6-14 provides a description of the Notification fields. For more information on the Notification Tool, see its standard online help that is accessed by clicking the Start button and then selecting Programs → Action Request System → Remedy Notifier Help.

Table 4.3.6-14. Notification Field Descriptions

Field Name	Data Type	Size	Entry	Description
Date	Character	Variable	System generated	Timestamp of the notification.
Source	Character	Variable	System generated	Source of the trouble ticket.
Message	Character	Variable	System generated	The short description from the trouble ticket.
Total number of notifications	Integer	Variable	System generated	Current count of the total number of notifications assigned.

4.3.6.2.13 Remedy's Import Tool

The Import Tool enables one to load data into a form from a data file previously created in the User Tool, a third party report writer, or another application. Figure 4.3.6-14 shows the Import Tool screen.



RelB-TT-ForwardToSite (f3mss01)

Assigned-Priority
Assigned-To
CI
Close-date
Closed-by
Closing Code

Import Fields

Figure 4.3.6-14. Import Tool GUI

Table 4.3.6-15 provides a description of the Import fields. For more information on the Import Tool, see its standard online help that is accessed by clicking the Start button and then selecting Programs → Action Request System → Remedy Import Help.

Table 4.3.6-15. Import Field Descriptions

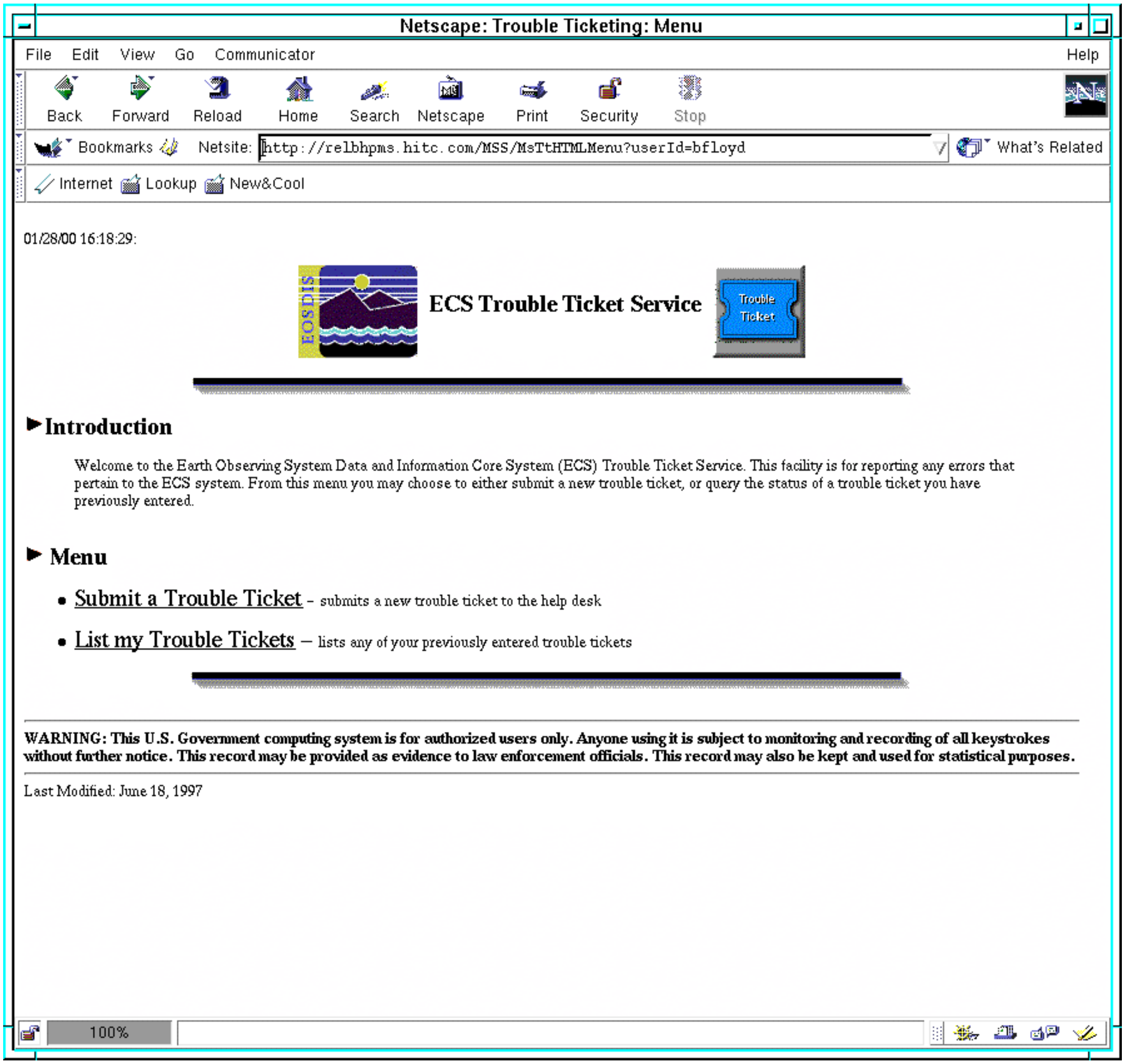
Field Name	Data Type	Size	Entry	Description
Form Fields	Character	Variable	System generated	Displays fields available in the selected form.
Mapping Values	Character	Variable	User Selected	Displays mapping values available for the selected form fields.
Form Field / Mapping Value	Character	Variable	System generated	Displays the chosen import / form mappings.

In addition to the fields described in Table 4.3.6-15, the Import Tool GUI has the following buttons:

- **Add**-- create a mapping between a destination form field and one of the import fields.
- **Delete**--delete the selected mapping in the mapping list.
- **Add All**--map all of the import fields directly to the destination form fields with the same field ID or field name.
- **Delete All**--delete all of the mappings from the mapping list.

4.3.6.2.14 Remedy's End-User Trouble Ticketing HTML Main Screen

The HTML Trouble Ticket main screen ("ECS Trouble Ticketing: Menu"), shown in Figure 4.3.6-15, provides an introduction on how to use the Trouble Ticketing HTML, and is used by User Services personnel to go to either the Submit page or List page. To get to the Trouble Ticket screen, execute Netscape and enter the URL for the Trouble Ticket web interface (e.g., enter: http://relbhpms.hitc.com/MSS/MsTtHTMLMenu?userId=bfloyd).



Netscape: Trouble Ticketing: Menu
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Netsite: http://relbhpms.hitc.com/MSS/MsTtHTMLMenu?userId=bfloyd What's Related
Internet Lookup New&Cool
EOSDIS
Trouble Ticket
100%

01/28/00 16:18:29:

ECS Trouble Ticket Service

Introduction

Welcome to the Earth Observing System Data and Information Core System (ECS) Trouble Ticket Service. This facility is for reporting any errors that pertain to the ECS system. From this menu you may choose to either submit a new trouble ticket, or query the status of a trouble ticket you have previously entered.

Menu

- Submit a Trouble Ticket – submits a new trouble ticket to the help desk
- List my Trouble Tickets – lists any of your previously entered trouble tickets

WARNING: This U.S. Government computing system is for authorized users only. Anyone using it is subject to monitoring and recording of all keystrokes without further notice. This record may be provided as evidence to law enforcement officials. This record may also be kept and used for statistical purposes.

Last Modified: June 18, 1997

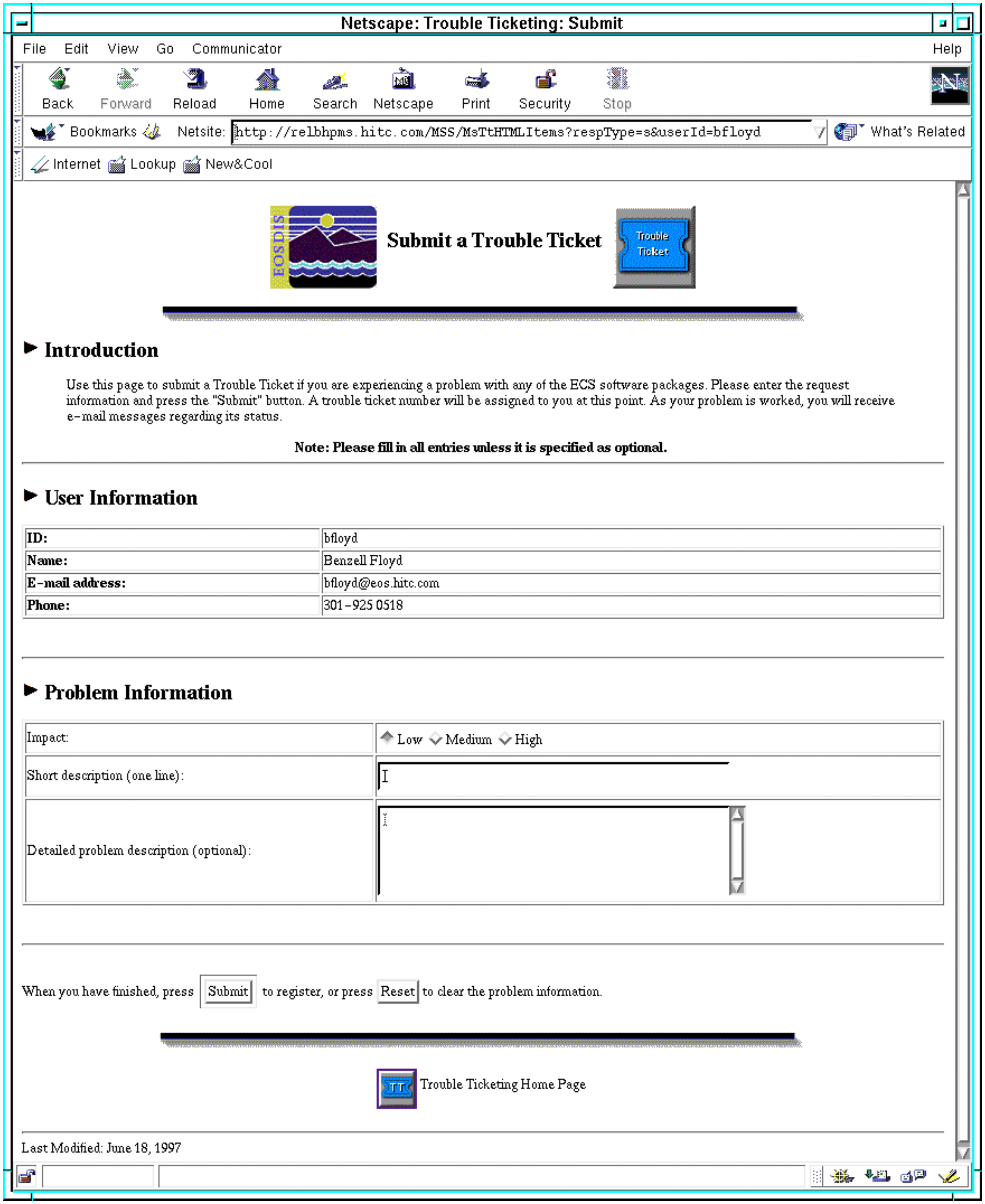
Figure 4.3.6-15. ECS Trouble Ticketing: (iPlanet) Menu GUI

Selecting **Submit a Trouble Ticket** brings up the Trouble Ticketing: Submit GUI described in Section 4.3.6.2.15.

Selecting **List [username] Trouble Tickets** brings up the Trouble Ticketing: List GUI described in Section 4.3.6.2.17.

4.3.6.2.15 Remedy's End-User Trouble Ticketing HTML Submit GUI

User Services personnel to submit a Trouble Ticket use the HTML Trouble Ticket Submit GUI, shown in Figure 4.3.6-16.


Netscape: Trouble Ticketing: Submit
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Netsite: http://relbhpms.hitc.com/MSS/MsTtHTMLItems?respType=s&userId=bfloyd What's Related
Internet Lookup New&Cool
EOSDIS
Trouble Ticket

Submit a Trouble Ticket

► Introduction

Use this page to submit a Trouble Ticket if you are experiencing a problem with any of the ECS software packages. Please enter the request information and press the "Submit" button. A trouble ticket number will be assigned to you at this point. As your problem is worked, you will receive e-mail messages regarding its status.

Note: Please fill in all entries unless it is specified as optional.

► User Information

ID:	bfloyd
Name:	Benzell Floyd
E-mail address:	bfloyd@eos.hitc.com
Phone:	301-925 0518

► Problem Information

Impact:	● Low ○ Medium ○ High
Short description (one line):	
Detailed problem description (optional):	

When you have finished, press Submit to register, or press Reset to clear the problem information.

Trouble Ticketing Home Page

Last Modified: June 18, 1997

Figure 4.3.6-16. Trouble Ticket HTML Submit GUI

Table 4.3.6-16 provides a description of the Trouble Ticket HTML Submit fields.

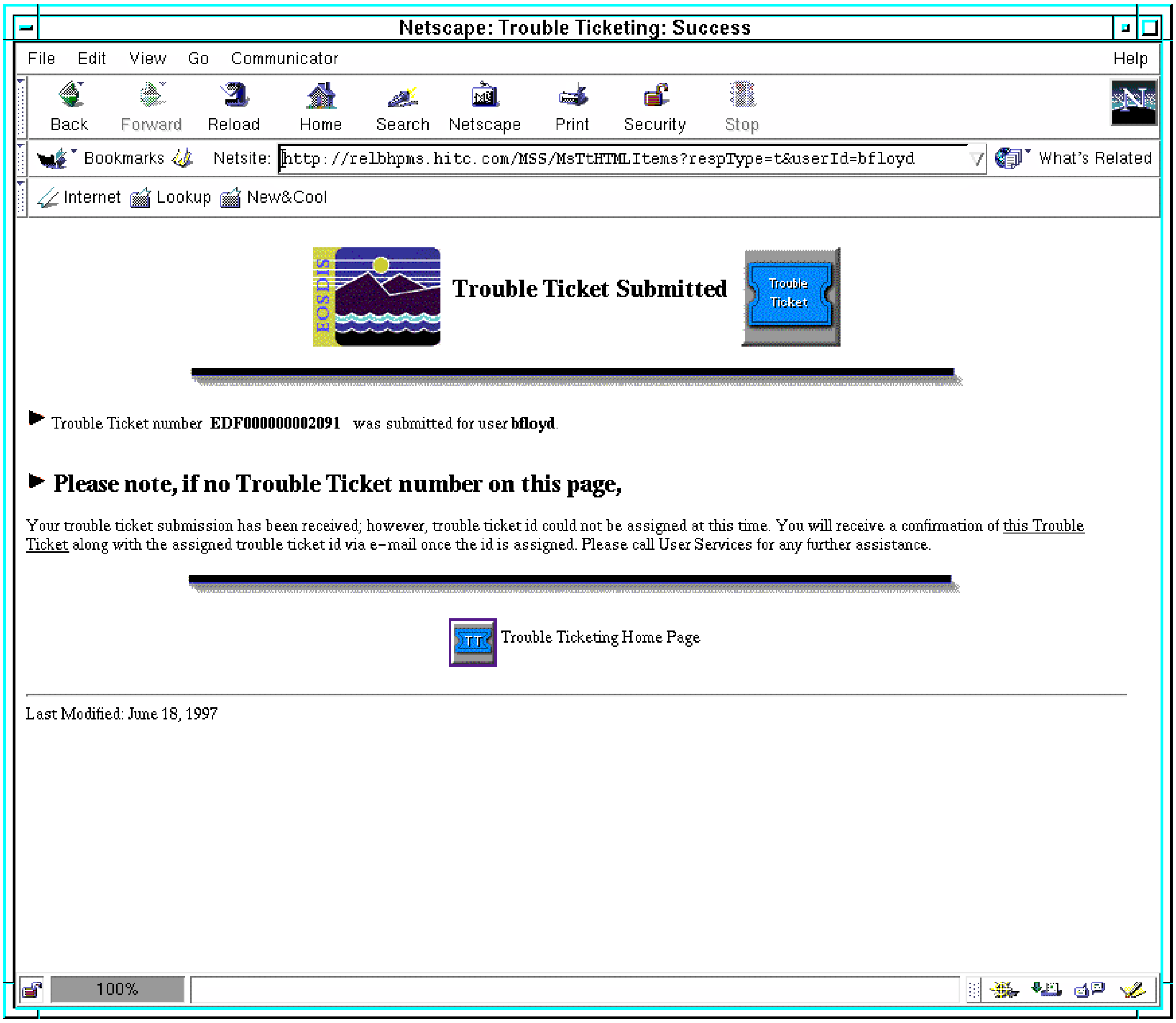
Table 4.3.6-16. Trouble Ticket HTML Submit Screen Field Descriptions

Field Name	Data Type	Size	Entry	Description
ID	Character	30	System generated	Submitter Id.
Name	Character	30	System generated	Submitter Name.
E-mail address	Character	64	System generated	Submitter E-mail Address.
Phone	Character	30	System generated	Submitter Phone Number.
Impact	Selection	4	Required	Impact to Submitter.
Short description	Character	125	Required	Short description of problem (one line).
Detailed problem description	Character	245	Optional	Long description of problem.

When the information is completed, the user can submit the Trouble Ticket by clicking on the **Submit** button on the lower half of the screen. Clicking on the Reset button can clear the Problem Information Fields. The user also has the choice of returning to the Trouble Ticketing Homepage by clicking on the icon at the bottom of the page.

4.3.6.2.16 Remedy's End-User Trouble Ticketing HTML Success GUI

User Services personnel to confirm successful submission and report Trouble Ticket Id use the HTML Trouble Ticket Success screen, shown in Figure 4.3.6-17.


Netscape: Trouble Ticketing: Success
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Netsite: http://relbhpms.hitc.com/MSS/MsTtHTMLItems?respType=t&userId=bfloyd
What's Related
Internet Lookup New&Cool
EOSDIS
Trouble Ticket
TT
100%

Trouble Ticket Submitted

▶ Trouble Ticket number **EDF000000002091** was submitted for user **bfloyd**.

▶ **Please note, if no Trouble Ticket number on this page,**

Your trouble ticket submission has been received; however, trouble ticket id could not be assigned at this time. You will receive a confirmation of <u>this Trouble Ticket</u> along with the assigned trouble ticket id via e-mail once the id is assigned. Please call User Services for any further assistance.

Trouble Ticketing Home Page

Last Modified: June 18, 1997

Figure 4.3.6-17. Trouble Ticket HTML Success GUI

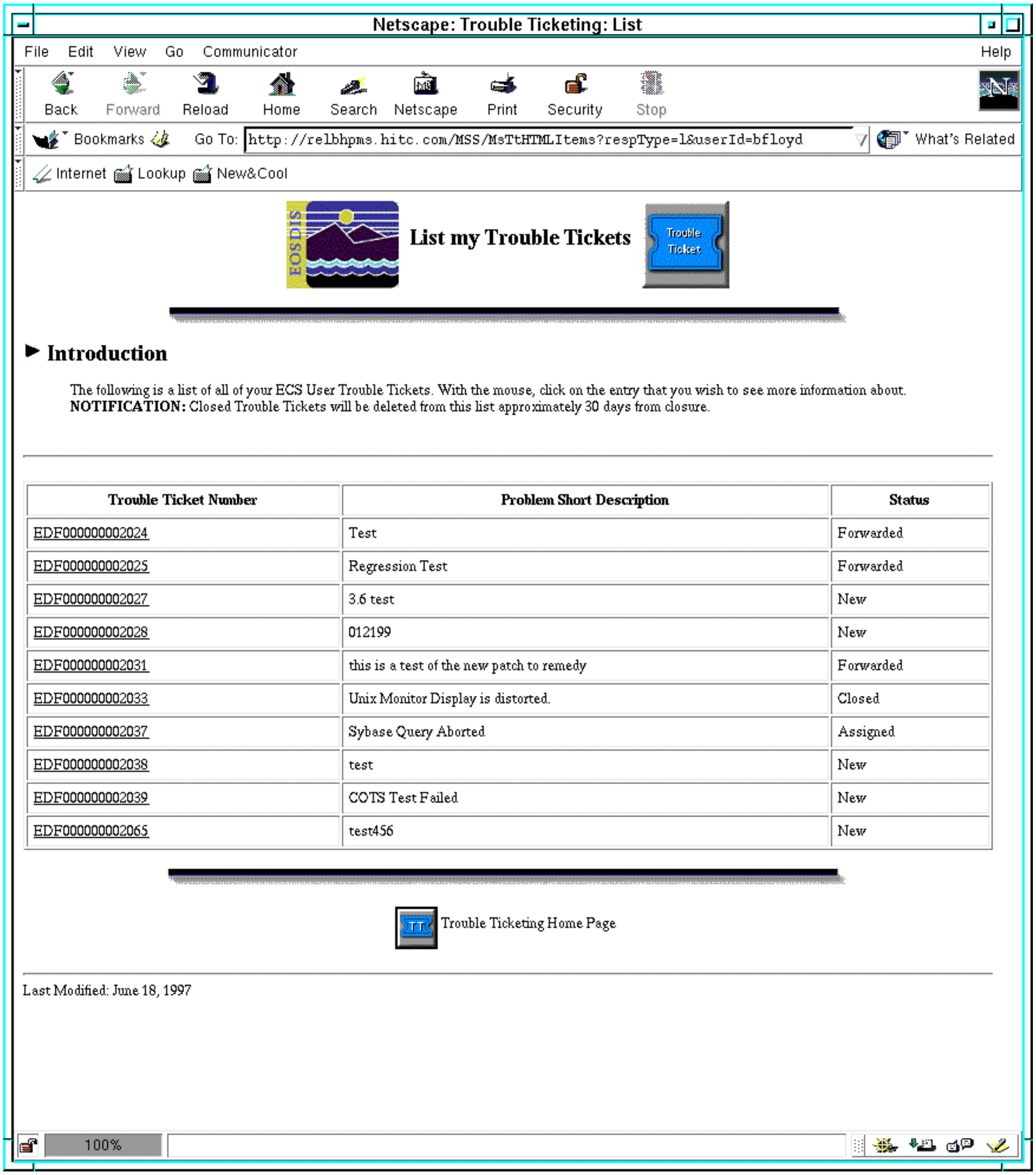
From this screen, the user is provided with the following information/options:

- The Trouble Ticket was successfully submitted, Trouble Ticket identification number and who submitted it
- Select "<u>this Trouble Ticket" in the second remark and </u>the contents of the trouble ticket is displayed via the Trouble Ticket Detailed Screen (see Section 4.3.6.2.18).

The user also has the choice of returning to the Trouble Ticketing Homepage (Section 4.3.6.2.14) by clicking on the icon at the bottom of the screen.

4.3.6.2.17 Remedy's End-User Trouble Ticketing HTML List GUI

User Services personnel use the HTML Trouble Ticket List GUI, shown in Figure 4.3.6-18 to List Trouble Tickets for a user and links the listed Trouble Ticket Number to the Trouble Ticket Detailed GUI (see Section 4.3.6.2.18).



List my Trouble Tickets

Introduction

The following is a list of all of your ECS User Trouble Tickets. With the mouse, click on the entry that you wish to see more information about.
NOTIFICATION: Closed Trouble Tickets will be deleted from this list approximately 30 days from closure.

Trouble Ticket Number	Problem Short Description	Status
EDF000000002024	Test	Forwarded
EDF000000002025	Regression Test	Forwarded
EDF000000002027	3.6 test	New
EDF000000002028	012199	New
EDF000000002031	this is a test of the new patch to remedy	Forwarded
EDF000000002033	Unix Monitor Display is distorted.	Closed
EDF000000002037	Sybase Query Aborted	Assigned
EDF000000002038	test	New
EDF000000002039	COTS Test Failed	New
EDF000000002065	test456	New

TT Trouble Ticketing Home Page

Last Modified: June 18, 1997

Figure 4.3.6-18. Trouble Ticket HTML List GUI

Table 4.3.6-17 provides a description of the Trouble Ticket HTML List fields.

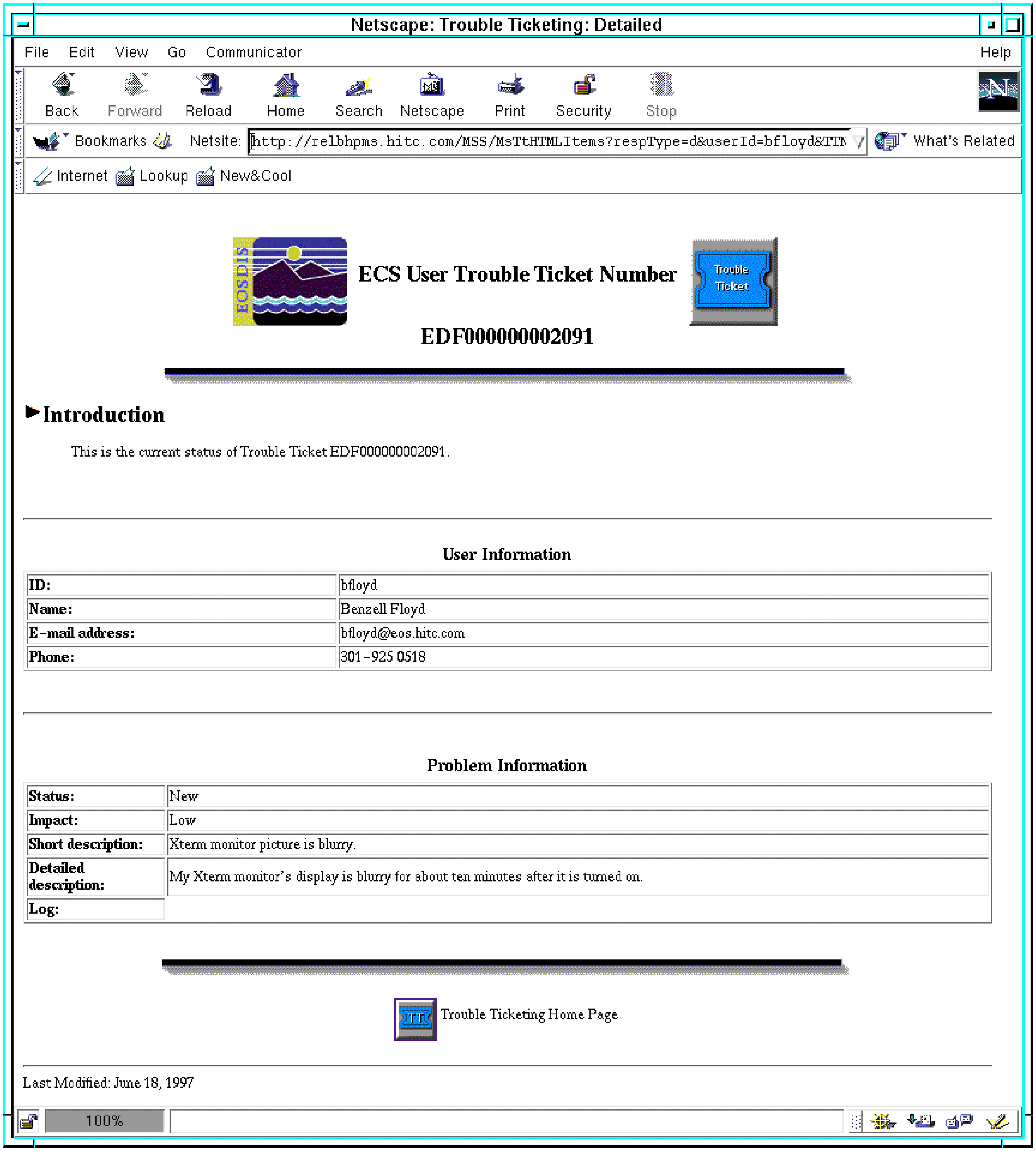
Table 4.3.6-17. Trouble Ticket HTML List Field Descriptions

Field Name	Data Type	Size	Entry	Description
Trouble Ticket Number	Character	15	System generated	Trouble Ticket ID.
Problem Short Description	Character	125	System generated	Short Description of Problem.
Status	Character	20	System generated	Status of Trouble Ticket (e.g. new—trouble ticket entered into the database but has not been evaluated; assigned--trouble ticket has been assigned to someone for resolution, etc.).

The user can click on one of the listed trouble ticket numbers and activate a display of the trouble ticket's content (Section 4.3.6.2.18). The user also has the choice of returning to the Trouble Ticketing Homepage by clicking on the icon at the bottom of the screen.

4.3.6.2.18 Remedy's End-User Trouble Ticketing HTML Detailed GUI

User Services personnel to see a more detailed output of a Trouble Ticket use the HTML Trouble Ticket Detailed GUI, shown in Figure 4.3.6-19.



Netscape: Trouble Ticketing: Detailed
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Netsite: http://relbhpms.hitc.com/MSS/MsTtHTMLItems?respType=d&userId=bfloyd&TTN
What's Related
Internet Lookup New&Cool
EOSDIS
Trouble Ticket
TT
100%

ECS User Trouble Ticket Number

EDF000000002091

▶ Introduction

This is the current status of Trouble Ticket EDF000000002091.

User Information

ID:	bfloyd
Name:	Benzell Floyd
E-mail address:	bfloyd@eos.hitc.com
Phone:	301-925 0518

Problem Information

Status:	New
Impact:	Low
Short description:	Xterm monitor picture is blurry.
Detailed description:	My Xterm monitor's display is blurry for about ten minutes after it is turned on.
Log:	

Trouble Ticketing Home Page

Last Modified: June 18, 1997

Figure 4.3.6-19. Trouble Ticket HTML Detailed GUI

Table 4.3.6-18 provides a description of the Trouble Ticket HTML Detailed fields.

