5. Security Services

EMD security architecture must meet the requirements for data integrity, availability, and confidentiality. EMD Security Services meets these requirements by incorporating a variety of mechanisms to establish and verify user accounts, issue and verify passwords, audit user activity, and verify and protect data transfer. Security logs are monitored and security reports generated by the System Administrator as required. Several open source products provide tools for authentication and network and systems monitoring - Crack, ANLpasswd, TCP Wrappers, and Tripwire. Crack and ANLpasswd provide brute force password cracking and password checking, respectively for local system and network access. Tripwire monitors for intruders by noting changes to files. F-Secure Secure Shell (ssh) provides strong authentication access and session encryption from external, non-trusted networks as well as internally within a DAAC. Security Services also supports detection of, reporting, and recovery from security breaches. Security scans of each system are performed periodically to prepare for the formal security scans done biannually by the ESDIS IV&V contractor. These preliminary scans are done using the FOUNDSTONE Security Scanner product.

The following section defines step-by-step procedures for Operations personnel to run the Security Services tools. These procedures assume that DAAC Management has already approved the requester's application for a Security process. It is recommended that access to these tools be controlled through the **root access only**.

5.1 Scanning Network Vulnerabilities

The EMD project no longer has responsibility for scanning the network and network-attached systems. However, the FOUNDSTONE Security Scanner is a licensed product that NASA uses extensively to detect system level vulnerabilities. GSFC has a site license to use the product and any of the supported DAACs may use that license since all DAACs are using GSFC IP address space. This product does NOT belong to EMD and as such there is not an official release of it. A license key is required which can be obtained from the ESDIS Computer Security Official. The information the ESDIS Computer Security Official needs includes the IP addresses of the Production and M&O LANs. The software runs on Microsoft Windows Server 2003.. A laptop is the only practical way to run it. The software and the keys must be obtained through ESDIS CSO.

5.2 Ensuring Password Integrity

One aspect of system security is discretionary access control based on user passwords. Passwords ideally would be so unique that they are virtually impenetrable to unauthorized users. Two products provide utilities to create effective password practices. "Crack" detects weak passwords that could be easily bypassed. It works in "batch" mode. ANLpasswd enforces strong password rules as the user is changing their password.

Table 5.2-1 contains the activity checklist for Crack.

Table 5.2-1. Crack - Activity Checklist

Order	Role	Task	Section	Complete?
1	SA	Configure Crack	(P) 5.2.1.1	
2	SA	Launching Crack	(P) 5.2.1.2	
3	SA	Creating Dictionaries	(P) 5.2.1.3.1	

Crack and ANLpasswd provide a comprehensive dictionary, which can be shared. These "source" dictionaries provide lists of words, which if used, would create vulnerable passwords. You can add other dictionaries, for example, acronym lists, to eliminate commonly used terms from being used as passwords.

Crack is installed in a secure location that has **root access only**. ANLpasswd is automounted in /tools/bin.

5.2.1 Detecting Weak Passwords

Running Crack against a system's password file enables a system administrator to assess how vulnerable the file is to unauthorized users and how well authorized users select secure passwords. Crack is designed to find standard UNIX eight-character DES-encrypted passwords by standard guessing techniques.

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

The instructions provided in the following sections are general in nature, because how you configure Crack is DAAC specific. Operations personnel should be familiar with these tasks to:

- Configure the Crack shell script and config.h files based on the README file and on requirements established for your site. See the Section on "Configuring Crack" below.
- Run Crack based on requirements established for your site. See "Running Crack" below.
- Customize the dictionaries. See Section "Creating Dictionaries" below.

Although Crack should already be configured for your system, the instructions are provided should you have to reconstruct the makefile as a result of file corruption. Crack has two configuration files: the Crack shell script, which contains all the installation-specific configuration data, and the file Sources/conf.h, which contains configuration options specific to various binary platforms. Use the following procedure for configuring crack.

5.2.1.1 Configuring Crack

1 In the Crack shell script, edit the CRACK_HOME variable to the correct value. This variable should be set to an absolute path name on which Crack will be run. (Path names relative to username are acceptable as long as you are using csh.)

- There is a similar variable, CRACK_OUT, which specifies where Crack should put its output files — by default, this is the same as $CRACK_HOME.

2 Edit the file Sources/conf.h and establish which switches to enable. Each #define has a small note explaining its purpose. Portability of certain library functions should not be a problem.

3 If using Crack-network (see Section 5.2.1.4, Options, below), generate a Scripts/network.conf file. This file contains:

- A list of hostnames that are rsh/ssh destinations.
- Their binary type (useful when running a network Crack on several different architectures).
- An estimate of their relative power (take your slowest machine as unary, and measure all others relative to it).
- A list of per-host flags to add to those specified on the Crack command line, when calling that host.
- There is an example of such a file provided in the Scripts directory.

4 To specify a more precise figure as to the relative power of your machines, play with the command *make* tests in the source code directory. This can provide you with the number of fcrypt()s that your machine can do per second. This number can be plugged into your *network.conf* as a measure of your machines' power (after rounding the value to an integer).

Crack is a self-installing program. Once the necessary configuration options for the Crack shell script and config.h have been set, the executables are created via *make* by running the Crack shell script.

NOTE: To run Crack on a YP password file, the simplest way is to generate a passwd format file by running the following command:

 # ypcat passwd > passwd.yp ↵

and then running Crack on the passwd.yp file.

5.2.1.2 Launching Crack

To launch Crack:

---

1 To change directory, type **cd /usr/local/solaris /crack**, and then press the **Return/Enter** key.

2 To execute the program, type **./Crack**, and then press the **Return/Enter** key.

3 For the single platform version, type **./Crack [options] [bindir] /etc/passwd [...other passwd files]** and then press the **Return/Enter** key.

4 To execute over the network, type **./Crack -network [options] /etc/passwd [...other passwd files]** and then press the **Return/Enter** key.

For a brief overview of the [options] available, see Section 5.2.1.4, Options, below. Section 5.2.1.5, Crack Support Scripts, briefly describes several very useful scripts.

Crack works by performing several individual passes over the password entries that are supplied. 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, located in the Scripts directory (see Section 5.2.1.5, Crack Support Scripts, below).

Rules in Scripts/gecos.rules are applied to data generated by Crack from the pw_gecos and pw_gecos entries of the user's password entry. The entire set of rules in gecos.rules is applied to each of these words, which creates many more permutations and combinations, all of which are tested. After a pass has been made over the data based on gecos information, Crack makes further passes over the password data using successive rules from the Scripts/dicts.rules by loading the whole of Dicts/bigdict file into memory, with the rule being applied to each word from that file. This generates a resident dictionary, which is sorted and made unique to prevent wasting time on repetition. After each pass is completed, the memory used by the resident dictionary is freed up, and re-used when the next dictionary is loaded.

Crack creates the Dicts/bigdict dictionary by merging, sorting, and making unique the source dictionaries, which are to be found in the directory DictSrc and which may also be named in the Crack shell script, via the $STDDICT variable. (The default value of $STDDICT is /usr/dict/words.)

The file DictSrc/bad_pws.dat is a dictionary that is meant to provide many of those common but non-dictionary passwords, such as 12345678 or qwerty.

5.2.1.3 Creating Dictionaries

---

1 Copy your dictionary into the DictSrc directory (use compress on it if you wish to save space; Crack will unpack it while generating the big dictionary).

2 Delete the contents of the Dicts directory by running Scripts/spotless. Your new dictionary will be merged in on the next run.

5.2.1.4 Options

Options available with the Crack command are:

-f Runs Crack in foreground mode, i.e., the password cracker is not put into the background, and messages appear on stdout and stderr as you would expect. This option is only really useful for very small password files, or when you want to put a wrapper script around Crack.

Foreground mode is disabled if you try running Crack-network -f on the command line, because of the insensibility of rsh'ing to several machines in turn, waiting for each one to finish before calling the next. For more information, read the section about Network Cracking without NFS/RFS in the README.NETWORK file.

-v Sets verbose mode, whereby Crack will print every guess it is trying on a per-user basis. This is a very quick way of flooding your filestore, but useful if you think something is going wrong.

-m Sends mail to any user whose password you crack by invoking Scripts/nastygram with their username as an argument. The reason for using the script is so that a degree of flexibility in the format of the mail message is supplied; i.e., you don't have to recompile code in order to change the message.

-nvalue Sets the process to be nice()ed to value, so, for example, the switch -n19 sets the Crack process to run at the lowest priority.

-network Throws Crack into network mode, in which it reads the Scripts/network.conf file, splits its input into chunks that are sized according to the power of the target machine, and calls rsh to run Crack on that machine. Options for Crack running on the target machine may be supplied on the command line (for example, verbose or recover mode), or in the network.conf file if they pertain to specific hosts (e.g., nice() values).

-r<pointfile>

This is only for use when running in recover mode. When a running Crack instance starts pass 2, it periodically saves its state in a point file, with a name of the form Runtime/P.* This file can be used to recover where you were should a host crash. Simply invoke Crack in exactly the same manner as the last time, with the addition of the **-r** switch (for example, **-rRuntime/Pfred12345**). Crack will startup and read the file, and jump to roughly where it left off. If you are cracking a very large password file, this can save a lot of time after a crash.

5.2.1.5 Crack Support Scripts

The Scripts directory contains a small number of support and utility scripts, some of which are designed to help Crack users check their progress. The most useful scripts are briefly described below.

Scripts/shadmrg

This is a small script for merging /etc/passwd and /etc/shadow on System V style shadow password systems. It produces the merged data to stdout, and will need to be redirected into a file before Crack can work on it.

Scripts/plaster

This is a simple front-end to the Runtime/D* diefiles that each copy of the password cracker generates. Invoking Scripts/plaster will kill off all copies of the password cracker you are running, over the network or otherwise. Diefiles contain debugging information about the job, and are generated so that all the jobs on the entire network can be called quickly by invoking Scripts/plaster. Diefiles delete themselves after they have been run.

Scripts/status

This script rsh's to each machine mentioned in the Scripts/network.conf file, and provides some information about processes and uptime on that machine. This is useful when you want to find out just how well your password crackers are getting on during a Crack -network.

Scripts/{clean,spotless}

These are just front ends to a makefile. Invoking Scripts/clean cleans up the Crack home directory and removes unwanted files, but leaves the pre-processed dictionary bigdict intact. Scripts/spotless does the same as Scripts/clean, but obliterates bigdict and old output files, too, and compresses the feedback files into one.

Scripts/nastygram

This is the shell script that is invoked by the password cracker to send mail to users who have guessable passwords, if the **-m** option is used. Edit it to suit your system.

Scripts/guess2fbk

This script takes your out* files as arguments and reformats the 'Guessed' lines into a feedback file, suitable for storing with the others.

An occasion where this might be useful is when your cracker has guessed a large number of passwords and then died for some reason (a crash?), before writing out the guesses to a feedback file. Running **Scripts/guess2fbk out* >> Runtime/F.new** will save the work that has been done.

5.2.1.6 Checking the Log

Crack loads dictionaries directly into memory, sorts and makes them unique, before attempting to use each of the words as a guess for each users' password. If Crack correctly guesses a

password, it marks the user as done and does not waste further time on trying to break that user's password.

Once Crack has finished a dictionary pass, it sweeps the list of users looking for the passwords it has cracked. It stores the cracked passwords in both plain text and encrypted forms in a feedback file in the directory **Runtime**. Feedback files have names of the form **Runtime/F***. This allows Crack to recognize passwords that it has successfully cracked previously, and filter them from the input to the password cracker. This provides an instant list of “crackable” users who have not changed their passwords since the last time Crack was run. This list appears in a file with name **out*** in the **$CRACK_OUT** directory, or on **stdout**, if foreground mode (**-f**) is invoked (see Section “Options”, above).

Similarly, when a Crack run terminates normally, it writes out to the feedback file all encrypted passwords that it has NOT succeeded in cracking. Crack will then ignore all of these passwords next time you run it.

Obviously, this is not desirable if you frequently change your dictionaries or rules, so, **Scripts/mrgfbk** is provided to allow for checking the “uncrackable” passwords. This script sorts your feedback files, merges them into one, and optionally removes all traces of "uncrackable" passwords, so that your next Crack run can have a go at passwords it has not succeeded in breaking before.

mrgfbk is invoked automatically if you run **Scripts/spotless** (see Section 5.2.1.5, Crack Support Scripts, above).

5.2.2 ANLpasswd

The Argonne National Laboratory wrote ANLpasswd and has made it available to everyone as freeware. There is a simple install script that will install the components on the automount host for both SGI and Sun architectures. ANLpasswd consists of a setuid C program that is used to call the anlpasswd Perl script. The Perl script uses the Crypt::Cracklib module, which is installed with the package, a dictionary generation tool, and dictionaries that are used to match attempted passwords against possible passwords that are in the dictionary file.

It is assumed that Perl 5.6 is properly installed in /tools/perl for Sun and SGI platforms. The binary ypstuff and the anlpasswd30 script (with its soft links to anlpasswd and yppasswd) are placed in /tools/bin. The Perl includes and dictionary file should also be NFS mounted and placed in /tools/lib/anlpasswd.

Once the package is configured, the only alteration may be in the dictionary files. There are a large number of dictionary files that are included by default in this release. If there are local requirements to change them (i.e. the default has too little security or too much security), the following procedure is applicable.

Table 5.2-2 contains the activity checklist for ANLpasswd.

Table 5.2-2. ANLpasswd - Activity Checklist

Order	Role	Task	Section	Complete?
1	SA	Configure ANLpasswd	(P) 5.2.2.1	

5.2.2.1 Configuring ANLpassword

1 Login to the automount host as root or **su** to root.

2 Modify the SGI /tools/lib/words directory as required (add files, modify files or remove files).

3 Remove the Sun /tools/lib/words directory contents, then copy the SGI (modified) directory to the Sun directory.

4 Login to an SGI as root or **su** to root.

5 From the SGI window, type **cd /tmp** and then press the **Return/Enter** key to change to the directory where anlpasswd-30.tar.gz is located.
- The directory is changed to **/tmp**.

6 To explode anlpasswd-30.tar.gz, type **gzip –dc anlpasswd-30.tar.gz | tar –xovf –** and ten press the **Return/Enter** key.
- The anlpasswd-30.tar.gz file is exploded.

7 From the SGI window, to change directory to the location of the make dictionary script, type **cd /tmp/anlpasswd/anlpasswd-3.0-sgi/cracklib25_small**, and then press the **Return/Enter** key.
- The directory is changed to /tmp/anlpasswd/anlpasswd-3.0-sgi/cracklib25_small.

8 To run the make dictionary script, type **./makedictionary.pl** and then press the **Return/Enter** key.
- ***Note*:** perl expected to be in /tools/perl
- The script runs.

9 From the SGI window, on completion, copy the pw_dict.* files to the automount host's /tmp directory.

10 From the automount host window, copy the /tmp/pw_dict.* files to the appropriate /tools/lib directories for ***both*** SGI and Sun architectures.

11 Logout from the automount host.

12 From the SGI window, **su** to a normal user account and check that the changes work by running /tools/bin/anlpasswd as a normal user and verify at least one of the changes and that the script still works normally (without errors).

13 From the SGI window, to delete the temp files, type **rm –rf /tmp/anlpasswd**, and then press the **Return/Enter** key.

- The temp files are deleted.

14 Logout from the SGI.

5.2.2.2 Installing ANLpasswd

Use the procedures provided in the Release Notes for the relevant version of ANLpasswd. Release Notes are available through the "Release Notes" link at the following URL:

http://cmdm-ldo.raytheon.com/baseline/

5.2.2.3 ANLpasswd readme

The following is the README.INSTALL from the tar file with comments. This work has already been incorporated in the release. It is provided here to facilitate understanding of how the product is put together.

```
ANLpasswd is used in ECS to provide interactive password checking. It is
installed on the network in the /tools/bin directory. Local installation
is not required.

PREREQUISITES
This installation requires:
Perl 5.6.1
50Mb of disk space
It will take approximately 30 minutes to complete this installation.

INSTALLATION INSTRUCTIONS
1.  Copy the anlpasswd-30.tar.gz file to a staging area. For
convenience, /tmp is used in these instructions.

2.  Login to the automount host as root or su to root.

3.  Change directory to /tmp and explode the tarball using the commands:
# cd /tmp ↵
# gzip -dc anlpasswd30.tar.gz | tar -xovf - ↵

4. Change directory to the top level directory and run the install script using the command:
# cd /tmp/anlpasswd ↵
# ./install_anlpasswd.pl ↵

This will install the /tools/bin/anlpasswd30 script with links to
/tools/bin/anlpasswd and /tools/bin/yppasswd, the dictionaries
themselves, and the dictionary indexes.
```

5. If you DO have a password aging method in place, skip to step 8. If you do not have a password aging method in place and are implementing the password aging script, copy the /tmp/anlpasswd/password_aging_notify.pl script to the NIS master server. To implement this script, the following information needs to be edited in the passwordage.pl script:

```
# Master NIS server

$master_host = "<NISMASTER>";

###

# Domain (used when building address to send users email)
$domain = "<DAACDOMAIN>";
#

# Location of the Shadow file
$shadow_file = "<SHADOWFILELOCATION>";

# The protected accounts - these accounts are immune to password aging


@protected_accounts = ('root');
###


# Location of the directory to backup copies of the shadow file in
$shadow_archive = "<SHADOWFILEARCHIVELOCATION>";
###

# Variables used when sendmail emails messages to users and SAs
# This to address is used when sending emails to the SAs
$to_address = "<SAADDRESSLIST>";
```

where:
<NISMASTER> is the fully qualified host name for the NIS master
<DAACDOMAIN> is the NIS domain name of the DAAC
<SHADOWFILELOCATION> is the location of the shadow file (normally /etc/shadow)
<SHADOWFILEARCHIVELOCATION> is the directory of the shadow file archive backups
< SAADDRESSLIST> is the email account(s) to send messages to SAs

6. Setup cron to run the script at a convenient time.

7. Logoff from the automount host.

8. Checkout the SGI installation from an SGI production host by logging in as a normal user.

9. Change your password using the command:

   % **/tools/bin/yppasswd ↵**

10. Logout

11. Wait a few minutes to make sure that the updates are completed.

```
12.  Checkout the Solaris installation from a Sun production host by
logging in as a normal user.

13.  Change your password using the command:
      % /tools/bin/yppasswd ↵

14.  Logout

15.  Wait a few minutes to make sure that the updates are completed.

That should be all that is needed to get this program up and running. If
there are any problems or inaccuracies in this documentation, or you
have any improvements or bug fixes, please send email to
"support@mcs.anl.gov"
```

5.3 Aging Passwords

Password aging is required by NPR 2810.1, *NASA Procedural Requirements: Security of Information Technology*. A perl script is provided as part of the ANLpasswd 3.0 release that will, after configuration, perform 120-day password aging. If your site already has a method of doing password aging, this section may be ignored. If your site does NOT have a password aging method in place and you are implementing the password aging script, copy the /tmp/anlpasswd/password_aging_notify.pl script to the NIS master server.

To implement this script, the following information needs to be edited in the password_aging_notify.pl script:

```
# Master NIS server
$master_host = "<NISMASTER>";
###

# Domain (used when building address to send users email)
$domain = "<DAACDOMAIN>";
#
# The protected accounts - these accounts are immune to password aging
@protected_accounts = ('root');

# Location of the directory to backup copies of the shadow file in
$shadow_archive = "<SHADOWFILEARCHIVELOCATION>";
###

# Variables used when sendmail emails messages to users and SAs
# This to address is used when sending emails to the SAs
$to_address = "<SAADDRESSLIST>";
```

where:
<NISMASTER> is the fully qualified host name for the NIS master
<DAACDOMAIN> is the NIS domain name of the DAAC
<SHADOWFILELOCATION> is the location of the shadow file (normally /etc/shadow)
<SHADOWFILEARCHIVELOCATION> is the directory of the shadow file archive backups
< SAADDRESSLIST> is the email account(s) to send messages to sys administrators

5.4 Secure Access through Secure Shell

The security risks involved in using "R" commands such as rlogin, rsh, rexec and rcp are well known, but their ease of use has made their use tempting in all but the most secure of environments. Ssh is an easy-to-use, drop in replacement for these commands developed by Tatu Ylonen. Ssh is a "user" level application. No changes to the host kernel are required. The UNIX server implements the commercial version of F-Secure. As of the F-Secure 3.2 release, only SSH Version 2 is included in pre-compiled, OS-specific packages.

As of the Secure Shell 2.0 release in May, 2000 and later, all of the files needed to function are loaded locally on each UNIX host in /usr/local/bin.

- ssh - replaces rsh, rlogin and rexec for interactive sessions
- scp - replaces rcp for interactive file transfer
- ssh-agent – application that allows a user to enter the passphrase once, then when other applications (e.g. ssh, scp) are used, one is not prompted for the passphrase – it is automatically negotiated.
- ssh-add - add access to a specific ssh host
- ssh-keygen - generates keys for the local host based on a passphrase (long password)
- ssh-signer – verifies that a key is genuine so that public key authentication may proceed
- sftp - secure ftp

The host daemon is in /usr/local/sbin which includes:

- sshd2 - the ssh version 2 daemon

Several files are generated on installation and when running and are installed locally:

- /etc/ssh2/ssh2_config - system-wide configuration for the ssh2 client
- /etc/ssh2/hostkey - contains the long number used for one of the ssh2 keys
- /etc/ssh2/hostkey.pub - contains the ssh2 key known to the public
- /etc/ssh2/random_seed - base number used in generating keys
- /etc/ssh2/sshd2_config - defines the local ssh2 security policy
- /etc/sshd2_22.pid - the process id of the ssh2 daemon currently running

The amount of disk space that the programs and the configurations require is less than 25 MB.

Table 5.4-1 contains the activity checklist for Services Access through Secure Shell.

Table 5.4-1. Secure Access through Secure Shell - Activity Checklist

Order	Role	Task	Section	Complete?
1	SA	Using Secure Shell	(P) 5.4.3.1	

5.4.1 Installation of SSH

Use the procedures provided in the Release Notes for the relevant version of ssh. Release Notes are available through the "Release Notes" link at the following URL:

http://cmdm-ldo.raytheon.com/baseline/

5.4.2 The SSH Encryption Mechanism[1]

Each host has a host-specific DSA key (normally 1024 bits) used to identify the host. Additionally, when the daemon starts, it generates a server DSA session key (normally 768 bits). This key is normally regenerated every hour if it has been used, and is never stored on disk.

Whenever a client connects the daemon, the daemon sends its host and server public keys to the client. The client compares the host key against its own database to verify that it has not changed. The client then generates a 256 bit random number. It encrypts this random number using both the host key and the server key, and sends the encrypted number to the server. Both sides then start to use this random number as a session key that is used to encrypt all further communications in the session. The rest of the session is encrypted using a conventional cipher. Under EMD the aes128 cipher is used. The client selects the encryption algorithm to use from those offered by the server.

Next, the server and the client enter an authentication dialog. The client tries to authenticate itself using .rhosts authentication, .rhosts authentication combined with DSA host authentication, RSA challenge-response authentication, or password-based authentication. (NOTE: In the EMD configuration, .rhosts is NOT available).

Rhosts authentication is disabled within the DAACs because it is fundamentally insecure.

If the client successfully authenticates itself, a dialog for preparing the session is entered. At this time the client may request things like allocating a pseudo-tty, forwarding X11 connections, forwarding TCP/IP connections, or forwarding the authentication agent connection over the secure channel.

5.4.3 Using Secure Shell

5.4.3.1 Using Secure Shell

1 To login, use the command:

% slogin defiant ↵

Enter the passphrase for the key (lotsofstuffhere): **br0wn cow 3ats grass ↵**
Last login: Sun Feb 22 06:50:59 1998 from echuser.east.hitc.com
No mail.
%

NOTE: The first time you login to a host the following message will pop up asking if you want to continue. In response, type **yes** and [**enter**]:

[1] From the *sshd* man page

Host key not found from the list of known hosts.
Are you sure you want to continue connecting (yes/no)? **yes** ↵
Host 't1acg01' added to the list of known hosts.

2 To transfer a file, use the command:

% scp hostone:/etc/info info ↵
Enter the passphrase for the key (lotsofstuffhere): **br0wn cow 3ats grass** ↵

- This will copy the file /etc/info from hostone to your local host. Note that your passphrase is needed to initiate the transfer.

IMPORTANT NOTE: The default directory on the *target* host is always the users HOME directory.

3 Also, one may send/receive files recursively using "-r" such as:

% scp -r ~/files/* hostone:~/files ↵

will send what is in the home directory files subdirectory to the target host hostone in the home files subdirectory.

4 To execute a command remotely, use the command:

% ssh whoisonfirst ps –ef ↵
Enter the passphrase for the key (lotsofstuffhere): **br0wn cow 3ats grass** ↵

5.4.4 A Layer of Convenience

If you are already a user of "r" commands, you probably know about the .rhost file. Ssh will allow a user to setup the .rhost equivalent called .shost in one's home directory. .Rhost and .shost contain the names of the hosts to which one normally connects. The nice thing about using it is one need not enter one's passphrase. Unlike "r" commands, however, ssh commands use long strings of numbers to authenticate the client, which makes it quite difficult for an intruder to impersonate a legitimate user. One word of caution, however, if you leave your terminal while logged on, a passerby could logon to any host in your .rhost/.shosts file and potentially cause malicious damage to you and your colleagues work. Be aware!

NOTE: ssh checks the mode of .shost, so change permission on .shost by typing:

% chmod 600 /home/JohnDoe/.shost ↵

where you must substitute your own home directory for /home/JohnDoe.

5.4.5 Multiple Connections

If you open multiple connections, it is more convenient to keep your keys in system memory. To do this requires executing two commands:

> **% ssa ↵**
> Enter the passphrase for the key (lotsofstuffhere):
> Enter passphrase: **br0wn cow 3ats grass ↵**
> Identity added: /home/JohnDoe/.ssh/identity (bpeters@nevermor)
> %

Now, one may make connections (slogin, scp, ssh) to hosts that are running ssh without being prompted for a passphrase.

5.4.6 Secure FTP

A secure version of ftp is available. Use the command:

> **% sftp user@remotehost ↵**
> Enter the passphrase for the key (lotsofstuffhere): **MY PASSPHRASE ↵**
> local directory - /home/user
> remote directory - /home/user
> **sftp> get thisisafilename ↵**
> **sftp> put thisotherfilename ↵**
> **sftp> quit ↵**

5.4.7 Other Notes

IMPORTANT: SSH will automatically "tunnel" X sessions without user involvement even through multiple hops. However, it is important that you do NOT change the DISPLAY parameter or X will not use the ssh tunnel!

5.4.8 Configuration of Secure Shell

5.4.8.1 Local Setup

Most users will start from the same host whether from an X terminal, a UNIX workstation, or a PC. Running the sss (sshsetup) script generates long strings called keys that make ssh work. One set of keys is needed for each home directory.

The only thing you need to know before executing the script is to pick a good passphrase of at least 10 characters. You can and should use spaces and multiple words with numbers, misspellings and special characters. Note that passwords are NOT echoed back to the screen.

PLEASE DO NOT USE THE PASSWORDS/PASSPHRASES USED HERE OR IN ANY OTHER DOCUMENTATION!

