FILE EXCHANGE INTERFACE

Multimission Ground Systems and Services Office – Instrument Operations Systems

USERS GUIDE

Version 5 - Release 2.2.6

THOMAS HUANG

CURRENT CUSTODIANS: HELEN MORTENSEN

NICHOLAS TOOLE

January 2011

National Aeronautics and Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

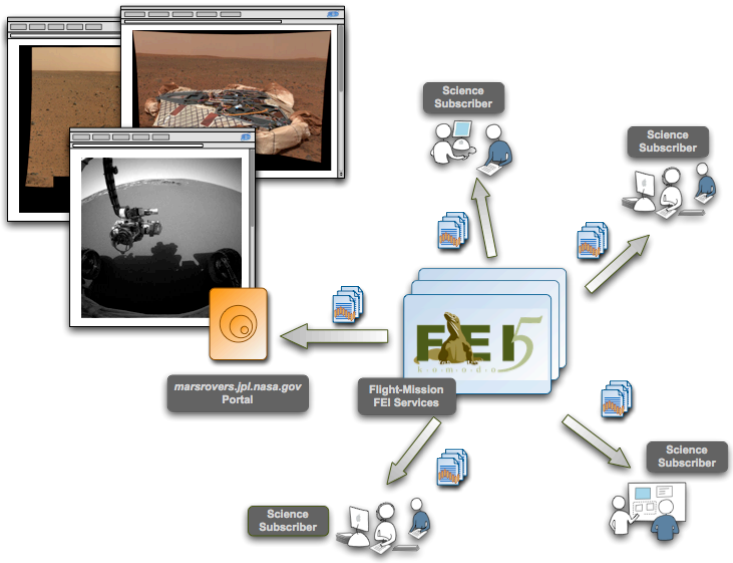
Document: D-31530 Rev. 5

4800 Oak Grove Drive, Pasadena, CA 91109, U.S.A. • telephone: 818.354.7974 • fax: 818.393.6962 • www-mipl.jpl.nasa.gov/mdms

Table of Contents

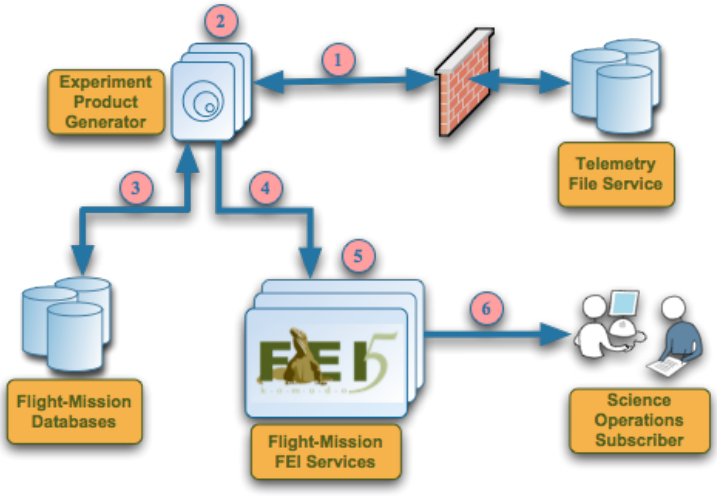
1. Introduction

The File Exchange Interface (FEI5) service offers secure file transaction, store, transport, and management services. FEI5 is the science data product management and distribution service used by most major space missions. The service offers a transaction-oriented approach in file management. That is, all concurrent updates to the same data product are prohibited. All uncommitted file transactions are automatically rolled back. The latest distribution, FEI5 software code name Komodo, is a complete redesign from its predecessors, which adopts the latest computing technologies and standards.



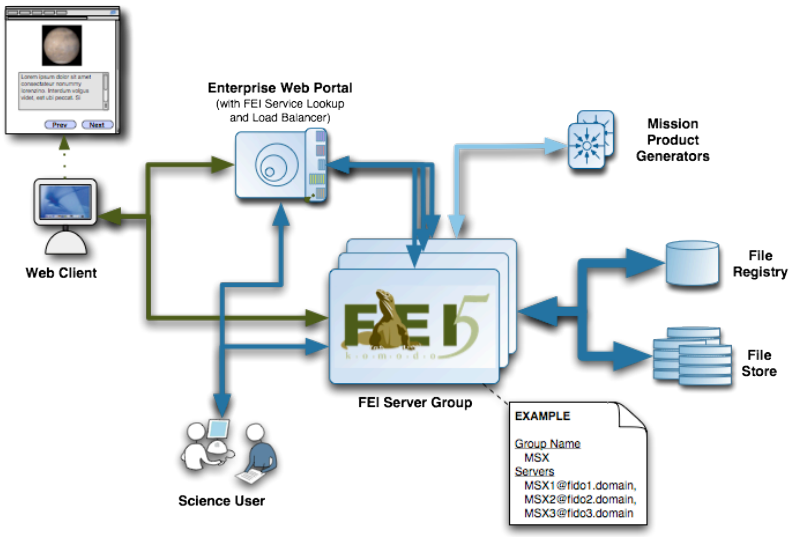
Architecture

FEI5 is a client-server application that is driven by the back-end data store and file system. The term data store is used and not database, because the architecture enables the service to be decoupled from the specified back-end data store. The service uses standard input/output to interact with the back end file system to enable portable support across various file systems. The architecture's main key attributes are plug-able security selection, transparent location of servers and clients, automatic file archiving, failover capability when multiple servers are in use, and high-speed file transfer. The diagram also illustrated the portability of the FEI5 software that is being supported on all current popular operating systems.



Server Groups

Traditional client server architecture consists of one server and many clients. The architecture is limited by the server machine performance and physical resource. High-profile missions always attract a large numbers of users. A scalable data distribution service must be able to handle large number of concurrent requests. In the case of a file transaction service, this also means to be able to handle large number of concurrent transactions as well. Since there are physical connection limits to any single host machine, the solution that FEI5 offers is to have a group of associated FEI5 servers, also known as server groups. File transactions performed by each FEI5 server, need to be reflected to and be visible by other FEI5 servers of the same server group. This is the durable property of ACID (Atomicity, Consistency, Isolation, and Durability). A federation model is used to utilize the service registry as the transaction communication channel among the servers. With a server grouping model, a mission can have a file distribution architecture that maps many clients to many servers. To the client, the server group is just a name associated to a collection of physical host machines.



File Types

FEI5 organizes files into file types. To transfer a file, or set of files, the client is only required to supply the file type and the files of that type to transfer. The location of the target FEI5 server and file directory are transparent to the user. FEI5 server administrators can change the FEI5 server configuration and file directory organization without impacting client applications. A good analogy of file types is mailboxes within an organization. The mail-stop, usually consists of a building and a room number, is a physical location where a set of mailboxes are hosted, much like the physical location of FEI5 server. The mailboxes are tagged with distinct employee names. Each mailbox can only be used to store mail for an individual employee.

2. Application Programs

The FEI5 software distribution is bundled with a GUI, command line, and interactive application programs. These three tools provide flexibility for using FEI5. The command line tools are often used in shell scripts to automate processing. The 'fei5gui' and 'fei5' interactive tools provide interactive access to the data in a single user session. The GUI application provides a simple user interface to the data.

Note: Command line tools start up a Java virtual machine (JVM) each time a command is issued. For tasks that involve multiple command-line operations, this can add some overhead to the response time. The 'fei5gui' and 'fei5' applications start up a JVM once and response time to data is faster.

APPLICATION	INTERACTIVE	CMD LINE	GUI	DESCRIPTION
fei5	X	X		The general FEI5 user session application, that allows the user to add, get, delete, replace files… etc.
fei5accept		X		Command to take a list of file names from standard-in and perform the add|replace|delete|get transaction for each file.
fei5add		X		Command to add/register file(s) to the FEI5 server.
fei5changepassword		X		Command to change the user's password for a non-LDAP configured system.
fei5check		X		Command to check the user connection to the FEI5 server for each file type listed on the client installation's domain file.
fei5checkfiles		X		Command to compare the file in a local directory with files listed in the FEI5 server's file type.
fei5comment		X		Command to add a comment to a file.

