DTS Testing Plan for the Swift Mission

Micah Johnson
Laboratory for High Energy Astrophysics, NASA/GSFC, Code 662, Greenbelt, MD 20771
micah@milkyway.gsfc.nasa.gov

1 Introduction

This document describes the planned tests for exercising the DTS protocol, which transfers data among separate data centers. It assumes that all sites participating in the testing have already set up a machine with suitable disk space.

General requirements for the machine hardware and software as well information and code for DTS are available at the following web address:

http://legacy.gsfc.nasa.gov/dts/dts.html

If questions arise concerning the usage of the DTS or a bug is discovered in the course of testing, please contact *micah@milkyway.gsfc.nasa.gov* . Provide a detailed description and the associated ".msg" file from the log directory if applicable.

2 Test Summary

There are four types of tests used to verify the operation of the DTS software across the Swift sites.

- The first consists of single site to single site transfers, e.g. each site transfers data to another site. This is to ensure that all sites can talk to each other.
- The second consists of single site to multiple sites transfers, e.g. one site will transfer data to all other sites that join the test. To be effective this test should be done in turn from each single site. The opposite tests should also be exercised: multiple sites transfer to a single site either in a syncronous or asyncronous way.

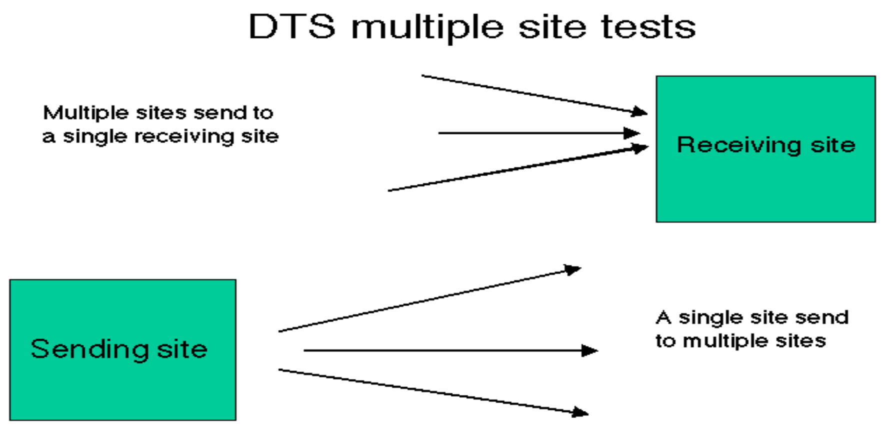

Figure 1: Multiple sites scenarios

- The third is a stress test to determine whether the DTS can handle simultaneous processing of many retrievals. This is achieved with one site sending a large number of transfers in a small amount of time.
- The last is a set of failure/recovery tests, used to determine whether the DTS properly handles transfer problems. The recovery process is tested on recoverable errors, while

unrecoverable errors are tested by ensuring proper notification of the problem is sent to the DTS operator.

All tests (described in detail in section 3.2 and above) are to be performed in combination with every other participating site.

2.1 General commands/procedures

- - SEND OPTIONS. The behavior of the send option can modified with several flags. All options are tested. These are :
  - Option -f . Used to send files listed in a text file (either flat or with directory structure):

    ```
    dts -se SITE -f filelist.txt
    ```

    Where filelist.txt contains

    ```
    example.gif
    example.evt
    ```

    -OR-

    ```
    obsid/inst1/screened/example.evt
    obsid/inst1/images/example.gif
    ```

  - Option -l. Used to send files listed on the command line (either flat or w/ directory structure). For example,

    ```
    dts -se SITE -l *.*
    ```

    -OR-

    ```
    dts -se SITE -l `find obsid -type f -print`
    ```

  - Option z. Compresses files with gzip in ftp directory before sending

    ```
    dts -se SITE -z ALL -f filelist.txt
    ```

  - Option -split. Splits files into smaller pieces in ftp directory before sending

    ```
    dts -se SITE -split -f filelist.txt
    ```

  - Option -t. If AUTORUN is true, type will spawn matching script in dts.scripts

    ```
    dts -se SITE -t TYPE -f filelist.txt
    ```

  - Send to multiple sites. The send command can be used to send data to multiple sites simultaneously by separating them with commas. All previous options may also be used.