Using the script sss should look like:

% sss ↵
Use a passphrase of at least 10 characters; which should include numbers
or special characters and MAY include spaces
New passphrase: **This is a silly test ↵**
Retype new passphrase: **This is a silly test ↵**
Generating ssh1 keys. Please wait while the program completes...
Generating ssh2 keys. This can take up to 240 seconds...
Done with sshsetup!
%

You are on the way!

NOTE: If you have accounts in the PVC, VATC and/or the EDF, at a DAAC production LAN or DAAC M&O LAN, do sss in EACH environment.

5.4.8.2 Remote Setup

If you need to access a host with a different home directory, you will need to run the ssr (ssh remote) script. NOTE: It is helpful to have run Secure Shell Setup (sss) in each environment first before doing the ssh remote script. This script sets up the destination host with the new set of keys and transfers the source (local) key to the destination and the destination key to the source. A new capability is to use different user names on the source and target hosts. This should look something like:

% **ssr ↵**
Remote user name (default: yourusername): ↵
Do you want to setup for:
1 VATC
2 PVC
3 GSFC DAAC
4 SMC
5 GSFC M and O
6 EDC DAAC
7 EDC M and O
8 LaRC DAAC
9 LaRC M and O
10 NSIDC DAAC
11 NSIDC M and O
x Exit from script
Select:
2
Working...
Accepting host p0spg07.pvc.ecs.nasa.gov key without checking.
yourusername@p0spg07.pvc.ecs.nasa.gov's password:
Authentication complete. Continuing with sshremote...

Downloaded remote keys.
Uploaded local keys.Keys concatenated.

Enter next site (press the enter-key and then x enter-key to exit)
Remote user name (default: yourusername): ↵

Do you want to setup for:
1 VATC
2 PVC
3 GSFC DAAC
4 SMC
5 GSFC M and O
6 EDC DAAC
7 EDC M and O
8 LaRC DAAC
9 LaRC M and O
10 NSIDC DAAC
11 NSIDC M and O

x Exit from script
Select:
x <enter>
bye!
%

5.4.8.3 Changing your Passphrase

To change your passphrase, use the following command:

% ssp ↵

Enter old passphrase: little 1amp jumb3d <enter>
Enter a new passphrase of at least 10 characters which should include numbers or special characters and MAY include spaces

New passphrase: **br0wn cows 3at grass ↵**
Retype new passphrase: **br0wn cows 3at grass ↵**

ssh2 key changed successfully.
Done with sshpass2!

5.4.9 Administration of Secure Shell

There is no administration of secure shell required except for general monitoring to make sure that the daemon process (/usr/local/sbin/sshd2) is running. Note, however, that the standard installation will establish a /var/log/ssh log file. It is recommended to review the /var/log/ssh and the system log file at least once a week.

5.5 Controlling Requests for Network Services (TCP Wrappers)

With TCP Wrappers, you can monitor and filter incoming requests for network services, such as FTP.

TCP Wrapper provides a small wrapper program for inet daemons that can be installed without any changes to existing software or to existing configuration files. The wrappers report the name of the client host and the name of the requested service; the wrappers do not exchange information with the client or server applications, and impose no overhead on the actual conversation between the client and server applications. The usual approach is to run one single daemon process that waits for all kinds of incoming network connections. Whenever a connection is established, this daemon runs the appropriate server program and goes back to sleep, waiting for other connections.

Operations personnel will monitor requests for these network services:

Client	Server	Application
ftp	Ftpd	file transfer
finger	Fingerd	show users

The **/var/log/wrappers log** file should be reviewed at least once a week. The log file provides information concerning who tried to access the network service. TCP Wrapper blocks any request made by unauthorized users. TCP Wrapper can be configured to send a message to any administrator whose request is rejected.

NOTE: All DAACs, except for NSIDC, do not use TCP Wrappers.

5.5.1 Installation, Configuration, and Testing for Wrappers

The installation of TCP Wrappers is part of the Secure Shell 2.0 and later packages. As of F-Secure SSH 3.2, it is a separate package and should be installed as part of the ssh installation. (Refer to the Release Notes as described in Section 5.4.1.) The location of most of the wrappers files have been changed to /etc/wrappers. Libwrap.a is in /usr/local/lib and tcpd.h is in /usr/local/include. The installation is automatic if wrappers has been previously installed. After installation, there are common operator functions, and the following checks should be made:

Table 5.5.1-1. Operator Functions

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.

5.5.2 Quick Start Using Tripwire

The following command is used to execute Tripwire from the command line prompt (as root):

/etc/tripwire/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.

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.

5.6 Monitoring File and Directory Integrity (Tripwire)

Tripwire is a tool that aids in the detection of unauthorized modification of files resident on UNIX systems. One important application of Tripwire is its use as the first and most fundamental layer of intrusion detection for an organization. Tripwire is automatically invoked at system startup. This utility will check the file and directory integrity by comparing a designated set of files and directories against information stored in a previously generated database. Tripwire flags and logs any differences, including added or deleted entries. When run against system files regularly, Tripwire spots any changes in critical system files, records these changes into its database, and notifies system administrators of corrupted or tampered files so that they can take damage control measures quickly and effectively. With Tripwire, system administrators can conclude with a high degree of certainty that a given set of files remain free of unauthorized modifications if Tripwire reports no changes. Tripwire works in conjunction with these other solutions to provide a "Defense in Depth"(trademark) security solution.

NOTE: Since system files should not change and users' files change constantly, Tripwire should be used to **monitor only system files**. The list of system files you want to monitor is stored in **./configs/tw.conf**.

The system administrator should install Tripwire on a clean system. This baseline database will then be used to compare possible changes to files and directories to make sure the system has not

been compromised. If the system has been compromised, information provided by Tripwire can be used to carry out a forensics investigation of the compromise. Forensics is the compiling of the chain of evidence necessary to prosecute offenders after an attack has occurred.

The system administrator should check any changes made to the system on a weekly basis or after an alert from a security organization like NASIRC or CERT has put out an alert on security vulnerabilities for any of the baseline operating systems or COTS software.

All reported changes need to be investigated right away. The investigator should be aware that most of the file changes are due to system updates. But each change should be traceable to a specific, baselined change. If no unexplained changes are detected, then the Tripwire database needs to be updated to reflect file updates. Tripwire should be configured to mail the system administrator any output that it generates.

5.6.1 Installation of Tripwire

Use the procedures provided in the Release Notes for the relevant version of Tripwire. Release Notes are available through the "Release Notes" link at the following URL:

http://pete.hitc.com/baseline/CUSTOM_SOFTWARE/ReleaseNotes.html

5.6.2 Updating the Tripwire Database

You can update your Tripwire database in two ways. The first method is interactive, where Tripwire prompts the user whether each changed entry should be updated to reflect the current state of the file, while the second method is a command-line driven mode where specific files/entries are specified at run-time.

5.6.2.1 Updating Tripwire Database in Interactive Mode

Running Tripwire in Interactive mode is similar to the Integrity Checking mode. However, when a file or directory is encountered that has been added, deleted, or changed from what was recorded in the database, Tripwire asks the user whether the database entry should be updated.

For example, if Tripwire is run in Interactive mode and a file's timestamp changed, Tripwire will print out what it expected the file to look like, what it actually found, and then prompt the user to specify whether the file should be updated. For example:

```
/etc/hosts.equiv
    st_mtime: Wed May  5 15:30:37 2006      Wed May  5 15:24:09 2006
    st_ctime: Wed May  5 15:30:37 2006      Wed May  5 15:24:09 2006
---> File:  /etc/hosts equiv
---> Update entry?  [YN(y)n?] y ↵
```

You could answer yes or no, where a capital 'Y' or 'N' tells Tripwire to use your answer for the rest of the files. (The 'h' and '?' choices give you help and descriptions of the various inode fields.)

While this mode may be the most convenient way of keeping your database up-to-date, it requires that the user be "at the keyboard." A more conventional command-line driven interface exists, and is described next.

5.6.2.2 Updating Tripwire Database in Database Update Mode

Tripwire supports incremental updates of its database on a per-file/directory or tw.config entry basis. Tripwire stores information in the database so it can associate any file in the database with the tw.config entry that generated it when the database was created.

Therefore, if a single file has changed, you can:

 # tripwire -update /etc/newly.installed.file ↵

Or, if an entire set of files that made up an entry in the tw.config file changed, you can:

 # tripwire -update /usr/local/bin/Local_Package_Dir ↵

In either case, Tripwire regenerates the database entries for every specified file. A backup of the old database is created in the **./databases** directory.

Tripwire can handle arbitrary numbers of arguments in Database Update mode.

The script **twdb_check.pl** script is an interim mechanism to ensure database consistency. Namely, when new entries are added to the tw.config file, database entries may no longer be associated with the proper entry number. The twdb_check.pl script analyzes the database, and remaps each database entry with its proper tw.config entry.

5.6.3 Configuring the tw.config File

Edit your **tw.config** file in the **./configs** directory, or whatever filename you defined for the Tripwire configuration file, and add all the directories that contain files that you want monitored. The format of the configuration file is described in its header and in the "man" page. Pay especially close attention to the select-flags and omit-lists, which can significantly reduce the amount of uninteresting output generated by Tripwire. For example, you will probably want to omit files like mount tables that are constantly changed by the operating system.

Run Tripwire with **tripwire -initialize**. This will create a file called **tw.db_[hostname]** in the directory you specified to hold your databases (where [hostname] will be replaced with your machine hostname).

Tripwire will detect changes made to files from this point on. You ***must*** be certain that the system on which you generate the initial database is clean; however, Tripwire cannot detect unauthorized modifications that have already been made. One way to do this would be to take the machine to single-user mode, reinstall all system binaries, and run Tripwire in initialization mode before returning to multi-user operation.

This database must be moved someplace where it cannot be modified. Because data from Tripwire is only as trustworthy as its database, choose this with care. It is recommended to place all the system databases on a read-only disk (you need to be able to change the disk to writeable

during initialization and updates, however), or exporting it via read-only NFS from a "secure-server." (This pathname is hardcoded into Tripwire. Any time you change the pathname to the database repository, you must recompile Tripwire. This prevents a malicious intruder from spoofing Tripwire into giving a false "okay" message.)

We also recommend that you make a hardcopy printout of the database contents right away. In the event that you become suspicious of the integrity of the database, you will be able to manually compare information against this hardcopy.

Once you have your database set up, you can run Tripwire in Integrity Checking mode by typing **tripwire** on the command line from the directory in which Tripwire has been installed.

5.7 Reporting Security Breaches

Reporting of Security breaches shall be in accordance with NPR 2810.1, *NASA Procedural Requirements: Security of Information Technology*. The specific location in the 2810 is the section on IT Security Incidents Reporting and Handling.

5.8 Initiating Recovery from Security Breaches

Recovery from Security breaches shall be in accordance with NPR 2810.1, *NASA Procedural Requirements: Security of Information Technology*. The specific location in the 2810 is the section on IT Security Incidents Reporting and Handling.

This page intentionally left blank.

6. Network Administration

This section covers the procedures necessary for the management operations that monitor and control the system network capabilities.

Detailed procedures for tasks performed by the Network Administrator are provided in the sections that follow. The procedures assume that the administrator is authorized and has proper access privileges to perform the tasks (i.e., root).

6.1 Network Documentation

EMD Network Administration requires access to restricted documents that are posted on the EMD Baseline Information System (EBIS) Site (http://pete.hitc.com/baseline/) but are not available on the public mirror site (http://cmdm-ldo.raytheon.com/baseline/). The following restricted documents provide network documentation:

- DAAC LAN Topology 921-TD*x*-001
  (*x* = DAAC designation: G = GSFC; L = LaRC; N = NSIDC; E = LP DAAC)
- [DAAC] Hardware/Network Diagram 921-TD*x*-002
- IP Address Assignment (DAAC Hosts) 921-TD*x*-003
- IP Address Assignment (DAAC Network Hardware) 921-TD*x*-004

The documents describe and depict the network layout and inter/intra-connections necessary to understand the system. Contact EMD Configuration Management for versions relevant to an individual site.

6.2 Network Monitoring

6.2.1 Big Brother - Better Than Free Edition

Big Brother Better Than Free Edition (BTF) is a network monitoring and notification COTS application. DAAC network administrators use it to monitor network devices and the services on those devices and to get feedback on their network's performance. Basic procedures common activities and additional information is available in 609-EMD-003, Release 7.20 Operations Tools Manual for the EMD Project

6.2.2 WhatsUp Professional 2006 Premium (WhatsUp Professional) -Removed

6.3 DAAC LAN Topology Overview

The LAN topology at each DAAC is unique. The detailed network topology for each DAAC is not presented in the student guide due to network security concerns. However, the details will be discussed during the class presentation.

In spite of the unique design of each DAAC's LAN, there are some common features.

- The Production Network at each Distributed Active Archive Center (DAAC) consists of an Ethernet Virtual Local Area Network (VLAN) supported by Ethernet 10-Mb/s, 100-Mb/s, and Gigabit Ethernet (GigE) connections and a GigE switch.
- The VLAN Ethernet switch is connected to a Portus Firewall by a Gigabit Ethernet connection.
- The firewall provides access to both the M&O Network and a VLAN outside the firewall that provides connections to external networks, such as the following services:
  - Campus network.
  - Very high performance Backbone Network Service (vBNS+).
  - Internet Protocol Network Operations Center (IPNOC) and other parts of the NASA Integrated Services Network (NISN).

System hosts within a DAAC are connected to the VLAN Ethernet switch. The switch is used to connect hosts at 10/100/1000 Mb/s. The VLAN Ethernet switch is connected to the Portus Firewall via a 1000-Mb/s (GigE) connection.

In addition to their connections to the VLAN Ethernet switch the ACM/Ingest hosts, PDR Server hosts, Data Server hosts, and science processors are interconnected through a backend GigE switched network. This setup provides a high-speed network to handle the large data transfers among subsystems. The GigE switch shifts the numerous transfers of large volumes of data onto a dedicated high-speed topology, freeing the VLAN to handle control flows and DAAC-to-DAAC processing flows.

6.4 Network Hardware Components

The DAAC LANs consist of the following major hardware components:

- Portus Firewall.
- VLAN Ethernet Switch.
- GigE Switch.

6.4.1 Portus Firewall

The Portus Firewall hardware consists of an IBM RS6000 server installed with the basic AIX 5.3operating system. It contains two 9GB internal disk drives that are mirrored, as well as a pair of redundant power supplies. Distributed FTProxy SOCKS is provided by Portus Forewall, which has replaced E-Borders Server.

6.4.2 VLAN Ethernet Switch

The Ethernet switch at each DAAC is a Cisco Catalyst 6506E , which provide a large number of 10/100/1000-Mb/s interfaces. The VLAN1 Ethernet switch interfaces with all Production hosts and the Portus Firewall. The VLAN10 Ethernet switch interfaces with the Portus Firewall and routers to external networks.

Maintenance and configuration of the Ethernet switche is considered non-trivial functions. Such tasks are addressed in special technical training provided by the vendor and supplemental training provided by EMD.

6.4.3 GigE Switch

The GigE Switch is Cisco Catalyst 3550. It connects some of the SGI hosts to a high-speed switching fabric.

Maintenance and configuration of the GigE Switch are considered non-trivial functions. Such tasks are addressed in special technical training provided by the vendor and supplemental training provided by EMD.

6.5 Domain Name Service (DNS) Structure

The parent DNS domain for the system is **ecs.nasa.gov**. These DNS servers reside at the SMC, and the NSIDC DAAC. In this domain are the SMC hosts, User hosts for all DAACs, and pointers to the DAACs' external DNS servers. The external DNS resides on the Portus Firewall.

The ecs.nasa.gov DNS servers are:

- m0isi08.ecs.nasa.gov
- n0fwi09u.ecs.nasa.gov

The DAACs' Production networks are a child domain of ecs.nasa.gov. They are:

- LaRC Production networks:
  - l0ins02.larcb.ecs.nasa.gov (internal)
  - l0css02.larcb.ecs.nasa.gov (internal)
  - l0isi08.larcb.ecs.nasa.gov (external)
- LP DAAC Production networks:
  - e0ins02.edcb.ecs.nasa.gov (internal)
  - e0css02.edcb.ecs.nasa.gov (internal)
  - e0fwi09.edcb.ecs.nasa.gov (external)
- NSIDC Production network:
  - n0ins02.nsidcb.ecs.nasa.gov (internal)
  - n0css02.nsidcb.ecs.nasa.gov (internal)
  - n0fwi09.nsidcb.ecs.nasa.gov (external)
- GSFC Production networks:
  - g0ins02.gsfcb.ecs.nasa.gov (internal)
  - g0css02.gsfcb.ecs.nasa.gov (internal)
  - g0fwi09.gsfcb.ecs.nasa.gov (external)

The DAACs' M&O networks are also a child domain of ecs.nasa.gov. They are:

- LaRC M&O network.
  - larcmo.ecs.nasa.gov
- LP DAAC M&O network.
  - edcmo.ecs.nasa.gov

- NSIDC M&O network.
  - nsidcmo.ecs.nasa.gov
- GSFC M&O network.
  - gsfcmo.ecs.nasa.gov

6.6 Host Names

A letter is appended to the production host name to distinguish which interface (and IP address) a user is accessing.

As an example, a GSFC DAAC host named g0acg01.gsfcb.ecs.nasa.gov is a host attached to the Production network.

6.7 Network Security

6.7.1 Network Connectivity

The system network was designed to minimize unauthorized user access, including the use of a firewall at each site. Ingest network access at a DAAC is limited to its Level 0 data provider(s), the SMC, and hosts attached to the DAAC's Production and M&O networks. No local campus, Internet or other DAAC access is provided. Access to a DAAC's Production network is limited to the SMC, the DAAC's M&O network, and other DAACs. No local campus, Internet, or Level 0 data provider(s) access is provided.

6.7.2 Troubleshooting - Verifying Connectivity

One of the key reasons for failure of data access and transfer is an error or problem in system connectivity. This can be caused by a myriad of glitches such as incorrect/outdated lookup tables, incorrectly assigned IP addresses, missing default route and more. Besides checking individual host/server operation with various tools such as ECS Assistant, you can use several command line entries to verify point-to-point communication between components.

There are three initial steps to help verify system connectivity:

- Ensuring connectivity is authorized (check the Network Connectivity matrix).
- Determining whether the Domain Name Service (DNS) is resolving host name and IP addresses correctly.
- Actively testing the connectivity using the ping function.

6.7.2.1 Checking Local Host Access to Another Local Host over the Network

1 To check the Domain Name Service entries (DNS) for the source host on workstation ***x0xxx##*** at the UNIX prompt enter:

nslookup <local_host>

- The screen display will be similar to the following:

```
g0spg01{mblument}[204]->nslookup g0spg01
Server:  g0css02.gsfcb.ecs.nasa.gov
Address:  xxx.xxx.xxx.xx
Name:    g0spg01.gsfcb.ecs.nasa.gov
Address:  xxx.xxx.xxx.xx
```

2 To check the DNS entries for other host on the Production network enter:

nslookup <other host>

- The screen display will be similar to the following:

```
g0spg01{mblument}[201]->nslookup g0css02
Server:  g0css02.gsfcb.ecs.nasa.gov
Address:  xxx.xxx.xxx.xx
Name:    g0css02.gsfcb.ecs.nasa.gov
Address:  xxx.xxx.xxx.xx
```

3 To determine the host's network interface parameters enter:

netstat -i

- The **netstat -i** command will provide the following information:

```
g0spg01{mblument}[201]->netstat –i
Name Mtu   Network     Address          Ipkts Ierrs    Opkts Oerrs  Coll
ipg0 4352  xxx.xxx.xxxg0spg01.gsfcb.   9182666     1  8103032     0     0
hip0 65280 xxx.xxx.xg0spg01h.gsfcb.  5554524     0  6776651     0     0
xpi0 4352  xxx.xxx.xxx.xxg0spg01u.ecs.   37850320     0 14109683     3     0
xpi1 0     none        none                0     0       0     0     0
et0* 1500  none        none                0     0       0     0     0
lo0  8304  loopback    localhost        314800     0   314800     0     0
```

4 To determine the host's network interface enter:

ifconfig <interface>

- Using **ipg0** from the **ifconfig <interface>** data as the interface parameter, **ifconfig ipg0**, will result in the following display:

```
g0spg01{mblument}[203]->ifconfig ipg0
ipg0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST>
inet xxx.xxx.xxx.xx netmask 0xffffff00 broadcast xxx.xxx.xxx.xxx
```

5 To ping the local host to verify inter-connectivity enter:

ping <local host>

- For example:

```
g0spg01{mblument}[232]->ping  g0spg01
PING g0spg01.gsfcb.ecs.nasa.gov (xxx.xxx.xxx.xx): 56 data bytes
64 bytes from xxx.xxx.xxx.xx: icmp_seq=0 ttl=255 time=0 ms
64 bytes from xxx.xxx.xxx.xx: icmp_seq=1 ttl=255 time=0 ms
64 bytes from xxx.xxx.xxx.xx: icmp_seq=2 ttl=255 time=0 ms
64 bytes from xxx.xxx.xxx.xx: icmp_seq=3 ttl=255 time=0 ms
64 bytes from xxx.xxx.xxx.xx: icmp_seq=4 ttl=255 time=0 ms
 ----g0spg01.gsfcb.ecs.nasa.gov PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0/0/0 ms
g0spg01{mblument}[233]->
```

6 To verify inter-connectivity with other hosts on the Production network enter:

ping <remote host>

- For example:

```
g0spg01{mblument}[202]->ping  g0css02
PING g0css02.gsfcb.ecs.nasa.gov (xxx.xxx.xxx.xx): 56 data bytes
64 bytes from xxx.xxx.xxx.xx: icmp_seq=0 ttl=255 time=2 ms
64 bytes from xxx.xxx.xxx.xx: icmp_seq=1 ttl=255 time=1 ms
64 bytes from xxx.xxx.xxx.xx: icmp_seq=2 ttl=255 time=1 ms
64 bytes from xxx.xxx.xxx.xx: icmp_seq=3 ttl=255 time=1 ms
64 bytes from xxx.xxx.xxx.xx: icmp_seq=4 ttl=255 time=1 ms
 ----g0css02.gsfcb.ecs.nasa.gov PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 1/1/2 ms
```

Note: You cannot ping hosts outside of the Production network because of the Protus firewall implemetation. See your local firewall administrator for assistance when troubleshoot connectivity issues with hosts outside of the Production LAN.

7 To check the health of the interface enter:

netstat -i

- The following type of result is returned:

```
g0spg01{mblument}[218]->netstat -i
Name Mtu   Network     Address                Ipkts        Ierrs        Opkts        Oerrs  Coll
ipg0 4352  xxx.xxx.xxxg0spg01.gsfcb.          9197317      1            8113487      0      0
hip0 65280 xxx.xxx.xg0spg01h.gsfcb.           5554541      0            6776668      0      0
xpi0 4352  xxx.xxx.xxx.xxg0spg01u.ecs.        37851779     0            14109837     3      0
xpi1 0     none            none               0            0            0            0      0
et0* 1500  none             none              0            0            0            0      0
lo0  8304  loopback    localhost              325510       0            325510       0      0
```

8 Examine the output of the **netstat** command to determine whether there are any Ierrs and/or Oerrs.

- One or two errors are acceptable, 100 errors are not.
  - A lot of errors indicate an interface problem; check the syslog for any startup or logged problems from the OS.

6.7.2.2 Checking Host Communication across external networks

You cannot ping or traceroute to hosts outside the Production network because of the firewall implemetation. See your local firewall administrator for assistance when troubleshoot connectivity issues with hosts outside of the Production network.

This page intentionally left blank.

7. System Monitoring

7.1 Overview

This chapter covers procedures for the management operations that monitor the network and server applications. Graphical tools available to monitor system status include **Big Brother (BTF)**, **ECS Health Check GUI**, **ECS Assistant/ECS Monitor**, **EcMs-Whazzup??**, and a script **EcCsIdPingServers** that permits an operator to ping all servers. These programs provide system monitors with real-time status of the system and indications of potential problem areas.

7.2 Checking the Health and Status of the Network

7.2.1 Big Brother

Big Brother Better Than Free Edition (BTF) is a network monitoring and notification COTS application. DAAC network administrators use it to monitor network devices and the services on those devices and to get feedback on their network's performance.

Big Brother is a Web-based COTS application used to monitor network devices and services on the EMD Production LANs. Big Brother capabilities are executed through the use of GUIs. The GUI's instructions are referenced in the EMD 609 document. Installation and configuraition of the product can also be found in EMD 914-TDA-382, the PSR document.

Common functions performed by Big Brother are shown below in Table 7.2-1:

Table 7.2-1. Common functions performed by Big Brother

Operating Function	GUI	Description	When and why to Use
View network devices/services status	View icon color and on web GUI; view quick status dialog box.	Icon color indicate the status of network devices and services.	To verify that all network devices and services on the devices are operational. To ascertain network devices and services that are not operating properly.
View network devices/services performance data	Logs and Report menus on GUI	A set of reports that can be viewed, printed, and/or its content transmitted to a file.	To obtain status information about monitored devices and services.

7.3 Using WhatsUp Professional Reports - Removed

7.4 Starting and Using the ECS Health Check GUI

The **ECS Health Check GUI** indicates the status of the EcDmV0ToEcsGateway and Data Pool drill-down search engine. At a specified rate it sends inventory searches to the EcDmV0ToEcsGateway and/or submits granule search requests to the Data Pool. It provides warnings by the following means when a failure is registered by the GUI during the current search:

- Visual warning (including details about the time and nature of the error).
- Audible alarm (when implemented).
- E-mail message.

Table 7.4-1 contains the Activity Checklist for ECS Health Check.

Table 7.4-1. ECS Health Check - Activity Checklist

Order	Role	Task	Section	Complete?
1	SA	Starting ECS Health Check	(P) 7.4.1	

7.4.1 Starting ECS Health Check

1 Log in to the Internalhost as cmshared.

- The EcCsHealthCheckStart script must be run by cmshared.
- Examples of Internal hosts include , **e4eil01**, **l4eil01**
- For detailed instructions refer to the **Log in to ECS** procedure (previous section of this lesson).

2 At the UNIX prompt type **cd /usr/ecs/<*MODE*>/CUSTOM/bin/CSS** and then press the **Return/Enter** key.

- Change to the directory containing the start script (i.e., EcCsHealthCheckStart) for the ECS Health Check GUI.
- The location is different from that of most other ECS start scripts because the COTS dependency on the V0ImsClientSimulator application requires that the parent application be run in the same directory as the V0ImsClientSimulator binary.

3 At the UNIX prompt type **EcCsHealthCheckStart <*MODE*>** and then press the **Return/Enter** key.

- The ECS Health Check GUI is displayed.

4 Click on the appropriate tab for the area to be monitored.

- The following choices are available:
  - **EcDmV0ToEcsGateway.**

- **ECS Datapools.**

5 In the **Specify timeout period minutes** and **seconds** text entry boxes type the number of minutes and seconds (respectively) before timeout.

- **Timeout period** specifies how long the GUI will wait for a response from the current search before flagging an error.

6 In the **Specify repeat period minutes** and **seconds** text entry boxes type the number of minutes and seconds (respectively) before repeating a search.