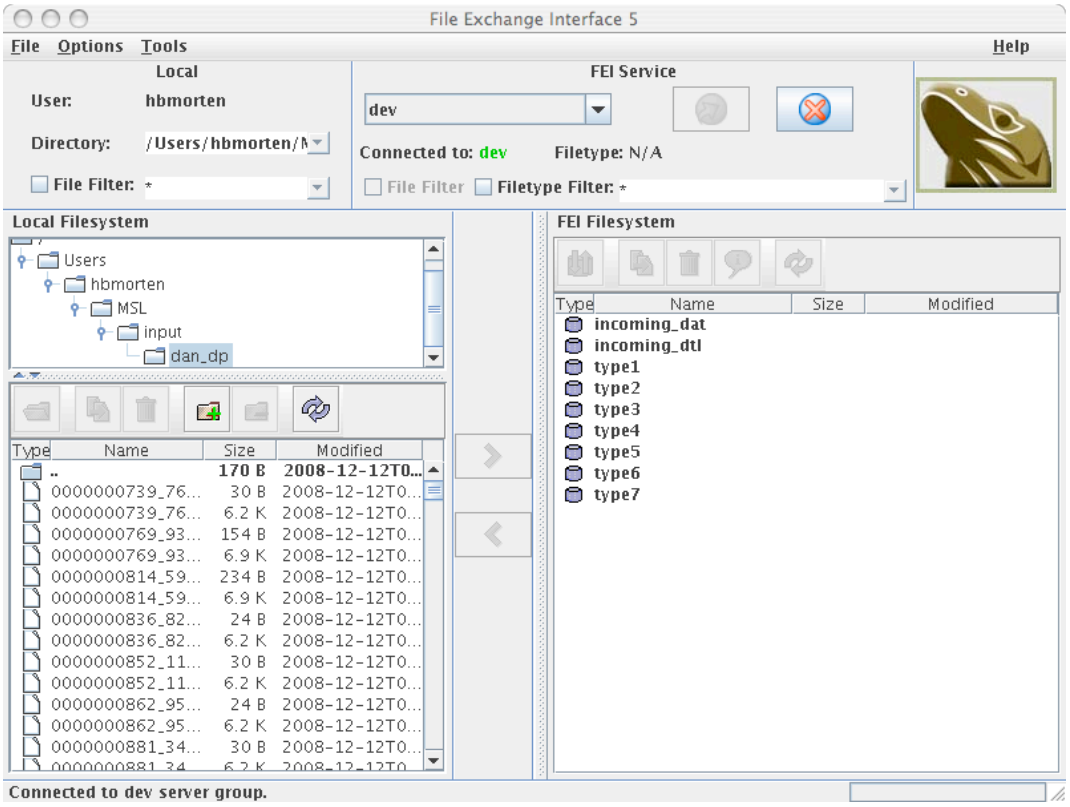
APPLICATION	INTER AC-TIVE	CMD LINE	GUI	DESCRIPTION
fei5crc		X		Command to compute and display the CRC values for local files.
fei5delete		X		Command to delete file(s) from the FEI5 server.
fei5display		X		Command to get a file from the FEI5 server and display its contents to stan-dard-out.
fei5filetypes		X		Command to query and list file types and server information in the client in-stallation's domain file.
fei5get		X		Command to get file(s) from the FEI5 server.
fei5guardian		X		An example utility command that will invoke a criteria test on the filename that must return success before running a FEI5 command-line operation.
fei5gui	X		X	The graphical GUI FEI5 user session application.
fei5kdestroy		X		Command to destroy the persistent login information.
fei5kinit	X			Command to create a persistent user login file.
fei5klist		X		Command to list information on the persistent user login file.
fei5list		X		Command to list files within a FEI5 server file type.
fei5locktype		X		Command to lock a filetype. Requires locktype capabilities.

APPLICATION	INTER AC-TIVE	CMD LINE	GUI	DESCRIPTION
fei5makeclean		X		Command to cleanup a file type on the FEI5 server.
fei5notify		X		Command to subscribe and return information on each newly modified or added file of the specified type.
fei5reference		X		Command to add a reference to a VFT.
fei5register		X		Command to register a record in the FEI5 database to an existing file located on a file system accessible by the server. This capability is not available in the fei5gui or fei5 applications.
fei5rename		X		Command to rename a file on the FEI5 server.
fei5replace		X		Command to replace a file on the FEI5 server.
fei5subscribe		X		Command to subscribe to a file type to have each newly modified or added file delivered automatically.
fei5unlocktype		X		Command to unlock a filetype. Requires locktype capabilities.
fei5unregister		X		Command to unregister the record in the FEI5 database without deleting the file from the underlying file system.

3. GUI Client

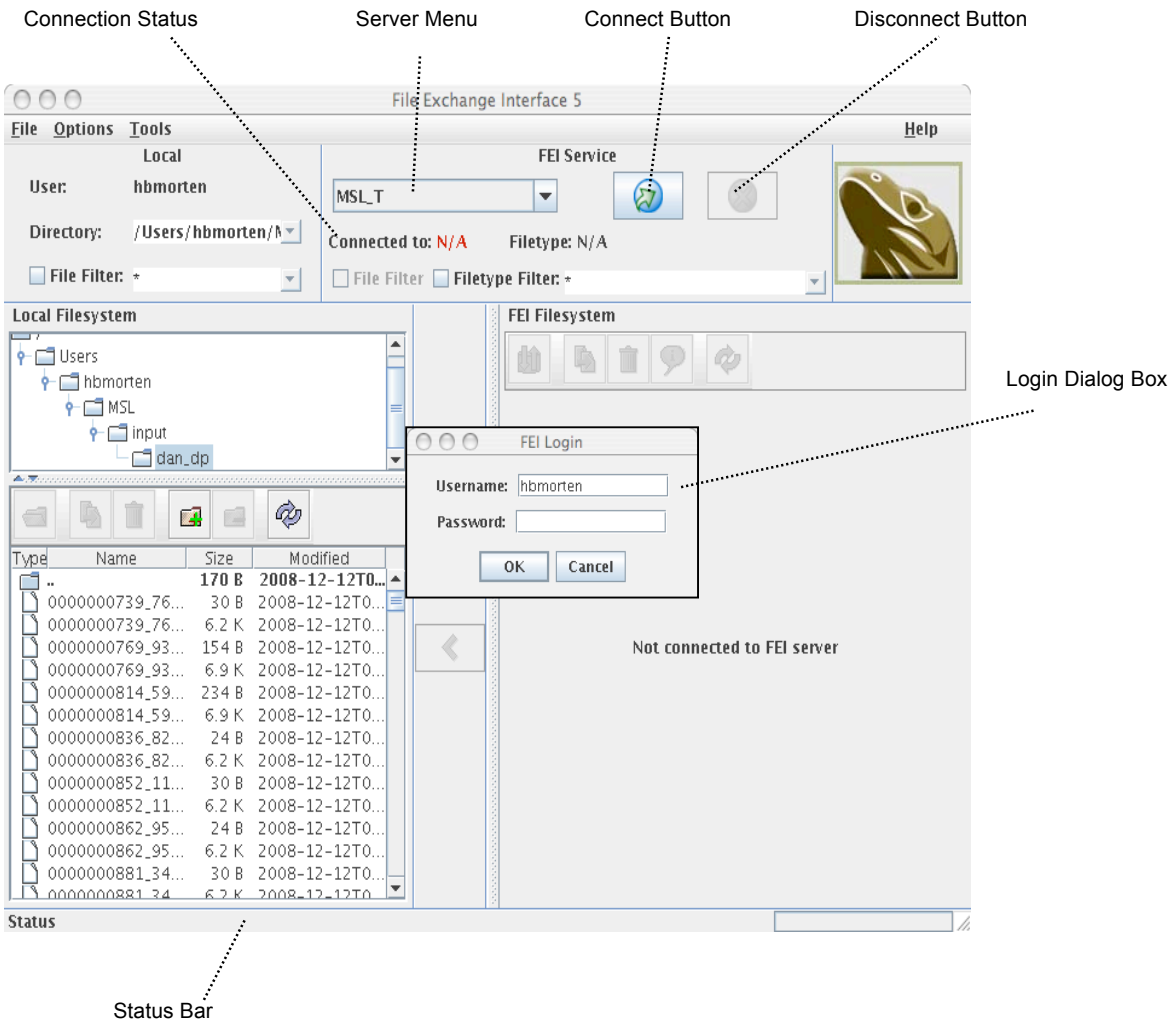
In release 1.6.2, the FEI5 development team introduced a beta release of the FEI5 Graphical User Interface (GUI) client. The GUI client provides point-and-click interface to the FEI5 service. It offers a single desktop interface to all server groups. Users can navigate between server groups and file types without having to open multiple application window sessions. File retrieval and delivery transaction can be performed in drag-and-drop fashion. Its powerful regular expression engine enables users to form complex filtering on local and remote files. In release 1.7.7 a subscription manager and log panel were added to the GUI client. The subscription manager enables the user to have multiple subscription sessions among available server groups within a single GUI session. The log panel logs all client messages with support for message search and formatted printing. The GUI client is developed using pure Java technology. It is available on all supported platforms.

To start the FEI5 GUI client, type **`fei5gui`** from the command line prompt. Select the desired server from the drop down menu in the FEI Service area and hit the Connect Button to start a session with a server.



