MDMS File Exchange Interface Utilities

Reference Guide

Release 3.2.0

October 2001

MIPL Data Management System

Thomas Huang, MDMS
Thomas.Huang@jpl.nasa.gov

Contents

Introduction

FEI provides a set of command line utilities that can be used from a command shell like the Unix C (csh) or T (tcsh) shells. Each utility provides a simple FEI function. Together the utilities offer FEI functionality integrated with the shell. Each command is described below. Command syntax and examples are supplied with the description.

FEI3 Command-Line utilities list:

```
      feiKinit/feikinit
      feiKlist/feiklist
      feiKdestroy/feikdestroy
      feiKsrvtgt/feiksrvtgt
      feiKsrvtgt/feiksrvtgt
      feiAccept/feiaccept
      feiAdd/feiadd
      feiadmin
      feiCheck/feicheck
      feiCheckFiles/feicheckfiles
      feiCrc/feicrc
      feiDelete/feidelete
      feiDisplay/feidisplay
      feiFileMaker/feifilemaker
      feiFileTypes/feifiletypes
      feiGet/feiget
      feiHelp/feihelp
      feiList/feilist
      feiListGet/feilistget
      feiMakeClean/feimakeclean
      feiNotify/feinotify
      feiRename/feirename
      feiReplace/feireplace
      feiSubscribe/feisubscribe
```

Command Line Argument Conventions

Command names begin with the word "fei" followed by either command name with first letter an upper case character or all letters in the command name as lower case letter (Example feiAdd or feiadd). Almost all of the commands are then followed by an FEI file type and possibly a file name or file name expression. These two arguments are position dependent. All additional arguments are supplied as a keyword or a keyword and a value. Keywords can be supplied in any order. The command syntax is constructed to read like fractured English. Hopefully this makes it easier to remember command syntax. All examples below will use the upper case in the first letter of the command name. Here is an example that lists all "test" files beginning with the name "file..." that have modification dates later than October 24, 2000. We've asked for the long form of the list which includes information besides the file's name:

```
% feiList image1 'file*' after 2000-10-24 long
```

Since keywords are order independent, we can also specify the command like this:

```
% feiList image1 'file*' long after 2000-10-24
```

To see a command's syntax, type the command name alone and press `<enter>` or `<return>`. For example:

```
% feiList
Copyright 2000, The California Institute of Technology. All rights reserved.
FEI API Version 3.1.0, May 27, 1999
```

```
feiList <file type> ['<file name expression>']
        {[on|before|after <date-time>]|[between <date-time1> and <date-time2>]
         [long]}
```

Variable names are in `<angle brackets>` and optional arguments are in `[square brackets]`. Argument keywords, like `long` in the example, are in plain text. The set of order independent keywords is placed in `{curly braces}` following the command name and any order dependent arguments.

File Name Expressions

Several of the commands accept file name expressions. An expression can be a file name, like file1 or a regular expression like file* (all files whose names begin with file...) or * (all files). When using file name expressions you must be sensitive to where the evaluation of the regular expression takes place. For the operations feiAdd and feiReplace, the local shell evaluates the expression. For all other operations, like feiGet and feiList, the expression is sent to an FEI server for evaluation. To prevent your local shell from evaluating an expression being shipped to an FEI server, place single quotes around the expression.

```
% feiAdd image1 file*       # Evaluated by your local shell

% feiGet image1 'file*'     # Evaluated by the FEI server
```

File names can also be specified along with fully qualified path.

```
% feiAdd image1 /home/bar/data/file*
```

Dates and Time Formats

Several of the commands take an FEI date-time value. Date-time syntax follows the Consultative Committee for Space Data Systems Time Code Formats as specified in Blue Book, CCSDS 301.0-B-2, Issue 2, April 1990. (Don't worry about it. Just read the next couple of paragraphs.) As the following syntax shows, you can use year-day of year or year-month-day format with or without a time. Time can also be specified alone:

```
yyyy-ddd[Thh:mi:ss.ddd]
yyyy-mm-dd[Thh:mi:ss.ddd]
hh:mi:ss.ddd

yyyy : year {1900 - 2036}
mm   : month of year {01 - 12}
dd   : day of month {01 - 31}
ddd  : day of year {001 - 365} or {001 - 366}
hh   : hour {00 - 23}
mi   : minutes {00 - 59}
ss   : seconds {00 - 59}
ddd  : decimal part of second {000 - 999}
```

Note: Decimal part of second resolution is dependent on the database management system used to store FEI information. For records returned by Sybase SQL Servers, the resolution is 3 microseconds. Decimal parts of seconds are ignored when looking at local operating system times.

Note: All date parts have a prescribed number of digits. The month of February is represented by '02' and not '2', and 5-minute is '05' and not '5'.