- **Repeat period** specifies how often the GUI sends an inventory search to the area being monitored.

7 Click on the **Start** button in the **Control** pane to start checking the selected area.

- The ECS Health Check GUI starts sending inventory searches to the selected area at the frequency specified in the **Specify repeat period** text boxes.
- **The Current Status is:** (as displayed on the GUI) changes from **Dormant** to **Running**.

8 Repeat Steps 4 through 7 to set up inventory searches of the other area (if applicable).

9 Observe information displayed in the **Visual Warnings** pane of the ECS Health Check GUI and listen for audible warnings (if enabled).

- Nothing much will appear to happen unless an error occurs in an inventory search sent to the EcDmV0ToEcsGateway/Data Pool, in which case the following indications will be evident:
  - **The Current Status is:** (as displayed on the GUI) changes from **Running** to **Failed** (in red).
  - Details concerning the time and nature of the error are displayed in the **Visual Warnings** text pane.
  - The **Mail Warnings** pane indicates that mail has been sent successfully to the recipients in the list.
  - A repeating audible alarm sounds (if enabled).
- If an error occurs in a search sent to the EcDmV0ToEcsGateway/Data Pool, no further inventory searches will be sent to the selected area unless restarted using the **Start** button in the **Control** pane (refer to Step 13).

10 To stop an audible alarm (when applicable) click on the **Stop** button in the **Audible Warnings** pane.

- The alarm stops sounding.

11 To clear error information from the ECS Health Check GUI (when applicable) click on the **Reset** button in the **Control** pane.

- The error information is cleared from the ECS Health Check GUI.

12 To stop inventory searches of the selected area (when applicable) click on the **Stop** button in the **Control** pane.

- The GUI goes to a dormant state.

13 To restart inventory searches of the selected area (when applicable) return to Step 7 (click on the **Start** button in the **Control** pane).

- The GUI goes to a dormant state.

14 To exit from the ECS Health Check GUI (when applicable) select **File → Exit** from the pull-down menu.

- The ECS Health Check GUI is dismissed.

7.5 Monitoring and Managing Server Applications

Whazzup??? is a management tool that provides operators and maintainers with a means of monitoring and checking servers, for quickly identifying servers that may have problems, and for isolating faults. It is a web-based application, and is therefore accessed by means of browser software. It provides the following general features:

- Host and mode views of network resources.
- Status information on resources (indicated by color coding: purple indicates inability to ping the specified host, blue indicates incomplete data collection, red indicates that the server is down, and yellow indicates that a warning threshold has been exceeded).
- Performance monitoring capability.

Another set of tools for monitoring and managing system resources is **ECS Assistant** and its companion, **ECS Monitor**, which offer:

- Installation support.
- Indication of network and server status and changes.

There is an accompanying script, **EcCsIdPingServers**, which provides the capability to ping all servers.

Table 7.5-1 contains the Activity Checklist for Monitoring and Managing Server Applications.

Table 7.5-1. Monitoring and Managing Server Applications - Activity Checklist

Order	Role	Task	Section	Complete?
1	SA	Launching Whazzup	(P) 7.5.1.1	
2	SA	Starting ECS Assistant	(P) 7.5.2.1	
3	SA	Starting ECS Monitor	(P) 7.5.2.2	
4	SA	Using EcCsIdPingServers to Ping All Servers in a Mode	(P) 7.5.2.3	

7.5.1 Whazzup??

A powerful COTS program that has been modified for monitoring the system is EcMsWz-Whazzup??. It is a web-accessed program that provides a graphical display of Host Status, Mode Status, Mode Verification and Performance Management. The welcome screen has buttons and links at the bottom permitting an operator to view status by various means (e.g., host, mode), verify modes and view what servers may be down, and access data on performance. The **Performance Stats** screen provides a quick overview of the system status; if Whazzup is unable to ping a host, the row for that host is highlighted in purple.

These functions of Whazzup?? provide graphical displays of host and software-server status in real-time mode. When used in conjunction with WhatsUp Professional and ECS Assistant, Whazzup?? can provide System Administrators with a comprehensive knowledge of the system's status.

7.5.1.1 Launching WhazzUp

1 On workstation ***x4spl01***, at the UNIX prompt in a terminal window, type **netscape &** and then press the **Return/Enter** key.

2 In the location field, type **http://*x4spl01*:5150** and press the **Return/Enter** key.
- The **EcMsWz-Whazzup???** screen is displayed.

3 At the bottom of the screen, click on the **Verify Mode** option button and, in the resulting pop-up menu, drag the cursor to highlight the option **What's Down**.
- The screen displays a table showing **Required Servers Currently Down . . .**, listing by mode the servers that are down.

4 Move the mouse to position the cursor on the **Performance** link, click the **Right Mouse Button**, and select **Open Link in New Window**.
- The **Performance Stats** screen is displayed in a new window, showing information that may help determine the reason for any servers being down.

5 If desired, click on the link for any host to obtain more detailed information.
- An information screen for the selected host is displayed, showing data on system memory, disk utilization, process information, and network information.

7.5.2 ECS Assistant and ECS Monitor

The ECS Assistant and ECS Monitor tools provide additional easy-to-use tools that offer a server monitoring capability (ECS Monitor) as well as a capability to start and stop servers (ECS Assistant).

7.5.2.1 Starting ECS Assistant

1 Log in to one of the host machines.

2 At the UNIX prompt on the host from which the ECS Assistant is to be run, type **setenv ECS_HOME /usr/ecs**, and then press the **Return/Enter** key.
- To verify the setting, type **echo $ECS_HOME**, and then press the **Return/Enter** key.

3 At the UNIX prompt, type **cd /tools/common/ea**, and then press the **Return/Enter** key.
- The working directory is changed to **/tools/common/ea**, the path where ECS Assistant is installed, and also where EcCoScriptlib may be found.

4 Type **EcCoAssist /tools/common/ea &**, and then press the **Return/Enter** key.
- The **ECS Assistant** GUI is displayed.

5 At the ECS Assistant GUI, click the **Subsystem Manager** pushbutton.
- The **Subsystem Manager** GUI is displayed.

6 Select a mode by clicking on the down arrow at the right end of the **Mode** field and then on the desired mode name in the resulting list.
- The selected mode is displayed in the **Mode** field and colored indicators show the installation status for components in that mode on the host; the legend for the color indications is at the lower right on the Subsystem Manager window.

7 In the list of subsystems, double click on the name of the subsystem of interest.
- One or more component groups appear below the selected subsystem name.

8 Double click on the name of a component group.
- One or more application groups appear below the selected component group name.

9 Double click on the name of the application group of interest.
- The applications or servers in the selected group are listed below the name of the group.

10 Single click on the name of an application or server of interest.
- The selected application or server is highlighted.
- Detailed installation information is displayed in the **Installation Statistics** window.

ECS Monitor provides a convenient way to monitor the status of the servers by listing their up/down condition. The **ECS Monitor** GUI has a status flag for a server indicating whether or not that server is running, and for a server that is running, the window shows the process ID (PID), the user ID, and the start time.

7.5.2.2 Starting ECS Monitor

1 Log in to one of the host machines.

2 At the UNIX prompt on the host from which the ECS Assistant is to be run, type **setenv ECS_HOME /usr/ecs**, and then press the **Return/Enter** key.
- To verify the setting, type **echo $ECS_HOME**, and then press the **Return/Enter** key.

3 At the UNIX prompt, type **cd /tools/common/ea**, and then press the **Return/Enter** key.
- The working directory is changed to **/tools/common/ea**, the path where ECS Monitor is installed, and also where EcCoScriptlib may be found.

4 Type **EcCoMonitorGui /tools/common/ea <*MODE*> &**, and then press the **Return/Enter** key.
- The **ECS Monitor GUI** is displayed, showing the status (**UP** or **DOWN**) of the servers on the current host in the mode specified in the command, as indicated near the top left corner of the window.
- The status "**UP/DOWN**" indicates whether a listed server is running.

5 To update the **Server Monitor** to show the current status at any time, click on the **update** button in the GUI.
- This causes the list to update to the current status.

6 To monitor other servers, log in to other hosts and launch the ECS Monitor GUI in the desired mode, as in Steps 1 through 4.

7 To exit click the **EXIT** button.
- This terminates display of the ECS Monitor GUI.

The script **EcCsIdPingServers** script works with a *Sweeper* binary to ping the servers and clients in a mode to determine their status.

7.5.2.3 Using EcCsIdPingServers to Ping All Servers in a Mode

1 Log in to one of the host machines.

2 At the UNIX prompt, type **cd /usr/ecs/<*MODE*>/CUSTOM/utilities**, and then press the **Return/Enter** key.
- The prompt reflects a change to directory **cd /usr/ecs/<*MODE*>/CUSTOM/utilities**, where **<*MODE*>** is likely to be **OPS**, **TS1**, or **TS2**.

3 Then type **EcCsIdPingServers <*MODE*>**, and then press the **Return/Enter** key.

- The result should appear similar to the following example:

  /usr/ecs/DEV03/CUSTOM/bin/CSS/Sweeper -nsh dss2 -nsp 22822
  FoSwSweeper application started...

---

8. Problem Management Procedures

EMD Problem Management is administered through system-level and site-level control board reviews. These control boards oversee the analysis, recommendations, and actions taken to resolve system/site problems concerning hardware, software, documentation, and procedures. Operations (OPS) Deployment and the site-level organizations resolve routine maintenance issues at the system-level and site-level, respectively, using the trouble ticket system for tracking system problems. Trouble Tickets (TTs) may evolve into Non Conformance Reports (NCR), as required, which may then be utilized to generate Configuration Change Requests (CCRs) to effect changes to the approved baseline. To ensure controlled change, NCRs are tracked using the TestTrack Pro system (TTPro) in the EDF (EMD Development Facility) and CCRs are tracked manually by Systems Engineering.

The trouble ticket process is the first vehicle used to record and report problems with the operational system. Trouble Tickets can be generated by operations, maintenance, development, and customer personnel as well as external users. The trouble ticket process uses an automated database that tracks the resolution activities associated with each trouble ticket. Documentation that is related to the problem, and is not in electronic form, or is in electronic form at the Distributed Active Archive Center (DAAC), is distributed by the local CM Administrator (CMA), and is listed as an attachment to the trouble ticket.

The CM Administrator at each site or DAAC serves as trouble ticket system administrator. Trouble Tickets are initially generated at a site, the resolution of which require changes to the system level baseline, are forwarded to the EMD Development Facility (EDF), where they are reviewed and *translated* into NCRs. Additionally, Trouble Tickets and CCRs that are generated at the sites, which are repaired locally, and result in site-unique extensions to the system level baseline, are forwarded to the SMC for tracking across the EMD baseline. The EMD Problem Review Board (PRB) designee is responsible for tracking TTs after they have been received from the sites, and for propagating system problem resolutions for site visibility.

CMAs also support the activities of the local Problem Review Board. This includes generating status reports, and implementing resolutions, instructions, and changes as directed by the Board. User Services Representatives monitor TTs to notify users concerning problem resolution and status. Maintenance Engineers at respective levels will record all activities in the trouble ticket. This information can be used to determine critical maintenance concerns related to frequency of occurrence, criticality level, and the volume of problems experienced. The maintainability analysis will guide critical changes, volume and type of support components to be utilized, and will focus further development efforts.

This section provides an overview of the Trouble Ticketing process and defines the procedures for processing and resolving trouble ticket submissions. In addition, this section provides a scenario for handling Emergency Fixes.

8.1 The Problem Resolution Process — An Overview

8.1.1 Internal Process

Any system user may submit Trouble Tickets (TTs) using the local trouble ticketing system at any site or DAAC. TT submission triggers an internal review by the site's review board. Primary objectives of the internal review are to quickly identify and correct problems that fall within the site's capability to maintain, review and validate the priority of the problem. Thus, then is to elevate to the system level those problems that either exceed their capability to repair, or that require a change to the system level baseline.

Problems passed from the sites to the ECS system level, are passed by transferring the trouble ticket from the local trouble ticket system to the EDF trouble ticket system. Here they are reviewed by the EMD Problem Review Board (PRB), which hosts daily teleconferences, known as the PRB Telecon.

The permanent membership of the Problem Review Board (PRB) is as follows:

a. Chair: Problem Management Lead or designee

b. Each DAAC: one member representative

c. Quality Assurance: one member representative

d. ESDIS: one member representative

The roles and responsibilities of the participants in the PRB Telecon are to:

a) Follow a nominal agenda that includes the following discussions:

i) New trouble tickets.

ii) Deferred trouble tickets.

iii) Aging high-status trouble tickets.

iv) Review of all Severity 1 (Sev 1) NCRs (OPS Only).

v) Review of new OPS NCRs. (Full Report Only)

b) Review severity of each NCR according to the guidelines specified in the

Operations Class NCR Management Process (MO-1-003-5)

c) Convert any TTs that identify a system non-conformance and have the appropriate information elevated to an NCR, by requesting that the Problem Management Lead forward the TT into TTPro (TestTrack Pro) tracking repository. NCRs document the system non-conformance and are worked by the Sustaining Engineering staff.

d) NCRs that are deemed to not have enough information for proper adjudication will either be rejected as incomplete or they will be deferred until a future meeting, as determined by the PRB chairperson. The PRB Administrator will maintain a list of deferred items and place those items on a future PRB agenda pending addition information

The PRB performs a preliminary review of each trouble ticket to confirm the severity assigned by the site, the completeness of information and data relevant to the problem, and whether it requires a change to the system level operational baseline.

The PRB forwards TTs to the Development Sustaining Review Board, which is comprised of a Development Lead and a COTS technical investigation, resolution proposal, and NCR preparation, when required. The CM Administrator (CMA) or DAAC Support Help Desk members provide administrative support to the PRB by publishing minutes of teleconference meetings, tracking TTs and action items, and by translating TTs to NCRs.

The Sustaining Engineering Review Board has the authority to direct resolutions to trouble ticket problems that do not change or which in any way affects the EMD operational baseline and baseline documentation. An NCR is required when the Technical Investigation (TI) determines that the operational baseline must be changed in order to correct the problem identified in the trouble ticket.

The PRB is not a voting board; the membership is appointed for the purpose of providing timely, direct technical support to the Chair, who has the decision-making responsibility and authority.

The Science Development (SCDV) CCB is not a voting board. It does have the authority and responsibility to approve Class II changes to the operational baseline. (Class II changes are described in the *Configuration Management Plan for the EMD Project* (110-EMD-001).) Specific responsibilities include:

a. Review, approve and schedule; review and backlog; or reject each NCR's proposed resolution, or cost and schedule input from the Responsible Engineer (RE).

b. Approve the schedule for the deployment of configuration changes in the form of a 'drop' to the SMC.

c. Manage and adjust the schedule and contents of each block in accordance with program priorities and the progress of NCR work-off.

d. Review the status of all backlogged NCRs on a periodic basis. Schedule NCRs for a future block, as appropriate

e. Collect and report on NCR statistics.

The permanent membership of the SCDV CCB is as follows:

a. Chair: EMD Systems Engineering and Integration Team (SEIT) designee.

b. Members of the EMD SEIT and designees.

c. Each DAAC: one representative.

8.2 Problem Management Procedures

The TestTrack Pro (TTPro) trouble tickets and NCRs system is database repository that has replaced the Remedy Action Request System. This repository is a Commercial Off-The-Shelf (COTS) product that provides a distributed trouble ticketing service which provides a common

environment and means of classifying, tracking, and reporting problem occurrence and resolution to both system users and operations personnel. The trouble ticketing service:

- Provides a GUI for operations personnel to access all Trouble Ticket services.
- Provides a common Trouble Ticket entry format.
- Stores Trouble Tickets and NCRs in one centralized repository.
- Retrieves Trouble Tickets and NCRs via ad hoc queries from internet or PC client.
- Allows operations personnel to forward problems from one DAAC to another.
- Produces stock and common reports.
- Provides an interface to user's and operator's e-mail to provide automatic notification.
- Offers portability of application from PC to PC.
- 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 re-prioritization and action rules. This is the role of the onsite Administrator who will manage the customizable TTPro repository at the DAAC.

In addition to the functionality provided by TestTrack Pro, the trouble ticketing service utilizes a set of custom HTML pages ("screens") to provide registered users with the ability 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 with which most are already familiar. Support staff members are able to enter Trouble Tickets through the trouble ticket system interface for problems received via other methods (for example, phone calls).

The TestTrack Pro system also functions as a User Contact Log and is configured to accept entries that User Services personnel enter for each contact that they receive from a user.

Users submit trouble tickets to the User Services Desk located at the DAACs. External users submit trouble tickets through the Internet [using a series of hypertext mark-up language (HTML) screens]. Site personnel submit trouble tickets via the trouble ticket system (TTPro).

User Services personnel process trouble tickets for problem resolution in accordance with local policy, ensuring each trouble ticket is logged into the database. Trouble tickets are first evaluated to determine the severity of the problem and assignment of on-site responsibility. Trouble tickets that can be resolved locally are assigned and tracked at the local center. The Operations Supervisor reviews each trouble ticket (for priority verification and problem description) and assigns it to an appropriate Maintenance Engineer for resolution.

Matters that require external or higher-leveled assistance, such as a repair which require changes to the system baseline are escalated via the PRB Telecon for discussion and disposition. The

telecon is held to coordinate trouble ticket activities within the EMD organization as well as with development, customer, and user organizations.

8.3 Using the Trouble Ticket System

1. User or Operator discovers a problem with the system (hardware, software, documentation, procedure) and documents this problem for later resolution. The submitter forwards a trouble ticket to User Services by:

    - activating the TestTrack Pro ticketing system via the Internet;
    - phoning User Services directly;

2. The trouble ticket is logged into the TTPro system. It automatically assigns "Open, not assigned" status to the trouble ticket and notifies the Operations Supervisor for assignment and prioritization. The status of each trouble ticket is the responsibility of Operations as it progresses through the resolution process.

3. Categorizing Problem Severity. The DAAC Help Desk will review the information provided by the DAAC to determine if the problem has been described in enough detail to warrant the recommended severity and category. If there is insufficient information, the DAAC submitter or point of contact will be contacted. The Trouble Ticket will be deferred to the Sustaining Engineering Review Board for further review.

    In determining severity of the problem, the DAAC Help Desk must consider the following factors:

    - Impact on the ability to ingest, process or distribute satellite data
    - Frequency of occurrence
    - Availability of an adequate work-around

    Refer to Table 8.3-1 to determine problem severity.

Table 8.3-1. Trouble Ticket Priority/NCR Severity

As Documented in NASA 420-05-03	As Used/Interpreted by the EMD Project
Category 1: System/Service cannot perform critical function or imposes major safety hazard. (Priority 1) Presents an immediate impact to development, operations, services, or data processing functions; imposes major safety hazard to personnel, systems, or space mission resources; or results in loss of one or more essential mission objectives.	**HIGH (Severity 1)**: An NCR which causes: – Inability to perform a mission-critical function (i.e., Ingest/Pre-Processing/Archiving of Science Data, Planned Processing, Browse/Order/Distribute); – Performance of a mission-critical function to be so degraded that production minimum goals cannot be achieved; – A mission-critical function to be performed improperly, resulting in permanent loss of data; ***and*** for which no workaround exists ***or*** for which no workaround can be accommodated by DAAC operators given a detailed workaround procedure is documented but the procedure is inadequate based upon the complexity of the procedure, the abilities of an adequately trained and experienced operator, or both.
Category 2: System/Service substantially impaired. (Priority 2) Substantially impacts development, operations, services, or data processing functions; fails to operate within critical performance specifications; or cannot effectively or efficiently fulfill baseline requirements.	**MEDIUM (Severity 2)**: An NCR with the consequence that: – The performance of a mission-critical function is degraded and may prevent achieving production minimum goals; – A mission-critical function can be only partially performed, or performs improperly, resulting in temporary loss of data or incorrect data results; – A situation (actually or potentially) severly compromises ECS mission readiness or operational integrity; – A condition exists to produce a severely degraded mission-critical function, but a workaround will allow operations to continue temporarily without permanent loss of data or severely impaired performance/workload/schedules.
Category 3: System/Service slightly impaired. (Priority 3) Causes minor or no substantial impact to development, operations, services, or data processing functions. Support may be degraded, but mission can still be accomplished.	**Severity 3**: An NCR with the consequence that: – A non-critical mission function (e.g., Advertising) cannot be performed, or yields incorrect results; – Unexpected events occur which can be corrected using normal operational procedures with minimal impacts to performance/workloads/schedules – A condition exists to produce a degraded mission-critical function, but a workaround will allow operations to continue indefinitely without severely impaired performance/workload/schedules.
	Severity 4: Improvement (Nuisance; e.g., a typo).
	Severity 5: Enhancement (Identified for next release).

4. All affected Operations Supervisors at the sites (SMC, DAACs, EDF) are notified by e-mail of the problem and solicited for inputs to problem assessment (impact) and resolution.
5. The Trouble Ticket database is updated to reflect any new information/coordination activity by the EDF Administrator or the DAAC Support Help Desk Staff
6. The Operations Supervisor assigns the problem to a Problem Investigator for further follow-up.
7. The Problem Investigator coordinates input from developers, vendors, and external organizations to affect the local resolution, presenting any significant issues at the PRB Telecon.
8. The Problem Investigator updates the trouble ticket database and forwards any information regarding proposed/implemented fixes to the established notification list.
9. In those cases where the problem resolution results in a change to the operational baseline, or where the local site wishes to elevate the TT to the PRB for advice or for resolution, the site CMA forwards the TT to the SMC.
10. The proposed resolution is then presented to the Problem Review Board for review, ratification or revision. The Problem Review Board can approve, implement, and/or close changes that do not affect controlled configuration items.

11. All system level changes are proposed in Non Conformance Reports (NCRs). The SMC opens an NCR in the Change Request Manager and closes the trouble ticket in TestTrack Pro. Emergency fixes (High TTs) can be made locally with the approval of the local CCB and then be forwarded to the SMC in a trouble ticket.
12. The off-site problem resolution process is monitored by the EMD Problem Review Board, which may also revise the proposed solution because of any system-level effect(s).
13. The NCR may be escalated to higher levels for system and/or external elements that may be involved in the resolution process.

8.3.1 Accessing the Trouble Ticket System

The Seapine TestTrack Pro (TTPro) trouble ticket system may be accessed through either the web or PC client. The web version is used by both User Services and the end user to submit, browse, edit, and query trouble tickets. Users can also use a number of defined templates (html formatted) to generate defect reports. Predefined and user customized filters allow display of only those issues of interest, including advanced filters created to perform detailed trouble tickets searches. Detailed instructions for TestTrack Pro may be found in the TestTrack Pro Web Client User Guide at the Seapine Software company's web site:

http://www.seapine.com/ttpevaltools.php

Through TestTrack Pro, the User clicks on the User Tool icon, which launch the PC client version which displays the client's secure login screen, listing of collectively centralized projects that directs access to both Trouble Tickets and NCRs. Once the users access the needed project, the TestTrack Pro (TTPro) Defects Project Window displays list of all or specified trouble tickets. Forwarding to a selected defect, the user has the ability to navigate through the trouble ticket, viewing its main form to submit, query, or work a Trouble Ticket. The Main TestTrack Pro Trouble Ticket GUI (Figure 8.3-1) is used to select the appropriate fields and workflow tabs for submitting, modifying, or displaying a trouble ticket and its related components. The Main Page data fields are identified in Table 8.3-3 and its associated tabs are identified in Table 8.3-4, the Source Code, Notify, and Links tab are not used, and therefore are omitted.

Table 8.3-2 provides the activity checklist for the Trouble Ticket System.

Table 8.3-2. Trouble Ticket System - Activity Checklist (1 of 2)

Order	Role	Task	Section
1	System Users	Access the Trouble Ticket System	8.3.1, 1
2	System Users	Submit Trouble Ticket	8.3.2.1
3	Maintenance Engineer	Review and/or Modify Open Trouble Ticket	8.3.3.1
4	Database Administrator	Add Users to Trouble Ticket System	8.3.5.1
5	CMA	Modify TestTrack Pro System User Privileges	8.3.6.1

Table 8.3-5. Trouble Ticket System - Activity Checklist (2 of 2)

Order	Role	Task	Section
6	CMA	Modify TestTrack Pro System Configuration	8.3.7.1
7	Maintenance Engineer, CMA	Generate Reports	8.3.8.1

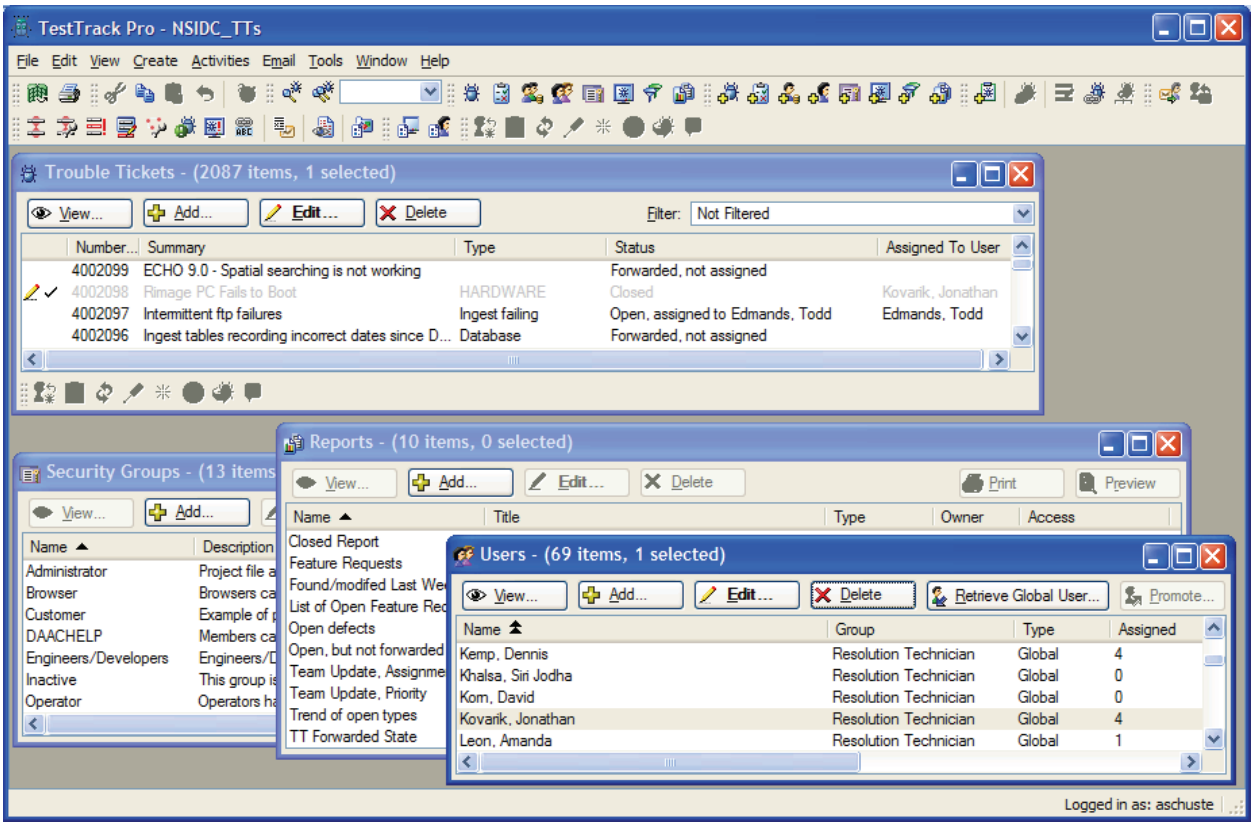

Figure 8.3-1. Main TestTrack Pro Trouble Ticket GUI

Table 8.3-3. TestTrack Pro Trouble Tickets MAIN GUI Field Description

Field Name	Data Type	Size	Entry	Description
Number	Integer	8	System generated	Ticket number, which is set and maintained by the system.
Summary	Character	154	Required	Short description of the problem. Displays as defect listing line in defects project window.
Status	Character	n/a	Automatic	Indicate the assignment and state of transition of defect
Type	Selection	30	Optional	Type of problem addressed by this Trouble Ticket (e.g., configuration error, hardware problem, software problem).
Product	Selection	*	Optional	Product exhibiting the problem or issue
Entered by	Character	30	Required	User name of the Submitter or person who created the defect.
Mode	Selection	*	Optional	Run mode in which problem was detected.
DAAC Trouble Ticket	Character	20	Optional	Unique identifier that is established at the origination site. Legacy identifier of defect (from REMEDY ARS)
DAAC POC	Character	n/a	Optional	Name of the issue's point of contact at the DAAC. Used when escalating Trouble Tickets to Landover PRB for advice or resolution.
Submitter Site	Selection	*	Optional	Home DAAC of the Submitter or defect's originating site.
Priority	Selection	*	Optional	Priority of trouble ticket assigned at the site (HIGH, MEDIUM, LOW).
Component	Selection	*	Optional	Product's component exhibiting the problem or issue. In legacy (REMEDY) defects, it's the name of the configuration item to which the problem is associated.
Severity	Selection	*	Required	Impact of the problem to the submitter (HIGH, MEDIUM, LOW).
Date Entered	Date	n/a	Optional	Date Trouble Ticket was created at the present site.
Description	Character	4060	Optional	Detailed description of the problem
Machine Name	Character	60	Optional	Hardware resource on which this problem was detected.
CCR/NCR	Character	10	Optional	Identifier of a related CCR or NCR. If more than one, separate each by a space or semicolon for readability.
Duplicate of	Character	10	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).
Current Report	Selection	*	Optional	Submitter and date of an occurrence of the problem or issue. Helps browse through multiple reports of the same issue.