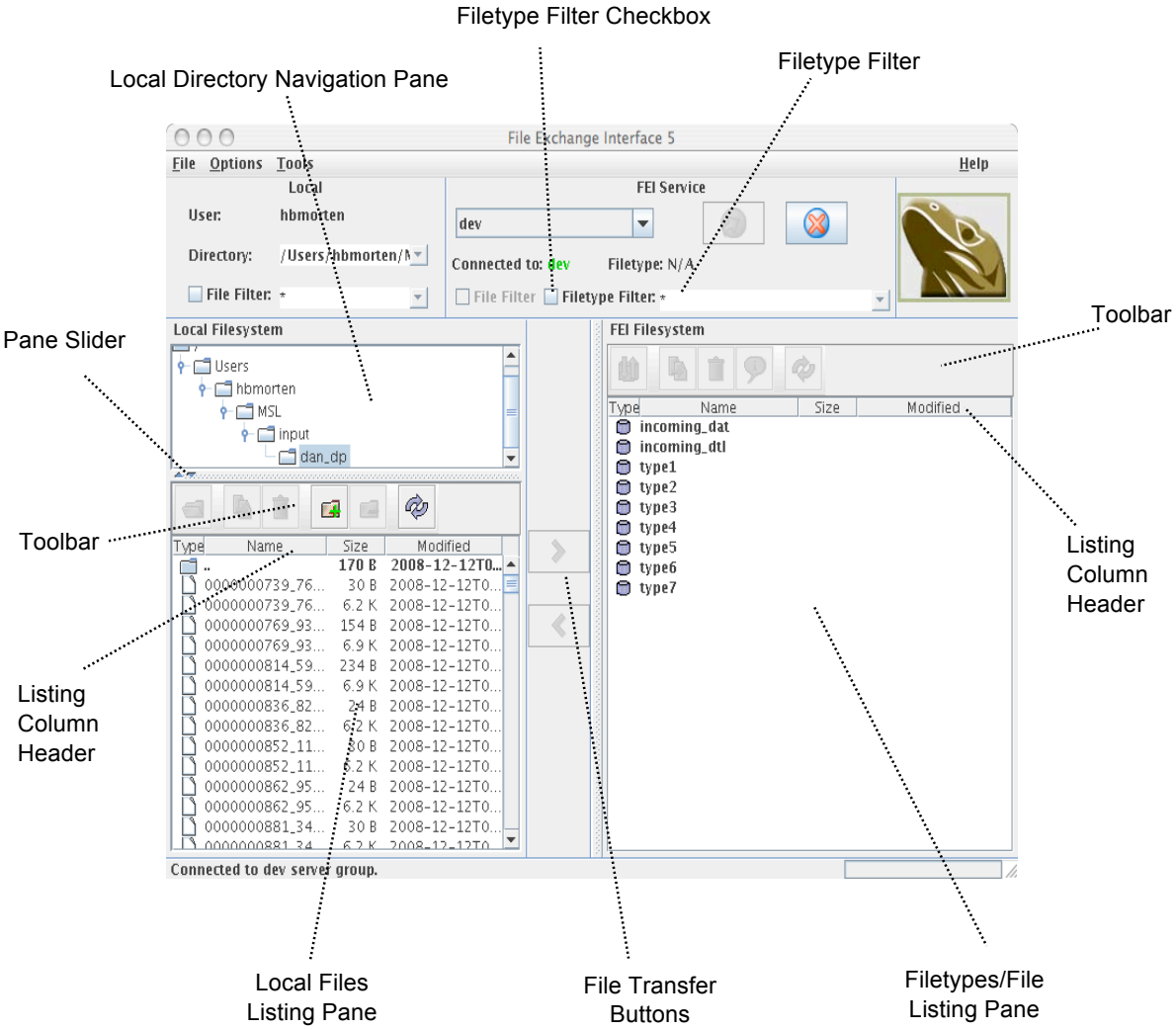
Login

The main screen is displayed upon startup of the FEI5 GUI. The screen is logically split into a left and a right half. The left side of the screen represents the user's local file system. The right half represents the FEI5 service and file system. The user can navigate up and down the file systems. The user first connects to the FEI5 server by selecting the server from the Server Menu and then selecting the Connect button. The Login Dialog Box requesting the Username and Password will be displayed. A successful connection will show the server name in the Connection Status. Command status is shown at the bottom of the GUI in the status bar area.



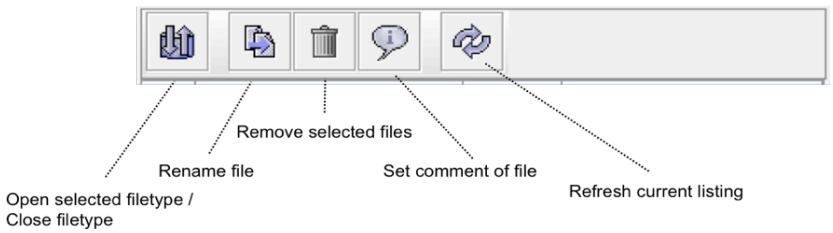
Navigating the Main Screen

After the user has successfully logged in, a list of the available filetypes is displayed in the FEI Filesystem area on the right half of the window in the Filetype Listing Pane. The user will navigate to a specific filetype and select the filetype of interest. When the list of filetypes is large, the user can scroll to the specific filetype or the user can apply a Filetype Filter to reduce the displayed list. The '*' is the wildcard character and can be placed anywhere in the filter expression. Multiple wildcards can be used. The user can activate the filter on and off by selecting or deselecting the Filetype Filter Check box to the left of the Filetype Filter. To select a filetype, double click on the filetype name in the Filetypes Listing Pane area or select the filetype name and then select the open selected filetype button from the FEI Filesystem toolbar. The list of files stored in the selected filetypes is displayed in the File Listing Pane area. The list of files can be sorted by simply selecting the Listing Column Header: name, size or modified. Double-clicking on a file name will display a dialog box containing file metadata.



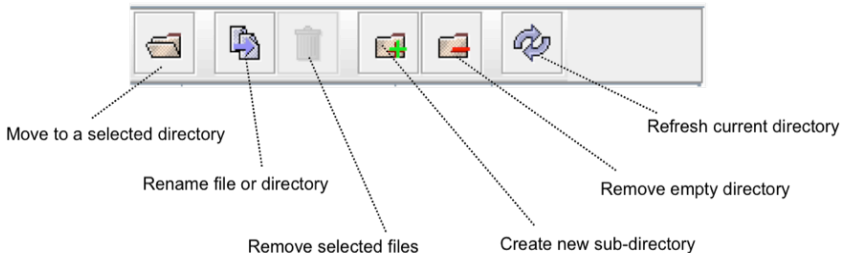
The FEI Filesystem toolbar has buttons for moving to a selected filetype, renaming a file (requires FEI rename capabilities), removing selected files (requires FEI delete capabilities), setting a comment for a file (requires FEI add capabilities), and refreshing the current listing. Select a file and then select a button. Place the cursor on top of a toolbar icon for a description.

FEI Filesystem Toolbar



The user can navigate their local file system by using the left side of the main screen. Double-click or enter the necessary information. The Local Filesystem Toolbar has buttons for moving to a select directory, renaming a file or directory, removing selected files to the trash can, creating new sub-directories or folders, removing empty directories and refreshing the current directory. Place the cursor on top of a toolbar icon for a description. The local file system also has a pane slider. You can use this to resize the navigation pane and the listing pane. Place the cursor over the pane slider until a slider arrow appears, then hold the mouse button down and drag the panes up or down to resize.

Local Filesystem Toolbar



- **Getting and Adding files via the main screen**

  To get files, select the files from the FEI Filesystem File listing pane and drag and drop the files onto the Local Filesystem listing pane. This can be done on individual files or a sequential set of selected files. To select a set of files, hold the mouse button down and drag the mouse across the set of files or select the first file and then hold the shift key down while selecting the last file of the list with the mouse button. The user can also select the files and then hit the file transfer button.

  Adding files, into the FEI server requires FEI add and/or replace privileges. To add files, select the files from the Local Filesystem File listing pane and drag and drop the files onto the FEI Filesystem listing pane. This can be done on individual files or a sequential set of selected files. To select a set of files, hold the mouse button down and drag the mouse across the set of files or select the first file and then hold the shift key down while selecting the last file of the list with the mouse button. The

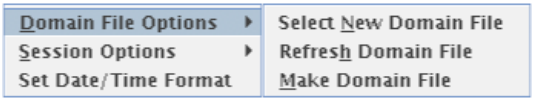
user can also select the files and then hit the file transfer button. Adding files requires access privileges. Contact the FEI5 System Administrator.

If a file already exists, a dialog will appear and asked if the user would like to replace the file.

- **The File Menu**

  The File menu allows the user to exit the FEI5 GUI application. The user also can exit the application with the key sequence of Alt-x or by the local system close window button. **Note**: When a user exits the GUI application, all settings are saved and restored upon the next login.

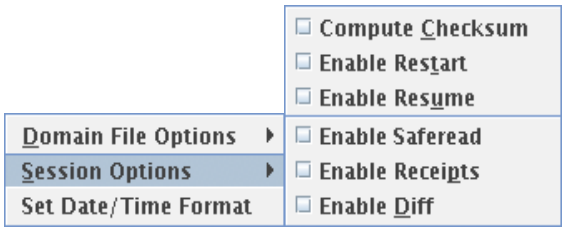
- **The Options Menu**



The Options menu allows the user to select, refresh or make a domain file, set session parameters or set the date/time format. The Domain File Options are not typically used. The Select New Domain File will present a open dialog box in which the user can locate a new domain file and ingest it. The domain file is an ascii XML file the client uses to determine what filetypes the user can see and where the servers reside.