    ```
    dts -se SITE1,SITE2,SITE3 -f filelist.txt
    ```

- - AUTOCLEAN. If set to TRUE, this option cleans out the ftp area of the sending site after the receiving site sends an acknowledgement email.
- - ONE TIME PASSWORD. To increase security a one time password mechanism has been implemented within DTS. To use the one-time password option to access FTP site, the key USEOTP in the configuration file needs to be set. However this was implemented before the 'sftp' protocol was available. If the machine where DTS run has installed 'sftp' the one time password feature is unnecessary. For our tests USEOTP is set to FALSE and USESFTP is set to TRUE.
- - RECOVERY MODE. There are potential cases where the transfer is interrupted but can be recovered:
  - Sending site's FTP server is temporarily down. {To simulate : Change to invalid ftp password or site in dts.sitelist}
  - Receiving site's staging directory is temporarily unavailable or out of disk space. { To simulate: Change DTS_STG to invalid directory in dts.config}

  To recover the transfer, the following DTS command is used :

  ```
  dts -r [dts_string].mail
  ```
- - FAILURE NOTIFICATION. In unrecoverable failure scenarios the DTS operator must be notified so that appropriate action can be taken.
  - Sending site's FTP directory unavailable or out of space. Send should fail before sending any message. {To simulate: Change DTS_FTP to invalid directory} This assumes that the send operation is occurring manually or the script that spawns dts can identify this failure.
  - Files deleted before retrieval. Delete directory from ftp area before informing receiving site. Email should be sent to DTS operator.
  - Files corrupted in transfer. Email should be sent to the DTS operator. {To simulate: Corrupt some files in FTP area, echo blah >> filename}

3 Test Detail

Each test will use the following general steps:

```
FROM SITE1 send to SITE2: dts -se SITE2 [send options]
SITE2 RECEIVING MODE    : dts -g
SITE1 CLEANING MODE     : dts -g   (AUTOCLEAN=TRUE)
```

Although the DTS will be largely hands-off once in operation, during the testing period it must be monitored closely to ensure that it is operating properly. Each site must have a contact person so that tests may be coordinated between sites. After each send an email must be sent to the the receiving site's contact person so that they can verify that the SEND message has arrived and can then run "dts -g" The email message must contain the identifier for the test that is being run, listed as "Test: TEST_NAME" in the procedures below. Once the files have been retrieved, the receiving site must verify that they have arrived intact by comparing them to the testbed. Then, they email the sending site's contact person about the results of the test. The sending site, then runs "dts -g" and verifies that the ftp area has been cleaned (AUTOCLEAN is TRUE for all tests).

While diagnosing problems with a particular test or verifying its proper operation it is necessary to know your way around the DTS. The central location for all DTS parameters is the dts.config file. The following parameters defined in dts.config are important for evaluating the test operations:

- DTS_IN - Location of logs when you are the receiving site
- DTS_OUT - Location of logs when you are the sending site
- DTS_STG - Location of incoming files, found in time-coded directories
- DTS_FTP - Location of outgoing files, found in time-coded directories
- MAILFILE - Location of mail file (In some cases it may be useful to view the contents of the mail file before running "dts -g")

After all testing is completed, the staging areas may be cleaned out to save disk space, however, we request that you retain the contents of the log directories, as they contain valuable information on transfer rate, etc.

3.1 Setup

All sites regardless of whether they send, receive, or both must retrieve the testbed from the website:

http://legacy.gsfc.nasa.gov/dts/protect/dtstest.tar.gz

Make a testbed directory, and unpack the files into that directory:

```
gunzip -c dtstest.tar.gz | tar xvf -
```

The top level of the testbed directory should contain the following:

```
ad71031000.002/
doc/
filelist.txt
flat/
scripts/
```

The testbed contains the test data and scripts for use with dts and the testing procedure.