*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 8.3-4. Trouble Ticket TAB Description

Tab Name	Description
Detail	User populates details of defect and any additional occurrences of the same problem. Fields captured on this tab are *Found By* (*Submitter)*, *Date* of problem, *Version* of software found, and *Description* of concise details of the problem that an engineer is able to perform an analysis.
Workflow	This tab automatically captures identified series of events or activities associated with the trouble ticket's workflow. This listing is populated after the User continues to move through the lifecycle of the defect. Fields displayed data of this tab includes the flow of the *Event*, *Date* of change or input, *Who* activated the event, *Other Information* about the event.
Workaround	This tab captures User's description of how to workaround the bug or requested feature until it is fixed or implemented.
Source Code	Not in use
Notify	Optional
Links	Not in use
History	Captures and displays the defects workflow.

In addition to the fields described in the preceding tables, the TestTrack Pro provide a number of dialogue box options for the purpose of documenting notes, dates, and other related information. The collection of this information is to better clarify the workflow the defect's activity changes

8.3.1.1 Access the Trouble Ticket System

1 For Web access:

1. **Activate internet browser** i.e., IE, Netscape. Recommended browsers are Internet Explorer 6.0+, Netscape 7.0+ or Firefox 2.0+.
2. **Enable "pop-ups"** on browser, and then **enter URL**:
   - http://i4apl01.hitc.com:20071/ttweb/login.htm (testing only)
   - https://links.gsfc.nasa.gov:20072/ttweb/login.htm (external)
   - http://links.hitc.com:20071/ttweb/login.htm (internal)

2 For PC Client access:

1. Call the Help Desk to **install Client onto PC**
2. From **Start** menu, select (All) **Programs**,
3. Select **Seapine Software,** select **TestTrack Pro**.
4. Select **TestTrack Pro Client** [or double-click **TestTrack Pro Client icon** on the Desktop]

If using TTPro for the first time, the Edit TestTrack Pro Server GUI will appear so you can define a TTPro server connection (Figure 8.3-2). Your TTPro Administrator can help you set up the connection.

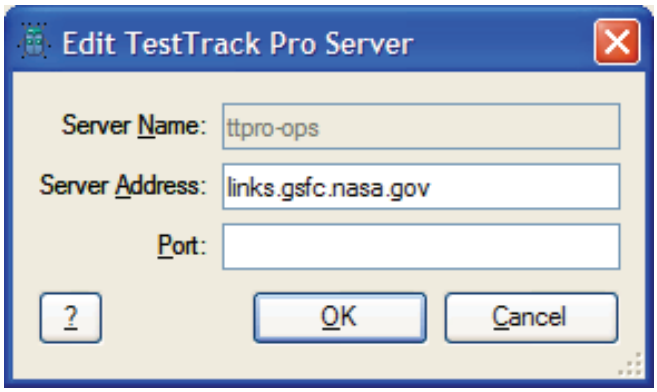

Figure 8.3-2. TestTrack Pro Log Server GUI

Table 8.3-5 provides a description of the Edit TestTrack Pro Server GUI's field.

Table 8.3-5. Edit TestTrack Pro Server GUI's Field Descriptions

Field Name	Data Type	Size	Entry	Description
Server Name	Character	> 200	Required	Name of TTPro server connection
Server Address	Character	> 200	Required	Fully qualified domain name of the TTPro server
Port	Integer	5	Required	Port on which TTPro clients communicate with the TTPro server

***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.

8.3.1.1.1 TestTrack Pro's Email Notification Features

The TestTrack Pro (TTPro) tool uses several email notification features to notify user of a defect's event/activity change. It allows automatic notification of defect changes be sent to selected recipients using the Notify tab, "Add Mail Recipients" dialog box. A general email can be sent from the main menu, using the "Send Mail" option to notify recipients of the defect or other general information. Defect details can be sent to the defect's founder using the "Reply to Submitter" Email, "Reply to submitter" option on the main menu. The system will automatically capture the data from the Detail Tab, description textbox. The email can also be modified.

8.3.2 Submitting a Trouble Ticket

When an operational issue or a problem in ECS is either found by or reported to User Services, follow the procedure applicable to your system, to create and log trouble tickets. Trouble tickets can be submitted into the Trouble Tickets main screen (Figure 8.3.-1) using the TTPro tool via accessing the web or via accessing the PC client.

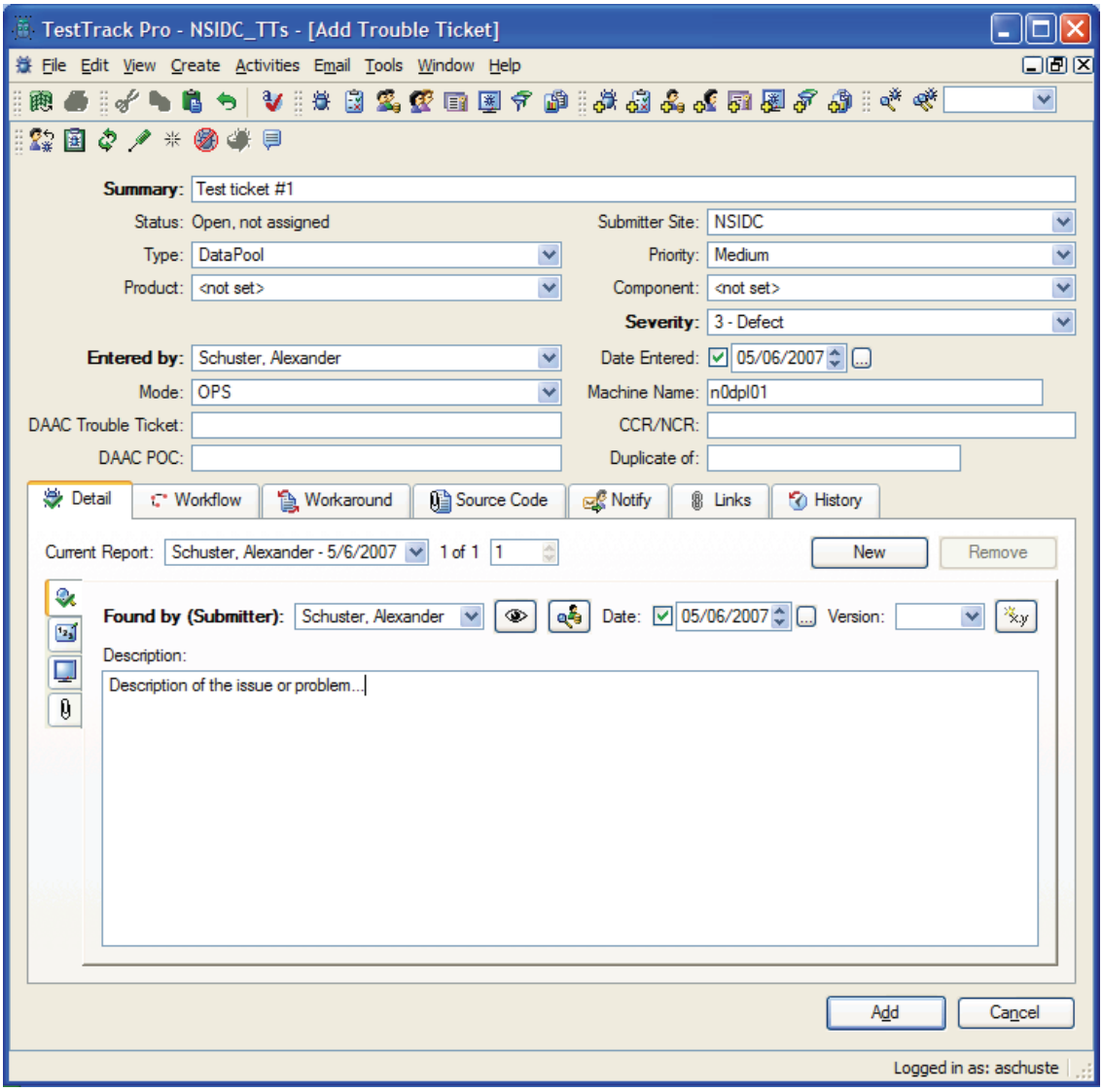

Figure 8.3-3. Add Trouble Ticket GUI

8.3.2.1 Submit Trouble Ticket

1 Click **Start** on taskbar, select (All) **Programs**, **Seapine Software, TestTrack Pro** to activate TestTrack Pro PC Client (or activate Web client).

2 Select **Server,** then **Project** from the Seapine TestTrack Pro Web Login GUI (Figure 8.3-4). Enter **User and Password**.

3 Select **Connect** – the selected Project's defect window displays.

4 From the **TestTrack Pro defect window**, from the main menu, select **File**, **New** (or click the "New" button) to activate a blank Trouble Ticket screen.

8.3.3 Reviewing and/or Modifying Open Trouble Tickets

Trouble tickets may be modified to better clarify the nature of problems defined and revised resolutions from the Maintenance Engineer investigations, Sustaining Engineering inputs, Developer inputs, Problem Review Board decisions, Change Control Board decisions, and/or Failure Review Board decisions. This screen will is used to advance the defect through its lifecycle states. The results will be factored into revisions and/ or additions to the Trouble Ticket log.

8.3.3.1 Review and/or Modify Open Trouble Ticket

1 **For Reviewing and Modifying Trouble Tickets via the Web:**

a) **Activate URL using internet browser** of TestTrack Pro Web Client. Trouble Tickets can be ***submitted, queried or modified***.

b) Select **Server** (as previously defined), then **Project** from the Seapine TestTrack Pro Web Login GUI (Figure 8.3-4). Enter **User and Password** information assigned by the System Administrator – initially setup as your UNIX login information. Then select where to **Start at – "Defect List",** which opens the List page and shows each Trouble Ticket's Identification, Short Description and Status.

c) Select **checkbox** next to the Trouble Ticket Id to get a more detailed description of that particular Trouble Ticket. Select "**View" or the "Edit"** button above selection to review or modify.

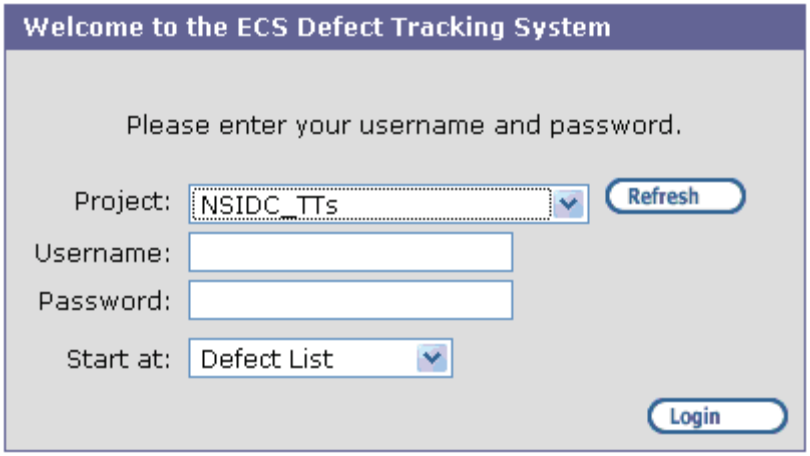

Figure 8.3-4. TestTrack Pro WEB Login GUI

2. For Reviewing and Modifying Trouble Tickets through PC Client:

a) Access **Seapine Software TestTrack Pro** application from the Start,

b) Select from the Seapine TestTrack Pro PC Client Login GUI (Figure 8.3-5) GUI: **Server**, **Project**, and enter **User and Password** information. Then click the **Connect** button to engage the "Defect List" window, which opens the List each Trouble Ticket's Identification, Short Description, and Status.

c) Select/Highlight the **Trouble Ticket Id** to get a more detailed description of that particular Trouble Ticket to review or modify. Select **"View" or the "Edit" button** above selection or double-click the on selection.

d) **Review and/or modify the Trouble Ticket fields** as necessary, **select tab options** such as Detail, Workflow, History, etc. Under the Workflow option, double-clicking the lifecycle state to update the existing date recorded in the activity dialog boxes.

e) Select **OK from the Trouble Ticket screen to commit update**.

NOTE: Clicking OK does not update the database within a dialog box or entry screen. The updates are captured/committed only when the Add/Edit screens are closed.

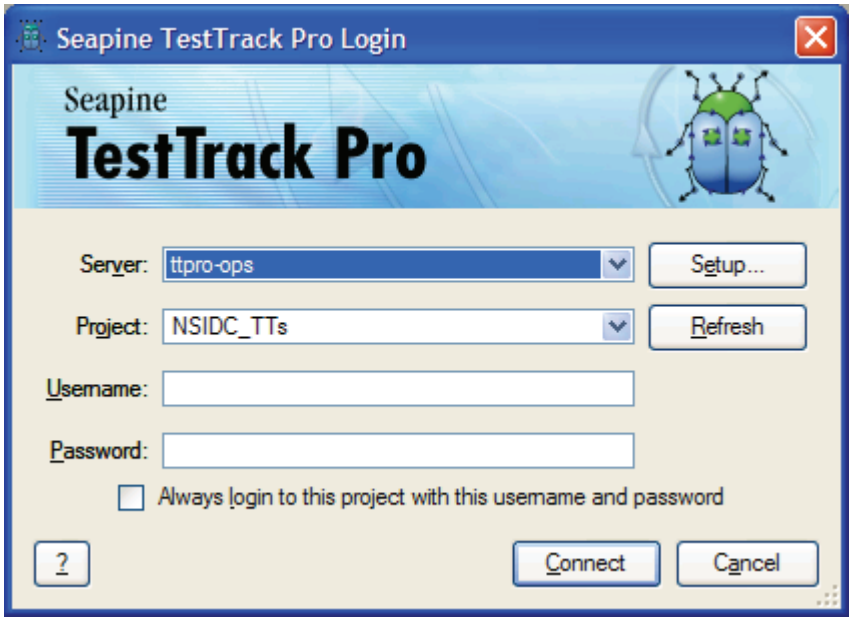

Figure 8.3-5. TestTrack Pro PC Client Login GUI

8.3.4 Moving Trouble Tickets Through Lifecycle State of Events

Trouble tickets (TTs) are transferred through the lifecycle using the Activities menu from the "Open" state to the "Closed" state (not necessarily in a sequential order). An example is the DAACs can forward an issue to the Landover's Problem Review Board (PRB) for advice or resolution. Using the "Escalate" event option, a trouble ticket is advanced to the "Forwarded: state for review to the EDF for review. Once the state of event is committed, automatic system notifications are generated to alert all relevant team members.

8.3.5 Adding Users to TestTrack Pro

The Trouble Ticket (TT) Administrator uses the TestTrack Pro (TT Pro) Security Groups screen to grant access to the tool. Security User's profiles are managed based on their defined roles and system privileges. Each project's security groups are defined by its local TT Administrator. Only one security group per project is granted or assigned to any one User. There are two license types that can be assigned to users:

1. The **Named License** is assigned to a user and is always reserved for that user. The TTPro System Administrator must have a Named license.
2. The **Floating Licenses** are not reserved to a single user; instead, they are available on a first-come, first-served basis. Each DAAC has a specified number of assigned Floating licenses. It is appropriate to assign Floating licenses to those who will need to modify Trouble Tickets (e.g., those who will be assigned to work and resolve problems).

The License Server Admin Utility is the gateway to the collection of screens for managing TTPro license server operations. Generally, access to the License Server Admin Utility screen (Figure 8.3-6) is limited to central TTPro System Administrators.

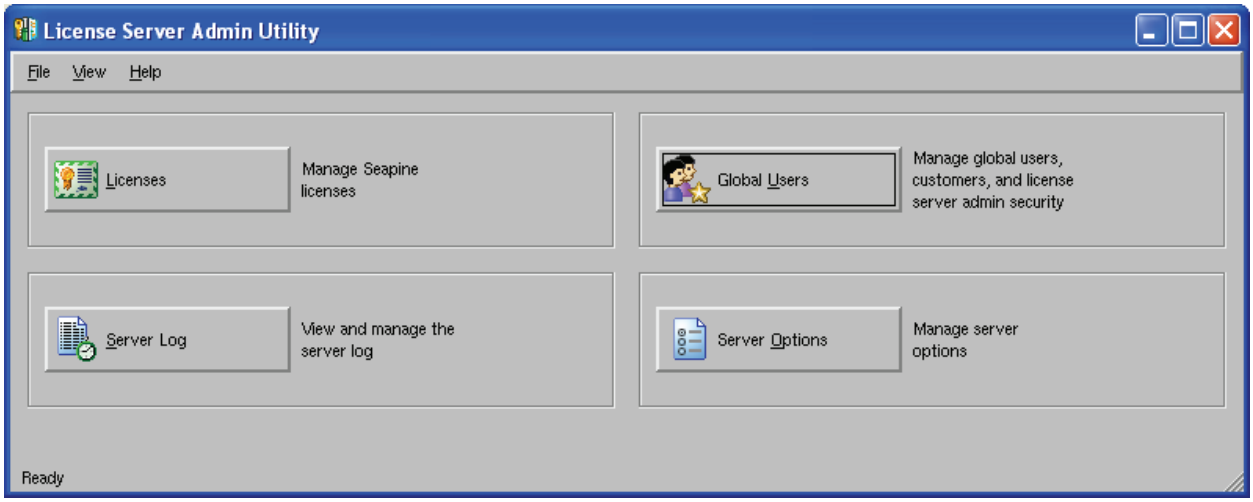

Figure 8.3-6. License Server Admin Utility GUI

8.3.5.1 Add Users to TestTrack Pro system

NOTE: The GUI can be started only from the command line on the TTPro server machine.

1 At the command line, **type <TTPro-root>/splicvr/bin &** to activate the License Server Admin Utility screen (Figure 8.3-6).

2 Click **Global Users button** to add (edit or delete) user profiles.

8.3.6 Privileges in TestTrack Pro

Changing and assigning privileges in TestTrack Pro (TTPro), or controlling privileges of those who have access to TTPro, is done by the TestTrack Pro System Administrator. There are numerous TTPro privilege groups for the system, and a change to the privileges of any group requires an approved Configuration Change Request (CCR). Access privileges determine which forms a user may open and specify whether a user may change information in a field or merely view it.

A user's privileges may be changed in two ways:

- Changing the group to which the user is assigned.
- Changing the access privileges of the group.

The **Security Groups GUI** is used to implement group assignment and to change the access privileges of a group. There are two primary ways to change the access privileges of a group. The use of the **Security Group** window and selecting one of the listed groups by:

- View… opens view of selected groups' privileges within the project.
- Add…opens view for adding a new group and its privileges to the project.
- Edit…opens view for updating selected groups' privileges within the project
- Delete…which deletes selected security groups. Users deleted from group(s) are no longer assigned to any project. They no longer have access to project(s) and can not receive project-issued email notifications.

One way to change access privileges is to use the **Users GUI** (Figure 8.3-7) to select which allows management profiles that define user's access to Trouble Tickets. **Users GUI** has no data entry fields and the following buttons are used to:

- View… opens view of selected user's profile within the project.
- Add…opens view for adding a new user profile to the project.
- Edit…opens view for updating selected user's profiles within the project.
- Delete…deletes selected user's profile. Deleting User's profile removes all references to that User from within the project's Trouble Tickets.
- Retrieve Global User – adds User to the project by retrieving the user's profile from the TTPro license server's global user records.
- Promote – adds selected local Users' profiles to TTPro license server's global user database.

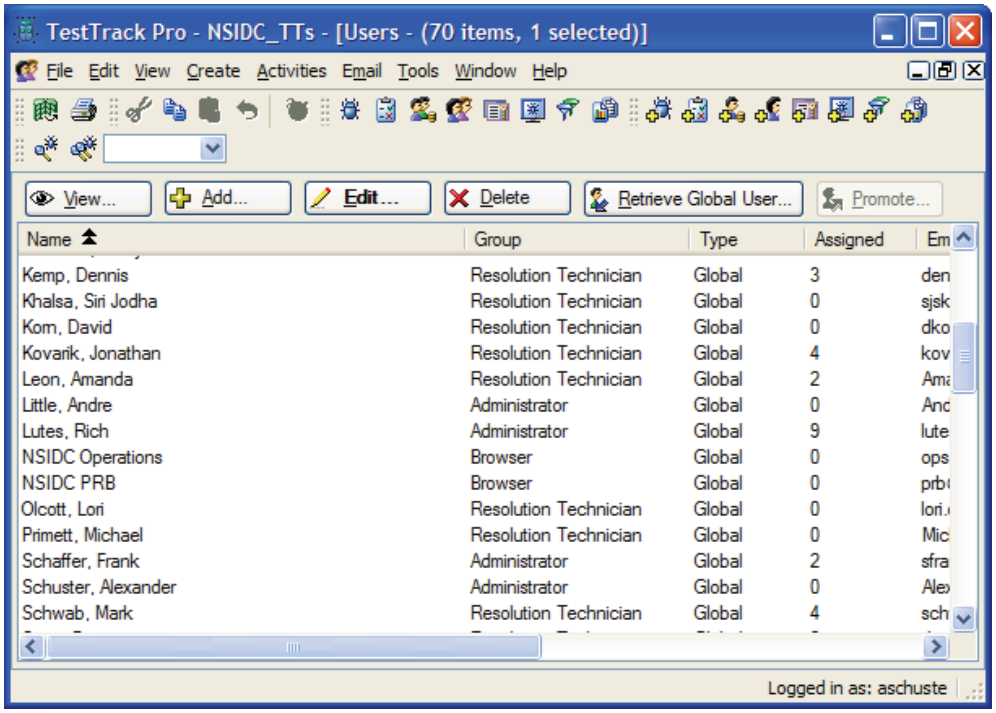

Figure 8.3-7. Users GUI

Groups have already been created to accommodate all privileges needed by TestTrack Pro Users for Release 7.20. These groups are identified in Table 8.3-6.

Table 8.3-6. Table of Access Control Groupings (1 of 2)

Groups	Description	Access Type
Operator	Operators have all permissions except for admin and event commands; cannot update closing code, assignments or fix event data; Submits trouble ticket internally.	Change/Global
Customer	Example of possible customer configurations	NA
User Services	Have all permissions except for admin commands and selected others considered admin commands; Submits trouble ticket internally for user.	Change/Global
DAAC HELP	Members can view escalated TTs and update the Forward checkbox in the Escalate event.	Change/Global
Inactive	This group is for inactive users so you do not have to delete the user. NO USER RIGHTS!	NA

Table 8.3-6. Table of Access Control Groupings (2 of 2)

Groups	Description	Access Type
Operations Supervisor	Have all permissions except for admin commands and selected otheres considered admin commands; Assigns problem priority and resolution responsibility. Can forward trouble ticket to another site.	Change/Global
Resource Manager	Have all permissions except for admin commands and selected others considered admin commands; Assigns problem priority and resolution responsibility. Can forward trouble ticket to another site.	Change/Global
Resolution Technician	Have all permissions except for admin commands and selected others considered admin commands; Attempts to resolve problem.	Change/Global
Problem Review Board Chair Person (or TT Review Board Chair)	Have all permissions except for admin commands and selected others considered admin commands; Reviews proposed solutions.	Change/Global
Administrator	Projects file administration. Adds groups and users. Changes permissions. Sets escalation times. Sets menu items. Etc.	Change/Global
Sub-Administrator	Same functions as Administrator but only with certain Schemas.	Change/Global
Browser	Read only permission. View records only, but can not update records.	Read/Global
Restricted View	Example of a user group using a filter to limit the viewing rights.	
Customize	Can use all features of the customize facility.	Change/Global
Submitter	Placeholder for anyone that submits a trouble ticket.	NA
Assignee	Place holder for anyone that is assigned a trouble ticket.	NA
Public	Read only permission. Guest users are automatically put in this group.	Read/Global
NotifyOpenEscal	Everyone that will be notified on an escalation due to trouble ticket being in “Open” status.	Read/Global
NotifyAssignEscal	Everyone that will be notified on an escalation due to trouble ticket being in “Assign” status.	Read/Global
NotifySolPropEscal	Everyone that will be notified on an escalation due to TT being in “Solution Proposed” status.	Read/Global
NotifyImpSolEscal	Everyone that will be notified on an escalation due to trouble ticket being in “Implement Solution” status.	Read /Global
NotifyImpSolEscal	Everyone that will be notified on an escalation due to TT being in “Implement Solution” status.	Read /Global

8.3.6.1 Modify TestTrack Pro User System Privileges

NOTE: The GUI can be started only from the command line on the TTPro server machine.

1 At the command line, **type <TTPro-root>/splicvr/bin &** to activate the License Server Admin Utility screen (Figure 8.3-6).