The character ‘T’ separates date and time. You can supply a date or a time separately. A date without a time assumes a time of 00:00:00.000. A time without a date assumes the current date. Also, when you specify the time alone, don't precede it with the time separator 'T'.

```
2000-10-24T14:21    => 24 Oct 2000 14:21:00.000
2000-298T14:21      => 24 Oct 2000 14:21:00.000
2000-298            => 24 Oct 2000 00:00:00.000
14:21               => <today> 14:21:00.000
```

You can omit time parts to the right in a specification when the precision is not required. Zeros are supplied in these cases. The following times are the same:

```
15:00:00.000
15:00:00
15:00
15
```

Preliminaries

Before you can use FEI utilities you must have a Kerberos security ticket which FEI servers uses for authentication. Supply a user name and password to the kinit program to get a ticket:

```
% feiKinit bar@MIPS-DEV.JPL.NASA.GOV
Password for bar@MIPS-DEV.JPL.NASA.GOV: <password>
```

The utilities also need to get connection information, which is stored in an FEI domain file. The domain file is referenced by the value of the environmental variable FEI, so you'll need that defined as well. To see if your environment is set-up correctly use the command:

```
% feiFileTypes
```

This returns the set of file types accepted by the FEI commands. If the command doesn't work, your environment is probably not correct. The error message returned by `feiFileTypes` should tell you what needs fixing.

Kerberos Utilities

All these utilities are the FEI version of Kerberos utilities. The user has choice to use either these utilities or the Kerberors utilities.

feiKinit

```
feiKinit [Kerberos principal name]
```

This is not an actual utility, but an alias pointing to the Kerberos utility `kinit`. The utility is use for getting a Kerberos ticket, which is needed before an FEI server can authenticate you as a valid user of the system. You should always run this utility and obtain a ticket before running any program that accesses FEI.

Kerberos tickets are good for some number of hours, typically between 5 and 23 hours. The value for your system is defined in the file `/etc/krb5.conf`. (The value is actually in 5-minute chunks, although it's usually set for some number of whole hours.)

If the alias `feiKinit` is not defined for your environment, you can do this by finding the path to your copy of `kinit`, '`% which kinit`', and then setting up the alias, alias feiKinit `<path>/kinit` in your '`.cshrc'` file - or whatever file you use to define your environment. Alternatively, you can use kinit directly wherever you see `feiKinit` called for. (The alias is used because some systems have several copies of `kinit`, and they do not all operates the same.)

In the following example a user with Kerberos principal name "bar" obtains a Kerberos ticket.

```
% feiKinit bar@MIPS-DEV.JPL.NASA.GOV
Password for bar@MIPS-DEV.JPL.NASA.GOV: <bar’s kerberos password>
```

feiKlist

```
feiKlist
```

This utility allows the user to view information, as shown below, pertain to the Kerberos security ticket currently held by the user.

```
% feiKlist
Ticket cache: FILE:/tmp/krb5cc_1326
Default principal: bar@MIPS-DEV.JPL.NASA.GOV

Valid starting    Expires           Service principal
10/25/00 10:00:32 10/25/00 20:00:32 krbtgt/MIPS-DEV.JPL.NASA.GOV@MIPS-DEV.JPL.NASA.GOV
10/25/00 10:00:42 10/25/00 20:00:32 fei_d/foo.jpl.nasa.gov@MIPS-DEV.JPL.NASA.GOV
```

feiKdestroy

```
feiKdestroy
```

This utility allows the user to destroy the previously held Kerberos security ticket.

feiKsrvtgt

```
    feiKsrvtgt name instance [[realm] srvtab]
```

The purpose of this utility is to fetch and store Kerberos ticket-granting ticket using a service key. This utility is intended primarily for use in shell scripts and other batch-type facilities.

`feiKsrvtgt` retrieves a ticket-granting ticket with a lifetime of five minutes for the principal name.instance@realm (or name.instance@localrealm if realm is not supplied on the command line), decrypts the response using the service key found in the file srvtab (or in /etc/srvtab if srvtab is not specified on the command line), and stores the ticket in the standard ticket cache.

General Utilities

feiAccept

```
feiAccept <file type> for <add|replace|get|remove>
        {[output <path>] [encrypt] [crc] [replace|version]}
```

Keywords

- `for`: operation type `add`, `replace`, `get`, or `remove`. Default: add operation.
- `output`: output path for the `get` operation. Default: current directory.
- `encrypt`: encrypt[1] the data stream being transferred. Default: none.
- `crc`: calculate the CRC value for each file. Default: none.
- `replace`: this option works with the `get` operation to replace existing files within the specified output directory. Default: none.
- `version`: this option works with the `get` operation to create versioning of existing files within the specified output directory. If both replace and version are specified, then version is used. Default: none.

Accepts a list of file names from standard-in (stdin). Each file name is transferred to FEI. If no "`for`" clause is given, files are added. The "`for`" clause should be used if you want the files replaced. The "`for`" clause also accepts a value of add to accommodate typing habits.

feiAccept is handy when you want to pipe a set of file names to FEI and then have the files transferred. You can use it with Unix commands like ls and grep. For example:

```
% ls  file* | feiAccept image1
% ls | feiAccept image1 for replace

% grep -l '[Mm]ars' file* | feiAccept image1 compress

% feiAccept image1 encrypt for replace < file11
```

feiAdd

```
Batch: feiAdd <file type> <file name expression>...
        {[on|before|after <date-time>]|[between <date-time1> and <date-time2>]
         [encrypt] [crc]}

Batch: feiAdd <using> <option file>
Option File Format (per line):
        <file type> <file name>...
          {[on|before|after <date-time>]|[between <date-time1> and <date-time2>]
           [encrypt] [crc]}

Interactive: feiAdd <file type> {[encrypt] [crc]}
```

Keywords

- `on`: add files with creation times equal to the specified time value. Default: none.
- `before`: add file with creation times before the specified time value. Default: none.
- `after`: add files with creation times after the specified time value. Default: none.
- `between… and…`: add files with creation times between the specified time range. Default: none.
- `encrypt`: encrypt the data stream being transferred. Default: none.