Table 4.3.6-18. Trouble Ticket HTML Detailed Field Descriptions

Field Name	Data Type	Size	Entry	Description
ID	Character	30	System generated	Submitter Id.
Name	Character	30	System generated	Submitter Name.
E-mail address	Character	64	System generated	Submitter E-mail Address.
Phone	Character	30	System generated	Submitter Phone Number.
Status	Selection	4	System generated	Status of Trouble Ticket.
Impact	Selection	4	System generated	Impact to Submitter.
Short description	Character	125	System generated	Short description of problem.
Detailed description	Character	245	System generated	Long description of problem.
Log	Character	Unlim.	System generated	Diary of problem resolution.

The user also has the choice of returning to the Trouble Ticketing Homepage by clicking on the icon at the bottom of the screen.

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 ReadMe file for each product. To find the ReadMe file for Remedy, use the XRP Baseline Manager to determine where in ClearCase the ReadMe file resides.

4.3.6.3.1 Interfaces and Data Types

Remedy's Action Request System exchanges data of various types through interfaces within and external to ECS. Table 4.3.6-19 lists Remedy's Action Request System interfaces.

Table 4.3.6-19. External Interface Protocols

Interface (facility)	Type of Primary Interface Protocols	Type of Backup Interface Protocols	Comments
Forwarding	E-mail	Default E-mail Backup Interface Protocols	Site to site forwarding of Trouble Tickets.
HTML	HTTP	Default HTTP Backup Interface Protocols	End user submission and queries.

4.3.6.4 Databases

Remedy's Action Request System is installed on Sybase; it creates, modifies, and deletes tables as forms are created, modified, and deleted with each column corresponding to a field in the forms. This is all done automatically and is invisible to the user.

4.3.6.5 Special Constraints

Note that while Trouble Tickets and the Contact Log forms are open to all operators, and operators have view privileges to the user form, only system administrators have the ability to modify the form and tools presented in this section. Privileges are set according to DAAC policy.

4.3.6.6 Outputs

Output from Remedy's Action Request System (besides output to the screen in the form of its GUIs) is in the form of a report either to the printer or to a file (reports discussed in Section 4.3.6.8) or a log entry as shown in Table 4.3.6-20.

In the Remedy **aradmin** tool, you may enable / disable logging at any time. Select File-> Server Information-> Log Files to display the current location of log files that have been enabled. The format of the messages is similar to the Unix syslog as seen in this example:

Table 4.3.6-20. Remedy Log File Messages Example

```
Mon Feb 23 16:28:16 1998  390600 : Failure during SQL operation to the database (ARERR 552)
Mon Feb 23 16:28:16 1998     Unable to connect: SQL Server is unavailable or does not exist. (Sybase 20009) :
Connection refused
Mon Feb 23 16:28:16 1998     Unable to connect: SQL Server is unavailable or does not exist. (Sybase 20009) :
Connection refused
```

4.3.6.7 Event and Error Messages

For Remedy's Action Request System's system messages see the *Action Request System Error Messages Guide*.

Table 4.3.6-21 lists non-system failure related messages, which appear on the operator's screen.

Table 4.3.6-21. Non-Failure Related Error Messages

Error Message String	Cause	Action
You have to assign the trouble ticket to somebody	Setting the Status to Assigned without setting the Assigned-to field.	Set Assigned-to field.
You have to assign a Closing Code to close	Setting Status to Closed without a Closing Code.	Set Closing Code field.
Trouble Ticket’s status must be “Closed” and its Closing Code field must have a value before forwarding is allowed with this button.	Attempting to forward an open ticket using the “Forward Closed TT to SMC” button.	Wait until the ticket is closed before attempting to forward it using the aforementioned button.
You can only submit a Trouble Ticket with your login id	Trying to submit Trouble Ticket using someone else’s user ID.	Use your user ID.
Trouble Ticket number $Ticket-Id$ has already been forwarded to or otherwise opened at $Forward-to$	Already forwarded Trouble Ticket to site in Forward-to field.	Check Forward-to site name against the sites that have already been forwarded the Trouble Ticket.
Must change status to "Forwarded" and fill in the "Forward-to" field	Must set the indicated fields before the Trouble Ticket can be forwarded.	Check Forward-to and Status fields to ensure they have the appropriate values.
This closed trouble ticket has already been sent to the SMC	Attempting to forward a copy of a closed ticket to the SMC and a copy has already been forwarded.	Dismiss Display window and make no further attempts to forward this ticket.
There is not an Associated Contact Log Id for this Trouble Ticket	Trying to access a Contact Log not associated with a Trouble Ticket through the RelB-Trouble Tickets form.	You can’t access the Contact Log for this Trouble Ticket because it does not exist.
There has not been a Trouble Ticket created for this log	Trying to access a Trouble Ticket not associated with a Contact Log through the RelB-Contact Log form.	You can’t access the Trouble Ticket for this Contact Log because it does not exist.
A Trouble Ticket is not created. A Trouble Ticket has already been opened for this log	Trying to create a Trouble Ticket via the Contact Log that has already been created.	You can’t open a Trouble Ticket for this Contact Log since one has already been opened.
A Trouble Ticket cannot be created. Contact Id required for Trouble Ticket submission	Contact ID is required for creation of a Trouble Ticket via the Contact Log.	Set the Contact ID field.
A Trouble Ticket cannot be created without a Log Id	Trying to create a Trouble Ticket via a Contact Log that has not yet been saved and hence has no Contact ID.	Select Apply to assign a Log ID then try and create a Trouble Ticket again.

4.3.6.8 Reports

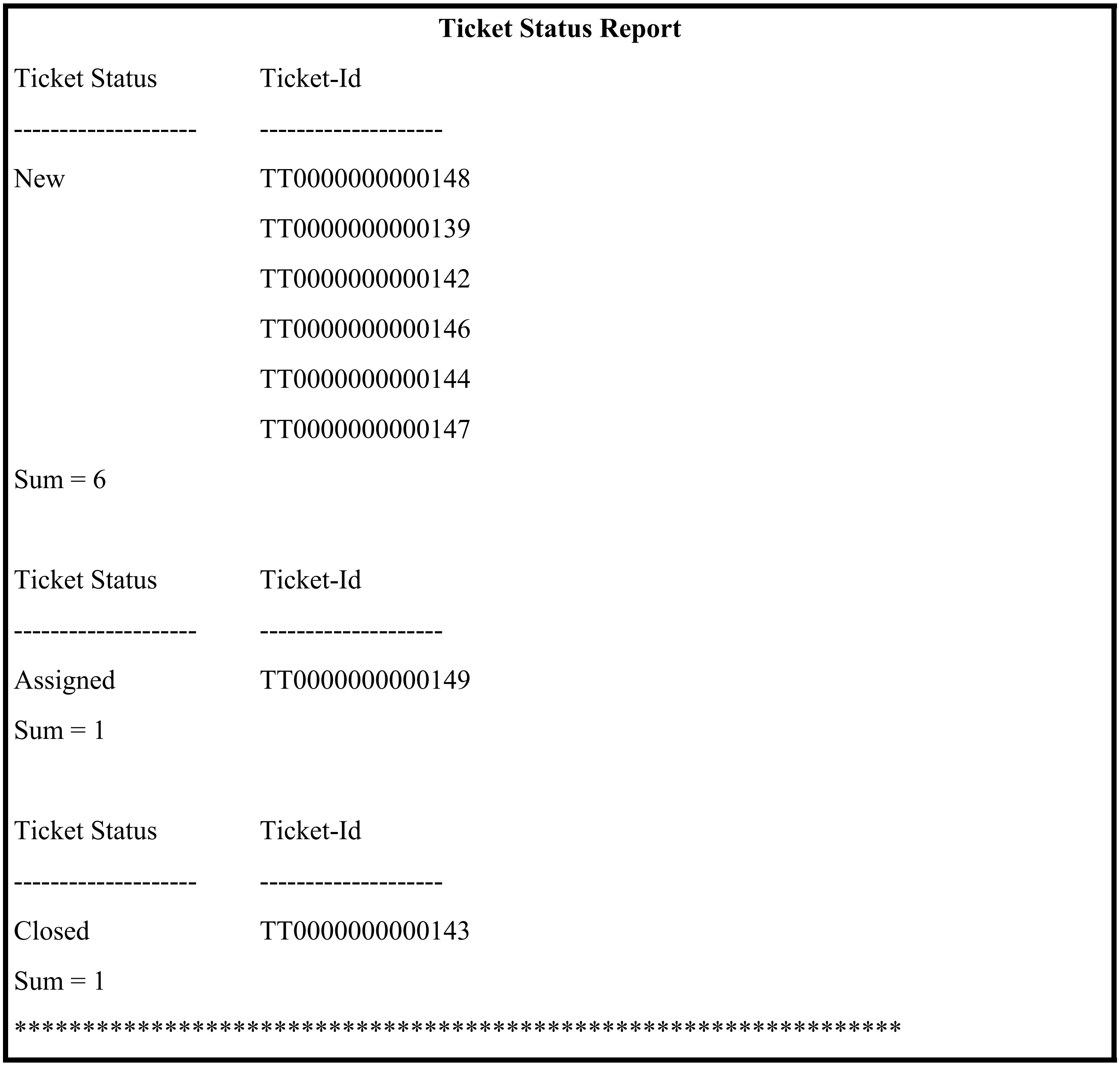
The Remedy Action Request System issues the reports described in Table 4.3.6-22.

Table 4.3.6-22. Reports

Report Type	Report Description	When and Why Used
Ticket Status Report	Indicates the status of a set of trouble tickets based on a particular criteria (e.g., by date range, assigned-user, status...).	When and if someone wants to know the status of a set of trouble tickets based on a particular criterion (e.g., by date range, assigned-user, status...).
Hardware Resource Report	Indicates by hardware resource, the number of problems encountered by the affected hardware resource.	When and if someone wants to know, by hardware resource, the number of problems encountered.
Trouble Ticket User Report (Number of Tickets by Submitter)	Indicates by submitter, the number and type of trouble tickets in the system.	When and if someone wants to know, by submitter, the number and type of trouble tickets in the system.
Trouble Ticket Statistics Report (Average Time to Close)	Indicates for a particular criterion, statistical information such as mean time to close.	When and if someone wants to know, for a particular criterion, statistical information such as mean time to close.
Number of Trouble Tickets by Status	Provides a summary of the number of tickets by status.	When and if someone wants to know a summary of the number of tickets by status.
Number of Tickets by Assigned Priority	Provides a summary of the number of tickets by priority.	When and if someone wants to know a summary of the number of tickets by priority.
Trouble Ticket Status Report (SMC)	Provides a summary of the tickets by status for importing into Excel.	When and if someone wants to import a summary of the tickets into Excel.
Custom reports	TTS allows for both extensive customization of the above reports and creation of new ones. The reporting capabilities include the capability to display not only data contained in the database but also statistical and correlation functions on that data.	When and if someone wants to know more than is available through the previous reports.

4.3.6.8.1 Sample Reports

The following are examples of sample reports that can be generated from the Trouble Ticket form. These sample reports (Figures 4.3.6-20 through 4.3.6-26) include: Ticket Status, Hardware Resource, Number of Tickets by Submitter, Average Time to Close, Number of Trouble Tickets by Status, Number of Tickets by Assigned Priority, and a Summary Report (imported into Excel).



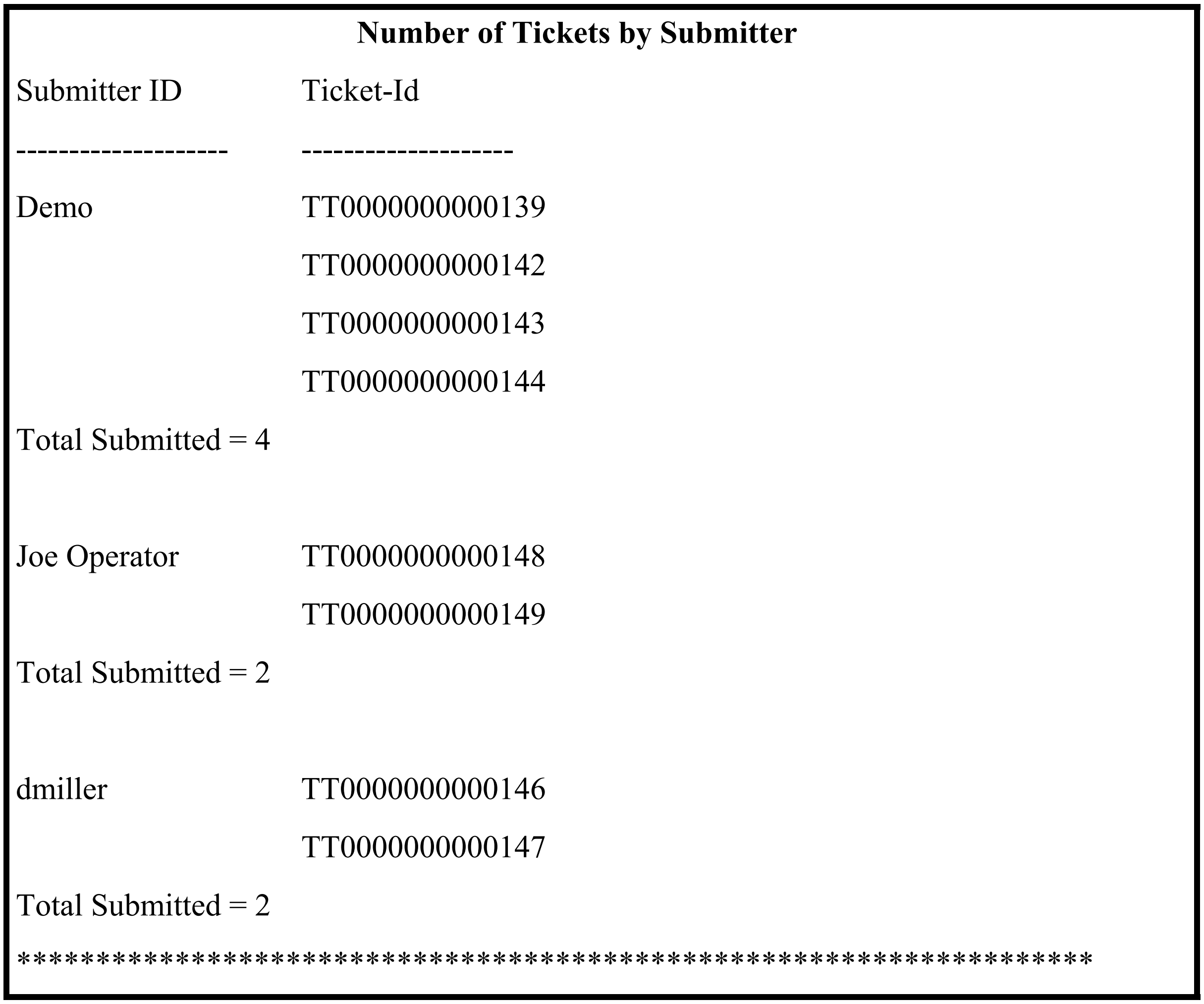
```
                              Ticket Status Report
Ticket Status          Ticket-Id
--------------------   --------------------
New                    TT0000000000148
                       TT0000000000139
                       TT0000000000142
                       TT0000000000146
                       TT0000000000144
                       TT0000000000147
Sum = 6

Ticket Status          Ticket-Id
--------------------   --------------------
Assigned               TT0000000000149
Sum = 1

Ticket Status          Ticket-Id
--------------------   --------------------
Closed                 TT0000000000143
Sum = 1
*********************************************************************
```

Figure 4.3.6-20. Trouble Ticket Status Report

Hardware Resource Report

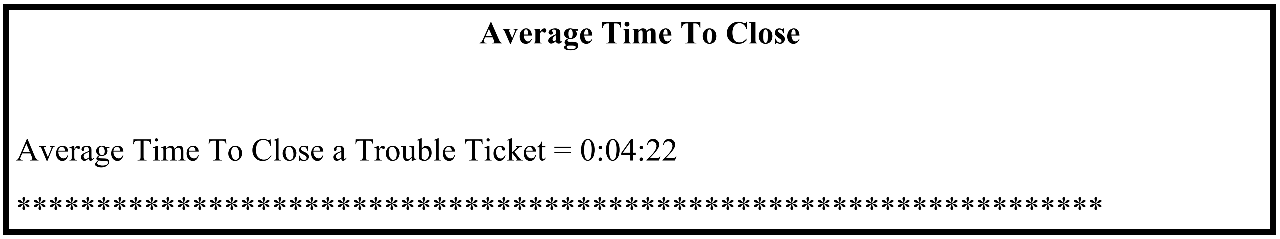
```
Hardware Resource

--------------------

slimer

Number of Associated Tickets = 1


epserver

Number of Associated Tickets = 4


cyclops

Number of Associated Tickets = 3
*********************************************************************
```

Figure 4.3.6-21. Hardware Resource Report



```
                    Number of Tickets by Submitter
Submitter ID        Ticket-Id
--------------------  -------------------
Demo                TT0000000000139
                    TT0000000000142
                    TT0000000000143
                    TT0000000000144
Total Submitted = 4

Joe Operator        TT0000000000148
                    TT0000000000149
Total Submitted = 2

dmiller             TT0000000000146
                    TT0000000000147
Total Submitted = 2
********************************************************************
```

Figure 4.3.6-22. Number of Tickets by Submitter Report



```
                         Average Time To Close


Average Time To Close a Trouble Ticket = 0:04:22
********************************************************************
```

Figure 4.3.6-23. Average Time to Close Report

```
Number of Trouble Tickets by Status


Ticket Status
--------------------
New
Number of Tickets = 6


Assigned
Number of Tickets = 1


Closed
Number of Tickets = 1
*******************************************************************************
```

Figure 4.3.6-24. Number of Tickets by Assigned Status Report

```
Number of Tickets by Assigned Priority


Assigned-Priority
--------------------
Low
Number of Tickets = 8
********************************************************************************
```

Figure 4.3.6-25. Number of Tickets by Assigned Priority Report

Summary Report to be imported into Excel (comma separated values)

```
"Ticket-Id","Assigned-Priority","Closing Code","Current Site","Hardware Resource","Key
Words","ProblemType","SoftwareResource","TicketStatus","New.TIME","Assigned.TIME","SolutionPropo
sed.TIME","ImplementSolution.TIME","SolutionImplemented.TIME","Closed.TIME","Forwarded.TIME","
Work Around.TIME","Not Repeatable.TIME"

"TT0000000000139","Low",,"cyclops","cyclops",,,,"New","05/22/96 11:06:44","", "","","", "","","",""

"TT0000000000142","Low",,"cyclops","cyclops",,,,"New","05/23/96 10:13:14","","","","", "","","",""

"TT0000000000143","Low","ConfigurationError","cyclops","cyclops",,,,"Closed","05/28/96
10:36:27","","","","05/28/96 10:40:49","05/28/96 10:40:54","","",""

"TT0000000000144","Low",,"cyclops","epserver",,,,"New","05/30/96 09:25:44","","","", "","","","",""

"TT0000000000146","Low",,"cyclops","epserver",,,,"New","05/30/96 13:47:53","","","", "","","","",""

"TT0000000000147","Low",,"cyclops","epserver",,,,"New","05/30/96 13:48:18","","","", "","","","",""

"TT0000000000148","Low",,"cyclops","epserver",,,,"New","05/31/96 11:54:28","","","", "","","","",""

"TT0000000000149","Low",,"cyclops","slimer",,,,"Assigned","06/07/96 14:04:03","06/07/96
14:06:17","","","","","","",""

**********************************************************************
```

Figure 4.3.6-26. Summary Report

4.3.6.8.2 Report Customization

See *Remedy's Action Request System User's Guide*, Chapter 5: "Reports", page 5-1.

This page intentionally left blank.

4.4 Security and Accountability

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

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

This page intentionally left blank.

4.4.1 TCP Wrappers

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

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

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

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

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

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

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

4.4.1.1 Quick Start Using TCP Wrappers

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

4.4.1.1.1 Command Line Interface

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

4.4.1.2 TCP Wrapper Main Screen

TCP Wrapper does not have a graphical user interface.

4.4.1.3 Required Operating Environment

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

4.4.1.4 Databases

None.

4.4.1.5 Special Constraints

None.

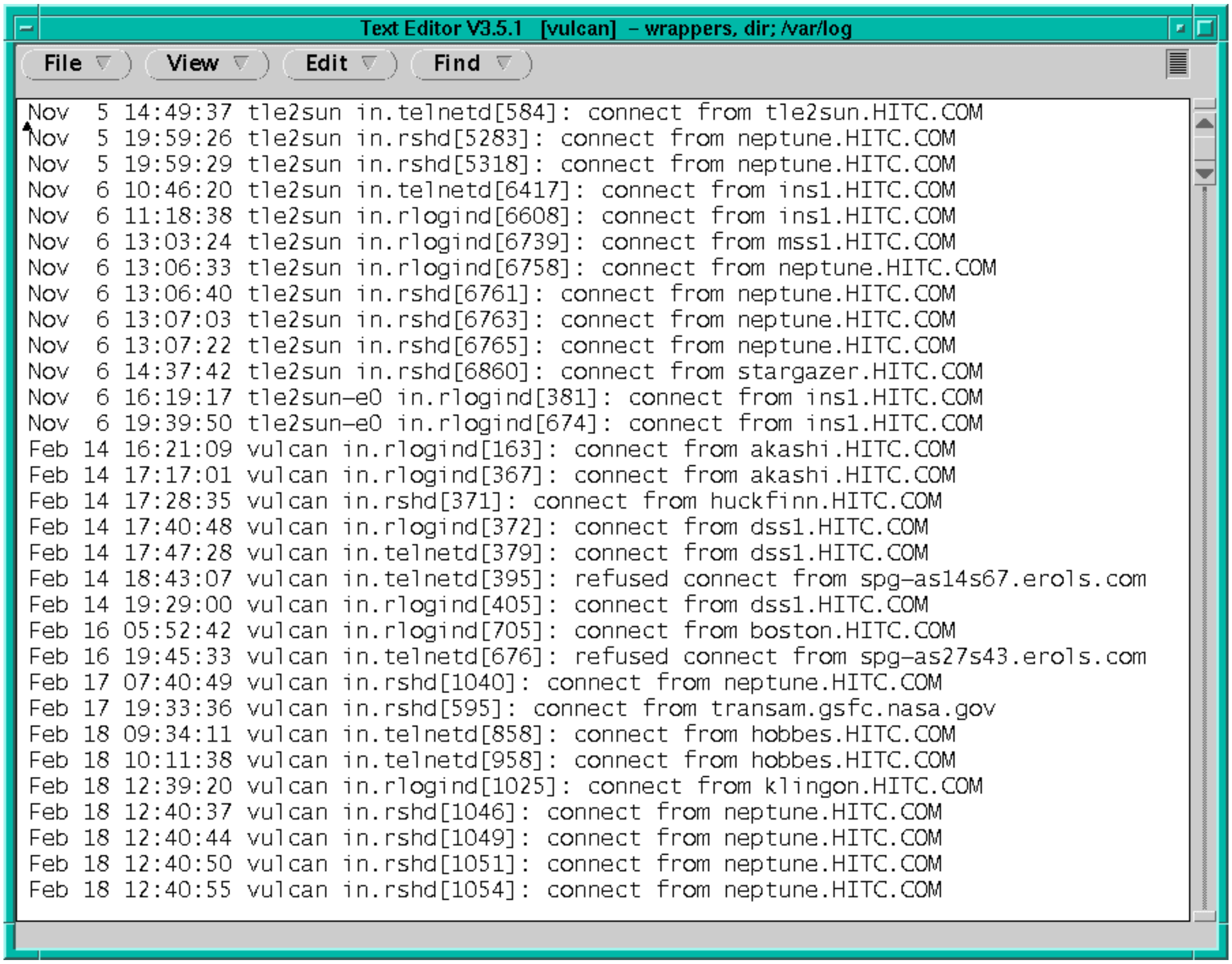
4.4.1.6 Outputs

Table 4.4.1-2 describes TCP Wrappers output.

Table 4.4.1-2. TCP Wrapper Outputs

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

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



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

Figure 4.4.1-1. Example of TCP Wrappers Log

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

4.4.1.7 Event and Error Messages

None.

4.4.1.8 Reports

None.

4.4.2 Crack

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

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

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

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

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

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

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

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

4.4.2.1 Quick Start Using Crack

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

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

4.4.2.1.1 Invoking Crack From the Command Line Interface

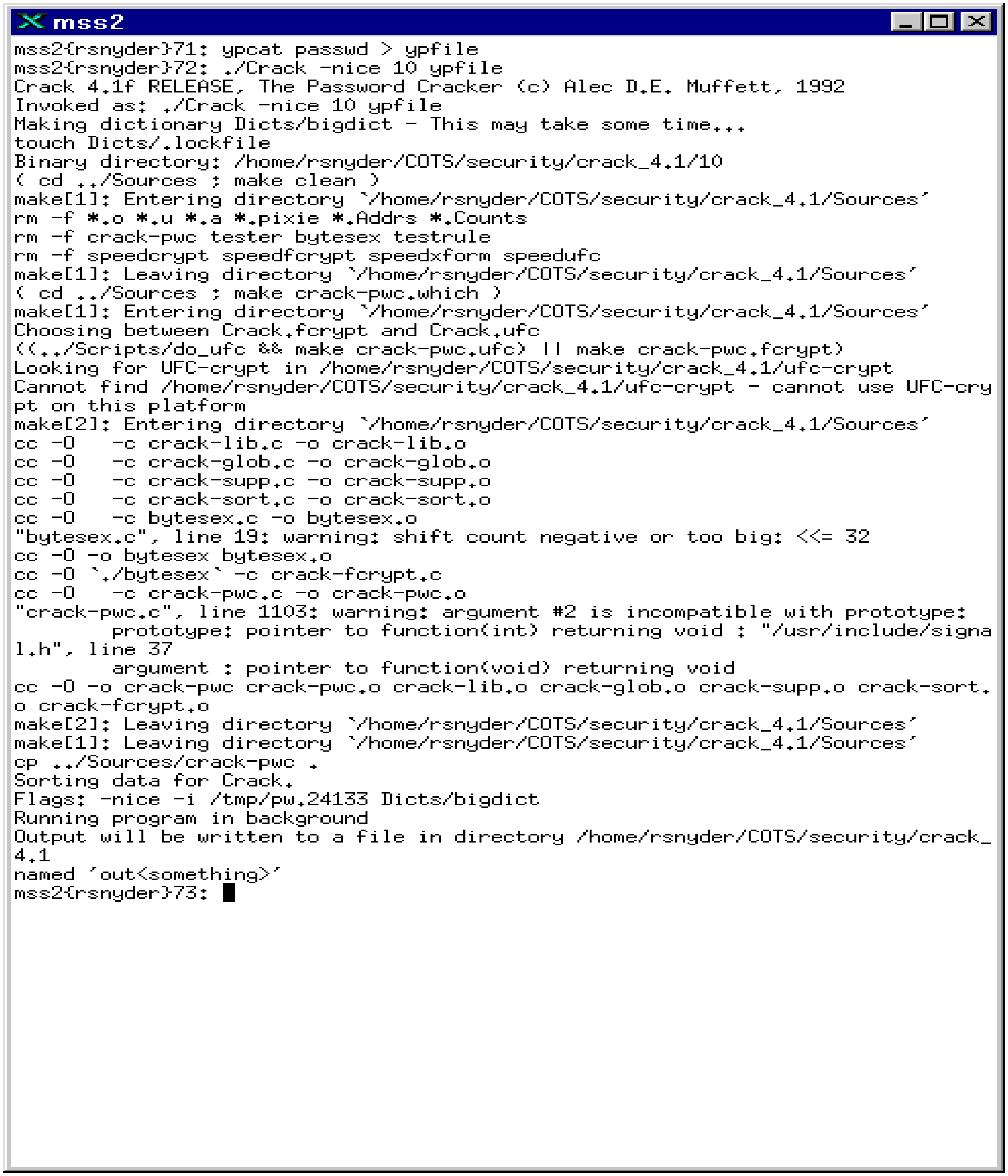
To execute Crack from the command line prompt use:

>Crack [*filename*]

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

4.4.2.2 Crack Main Screen

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



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

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

4.4.2.3 Required Operating Environment

Crack can run on any 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.

4.4.2.4 Databases

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

4.4.2.5 Special Constraints

None

4.4.2.6 Outputs

Table 4.4.2-2 describes the output for Crack.

Table 4.4.2-2. Outputs for Crack

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

4.4.2.7 Event and Error Messages

None.

4.4.2.8 Reports

Crack 4.1 does not support reports.