2 Click **Global Users button** to add (edit or delete) user profiles.

8.3.7 Modifying TestTrack Pro's Configuration

The System Administrator can configure the variables and/or environment of the TestTrack Pro Client to accommodate the site's user environment. There are several areas of configuration that can be utilized:

8.3.7.1 Modify TestTrack Pro System Configuration

To modify the TestTrack Pro environmental variables, refer to the TestTrack Pro operator's manuals or online help documentation at the company's web site

NOTE: No administrative configuration should be made without proper configuration change approval. Typically, usage of these options is managed by the Administrator.

1 **Modifying Setup <Field> Names:** To modify the TestTrack Pro field names, which is used to pre-define values that can be entered via pull-down menus relative to specific TTPro fields. These fields are list box selection populated for user selection when entering trouble ticket data.

1. To modify fieldname, from the Menu, select **Tools**
2. Click "**Configure List Values" <fieldname> Values** from dialog box.

2 **Configure System Notification Rules of Field names**: To modify the TestTrack Pro field notification rules, which is used for defining the conditions for automatically notifying uses about defect changes. Notifications can be issued for all or any subset of records, using a pre-defined or a custom email template within an accessibly authorized project.

1. To modify notification rules, from the Menu bar, **select Tools**
2. Select "**User Options**" to evoke the User Options dialog box.
3. Select Notifications from right pane menu, make changes

8.3.8 Generating Trouble Ticket Reports

A set of predefined reports is available in TestTrack Pro (TTPro). These reports are trouble ticket administrative reports generated for local and system-wide usage. There are several types

of predefined reports that can produce detail, list, distribution and trend reports of defects. Descriptions as defined in Table 8.3-7:

Table 8.3-7. Report Types in TestTrack Pro

Report Type	Report Description	When and Why Used
Detail – Display of Open Defects	A full report of every Trouble Ticket not in a Closed state, sorted by Trouble Ticket number.	When and if someone wants a copy of all open Trouble Tickets.
List – Summary of Problems	A list of the Trouble Tickets found or modified during the week prior to the report, containing only key details and sorted by Trouble Ticket number.	When and if someone wants a list of the Trouble Tickets opened or updated during the past week.
Distribution – Team Assignment Report	A distribution report identifying the Trouble Tickets found or modified during the week prior to the report, containing only key details and sorted by Trouble Ticket number.	When and if someone wants to know how evenly work is distributed among the staff.
Trend – Trend of Open Defects and Types	A trend report identifying the number of Trouble Tickets of each problem type in the Open state over time, grouped and ordered by month.	When and if someone wants to review (or forecast) trends among the types of problems reported.

Most often, users will select a system generated report from the list, using the **Report** option, from then **Create** option on the TestTrack Pro menu bar. If you need a custom report, it is possible to select alternate fields to customize a pre-defined report template or create/generate a new template.

8.3.8.1 Generate Reports

1 From the TestTrack Pro menu, select **Create**.

2 Select **Report**, the select **Detail** (or Distributed, List, Trend) to display of open TTs.

8.4 Emergency Fixes

Emergencies may be in real time with the understanding that the trouble ticket system must be brought up-to-date as soon as possible after implementing the repair. The example presented below, involves a hardware failure. The problem needs to be resolved quickly to bring a system back into operation. The resolution requires emergency replacement of a component that is of a later version than is contained in the original equipment. The scenario is summarized in Table 8.4-1.

Scenario —Example of an Emergency Change Procedure

It is 7:00 on a Saturday evening. The DAAC operator detects a problem with the automated tape library (ATL) and reports the problem to the trouble ticket system. The trouble ticket is routed to the System Administrator, who confirms that the system will not operate and notifies the site Maintenance Engineer. After running further diagnostics, the Maintenance Engineer reports the problem and symptoms to the OEM's maintenance desk. The original equipment manufacturer (OEM) maintenance representative arrives and concludes that a controller card has failed. The only card the OEM has immediately available is of a later version and no spares are available on site. It will be Monday at the earliest before a replacement board of the same revision level can be located. The site maintenance engineer reports this to the operations Crew Chief (i.e., shift leader) for a decision.

The DAAC cannot afford to have the ATL down until Monday. The Crew Chief calls the DAAC manager at home, apprises him of the situation, and obtains approval to replace the board with the later version if tests conclude that it works properly. The OEM's maintenance representative installs the board. The site's sustaining engineer tests the new controller board, finds that it works properly, and brings the ATL back on-line. The sustaining engineer updates the trouble ticket to document the configuration change and the authority for the change, and forwards it to the site CMA. The site Maintenance Engineer updates the property record with the model, version, and serial number of the new board.

The site CM Administrator reviews the trouble ticket, and presents it to the local CCB for approval. The CMA then updates the Baseline Manager with the new configuration and TT number authorizing the change. At this point, the site is operational at variance from the system baseline (i.e., site unique) and is at risk of losing maintenance support from OPS Deployment.

The site CMA forwards the trouble ticket to the SMC, presented to the PRB for priority review and is solved or translated into an NCR. The PRB reviews all emergency TTs to assess whether there may be impacts to the system and/or applicability to other sites. The SCDV CCB monitors all open NCR promotion and approves them for closure.

In the event that it is later discovered that the new version controller board has adverse impacts when operating in the system configuration, a board of the original version will have to be obtained to replace the newer version. In such cases, the action will be recorded on a new trouble ticket, citing the previous CCR.

Table 8.4-1 summarizes emergency procedures that might be taken during an after-hours, over-the-weekend emergency hardware failure.

Table 8.4-1. Example of Emergency Change Procedure

Operator/User	System
Operator prepares trouble ticket to report ATL controller failure.	Trouble ticket recorded.
System Administrator and Maintenance Engineer confirm ATL controller failure, call ATL maintenance vendor, report call and time in Trouble ticket.	Diagnosis and vendor call recorded in trouble ticket.
Maintenance vendor isolates failure to the controller card. The later version card is the only card available.	
Crew Chief notified of situation and decision needed to bring ATL up to full operating capability. Approves use of the newer version card, records decision in the trouble ticket, forwards trouble ticket to Sustaining Engineer.	
Maintenance vendor installs card, tests using hardware diagnostics. Crew Chief authorizes controller to be brought back on-line.	
Maintenance Engineer records card installation by model/version into the trouble ticket.	Trouble ticket action recorded.
Sustaining Engineer reads trouble ticket and prepares for discussion at 8:30 am meeting. Updates the TT.	Install action recorded in TT. TT routed to the CMA.
CMA updates site baseline, forwards TT to the CCB. When CCB approves the action, CMA forwards to SMC.	Site ATL baseline updated in Baseline Manager.
PRB reviews emergency NCR, checks for applicability to other sites.	

This page intentionally left blank.

9. Configuration Management Procedures

The prepared procedures are applicable to all EMD hardware, software, and firmware components of systems or subsystems developed and acquired by the EMD contract and/or delegated to configuration management control by the operational site-level organizations. The procedures are applicable to all items maintained by the EMD Sustaining Engineering Organization in support of EMD mission-specific projects and multiple mission-specific institutional facilities. The procedures are not applicable to those entities controlled by higher level ESDIS Project Office CM Plans. CM procedures already in place may be used by the contractor subject to direction from the Change Control Board (CCB) chairperson.

Some major features of the approach being described here include:

- Customers participate in establishing the procedures;
- The SCDV CCB performs a support role for ESDIS and its designated on-site CCBs by processing system-level CCRs & Trouble Tickets (also know as defects);
- Prioritization, automated tools, and procedures are used for handling change requests;
- Diverse/Strategic representation at hierarchical CCBs facilitates a path for speedy escalation/resolution of problems/issues;
- Local organizations have the needed autonomy to accomplish their mission with the minimum necessary outside intervention to promote timely resolution of local problems and enable timely production of data products;
- Proper use and deployment of CM database assets to support all CCBs allows management monitoring, control, and analysis of activities;
- Coordination with the Failure Review Board allows coordinated response to problems and filtering of prioritized issues; and
- Common CM tools will be used in all elements of the EMD Project during operations.

The procedures are organized into seven major sections that address the EMD system-level flow-down of procedures to the site-level which references applicable site-tailored procedures. The topics include:

- (Section 9.1) Configuration Identification
- (Section 9.2) Change Control Processes
- (Section 9.3) Configuration Status Accounting
- (Section 9.4) Configuration Audits
- (Sections 9.5 and 9.6) Software CM Manager (ClearCase)

- (Section 9.7) Baseline Manager (ClearCase).

9.1 Configuration Identification Procedure

9.1.1 Purpose

The purpose of configuration identification during maintenance and operations is to incrementally establish and maintain the definitive basis of control and status accounting for the EMD control items. To accomplish configuration identification for both hardware and software, the configuration management (CM) administrator (CMA) shall ensure the maintenance of each EMD configuration controlled item in an operational baseline by executing the following tasks:

a. Assign identifiers to configuration items (CIs) and their component parts and associated configuration documentation, including revision and version number where appropriate. Assign serial and lot numbers, as necessary, to establish the CI effectivity of each configuration of each item of hardware and software.

b. Follow EMD developer guidelines as referenced below in Section 9.1.3.

c. Follow vendor nomenclature for COTS items.

d. Apply operation and maintenance (O&M) version name extensions to EMD modified item nomenclature following the rules in Section 9.1.4.

e. Follow author-designated version control and nomenclature for documents and follow the EMD Library guidelines (cf. Chapter 20, *Library Administration*) administered by the EMD Librarian.

f. Support the EMD Librarian's efforts to maintain linkage of the EMD documentation to EMD configuration items in the Baseline Manager tool (cf. Section 9.7). Ensure that the marking and labeling of items and documentation with their applicable identifiers enables correlation between the item, configuration documentation, and other associated data.

g. Maintain a release system for configuration changes (cf. Section 9.2, *Configuration Change Control Procedures)*.

h. Maintain views of operational baselines using the Baseline Manager tool.

9.1.2 Applicability

All EMD CM Administrators and support personnel.

9.1.3 References

ESDIS CM Plan

Configuration Management Plan for the EMD Project	110-EMD-001
EMD Software Build Process	CM-1-045
Data Identification Numbering	DM-002

9.2 Configuration Change Control Procedures

9.2.1 Purpose

The ESDIS CCB chartered Change Control Boards (CCBs) shall apply configuration control measures to all the EMD configuration items and the associated documentation prior to the time it is baselined for operations. The CCBs shall apply configuration control measures to accomplish the following objectives:

a. Ensure effective control of all CIs and their approved documentation.

b. Provide effective means, as applicable, for (1) proposing engineering changes to CIs, (2) requesting deviations and waivers pertaining to such items, (3) preparing notices of revision, and (4) preparing Specification Change Notices.

c. Ensure the implementation of approved changes.

9.2.2 Applicability

All ESDIS chartered EMD CCBs.

9.2.3 References

ESDIS CM Plan

EMD Configuration Management Plan	110-EMD-001
CCB Change Control Process	CM-004

9.2.4 Procedures

9.2.4.1 Configuration Change Request Preparation

The Configuration Change Request (CCR) form in Figure 9.2-1 has been developed as a medium for the drafting of CCRs throughout the EMD Operations Deployment environment for changes processed locally at EMD site-level chartered CCBs at the SMC, EOC, and DAACs (GSFC, LaRC, LP DAAC and NSIDC). For changes processed by the ESDIS Change Control Board (CCB), use ESDIS own forms. For EMD CCBs, use the new EMD Electronic CCR Tool. There are numbered items on the form that correspond exactly to the data entry required to be performed by the respective originator of the CCR. Each CCB will have unique CCR identification sequence numbers. Each CCB can forward CCRs and reports from the Change Request Manager to SMC, which processes system-level CCRs for ESDIS CCB. The ESDIS CM Plan will determine the charter of the respective CCBs and thus the scope of CCR issues to be addressed by the site CCBs.

Earth Observing System Data and Information System (EOSDIS)
Core System (ECS)
Configuration Change Request (CCR)

1. **Configuration Control Board (CCB)**
ESDIS:___ ECS:___ SMC:___ DAAC: GSFC___, LaRC___, EDC___, NSIDC___,

2. CCR No.

3. Submitted Date:	4. Revision	5. Priority Emergency ☐ Urgent ☐ Routine ☐	6. Change Class	7. Status

8. CCR Title:

9. Originator: Org: e-mail: phone:

10. Approval:_____ _____
signature date

11. Reason for Change

(indicate attachment ___)

12. Description of Change

(indicate attachment ___)

13. Impact Analysis:
Cost: ☐ None ☐ Small (Not exceeding $100,000) ☐ Medium ($100,000 to $500,000) ☐ Large (Over $500,000)
Evaluation Engineer: Org: e-mail: phone:
Impact Evaluators: ESDIS___; ECS Dev___; SEO___; SMC___; DAACs: GSFC___, LaRC___, EDC___, NSIDC___, EOC___;
Others_____ (indicate attachment ___)

14. Comments: (Indicate Sites/ Organizations Affected)

(indicate attachment ___)

15. Board Action: ☐ Approved ☐ Withdrawn ☐ Disapproved ☐ Deferred Until_____
date
Further Action Required: ☐ ECP ☐ Waiver ☐ Deviation ☐ Tech Direction ☐ Contract Mod. ☐ DCN Other:_____

16. CCB Approval
Chair:_____ _____
signature date

17. CCR Implemented
CM Admin. signature:_____ date:_____

Figure 9.2-1. Configuration Change Request (CCR) Form

The following enumerated text corresponds to the numbered items on the CCR form:

(1) **Change Control Board (CCB) --** The designated CCB is checked-off for changes processed by the ESDIS Change Control Board (CCB) and its EMD site-level chartered CCBs at the SMC, EOC, and DAACs (GSFC, LaRC, ,LP DAAC and NSIDC).

(2) **CCR Number --** The unique serialized CCR number is applied at each site.

(3) **Submitted Date --** The date that the CCR was prepared is documented.

(4) **Revision --** The current revision is designated for tracking changed versions of original CCR.

(5) **Priority --** The priority level of the CCR is assigned. Emergency CCRs may have already been implemented on a temporary basis by the Trouble Ticket Review Board (TTRB) with concurrence from the CCB Chair who later receives the CCR to document /implement the permanent change. Urgent items will be reviewed by the next CCB meeting. Routine items will be reviewed as soon as the schedule permits.

(6) **Change Class --** Change Classes are either I or II. Class I will be handled by ESDIS-only because of cost, schedule, and/or mission impacts that may require requirements changes. Class II items do not affect mission requirements, but may have cost and/or schedule implications which affect maintenance, operations, procedures, documentation, site-tailored items, COTS implementation, site installations of core system changes, science SW changes, etc.

(7) **Status --** Table 9.2-1 is a summary of the workflow of events (Figure 9.2-2) of the CCR states as defined in the TestTrack Pro defect tracking database tool (which implements the Change Request Manager). Note that the hard copy form will not be updated chronically, but will be kept in the master suspense file of the CM Administrator until stamped closed-out (item #7 & 15) and appropriate signatures (viz., items 16, & 17) are acquired.

Table 9.2-1. State Table Composition in TestTrack Pro Format

State	**Event**
Open	Open – Initial Escalation from DAACs
Assigned	Assign
Estimate	Estimate
Start Work	Start Work
Fix	Fix
Merged	Merge
Tested (Integration)	Test
Verified (PVC/DAAC)	Verify (PVC/DAAC)
Verified (Failed) as Open	Assign, Open, Defer, Close
Verified (PVC/DAAC) as Closed	Close, Deliver

Transition (NCR State)	Transition (NCR Event)	Action or Allowed Next Events
Pending (PRB review) ⇒ **Open**	Initial state of NCR escalated from DAACs– Open	Perform initial review
Open ⇒ **Assigned**	Assign	Problem assigned to engineer(s) for technical assessment; cost/schedule/impact
Open (assigned) ⇒ **Estimate**	Estimate	Determined/Enter workaround information, if applicable; record cost, schedule and impact information; Technical Assessment Approval
Open (assigned) ⇒ **Start Work**	Start Work	Work on fix has begun
Open (assigned) ⇒ **Fix**	Fix	Fix has been built, installed and is ready for testing
Fixed ⇒ **Merged**	Merge	After merge approved, maintenance engineer merge code (to Baseline(s) 1, 2, 3)
Merged ⇒ **Tested (integration)**	Test	Product has undergone integration testing in test environment (PVC, VATC, EDF); ready for verification testing
Tested (integration) **Verified (PVC/DAAC)**	Verify (PVC/DAAC)	Fix was verified by Test; added test procedure (results of a fix verification); Quality verification of fix; delivery and closure recommended
Verified (Failed) ⇒ ***Open***	*Assign, open, defer, close*	*Fix failed verification testing and can be re-worked, deferred, or closed*
Verified (PVC/DAAC) ⇒ **Closed**	Close, Deliver	With PRB/DAAC Approval –directed that problem be closed

Activities
- Open...
- Assign...
- Estimate...
- Start Work...
- Defer...
- Fix...
- Merge...
- Deliver...
- Test...
- Verify (PVC)...
- Verify (DAAC)...
- Verify (Failed)...
- Close...
- Duplicate...
- Escalate...
- DDTS History...
- New Reported By...
- Delete Reported By
- Assign NCR Numbers
- Renumber NCRs...
- Bulk Field Changes...
- Add Link...
- Diagram Links...
- Override Workflow

Figure 9.2-2. TestTrack Pro Workflow of Events

Explanations of DDTS' State Table Content:

Uppercase character in the 1st column is the character stored in the change request record to indicate what state a change request is in. DDTS uses this character to go into the table and extract the descriptive name for display in reports.

Names in the second column are the state in the present tense. They are shown on the DDTS list of states that are available for selection during input. It's also used by some of DDTS' query and report code. It facilitates querying based on descriptive names opposed to a single letter.

Names in the third column are the state in the past tense. They are shown on the DDTS change request record. It's also used by some of DDTS query and report code. It facilitates querying based on descriptive names opposed to a single letter.

Definitions (a state is the stage that a proposed change has reached in its life cycle.):

New - the initial state for all newly entered change requests.

Assign-Eval- state entered when the change request is being assigned to an engineer for evaluation/analysis.

Assign-Implement- state entered when the change request is being assigned to an engineer for development.

Implement-state entered when the proposed change has been developed.

Assign-Verify-state entered when the developed change is being assigned to an engineer for verification testing.

Verify- state entered when a developed change has been tested and verified that it functions properly.

Close- state entered when all activity specified in the change request has been completed or that the approval authority has decided to close it prior to completion of all activity.

Duplicate-state entered when a change request is determined to be a duplicate of an existing change request. Duplicate change request identifies change request being duplicated.

Forward-state entered when a change request needs to be forwarded to another DDTS defined project (In DDTS terminology, a project is a grouping of change requests. For example, a change request from a site project can be forwarded to an EMD project.).

Defer- state entered when activity on a change proposal has to be postponed.

(8) **CCR Title --** The CCR title is supplied by the originator.

(9) **Originator --** The originator name, organization, e-mail address, and phone number is given.

(10) **Approval --** The CCR is approved by the designated management authority that is assigned by the CCB. This sponsorship requirement acts as a primary filter to eliminate from consideration those CCRs that cannot be implemented or which have no EMD site management support.

(11) **Reason for Change --** The reason for the change is narrated on the form and/or the designated attachment.

(12) **Description of Change --** The proposed implementation of the change is narrated along with any known impacts, resources, and expenses to be incurred.

(13) **Impact Analysis --** Impact analysis is documented in the form of Figure 9.2-3. The impact analysis is collected by the CCB Chair appointed Evaluation Engineer in coordination with the CM Administrator who maintains the CM records and assembles the review package for the CCB. The Evaluation Engineer documents the list of Impact Evaluators and derives and/ or verifies cost, technical, and schedule impact of the proposed change based on all inputs received. The results of the coordinated CCR Impact Analysis inputs are presented in the CCR Impact Summary form shown in Figure 9.2-3 part of the CCR review package.

(14) **Comments** -- Comments are added to the CCR to summarize sites and/ or organizations affected by the CCR. Additional comments may address proposed CCB dispositions and recommendations to be indicated by resolutions in item #15.

(15) **Board Action --** CCB actions and follow-up actions that will be facilitated and tracked by the CM Administrator are indicated. Possible CCB dispositions are given as

approved, withdrawn, disapproved, and deferred (pending follow-up activities by the indicated schedule date). Further actions are indicated as:

Engineering Change Proposal (ECP)- changed scope of contract requirements

Waiver-- declaration that certain contract requirements no longer apply

Deviation-- change of contract terms or substitution of terms or deliverable requirements

Technical Direction-- order by Contracting Officer's Technical Representative (COTR) to perform certain tasks within the scope of the contract

Contract Modification-- changes to the terms of a contract

Document Change Notice (DCN)- notification of changes to published documents

Others-- Engineering Change Notice, Change Order, Escalate to higher CCB authority, etc.

(16) **CCB Approval --** CCB approval signature authority by CCB Chair or designate.

(17) **CCR Implemented --** This signature and close-out stamp (item #7) are executed by the CM Administrator witnessing the completion of the CCR implementation process which is tracked in the Change Request Manager automated tool DDTS and updated in Baseline Manager for affected version control status changes.

CCR Impact Analysis

Responder Request Number: ____
Responder:_____
Responder Point of Contact:
address:_____
_____
phone:_____
e-mail:_____
CCB Schedule Date:_____
CCR Number:_____
CCR Log Date:_____
CCR Originator:_____
CCR Originator Point of Contact:
address:_____
_____
phone:_____
e-mail:_____

Evaluation Engineer:_____
Evaluation Engineer Point of Contact:
address:_____
_____
phone:_____
e-mail:_____
Requested Return Date:_____

Rough Order of Magnitude (ROM) Impact Analysis

Basis of Estimate:

Technical Assumptions and Comments:

Cost Impact:
None [ ]
Small [ ] < $ 100, 000
Medium [ ] $100,000 < x < $500,000
Large [ ] > $500,000

Schedule Impact:

Technical Assessment: (Your impact analysis should consider the implementation approach; interfaces affected; HW or SW changes required; documentation changes required-- change from/to pages; suggested alternatives, if any; and impact to security features. If your system is not impacted, please provide that information to the CM Administrator.)

Comments:

Signed:_____
(Responder)

Date:_____

Figure 9.2-3. EMD CCR Impact Analysis (1 of 2)

CCR Impact Summary

Evaluation Engineer:_____
Evaluation Engineer Point of Contact:
address:_____
_____
phone:_____
e-mail:_____
CCR Board Date:_____

Resources Summarized:

Technical Summary:

ROM Summary (BOE, Cost, and Schedule):

Recommendation:

Signed:_____
(Evaluator)

Date:_____

Figure 9.2-3. EMD CCR Impact Summary (2 of 2)

9.2.4.2 Change Control Board Process (System and Site-level CCBs)

The EMD organization will provide administrative and technical support services for the CCB at each site. Each site's CCB is controlled by the host site organization and provides the authority and direction for the EMD contractor to modify the operational baseline. The ESDIS CCB has chartered an EMD Review Board to coordinate EMD system-level changes and problem management via the EMD contractor and on-site Review Boards that also act as site CCBs. This is illustrated using the CM Administrator's workflow for the sustaining engineering support of the EMD Review Board in Figure 9.2-4 and the On-Site CM Administrator's workflow for sustaining engineering support of the on-site CCB in Figure 9.2-5. The problem management process was discussed in detail in Chapter 8 of this document. Both diagrams illustrate the flow of CCRs through the respective CCBs with inputs from the review boards and evaluators that determine the disposition of proposed changes. Details of this process are given below:

System-level Change Control Procedures

(The enumeration corresponds to the diagram of Figure 9.2-4)

(1) Configuration Change Requests are received by the EMD CM Administrator from all sources with regard to the operational EMD Core System as described in Section 9.2.4.1. These changes designated as from other sources could involve system enhancements, procedures, interfaces (both external and internal), documentation changes, etc. that are not the subject of contemporaneous problem reports which would be first deliberated by the Trouble Ticket Review Board (TTRB) and/or Failure Review Board as explained below.

(2) Proposed common baseline changes will be proposed based on Trouble Ticket (TT) resolutions obtained from the respective review boards (see Chapter 8 for details). The respective TT would be closed via a corresponding CCR to either ratify, i.e., to make permanent the prior temporary/emergency action taken by the TTRB or to consider normal priority (scheduled) changes for incorporation into future change releases.

(3) The EMD CM Administrator is responsible for logging the CCR into the Change Request Manager.

(4) The CCB chair assigns an evaluator and the EMD CM Administrator coordinates impact assessment.

(5) Class I change requests (proposed changes that affect controlled milestones, schedules, budget, cost and requirements) are forwarded to the ESDIS CCB for consideration with recommendations from the EMD Review Board.

(6) Class II change requests (proposed changes affecting documentation, hardware [alternative use of], software [correction of errors], and COTS substitution without a Class I impact) are considered by EMD Review Board deliberations.

(7) Notice of proposed changes is distributed to affected parties and review board members to obtain and coordinate impact assessment and optimize the approach to implement proposed changes.

(8) The results of EMD Review Board deliberations are factored into review board resolutions which determine whether, when, or where the system changes will be implemented.

(9) Approved changes are processed by the EMD CM Administrator to the support activities, i.e., site CCBs, support personnel (EMD), vendors, etc. who are provided with change orders, schedule, and implementation instructions.

(10) Disapproved changes are processed by the EMD CM Administrator via official notifications, memo to the file, and update of the Change Request Manager (CRM).

(11) The EMD CM Administrator tracks implementation and closure of CCRs via directions to implementing organizations and their acknowledgements using the CRM tracking and statusing features.

(12) New versions and/or maintenance updates are annotated in Baseline Manager at the SMC and at the affected sites by following the procedures for configuration identification, activation dates, deactivations dates, and issuing version description documents.

(13) Simultaneously, the SW Change Manager (ClearCase) is updated with directory trees, installation files, and software as required by SW maintenance.

(14) Status of this activity to implement changes and assigned responsibilities is tracked through closure in the CRM at SMC and at the sites.

(15) The databases are synchronized by manual checking between applications Baseline Manager vs. CRM vs. SW Change Manager) and automated verification by the SW CM Manager for purposes of SW distribution and maintenance.

(16) The TT Review Board is empowered to make emergency fixes without common baseline changes and update these changes directly to Baseline Manager with documentation to follow via the CCR submitted to the appropriate CCB. Proposed common baseline changes must be submitted by CCR to the EMD Review Board.