[1] Due to the non-portable nature of the current encryption algorithm, the encryption feature is only supported on big endian systems. A portable solution is still under evaluation.

- `crc`: calculate the CRC value for each file. Default: none.
- `using`: keyword to supply a list of commands in an options file. Default: none.

Add one or more files to FEI. You can use a list of files or a file glob specification in place of the file name. Also, you can include an FEI date-time specification following one of the time keywords "before", "on", "after", "between… and…" along with a file glob specification if you want to limit the list to files modified since a specified time. The same rules can apply to the options file.

A file is not added if it already exists in FEI. Use `feiReplace` to do that.

Batch:
```
    % feiAdd image1 file1
    % feiAdd image1 file1 file2 file3
    % feiAdd image1 file*
    % feiAdd image1 file* after 2000-10-24
    % feiAdd image1 file* after 2000-10-24T14:21
    % feiAdd image1 file* between 2000-10-24T14:20 and 2000-10-24T14:30
```

Batch:

% cat addlist.txt
```
    image1 /home/bar/data/file0
    image1 file1 file2  # list of file names
    image1 file3 encrypt # encrypt the file
    image1 file* between 2000-255 and 2000-256
```

% feiAdd using addlist.txt

Note: If you use a file glob, the shell expands the expression into a list of files on the command line just as though you had listed them yourself.

You can also use `feiAdd` with just a file type. This puts you into interactive mode and you're prompted for each file you want to add. This mode also allows you to supply an FEI file name that is different from the one in your directory. In the following example, the local file file1 is named test1 when sent to FEI. If you don't supply a second file name, the local file name also given to the file within FEI. To exit interactive mode, just press the `<enter>` or `<return>` key at the prompt and the session ends. For example:

Interactive:
```
    % feiAdd image1
    Press the <enter> or <return> key to exit.
    feiAdd: [<local file name>] <FEI file name>

    feiAdd: file1 test1
    feiAdd: file2
    feiAdd: <enter>
    %
```

feiDelete

```
    Batch: feiDelete <using> <option file>
    Option File Format (per line):
            <file type> <file name>

    Interactive: feiDelete <file type> ["<file name expression>"]
```

Keyword
- `using`: keyword to supply a list of commands in an options file. Default: none.

`feiDelete` takes the file type and file name or expression uses the information to return a list of files. Each file in the list is then individually displayed as part of a prompt. If you want to delete the file, type the letter 'd'. To skip file deletion just press the `<enter>` or `<return>` key. When you have examined all the items in the list, the utility exits.

Batch:
```
% cat delList.txt
image1 file2
image1 file3
image1 file4
image1 file5
image1 file6

% feiDelete using delList.txt
```

Interactive:
```
% feiDelete image1 file1

Press <enter> or <return> to skip a file.  Type 'd' to delete the file.
Delete "file1": d
Deleting from file type "image1":
        "file1" deleted
```

You can also use a file glob expression in place of a file name.
Note: Since the expression is evaluated by FEI, it must be protected from shell interpretation by placing single quotes around the expression.

```
% feiDelete image1 "file*"

Press <enter> or <return> to skip a file.  Type 'd' to delete the file.
Delete "file0": d
Deleting from file type "image1":
        "file0" deleted

Delete "file7": d
Deleting from file type "image1":
        "file7" deleted

Delete "file8": d
Deleting from file type "image1":
        "file8" deleted
%
```

feiDisplay

```
feiDisplay <file type> <file name> {[encrypt] [crc]}
```

Keywords
- `encrypt`: encrypt the data stream being transferred. Default: none.
- `crc`: calculate the CRC value for each file. Default: none.

feiDisplay gets file and writes its contents to standard-out (stdout). The file name is also written, but it's written to standard-error (stderr). Use this utility when you want to display the contents of a file or pipe the contents to another program. In one of the following examples we show a file whose contents are piped to an image display program.

Note: Normally you only supply one file name to this utility; however, you can use a list or file glob if that's really what you want. (A file glob in this instance is evaluated by FEI, so it should be protected from shell evaluation by surrounding it with single quotes.)

```
% feiDisplay image1 doc1.txt
image1 file2
image1 file3
image1 file4
image1 file5
image1 file6
%
```

```
% feiDisplay test image1 | imageViewer
image displayed in "imageViewer" window
```

See also: `feiGet`.

feiHelp

feiHelp

Keyword

None

It displays the names of each FEI utility program and gives a short explanation of each utility's function. The last line reminds you that the syntax of each command is displayed by typing the utility's name with no arguments and then pressing `<enter>` or `<return>`.

feiFileTypes

```
feiFileTypes [<domain file name>]
```

Keyword

None

List the file types in your local FEI domain file. If no domain file name is supplied, FileTypes lists the file types in the default domain, $FEI/feiDomain. For example:

```
% feiFileTypes
File types in "feiDomain"
image1                           image2
image3                           image4
image5
%
```

feiGet

```
Batch: feiGet <file type> <file name expression>
        {[output <path>] [on|before|after <date-time>]|
         [between <date-time1> and <date-time2>]
         [encrypt] [crc] [replace|version]}

Batch: feiGet <using> <option file>
Option File Format (per line) :
        <file type> <file name expression>
          {[output <path>] [on|before|after <date-time>]|
           [between <date-time1> and <date-time2>]
           [encrypt] [crc] [replace|version]}

Interactive: feiGet <file type> {[encrypt] [crc] [replace|version]}
```

Keywords

- `output`: output path. Default: current directory.
- on: get files with FEI modification dates equal to the specified time value. Default: none.
- `before`: get files with FEI modification dates before the specified time value. Default: none.
- `after`: get files with FEI modification dates after the specified time value. Default: none.