Note that the test procedures are written primarily from the perspective of the sending site, however, special notes to the receiving site are preceded by "Receiving site:"

3.2 Single Site to Single Site

The suite of single site to single site transfer tests is designed to test the various options of DTS.

Test: SIMPLE_FLAT

Change to the 'flat' directory in your testbed. NOTE: at this point dts should be in your path and the DTS_CONFIG environmental variable should be set. Edit the VERSION file in this directory to reflect your site id (e.g. HEASARC, SDC, ISAC, etc.) and the test name (i.e. SIMPLE_FLAT).

```
dts -se SITE2 -l *
```

This should send the VERSION, p0007422101m1s001mievli0000.fit and rp200008.evt files to SITE2. Inform the receiving site.

Receiving site: After running "dts -g" change to the staging area (DTS_STG in dts.config) and go into the most recent directory. Verify that the VERSION file reflects the expected site and test name. Use diff or the ftool fdiff to compare the received files with their counterparts in the local testbed. NOTE: Due to forking operations, the "dts -g" command will not print the shell prompt after it completes. After the DTS messages are done printing, press Enter to return the prompt.

Test: COMPRESS_FLAT

Edit the VERSION file to reflect the test name. From the testbed's 'flat' directory:

```
dts -se SITE2 -z ALL -l *
```

This should send the VERSION, p0007422101m1s001mievli0000.fit and rp200008.evt files to SITE2, compressing them with gzip in the process. Inform the receiving site.

Receiving site: Verify the VERSION file. Either uncompress the files and then diff or use the ftool fdiff which uncompresses on-the-fly to compare the received files with their counterparts in the local testbed.

Test: SPLIT_FLAT

For this test we will split the files into 200 kilobyte pieces before transfer. The split path should be defined in dts.config as follows:

```
SPLITPATH="/usr/bin/split -b 200k"
```

The local split command may be located elsewhere. Modify the path to reflect it. Edit the VERSION file to reflect the test name. From the testbed's 'flat' directory:

```
dts -se SITE2 -split -l *
```

This should send the p0007422101m1s001mievli0000.fit and rp200008.evt files to SITE2, splitting them into smaller pieces in the process.

Receiving site: Verify the VERSION file. The files should be reconstituted in the staging area. Use diff or fdiff to compare the received files with their counterparts in the local testbed.

Test: SCRIPT_PATH

Change to the top level directory of the testbed. Edit the ad71031000.002/VERSION file to reflect your site id and the test name.

```
dts -se SITE2 -type PTEST -f filelist.txt
```

Receiving site:

This test takes more preparation than prior tests. The transmission of the directory structure must be verified through the use of a simple script. Edit dts.scripts to associate PTEST with the full path to scripts/dirlist.pl in the testbed. Edit the first line of dirlist.pl to have the path to perl on your system. Run dirlist.pl manually with no arguments before running DTS. The following should be printed:

```
AUTORUN MESSAGES: dirlist.pl
ERROR: No files
```

If it is not already, set AUTORUN to TRUE in dts.config. Now, run "dts -g" The script should create a text file named dirstruct.txt, which contains the directory structure of the original files, inside the staging directory with the transferred files.

In order to compare the received files with the testbed, it is necessary to restore the directory structure. From within the directory that contains the retrieved files, run the dirbuild.pl script from the testbed, giving dirstruct.txt as the argument. For example, if the testbed directory is in the dtsops home directory:

```
~/testbed/scripts/dirbuild.pl dirstruct.txt
```

This should reconstruct the directory structure and allow comparison with the testbed using the "diff -r" command.

Note: All remaining tests will use this procedure to verify the data integrity of the transferred files.