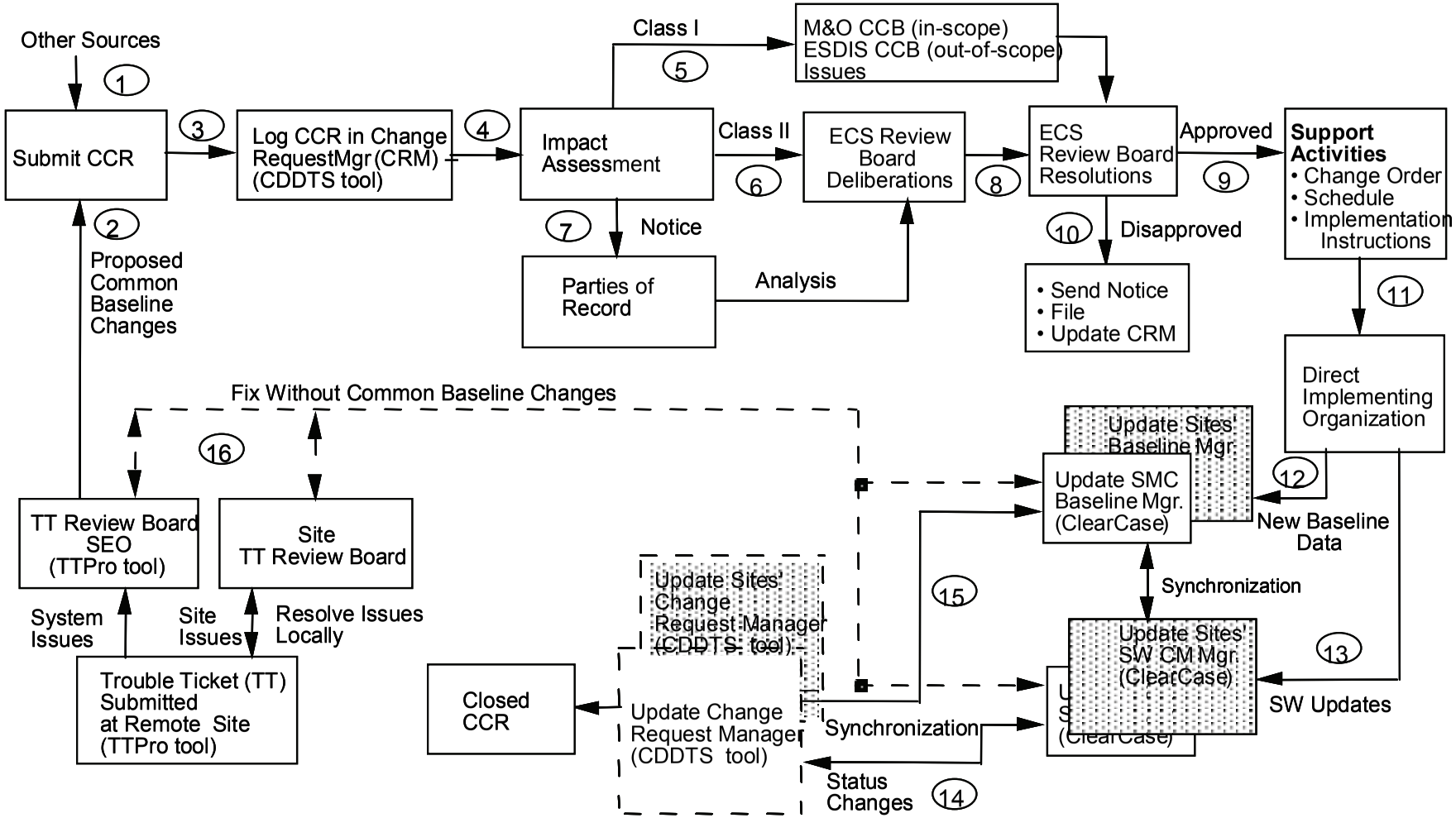

Figure 9.2-4. Workflow Diagram for EMD CM Administrator

Site-level Change Control Procedures

(The enumeration corresponds to the diagram of Figure 9.2-5)

(1) Configuration Change Requests are received by the Site CM Administrator from all sources with regard to the **site unique extensions** to the operational EMD Core System as described in Section 9.2.4.1. These changes designated as from other sources could involve system enhancements, procedures, interfaces (both external and internal), documentation changes, etc. that are not the subject of contemporaneous problem reports which would be first deliberated by the Site/ EMD Trouble Ticket Review Board (TTRB) and/or Failure Review Board as explained below.

(2) Proposed site baseline changes will be proposed based on Trouble Ticket (TT) resolutions obtained from the respective review boards (see Chapter 8 for details). The respective TT would be closed via a corresponding CCR to either ratify, i.e., to make permanent the prior temporary/emergency action taken by the TTRB or to consider normal priority (scheduled) changes for incorporation into future change releases.

(3) The Site CM Administrator is responsible for logging the CCR into the Change Request Manager.

(4) The CCB chair assigns an evaluator and the Site CM Administrator coordinates impact assessment.

(5) Class I/System Issues change requests (proposed changes that affect controlled milestones, schedules, budget, cost and requirements) are forwarded to the EMD Review Board for consideration with recommendations from the Site CCB. Class I issues are further forwarded with recommendations by the EMD Review Board to the SCDV CCB for in-scope issues and to the ESDIS CCB for consideration of out-of scope issues with respect to the SOW of the EMD Contract.

(6) Class II change requests (proposed changes affecting documentation, hardware [alternative use of], software [correction of errors], and COTS substitution without a Class I impact) are considered by Site CCB deliberations.

(7) Notice of proposed changes is distributed to affected parties and review board members to obtain and coordinate impact assessment and optimize the approach to implement proposed changes.

(8) The results of Site CCB deliberations are factored into CCB resolutions which determine whether, when, or where the system changes will be implemented.

(9) Approved changes are processed by the Site CM Administrator to the support activities, i.e., other CCBs, support personnel vendors, etc. who are provided with change orders, schedule, and implementation instructions.

(10) Disapproved changes are processed by the Site CM Administrator via official notifications, memo to the file, and update of the Change Request Manager (CRM).

(11) The Site CM Administrator tracks implementation and closure of CCRs via directions to implementing organizations and their acknowledgements using the CRM tracking and statusing features.

(12) New versions and/ or maintenance updates are annotated in Baseline Manager at the affected sites and the SMC by following the procedures for configuration identification, activation dates, deactivations dates, and issuing version description documents.

(13) Simultaneously, the SW Change Manager (ClearCase) is updated with directory trees, installation files, and software as required by SW maintenance.

(14) Status of this activity to implement changes and assigned responsibilities is tracked through closure in the CRM at the sites.

(15) The databases are synchronized by manual checking between applications Baseline Manager vs. CRM vs. SW Change Manager) and automated verification by the SW CM Manager for purposes of SW distribution and maintenance.

(16) The on-site TT Review Board is empowered to make emergency fixes without common baseline changes and update these changes directly to Baseline Manager with documentation to follow via the CCR submitted to the appropriate CCB. Proposed common baseline changes must be submitted by CCR to the EMD Review Board.

Each site's CCB accepts initial release or updates from the ESDIS CCB. Similarly, the Distributed Active Archive Center (DAAC) CCBs will accept product generation software from an ESDIS authority. Local tailoring and installation decisions are determined by the site CCB.

In the case of Evaluation Package (EP) and Prototype deliveries, the EMD CCB as directed by ESDIS will provide a configured, documented, executable with supporting files. Again, installation decisions are determined by the site CCB.

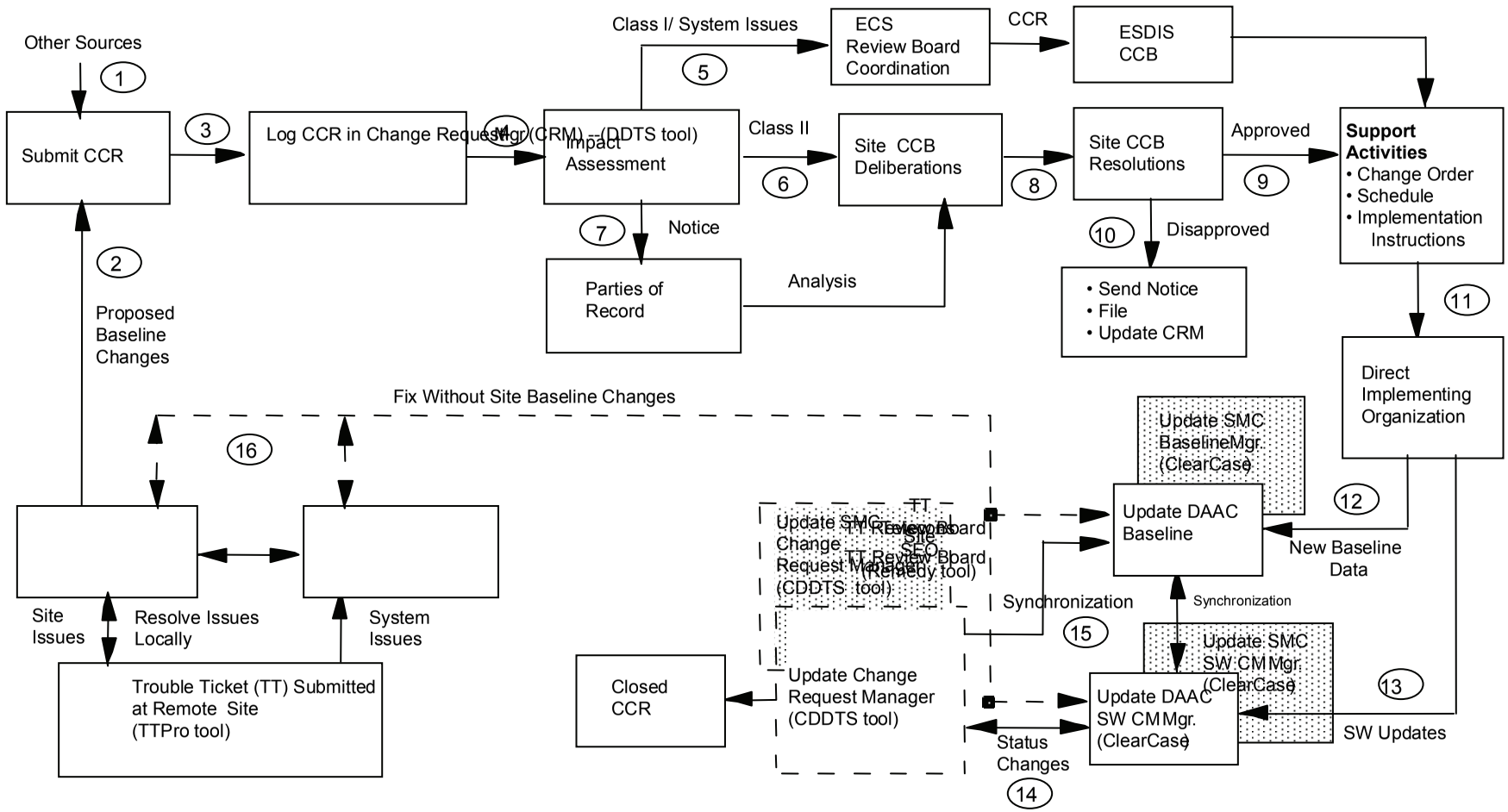

Figure 9.2-5. Workflow Diagram for Site-level CM Administrator

Each Science Computing Facility (SCF) is assumed to have a configuration control function. For commonality with other sites, it is assumed that this function will be performed by a CCB. A major difference, though, is the EMD contractor does not have an active role in the support of this CCB.

The SCF CCB will provide two types of configuration control:

(1) Configuration control of software and databases that are to be executed in another site's environment. In this mode, it operates very much as the EMD CCB does to establish a product baseline.

(2) Configuration control of SCF resources that are made available to the EOSDIS community. In this mode, its functions are the same as a DAAC CCB.

The EMD CM function at each DAAC will accept science SW and data items from the SCF CCB. These items will be incorporated into the DAAC's operational baseline as directed by the DAAC CCB.

The EOC CCB will control the operational configuration of the required EOC operational baseline. EMD CM will provide services as directed by the CCB.

The EMD Review Board will be charged with the responsibility for centralized coordination and control of EMD CM activities to ensure:

- EMD integrity and quality of service
- Successful coordination with both internal and external networks, systems, and on-site facilities
- Timely ESDIS CCB visibility into and oversight of EMD operations
- Convenient user administrative services

9.3 Configuration Status Accounting Procedures

9.3.1 Purpose

Operational phase configuration status accounting (CSA) consists of recording and reporting information about the configuration status of the EMD Project's documentation, hardware and software throughout the Project life cycle. Periodic and ad hoc reports keep ESDIS informed of configuration status as the operational mission evolves. Reports to support reviews and audits will be extracted as needed starting from the RRR.

The Baseline Manager tool described in Section 9.7 records and tracks as-built products designated as EMD control items (i.e., custom, COTS, science, toolkit, etc.) SW and HW items along with their associated documentation and records) and historical versions of EMD operational configurations. Baseline Manager, which is updated with the acceptance tested version of the EMD baseline at RRR, records and reports EMD document change status and histories, mission milestone baselines, and change status.

CSA entails maintaining version histories of delivered and maintained products as well as histories of operational baselines and changes made to each baseline. Additionally, CSA tracks the status of proposed changes from initial CCR submission to ultimate disposition and/or implementation. CSA also maintains historical records of CCRs.

9.3.2 Applicability

All ESDIS chartered CCBs

9.3.3 References

ESDIS CM Plan

EMD Configuration Management Plan 110-EMD-001

9.3.4 Procedures

The following are topical items subject to periodic or ad hoc reporting on behalf of the respective CCB or a system-level summary of information that will be reported by the CM Administrator representing the operational baseline for all the EMD sites.

(a) **New CCRs and Revisions.** This is a standard Change Request Manager report. This report will be issued monthly and summarized annually.

(b) **CCB Review.** Distribute CCR copies for review (and Impact Analysis forms if applicable). Print the agenda and distribute prior to the meeting.

(c) **Open Action Items.** Open action items should be statused regularly between meetings.

(d) **CCB Meeting.** Record the CCB's disposition of each CCR.

(e) **Record Action Items.** Record actions, assignments, and due dates.

(f) **CM Librarian Maintained Document Changes.** When all authorized document changes have been accomplished prepares DCN, post the final version on the EMD Document Data Server and distribute hardcopy as required.

(g) **Minutes Distribution.** Distribute minutes to the standard distribution and inform actionees of assigned action items.

(h) **CCR Implementation Status.**

- After CCB disposition, regularly status opens CCRs until closure.
- Class I events include: CCR to EMD Review Board for review/approval; Technical Review Board; and ESDIS Disposition
- Further events are as follows for implementation status: Consent Obtained; Item Received; Installed; Document Completed; etc.
- CCR CLOSED: A Class I CCR is not closed until the ESDIS contract officer's authorization is received or the reference CCR has been withdrawn.

- Class II document change CCRs may be closed with the CM Administrator's issuance of the DCN.
- Other non-document change CCRs may be closed when the originator verifies to the CM Administrator that all specified changes have been implemented.

9.4 Configuration Audits

9.4.1 Purpose

EMD will support Functional Configuration Audit /Physical Configuration Audit (FCA/PCA). EMD will also support audits by ESDIS and our own Quality Office functions. Internal CM self-audits will be conducted by EMD. Self-audits evaluate the Project's compliance with the Configuration Management Plan for the EMD Project and the ESDIS CMP. The CM self-audits will verify:

- That CM policies, procedures, and practices are being followed.
- That approved changes to documentation, and to software and hardware products are properly implemented.
- That the baseline documentation of each CI agrees with the as-deployed configuration and that adequate records of differences are available at all times.

A post-audit report is written outlining the specific items audited, audit findings, and corrective actions to be taken. All action items are tracked to closure.

In addition, EMD supports formal audits scheduled and conducted by ESDIS. These audits are conducted to validate that each EMD CI is in conformance with its functional and performance requirements defined in the technical documentation. The audits validate that:

- The as-built configuration compares directly with the documented configuration identification represented by the detailed CI specifications.
- Test results verify that each EMD product meets its specified performance requirements to the extent determinable by testing.
- The as-built configuration being shipped compares with the final tested configuration. Any differences between the audited configuration and the final tested configuration are documented.
- When not verified by test, the compatibility of EMD products with interfacing products or equipment is established by comparison of documentation with the interface specifications that apply.
- COTS products are included in audits as integral parts of the EMD baseline.

9.4.2 Applicability

All ESDIS chartered CCBs

9.4.3 References

Configuration Management Plan for the EMD Project 110-EMD-001

9.4.4 Procedures

The audits are standardized for a limited set of issues that drive the process for which the audit is taken, FCA/PCA, Security Issues, General Accounting, Test Readiness Review, or Operational Certifications. The baseline references for the audits will be maintained by the ClearCase Baseline Manager tool (cf. Section 9.7). Release Notes documents will be used to document auditable changes to configured articles that are issued at the EMD configuration item (CI) level. Audit processes, including Project Instruction documentation, are discussed in the *Configuration Management Plan for the EMD Project*, 110-EMD-001.

The release notes will contain the prioritized current status summary of any Trouble Tickets/Discrepancy Reports against the CI that is being issued per the change request.

Some general guidelines and/or items that must be tailored for the specific size and scope of configuration audit to be conducted include:

(a) Audit Plan;

(b) Conference Agenda;

(c) Location to collect and analyze data; conduct meetings;

(d) Applicable specifications, drawings, manuals, schedules, design and test data;

(e) Test Results Analysis;

(f) Meeting minutes including resulting audit action items;

(g) Tools and inspection equipment necessary for evaluation and verification;

(h) Unencumbered access to the areas and facilities of incoming inspection, fabrication, production, and testing;

(i) Personnel from each engineering, production, and quality department to be available for discussion of their respective areas;

(j) Copies of inspection reports, process sheets, data sheets, and other documentation deemed necessary by the Government FCA/ PCA teams; and

(k) Isolation of the item(s) and detailed parts to be reviewed.

9.5 Archiving Procedures for the SW CM Manager (ClearCase)

9.5.1 Purpose

This section details the concerns and procedures used by Configuration Management for the backup of ClearCase Version Object Base, public storage area for files (VOB) data.

9.5.2 Applicability

All EMD CM Administrators, SW Maintenance Engineers, and Sustaining Engineers

9.5.3 References

EMD Configuration Management Plan 110-EMD-001

9.5.4 Definitions

Build - an assemblage of threads to produce a gradual buildup of system capabilities.

EMD Development Facility (EDF) - the software development environment including data, hardware, software, networks, facilities and procedures used to support EMD software development and testing.

Software - for the purposes of this instruction, software includes all EMD-developed application software, COTS software, build and environmental instructions, and databases used in the execution of these products.

System-level - for the purpose of this instruction, system-level includes all EMD integration and test activities beginning with installation of software.

Software Development File - a repository for a collection of material pertinent to the development or support of software.

Thread - a set of components (software, hardware, and data) and operational procedures that implement a scenario, portion of a scenario, or multiple scenarios.

View - a unique workspace (operator private storage) management or "sandbox management" that provides developers with transparent, file-level access to any version of any element through the use of dynamically-evaluated, user-specified version selection rules.

VOB - Versioned Object Base. Secure, permanent, virtual file system that is mountable. Repository for public area storage of version-controlled data/files.

9.5.5 General

1. IT Infrastructure maintains a backup for all EMD systems. This procedure documents the additional steps CM has taken to ensure that the Infrastructure backups include the nightly backup of the ClearCase Versioned Object Base (VOB) Data. These VOBs store all versions of all of all custom software developed for the EMD project. In the event of a catastrophic failure, everything can be restored from the VOB data.
2. There is a cron job that runs on each VOB server (sweat, tears, nelix, janeway and triton) at 11:55 pm as the user id "vobadm" to lock down the individual VOBs (necessary to flush the database caches back to disk, in order to obtain a valid file for backup), tar up important VOB storage directories (the entire source pool, and the local copies of the VOB storage directories, which includes the VOB databases), and copy that data across

the network to a staging area (located in /tools/ccbackup) along with log files of the backup process. This process takes approximately 55 minutes.

3. The staging area is then backed up by IT Infrastructure using their standard backup procedures. IT Infrastructure maintains it's own process of offsite storage for the EMD systems backups.
4. For EMD Evolution, the ClearCase view servers and VOB servers have been moved from Solaris 8 hosts to Linux hosts. A noted benefit of the porting of ClearCase has been speed. ClearCase functions now execute between 40-80 times faster, due to the host network interface improvements (x10) and the much faster processors. This primarily covers compilations and file preparations.

9.5.6 Procedures

No additional procedures are needed on Configuration Management's part.

9.6 Software Delivery and Installation

9.6.1 Purpose

This section describes the delivery of Sustaining Engineer Organization-developed software from the Landover facility to a remote site (a DAAC), and subsequent installation of the software onto target hosts, in accordance with configuration management controlled processes. The process begins with an approved CCR. The CCR precisely defines the software to be released to the remote sites. Software is prepared and delivered using the *DeliveryTool*. The *DeliveryTool* is a custom ClearCase tool that was developed to ensure accurate and controlled release of software. Software installation is controlled by each sites' Configuration Change Board (CCB). Landover CSG prepares and releases the software to the DAACs, but the DAACs control the installation of the software into their operational modes.

9.6.2 Applicability

All EMD sites' Sustaining Engineers, System Administrators, CM Administrators, and Maintenance Engineers

9.6.3 References

Configuration Management Plan for the EMD Project	110-EMD-001
COTS and Custom Software Preparation and Delivery	CM-1-032-1

9.6.4 Procedures

9.6.4.1 Overview

This section describes the release of CM controlled software under the authority of the SCDV CCB's approved CCRs.

Assumptions:

- CM maintains records of software preparations and deliveries.
- Baseline records are maintained in the Baseline Manager (ClearCase tool)
- Electronic Delivery is performed via Secure Copy (scp) to target DAACs.
- Resource Planning, Mode Management, and other issues are not addressed in this scenario.

Summary of Procedures:

- Software is prepared delivered using a CM operated *DeliveryTool*. An approved CCR specifies which previously prepared software is to delivered and to which sites. Refer to CM-1-032-1, *COTS and Custom Software Preparation and Delivery*
- DAAC CCB Approves the Installation of the delivered software into a DAAC Operational Baseline (e.g., a mode, "OPS", "TS1", etc.)
- Software is installed in accordance with CCR installation instructions, and in accordance with local DAAC procedures/processes
- Delivery Tracking pages are maintained by EMD CM, located at:

http://pete.hitc.com/baseline/

LPDAAC	http://e0ins01.edcb.ecs.nasa.gov:10160/baseline/
GES	http://g0ins01.gsfcb.ecs.nasa.gov:10160/baseline/
LaRC	http://l0ins01.larc.ecs.nasa.gov:10160/baseline/
NSIDC	http://n0ins02u.ecs.nasa.gov:10160/baseline/
SMC	http://m0mss16.ecs.nasa.gov:10160/baseline/

9.6.4.2 Functional Roles

EMD CM Administrator--Ensures that changes to the hardware, software, and procedures are properly documented and coordinated. Recommends configuration of all EMD configured hardware and software.

DAAC/SMC CM Administrator--Ensures that changes to the hardware, software, and procedures are properly documented and coordinated. Maintains control of all configured hardware and software. Assists in the development and administration of the library with respect to local configuration management procedures.

DAAC Sustaining Engineering--SW Maintainer--Produces, delivers, and documents the corrections, modifications, and enhancements made to EMD software (including COTS), and/or adaptations or incorporations of EMD COTS software.

9.7 Baseline Manager

9.7.1 Overview

The EMD provides a *ClearCase* Baseline Manager (BLM) tool to assist in documenting changes to the baseline, and to maintain a historical record of those changes. The tool is used by CM Personnel to manage baseline data about resources deployed to all external ECS sites, including the DAACs and the SMC, as well as the three internal ECS sites, the PVC, VATC, and the EDF2 string (Evolution). The BLM tool is used at the EMD Development Facility to maintain system-level records and site-level records; baseline reports are accessible at the operational sites by accessing the URL http://pete.hitc.com/baseline/. This is the EMD Baseline Information System (EBIS). Each site has a replicated EBIS that is served locally. Selection of the "Technical Documents" button provides detailed reports regarding the EMD baseline. As part of Evolution, the Solaris 8 host "pete.hitc.com" has been replaced by Linux host "c4cbl02.hitc.com". To maintain transparency to the EDF users, an alias was created so that the usual EBIS URL could be used. The EDF EBIS though is now served from Linux host "c4cbl02.hitc.com".

9.7.2 Baseline Terms and Concepts

Baseline management is a process to identify and control versions of hardware and software, to provide a standard configuration of systems throughout all sites, and allow unique site-configured systems and baselines. It identifies interdependencies between hardware and software items, and permits maintenance of a complete history of baseline changes throughout the life of the project. For EMD baseline management and BLM tools, certain terms and concepts are key to understanding how data on the system baseline are stored and tracked.

Control Item – any EMD item under version control by Configuration Management.

Configuration Item – an aggregation of hardware, firmware, software, or any discrete component or portion, which satisfies and end user function and is designated for configuration control.

Baseline – a configuration identification document or set of such documents formally designated by the Government at a specific time during the life cycle of a configuration item (CI).

Configured Article – a control item reportable as part of the Configured Articles List (CAL).

CIL – a Configuration Items List (CIL) identifies the approved set of CIs that are subject to CM requirements and procedures.

CAL – a Configured Articles List (CAL) describes all CIs, critical item hardware and software, and supporting documentation by which the exact configuration definition of the hardware and software can be determined.

Additional terms, some of which address specific entries in the BLM tool, further define how data on the system baseline items and structure are tracked.

Assembly –	An item made up of other items.
Bill of Material –	The list of items that comprise an assembly.
Product Structure –	The parent-child pairings that define the bill of material for an assembly; each product structure record specifies the effective dates and quantities for a single component of a parent for each engineering change.
Active Date –	The date a component becomes effective in an assembly's bill of material. This date is the CCR approval date.
Inactive Date –	The date a component is no longer effective in an assembly's bill of material. This date is when a new CCR displaces or affects the item.
Engineering Change –	A mechanism for grouping, reporting, and controlling product changes collectively.
Revision –	The sequence number of a product structure change to an assembly; it signifies a change to the configuration of an assembly that does not alter its form, fit, or function.
Implementation Status –	A record describing the deployment of a control item to a site and the current state and associated date of its implementation; each control item has one record for each site to which it is deployed.
Exporting Data –	Creating a formatted file or records extracted from the BLM database; control item engineering change, product structure, and interdependency records may be extracted and sent to other application software at any site.
Importing Data –	Loading BLM data from a formatted file.

At the lowest level, the baseline is composed of configured articles that are the specific types of items that make up EMD and are tracked using the BLM tool. It is important to recognize, however, that we impose a conceptual structure on those configured articles to help us think about the system. In fact, it is possible to conceptualize the structure of the system in a number of different ways, and we may select a different conceptual structure based on the requirements of the situation. The EMD baseline management approach and the BLM tool permit recording and tracking these different conceptual baselines, which can be related to the same records of the configured articles.

For example, system designers may conceptualize the system in terms that will help them track subsystems and the configuration items for which each subsystem team is responsible. This may produce a baseline structured according to a design view, such as that shown in Figure 9.7-1.