- `between… and…`: get files with FEI modification dates between the specified time range. Default: none.
- `encrypt`: encrypt the data stream being transferred. Default: none.
- `crc`: calculate the CRC value for each file. Default: none.
- `replace`: replace existing files within the specified output directory. Default: none.
- `version`: create versioning of existing files within the specified output directory. If both replace and version are specified, then version is used. Default: none.
- `using`: keyword to supply a list of commands in an options file. Default: none.

Get's a file, or set of files if you supply a file glob as the file name expression. With a file glob, you can also supply an FEI date-time. Only files with FEI modifications values later than that are returned.

Note: A file glob is evaluated by FEI, so it must be surrounded by single quotes to protect it from shell evaluation. Get will not replace a file that already exists in your local directory unless you supply the keyword "with" followed by "skip" (the default), "replace" or "versions". "Replace" replaces the current local file while "versions" creates a version of the current local file and then writes the FEI file. (Versioning is done by appending an integer – it is really a time stamp value - to the file name.) The overwrite option "with" can come before or after a date-time specification if one is also included.

Batch :
```
% feiGet image1 file1
% feiGet image1 file1 replace
% feiGet image1 'file*'
% feiGet image1 '*' after 2000-10-24
% feiGet image1 'file*' version after 2000-298T14:20
% feiGet image1 'file*' after 2000-298T14:20 replace
% feiGet image1 'file*' between 2000-298T14:20 and 2000-298T14:50
```

Batch:
```
% cat getlist.txt
image1 file0 output /home/bar/data         # directing output
image1 file1 replace                       # use replacng
image1 file2 version                       # use versioning
image1 file3 encrypt                       # use encryption
image1 file* output /home/bar/data replace # use wild card

% feiGet using getlist.txt
Getting from file type "image1":
  1. Got "file0".

Getting from file type "image1":
  1. Got "file1".

Getting from file type "image1":
  1. Got "file2".

Getting from file type "image1":
  1. Got "file3" encrypted.

Getting from file type "image1":
  1. Got "file0".
  2. Got "file1".
  3. Got "file2".
  4. Got "file3".
  5. Got "file4".
  6. Got "file5".
  7. Got "file6".
  8. Got "file7".
  9. Got "file8".
 10. Got "file9".
 11. Got "file10".
```

You can get files in interactive mode by only supplying the file type. At the prompt, supply the FEI file name. If you want to use a different name for the local copy of the file, use the `feiRename` Command. Exit the utility from interactive mode by pressing the `<enter>` or `<return>` key at the prompt.

```
% feiGet image1
Press the <enter> or <return> key to exit.
feiGet: <file name expression>
        {[output <path>] [on|before|after <date-time>]|
         [between <date-time1> and <date-time2>]}

feiGet <image1>: file0
1. Got "file0".
feiGet <image1>: <enter>
%
```

See also: `feiDisplay, feiListGet`.

feiList

```
feiList <file type> ['<file name expression>']
        {[on|before|after <date-time>]|[between <date-time1> and <date-time2>]
         [long]}
```

Keywords

- `on`: get files with FEI modification dates equal to the specified time value. Default: none.
- `before`: get files with FEI modification dates before the specified time value. Default: none.
- `after`: get files with FEI modification dates after the specified time value. Default: none.
- `between… and…`: get files with FEI modification dates between the specified time range. Default: none.
- `long`: lists output with FEI modification date and file size in bytes. Default: none.

This utility gets a list of file names from FEI.

- If the file name expression is omitted, all files of the specified type are listed.
- If a file glob is used, it's evaluated by FEI and matching file names are returned.

Note: A file glob is evaluated by FEI, so it must be surrounded by single quotes to protect it from shell evaluation.

```
% feiList image1
file0
file1
file2
file3
file4

% feiList image1 'file*'
file0
file1
file2
file3
file4

% feiList image1 "file*" long

    1.  2000-298T14:21:27.256,            5, file0
    2.  2000-298T14:21:27.426,            5, file1
    3.  2000-298T14:21:27.550,            5, file2
    4.  2000-298T14:21:27.686,            5, file3
    5.  2000-298T14:21:27.816,            5, file4
    6.  2000-298T14:55:06.753,       151292, file5

% feiList image1 "file*" after 2000-298T14:30 long

    1.  2000-298T14:55:06.753,       151292, file5
```

See also: `feiNotify`.

feiListGet

```
feiListGet <file type> ['<file name expression>']
        {[output <path> [on|before|after <date-time>]
         |[between <date-time1> and <date-time2>]
         [encrypt] [crc] [replace|version] [long]}
```

Keywords

- `output`: output path. Default: current directory.
- on: get files with FEI modification dates equal to the specified time value. Default: none.
- `before`: get files with FEI modification dates before the specified time value. Default: none.
- `after`: get files with FEI modification dates after the specified time value. Default: none.
- `between…` `and…`: get files with FEI modification dates between the specified time range. Default: none.
- `encrypt`: encrypt the data stream being transferred. Default: none.
- `crc`: calculate the CRC value for each file. Default: none.
- `replace`: replace existing files within the specified output directory. Default: none.
- `version`: create versioning of existing files within the specified output directory. If both replace and version are specified, then version is used. Default: none.
- `long`: lists output with FEI modification date and file size in bytes. Default: none.