4.4.3 Tripwire

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

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

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

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

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

Table 4.4.3-1. Common ECS Operator Functions Performed

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

4.4.3.1 Quick Start Using Tripwire

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

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

The following is the general syntax of executing Tripwire

*tripwire [**options** ...] >**filename***

Where ***options*** are:

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

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

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

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

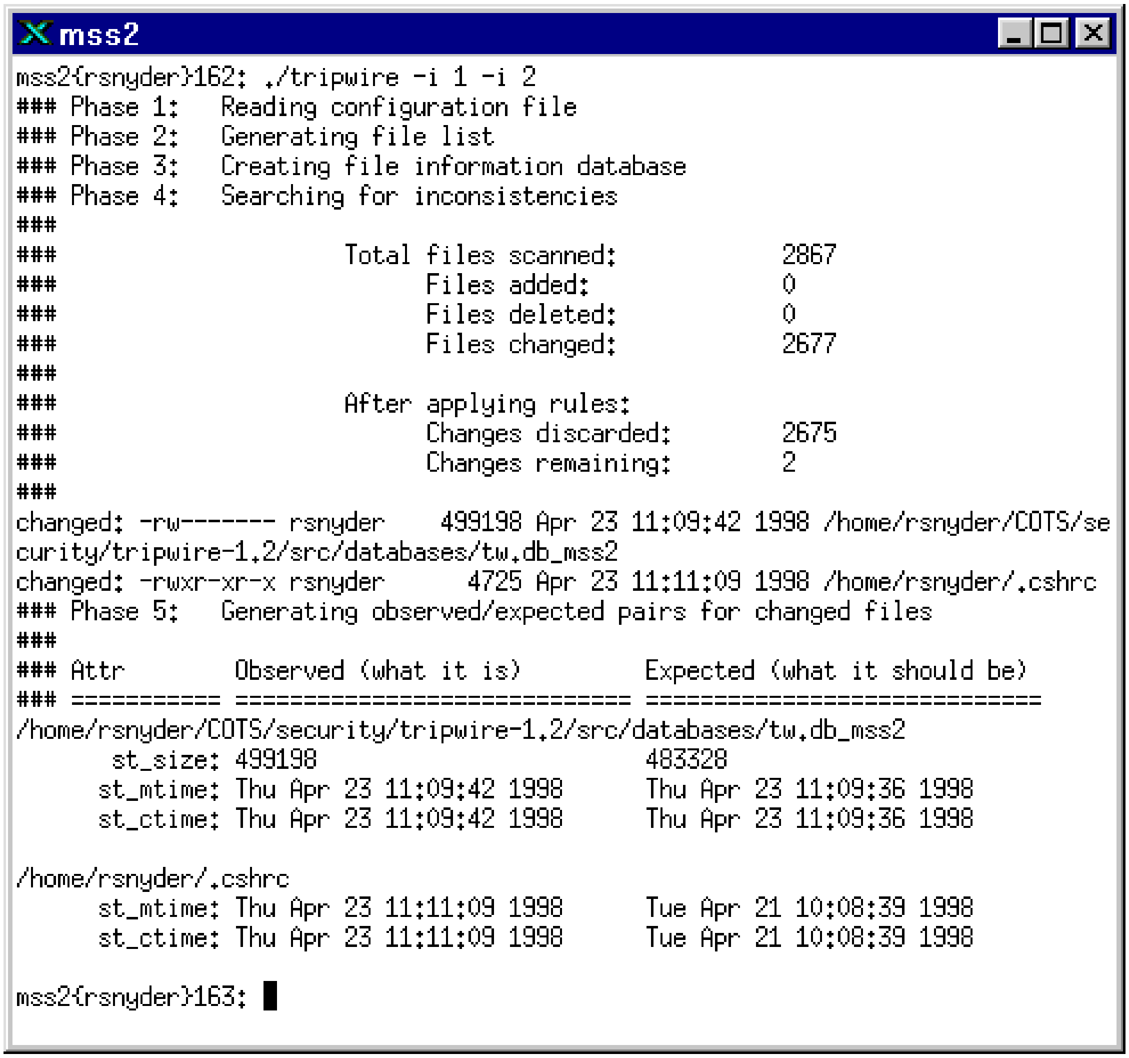
http://www.alw.nih.gov/Security/prog-monitor.html

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

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

4.4.3.2 Tripwire Main Screen

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



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

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

mss2{rsnyder}163: █
```

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

4.4.3.3 Required Operating Environment

Tripwire runs on all Sun and Origin servers.

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

4.4.3.4 Databases

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

4.4.3.5 Special Constraints

None.

4.4.3.6 Outputs

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

Table 4.4.3-2. Tripwire Outputs

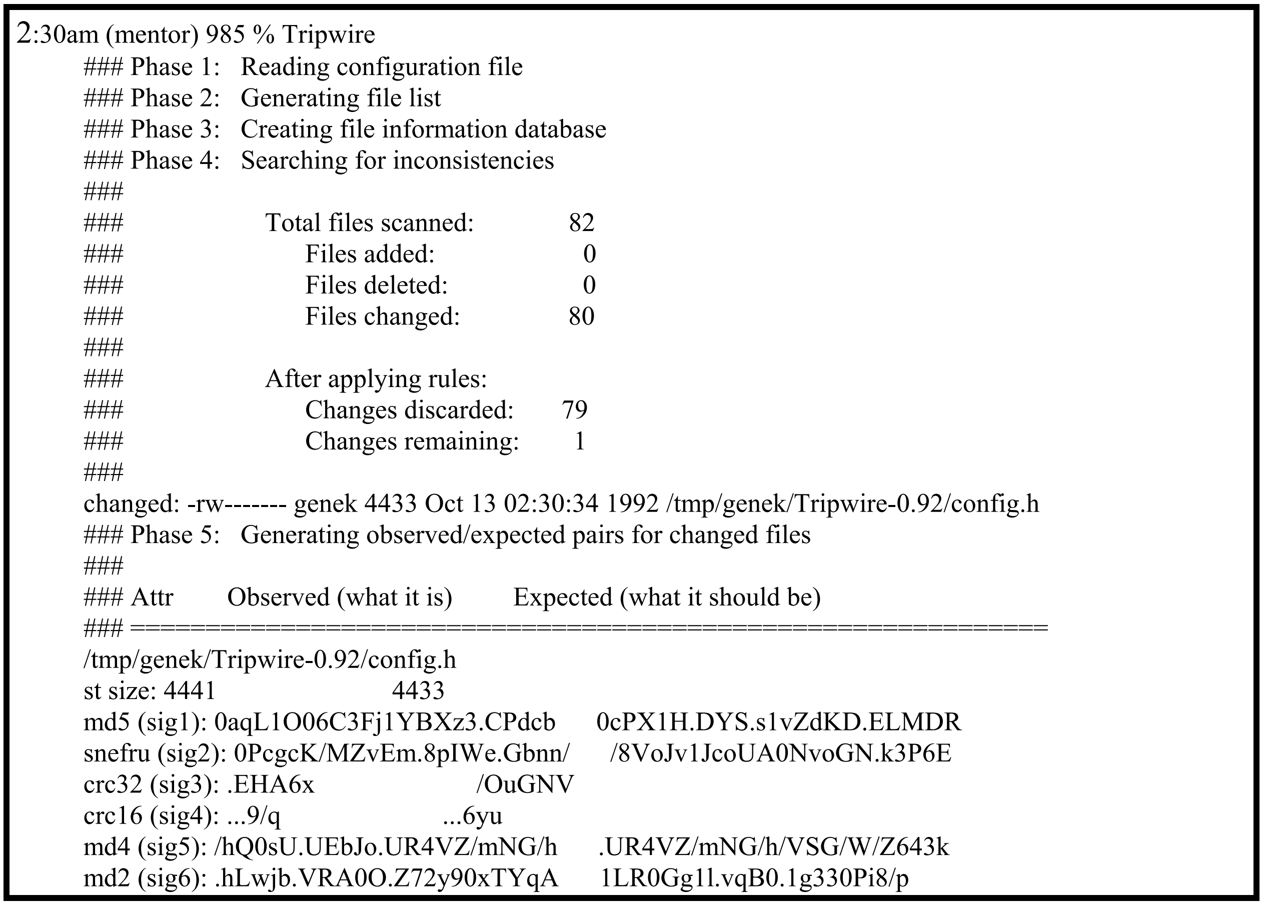
Output	Description and Format
Tripwire compares the new datastore with the existing Tripwire datastore stored on the file system, reporting added or deleted files, as well as those files that have changed.	See 4.4.3.8, the Report section, for a sample of Tripwire output.
Email to the operator.	Email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

4.4.3.7 Event and Error Messages

None.

4.4.3.8 Reports

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



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

Figure 4.4.3-2. Tripwire Report

4.4.4 Cryptographic Management Interface (CMI)

The Cryptographic Management Interface (CMI) GUI program, EcSeAuthnProg, is used by operations personnel to generate a randomized username and password (though only the password is currently used) given a key. There is one key for each 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 4.4.4-1.

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

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

4.4.4.1 Quick Start Using CMI

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

To execute CMI from the command line prompt, enter:

```
> EcSeAuthnProg
```

4.4.4.2 CMI Main Screen

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

- Application Key field
- User Id field
- Password field

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

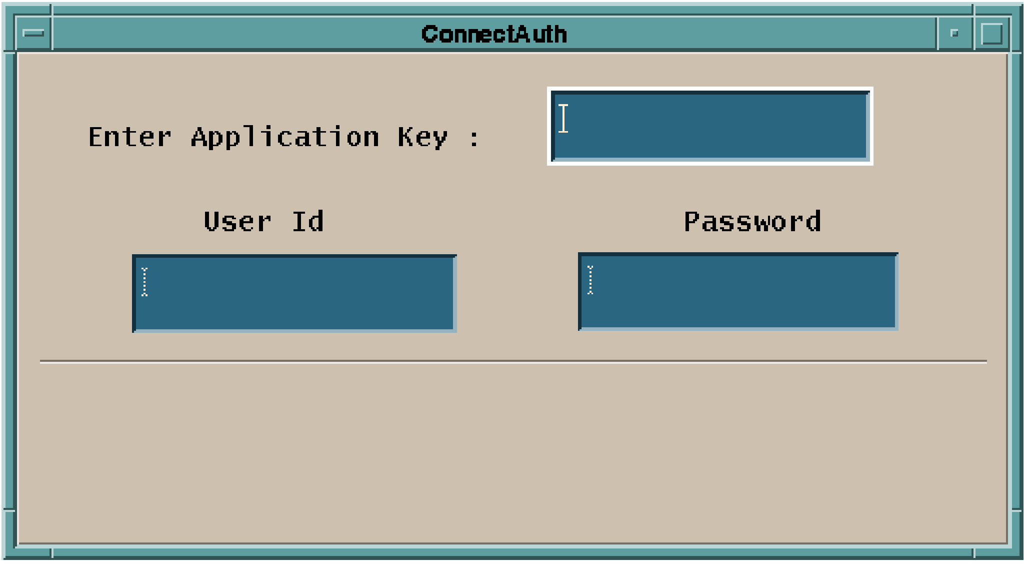


Figure 4.4.4-1. CMI Main Screen

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

Table 4.4.4-2. CMI Field Descriptions

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

4.4.4.3 Required Operating Environment

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

4.4.4.3.1 Interfaces and Data Types

CMI utilizes no special data types or interfaces.

4.4.4.4 Databases

None.

4.4.4.5 Special Constraints

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

4.4.4.6 Outputs

All information is displayed on the CMI screen.

4.4.4.7 Event and Error Messages

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

4.4.4.8 Reports

None.

This page intentionally left blank.

4.5 Science Software Integration and Test (SSI&T)

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

This page intentionally left blank.

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

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

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

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

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

Operating Function	Command/Script or GUI	Description	When and Why to Use
Prepare SSI&T checklist.	SSI&T Manager GUI	• SSI&T Manager GUI requires a checklist it can access. • The checklist contains all the operational procedures for science software integration and test.	• During normal SSI&T operations, used to keep track of activities pending and completed. • Checklist of operational procedures must first be prepared using a text editor. • SSI&T Manager must then be linked to the checklist before invoked.
Change SSI&T Checklist.	SSI&T Manager GUI	• SSI&T Manager GUI requires changing the checklist. • The checklist contains all the operational procedures for science software integration and test.	• Update tracking of activities pending and completed. • Checklist of operational procedures must first be prepared using a text editor. • SSI&T Manager must then be linked to the checklist before invoked.

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

Operating Function	Command/Script or GUI	Description	When and Why to Use
Open xterm session	• Open xterm window via Tools pull-down menu. • Also can be opened via Unix command *xterm.*	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: • FORCHECK • Prohibited Function Checker • Process Control File (PCF) Checker • Prolog extractor	• Check FORTRAN 77 science software. • Check if certain functions are used in the science software, which conflict with the production environment. • Check the syntax of the data in the Process Control File. • Extract prologs from science software.	• To ensure that science code conforms to ECS standards. • To ensure the delivered PCF is of the proper syntax. • To extract prologs from science software.
Product Examination	Select the following via Tools: Product Examination pull-down menu: • IDL • EOSView	Opened via Tools pull-down menu.	• Ad-hoc graphical analysis. • For viewing an arbitrary file (e.g., standard product) in HDF format.
File Comparison	Select the following via Tools: Product Examination File Comparison • ASCII files • Binary files • HDF files (GUI) • HDF files (hdiff)	Compares the outputs of the science software between the DAAC and SCF.	Ensures output generated at the SCF when running the science software is the same output generated at the DAAC.

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

Operating Function	Command/Script or GUI	Description	When and Why to Use
Edit text	Select the following via Tools: Text Editors pull-down menu: • Emacs • Xedit	Text editors.	Edit arbitrary text file.
PDPS Database	Select the following via Tools: PDPS Database pull-down menu: • PCF ODL Template • Check ODL Files • SSI&T Science Metadata Update • SSI&T Operational Metadata Update GUI	• Creates an ODL file template from the science software PCF. • Check the ODL file updates PGE and ESDT SCIENCE metadata in the PDPS /SSI&T database. • Updates PGE OPERATIONAL metadata via GUI in the PDPS /SSI&T database.	To initialize and update the Planning/Production (PDPS) databases: • SSI&T version • Production version
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 Delivery Archive Package (DAP). 2. Inserts static input file. 3. Inserts test dynamic input file. 4. Inserts tar file with files needed for processing. 5. Edits Science Software Archive Package (SSAP) components. 6. Acquires Metadata Configuration Files (MCFs) from the Data Server.	1. After DAP notification received by email. 2. After ESDT is registered in Data Server, before test PGE run. 3. After ESDT is registered in Data Server, before test PGE run. 4. After PGE compilation, before test PGE run. 5. After PGE testing is complete, at time of promotion to Production, as needed to edit/review SSAP components. 6. As needed to retrieve MCF.

4.5.1.1 Quick Start Using SSI&T Manager

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

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

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

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

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

where:

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

4.5.1.1.1.1 Sun Platform

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

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

Command Line Interface	Description and Format	When and Why Used
EcDpAtMgrStart	Startup script for SSI&T Manager.	To do SSI&T, and record items accomplished in the log.
EcDpAtMgrLogDump	Used to dump/print a log file to the screen.	As needed.
xterm	Open a Unix command line window.	As needed.
sparcworks	Ad hoc code analysis	As needed.
ghostview	Postscript file viewer	As needed.
netscape	WWW browser Netscape	As needed.
acroread	PDF file viewer Adobe Acrobat	As needed.
DpAtMgrForcheck	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.

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

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

4.5.1.1.1.2 SGI Platform

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

Table 4.5.1-3. Command Line Interfaces (SGI)

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

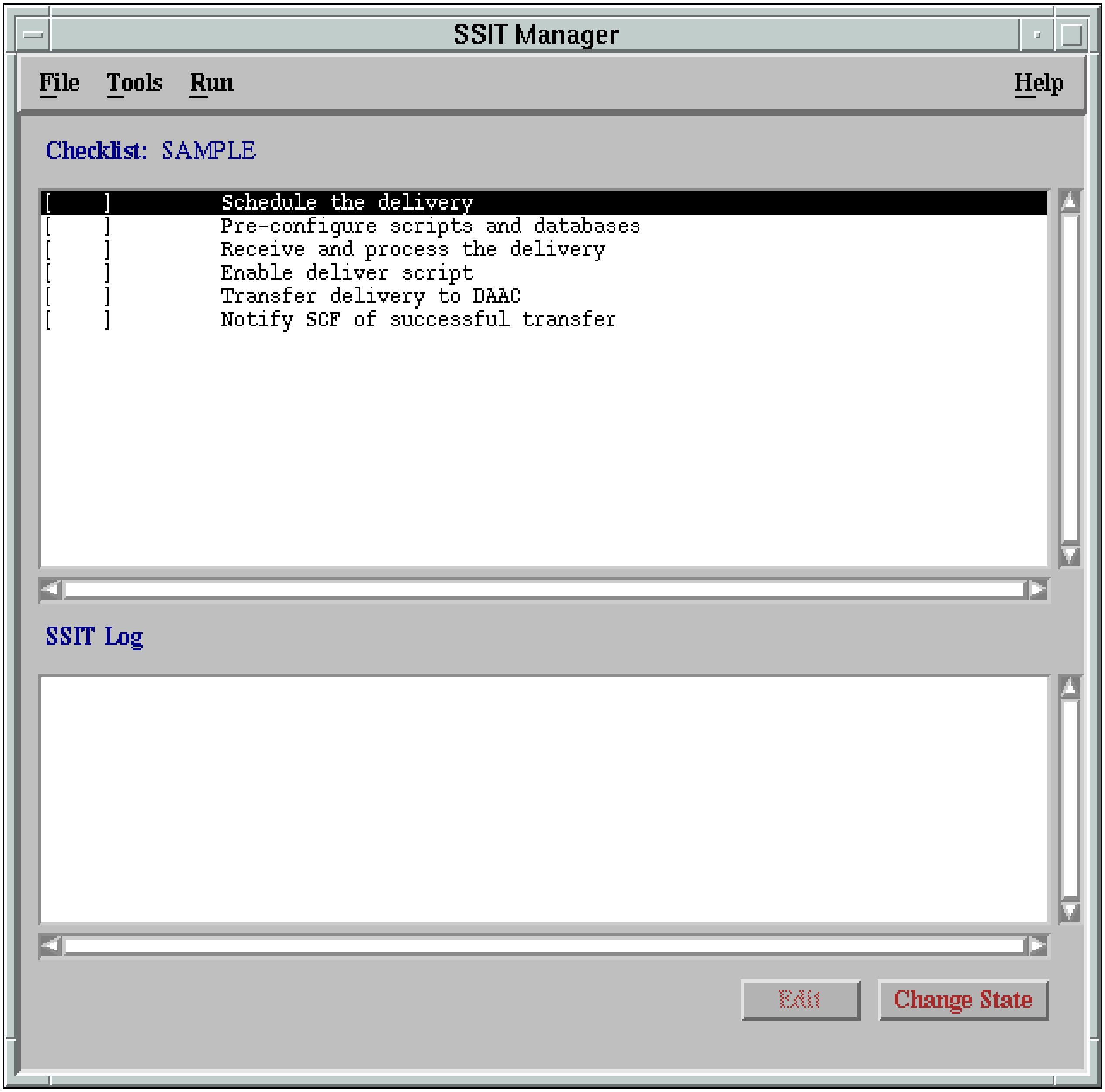
Table 4.5.1-4 lists SGI platform tools associated with the SSI&T process.

Table 4.5.1-4. SGI Platform Tools Description

Categories/Tools	Tool Description & Use	Further Information
ProDev Workshop	• ProDev Workshop is a COTS package developed by SGI • This tool is targeted within ECS for applications running on the SGI science processors • ProDev Workshop is a software development support tool which includes several tools, which have applicability to SSI&T • Among these tools is the capability to perform static code analysis to aid in the detection of memory leaks	• ProDev Workshop includes online documentation describing its features • Other ProDev Workshop documentation is delivered with ECS. • ProDev Workshop is not available from the SSI&T menu. This tool must be started from the Command Line Interface: see Table 4.5.1-3.
PGE Performance	• DpAtRusage is a custom tool developed by ECS • It measures performance parameters such as CPU time used for a PGE linked to the SDP Toolkit, SCF version.	A help message is printed if the tool is invoked without input parameters.

4.5.1.2 SSI&T Manager

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



Checklist: SAMPLE

```
[     ]          Schedule the delivery
[     ]          Pre-configure scripts and databases
[     ]          Receive and process the delivery
[     ]          Enable deliver script
[     ]          Transfer delivery to DAAC
[     ]          Notify SCF of successful transfer
```

Figure 4.5.1-1. SSI&T Manager Main Screen

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

The Checklist Window

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

The SSI&T Log Window

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

The Menu Bar

The menu bar includes a simple selection of options:

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

SSI&T Manager Buttons

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

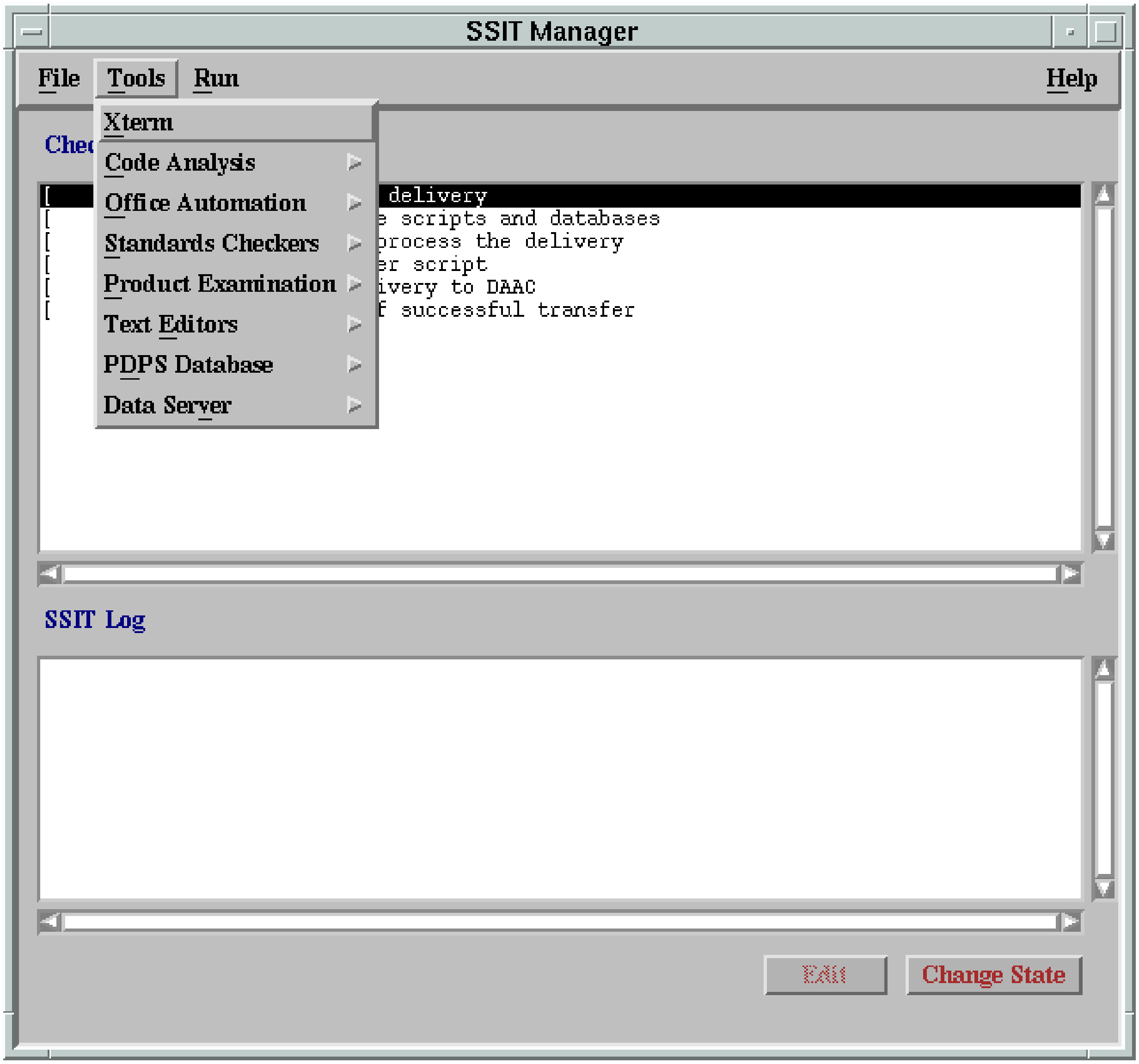
- **Edit** - The SSI&T specialist, to include additional detail on the SSI&T process, can edit the checklist items. Clicking the Edit button brings up the text editor for this purpose
- **Change State** - This toggle button allows the SSI&T specialist to switch between “checked” and “unchecked” for the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

4.5.1.2.1 SSI&T Manager Tools

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

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

Many of the items accessed through the Tools menu are COTS products for which documentation exists. Some items are custom applications from other ECS subsystems (e.g., EOSView). The complete description and documentation for these items is provided in other parts of this document. The remaining items are custom applications and are described in the following subsections. Note, since SSI&T is a collection of various tools, there is no specific order in which they must be run. Most tools can be brought up from the SSI&T Manager Main screen as well as be started on their own. The File menu provides the capability to exit the manager, and the Tools menu gives access to various tools that make up SSI&T.



```
[          delivery
[         e scripts and databases
[         process the delivery
[         er script
[         ivery to DAAC
[         f successful transfer
```

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

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

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

Category	Tools	Tool Description & Use	Further Information
UNIX Access	Xterm	This tool's option allows the operator to open an Xterm window and enter Unix commands, as needed. The other SSI&T Manager windows remain available while an Xterm window is open.	Unix documentation is available online via 'man' command.
Code Analysis	SPARCworks	SPARCworks is a COTS package developed by Sun Micro Systems Selecting this option from the Tools list starts the SPARCworks tool, which opens a separate window. Among the tools available in SPARCworks is the code analysis tool, which allows for static code analysis to aid in detecting memory leaks. For the SGI systems the ProDev Workshop product provides similar functionality.	• SPARCworks includes online documentation describing its features • Other SPARCworks documentation is delivered with ECS. This documentation includes: – SPARCworks Browsing Source Code – Sparcworks Debugging a Program 3.0.1 – SPARCworks Tutorial 3.0.1 – Sparcworks Performance Tuning an Application 3.0.1
	ProDev Workshop	ProDev Workshop is a COTS package developed by SGI. This tool is targeted within ECS for applications running on the SGI science processors. ProDev Workshop is a software development support tool, which includes several tools having applicability to SSI&T. Among these tools is the capability to perform static code analysis to aid in the detection of memory leaks	• ProDev Workshop includes online documentation describing its features • Other ProDev Workshop documentation is delivered with ECS. • ProDev Workshop is not available from the SSI&T menu. This tool must be started from the Command Line Interface: see Table 4.5.1-3.
Office Automation	GhostView	Selecting Ghostview from the Tools menu item starts the Ghostview application. Ghostview is a freeware tool. This is used for displaying Postscript format files.	Ghostview includes online documentation on its use. Since this is a freeware product, there is no other documentation provided by ECS for Ghostview.

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

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

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

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

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

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

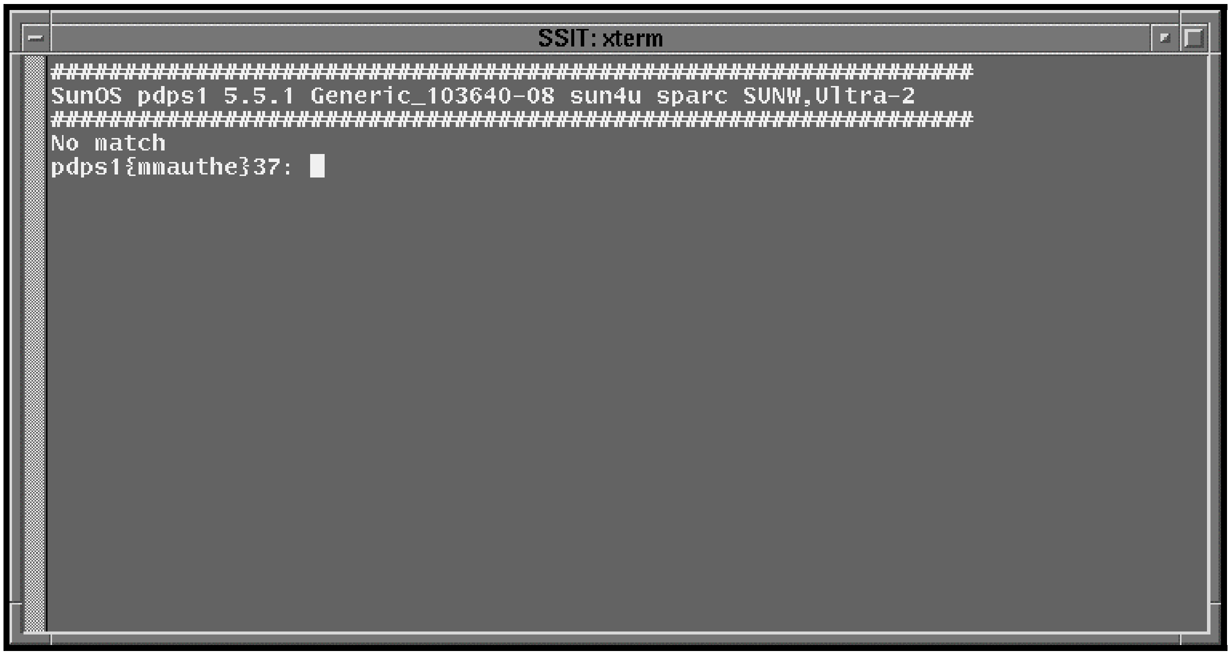
Table 4.5.1-5. SSI&T Manager Tools Description (5 of 5)

Category	Tools	Tool Description & Use	Further Information
PDPS Database	PCF ODL Template	Creates a PDPS PGE SCIENCE metadata ODL file from science software PCF.	This tool is described in detail in section 4.5.1.2.8.1.
	Check ODL	Parses ODL file and records errors. This tool accesses the PDPS DB for information on PGE/ESDT who's ODL is already available.	This tool is described in detail in section 4.5.1.2.8.2.
	SSI&T Science Metadata Update	Updates PDPS/SSI&T database with PGE and ESDT SCIENCE metadata read from ODL files.	This tool is described in detail in section 4.5.1.2.8.3.
	SSI&T Operational Metadata Update GUI	Updates PDPS/SSI&T database with PGE OPERATIONAL metadata input by user.	This tool is described in detail in the following 4.5.1.2.8.4 section.
Data Server	Acquire DAP	Acquires a DAP from the Data Server.	This tool is described in detail in section 4.5.1.2.9.1.
	Get MCF	Gets information about the ESDT the MCF is acquired for and interface with the Data Server to get the desired MCF.	This tool is described in detail in section 4.5.1.2.9.2.
	Insert Static	Inserts a static input file to the Data Server.	This tool is described in detail in section 4.5.1.2.9.3.
	Insert Test Dynamic	Inserts a test dynamic input file to the Data Server.	This tool is described in detail in section 4.5.1.2.9.4.
	Insert EXE TAR	Inserts tar file of executables, etc. needed to run science software to the Data Server.	This tool is described in detail in section 4.5.1.2.9.5.
	SSAP Editor	Read/Write Metadata and record errors.	This tool is described in detail in section 4.5.1.2.9.6.

4.5.1.2.2 Xterm

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

There is no submenu for the Xterm menu.