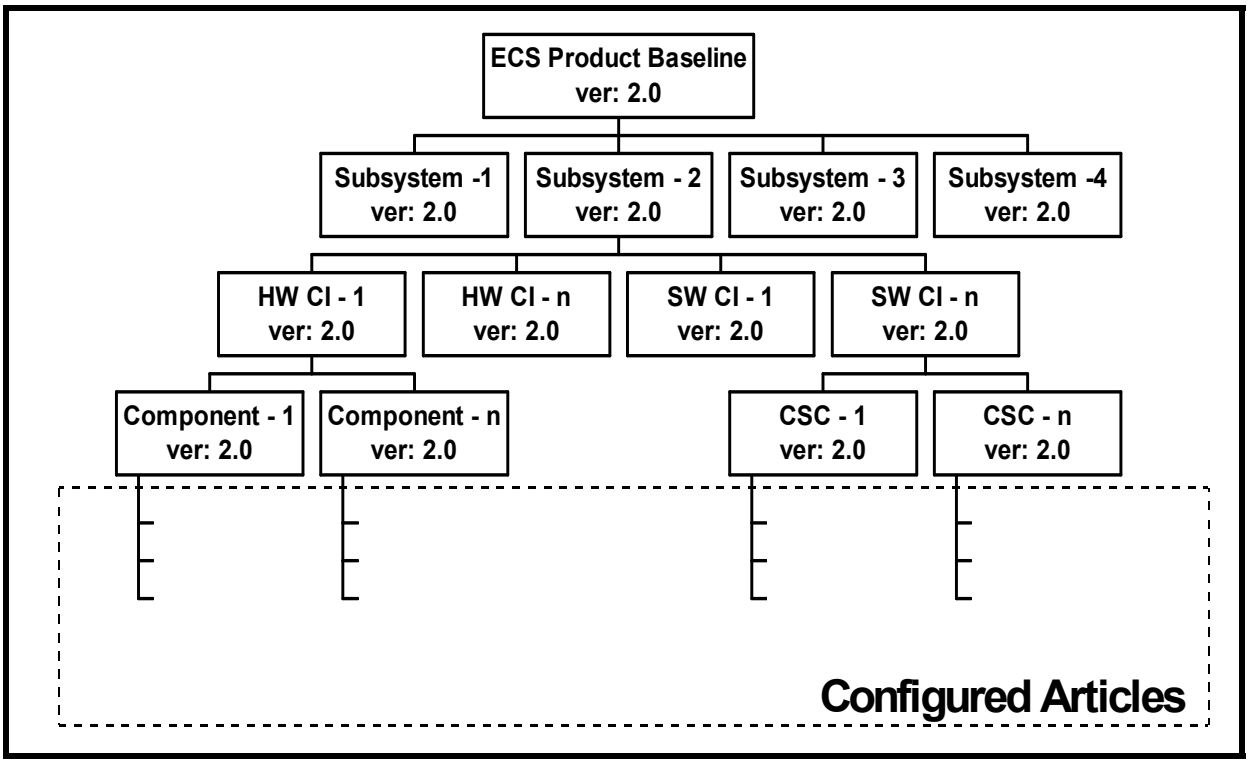

Figure 9.7-1. EMD Baseline Concept from a Design (CIL/CAL) View

At an operations site, the concept reflected in the upper layers of the Design View baseline structure may not be particularly useful. Although the same configured articles are involved, it may be desirable, for instance, to track items from the viewpoint of network administration. The resulting baseline product structure may reflect that shown in Figure 9.7-2.

Even if an operations site is to view EMD product structure as composed of subsystems, it is likely that the concept of CIs will be of little use. Instead, the site is likely to be focused on what hosts make up the subsystems. Therefore, the subsystem view at an operations site may be similar to that illustrated in Figure 9.7-3.

The Baseline Manager database implemented at the EMD Development Facility reflects EMD-developed product structures, and site personnel may not normally need all the data necessary to define these product structures. Instead, BLM tasks are likely to be limited to areas such as noting system changes, perhaps in the context of site-unique requirements and data. However, an understanding of the different ways of conceptualizing the system will help in interpreting baseline data reflected in the BLM.

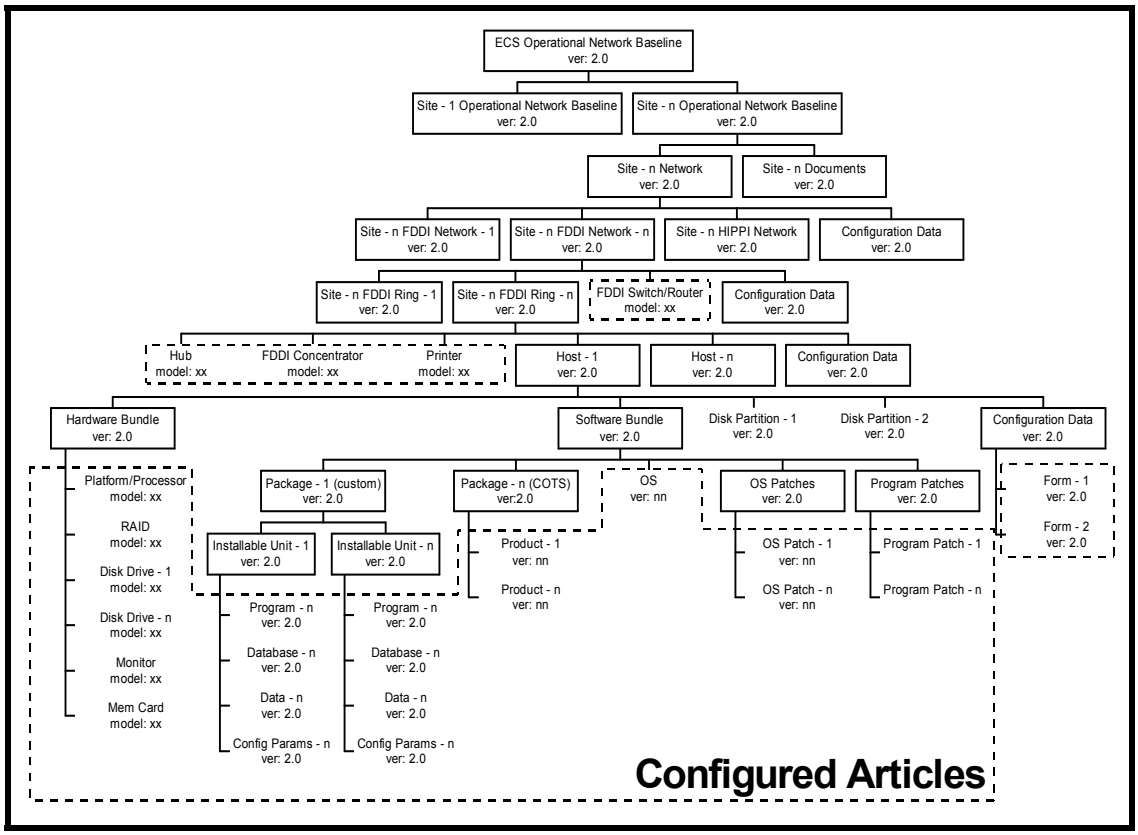

Figure 9.7-2. EMD Baseline Concept from an Operational (Network) View

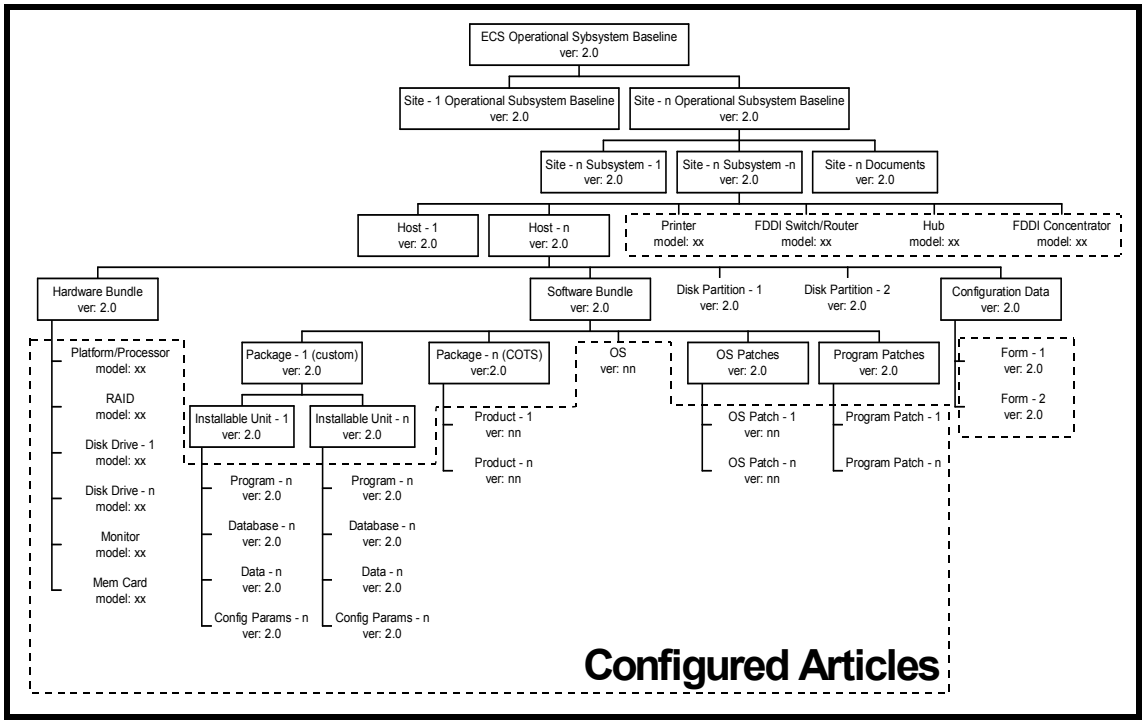

Figure 9.7-3. EMD Baseline Concept from an Operational (Subsystem) View

9.7.3 Baseline Manager (BLM) Outputs Useful at the Sites

The BLM manages the COTS software, operating system patches, and COTS software patch baselines. The BLM records, including information on all scripts, data, and GUIs, are maintained and managed at the EMD Development Facility using ClearCase. The BLM tool produces some of the 910/920 Technical Document reports, with automated posting to EMD Baseline Information System (EBIS) and replication to mirror sites at each DAAC. The reports include the following documents that affect all sites:

- COTS Software Versions 910-TDA-003.
- Site-Host Map 910-TDA-005.
- Critical COTS Software List 910-TDA-023.
- COTS S/W Where-Used Reports 910-TDA-030.

The reports also include the following documents that are site specific:

- Hardware-Software Maps 920-TD(*x*)-002.

  (*Note*: The *x* represents a letter designating specific DAACs. *e.g.*, g = GSFC, l = LaRC, e = LP DAAC, s = SMC, v = VATC, p = PVC, and n = NSIDC.)
- O/S and COTS S/W Patch Maps 920-TD(*x*)-014.

All BLM records are related to approved Configuration Change Requests (CCR) and Release Notes documents (i.e., series 914-TDA-xxx).

The Configuration Management (CM) organization is the principal user of the BLM tool, using it to implement changes to the baseline. The system is used daily to describe CCB-approved system components and to track sites and machines where version-controlled items are configuration controlled. In addition BLM supports other functions such as configuration audits, system engineering and deployment activities. The BLM records describe the hosts for each site. The sites are the operational Distributed Active Archive Centers (DAACs), Verification and Acceptance Test Center (VATC) and Performance Verification Center (PVC). The system also tracks the COTS software and patches that are mapped to their respective hosts. EBIS accommodates the identification of all configuration-controlled items such as documents, and disk partitions.

The BLM capabilities are used to:

- Maintain records that identify what items comprise individual, baselined, system configurations
- Identify the versions and variants of hardware and software items that are currently baselined together with the assemblies (e.g., hosts, subsystems, and networks) that use them
- Record item interdependencies and the sites to which baselined items are deployed

- Keep chronological histories of baseline changes and traceability of items to predecessor versions and system releases

9.7.4 Procedure for Retrieving Baseline Reports

When the EMD software baseline is changed (e.g., addition of a script, update or replacement of a Graphical User Interface (GUI) package), the change must be reflected in the collection, or "catalog," of control items that make up the affected Computer Software Component (CSC) assembly in the EMD product structure. In the BLM software at the EDF, to document the change it is necessary to add the new element to the catalog of version-controlled items, define an engineering change for the CSC assembly, and include the element in the list of items that will now make up that assembly. Once the change is documented, baseline reports reflect the new information. These reports may be accessed through the EMD CMDM server for the EMD Baseline Information System (EBIS). See Figure 9.7-4 .

The EBIS page is obtained by a click on the **EMD Baseline** button at the end of the Tools line near the top of the EMD CMDM Server page. On the EBIS page, the EMD Baseline Information Technical Documentation is accessible through use of the **Technical Documents** button at the top of the row of buttons on the left side.

The resulting **EMD Baseline Technical Documentation** page lists the document series, title, and document number. The document numbers are links that provide access to the listed documents. The titles of some documents indicate BLM origin by inclusion of the parenthetical notation **(ClearCase)**.

9.7.4.1 Retrieving baseline reports

1. At the UNIX command shell prompt, type **setenv DISPLAY *clientname*:0.0** and then press the **Return/Enter** key.
   - For ***clientname***, use either the local terminal/workstation IP address or its machine name.
2. Start the log-in to a Netscape host by typing **/tools/bin/ssh *hostname*** *(e.g., x4oml01)* at the UNIX command shell prompt, and press the **Return/Enter** key.
   - If you receive the message, **Host key not found from the list of known hosts. Are you sure you want to continue connecting (yes/no)?** type **yes** ("**y**" alone does not work).
   - If you have previously set up a secure shell passphrase and executed **sshremote**, a prompt to **Enter passphrase for RSA key '<*user@localhost*>'** appears; continue with Step 3.
   - If you have not previously set up a secure shell passphrase; go to Step 4.
3. If a prompt to **Enter passphrase for RSA key '<*user@localhost*>'** appears, type your ***Passphrase*** and then press the **Return/Enter** key. Go to Step 5.

4. At the ***<user@remotehost>*'s password:** prompt, type your ***Password*** and then press the **Return/Enter** key.
   - You are logged in and a UNIX command shell prompt is displayed.
5. Type **netscape &** and then press the **Return/Enter** key.
   - The Netscape web browser is displayed.
6. Click in the **Netsite:** field.
   - The field is highlighted.
7. Type the Universal Resource Locator (URL) for the EMD Baseline Information Server (**http://pete.hitc.com/baseline/**) and then press the **Return/Enter** key.
   - The **EMD CMDM Server** home page is displayed, offering access to EMD Baseline information as well as a number of tools, EMD web sites, and NASA EOS web sites.
8. On the Tools line click on the **EMD Baseline** button.
   - The **EMD Baseline Information System** page is displayed
9. Click on the **Technical Documents** button.
   - The **EMD Baseline Technical Documentation** page is displayed.
10. Locate the desired report, scrolling down as necessary.
    - Reports derived from the BLM may be indicated with a parenthetical notation **(ClearCase)** in the title entry.
11. Click on the link for the document to be accessed.
    - A directory is displayed with one or more document numbers and versions indicated as links.
12. Click on the link for the document and version desired.

    The document is displayed, and can be printed and searched.

    A sample of the EBIS page is shown in Figure 9.7.4 -1

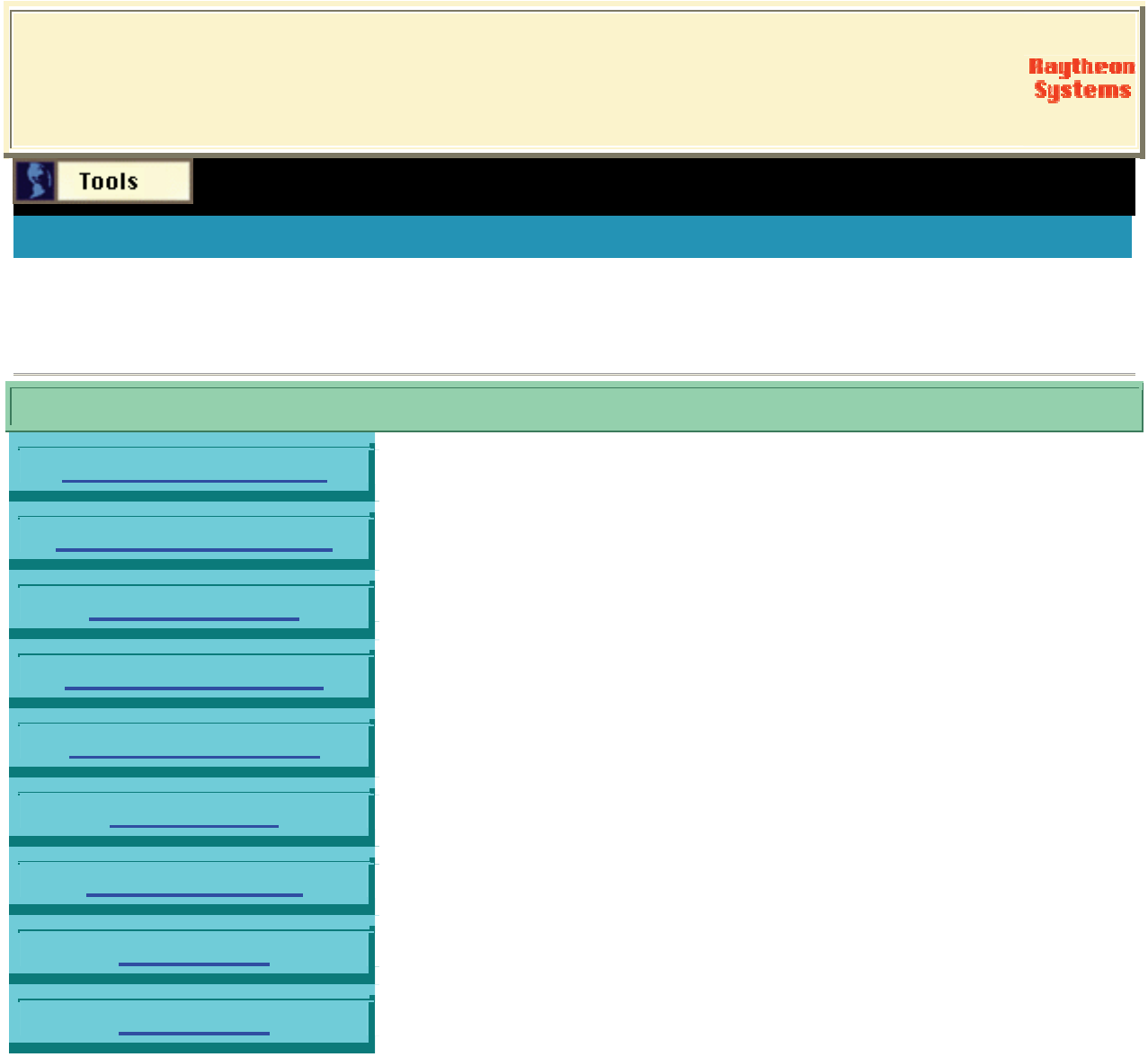

Figure 9.7-4. Sample EBIS Page

This page intentionally left blank.

10. Metadata Administration

Every science data product generated and archived by the system must be described to the system by metadata that are put into an inventory and then used to retrieve and distribute the data to users of the system. The Earth Science Data Model, described in document 420-EMD-001, Release 7 Implementation Earth Science Data Model, organizes the metadata into groups of related attributes and services to be performed on the data products. These "core" attributes are necessary to identify, interpret and perform services on granules and collections. The Data Model also provides for "product-specific" attributes (PSAs), i.e., attributes which are unique to a specific data product.

The smallest aggregation of data that is independently described and inventoried in the system is referred to as a data granule. Granules are organized into logical groupings called collections in which the granule metadata varies principally by time or location, called single-type collections.

Every collection is described by an Earth Science Data Type (ESDT) and is made known to the system by an ESDT descriptor file and associated software code that is built into the Data Server's dynamic link library (DLL) to perform the services.

Metadata administration includes creating and updating ESDTs. Collections may be modified and updated over time. In addition, quality assurance is performed after ESDTs have been installed and granules have been generated and stored in the archives. Collection-level metadata can be updated by updating the ESDT. Granule-level metadata can be updated manually (i.e., not as a result of an operation such as subsetting, which modifies the science data content of a granule) by setting the Quality Assurance flags and explanations. Procedures for updating these flags are provided in Chapter 15, Quality Assurance.

10.1 ESDT Descriptor Files

The primary task in establishing a collection is providing the core and product-specific metadata attribute values. This is done by creating an Earth Science Data Type (ESDT) descriptor file. The descriptor file is also used to specify the data services that are available for granules that belong to the collection. These services are implemented as methods of a Dynamic Link Library (DLL) containing C++ code to accomplish each service. The descriptor file and the DLL are the means by which a collection is made known to the Science Data Server (SDSRV).

Table 10.1-1 contains the activity checklist for maintaining ESDT Descriptor Files

Table 10.1-1. ESDT Descriptor - Activity Checklist

Order	Role	Task	Section	Complete?
1	DBA	Generate Descriptor Files	(P) 10.1.1	
2	DBA	Verifying Descriptor Files	(P) 10.1.2	

The ESDT descriptor is composed of sections, all in ODL, containing the following information:

- Collection level metadata attributes with values contained in the descriptor.
- Granule level metadata attributes whose values are supplied primarily by the Product Generation Executives (PGEs) during runtime.
- Valid values and permitted ranges for all product-specific attributes.
- List of services for all the granules in the collection and events that trigger responses throughout the system.

The services that apply to a collection are specified in the ESDT descriptor file. Product-specific services, such as subsetting or a product-specific acquire, require executable code to enact those services. This code is contained in the DLL. The DLL is written and tested by EMD sustaining engineering personnel.

After the ESDT (both descriptor file and DLL) has been generated it must be installed on the Science Data Server before the first data granule can be inserted. During this installation process, information from the ESDT Descriptor File is propagated to the Data Management Subsystem, the Subscription Server, and the Spatial Subscription Server, all of which must be operating during the ESDT installation process. The detailed procedures for ESDT installation are described in Chapter 26, Science Software Integration and Test.

10.1.1 Generating Descriptor Files

1 Identify desired collection-level metadata attributes.
- For permanent and interim files use only the minimum attributes.
- For distributable products identify all applicable attributes. This will involve reading appropriate documentation and interacting with the data provider.

2 Identify granule-level attributes.
- If a sample metadata configuration file is available from the data provider, use this.

3 Check “valids” (allowable metadata values) for core attributes (write CCR if new valids are required).

4 Check PSAs (register PSAs if new).

5 Use custom built scripts and a text editor to generate the descriptor file.

6 Verify the descriptor file as outlined in Section 10.1.2.

7 Check descriptor files into ClearCase.

10.1.2 Verifying Descriptor Files

1 Run the PERL script "update.pl", following the instructions in the script prologue.
- This script makes sure that the inventory metadata attributes are all listed as event qualifiers in the EVENT group.

2 Run the PERL script esdtQC.pl following the instructions in the script prologue.
- This script checks for more than 30 common descriptor file errors.

3 Make any necessary corrections in response to errors issued.

4 Rerun the PERL script esdtQC.pl.

5 Repeat Steps 3 and 4 until there are no errors.

6 Run the testodl utility to ensure that there are no errors in the ODL structure for the descriptor file.

7 Make any necessary corrections in response to errors issued.

8 Rerun the testodl utility.

9 Repeat Steps 9 and 10 until there are no errors.

10.2 Preparation of Earth Science Data Types

An ESDT goes through pre-operational life cycle steps starting with an analysis of the collection's need and continuing through development and operational installation. This process involves actions by the Data Provider or User in addition to EMD. The procedures are detailed in Software Development (SD) Project Instruction SD-038 ESDT Creation, Testing, Maintenance,& Integration at http://dmserver.hitc.com/EMD_PAL/index.html). The overall workflow is shown in Figure 10.2-1.

10.2.1 Definitions

Archive - A File Type indicating granules will be inserted to Data Server for long-term storage and acquisition for distribution.

Full - A level of metadata coverage intended for data products that are produced within the system.

Collection - A related group of data granules.

Granule - The smallest data element that is identified in the inventory tables.

Interim - A File Type indicating granules are temporarily stored in support of product generation.

Intermediate - A level of metadata coverage intended for contemporaneous data products that are not produced within the system.

Limited - A level of metadata coverage intended for heritage data products brought into the system for distribution

Minimal -A level of metadata coverage sufficient to uniquely identify a collection or granule.

Permanent - A File Type indicating static or semi-static granules that are used only as inputs in product
generation.

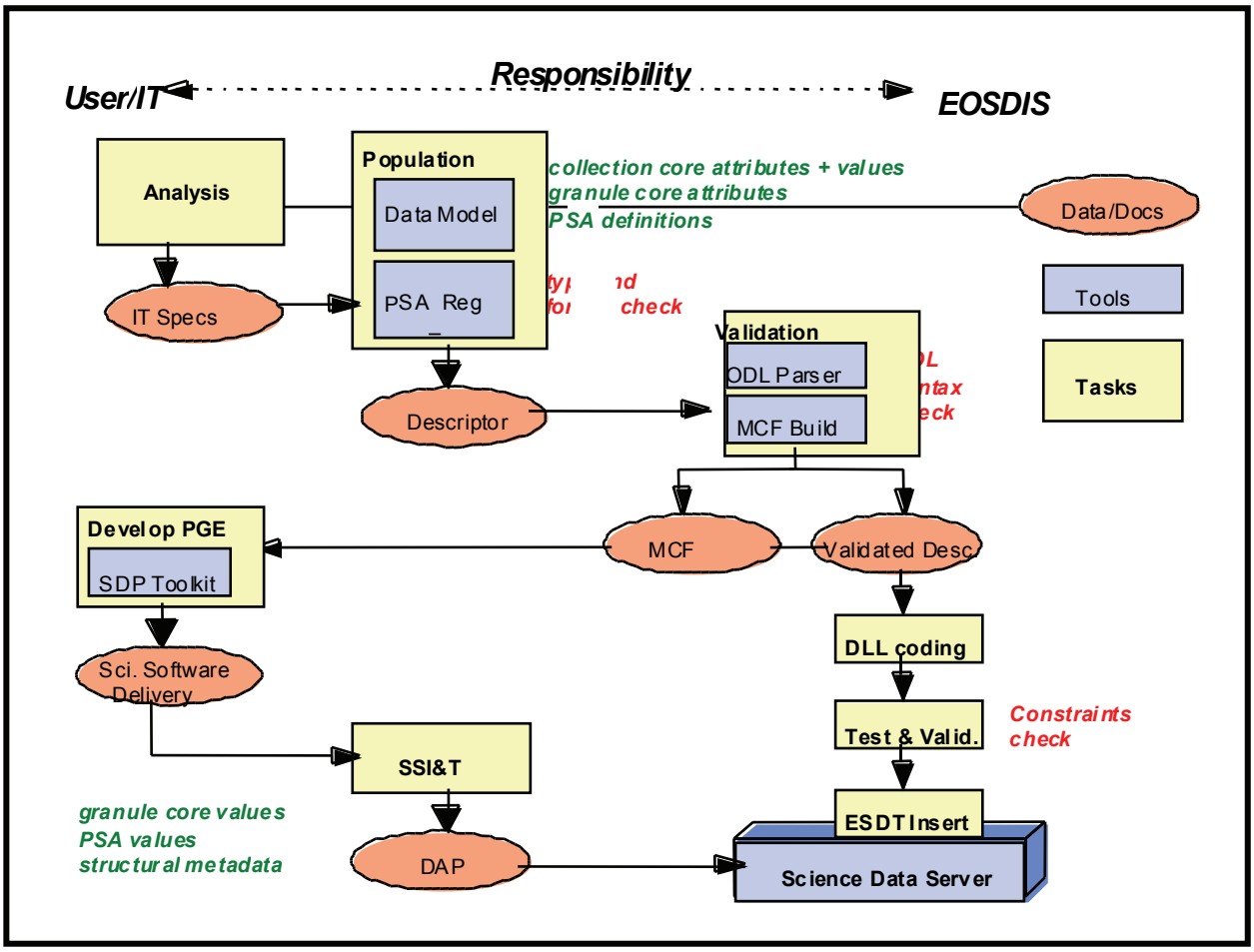

Figure 10.2-1. The ESDT Generation Process

Product Specific Attributes - Attributes that are defined by the data provider in support of searching for specific granules

Valid - An allowable metadata value.