This utility gets a list of file names from FEI based on the command line arguments. Each file name is displayed as a prompt. To get the file, type the letter 'g' following the prompt. To skip it, press the `<enter>` or `<return>` key. When the list is exhausted, the utility exits.

```
% feiListGet image1 "file*"
Press the <enter> or <return> key to skip file.
Type 'g' to get the file.

feiListGet/<image1>@Get "file0": g
  1. Got "file0".

feiListGet/<image1>@Get "file1": g
  1. Got "file1".

feiListGet/<image1>@Get "file2": <enter>


% feiListGet image1 "file*" between 2000-298T14:30 and  2000-298T15:00
Formated 'between' 2000-298T14:30:00 'and' 2000-298T15:00:00

Press the <enter> or <return> key to skip file.
Type 'g' to get the file.

feiListGet/<image1>@Get "file5": g

  1. Got "file5".
```

See also: `feiList` and `feiGet`.

feiNotify

```
feiNotify <file type>
        {[output <path>] [restart] [using <option file>]}
Option File Format:
        logFile <file name>
        displayMessages
        bell
        mailMessageTo <email address>
        mailReportTo <email address>
        mailReportAt <list of times of day>
```

```
            limitMessages
            invoke <command>
```

Keywords

- `output`: output path of the restart file. Default: current directory.
- `restart`: flag to restart the previous notification session. Default: none.
- `using`: to supply an input options file. See 'The Options File' below. Default: none.

`feiNotify` is a type of subscription that returns information on each file of the specified type as it received by FEI. (If the long keyword is included, FEI modification date and file size are displayed along with the file name.) The output looks similar to `feiList`. Use this, and not `feiList`, which performs a database query for each list of file names retrieved, when you want to receive the names of newly arrived files.

Restart

When you start a notification session, a restart file is automatically created for you in your current working directory. The naming convention for the file is `<file type>.notify`. If you supply a value for the '`output`' keyword (name of the directory), the `feiNotify` utility will lookup and output its restart file to the specified directory. If '`output`' is not specified, the default is the current working directory. If a subscription session is stopped for any reason, you can restart it at the place it left off using the state information saved in the restart file. Do this by including the restart keyword:

```
   % feiNotify image1 restart
```

Note: Don't try to alter the contents of a restart file. Only bad things will result.

Note: You'll find a second restart file named the same as the first with a trailing tilde character. This is a backup in case the restart file is damaged.

Note: You can restart the subscription from any directory as long as you supply the pathname of the directory where the restart file is located. You don't have to move the restart file in a directory to restart the subscription.

Reconnect

If `feiNotify` looses its connection to the target FEI Server, it will keep trying to reconnect at regular intervals of 1, 5, 10, and 15 minutes. After it reaches the 15-minute interval it continues to try at that interval until it reconnects or until you stop the program.

If you have enabled logging or email (discussed below in The Options File), `feiNotify` will send a message each time it tries to connect until it is successful. To suppress this, include `limitMessages` as an option. Then you only learn when a connection was lost and regained, but nothing in between.

Shutdown

The notification sessions run until they receive a SIGINT or SIGTERM signal, usually as a result of `Control-C` or `kill` request. This will cause the sessions to shutdown gracefully. On Unix systems, if feiNotify is running in the foreground, you can use <`Control-C`> to issue a SIGINT to the process. (This functionality is not supported under Microsoft Windows. The results are indeterminate according to MS.) On any system, use the `kill` command followed by the process ID (pid) of the process to send a SIGTERM signal to the process. For Unix systems, use this when the process in running in the background.

The Options File

An options file contains additional parameters that you can supply to a subscription session. All of the parameters are optional. The parameter names are case insensitive but the values following them are not.

Option file parameters:

1. `bell` : Rings a bell.
2. `logFile <file name>` : Error and other messages are normally sent to stderr. Supply this keyword along with the name of a file to have messages written to a log file instead.
3. `displayMessages` : When a log file is specified, error and other messages are not displayed to stderr. To override this behavior, use this parameter. The parameter takes no value.
4. `mailReportTo <e-mail address>` : An email address to which messages are sent. The subject of the message includes the subscription file type for easy identification. For example: "Re: subscription to image".
5. `mailReportAt <time of day>` : Specify a time of day, without a date, and the names of newly arrived files will be sent to you at that time using your mailReportTo option. (Ignored if mailReportTo and mailReportAt are not specified.
6. `mailMessagesTo` : List of e-mail addresses to which you wish to send administrative messages about FEI.
7. `limitMessages` : A parameter that takes no value. If included, Subscribe only notifies you when you loose a connection and when you regain it again. But any attempts in between go unrecorded.
8. `invoke <command>` : Invokes a command each time a new file is written to the local system. The command might do something like moving a file to another directory, image conversion, or image display. A command can be anything executable from your system's command processor, or shell, so C, C++ or Perl programs are appropriate. The command can contain three variables that Subscribe will replace before the command in invoked:
   - `$fileName` : The name of the file just arrived.
   - `$rootName` : The name of the file just arrived minus any trailing extention preceeded by a dot character. For example the root of "file1.extension" is "file1".
   - `$fileType` : The subscription session's file type.