```
#################################################################
SunOS pdps1 5.5.1 Generic_103640-08 sun4u sparc SUNW,Ultra-2
#################################################################
No match
pdps1{mmauthe}37: 
```

Figure 4.5.1-3. Xterm Unix Terminal Window

4.5.1.2.3 Code Analysis

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

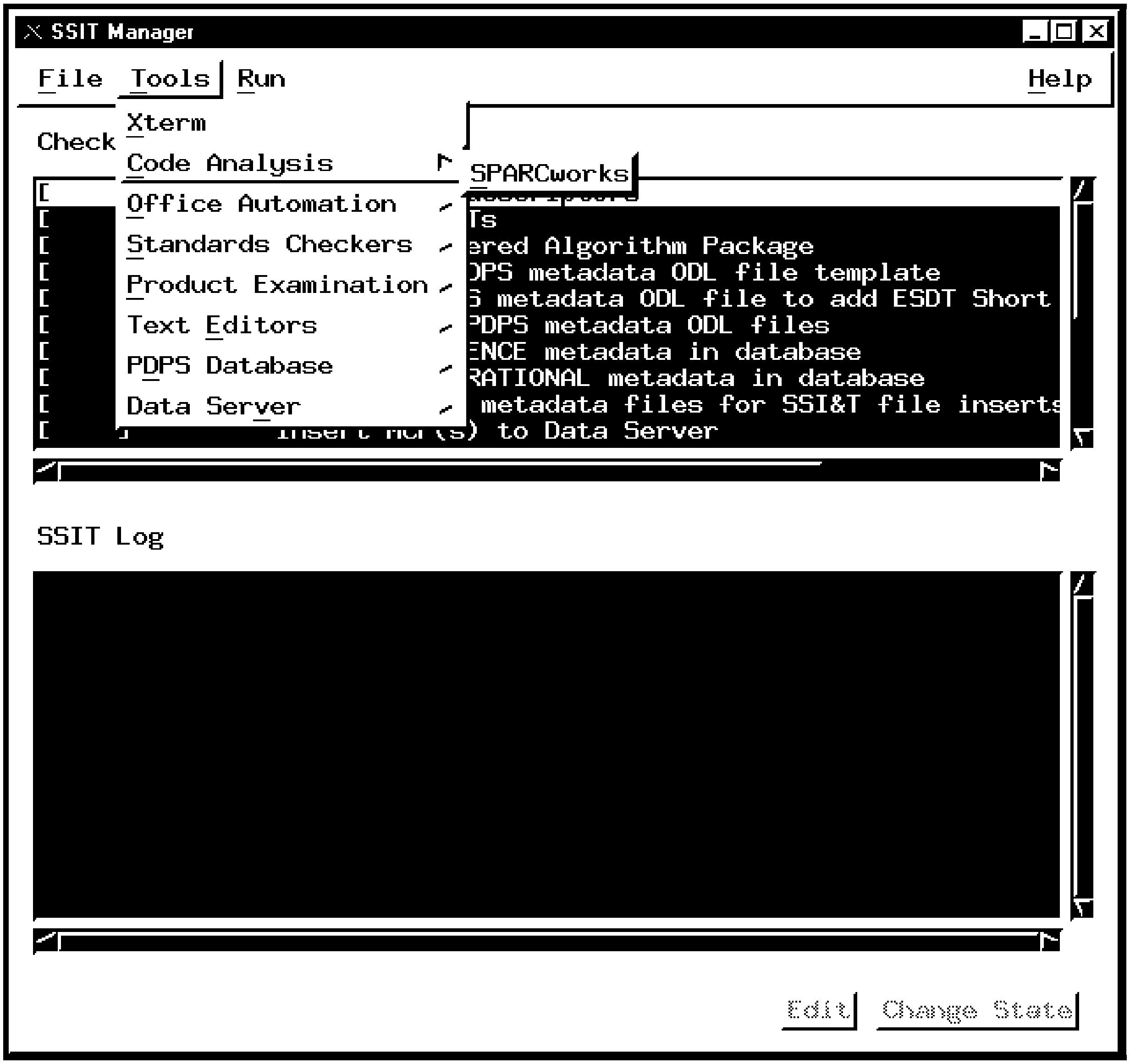


Figure 4.5.1-4. SSI&T Manager Main Screen with Code Analysis Tool Menu Displayed

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 results in significant performance reduction.

4.5.1.2.4 Office Automation

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

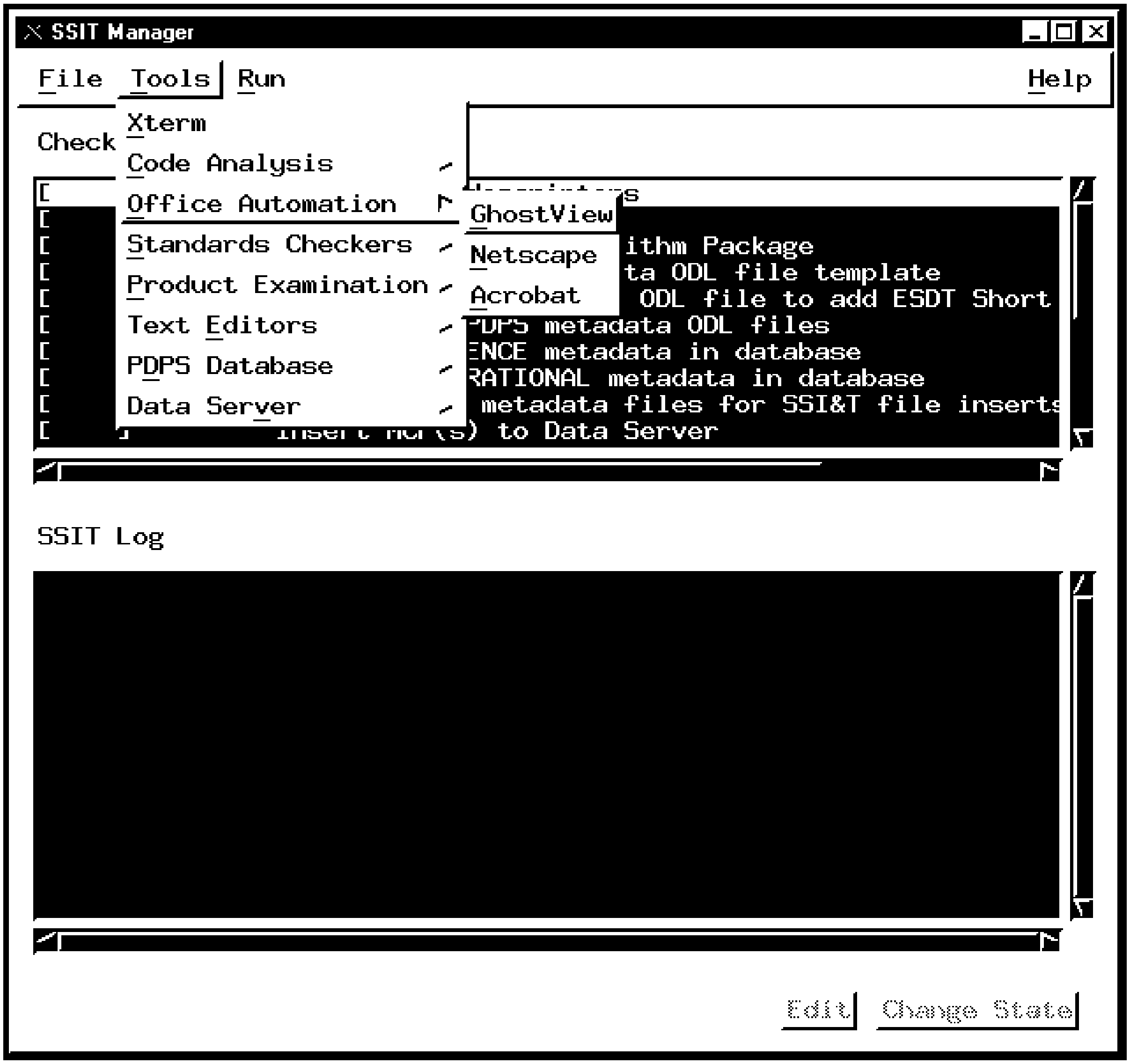


Figure 4.5.1-5. SSI&T Manager Main Screen with Office Automation Tool Menu Displayed

4.5.1.2.4.1 GhostView

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

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

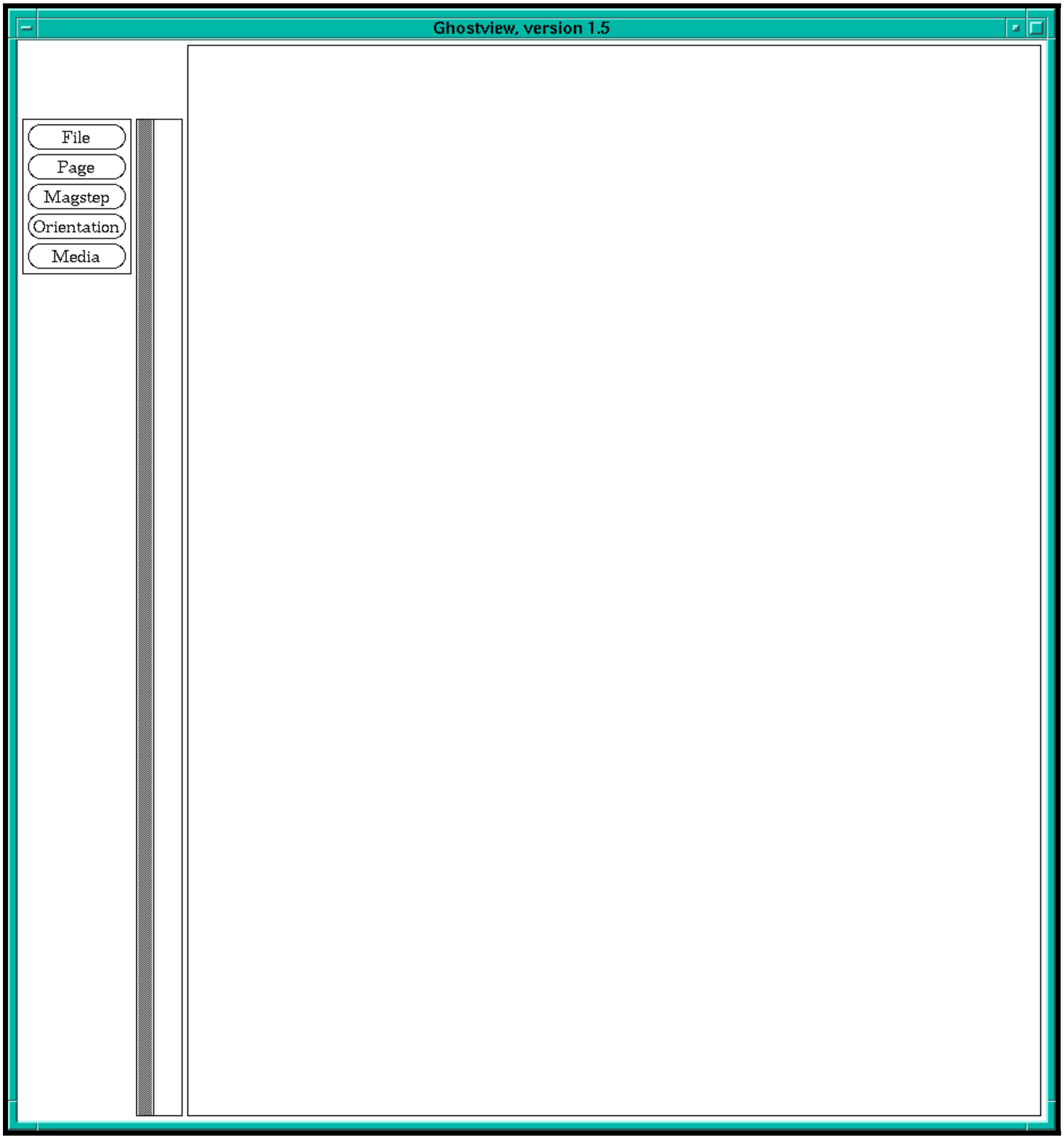


Figure 4.5.1-6. GhostView Main Screen

4.5.1.2.4.2 Netscape

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

4.5.1.2.4.3 Acrobat

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

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

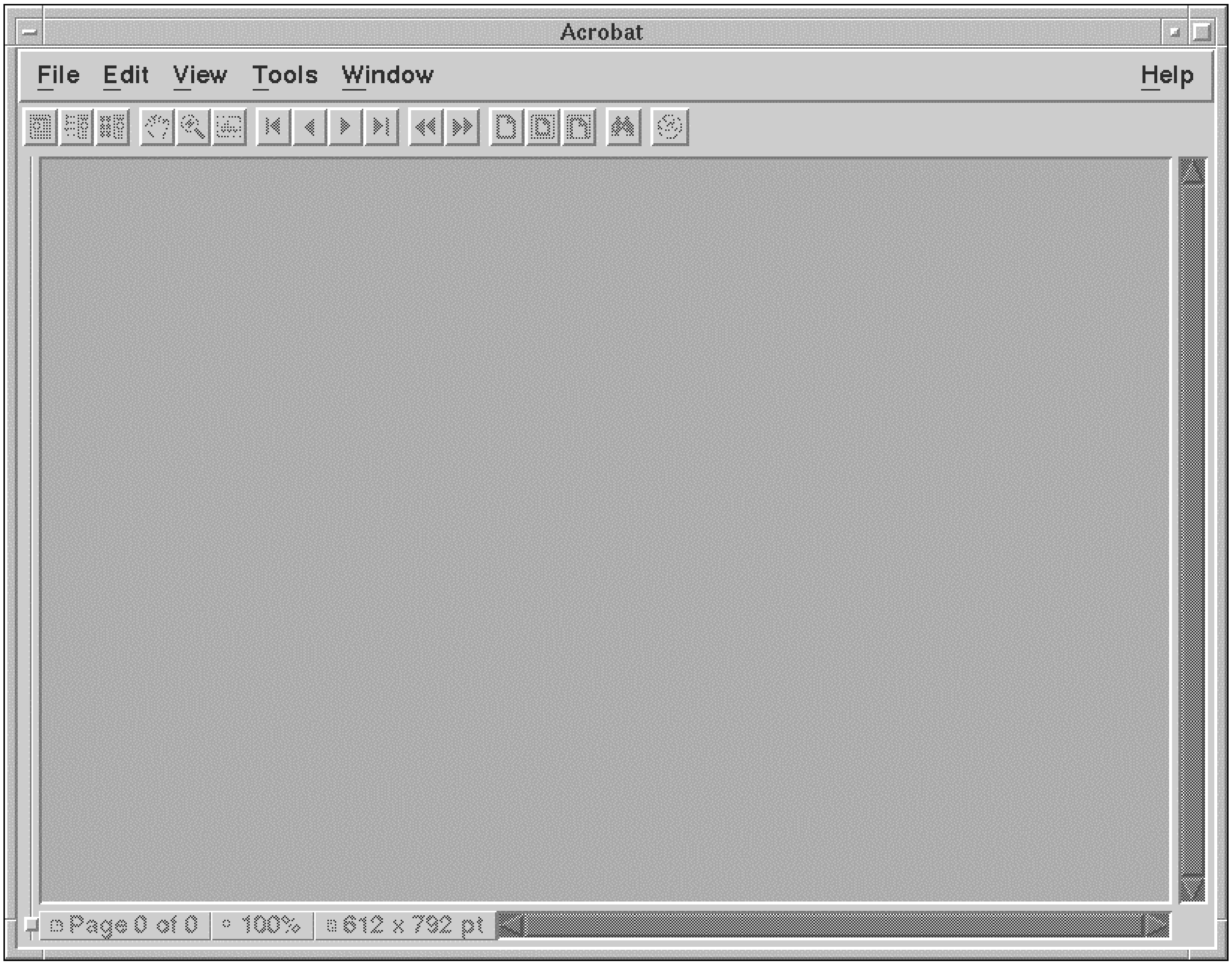
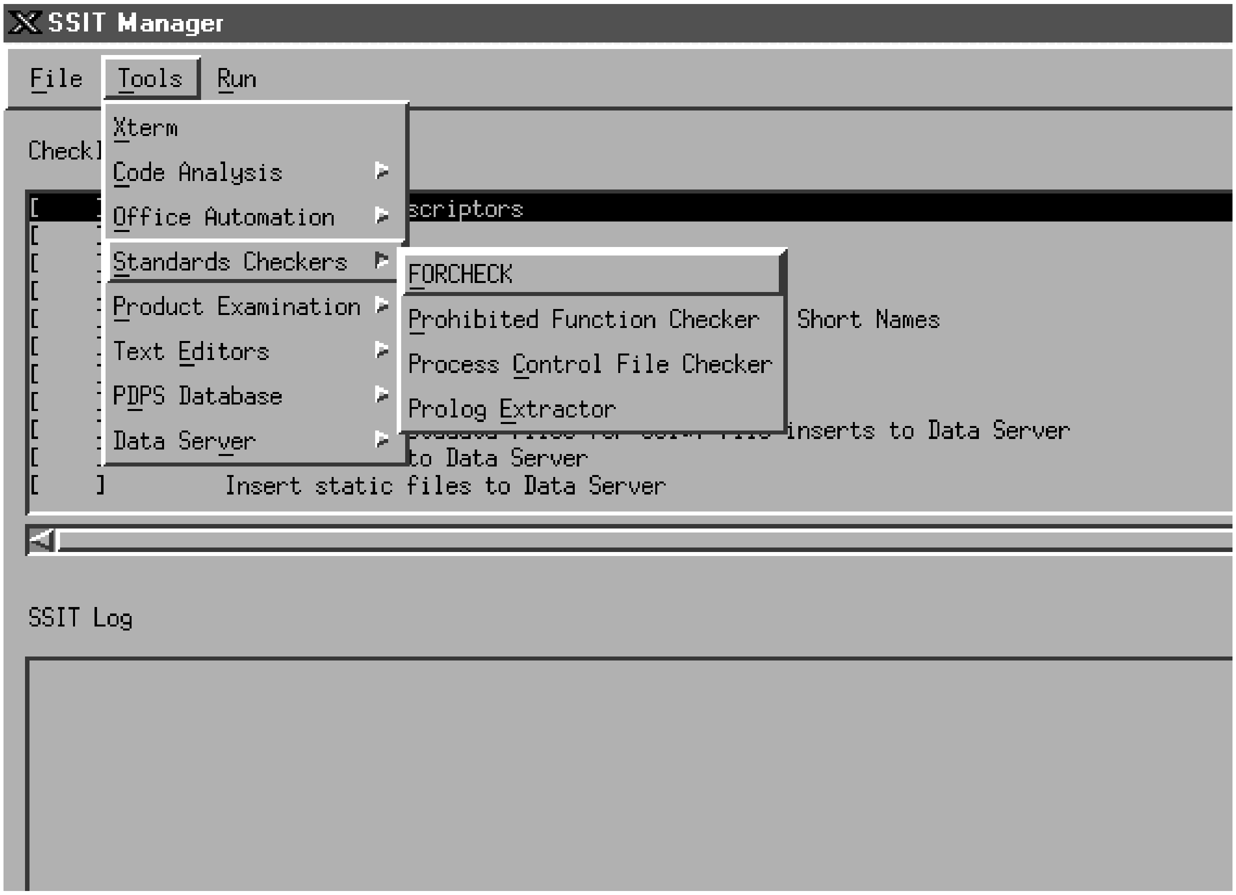


Figure 4.5.1-7. Acrobat Main Screen

4.5.1.2.5 Standards Checkers

The Standards Checkers menu shown in Figure 4.5.1-8 provides tools to assist operators in validating DAPs.



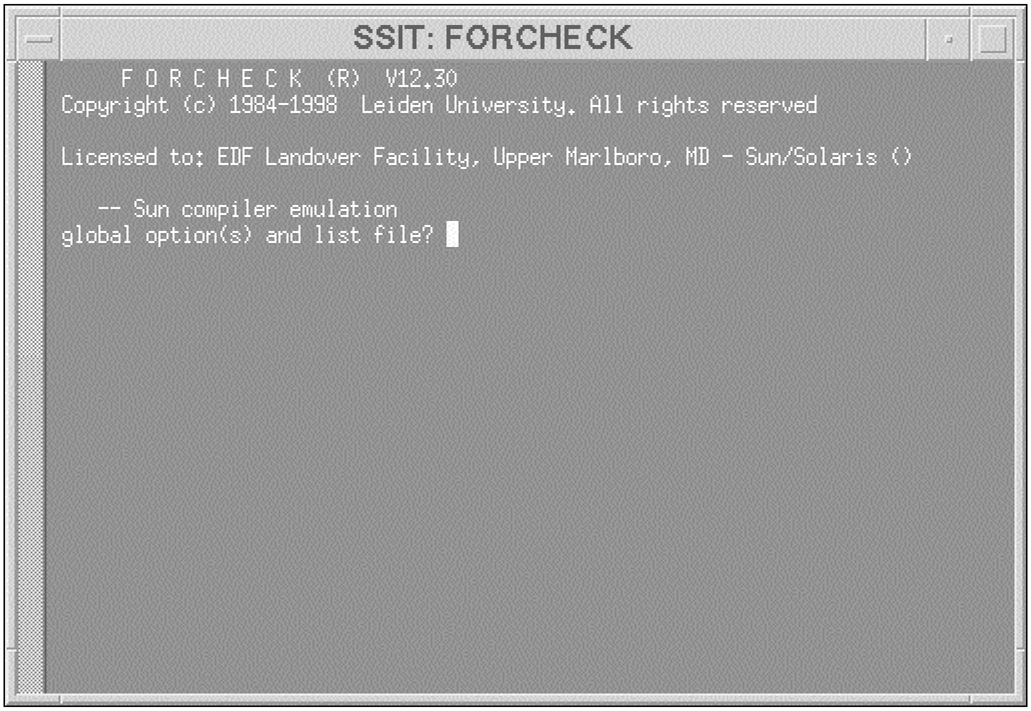
[ ] Insert static files to Data Server

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

4.5.1.2.5.1 FORCHECK

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



```
     F O R C H E C K  (R)  V12.30
Copyright (c) 1984-1998  Leiden University. All rights reserved

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

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

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

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

4.5.1.2.5.2 Prohibited Function Checker

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

The PFC GUI provides the following functionality:

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

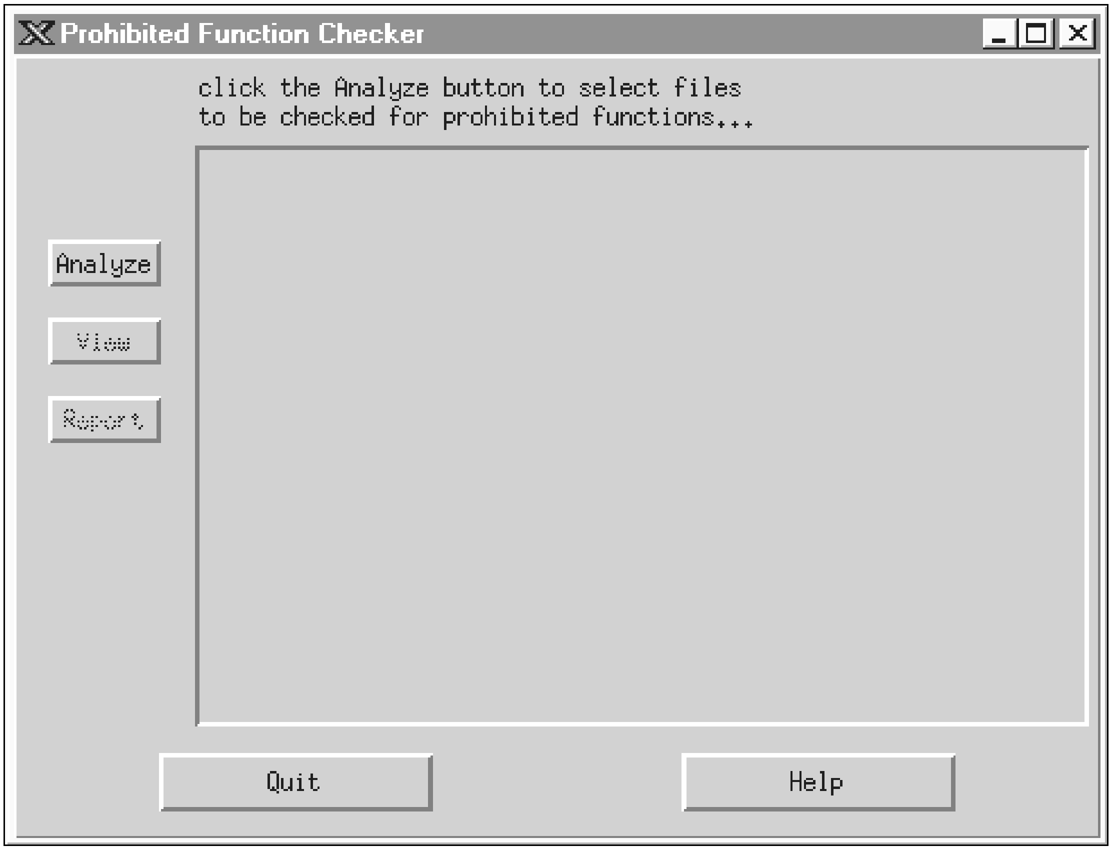
Table 4.5.1-6 lists the language and associated file extensions that can be checked for prohibited function calls using the PFC Tool.

Table 4.5.1-6. File Name Extensions

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

4.5.1.2.5.2.1 Checking for Prohibited Functions

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



click the Analyze button to select files
to be checked for prohibited functions...

Figure 4.5.1-10. Prohibited Function Checker Pop-up

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

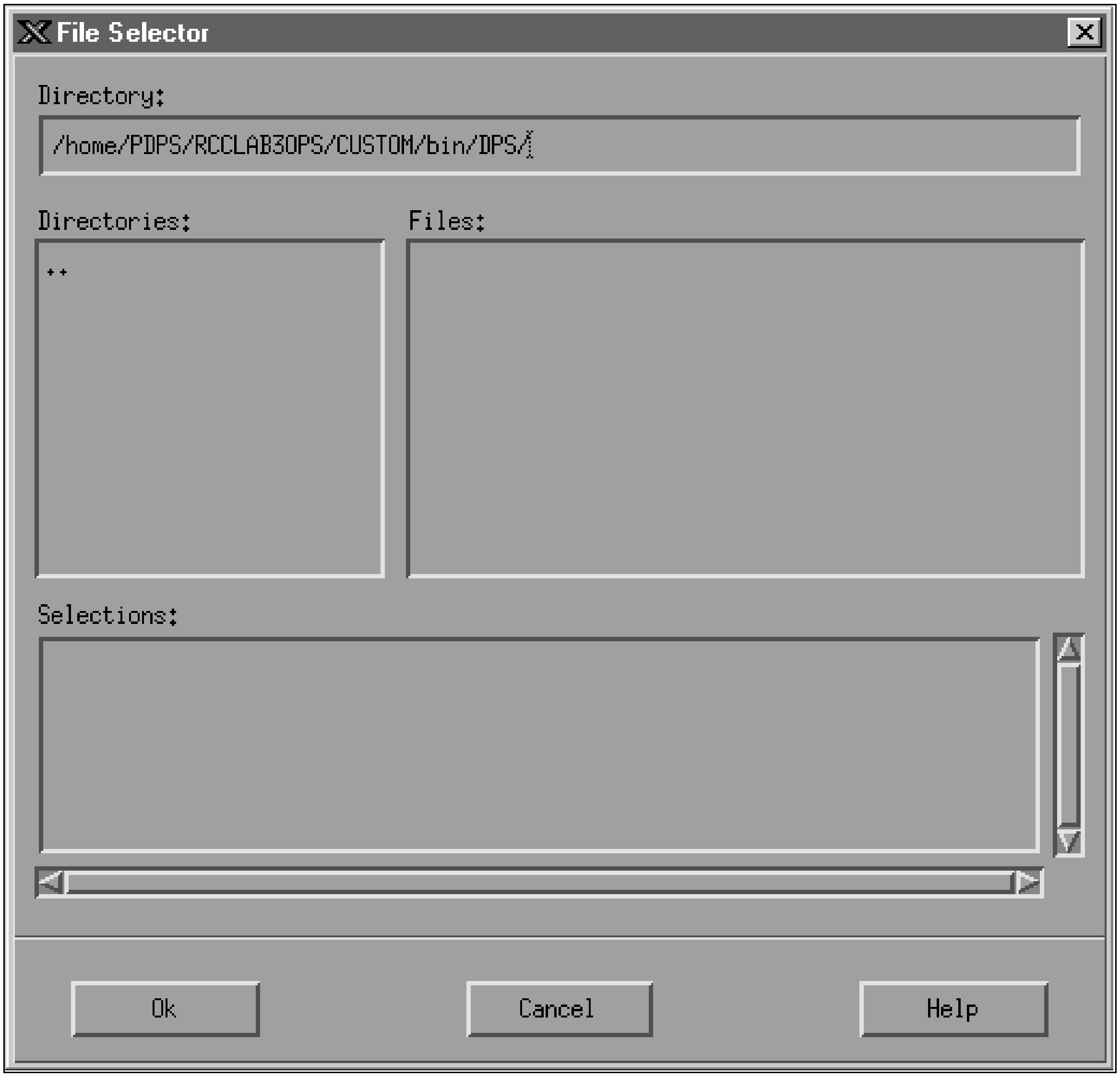
Table 4.5.1-7. Prohibited Function Checker Pop-up Field Descriptions (1 of 2)

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

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

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

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



/home/PDPS/RCCLAB3OPS/CUSTOM/bin/DPS/

..