The Refresh Domain File allows the user to restore the domain file with the one installed with the client. It will disconnect the user from the current server.

The Make Domain File allows the user to create a new domain file from the server configuration. This is useful when new filtypes are added. The user can create a new file and replace the current one located in the $FEI5 location.



The Session Options allows the user to set parameters for the client session. The Compute Checksum option turns on checksum computation. All gets and adds will ensure the checksums of files on both the local and server side match. All added files will have the server store the checksum. The checksum capability can also be controlled at the server level.

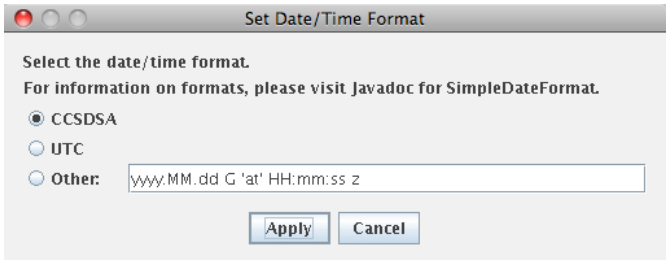
Enable Restart will track will track date of last download for given file type. When used with `computeChecksum` on, it allows transfer to resume.

Enable resume turns both checksum and restart on. This will allow a user to restart the acquisition of a file from the point where it left off.

Enable Saferead will transfer data into a hidden shadow directory on gets and then move it into the output directory only after the data is completely copied over. This ensures the data is not used while in transit.

Enable receipts allows the user to generate a receipt in the server for all adds and gets. The Compute Checksum feature is enabled with receipts. The receipt capability can also be controlled at the server level. This requires receipt capabilities.

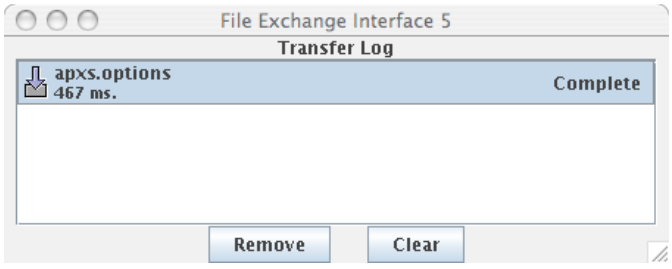
Enable Diff will allow an existing file to be replaced in the FEI5 server if the file to add differs from the file in the FEI5 server. On a get, Enable Diff will allow an existing file to be replaced in the user's local directory if the file to get from the server differs from the file on disk



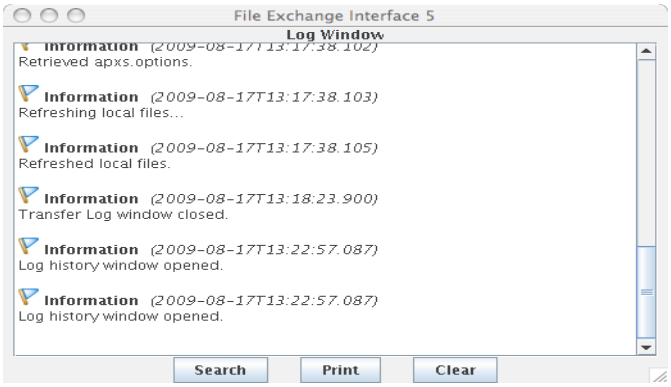
The Set Date/Time Format option will present a dialog box that allows the user to change the date/time format. The default is CCSDSA format.

- **The Tools Menu**

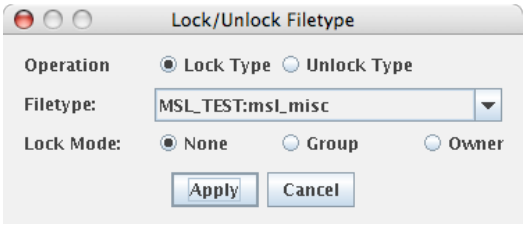
The Transfer History Tool will open a dialog box and show status on all adds and gets with the time it took to transfer the file. The user can removed or clear the contents of the dialog box.



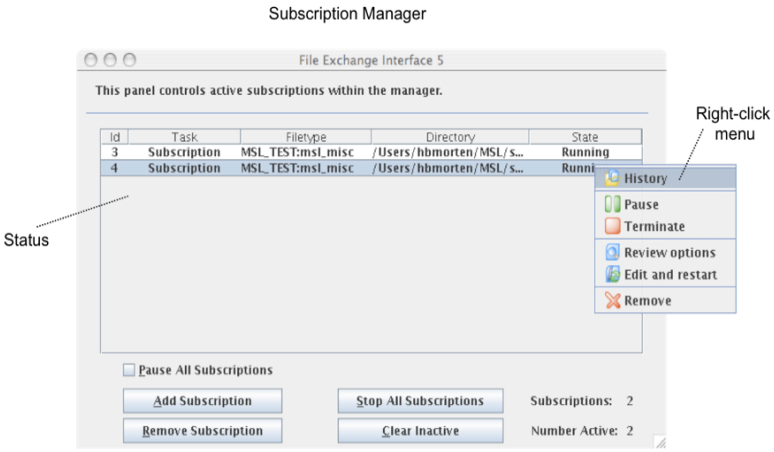
The Log Window Tool will open up a dialog box and display message about what the client is doing. The user can search, print or clear the contents of the dialog box.



The Change Password Tool will bring up a dialog box to change the user's password. If the server is configured for LDAP authentication, the user will need to follow the LDAP service instructions on how to change a password. The fei5kinit command should be performed after any password change if the batch commands are being used.



The Lock/Unlock Filetype Tool will allow a user to lock a filetype for read only. The lock/unlock function can be performed by a user with locktype capabilities.

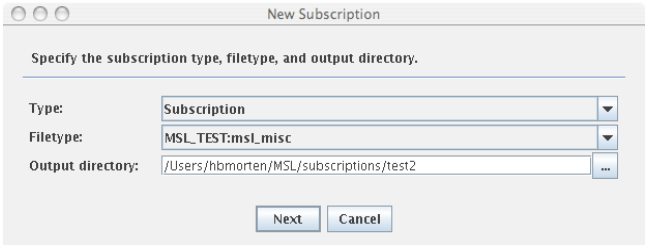


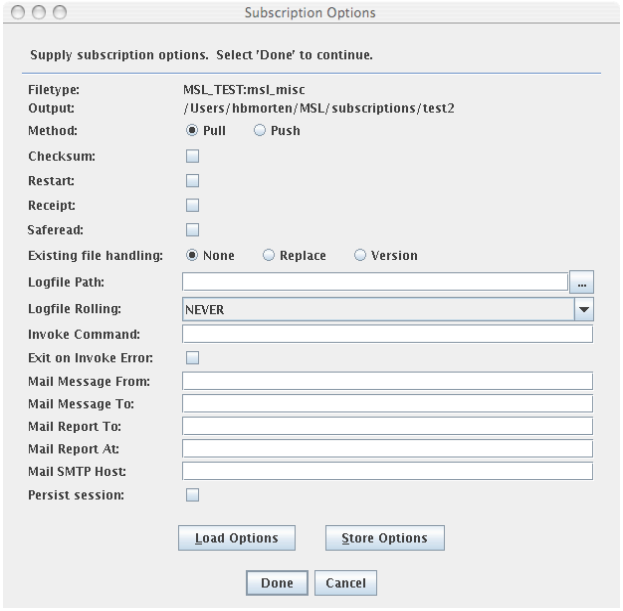
The Subscription Manager Tool will bring up a dialog that can be used for starting subscriptions from the GUI. The user must be aware that the subscriptions will shut down when the GUI is exited. **NOTE:** While subscriptions can be persisted in the GUI and restarted upon login, if the user

needs continuous subscriptions, the user should use the fei5subscribe or fei5notify batch commands.

Double clicking on a subscription from the Status area will bring up the Subscription History. Right-clicking on a subscription from the status area will bring up a subscription specific menu with the ability to show the subscription history, pause a subscription, terminate a subscription, view the subscription options, edit the subscription options and restart the subscription, and remove or stop the subscription. Editing a subscription will stop the subscription and restart. Make sure the restart option is set so the user can pick up files that may have been added during editing. Subscription options are described under the fei5subscribe command line utility in Section 4.

The Add Subscriptions button will guide the user in setting up a subscription via the New Subscription and Subscription Options dialog boxes. The Subscription Type can be subscribe or notify subscriptions and either pull or push subscriptions. Refer to Section 4 fei5subscibe and fei5notify command line descriptions for more information. The subscribe subscription copies the files over, the notify subscription just reports a file has been received in the log window or in reports, but combined with the invoke command can be used to send email notification or perform other functions. Subscriptions can be persisted or saved via the Subscription Options dialog so the user can close the GUI application and have the subscription restarted upon the next FEI5 GUI login.