An options file can contain comment and blank lines. Each parameter must be placed on a line by itself. Comments start with the character '#' and must be the first non-blank character on a line.

Here is a the beginning of a notify session:

```
% cat notify.opt
logfile notify.log                   # generate a log file
displayMessages                      # output all messages
bell                                 # ring when new image is posted
mailMessageTo bar@mipl.jpl.nasa.gov  # send administrative messages
mailReportTo bar@mipl.jpl.nasa.gov   # send report
mailReportAt 15:25:00                # time to mail
invoke feiList image1 $fileName long # execute command


% feiNotify image1 using notify.opt
FEI Information on 2000-298T15:25:34
PID:10325.  Listing file of type "image1".

FEI Information on 2000-298T15:26:33
received "file0" 5 bytes 2000-298T15:26:33.223

FEI Information on 2000-298T15:26:33
Invoke command: "feiList image1 file0 long"


    1.  2000-298T15:26:33.223,            5, file0
FEI Information on 2000-298T15:26:33
received "file1" 5 bytes 2000-298T15:26:33.543

FEI Information on 2000-298T15:26:33
Invoke command: "feiList image1 file1 long"


    1.  2000-298T15:26:33.543,            5, file1
```

```
<Control-C>
FEI Information on 2000-298T15:27:18
Disconnected from FEI.  Start shutting down.
FEI Information on 2000-298T15:27:19
Shutdown complete.  Normal termination.
```

See also: `feiSubscribe`, `feiList`.

feiRename

```
Batch: feiRename <file type> <old file name> <new file name>

Batch: feiRename <using> <file name>
Option File Format (per line):
        <file type> <old file name> <new file name>

Interactive: feiRename <file type>
```

Keyword

- `using`: keyword to supply a list of commands in an options file. Default: none.

Rename a file in FEI. If you only supply a file type, you enter interactive mode so you can supply sets of old and new file names. Exit interactive mode by pressing <enter> or <return> at the prompt.

Batch:

```
% feiRename image1 file0 fido_0
```

Batch:

```
% cat rename.txt
image1 file1 fido_1

% feiRename using rename.txt

% feiList image1
fido_0
fido_1
```

Interactive:

```
% feiRename image1
Press the <enter> or <return> key to exit.
feiRename: <old file name> <new file name>

feiRename: fido_0 file0
feiRename: fido_1 file1
feiRename: <enter>

% feiList image1
file0
file1
```

feiReplace

```
Batch: feiReplace <file type> <file name expression>...
        {[on|before|after <date-time>]|[between <date-time1> and <date-time2>]
         [encrypt] [crc]}

Batch: feiReplace <using> <option file>
Option File Format (per line):
        <file type> <file name>...
          {[on|before|after <date-time>]|[between <date-time1> and <date-time2>]
           [encrypt] [crc]}
```

```
Interactive: feiReplace <file type> {[encrypt] [crc]}
```

Keywords

- `on`: get files with FEI modification dates equal to the specified time value. Default: none.
- `before`: get files with FEI modification dates before the specified time value. Default: none.
- `after`: get files with FEI modification dates after the specified time value. Default: none.
- `between… and…`: get files with FEI modification dates between the specified time range. Default: none.
- `encrypt`: encrypt the data stream being transferred. Default: none.
- `crc`: calculate the CRC value for each file. Default: none.
- `using`: to supply a list of commands in an options file. Default: none.

This command works just like feiAdd but a file is replaced in FEI if it already exists. Here are some examples showing command line keywords:

Batch:
```
% feiReplace image1 file0
Replacing to file type "image1":
  1. Replaced "file0".

% feiReplace image1 file0 file1
Replacing to file type "image1":
  1. Replaced "file0".
  2. Replaced "file1".

% feiReplace image1 file*
Replacing to file type "image1":
  1. Replaced "file0".
  2. Replaced "file1".

% feiReplace image1 file* between 2000-298T15:40 and 2000-298T15:50
Formated 'between' 2000-298T15:40:00 'and' 2000-298T15:50:00

Replacing to file type "image1":
Formated file <file0> time: 2000-298T15:44:04
Formated file <file1> time: 2000-298T15:44:04
  1. Replaced "file0".
  2. Replaced "file1".
```

Interactive:
```
% feiReplace image1
Press the <enter> or <return> key to exit.
feiReplace: [<local file name>] <FEI file name>

feiReplace: file0 fido_0
Replaced "fido_0".
feiReplace: file1 fido_1
Replaced "fido_1".
feiReplace: <enter>
%
```

feiSubscribe

```
feiSubscribe <file type>
        {[output <path>] [restart] [using <option file>]}
Option File Format:
        crc
        encrypt
        replace
        version
        logFile <file name>
        displayMessages
        bell
        mailMessageTo <email address>
        mailReportTo <email address>
        mailReportAt <list of times of day>
        limitMessages
        invoke <command>
```

Keywords

- `output`: output path of the restart file and the subscribed data. Default: current directory.
- `restart`: flag to restart the previous notification session. Default: none.
- `using`: to supply an input options file. See 'The Options File' below. Default: none.