Figure 4.5.1-11. File Selector Pop-up

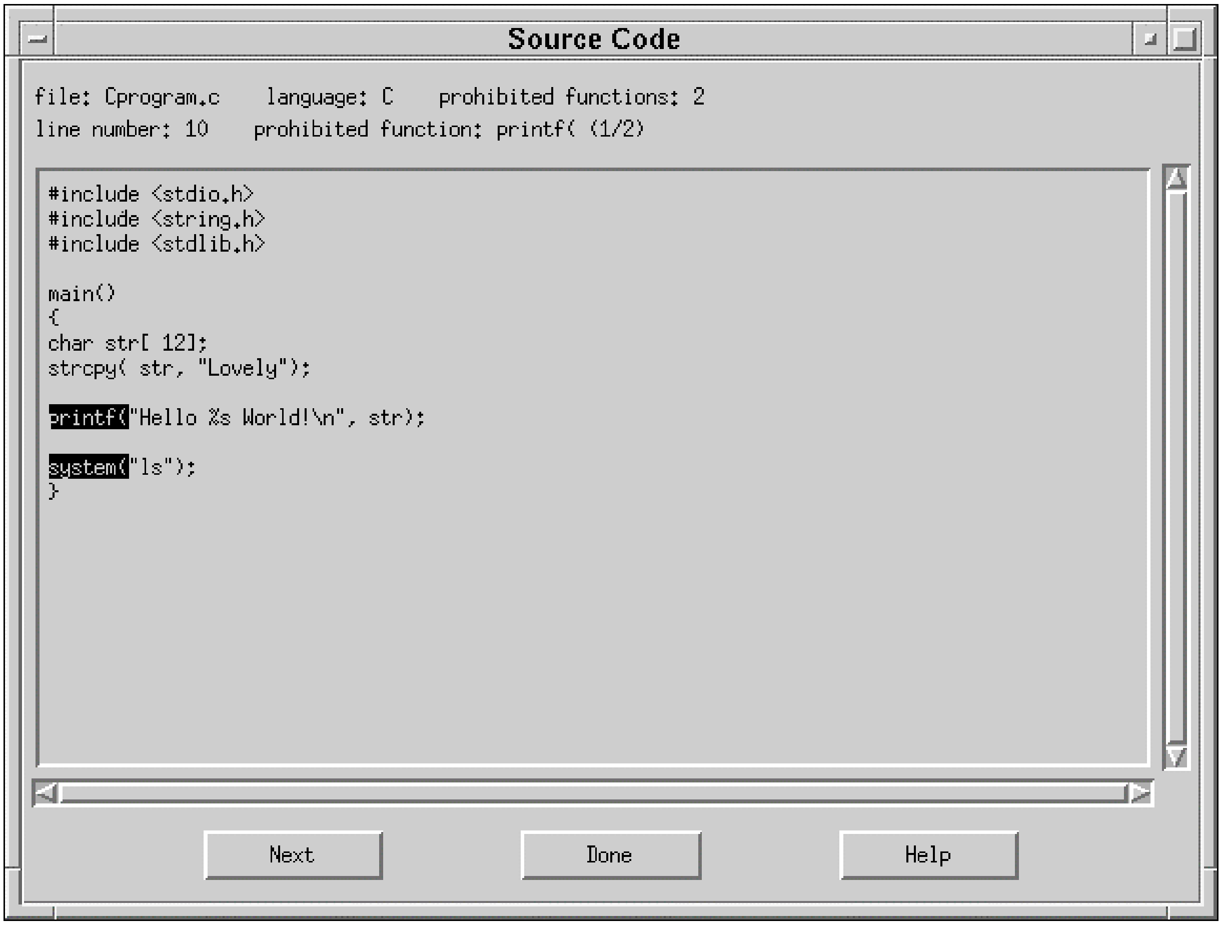
Table 4.5.1-8 describes the fields on the File Selector pop-up window.

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

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

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

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



file: Cprogram.c   language: C   prohibited functions: 2

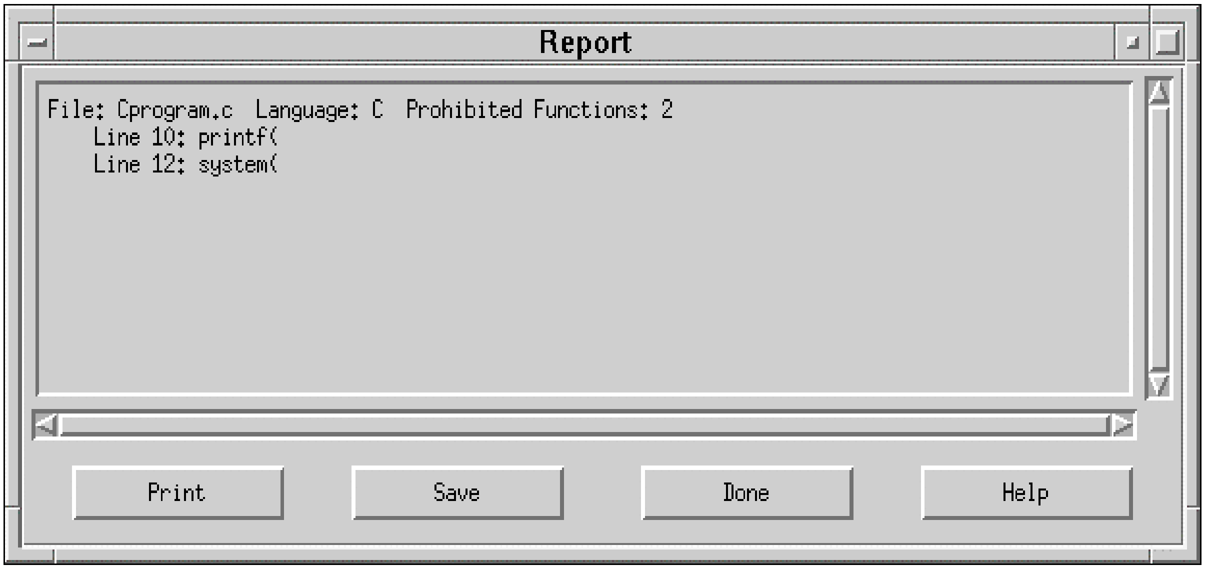
line number: 10   prohibited function: printf((1/2)

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

main()
{
char str[ 12];
strcpy( str, "Lovely");

printf("Hello %s World!\n", str);

system("ls");
}
```

Figure 4.5.1-12. Source Code Pop-up

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

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



```
File: Cprogram.c  Language: C  Prohibited Functions: 2
    Line 10: printf(
    Line 12: system(
```

Figure 4.5.1-13. Report Pop-up

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

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

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

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

4.5.1.2.5.3 Process Control File Checker

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

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

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

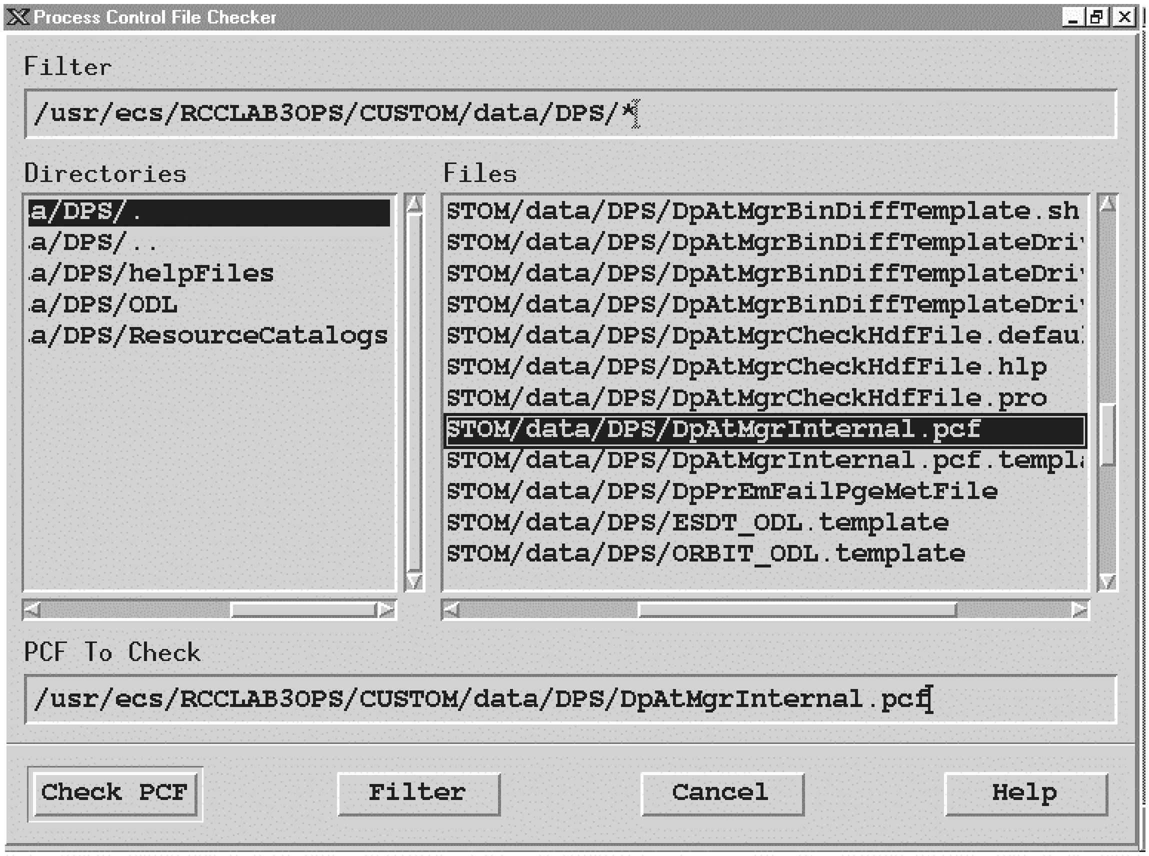
The PCFs to be checked are available, accessible, and have read permission for the user.

4.5.1.2.5.3.1 Process Control File Checker

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

4.5.1.2.5.3.2 Quick Start Using This Tool

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



Filter
/usr/ecs/RCCLAB3OPS/CUSTOM/data/DPS/*

Directories
```
a/DPS/.
a/DPS/..
a/DPS/helpFiles
a/DPS/ODL
a/DPS/ResourceCatalogs
```

Files
```
STOM/data/DPS/DpAtMgrBinDiffTemplate.sh
STOM/data/DPS/DpAtMgrBinDiffTemplateDri
STOM/data/DPS/DpAtMgrBinDiffTemplateDri
STOM/data/DPS/DpAtMgrBinDiffTemplateDri
STOM/data/DPS/DpAtMgrCheckHdfFile.defau
STOM/data/DPS/DpAtMgrCheckHdfFile.hlp
STOM/data/DPS/DpAtMgrCheckHdfFile.pro
STOM/data/DPS/DpAtMgrInternal.pcf
STOM/data/DPS/DpAtMgrInternal.pcf.templ
STOM/data/DPS/DpPrEmFailPgeMetFile
STOM/data/DPS/ESDT_ODL.template
STOM/data/DPS/ORBIT_ODL.template
```

PCF To Check
/usr/ecs/RCCLAB3OPS/CUSTOM/data/DPS/DpAtMgrInternal.pcf

Figure 4.5.1-14. PCF Checker Pop-up

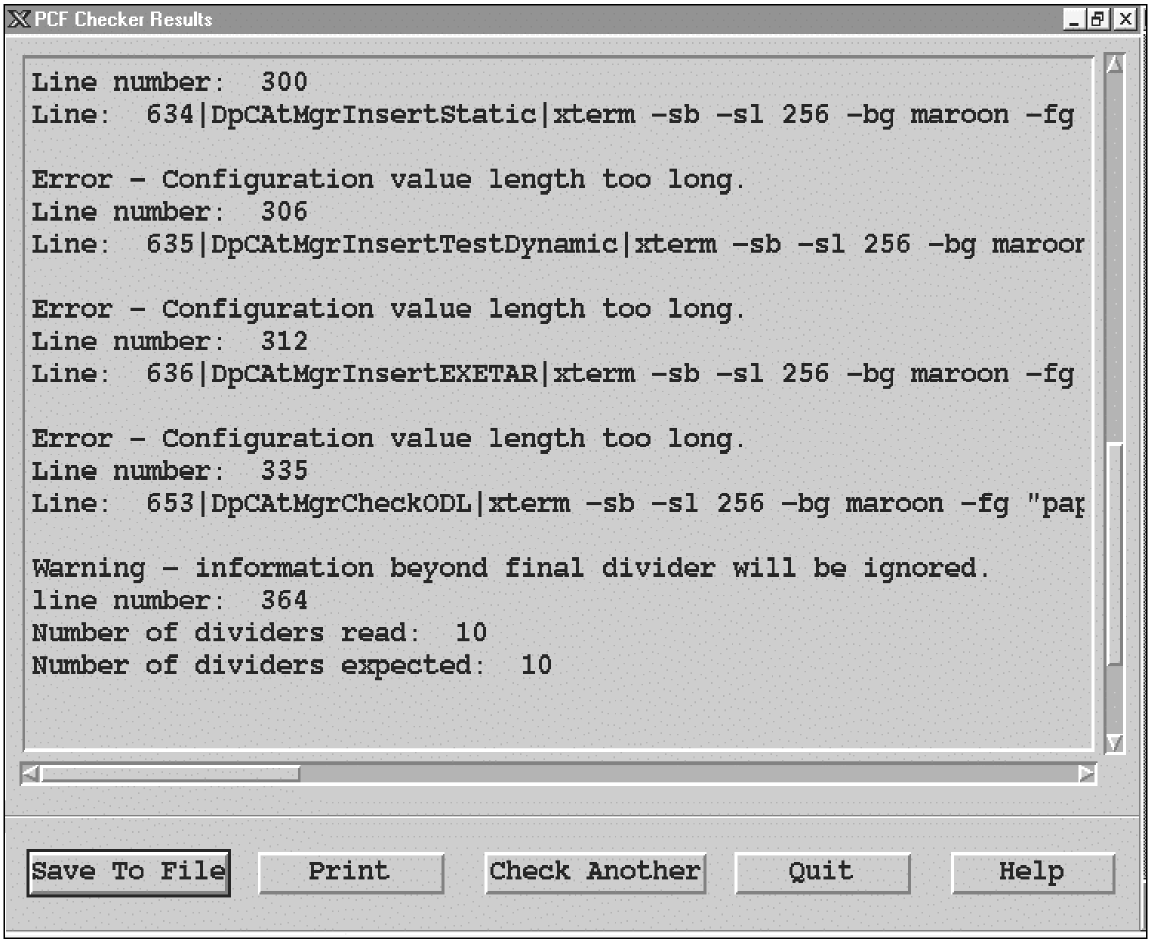
Table 4.5.1-9 describes the PCF Checker's fields.

Table 4.5.1-9. PCF Checker Field Description

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

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

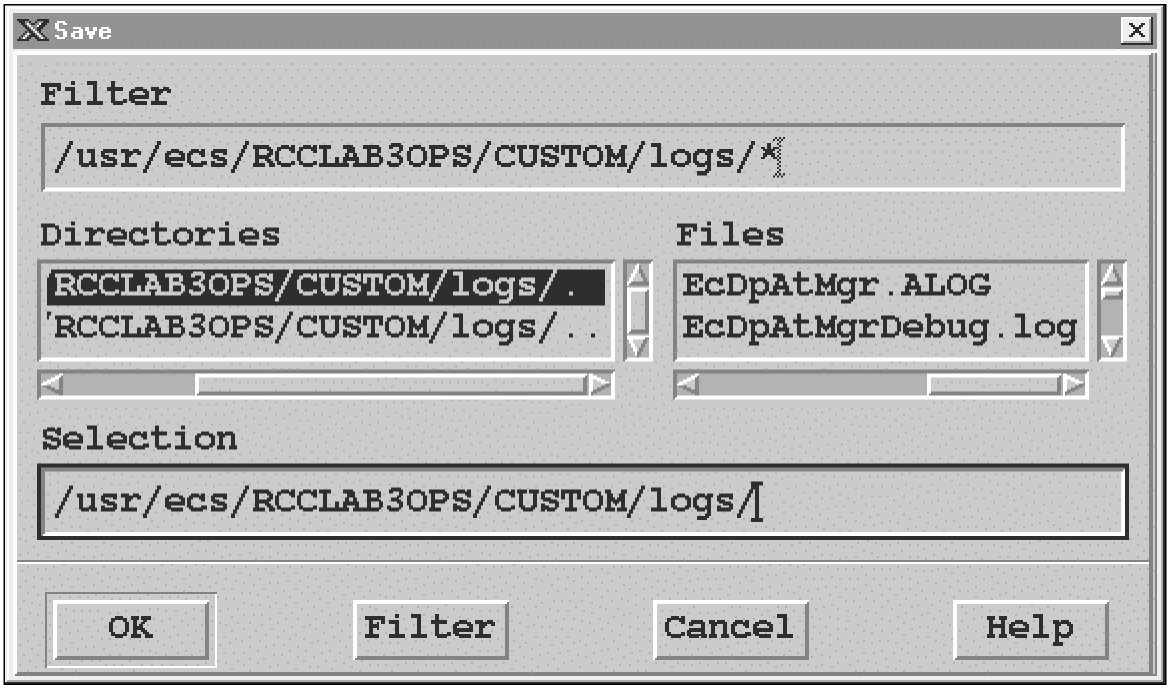
If the selected file does not exist, an error dialog is displayed to inform the user. Similarly, if the specified PCF exists but cannot be opened, an error dialog pops up to inform the user. Dialogs also pop up to inform the user of any fatal system errors or errors encountered in calls to toolkit functions.



```
Line number:  300
Line:  634|DpCAtMgrInsertStatic|xterm -sb -sl 256 -bg maroon -fg

Error - Configuration value length too long.
Line number:  306
Line:  635|DpCAtMgrInsertTestDynamic|xterm -sb -sl 256 -bg maroor

Error - Configuration value length too long.
Line number:  312
Line:  636|DpCAtMgrInsertEXETAR|xterm -sb -sl 256 -bg maroon -fg

Error - Configuration value length too long.
Line number:  335
Line:  653|DpCAtMgrCheckODL|xterm -sb -sl 256 -bg maroon -fg "par

Warning - information beyond final divider will be ignored.
line number:  364
Number of dividers read:  10
Number of dividers expected:  10
```

Figure 4.5.1-15. PCF Checker Results Pop-up

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



Filter

/usr/ecs/RCCLAB3OPS/CUSTOM/logs/*

Directories

RCCLAB3OPS/CUSTOM/logs/.
RCCLAB3OPS/CUSTOM/logs/..

Files

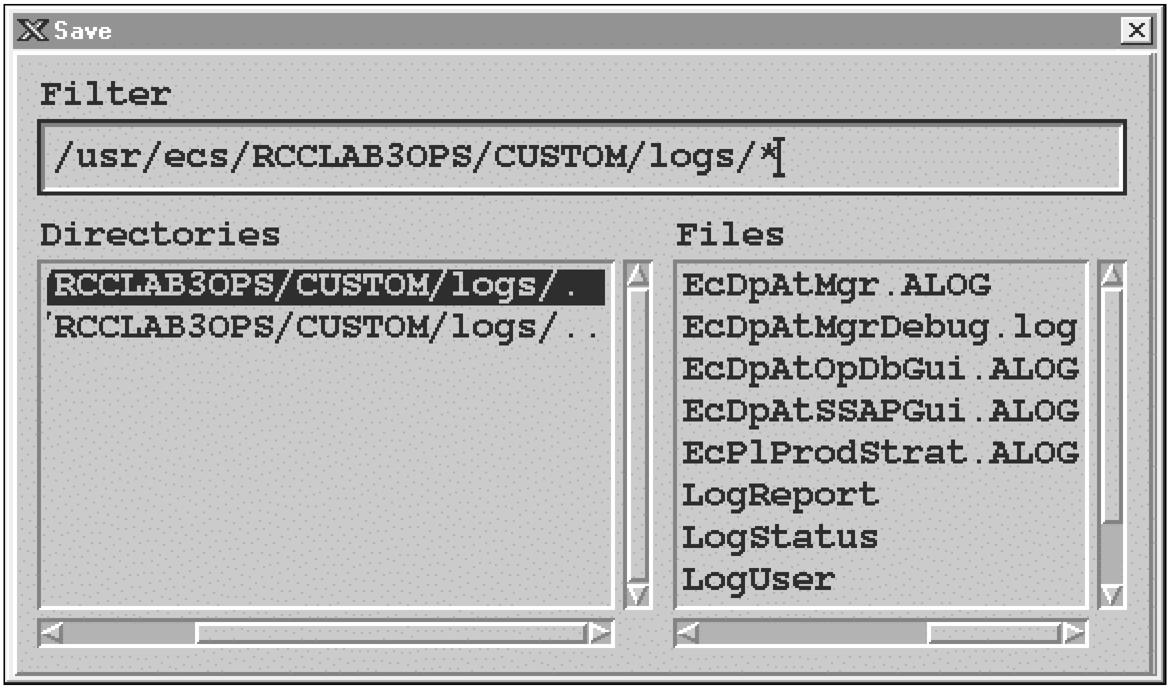
EcDpAtMgr.ALOG
EcDpAtMgrDebug.log

Selection

/usr/ecs/RCCLAB3OPS/CUSTOM/logs/

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

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



Filter

/usr/ecs/RCCLAB3OPS/CUSTOM/logs/*

Directories

RCCLAB3OPS/CUSTOM/logs/.
'RCCLAB3OPS/CUSTOM/logs/..

Files

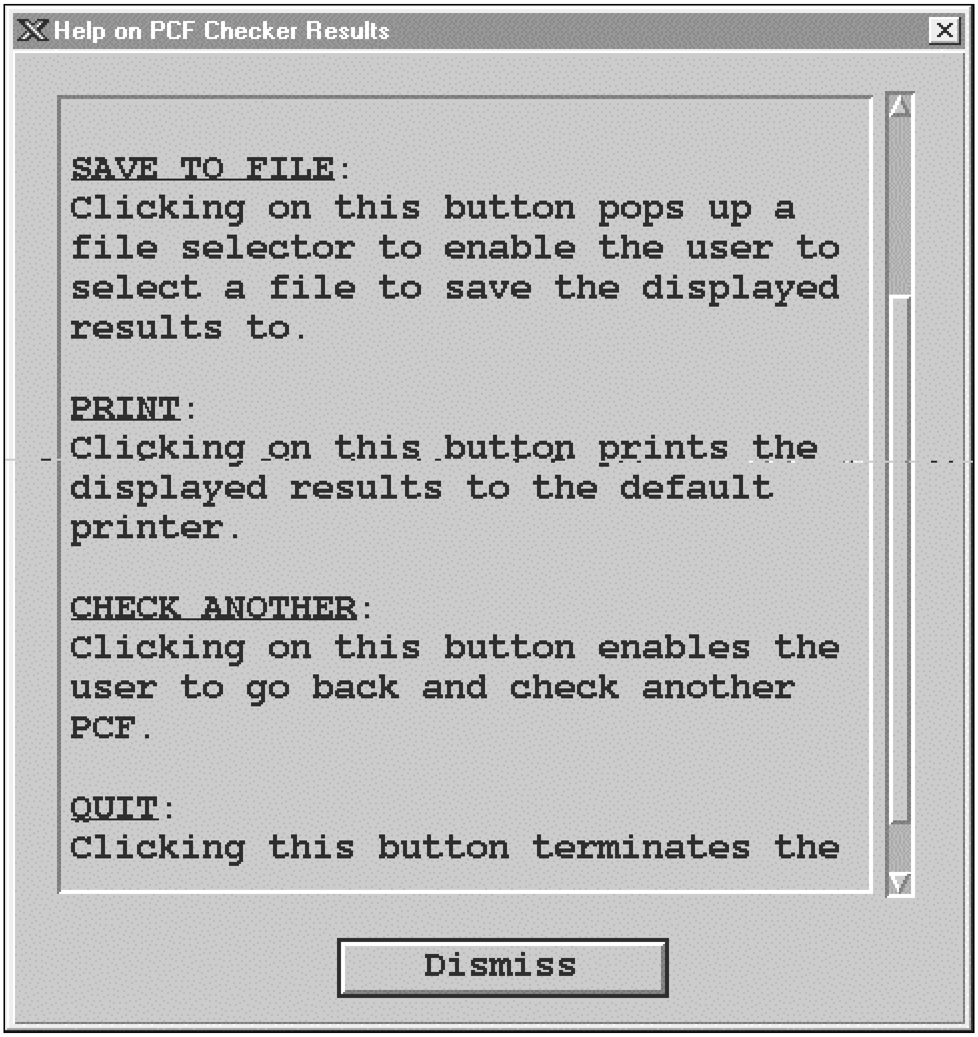
EcDpAtMgr.ALOG
EcDpAtMgrDebug.log
EcDpAtOpDbGui.ALOG
EcDpAtSSAPGui.ALOG
EcPlProdStrat.ALOG
LogReport
LogStatus
LogUser

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

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

The Cancel button terminates the running (open) application.

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



```
SAVE TO FILE:
Clicking on this button pops up a
file selector to enable the user to
select a file to save the displayed
results to.

PRINT:
Clicking on this button prints the
displayed results to the default
printer.

CHECK ANOTHER:
Clicking on this button enables the
user to go back and check another
PCF.