4. Command Line Utilities

The FEI5 client command line utilities are a set of batch commands for interacting with FEI5 file types. These flexible commands are designed to offer a quick and simple interface for the users to interact with their file types. They can be used for developing FEI5-enabled system applications or scripts and also can be employed for integrating FEI5 into other services (e.g. web service).

Getting help

To get help on each command, simply type the command followed by the keyword **help**.

```
% fei5list help
Copyright 2002-2010 Jet Propulsion Laboratory, Caltech, NASA
FEI5 release 2.2.4, April 2010
Komodo API release 3.0.1, December 2009


Usage:  fei5list [<server group>:]<file type> ['<file name expression>']
        {[before|after <date-time>] | [between <date-time1> and <date-time2>]}
        [format '<date format>'] {[long | verylong]}
        [query <queryfile>] [help]
```

Exit values

Most of the command line utilities are batch commands. They can be embedded into systematic scripts for system integration development and data processing automation. Each batch command is expected to return 0 (zero) as the success status and a positive value to indicate the number of failed transactions.

```
% fei5list MDMS_DEV:image1
bigfile.tst

% echo $status (csh)

0

% fei5list MDMS_DEV:bad_file_type
```

FEI5_ERROR::Unable to find file type "MDMS_DEV:bad_file_type" in Domain

```
% echo $? (bash)
1
```

File name expressions

A file name expression is a simple yet powerful method for users to selectively filter files names using the wild-card symbol asterisk (`*`). For example, to filter all earth jpeg images, one might use the file name expression `earth*.jpg`. Both the FEI5 client and server are equipped to handle file name expressions. For the command-line client utilities, users should be aware that all file name expressions, by default, are being evaluated by the client's local shell or terminal before passing the evaluated list to the target application program. The file name expression `earth*.jpg` will return a list of local file names that begin with the string '`earth`' and end with the extension '`.jpg`'.

Command-line utilities such as **fei5add** and **fei5replace** are designed to handle a list of file names from the command-line. That is, to add all earth jpeg images to FEI5 server, the user can simply issue **fei5add MDMS_DEV:image1 earth*.jpg**. After evaluated by the client's local shell or terminal, the program is really being invoked as **fei5add MDMS_DEV:image1 earth1.jpg earth2.jpg earth3.jpg...** etc.

Command-line utilities such as **fei5list** and **fei5get** are for the user to retrieve files from the FEI5 server. A file name expression can be used by these commands, but the expression should be evaluated on the FEI5 server, not on the client's local file system. To prevent file name expressions from being evaluated locally, the file name expression must be passed as single or double-quoted strings (double-quoted only for Windows). For example, to display names of all earth jpeg files registered under the file type `MDMS_DEV:image1`, the user can simply issue **fei5list MDMS_DEV:image1 "earth*.jpg"**.

Unix users should be aware that the Unix shells handle single and double quotes differently. Double-quote (" ") enables evaluation. That is, variables (all strings beginning with the character `$`) referenced within the double-quoted string will be first evaluated by the Unix shell before passing to the targeted application program. For example, to list all 'jpg' files on the user's home directory, the user might issue **ls "$HOME/*jpg"**. The environment variable `$HOME` will be replaced with the user's local session value before passing to the application program `ls`. Single-quote (' ') disables evaluation. If we replace the double-quotes with single-quotes, the '`ls`' statement will return an error that said `ls: $HOME: No such file or directory.`

Date formats

FEI5, by default, accepts date input in the CCSDS [*http://www.ccsds.org*] ASCII ("CCSDSA") calendar segmented time code format:

YYYY-MM-DDThh:mm:ss.SSS[1]

DATE PART	DESCRIPTION
`yyyy`	Year in four-character subfield with values 0001-9999 (two-character year input is not supported)
`MM`	Month in two-character subfield with values 01-12
`dd`	Day of month in two-character subfield with values 01-28, -29, -30, or -31
`'T'`	Calendar-Time separator
`hh`	Hour in two-character subfield with values 00-23

[1] The support for fraction of a second is specific to the file registry implementation. In the case of using DBMS as the file registry, this field is DBMS-specific and its precision may vary.

DATE PART	DESCRIPTION
mm	Minute in two-character subfield with values 00-59
ss	Second in two-character subfield with values 00-59
SSS	Decimal fraction of a second in one to three-character subfield where each d has a value of 0-9

Note: The hyphen '-', colon ':' letter 'T' and period '.' are used as specified subfield separators. All subfields must include leading zeros if the input date does not use all available characters. Named format date inputs can be specified without a time. The truncated parts are automatically padded with zeros. Which implies midnight for that time. That is, a date value of **2003-09-14** gets translated into **2003-09-14T00:00:00.000**.

As of release 2.0, the format of the date may be controlled to include other formats such as UTC and any java date formats that conform to the Java **SimpleDateFormat** API (http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html). The Java date pattern formats cannot be truncated and thus the entire format must be specified. The Java format is case sensitive.

```
% fei5list image1 "file*" long
    1. 2004-03-16T14:52:32.793,          4310, file5
    2. 2004-03-16T14:52:33.586,          4310, file6
    3. 2004-03-16T14:52:54.020,          4310, file1
    4. 2004-03-16T14:53:04.423,          4310, file0

% fei5list image1 "file*" long before 2004-03-16T14:53

    1. 2004-03-16T14:52:32.793,          4310, file5
    2. 2004-03-16T14:52:33.586,          4310, file6
    3. 2004-03-16T14:52:54.020,          4310, file1

% fei5list image1 "file*" long format "utc"

    1. 2004-075T14:52:32.793,          4310, file5
    2. 2004-075T14:52:33.586,          4310, file6
    3. 2004-075T14:52:54.020,          4310, file1
    4. 2004-075T14:53:04.423,          4310, file0

% fei5list image1 "file*" long format "utc" before 2004-075T14:52:54.020

    1. 2004-075T14:52:32.793,          4310, file5
    2. 2004-075T14:52:33.586,          4310, file6
    3. 2004-075T14:52:54.020,          4310, file1

% fei5list image1 "file*" long format "EEE, MMM d, ''yy"

    1. Tues, Mar 15, '04,          4310, file5
    2. Tues, Mar 16, '04,          4310, file6
    3. Tues, Mar 16, '04,          4310, file1
    4. Tues, Mar 16, '04,          4310, file0
```

```
% fei5list image1 "file*" long format "EEE, MMM d, ''yy" after "Tues, Mar 16, '04"
    1. Tues, Mar 16, '04,          4310, file6
    2. Tues, Mar 16, '04,          4310, file1
    3. Tues, Mar 16, '04,          4310, file0
```

Time queries may either be inclusive or exclusive. Commands that use the before or between parameter provide results in which the date/times queried are included. Commands that use the after parameter provides results in which the date/time is excluded.

As shown in the examples above, the most common date formats, CCSDSA and UTC, can be referenced by name instead of a format string ('ccsdsa' and 'utc', respectively). This capability was specifically included to assist users who may often use those formats and wish to avoid specifying to a well-formed format. The ability to declare dates with truncated times (i.e. "2006-205" instead of "2006-205T00:00:00.000" exists for the ccsdsa and utc named formats. The set of available named formats might increase depending on user feedback.

Note: The time reported by FEI5 might be different than the time reported by the unix command "date". The operating system of the machine might use several files that set the time zone for the client host. Java might use a file that is different than from what the unix date command uses to determine time zones. Have your SA set all files to the same time zone if you wish to have the FEI5 date and time be the same as the Unix "date" command.

Capabilities

A capability specifies the action(s) the user can perform on a given file type and the files. Capability values are set on a per-user basis by the FEI5 server administrator. A user can view their capabilities in the interactive 'fei5' application by typing the **`showCapabilities`** command. Some command line command parameters can only be used if the user has the correct capabilities. See section 4 for more information.

Command Timeout

All commands will timeout and produce an error if the command fails to communicate with the server after 5 minutes. The subscription commands (subscribe, notify) will timeout after 5 minutes if it cannot communicate with the server and restart itself.

Note: The time-out value can be overridden using the Java property *komodo.client.timeout*. For example, to set the timeout to ten minutes (60000 milliseconds), you would add the following property to your Java command-line invocation: *-Dkomodo.client.timeout=60000*.

Working with file types

FEI5 organizes files by base file types or simply file types, which is a virtual interface to the file system. Users do not have to know the physical location of the file types and the files they manage, because of the FEI5 file type abstraction. FEI5's data store keeps metadata about each file within a file type.

The user can see the list of file types that are registered within the domain file

```
% fei5filetypes
Domain file: /usr/local/fei5/config/domain.fei
Server Group: MDMS_DEV
   image1
   image2
   image3
   image4
```

In this example, there are four file types. These file types come from the domain file. If a new file type is added, the FEI5 administrator will notify the user and the user will use the **makedomainfile** command from the **fei5** interactive tool to create a new domain.fei file and replace the domain.fei file in the $FEI5/ directory. The server group identifies the server the file types are associated with. File type names are unique to a server group. If the user maintains more than one server group in their domain.fei file, they may need to specify the server group along with the file type name in commands.