Start a subscription session for a file type. For example:

```
% feisubscribe image1
```

Restart

When you subscribe, a restart file is automatically created for you in your current working directory. The naming convention for the file is `<file type>.subscribe`. If a subscription session is stopped for any reason, you can restart it at the place it left off using the state information saved in the restart file. Do this by including the restart keyword:

```
% feiSubscribe image1 restart
```

Note: Don't try to alter the contents of a restart file. Only bad things will result.

Note: You'll find a second restart file named the same as the first with a trailing tilde character. This is a backup in case the restart file is damaged.

Note: You can restart the subscription from any directory as long as you supply the pathname of the directory where the restart file is located. You don't have to move the restart file in a directory to restart the subscription.

Reconnect

If `feiSubscribe` looses its connection to the target FEI Server, it will keep trying to reconnect at regular intervals of 1, 5, 10, and 15 minutes. After it reaches the 15-minute interval it continues to try at that interval until it reconnects or until you stop the program.

If you have enabled logging or email (discussed below in The Options File), `feiSubscribe` will send a message each time it tries to connect until it is successful. To suppress this, include `limitMessages` as an option. Then you only learn when a connection was lost and regained, but nothing in between.

Shutdown

Subscription sessions run until they receive a SIGINT or SIGTERM signal. They then shutdown gracfully. On Unix systems, if feiSubscribe is running in the forground, you can use `<control-c>` o issue a SIGINT to the process. (This functionality is not supported by Microsoft. The results are indeterminate says MS.) On any system, use the kill command followed by the pid of the process to send a SIGTERM signal to the process. For Unix systems, use this when the process in running in the background.

The Options File

An options file contains additional parameters that you can supply to a subscription session. All of the parameters are optional. The parameter names are case insensitive but the values following them are not.

Option file parameters:

1. `encrypt` : Encrypt files as they are sent.
2. `crc` : Compute CRC value for files as they are sent.
3. `bell` : Rings a bell.
4. `replace` : Replace files that already exists.
5. `version` : Create a version of any file that already exists.

6. `logFile <file name>` : Error and other messages are normally sent to stderr. Supply this keyword along with the name of a file to have messages written to a log file instead.
7. `displayMessages` : When a log file is specified, error and other messages are not displayed to stderr. To override this behavior, use this parameter. The parameter takes no value.
8. `mailReportTo <e-mail address>` : An email address to which messages are sent. The subject of the message includes the subscription file type for easy identification. For example: "Re: subscription to image".
9. `mailReportAt <time of day>` : Specify a time of day, without a date, and the names of newly arrived files will be sent to you at that time using your mailReportTo option. (Ignored if mailReportTo and mailReportAt are not specified.
10. `mailMessagesTo` : List of e-mail addresses to which you wish to send administrative messages about FEI.
11. `limitMessages` : A parameter that takes no value. If included, Subscribe only notifies you when you loose a connection and when you regain it again. But any attempts in between go unrecorded.
12. `invoke <command>` : Invokes a command each time a new file is written to the local system. The command might do something like moving a file to another directory, image conversion, or image display. A command can be anything executable from your system's command processor, or shell, so C, C++ or Perl programs are appropriate. The command can contain three variables that Subscribe will replace before the command in invoked:
    - `$fileName` : The name of the file just arrived.
    - `$rootName` : The name of the file just arrived minus any trailing extention preceeded by a dot character. For example the root of "file1.extension" is "file1".
    - `$fileType` : The subscription session's file type.

An options file can contain comment and blank lines. Each parameter must be placed on a line by itself. Comments start with the character '#' and must be the first non-blank character on a line.

A pathname name can also be appended in front of the option file if the option file resides in a directory other than the current directory.

If the user decides to receive files from the server in a directory other than the current directory by supplying a pathname on the command line and also uses an option file which might contain an Invoke command that will invoke on the incoming files from the server, the user is now required to supply the same path name in front of the `$fileName` variable in the option file until this problem is corrected in the software. The placement of the pathname in front of the `$fileName` variable is the temporary workaround for now.

Here's subscription command using an option file. The file's contents are printed below the command.

```
% cat subscribe.opt
replace                            # replace files
logfile notify.log                 # generate a log file
displayMessages                    # output all messages
bell                               # ring when new image is posted
mailMessageTo bar@mipl.jpl.nasa.gov  # send administrative messages
mailReportTo bar@mipl.jpl.nasa.gov   # send report
mailReportAt 15:25:00              # time to mail
invoke feiList image1 $fileName long # execute command

% feiSubscribe image1 using subscribe.opt
FEI Information on 2000-298T16:10:31
PID:10497. Subscribing to "image1" file type.

FEI Information on 2000-298T16:12:04
received "fido_0" 6 bytes 2000-298T16:12:04.543

FEI Information on 2000-298T16:12:04
Invoke command: "feiList image1 fido_0 long"
```

```
    1.  2000-298T16:12:04.543,            6, fido_0
FEI Information on 2000-298T16:12:13
received "fido_1" 6 bytes 2000-298T16:12:13.363

FEI Information on 2000-298T16:12:13
Invoke command: "feiList image1 fido_1 long"


    1.  2000-298T16:12:13.363,            6, fido_1
FEI Information on 2000-298T16:14:18
Disconnected from FEI.  Start shutting down.
FEI Information on 2000-298T16:14:19
Shutdown complete.  Normal termination.
```

See also: `feiNotify`, `feiCheckFiles`, `feiGet`, `feiListGet`.