QUIT:
Clicking this button terminates the
```

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

4.5.1.2.5.4 Prolog Extractor

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

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

Table 4.5.1-10. Prolog Extractor standard delimiters

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

The end delimiter is always !END.

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

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

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

Table 4.5.1-11. Prolog Extractor file extensions

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

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

Output from this tool is always written to a file.

By default, the SSI&T Manager internal PCF specifies that this output file be named prologs.txt in the local directory. You can change this by editing the internal PCF, 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. Figure 4.5.1-19 shows the Product Examination tool menu options on the SSI&T Manager Main screen. These tools assist in the maintenance of the science software by allowing the operator to view programs in different languages, view science data, or compare program or data output files.

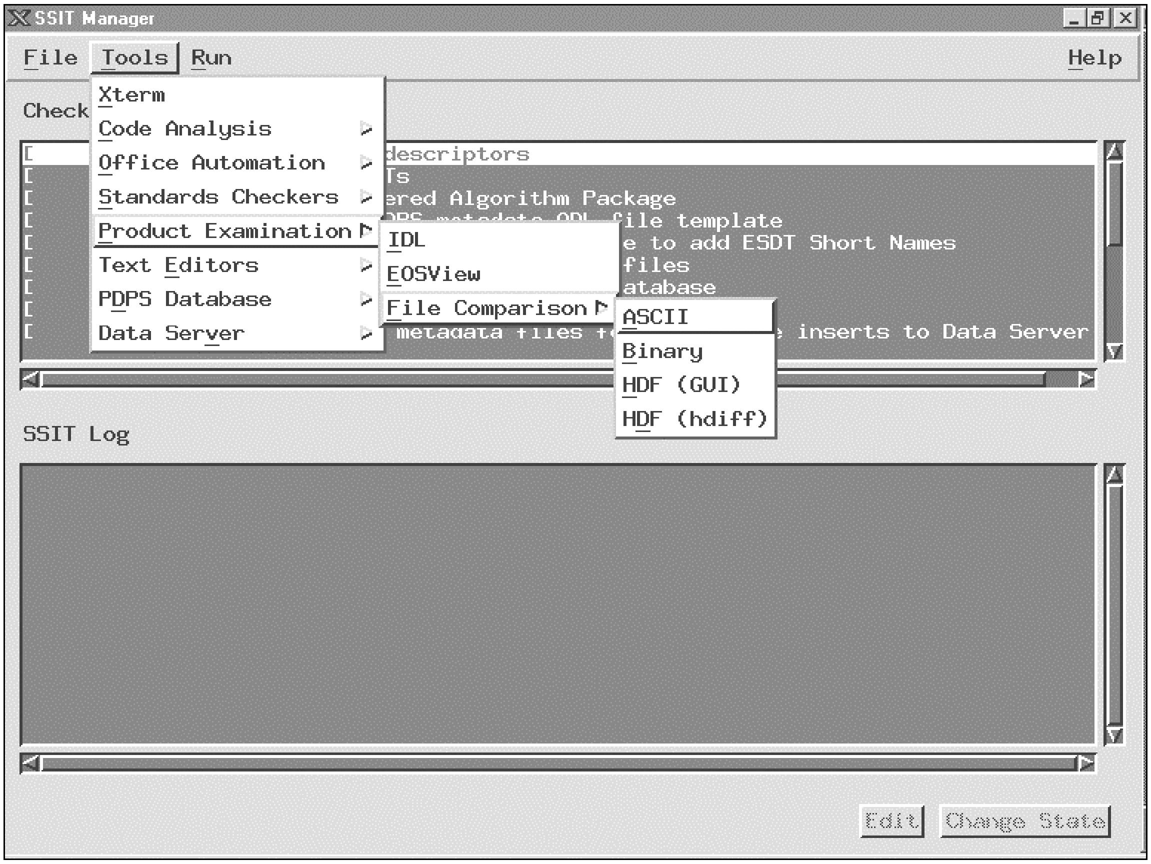


Figure 4.5.1-19. SSI&T Manager Main Screen with Product Examination Tools Displayed

4.5.1.2.6.1 IDL

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

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

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

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

4.5.1.2.6.2 EOSView

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

EOSView is described in Section 4.12.5, "EOSView."

4.5.1.2.6.3 File Comparison Tools

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

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

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

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

4.5.1.2.6.3.1 ASCII File Comparison Tool

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

4.5.1.2.6.3.2 Binary File Difference Environment

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

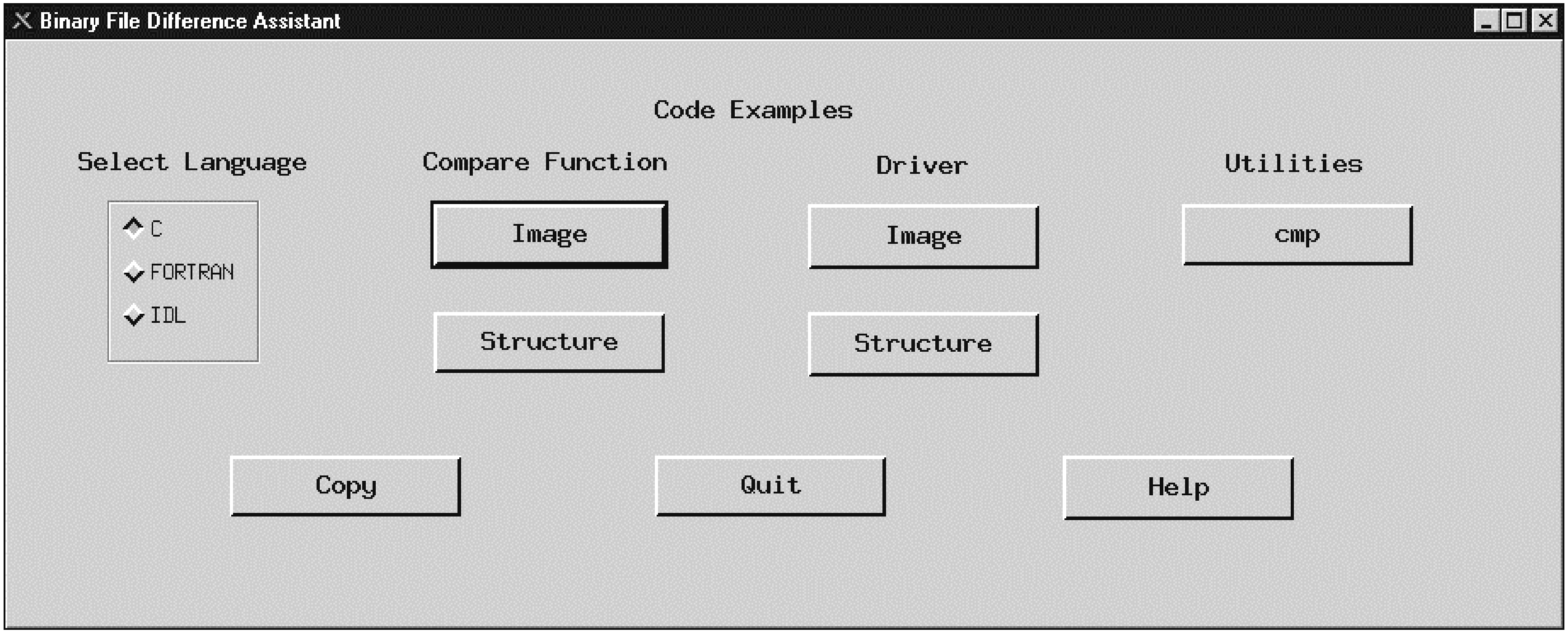


Figure 4.5.1-20. Binary File Comparison Pop-up

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

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

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

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

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

4.5.1.2.6.3.3 HDF GUI

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

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

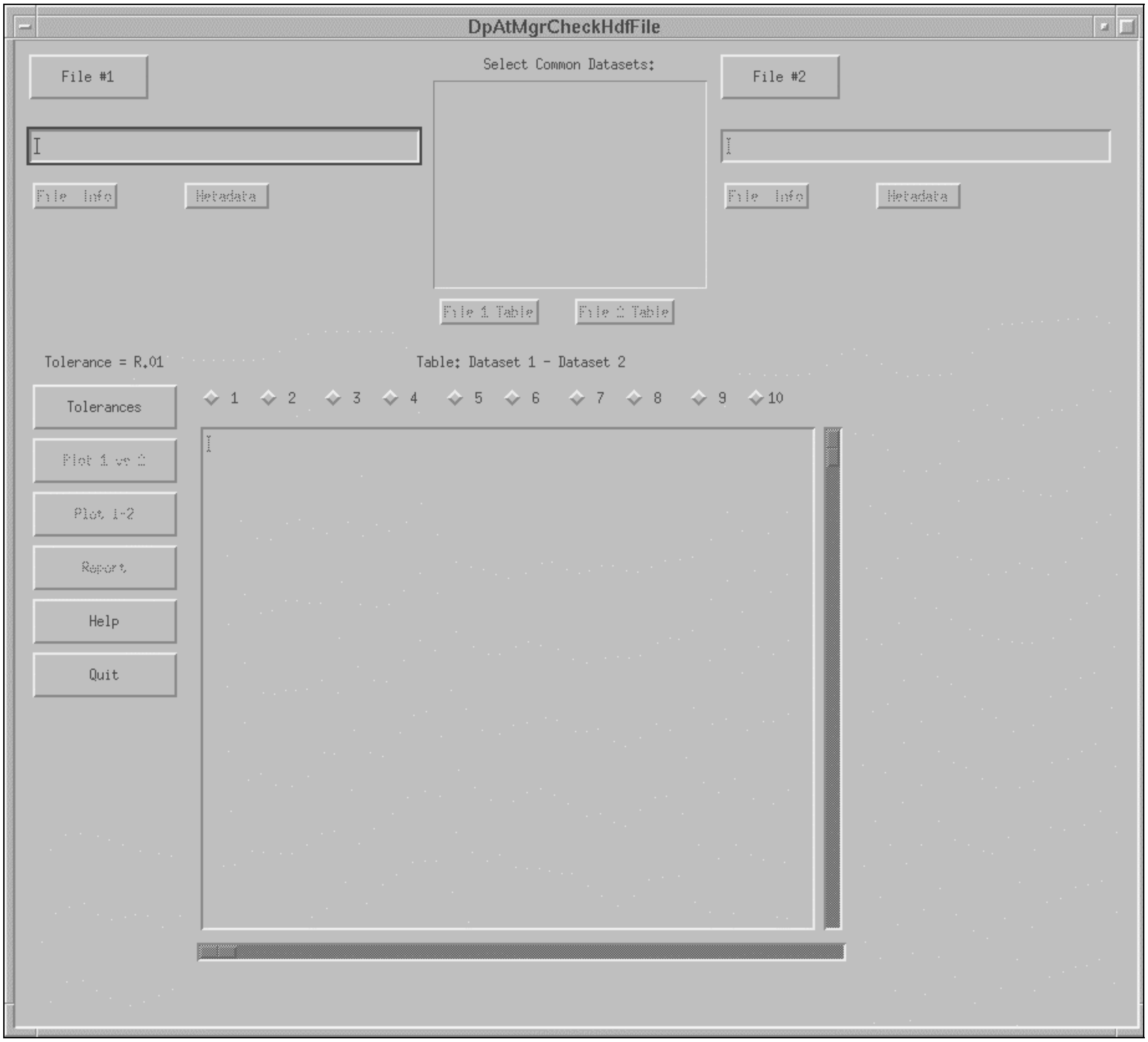
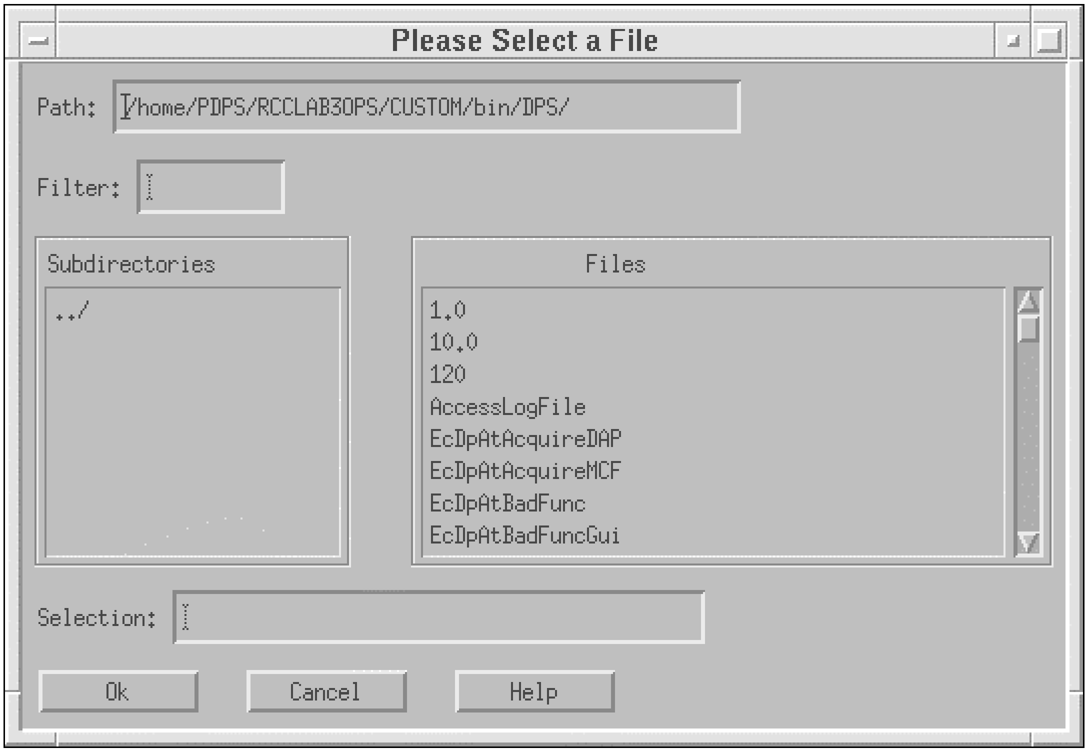


Figure 4.5.1-21. HDF Pop-up

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

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



Path: /home/PDPS/RCCLAB3OPS/CUSTOM/bin/DPS/

Filter:

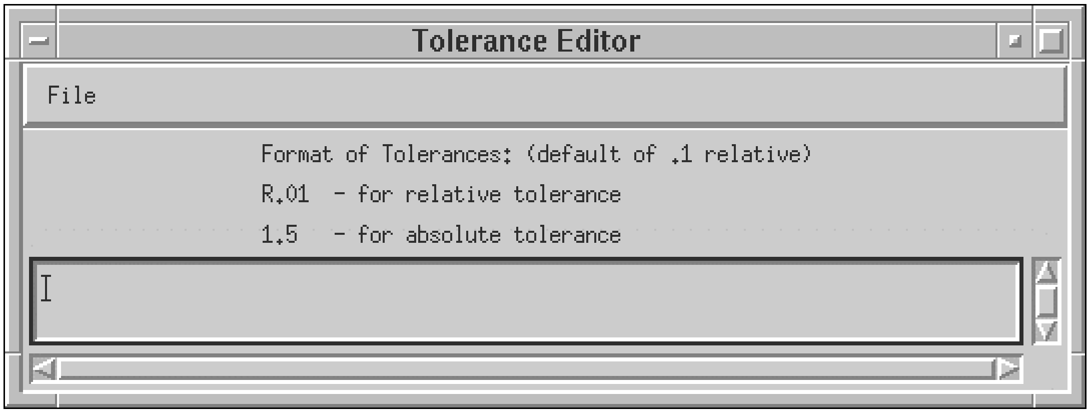
../

1.0
10.0
120
AccessLogFile
EcDpAtAcquireDAP
EcDpAtAcquireMCF
EcDpAtBadFunc
EcDpAtBadFuncGui

Selection:

Figure 4.5.1-22. File Selection Dialogue Pop-up

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



Format of Tolerances: (default of .1 relative)

R.01 - for relative tolerance

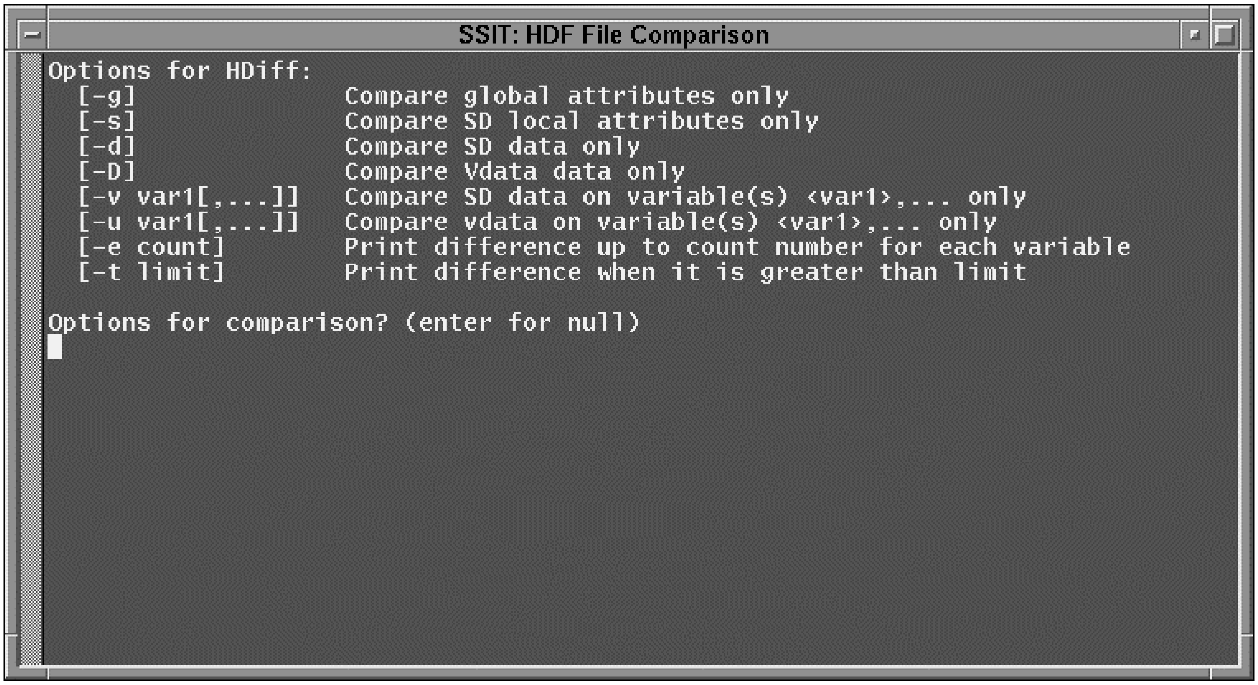
1.5 - for absolute tolerance

Figure 4.5.1-23. Tolerance Editor Pop-up

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

4.5.1.2.6.4 HDF File Comparison - hdiff

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



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

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

Figure 4.5.1-24. HDF (hdiff) Options

4.5.1.2.7 Text Editors

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

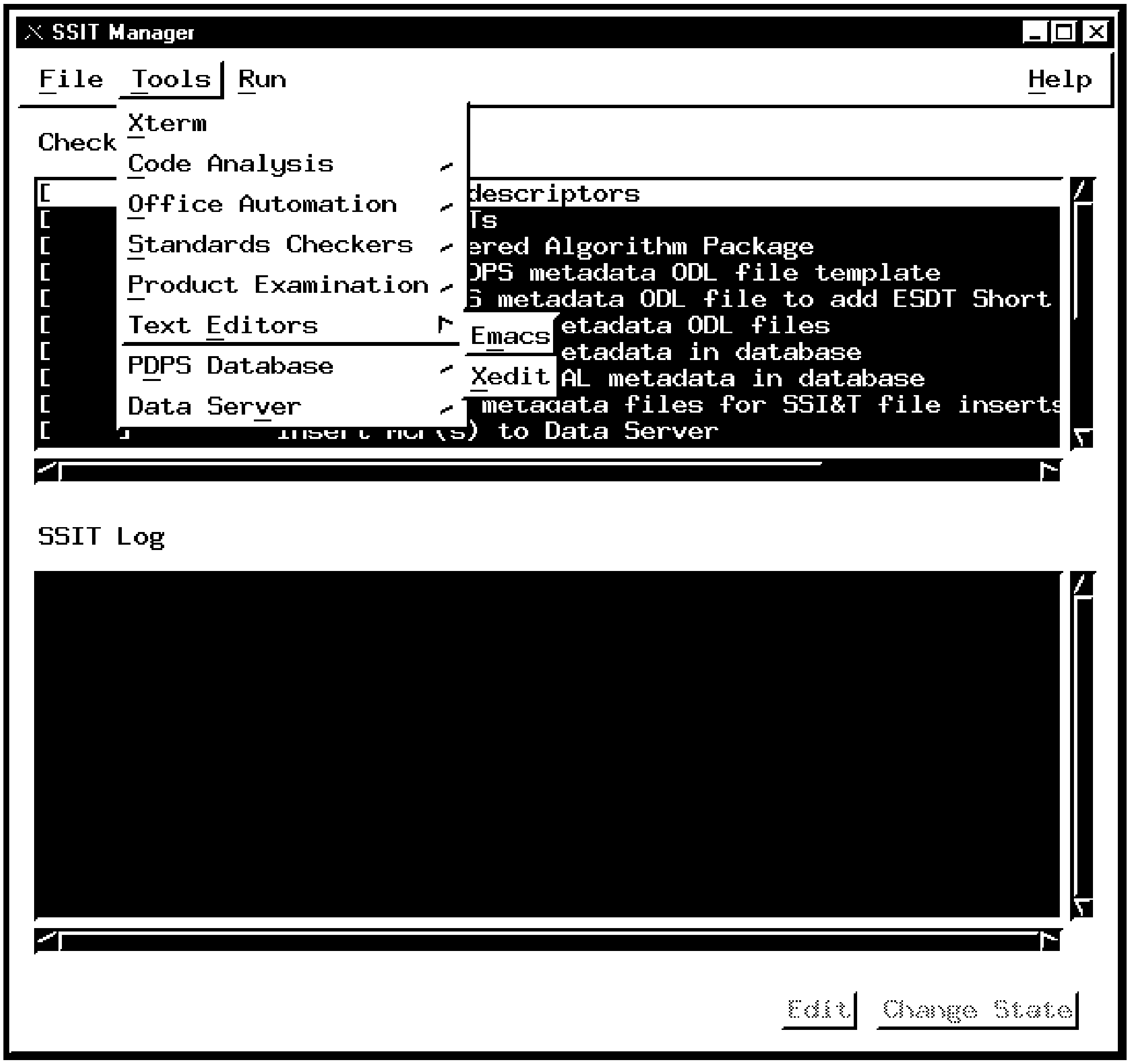
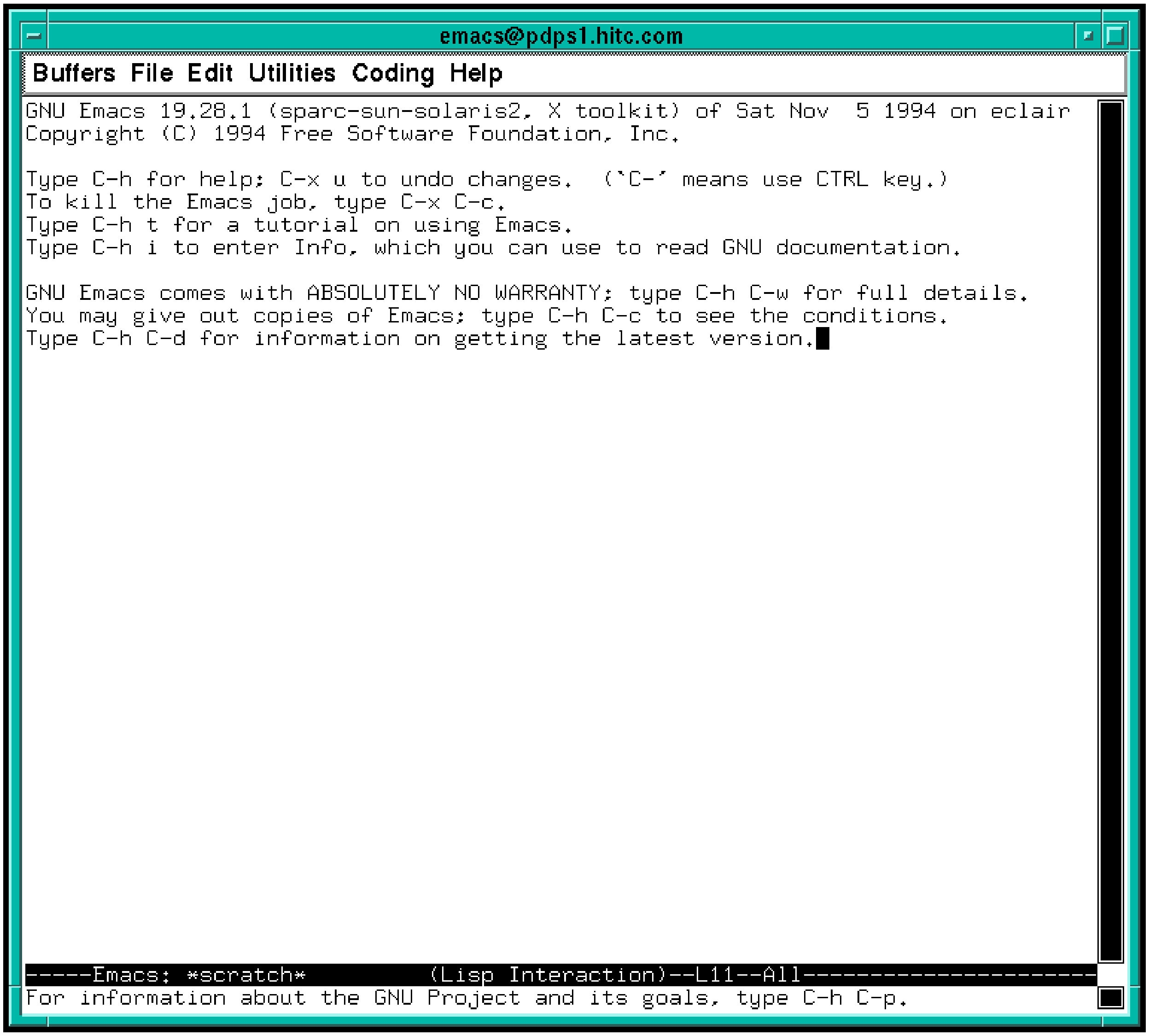


Figure 4.5.1-25. SSI&T Manager Main Screen with Text Editors Tool Menu Displayed

4.5.1.2.7.1 Emacs

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



```
GNU Emacs 19.28.1 (sparc-sun-solaris2, X toolkit) of Sat Nov  5 1994 on eclair
Copyright (C) 1994 Free Software Foundation, Inc.

Type C-h for help; C-x u to undo changes.  (`C-' means use CTRL key.)
To kill the Emacs job, type C-x C-c.
Type C-h t for a tutorial on using Emacs.
Type C-h i to enter Info, which you can use to read GNU documentation.

GNU Emacs comes with ABSOLUTELY NO WARRANTY; type C-h C-w for full details.
You may give out copies of Emacs; type C-h C-c to see the conditions.
Type C-h C-d for information on getting the latest version.
```

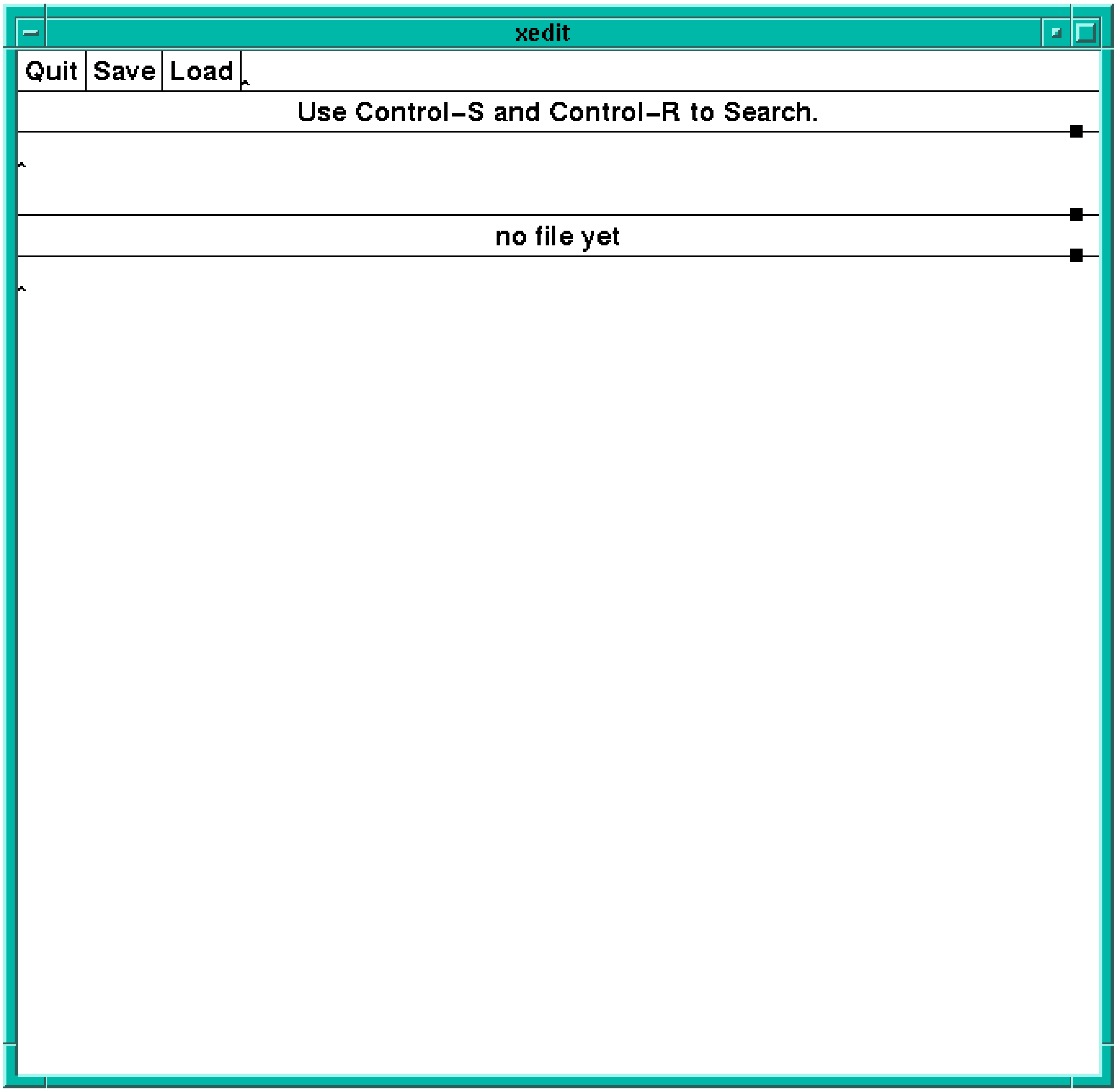
```
-----Emacs: *scratch*         (Lisp Interaction)--L11--All----------------------
For information about the GNU Project and its goals, type C-h C-p.
```

Figure 4.5.1-26. Emacs Pop-up

4.5.1.2.7.2 Xedit

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

>**man xedit**



Use Control-S and Control-R to Search.

no file yet

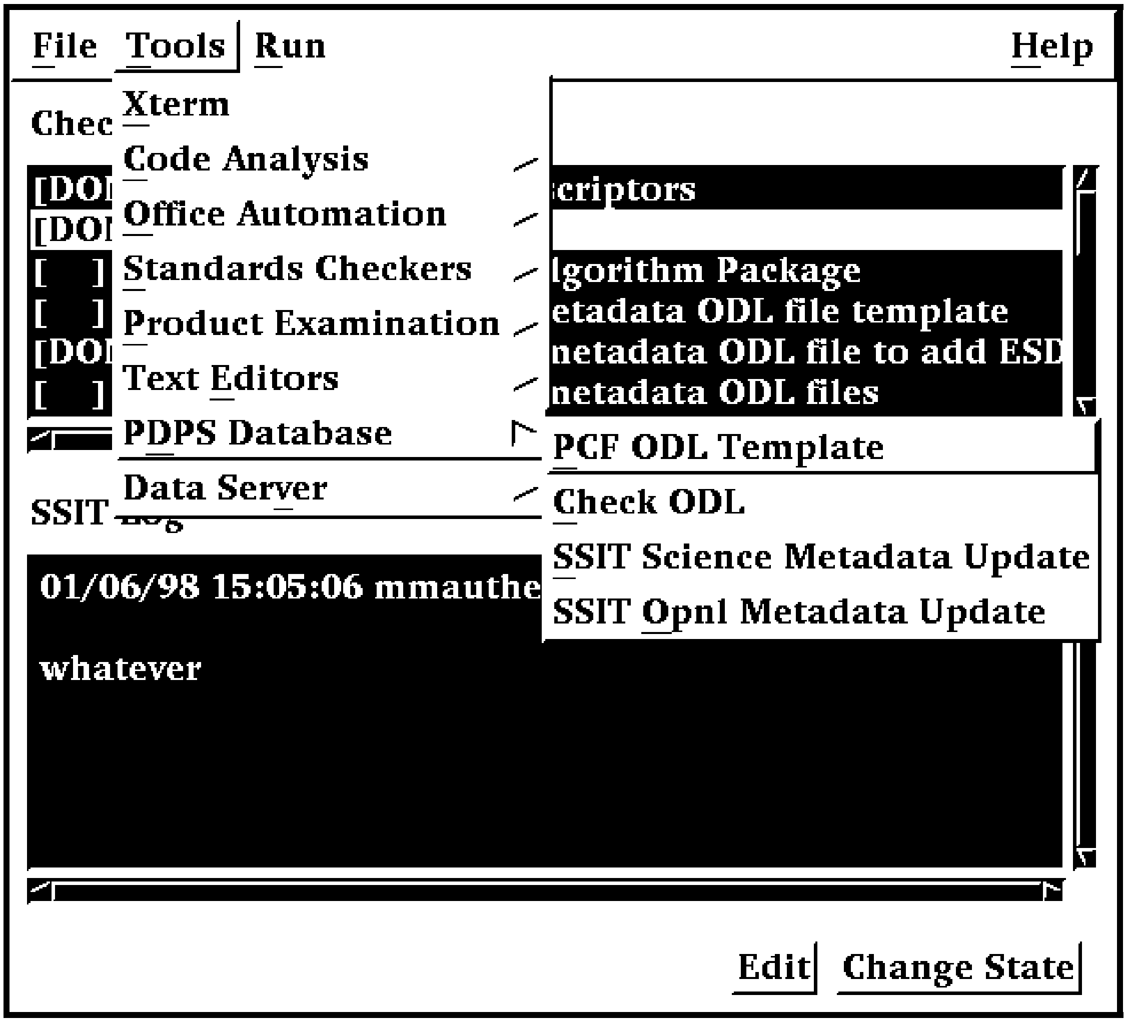
Figure 4.5.1-27. Xedit Pop-up

4.5.1.2.8 PDPS Database

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

The PDPS Database Update Tools consist of four components;

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