- **Insuring file integrity**

  FEI5 uses file checksums to ensure a file's integrity when transmitted from the server to a client. FEI5 uses SHA1. Checksums are not enabled by default in FEI5 due to the overhead required to compute them. If enabled, a checksum is computed for a file as it is being transmitted or received, depending on the operation. For 'get' operations, the receiver (client) also calculates the checksum as the data is received. The sender (server) checksum is transmitted to the receiver after the file has been fully transmitted. The sender and receiver checksums are compared. If the file was altered during transmission, the checksum values will not match and the file will be rejected. By default, only the file size is checked. An automatic retry feature is included in the client software as of release 1.8.0. The commands fei5get and fei5subscribe will attempt up to 3 times to transfer the file successfully; otherwise an error will be reported.

  FEI5 file types can be configured by the FEI5 administrator to automatically compute checksums on all added files. Users can control file checksums with the 'crc' parameter in the fei5add/replace, the fei5accept for add/get, the fei5get and the fei5subscribe commands. See the specific commands for more information.

- **Resume transfer**

  FEI5 has the capability to allow clients to resume the acquisition of a file in the event that communication was severed before file transfer completes. Using the ***crc*** parameter in the **fei5get** and **fei5subscribe** commands to enables resume transfer. Resume transfer requires file checksum data to detect file corruption. The resume transfer capability benefits users who must transfer large files and don't have time to retransfer the entire file if communication is severed.

  EXAMPLE User logs on to FEI5 and connects to image file type.

```
% fei5get MDMS_DEV:image1 largefile.jpg crc
File resume transfer enabled

*** communication is severed before transfer completes ***
```

  User reissues the **fei5get** command with the crc option.

```
% fei5get MDMS_DEV:image1 largefile.jpg crc
File resume transfer enabled
  1. Got "largefile.jpg".  CRC:a0d271f0248d488c4362578
%
```

  If the file had been corrupted during the first transfer, you would get the following after the corruption was detected during the 2nd call.

```
% fei5get MDMS_DEV:image1 largefile.jpg crc
File resume transfer enabled
Verification error (CRC) occurred for "largefile.jpg" on attempt 1.  Trying again…
  1. Got "largefile.jpg".  CRC:a0d271f0248d488c4362578
%
```

▸ User login utilities

These commands are for user login initialization. They are intentionally named similarly to the legacy Kerberos ticket management utilities to simplify migration of existing processes, but they are not associated with Kerberos in any way. The utilities are for login cache file management only.

- **fei5kinit**

DESCRIPTION

This utility is used to create a user login cache file that will be accessed by other general FEI5 batch utilities. The file will persist until a fei5kdestroy command is issued. The user login is not authenticated after the fei5kinit command is executed. The validation of the user login is not done until the user issues any FEI5 batch command.

NOTE: **fei5kinit** will need to be executed after any password change.

SYNOPSIS

fei5kinit **[<user name>] [<server group>] [*help*]**

SINCE

Release 1.4.0.

KEYWORDS

- ***user name:*** provided by the FEI5 Administrator
- ***server group:*** server group login credentials associated with (default is '*' which means for all server groups)
- ***help***: display the command line help message.

EXAMPLE To create and validate a login cache file for the user name `ops`.

```
% fei5kinit
User name>> ops
Password>> *****
Server group [ENTER for default]>> MDMS_DEV

% fei5list MDMS_DEV:image1
test1.img
test2.img
```

EXAMPLE Creation of a bad login cache file.

```
% fei5kinit ops
Password>> *****
% fei5list MDMS_DEV:image1
FEI_ERROR::Invalid login
```

- **fei5klist**

DESCRIPTION

This utility allows the user to view information on their login cache file. The login cache is located in the user's home directory by default in the .komodo directory. To create the login location in another directory, set the FEI5CDDIR environment variable. This is useful if a user is running 2 different versions of the FEI5 client for different projects and they use different authentication protocols. FEI5 V224 and up will handle multiple servers in a single login cache.

SYNOPSIS

fei5klist **[*help*]**

SINCE

Release 1.4.0.

KEYWORDS

- ***help***: display the command line help message.

EXAMPLE

```
% fei5klist
Copyright 2002-2008 Jet Propulsion Laboratory, Caltech, NASA
FEI5 release 2.1.0, April 2008
Komodo API release 2.9.6, April 2008

Credential cache file: /home/ops/.komodo/.komodologin

File modified on: Wed Feb 09 14:01:33 PST 2005
Default principal: ops
```

- **fei5kdestroy**

DESCRIPTION

This utility allows the user to destroy their login cache file.

SYNOPSIS

fei5kdestroy **[*help*]**

SINCE

Release 1.4.0.

KEYWORDS

- ***help***: display the command line help message.

EXAMPLE

```
% fei5kdestroy
% fei5klist
Login Error! Please acquire credentials with login utility.
```

General utilities

- **fei5accept**

DESCRIPTION

This utility accepts a list of file names from standard-in. The **for** operation will be performed for each file name. **fei5accept** is useful in an automated data processing environment. A data processor can keep track of the list of files it creates by logging the file names to a text file during each run. At the end of each processing cycle, the list of file names can be piped to **fei5accept** for delivery to the FEI5 server.

SYNOPSIS

```
fei5accept [<server group>:]<file type> for <add|replace|get|delete>
           [output <path>] [crc] [saferead] [autodelete]
           {[replace|version]} [diff] [help]
```

SINCE

Release 1.7.0.

KEYWORDS

- **autodelete**: this option works with the `add` and `replace` operations to delete the local copy of the file after the successful completion of add/replace operation. If specified with get or delete, it will be ignored. NOTE: For add, if the file already exists in the server, the local file will not be deleted. Default: off.
- **crc**: calculate the CRC value (currently using SHA1) for each file and save the value in the database for add or replace. With the *for get* option, the file will be transferred and the CRC value will be compared to the value returned by the server. If the two CRC values do not match, another attempt to copy the file will be made. The command will attempt a retry up to 3 times. Default: off.
- **diff**: used in conjunction with the **replace or version** option. Enables a get on replace of a file only if the file is different from an existing file in the output directory otherwise the file is skipped. Default: off.
- **for**: operation type **add**, **replace**, **get**, or **delete**. Default: none.
- **help**: display the command line help message.
- **output**: output path for the **get** operation. Default: current working directory.
- **replace**: this option works with the **get** operation to replace existing files within the output directory. Default: off.
- **saferead**: this option works with the **get** operation to only make the file visible to the output directory when the file has been fully received. Default: off.
- **version**: this option works with the **get** operation to create versioning of existing files within the output directory. If a newly retrieved file already exist within user's output directory, the current copy will be renamed by attaching a time value to the end of the file before writing the newly retrieved file to the output directory. Default: off.

EXAMPLE Getting a list of files by piping a list of file names to `fei5accept` and apply get on each.

UNIX:

```
% cat list.txt
file1.jpg
file2.jpg
file3.jpg
file4.jpg
file5.jpg

% cat list.txt | fei5accept MDMS_DEV:image1 for get
```

Windows:

```
% type list.txt
file1.jpg
file2.jpg
file3.jpg
file4.jpg
file5.jpg

% type list.txt | fei5accept MDMS_DEV:image1 for get
```

EXAMPLE Getting files by redirecting a list of file names to `fei5accept` and apply get on each.

```
% fei5accept MDMS_DEV:image1 for get < list.txt
     1. Got "file1.jpg".
     2. Got "file2.jpg".
     3. Got "file3.jpg".
     4. Got "file4.jpg".
     5. Got "file5.jpg".
```

EXAMPLE Get a list of files from fei5list and pipe the list to `fei5accept` and apply get on each.

```
% fei5list MDMS_DEV:image1 "file*" | fei5accept MDMS_DEV:image1 for get
     1. Got "file1.jpg".
     2. Got "file2.jpg".
     3. Got "file3.jpg".
     4. Got "file4.jpg".
     5. Got "file5.jpg".
```

EXAMPLE Replacing a list of files (use DIR /b for windows instead of ls)

UNIX:

```
%  ls *.jpg | fei5accept MDMS_DEV:image1 for replace

     1. Replaced "file1.jpg".
     2. Replaced "file2.jpg".
     3. Replaced "file3.jpg".
     4. Replaced "file4.jpg".
     5. Replaced "file5.jpg".
```

Windows:

```
%  dir /b *.jpg | fei5accept MDMS_DEV:image1 for replace

     1. Replaced "file1.jpg".
     2. Replaced "file2.jpg".
     3. Replaced "file3.jpg".
     4. Replaced "file4.jpg".
     5. Replaced "file5.jpg".
```

- **fei5add**

DESCRIPTION

This utility adds one or more files to the FEI5 server. The user can use a list of files or a file name expression in place of the file name. Also, the user can include an FEI5 date-time specification following one of the time keywords **before**, **after**, **between**... **and**... along with a file name expres-

sion to limit the list to files modified within a specified time range. The same rules apply to the options file. A file is not added if it already exists in the FEI5 server. Use **fei5replace** to replace files that exist in the FEI5 server.