3.3 Single Site to/from Multiple Sites

Test: MULT_SEND

Edit the ad71031000.002/VERSION file to reflect the test name. From the testbed directory:

```
dts -se SITE2,SITE3 -type PTEST -f filelist.txt
```

All sites involved in the test should be included as part of the comma-delimited list. Special attention should be paid to the cleaning process. It is recommended that you ask each site to perform "dts -g" in turn. As soon as any site informs you of a successful transfer, "dts -g" may be run to clean, however, only the ftp directory corresponding to that site should be cleaned. The recommended procedure follows:

- `dts -se SITE2,SITE3 -type PTEST -f filelist.txt`
- Email contact at SITE2
- Wait for confirmation from SITE2 contact

- `dts -g`
- Verify that only SITE2's directory has been cleaned
- Email contacts at remaining sites
- Wait for confirmation from all remaining sites
- `dts -g`
- Verify that ftp area is clean

Receiving site: This test is a repeat of SCRIPT_PATH from your perspective. Perform the same verification procedure.

Test: MULT_GET

This test requires the most coordination between sites. All sites involved should agree to a time taking into account time zone differences. The recommended course of action is for the receiving site to send out a request to the participating sites requesting a DTS send from all of them.

Sending sites:

Edit the ad71031000.002/VERSION file to reflect the test name. From the testbed directory:

```
dts -se SITE2 -type PTEST -f filelist.txt
```

Email the receiving site when it has been sent.

Receiving site:

After all sites have sent the data, run "dts -g". Multiple dts processes will fork and begin retrieving the files concurrently. After all have completed, compare the new arrivals (one from each participating site) with the testbed (see SCRIPT_PATH).

3.4 Stress Test

Test: STRESS

The purpose of this test to stress the DTS by requiring it to have many more transfers waiting in the mailbox than it can run at one time. This exercises the queueing process which allows up to 5 processes (the default value for CONCUR_MAX) to run at one time. Therefore, we must choose a value larger than CONCUR_MAX. We will use 2*CONCUR_MAX = 10.

A simple script has been provided to run this test. From the testbed directory:

```
scripts/stress.pl SITE2
```

After the script completes, running DTS send 10 times, inform the receiving site.

Receiving site: After running "dts -g" monitor the dts process with the top command. You should see at most 5 such processes at any one time. After all have completed, compare the received files with the testbed (see SCRIPT_PATH).

As a shortcut, compare all directories associated with the stress test to the most recent directory. They should all match except for the VERSION file. Then, reconstruct the directory structure of the one all were compared to, and compare the the reconstructed directory to the testbed. If there is some confusion as to which directories are associated with the stress test, use "grep STRESS */VERSION" from the staging directory.

3.5 Failure/Recovery Tests

Test: RECO_BADPASS

This test simulates the failure that would occur when a sending site's ftp site is temporarily unavailable.

Edit the ad71031000.002/VERSION file to reflect the test name. From the testbed directory:

```
dts -se SITE2 -type PTEST -f filelist.txt
```

Inform the receiving site.

Receiving site:

Edit dts.sitelist, inserting an 'X' before the encoded password of the sending site. This renders the password invalid. Now, run "dts -g". It should fail and send and email to dtsops at your site. Edit dts.sitelist, removing the 'X'. In the log directory (DTS_IN in dts.config) run "dts -r" on the most recent file with ".mail" as its extension.

```
dts -r dts######_######G.mail
```

After completion, compare the received files with the testbed (see SCRIPT_PATH).

Test: FAIL_CORRUPT

This test simulates the failure when files are corrupted in transfer.

Edit the ad71031000.002/VERSION file to reflect the test name. From the testbed directory:

```
dts -se SITE2 -type PTEST -f filelist.txt
```

Change to the directory in the ftp area that was just "sent." You may want to look at the VERSION file to be sure. Corrupt a few files in that directory (e.g. `echo blah >> filename`). Inform the retrieving site.

Receiving site:

The "dts -g" operation should detect an error and send email to the operator account associated with your DTS.