```
01/06/98 15:05:06 mmauthe

whatever
```

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

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

4.5.1.2.8.1 PCF ODL Template and ODL file Checking

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

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

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

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

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

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

4.5.1.2.8.2 Check ODL

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

The ODL Checker program is command line driven.

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

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

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

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

ECS Mode of operations?

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

PGE name (max 10 characters)?

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

PGE version (max 5 characters)?

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

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

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

Any success or error messages are then displayed followed by:

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

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

Check ODL program output.

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

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

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

```
DpAtCheckOdlDpAtScienceMd::ProcessOdlFile (PGE):
  ODL Error in file /ecs/formal/PDPS/DPS/SSIT/src/Metadata/Test/PGE_BAD#ODL#09.odl
    Check LogStatus file
```

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

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

```
DpAtCheckOdlDpAtScienceMd::ProcessOdlFile (PGE):
  Unable to open PGE Science Metadata ODL file
          {FILENAME}
```

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

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

```
********** Check of PDPS SCIENCE metadata SUCCESSFUL **********
           NO ERRORS FOUND
```

4.5.1.2.8.3 SSI&T Science Metadata Update

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

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

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

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

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

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

An example of this file can be found in:

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

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

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

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

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

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

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

Once these files (PGE, ESDT, and RULE) have been created and all the blanks have been completely filled in, the user selects *SSI&T Science Metadata Update* from the *PDPS Database* submenu to run the program. If a PGE already exists, the operator is prompted to update it. If a PGE does not already exist, a character-based window opens and prompts the user for information needed to identify the PGE, configuration filename, ECS mode of operations, PGE name, version and profile ID. With this information the program updates the SSI&T version of the PDPS database with PGE and ESDT Science metadata read from the ODL files.

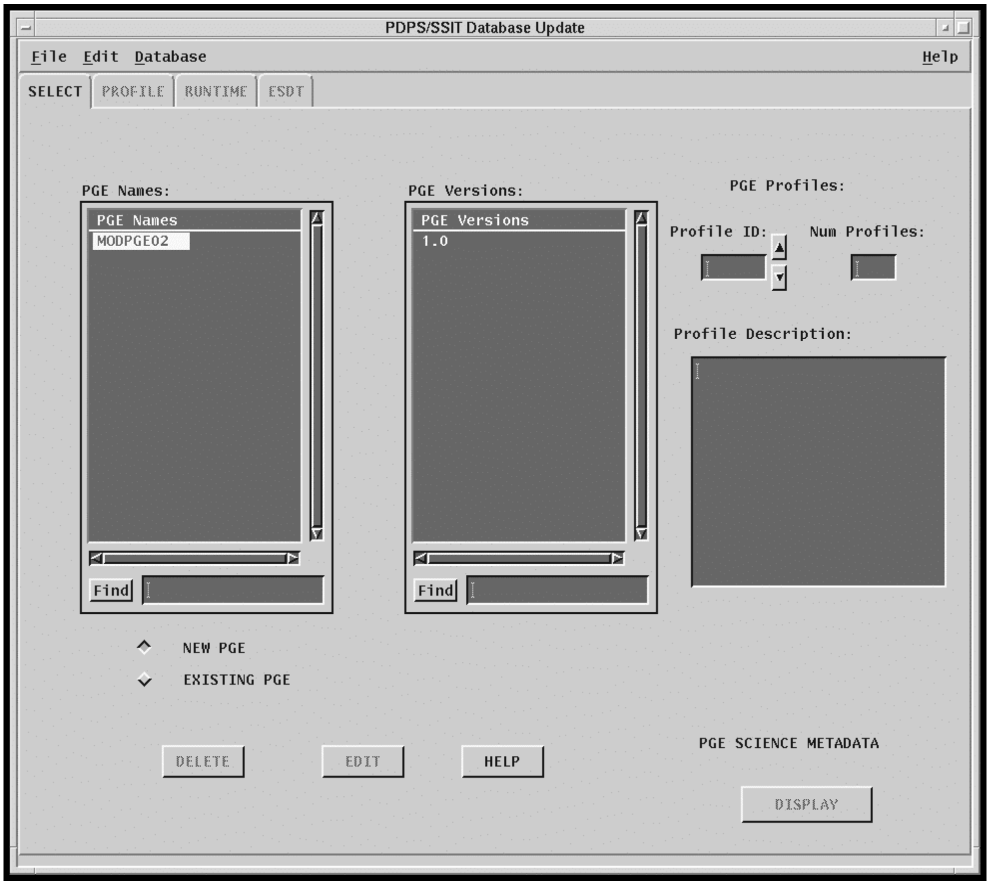
4.5.1.2.8.4 SSI&T Operational Metadata Update

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

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

Select View

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



PGE Names: PGE Names MODPGE02

PGE Versions: PGE Versions 1.0

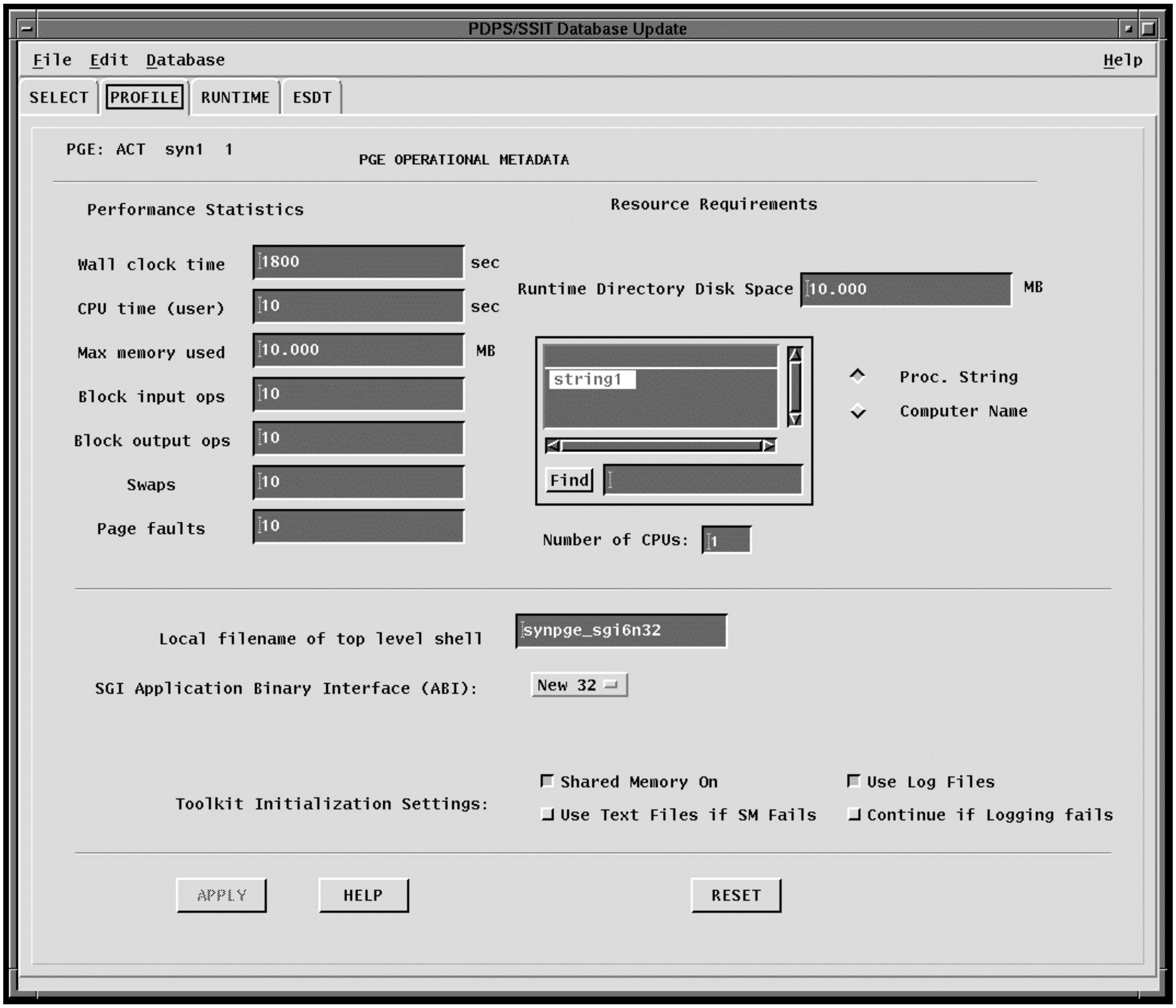
Figure 4.5.1-29. PDPS/SSI&T Database Operational Metadata Update Pop-up – Showing the SELECT Tab

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

Profile View

Selecting the *PROFILE* tab displays the profile information of the selected PGE. This view is shown in Figure 4.5.1-30. The view contains fields showing Resource Requirements and Performance Statistics of the selected PGE. Descriptions of each field are presented in Table 4.5.1-10. Default values for these fields (i.e. 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 the user must click on the *APPLY* button.

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



PGE: ACT syn1 1

PGE OPERATIONAL METADATA

Performance Statistics

Wall clock time 1800 sec

CPU time (user) 10 sec

Max memory used 10.000 MB

Block input ops 10

Block output ops 10

Swaps 10

Page faults 10

Resource Requirements

Runtime Directory Disk Space 10.000 MB

string1

◇ Proc. String

◇ Computer Name

Number of CPUs: 1

Local filename of top level shell synpge_sgi6n32

SGI Application Binary Interface (ABI): New 32

Toolkit Initialization Settings:

☑ Shared Memory On

☐ Use Text Files if SM Fails

☑ Use Log Files

☐ Continue if Logging fails

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

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

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

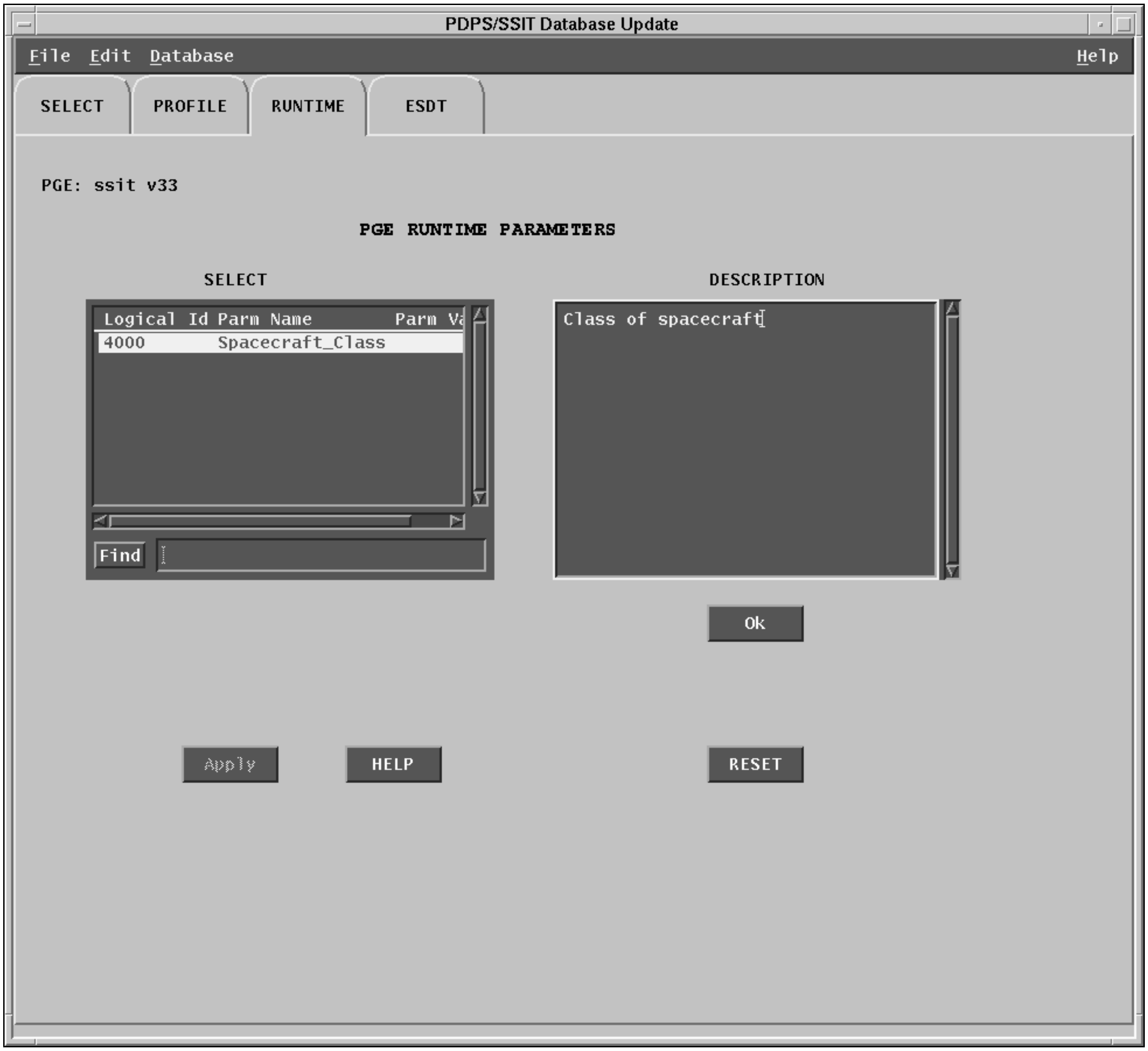
Field Name	Data Type	Size	Entry	Description
Wall clock Time	Float	N/A	User	Real time used by the process (in seconds); filled in both after processing with SCF Toolkit and with DAAC Toolkit (i.e., with AutoSys).
CPU Time	Float	N/A	User	CPU time used by the process in seconds.
Max Memory Used	Float	N/A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

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

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

Runtime View

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



PGE: ssit v33

Logical Id	Parm Name	Parm Va
4000	Spacecraft_Class	

Class of spacecraft

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

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

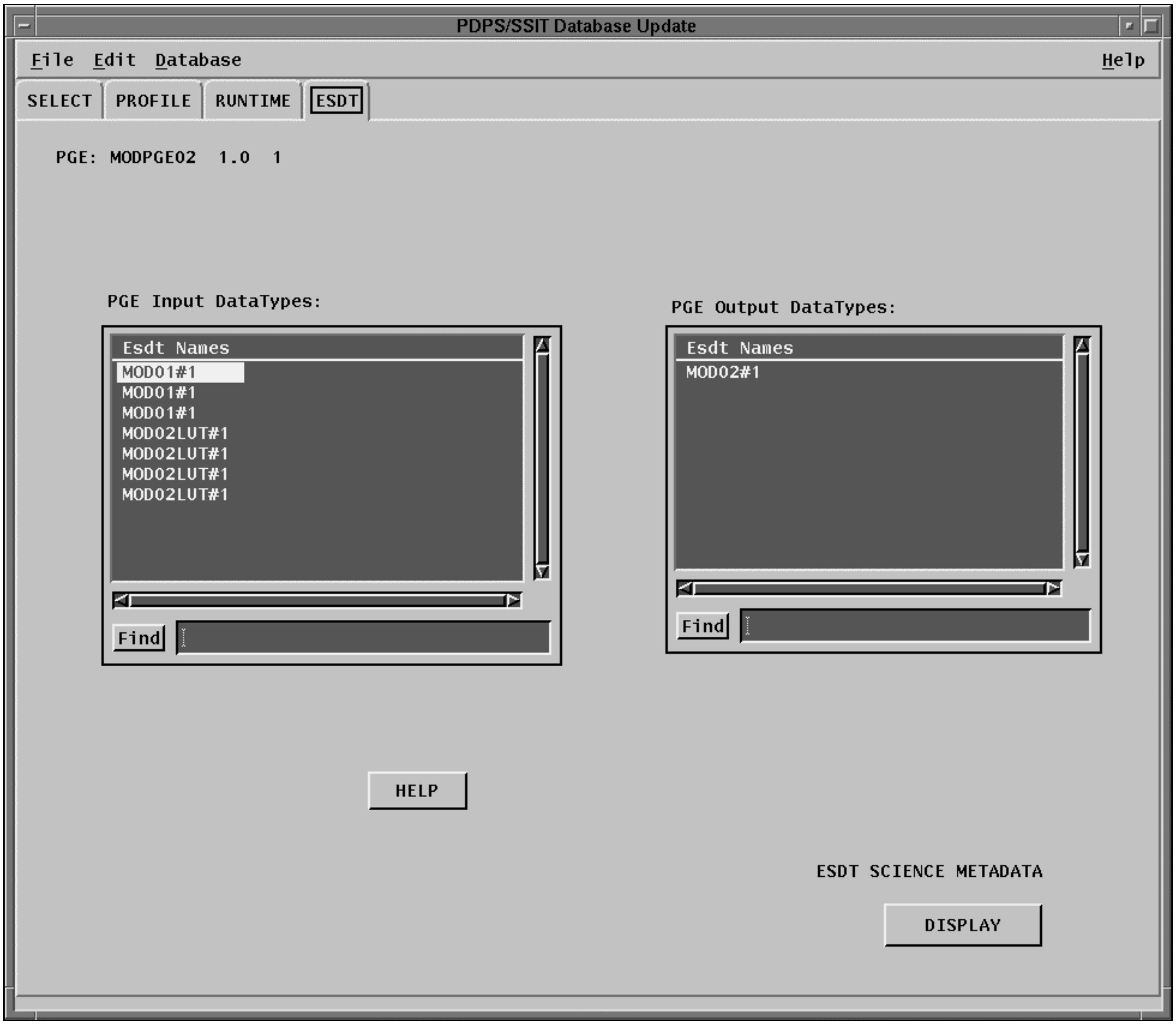
Table 4.5.1-13. PDPS/SSI&T Database Operational Metadata Update Runtime View Field Description

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

ESDT View

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

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



PGE: MODPGE02 1.0 1

PGE Input DataTypes:

Esdt Names
MOD01#1
MOD01#1
MOD01#1
MOD02LUT#1
MOD02LUT#1
MOD02LUT#1
MOD02LUT#1

PGE Output DataTypes:

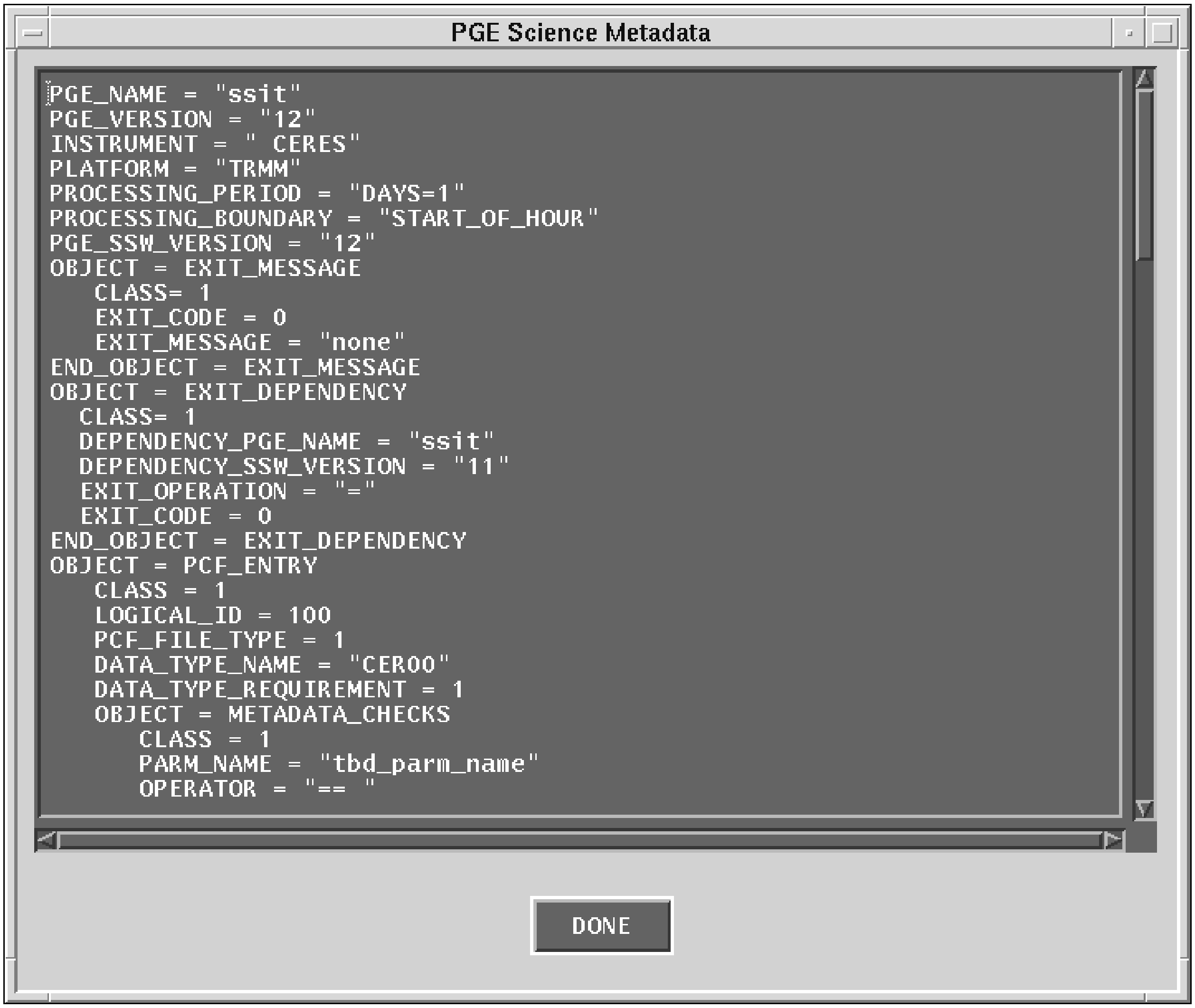
Esdt Names
MOD02#1

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

Display Screen

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

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

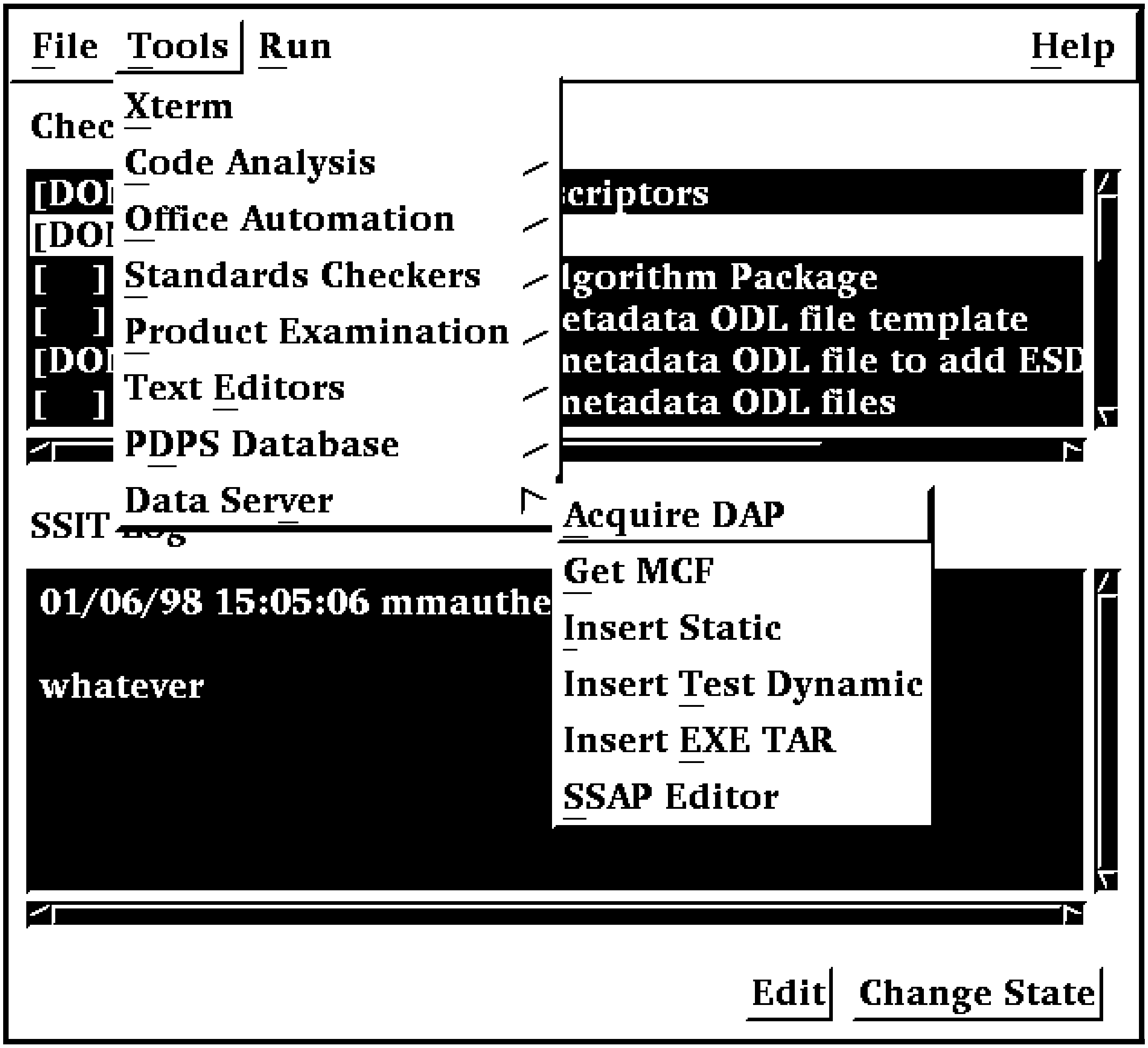


```
PGE_NAME = "ssit"
PGE_VERSION = "12"
INSTRUMENT = " CERES"
PLATFORM = "TRMM"
PROCESSING_PERIOD = "DAYS=1"
PROCESSING_BOUNDARY = "START_OF_HOUR"
PGE_SSW_VERSION = "12"
OBJECT = EXIT_MESSAGE
   CLASS= 1
   EXIT_CODE = 0
   EXIT_MESSAGE = "none"
END_OBJECT = EXIT_MESSAGE
OBJECT = EXIT_DEPENDENCY
  CLASS= 1
  DEPENDENCY_PGE_NAME = "ssit"
  DEPENDENCY_SSW_VERSION = "11"
  EXIT_OPERATION = "="
  EXIT_CODE = 0
END_OBJECT = EXIT_DEPENDENCY
OBJECT = PCF_ENTRY
   CLASS = 1
   LOGICAL_ID = 100
   PCF_FILE_TYPE = 1
   DATA_TYPE_NAME = "CER00"
   DATA_TYPE_REQUIREMENT = 1
   OBJECT = METADATA_CHECKS
      CLASS = 1
      PARM_NAME = "tbd_parm_name"
      OPERATOR = "== "
```

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

4.5.1.2.9 Data Server

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