SYNOPSIS

```
fei5add [<server group>:]<file type> <file name expression>
        {[before|after <date-time>] |
          [between <date-time1> and <date-time2>]}
        [format '<date format>']
        [comment '<comment text>'] [crc] [receipt] [autodelete] [help]

fei5add using <option file>
Option File Format (per line):
        [<server group>:]<file type> <file name>...
        {[before|after <date-time>] |
          [between <date-time1> and <date-time2>]}
        [format '<date format>']
        [comment '<comment text>'] [autodelete] [crc] [receipt]
```

SINCE

Release 1.4.0. (**format** option introduced in 2.0.0)

KEYWORDS

- ***after***: add files with creation times after the specified time value. Default: none.
- ***autodelete***: automatically delete the local file after it is successfully added to the server. If unsuccessful in adding the file, the local file will not be deleted Default: off. **Since: Release 1.7.0.**
- ***before***: add files with creation times before the specified time value. Default: none.
- ***between*...** ***and*...**: add files with creation times between the specified time range values. Default: none.
- ***format:*** overrides default date-time format with the specified pattern or name. Default: ccsdsa.
- ***comment***: add a description to the file. A comment longer than 255 characters will be truncated. Default: none.
- ***crc***: calculate the CRC (SHA1) value for each file and save in the database. Checksum validation is enabled. Default: off.
- ***help***: display the command line help message.
- ***receipt***: file delivery status is logged in FEI5 server and a tracking receipt ID is sent to the user. Checksum validation is enabled. Requires 'receipt' capability. Contact the FEI5 administrator. Default: off. **Since: Release 1.7.1.**
- ***using***: keyword to supply a list of commands in an options file. Default: none.

EXAMPLE Simple add operations

```
% fei5add image1 file1
% fei5add image1 file1 file2 file3
% fei5add image1 file*
% fei5add image1 file* after 2003-10-24
% fei5add image1 file* after 2003-10-24T14:21
% fei5add image1 file* between 2003-10-24T14:20 and 2003-10-24T14:30
```

EXAMPLE Add files using options file

```
% cat addlist.txt
MDMS_DEV:image1 /home/ops/data/file0
MDMS_DEV:image1 file1 file2  # list of file names
MDMS_DEV:image1 file* between 2003-255 and 2003-256 format 'utc'

% fei5add using addlist.txt
```

- **fei5changepassword**

DESCRIPTION

This utility allows the user to change their non-LDAP password. Servers configured with LDAP need to follow their LDAP service provided instructions for changing passwords. The user should follow any changed password with the **fei5kinit** command. **Note**: Do not execute the **fei5kinit** prior to changing the password because the **fei5changepassword** command needs to login into the server first before it can change the password and needs a valid credentials cache file. The user will get an Invalid login if the credential cache file contains the incorrect information.

SYNOPSIS

***fei5changepassword* <server group>**

SINCE

Release 2.1.4

KEYWORDS

- ***help***: display the command line help message.
- ***servergroup***: Name of FEI5 server group.

EXAMPLE Change password

```
% fei5changepassword
Changing password for user 'username'
Enter new password (or type "abort" to quit)>>
Re-enter new password>>
% fei5kinit
User name>> ops
Password>> ****
% fei5klist MDMS_DEV:image1
test1.img
```

- **fei5check**

DESCRIPTION

This utility is used to check the correctness and completeness of an FEI5 client installation. See the 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 user login cache exists, it will be used for login information; otherwise a '**fei5kinit'** command may need to be issued first.

SYNOPSIS

***fei5check* [*help*]**

SINCE

Release 1.7.0

KEYWORDS

- ***help***: display the command line help message.

EXAMPLE A Successful Check

```
% fei5check
Trying connection to file type "MDMS_DEV:image1"
OK
```

```
Trying connection to file type "MDMS_DEV:image2"
OK
Trying connection to file type "MDMS_DEV:image3"
OK
Trying connection to file type "MDMS_DEV:image4"
OK
Trying connection to file type "MDMS_DEV:image5"
OK
Trying connection to file type "MDMS_DEV:process_init"
OK
Trying connection to file type "MDMS_DEV:prod_available"
OK
Trying connection to file type "MDMS_DEV:sigevent"
OK
```

EXAMPLE An Unsuccessful Check

```
% fei5check
Trying connection to file type "MDMS_DEV:image1"
OK
Trying connection to file type "MDMS_DEV:image2"
OK
Trying connection to file type "MDMS_DEV:image3"
FEI_ERROR::Unable to connect to filetype "MDMS_DEV:image3" Server(s)
```

- **fei5checkfiles**

DESCRIPTION

This utility queries a list of files from FEI5 based on the input parameters supplied, and then, for each file name returned, **fei5checkfiles** tries to open the file in the current working directory. It prints a message based on the outcome of that attempt. (**fei5checkfiles** opens and closes files but does nothing to the content of a file.)

The user should use this utility to determine if they have any missing copies of files found in the FEI5 server. The user should use **fei5get** to retrieve any missing files.

SYNOPSIS

```
fei5checkfiles [<server group>:]<file type> ['<file name expression>']
               {[before|after <date-time>] |
                 [between <date-time1> and <date-time2>]}
               [format '<date format>'
               {[long | verylong]} [help]

fei5checkfiles using <option file>
Option File Format (per line):
        [<server group>:]<file type> '<file name expression>'
        {[before|after <date-time>] |
         [between <date-time1> and <date-time2>]}
        [format '<date format>'
        {[long | verylong]}
```

SINCE

Release 1.7.0.

KEYWORDS

- ***after***: compare files with FEI5 modification dates after the specified time value. Default: none.
- ***before***: compare files with FEI5 modification dates before the specified time value. Default: none.
- ***between****... and...*: compare files with FEI5 modification dates between the specified time range value. Default: none.
- ***format:*** overrides default date-time format with the specified pattern or name. Default: ccsdsa.
- ***help***: display the command line help message.
- ***long***: output with file modification date and file size in bytes. Default: off.
- ***using***: keyword to supply a list of commands in an options file. Default: none.
- ***verylong***: lists output with file modification date, file size in bytes, contributor, creation date, and any associated comments and archive notes. Default: off.

EXAMPLE

```
% ls
file1.jpg       file2.jpg       file3.jpg       file4.jpg       file5.jpg

% fei5list image1

file6.jpg
file7.jpg
file1.jpg
file2.jpg
file3.jpg
file4.jpg
file5.jpg

% fei5checkfiles image1

    1. file6.jpg
       - No such file.
    2. file7.jpg
       - No such file.
    3. file1.jpg
    4. file2.jpg
    5. file3.jpg
    6. file4.jpg
    7. file5.jpg
```

- **fei5comment**

DESCRIPTION

This utility adds a comment to the metadata of an existing file on the FEI5 server.

SYNOPSIS

```
fei5comment [<server group>:]<file type> <file name>
            comment "<comment text>" [help]
```

SINCE

Release 1.4.0.

KEYWORDS

- ***comment***: add description to the file. A comment longer than 255 characters will be truncated to only 255 characters.
- ***help***: display the command line help message.

EXAMPLE

```
% fei5comment MDMS_DEV:image1 file1.jpg comment 'latest earth image'
     1. Commented "file1.jpg".

% fei5list MDMS_DEV:image1 file1.jpg verylong

    1. 2005-02-24T17:41:00.346,         2913, file1.jpg
       [Contributor] ops
       [Created] 2005-02-24T17:05:14.030
       [Remote location] /data/MDMS_DEV/image1
       [Comment] "latest earth image1
```

- **fei5crc**

DESCRIPTION

This utility computes the SHA1 CRC value for a local file.

SYNOPSIS

```
fei5crc <file name expression> [help]
```

SINCE

Release 1.7.0.

KEYWORDS

- ***help***: display the command line help message.

EXAMPLE

```
% fei5crc file1.jpg
File:"file1.jpg"  Checksum:"93070c6bcd3fa094623151ceb00f39cc15c9f996"
```

- **fei5delete**

DESCRIPTION

This utility takes the file type and file name expression and deletes all files registered to the file type that matches the specified file name or expression.

Note: The FEI5 administrator can add a logDeleteRecord option to a filetype. If this is done, the server will mark the database record as a deleted file and delete the file from disk. If this file is ever added again, the database record marked as deleted will be used and updated for the newly added file and the original creation time from the first add will be retained. Contact the FEI5 administrator for more information.

SYNOPSIS

```
fei5delete [<server group>:]<file type> '<file name expression>' [help]

fei5delete using <option file>
Option File Format (per line):
        [<server group>:]<file type> '<file name expression>'
```

SINCE

Release 1.4.0.

KEYWORDS

- ***help***: display the command line help message.
- ***using***: keyword to supply a list of commands in an options file. Default: none.