Administrative Utility

feiadmin

```
feiadmin <server name> showusers
feiadmin <server name> hotboot
feiadmin <server name> shutdown <level {0|1|2}>
```

Keywords

- showusers: lists the current connected users.
- hotboot: notifies the server to reconfigure itself without shutting down the server.
- shutdown: signals the server to shutdown. Level 0 signals the server to shutdown immediately.

Show Users

The '`showusers`' keyword displays a list of user principal names, thread ID, current command, and time of last command execution.

Hot Boot

The 'hotboot' keyword tells the server to read its configuration files again and reconfigure based on the information found. All connections are maintained during this time, but all file transfers are suspended until the process is completed. This command will not return until the Hot Boot process is complete.

Shutdown

- Level 0: Immediate shutdown. Any transfers in progress are interrupted and all connections are closed immediately.
- Level 1: Finish any transfers currently in progress but accept no new ones. Terminate once all current transactions have finished.
- Level 2: Same as Level 1, but all subscription queues are read by subscribers before termination. No new files are added to subscription queues once the termination notice is accepted by the server.

This command is used to shutdown the FEI server. The level value indicates the type of shutdown. Note: if you use a level 0 shutdown, you can end-up with partial files in FEI. Check the status of files in the FEI database table files using the stored procedure `showBadFiles` after this type of shutdown.

Testing Utilities

feiFileMaker

```
feiFileMaker <file type>
        {[for (add | replace)] [using <content file name>]
         [<startWith <index>] [delay <seconds>] [encrypt] [crc]}
```

Keywords

- `for`: operation type `add` or `replace`. Default: add operation.
- using: supply a data file to transfer. Default: none.
- `startWith`: file name index starting number. Default: 0.
- `delay`: time delay between each add/replace operation. Default: 0.
- `encrypt`: encrypt the data stream being transferred. Default: none.
- `crc`: calculate the CRC value for each file. Default: none.

Creates the name of a test file and adds the file to FEI using the name of the supplied file type. If the keyword replace is included, the file is replaced if it already exists in FEI; otherwise it's added just as in the default case. Generated file names are made unique by adding an index number using the syntax:

```
   file<index>
```

The first index number to use can be supplied following the keyword `startWith`. If it's not supplied, 1 is used. By default, the content of the file is the file name. Another file's content will be copied and used instead if the content provider's name is supplied following the content keyword. Normally, `feiFileMaker` adds files as quickly as possible. You can add a delay between transactions by supplying the number of seconds to delay following the keyword delay. Examples:

```
% feiFileMaker image1
% feiFileMaker image1 startWith 1000 using fido_0
% feiFileMaker image1 startWith 1000 using fido_0 delay 10
```

FileMaker runs until stopped by a signal. Use SIGTERM or SIGINT.

See also: `feiMakeClean`

feiMakeClean

```
feiMakeClean <file type> <file name expression>
```

Keyword

None

Removes all of the files from FEI for the name of the file type supplied. For example:

```
% feiMakeClean image1
```

Validation Utilities

feiCheck

```
feiCheck [<FEI domain file name>]
```

Keyword

None

Used to check the correctness and completeness of an FEI client installation. See the Client Installation Guide for more information. The command normally is used to check the installation for the default domain, in which case no parameter is used. If a parameter is provided, it is the name of an FEI domain file used during the checking process:

```
% feiCheck
Using FEI domain file "/home/bar/feiDomain".
Trying connection to file type "image1"
OK
Trying connection to file type "image2"
OK
Trying connection to file type "image3"
OK

% FeiCheck tokyoDomain
```

feiCheckFiles

```
feiCheckFiles <file type> [<file name expression>]
        {[on|before|after <date-time>]|
         [between <date-time1> and <date-time2>] [long]}

feiCheckFiles <using> <file name>
Option File Format (per line) :
        <file type> [<file name expression>]
          {[on|before|after <date-time>]|
           [between <date-time1> and <date-time2>] [long]}
```

Keyword

- `on`: check files with FEI modification dates equal to the specified time value. Default: none.
- `before`: check files with FEI modification dates before the specified time value. Default: none.
- `after`: check files with FEI modification dates after the specified time value. Default: none.
- `between… and…`: check files with FEI modification dates between the specified time range. Default: none.
- `long`: lists output with FEI modification date and file size in bytes. Default: none.
- `using`: to supply a list of commands in an options file. Default: none.

Gets a list of files from FEI based on the input parameters supplied, and then, for each file name returned, feiCheckFiles tries to open the file in the current working directory. It prints a message based on the outcome of that attempt. (feiCheckFiles opens and closes files but does nothing to the content of a file.)

Use this utility if you think you may be missing some copies of files found in FEI. Once you've determined which files these are, use feiGet, or feiListGet to get them.

```
% feiList image1
% feiCheckFiles image1 'file*' after 1997-02-24
```

feiCrc

```
feiCrc <file name>
```

Keyword

None

Computes and display the CRC[2] of the local file

```
% feiCrc file1
```

[2] Due to the non-portable nature of the current CRC algorithm, the CRC feature is only supported on big endian systems. A portable solution is still under evaluation.