```
01/06/98 15:05:06 mmauthe

whatever
```

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

4.5.1.2.9.1 Acquire DAP

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

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

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

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

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

4.5.1.2.9.2 Get MCF

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

4.5.1.2.9.3 Insert Static

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

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

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

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

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

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

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

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

ECS Mode of operations?

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

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

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

ESDT Version for the file(s) to insert?

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

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

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

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

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

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

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

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

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

Single Filename to Insert (including FULL path)?

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

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

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

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

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

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

Name of MFG file (enter to end list)?

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

Associated ASCII Metadata Filename to Insert?

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

Any success or error messages are then displayed followed by:

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

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

4.5.1.2.9.4 Insert Test Dynamic

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

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

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

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

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

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

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

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

ECS Mode of operations?

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

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

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

ESDT Version for the file(s) to insert?

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

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

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

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

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

Single Filename to Insert (including FULL path)?

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

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

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

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

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

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

Name of MFG file (enter to end list)?

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

Associated ASCII Metadata Filename to Insert?

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

Any success or error messages are then displayed followed by:

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

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

4.5.1.2.9.5 Insert EXE TAR

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

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

Preconditions to running the Insert Test Dynamic File program

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

- An ESDT for EXE TAR files must have been installed at the Data Server.
- The PGE must have been registered to PDPS via the SSI&T Science Metadata Update tool.
- An ASCII Metadata file (.met) for the PGEEXE 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/PGEEXE.met.template.
- The tar file must be prepared with the files needed to execute the PGE (see below for a list).
- Both the tar file in question and its corresponding MCF must reside on the local machine.
- The directory where the tar file and the ASCII Metadata File exist must be cross-mounted to the Data Server machines.

The Exe Tar file must include

⇒ PGE compiled binary executable(s),

⇒ PGE scripts (if any),

⇒ SCF-generated SDP Toolkit compiled runtime message files (if any),

⇒ Any dynamic link libraries needed by the PGE at runtime.

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

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

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

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

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

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

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

ECS Mode of operations?

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

Name of PGE?

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

Science Software version of PGE?

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

Staged filename to Insert (including FULL path)?

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

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

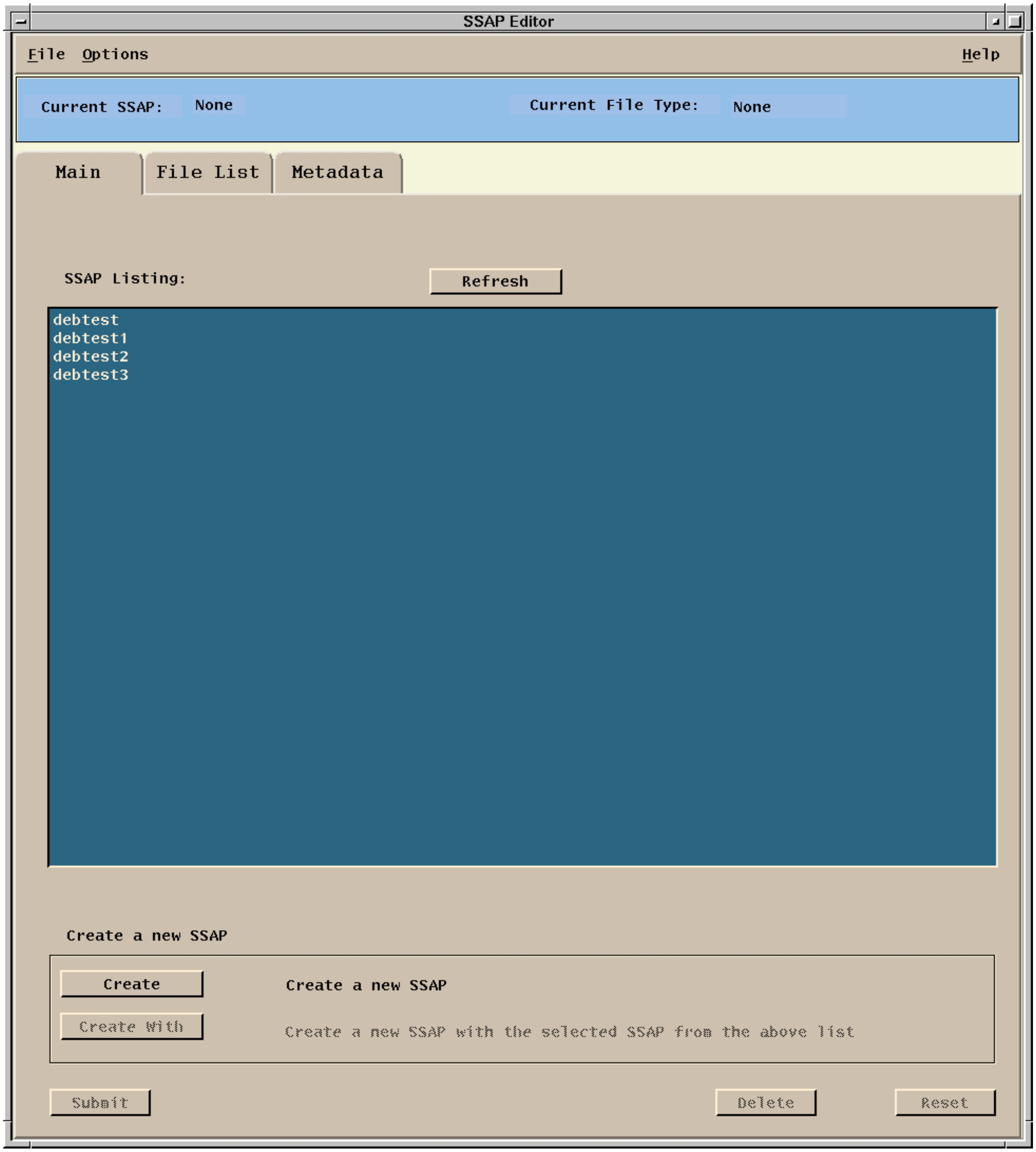
This is the name of the ASCII Metadata File (.met) associated with the tar file. The full path to the file must be included so the tool can find it. Use the /usr/ecs/{MODE}/CUSTOM/data/DPS/PGEEXE.met.template file to create one for the tar file to be inserted.

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

4.5.1.2.9.6 Science Software Archive Packages (SSAP) Editor

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

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



Current SSAP: None

Current File Type: None

SSAP Listing:

debtest
debtest1
debtest2
debtest3

Create a new SSAP

Create a new SSAP

Create a new SSAP with the selected SSAP from the above list

Figure 4.5.1-35. SSAP Editor Main Pop-up

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

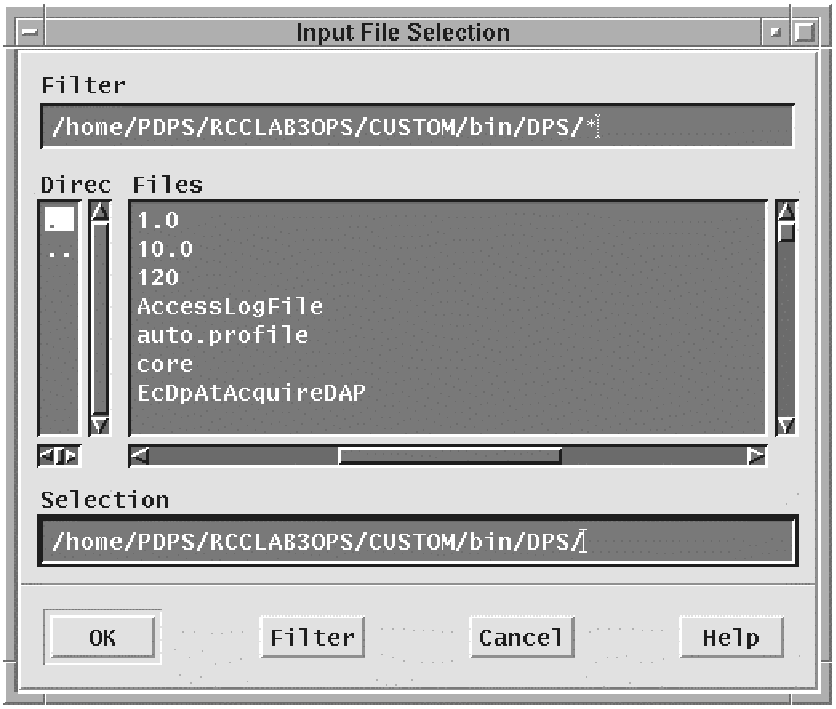
EcDpAtSSAPGui ConfigFile <CONFIG_FILE> ecs_mode <MODE>,

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

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

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

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



Filter

/home/PDPS/RCCLAB3OPS/CUSTOM/bin/DPS/*

```
1.0
10.0
120
AccessLogFile
auto.profile
core
EcDpAtAcquireDAP
```

Selection

/home/PDPS/RCCLAB3OPS/CUSTOM/bin/DPS/

Figure 4.5.1-36. Input File Selection Pop-up

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

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

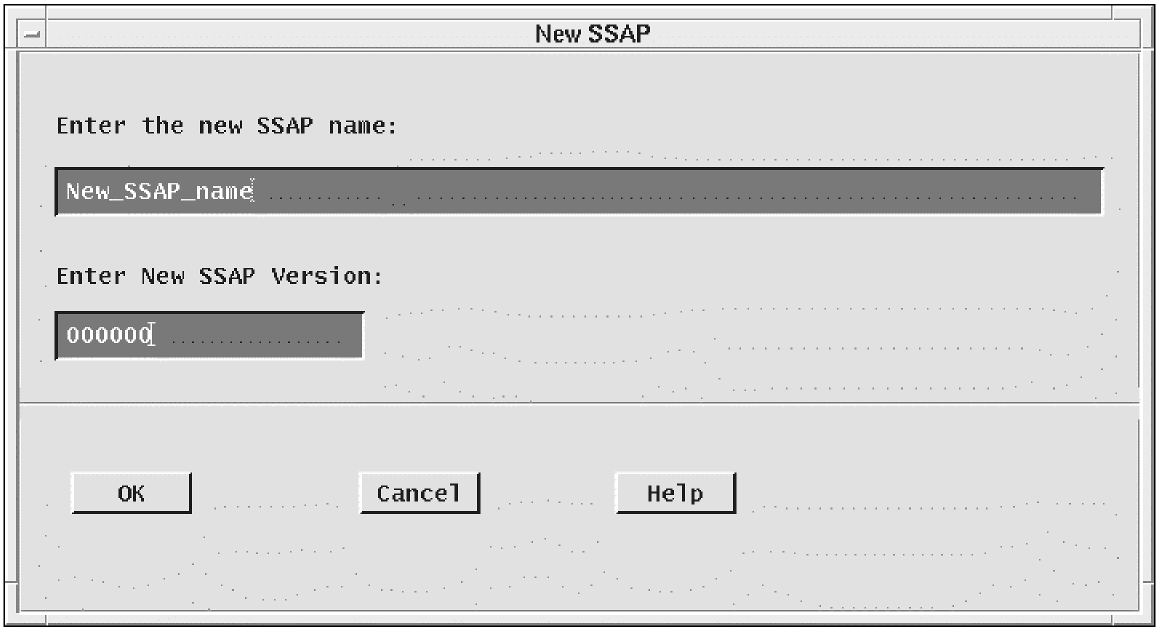
The SSAP Editor Main Pop-up screen has three tabs, i.e., **Main**, **File List** and **Metadata.**

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

4.5.1.2.9.6.1 SSAP Editor Main View

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

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


New SSAP

Enter the new SSAP name:

New_SSAP_name

Enter New SSAP Version:

000000

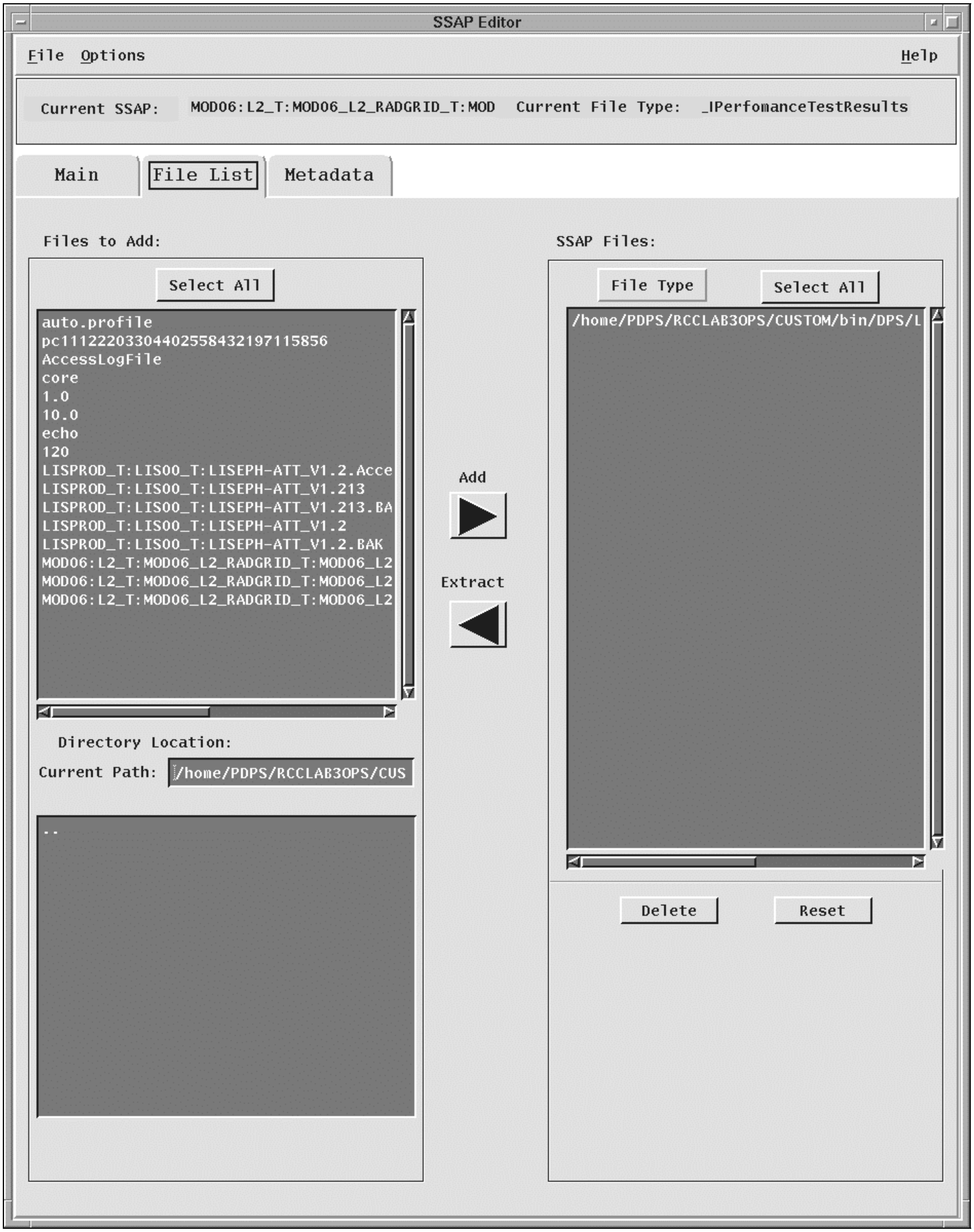
Figure 4.5.1-37. New SSAP Window

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

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

4.5.1.2.9.6.2 File List View

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



Current SSAP: MOD06:L2_T:MOD06_L2_RADGRID_T:MOD   Current File Type: _IPerfomanceTestResults

Files to Add:

```
auto.profile
pc11122203304402558432197115856
AccessLogFile
core
1.0
10.0
echo
120
LISPROD_T:LIS00_T:LISEPH-ATT_V1.2.Acce
LISPROD_T:LIS00_T:LISEPH-ATT_V1.213
LISPROD_T:LIS00_T:LISEPH-ATT_V1.213.BA
LISPROD_T:LIS00_T:LISEPH-ATT_V1.2
LISPROD_T:LIS00_T:LISEPH-ATT_V1.2.BAK
MOD06:L2_T:MOD06_L2_RADGRID_T:MOD06_L2
MOD06:L2_T:MOD06_L2_RADGRID_T:MOD06_L2
MOD06:L2_T:MOD06_L2_RADGRID_T:MOD06_L2
```

Directory Location:

Current Path: /home/PDPS/RCCLAB3OPS/CUS

```
..
```

SSAP Files:

```
/home/PDPS/RCCLAB3OPS/CUSTOM/bin/DPS/L
```

Figure 4.5.1-38. File List View Pop-up

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

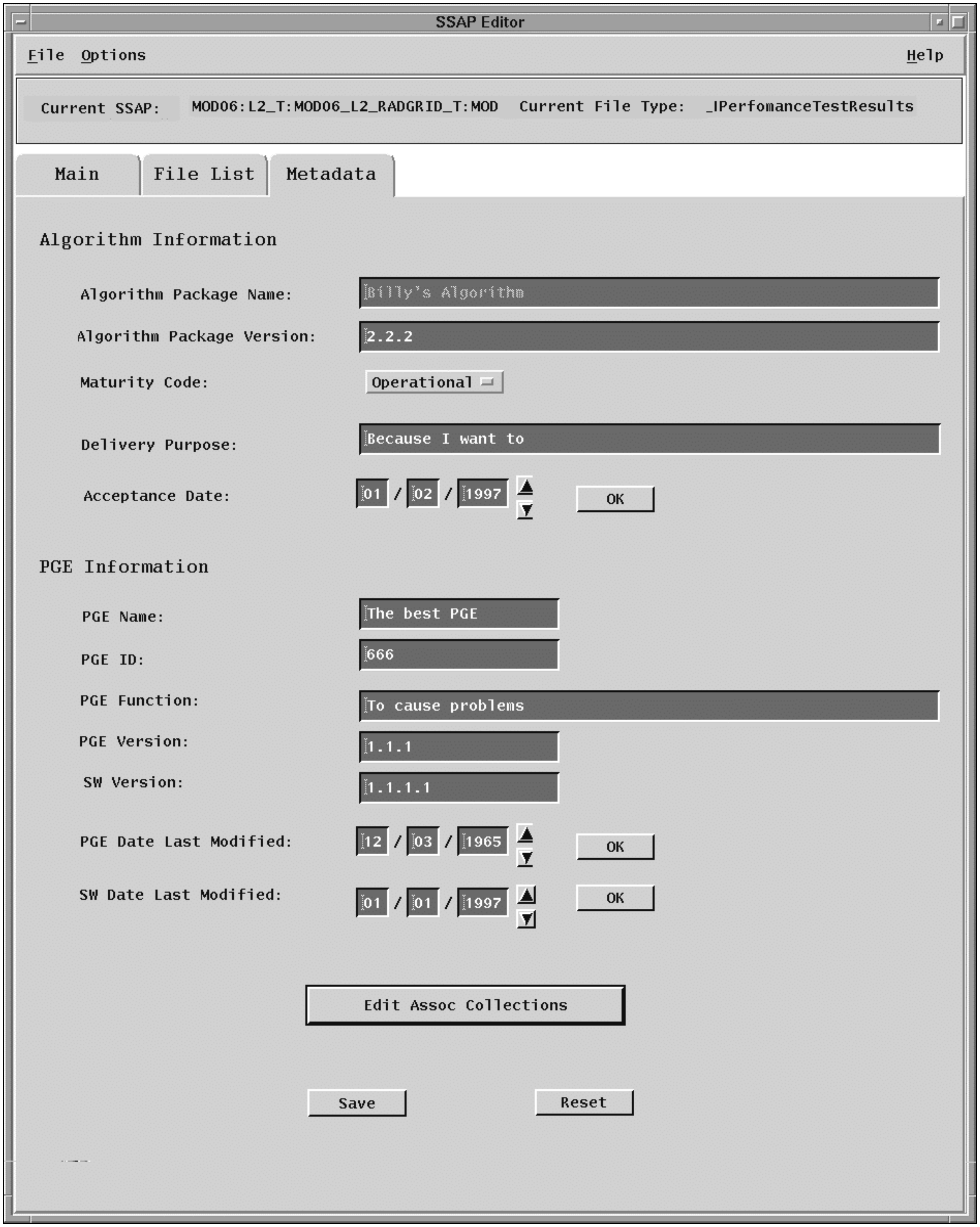
- By clicking on the **File Type** button, selected SSAP files can be further identified as one of the following: Algorithm Description, Change Log, Context Dialogue, Delivery List, PGE Information and Software. A **File Type** must be selected before file list changes can be made

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

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

4.5.1.2.9.6.3 Metadata View

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



Current SSAP: MOD06:L2_T:MOD06_L2_RADGRID_T:MOD Current File Type: _IPerfomanceTestResults

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

PGE Information

PGE Name: The best PGE

PGE ID: 666

PGE Function: To cause problems

PGE Version: 1.1.1

SW Version: 1.1.1.1

PGE Date Last Modified: 12 / 03 / 1965

SW Date Last Modified: 01 / 01 / 1997

Figure 4.5.1-39. Metadata View Pop-up

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

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

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

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

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

The **Edit Assoc Collection** button allows the operator to edit the ESDT's the original PGE is associated with. By clicking on this button, the Associated Collections pop-up appears as shown in Figure 4.5.1-40.

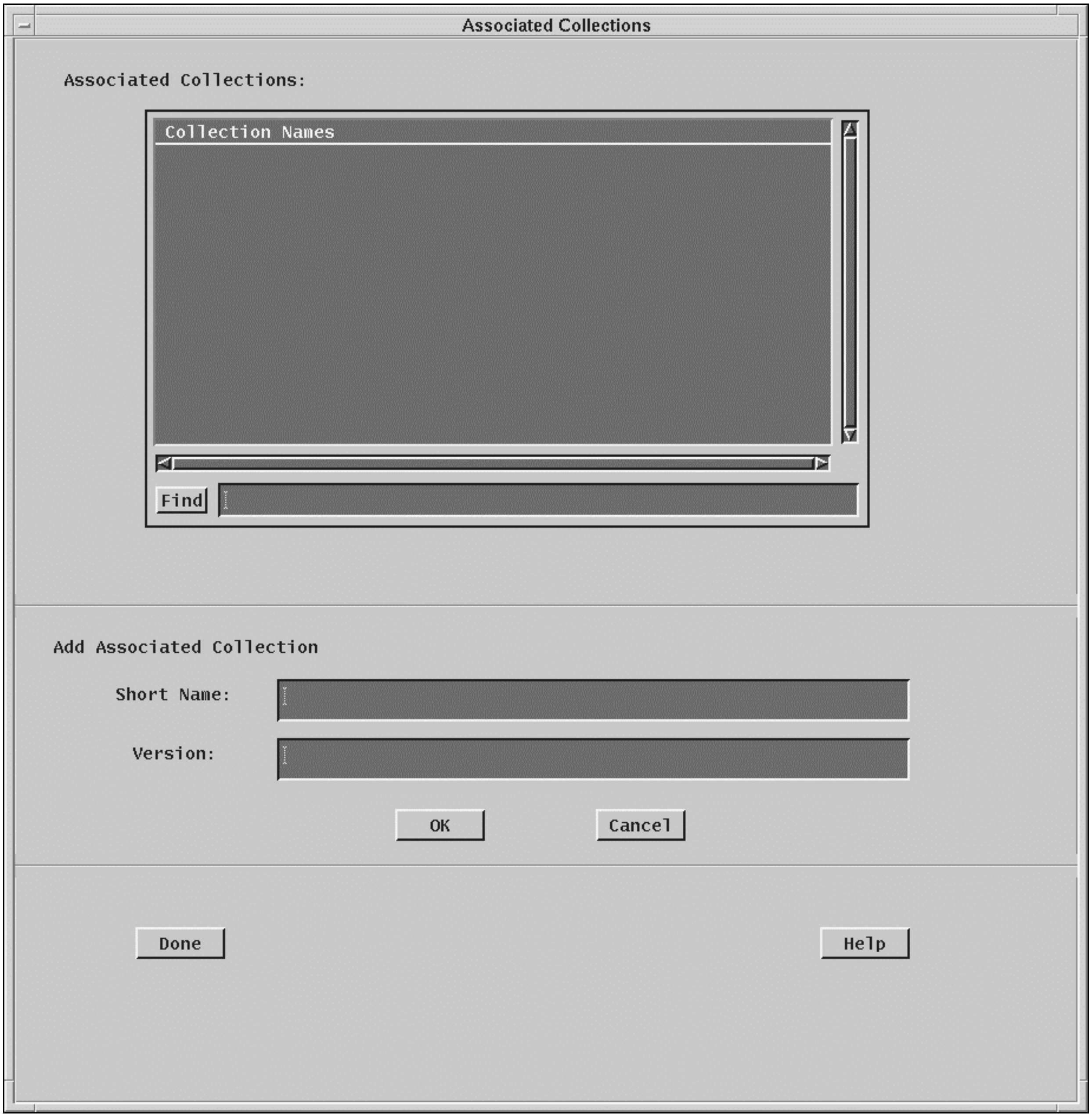


Figure 4.5.1-40. Associated Collections Pop-up

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

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

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

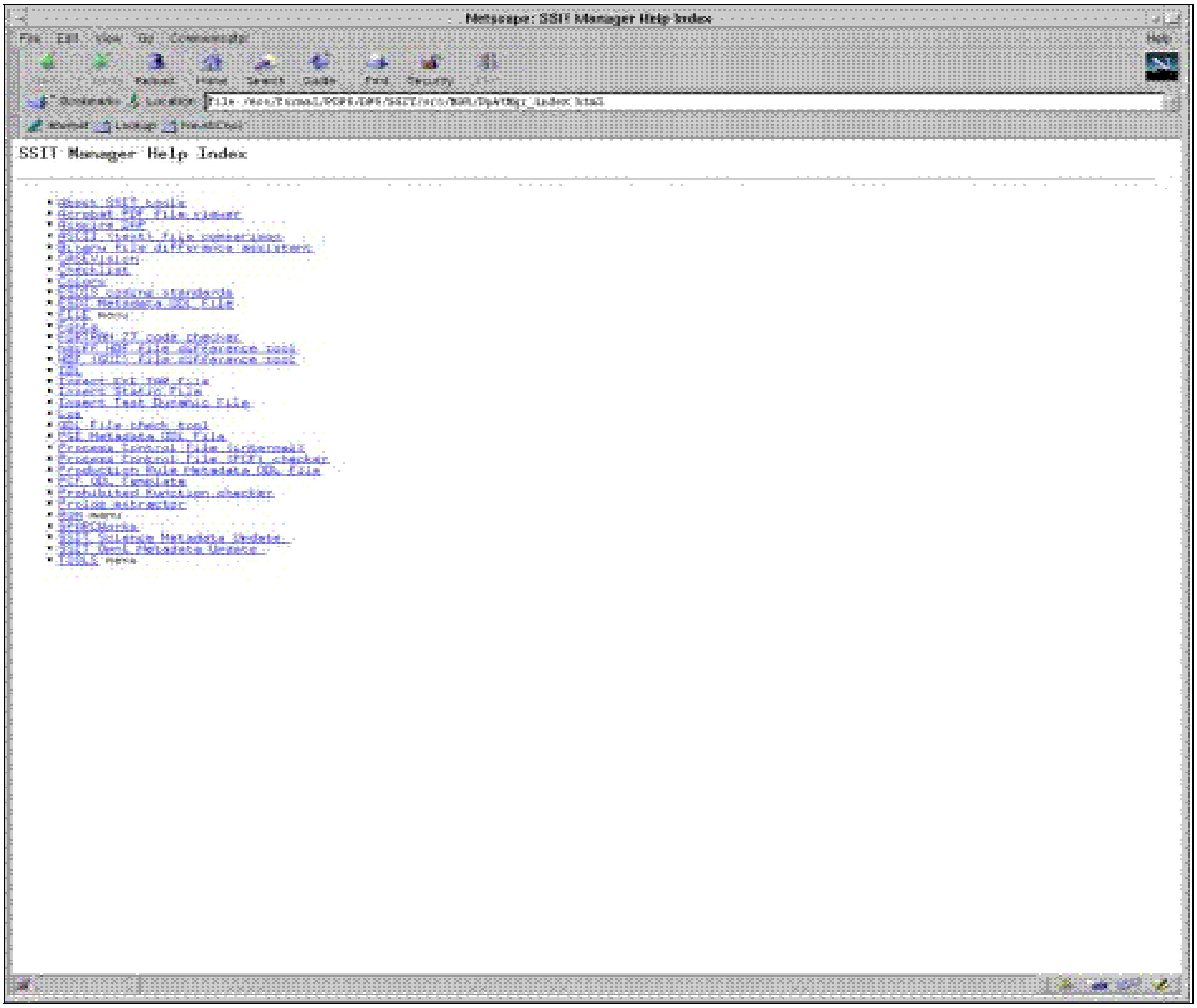
- Once the selection and/or the addition have been completed, the operator can confirm the changes by clicking on the **Ok** button. Further context sensitive help is available to the operator by clicking on the **Help** button

- Once **Done** is clicked, the operator is returned to the Metadata screen, shown in Figure 4.5.1-39. The **File** button at the upper left hand corner of the screen is clicked, to return to the SSI&T Manager menu, and finally to quit the operation

4.5.1.2.10 Help Menu

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

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



File: /ecs/formal/PDPS/DPS/SSIT/src/GUI/DpAtMgr_index.html

SSIT Manager Help Index

- About SSIT tools
- Acrobat PDF file viewer
- Acquire DAP
- ASCII (text) file comparison
- Binary file difference assistant
- CASEVision
- Checklist
- emacs
- ESDIS coding standards
- ESDT Metadata ODL File
- FILE menu
- Forte
- FORTRAN 77 code checker
- hdiff HDF file difference tool
- HDF (EOS) file difference tool
- IDL
- Insert Ext TAR File
- Insert Static File
- Insert Test Dynamic File
- Log
- ODL file check tool
- PGE Metadata ODL File
- Process Control File [illegible]
- Process Control File (PCF) checker
- Production Rule Metadata ODL File
- PCF ODL Template
- Prohibited Function checker
- Prolog extractor
- RUN menu
- SPARCWorks
- SSIT Science Metadata Update
- SSIT Oper. Metadata Update
- TOOLS menu

Figure 4.5.1-41. SSI&T 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 SSI&T, 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-TDA-022 "Custom Code Configuration Parameters" documentation series.

4.5.1.3.1 Interfaces and Data Types

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

Table 4.5.1-16. SSI&T Manager Interface Protocols

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

4.5.1.4 Databases/ Files

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

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

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

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

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

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

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

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

4.5.1.5 Special Constraints

None.

4.5.1.6 Outputs

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

4.5.1.7 Event and Error Messages

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

4.5.1.8 Reports

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

Table 4.5.1-18. Reports (1 of 2)

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

Table 4.5.1-18. Reports (2 of 2)

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

4.5.1.8.1 Sample Reports

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