10.2.2 Process

1 Need Analysis

- The baseline list of science ESDTs and their services is controlled by the ESDIS CCB. This baseline was established through an analysis of the system Functional and Performance Requirements Specification, the Technical Baseline established from inputs from the Ad Hoc Working Group on Production, and meetings with the individual data providers to define the basic requirements of each ESDT.
- The basic requirements are:
  - Data Provider File Designation,
  - File Type (Permanent, Interim, Archive)
  - Level of Metadata Coverage (Minimal, Limited, Intermediate, Full)
- For new ESDTs not currently in the development baseline, the result of the Need Analysis forms the basis for approving the inclusion of the ESDT into the system. This is accomplished through the CCR process.

2 ESDT Specification

- This step results in a set of specifications extending the results of the needed analysis and providing the information needed to implement an ESDT. This step is executed only if the ESDT has been included in the baseline. The roles and responsibilities for developing the specification are as above.
- The specifications must include:
  - ShortName and VersionID of the ESDT
  - A list of the metadata attributes needed, valids, and any constraints on attributes.
  - A list and specification of the services needed (e.g., specification of the INSERT, SEARCH, ACQUIRE and SUBSCRIPTION semantics).

3 ESDT Generation

- Once the ESDT Specification has been developed and the applicable attributes identified, the necessary metadata has to be gathered, the metadata values checked against the valid values and the product-specific attributes (PSA) need to be checked against the list of PSAs that are already defined (see Figure 10.2-2).
- Once the collection-level metadata and granule-level attributes have been checked, then the descriptor file is generated, the Dynamic Link Library produced (if a custom DDL is needed), and testing and validation of the ESDT performed. This process is further elaborated in the sections below.
- For a one-of-a-kind, distributable product with Full metadata coverage, this process can take up to six weeks to accomplish. For a related group of products with identical services, much of the Descriptor File and DLL of the first ESDT can be reused, and the cycle time for preparing subsequent ESDTs in the related group is much less.

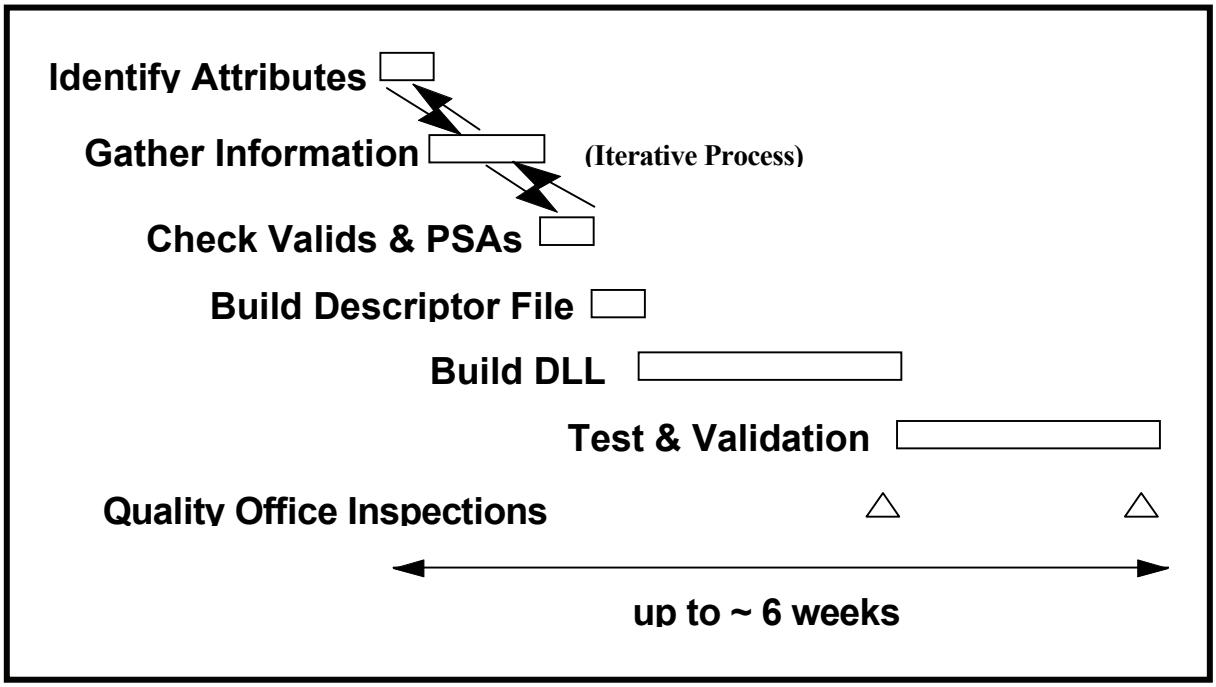

Figure 10.2-2. Steps in ESDT Development

10.3 Metadata Population

10.3.1 Collection-Level Metadata

A majority of the attributes in the Data Model apply to all the granules in the collection. These are known as collection-level attributes. There can be both core and product-specific collection-level attributes, defined once prior to establishing the collection.

Collection-level metadata is input either a text editor or a custom built script.

10.3.2 Granule-Level Metadata

The attributes in the Data Model that can vary on a granule-by-granule basis are known as granule-level attributes. There can be both core and product-specific granule-level attributes.

Granule-level metadata are specified and populated using the Metadata Configuration File (MCF). The MCF is derived from information contained in the ESDT descriptor file and is delivered by the Science Data Server for use by the Data Processing Subsystem or Ingest Subsystem. The MCF specifies how the searchable metadata attributes will be populated in the SDSRV database. For data products generated within the system, the science software or Product Generation Executive (PGE) interacts with the MCF using metadata tools contained in the Science Data Processing Toolkit. Through this process values are set for metadata attributes specified in the "source" MCF, such as the temporal or spatial coverage of each granule. These

values are then inserted into a "target" MCF at PGE run time. The MCF is used in a similar manner for data entering the system through the Ingest Subsystem.

Procedures for entering data into the system through the Ingest Subsystem are described in Chapter 16, Ingest. Procedures for running a PGE are described in Chapter 13, Production Planning and Chapter 14, Production Processing.

The actual population of the granule-level attribute values into SDSRV inventory databases takes place during the insert of a data granule into the SDSRV. Each data granule consists of one or more physical files. Accompanying each granule is a metadata record; i.e., an ASCII file containing the granule-level attributes and their values in ODL. Only one metadata record is allowed per granule, i.e., no sub-granule records are allowed, and no metadata records are shared between granules.

10.3.3 Product-Specific Metadata

Product-specific metadata can be at both the granule level and the collection level. Product-specific metadata may (at the data provider's election) be contained in the inventory tables in the database, in which case it will be searchable by the system. There is also a provision to store product-specific metadata within granules that is available only when the granule has been ordered and delivered. This is termed archive metadata and is specified in a separate ODL group in the MCF.

In the granule metadata, the core attribute that is available to store product-specific metadata is called ParameterValue. The metadata describing this attribute is specified by the data provider through the AdditionalAttributes class at the collection-level. The units of measure, range, accuracy, and resolution for this are specified in the PhysicalParameterDetails class, also at the collection-level.

Product-specific metadata at the collection level is specified at the time the other collection level metadata attributes values are defined. At the granule-level, product-specific metadata is defined in the MCF. In both cases, a list of valid values and permitted ranges are specified in the ESDT data dictionary.

10.4 Testing and Validation

Testing and validation involve installation of the ESDT on a Data Server (as described in Chapter 26), and subsequent tests of the data services for the ESDT. The tests include insertion of actual or simulated data, search, acquire, and other services that may apply and be available and supported under the extant version of the system.

After testing, the ESDT Descriptor File and DLL are promoted to the development CM environment if pre-release, or to the operational environment after the release is made operational.

This page intentionally left blank.

11. Production Rules

This section has been removed.

This page intentionally left blank.

12. Resource Planning

This section has been removed.

This page intentionally left blank.

13. Production Planning

This section has been removed.

This page intentionally left blank.

14. Production Processing

This section has been removed.

This page intentionally left blank.

15. Quality Assurance

In release 7.20 the SDSRV and the Data Pool subsystems contain utilities for managing QA metadata. The process begins by running the SDSRV QAMUT utility. This utility updates the QA metadata within the SDSRV database with the new QA values, it records the updates so that BMGT can later export them to ECHO, and finally it provides an output ("Undo") file that is used by the DataPool QA Update utility. The DataPool QA Update utility takes the output ("Undo") file from QAMUT and applies the changes to both the QA metadata attributes in the DPL database and the QA metadata stored in the XML files within the DPL file systems.

This section describes the procedures for setting Quality Assurance flags in the metadata for science granules using QA Metadata Update Tool (QAMUT).

The QAMUT provides a tool for updating the QA metadata for multiple granules in a batch.

All sections related to Quality Assurance address procedures for the following functions:

- Section 15.1 Using the QA Metadata Update Tool.

.

For each function, an **Activity Checklist** table provides an overview of the tasks to be completed. The outline of the Activity Checklist is as follows:

Column one - ***Order*** shows the order in which tasks could be accomplished.

Column two - ***Role*** lists the Role/Manager/Operator responsible for performing the task.

Column three -***Task*** provides a brief explanation of the task.

Column four - ***Section*** provides the Procedure (P) section number or Instruction (I) section number where details for performing the task can be found.

Column five - ***Complete?*** is used as a checklist to keep track of which task steps have been completed.

15.1 Using the QA Monitor

Using the QA Metadata Update Tool

The QA Metadata Update Tool (QAMUT) is an operational support tool used for updating the values of the Quality Assessment (QA) flags in the inventory metadata in the Science Data Server database. The QAMUT sets QA values for data granules containing one or more measured parameters after they have been assessed by Science Computing Facility (SCF) or DAAC staff to determine their quality.

The QAMUT is used to update the Science and Operational QA flags and the corresponding fields only.

QA flags can have the following values:

- Passed.
- Failed
- Being Investigated.
- Not Investigated.
- Inferred Passed.
- Inferred Failed.
- Suspect
- Hold

Table 15.1-1 provides an Activity Checklist for Using the QA Mut.

Table 15.1-1. Using the QAMUT - Activity Checklist

Order	Role	Task	Section	Complete?
1	System Administrator/ Database Administrator	Configure the QAMUT	(P) 15.1.1.1	
2	Production Monitor	Populate DsQAMUTESDTSite Table	(P) 15.1.2.1	
3	Production Monitor	Update QA Metadata Flags Using QAMUT	(P) 15.1.3.1	

During one run the QAMUT can update the metadata QA flags for multiple granules. In fact, the strength of the tool derives from its ability to update batches of granules at a time. This is in contrast to the QA Monitor GUI, which can be used to update the QA flags for just one granule at a time. There is no set limit on the number of granules that may be specified for a run. In fact, depending on how frequently the originators of requests for QA flag updates submit their requests, the DAAC may receive requests for updates of thousands of granules at a time. However, this creates the potential for extreme database loading (e.g., requirements for temporary storage of granule information). Specific practical limits may depend on individual site capacities and requirements, and the DAAC may need to work with the originators of requests to formulate requests of appropriate size to minimize QAMUT processing times and associated database impacts. In practice, it is likely that requests should be kept to updates for no more than 10,000 granules at a time. If a request is for significantly more than that, consideration should be given to breaking it up into multiple requests.

The granules with QA flags to be updated using the QAMUT may each contain several different measured parameters. The tool can update the QA flag associated with each parameter for each granule listed in a metadata update request. Updates for different measured parameters related to a particular granule may be grouped contiguously on separate lines in the request so that all the updates for the granule are accomplished at the same time.

The input needed to run the QAMUT is a uniformly formatted update request. SCF personnel typically send their metadata update requests to the DAAC by e-mail. Each update request contains an e-mail header (including the requester's return address) and a list of the granules to be updated, along with the new QA flag values for the specified parameters.

The body of the request starts with the statement "begin QAMetadataUpdate <Science or Operational> <LGID, GranuleUR or ESDT>". The body ends with an "end QAMetadataUpdate" statement. Each request can be based on 3 possible origins: LGID, GranuleUR, or ESDT with temporal range. In between is at least one parameter/QA value statement with the following components (which are separated by tabs):

- Short Name
- Version ID
- LGID, GranuleUR, or Range Beginning Date <tab> Range Ending date
- Measured Parameter Name/ALL
- QA Flag Value
- QA Flag Explanation Value

Each parameter/QA value statement starts on a new line.

The example in Figure 15.1-1 is an ESDT type of change and has four statements requesting science QA flag updates to parameters associated with four different granules. All are to be set to "Passed" based on a Performance Test.

This information must be properly arranged and placed in the SDSRV database (a designated directory or file).

Once a request to update the metadata has been received, the correctly formatted information must be saved to the designated directory or file. Once the data has been copied to this directory, the metadata can be updated by using QAMUT (see procedure 15.1.3).

The QAMUT has been designed to run independently of the SDSRV process. The system directly updates the inventory metadata database instead of going through the SDSRV services to update the database. The QAMUT is accessible on the SDSRV Server database host (e.g., x4oml01).

The QAMUT includes three scripts:

- **EcDsQAMUT.pl** -- the main script that does the update.
- **EcDsQAMUTBcp.pl** -- an assistant script that helps the DAAC load the DsQAMUTESDTSite table in the SDSRV database with information about the sites and the related ESDTs each site may request to have updated.
- **EcDsQAMUTEmailScript.pl** -- a script generated by EcDsQAMUTEmailScriptMkcfg and used to direct the email QAMUT update requests to the proper request directories.

```
begin QAMetadataUpdate Science ESDT

DFLAXENG   1  May 27 1999  9:00:00:000PM  May 28 1999  9:00:00:000PM   ALL   Passed   ESDT Perf Test
DFLAXLSM   1  May 27 1999  9:00:00:000PM  May 28 1999  9:00:00:000PM   ALL   Passed   ESDT Perf Test
DFLAXSTR   1  May 27 1999  9:00:00:000PM  May 28 1999  9:00:00:000PM   ALL   Passed   ESDT Perf Test
DFLAXMIS   1  May 28 1999 12:00:00:000AM  May 28 1999  9:00:00:000PM   ALL   Passed   ESDT Perf Test
End QA Metadata Update Science ESDT
```

Figure 15.1-1. Sample Metadata QA Update Request

15.1.1 Configure QAMUT

There are two configuration files developed for the QAMUT. These files are always developed as part of at the installation or when new ESDTs are added to the system. The site installer or Database Administrator is responsible for maintaining these files.

The first configuration file, used by the DAAC operator, contains a mapping of ESDT names and SCF sites. This file is a configurable data file that must be created and maintained by the operator, using an available editor (e.g., **vi**), in order to populate DsQAMUTESDTSite table in the SDSRV database using **EcDsQAMUTBcp.pl**. The file may be named appropriately by the operator when it is created (e.g., **bcpfile**). This file must use the following format:

<ESDTShortName><tab><SITEName>

Repeat this format for each SCF Site and all the ESDTs it can update. Note, no blank line is allowed in the file.

The second configuration file, used by the QAMUT, is called **EcDsQAMUT.CFG**. It contains details about how to connect to the Sybase database as well as DAAC specific information. Without this file, the utility cannot run. The configuration file must be a single entry plain text ASCII file which has the following format:

SYB_USER = <string>
SYB_SQL_SERVER = <string>
SYB_DBNAME = <string>
SYB_PASSWD =<string>
NUM_RETRIES = <number
SLEEP_SEC <integer>

QAMUTRequestDIR = <string>
QAMUTCompleteRequestDir = <string>
QAMUTErrRequestDir = <string>
QAMUTUndoRequestDir = <string>
MAILX = <string>

<SCFSite>_FromAddress = <string1,string2,string3…>
…(repeat for each site)

<SCFSite>_ReplyAddress = <string1>
…(repeat for each site)

DAACAddresses=<string1,string2,string3>

<SCF>_Notification={Y/N}
…(repeat for each site)

VALIDQAFLAG = <string>
value1,value2,value3,value4,value5,value6,value7,value8
MAX_NUM_GRANULES = <integer>
UpdateBatchSite = <integer>

Table 15.1-2 lists these configuration parameters and provides a brief description of each.

Table 15.1-2. Configuration File Parameters for QAMUT (1 of 2)

Parameter Name	Description
SYB_USER	The user name for the Sybase connection.
SYB_SQL_SERVER	The name of the SQL server for this Sybase connection.
SYB_DBNAME	The name of the database you intend to connect to
SYB_PASSWD	Program ID used to get Sybase password through a decryption program called EcDsDcrp.
NUM_RETRIES	The maximum number of times the utility will try to connect to the database or retry deadlock. The recommended default is 5.
SLEEP_SEC	The number of seconds the utility will wait ('sleep') between connect. Recommended default is 10.
QAMUTRequestDir	The directory where all the QAMUT update request files reside
QAMUTCompleteRequestDir	The directory where all the completed QAMUT update request files reside.
QAMUTErrRequestDir	The directory where all the QAMUT update requests with non retryable errors reside
QAMUTUndoRequestDir	The directory where all the information required to undo each request is stored in the same filename as the original request. If a request is run multiple times due to recoverable errors, there will be multiple

Table 15.1-2. Configuration File Parameters for QAMUT (2 of 2)

Parameter Name	Description
MAILX	The command including the full path for “mailx”
<SCFSite>_FromAddresses	Each SCF site can have multiple From email addresses separated by “,”. These email address are used for authentication as well as email addresses for QAMUT to send notification back
<SCFSite>_ReplyAddress	Each SCF site can have 0 to 1 Reply-To address. Enter nothing after the “=” sign if the site doesn’t have any Reply-To address. This address is used for email notification.
DAACAddresses	A list of internal DAAC e-mail addresses separated by “,” to which email notification are Copied upon completion of a QA update run. The completion means finishes without retryable errors--internal errors in the DAAC.
<SCFSite>_Notification	Email notifications are sent in the following situations: Authentication failure. ESDT update not allowed. Format errors of the request Non retryable errors in the request and no retryable errors occurred (QAMUT reprocesses the request until it overcomes all the retryable errors before notifying the requester of the non retryable errors) Successful completion. This is when the Notification option makes the difference. If “Y”, a notification will be sent upon successful completion, otherwise no notification will be sent.
VALIDQAFLAGs	Contains 8 valid values separated by “,”
MAX_NUM_GRANULES	Contains the DAAC configurable maximum threshold
UpdateBatchSize	The number of granules the utility will update in a batch.

15.1.1.1 Configure the QAMUT

1 Log into the host for the SDSRV database (e.g., x40ml01).

2 To change to the directory for configuration files, type the command:

cd /usr/ecs/<*MODE*>/CUSTOM/cfg

and then press the **Return/Enter** key.

- The working directory is changed to **/usr/ecs/<*MODE*>/CUSTOM/cfg**.

3 To start the **vi** editor and specify **bcpfile** as the name of the configuration file for mapping ESDT names and sites, type the command:

vi bcpfile

- A new file is opened for editing, or, if a file named **bcpfile** already exists, the contents of the file are displayed, and the cursor is displayed on the first character at the upper left corner of the file.
- ***Note*:** This procedure assumes use of the vi editor and naming of the file bcpfile. Other editors may be used, and the file may be given a different name.

4 If necessary, use the down arrow key on the keyboard to move the cursor down to a blank line.

- The cursor is displayed at the beginning of the selected line.

5 Type **i** to put the **vi** editor into the insert mode.

- The **vi** editor is in the insert mode, but no feedback is provided.

6 Enter data listing ESDTs and sites, one ESDT - site pair per line and the ESDT and site separated by a tab, with no blank lines, in the following format:

<ESDTShortName><tab><SITEName>

Continue until all ESDTs that may be updated by each site are entered. ***Note***: To facilitate this data entry, it may be desirable to obtain lists from the SDSRV database using **isql** and copy the information into the file

7 To leave the insert mode and return to the command mode, press the **Esc** key.

- The cursor moves one character to the left and the **vi** editor is in the command mode.

8 Type **ZZ** to save the file and exit the **vi** editor.

9 To start the vi editor and specify EcDsQAMUT.CFG as the name of the configuration file to be used by QAMUT, type the command:

vi EcDsQAMUT.CFG

- A new file is opened for editing and the cursor is displayed on the first character at the upper left corner of the file.
- ***Note*:** This procedure assumes use of the vi editor. Other editors may be used.

10 Type **i** to put the **vi** editor into the insert mode.

- The **vi** editor is in the insert mode, but no feedback is provided.

11 Enter data to specify how to connect to the Sybase database and provide necessary DAAC-specific configuration information (see Table 15.1-2).

12 To leave the insert mode and return to the command mode, press the Esc key.
- The cursor moves one character to the left and the **vi** editor is in the command mode.

13 Type **ZZ** to save the file and exit the **vi** editor.

15.1.2 Populate DsQAMUTESDTSite Table

Table 15.1-3 presents the steps required to populate the DsQAMUTESDTSite table in the SDSRV database. If you are already familiar with the procedure, you may prefer to use this quick-step table. If you are new to the system, or have not performed this task recently, you should use the following detailed procedure.

15.1.2.1 Populate QAMUT Site Table

1 Log into the host for the SDSRV database (e.g., x4oml01).

2 To change to the directory containing the QAMUT scripts, type the command:

cd /usr/ecs/<*MODE*>CUSTOM/utilities

and then press the **Return/Enter** key.
- The working directory is changed to **/usr/ecs/<*MODE*>CUSTOM/utilities**.

3 Type **EcDsQAMUTBcp.pl <*MODE*> *filename***.
- In the command, ***filename*** is the name of the configuration file containing the mapping of ESDTs and sites (e.g., **bcpfile**).
- The contents in the DsQAMUTESDTSite table in the SDSRV database are replaced with the content in the named file (e.g., **bcpfile**). Before the replacement, the current content of the table is saved in a file called DsQAMUTESDTSite.out in the directory **/usr/ecs/<*MODE*>/CUSTOM/data/DSS/QAMUT/QAMUTUndo**.
- ***Note***: The script completely replaces the content of the DsQAMUTESDTSite table; it does not merely append data. Therefore, to update the table, the entire desired content of the table must be reflected in the configuration file (e.g., **bcpfile**).

Table 15.1-3. Populate DsQAMUTESDTSite Table

Step	What to Do	Action to Take
1	Log in at the host for the SDSRV database	**enter text; press Return/Enter**
2	**cd /usr/ecs/<*MODE*>/CUSTOM/utilities**	**enter text; press Return/Enter**
3	**EcDsQAMUTBcp.pl <*MODE*> *filename*** (e.g., **bcpfile**)	**enter text; press Return/Enter**

15.1.3 Update QA Metadata Flags Using QAMUT

Access to the QAMUT must be gained through the use of UNIX commands. The process of updating QA metadata flags using the QAMUT start-up scripts starts with the following assumptions:

- The applicable servers are running.
- The DAAC operator has logged in to the system.
- A request for metadata update has been received in an acceptable format, such as that shown in Figure 15-2.
- The update request has been saved with the appropriate file name (i.e., **QAupdate_science.txt** or **QAupdate_operational.txt** as the case may be) in the **/usr/ecs/*MODE*/CUSTOM/utilities** subdirectory on the SDSRV database host (e.g., x4oml01).

15.1.3.1 Update QAMUT Flags

1 At a UNIX system prompt type **setenv DISPLAY clientname:0.0**

- Use either the X terminal/workstation IP address or the machine-name for the ***clientname***.
- When using secure shell, the **DISPLAY** variable is set just once, before logging in to remote hosts. If it were to be reset after logging in to a remote host, the security features would be compromised.

2 Start the log-in to the SDSRV database host by entering **/tools/bin/ssh *hostname*** (e.g., e0dbl01, , l0dbl01, n0dbl01) then pressing the Return/Enter key.

- If you receive the message, **Host key not found from the list of known hosts. Are you sure you want to continue connecting (yes/no)?** type **yes** ("**y**" alone will not work).
- If you have previously set up a secure shell passphrase and executed **sshremote**, a prompt to **Enter passphrase for RSA key '<*user@localhost*>'** appears; continue with Step 3.
- If you have not previously set up a secure shell passphrase, go to Step 4.

3 If a prompt to **Enter passphrase** for RSA key '<user@localhost>' appears, type your **Passphrase** then press the **Return/Enter** key.

- Go to Step 5.

4 At the <user@remotehost>'s password: prompt type your **Password** then press the **Return/Enter** key.

5 Type **cd /usr/ecs/<*MODE*>/CUSTOM/utilities** then press the **Return/Enter** key.

- Change directory to the directory containing the custom software start-up scripts for the applicable mode.

- The **<*MODE*>** will most likely be one of the following operating modes:
  - OPS (for normal operation).
  - TS1 (for SSI&T).
  - TS2 (new version checkout).

6 Type **ls** then press the **Return/Enter** key.

- The monitor displays a list of the files in the **/usr/ecs/<*MODE*>/CUSTOM/utilities** directory.

7 Type one of the following command line entries and then press the **Return/Enter** key:

- **EcDsQAMUT.pl <*MODE*>**
  - This command processes data from the designated directory and displays detailed information to the operator about granules as they are updated. The operator is asked for a confirmation before the update. The utility exits the request when the first error occurs.
- **EcDsQAMUT.pl <*MODE* > -noprompt**
  - This command processes data from the designated directory but does not display detailed information to the operator about granules as they are updated. The operator is not asked for a confirmation before the update. The utility exits the request when the first error occurs.
- **EcDsQAMUT.pl <*MODE*> -noexitonerr**
  - This command processes data from the designated directory and displays detailed information to the operator about granules as they are updated. The operator is asked for a confirmation before the update. The utility continues processing when an error occurs.
- **EcDsQAMUT.pl <*MODE*> -file <*filename*>**
  - This command processes data from the designated file and displays detailed information to the operator about granules as they are updated. The operator is asked for a confirmation before the update. The utility exits the request when the first error occurs.
- **EcDsQAMUT.pl <*MODE* > -file <*filename*> -noprompt**
  - This command processes data from the designated file but does not display detailed information to the operator about granules as they are updated. The operator is not asked for a confirmation before the update. The utility exits the request when the first error occurs.
- **EcDsQAMUT.pl <*MODE*> -file <*filename*> -noexitonerr**
  - This command processes data from the designated file and displays detailed information to the operator about granules as they are updated. The operator is asked for a confirmation before the update. The utility continues processing when an error occurs.
- **EcDsQAMUT.pl <*MODE* > -file <*filename*> -noprompt**
  - This command processes data from the designated file but does not display detailed information to the operator about granules as they are updated. The

operator is not asked for a confirmation before the update. The utility continues processing when an error occurs.

8 If an additional request is to be processed, copy the request message into the utilities subdirectory with the appropriate file name.

- For example:

  cp /home/allmode/mail/ScienceUpdateRequest /usr/ecs/<*MODE*>/CUSTOM/utilities/QAupdate_<*science/operational*>.txt

9 Repeat Steps 7 and 8 as necessary to process additional requests for QA metadata update.

This page intentionally left blank.