EXAMPLE Delete using an options file

```
% cat ilena.txt
MDMS_DEV:image1 file2
MDMS_DEV:image1 file3
MDMS_DEV:image1 file4
MDMS_DEV:image1 file5
MDMS_DEV:image1 file6

% fei5delete using ilena.txt
```

EXAMPLE Delete using server-side file name expression

```
% fei5delete MDMS_DEV:image1 'file*'
     1. Deleted "file2".
     2. Deleted "file3".
     3. Deleted "file4".
     4. Deleted "file5".
     5. Deleted "file6".
```

- **fei5display**

DESCRIPTION

This utility retrieves a file from the FEI5 server and writes its contents to standard-out. The file name is also written, but it's written to standard-error. The user can use this utility to display the contents of a file or pipe the contents to another program. Warning: Running this command with a binary file is not recommended as it may corrupt your standard output terminal session.

SYNOPSIS

```
fei5display [<server group>:]<file type> <file name> [help]
```

SINCE

Release 1.7.0.

KEYWORDS

- ***help***: display the command line help message.

EXAMPLE

```
% fei5list image1 filelist.txt
filelist.txt
```

```
% fei5display image1 filelist.txt
file1.jpg
file2.jpg
file3.jpg
file4.jpg
file5.jpg
```

EXAMPLE Process chaining

```
% fei5list image1
file1.jpg
file2.jpg
file3.jpg
file4.jpg
file5.jpg
filelist.txt


% fei5display image1 filelist.txt | fei5accept image1 for get
     1. Got "file1.jpg".
     2. Got "file2.jpg".
     3. Got "file3.jpg".
     4. Got "file4.jpg".
     5. Got "file5.jpg".
```

- **fei5filetypes**

DESCRIPTION

This utility lists the files types in the user's local FEI5 domain file. As of version 2.0.1, the domain file defines a default server group. If only one server group is defined, it will be the default. If more than one server group is defined in the domain file, the output will indicate which one is the default.

SYNOPSIS

```
fei5filetypes {'[<server group>:][<file type expression>]'|[srvgroups]}
              [classic] [help]
```

SINCE

Release 1.7.5

KEYWORDS

- ***classic***: output the file types list in classic format. Default: off
- ***help***: display the command line help message.
- ***srvgroups***: output information on each server group listed in the domain file. Do not provide a <server group>: name when using this parameter. If more than one server group is defined in the domain file, the default server group will be identified in the output.

EXAMPLE Querying using file type name expression.

```
% fei5filetypes "MDMS_DEV:image*"
Domain file: /home/ops/fei5/config/domain.fei
Server Group: MDMS_DEV
   image1
   image2
   image3
   image4
   image5
```

EXAMPLE Querying using file type name expression and output in classic format.

```
% fei5filetypes "MDMS_DEV:image*" classic
Domain file: /home/ops/fei5/config/domain.fei
image1                image2
image3                image4
image5
```

EXAMPLE List server group names

```
% fei5filetypes srvgroups

Server groups in "domain.fei"


 MDMS_DEV
```

- **fei5get**

DESCRIPTION

This utility retrieves a file or set of files. With a file name expression the user can also query by file modification time for all files that satisfies the file name expression that were modified within the specified date-time. By default this utility will not replace a file that already exists in the user's local directory unless the user supplies the keyword **replace** or **version**. **Replace** replaces the current local file. **Version** renames the current copy of the file by attaching a timestamp to the end before writing the new file. A new added feature as of release 2.0 is if the client is unable to reach the server due to network issues, after a default of five minutes, the client will stop with an error.

SYNOPSIS

```
fei5get [<server group>:]<file type> ['<file name expression>']
        [output <path>] {[before|after <datetime>] |
          [between <datetime1> and <datetime2>]}
        [format '<date format>'] [crc] [saferead] [receipt]
        {[replace|version]} [diff] [query <queryfile>] [replicate]
        [replicateroot <rootdir>] [filehandler] [help]

fei5get using <option file>
Option File Format (per line):
        [<server group>:]<file type> ['<file name expression>']
        [output <path>] {[before|after <date-time>] |
          [between <date-time1> and <date-time2>]}
        [format '<date format>'] [crc] [saferead]
        [receipt] {[replace|version] [diff] [replicate]
        [replicateroot <rootdir>] [filehandler]
```

SINCE

Release 1.4.0

KEYWORDS

- ***after***: get files with FEI5 modification dates after the specified time value. Default: none.
- ***before***: get files with FEI5 modification dates before the specified time value. Default: none.
- ***between****... **and**...*: get files with FEI5 modification dates between the specified time range. Default: none.
- ***crc***: calculate the CRC value using SHA1 for each file. This automatically enables resume file transfer. See section 4 – 'Resume Transfer' section for more details. The file will be transferred and the CRC value will be compared to the value returned by the server. If the two CRC values do not match, another attempt to copy the file will be made. The command will attempt a retry up to 3 times. Default: off.

- ***diff***: used in conjunction with the **replace or version** option. Enables a get on replace of a file only if the file is different from an existing file in the output directory otherwise the file is skipped. Default: off.
- ***filehandler:*** Enables file management when using replication where older files will be deleted when a specified threshold is reached. Requires a configuration file. Default: off.
- ***format:*** overrides default date-time format with the specified pattern or name. Default: ccsdsa.
- ***help***: display the command line help message.
- ***output***: output path. Default: current working directory.
- ***query***: name of a queryfile for queries based on data meta-data. Requires the server to be configured to a meta-data database. By default, this is not available. Contact the FEI5 administrator to find out if your server is configured for this capability.
- ***receipt***: file delivery status is logged in FEI5 server and a tracking receipt ID is sent to the user. Checksum validation is enabled. Requires 'receipt' capability. Contact the FEI5 administrator. Default: off. **Since: Release 1.7.1.**
- ***replace***: replace existing files within the output directory. Default: off.
- ***replicate***: Enables the client to place the copied file into the same location as the server location. This may be required in instances where other software is dependent on the same directory structure as the server. This requires that the user have privileges on the local host that allow the user to create directories and files. Contact your project representative for more information.
- ***replicateroot***: Allows a root prefix to the server location. Useful in testing.
- ***saferead***: files are to only appear in the output directory when it has been fully received. Default: off. **Since: Release 1.7.0.**
- ***version***: create versioning of existing files within the output directory. If a newly retrieved file already exists within the user's output directory, the current copy will be renamed by attaching a time value to the end of the file before writing the newly retrieved file to the output directory. Default: off.
- ***using***: keyword to supply a list of commands in an options file. Default: none.

EXAMPLE Simple get operations

```
% fei5get image1 file1
% fei5get image1 file1 replace
% fei5get image1 'file*'
% fei5get image1 '*' after 2003-10-24
% fei5get image1 'file*' version after 2003-298T14:20 format 'utc'
% fei5get image1 'file*' after 2003-298T14:20 replace format 'utc'
% fei5get image1 'file*' between 2003-298T14:20 and 2003-298T14:50 format 'utc'
```

EXAMPLE Using an options file

```
% cat getlist.txt
image1 file0 output /home/ops/data         # directing output
image1 file1 replace                       # use replacing
image1 file* output /home/ops/data replace # use wild card


% fei5get using getlist.txt
Getting from file type "image1":
     1. Got "file0".

Getting from file type "image1":

     2. Got "file1".

Getting from file type "image1":

     3. Got "file0".
     4. Got "file1".
     5. Got "file2".
     6. Got "file3".
     7. Got "file4".
     8. Got "file5".
```

- **fei5guardian**

DESCRIPTION

A script that invokes criteria tests on files to determine if the FEI operation should be invoked. The return status is 0 if all files satisfy all criteria; else >0 for errors.

SYNOPSIS

```
fei5guardian [ –f filename | -l fileList ]
             [-c criteriaName | -q queryList ]
             [ -o | -a ]  [ -s ] [ -h ]
             [feiCommandWithArgs]
```

OPTIONS

-f filename	-	Name, or path, of the file to test. Can be used more than once to handle a file set.
-c criteriaName	-	Name of criterion script to invoke. Location must be in PATH environment variable, or the criteriaName must be an absolute path. . Can be used more than once to handle multiple criteria.
-l fileList	-	File containing list of files to test.
-q criterionList	-	File containing list of criteria to use.
-a	-	And flag. File(s) is successful if it passed all criteria. (default)
-o	-	Or flag. File(s) is successful if it passed at least one criterion.
-s	-	Prints status of each file and criteria.
-h	-	Prints this message and exits.
feiCommand…	-	Any FEI command that accepts filenames as last arguments. Invoked only if files pass the criteria test. If no feiCommand is supplied, then the return value of the script is 0 (all files passed all criteria); 1 otherwise. If a feiCommand is supplied and all files passed, the exit value will be of the executed feiCommand; 1 otherwise.

OPTIONS CONSTRAINTS:

Either one or more –f options or one –l option.

Either one or more –c options or one –q option.

SINCE

Release 2.0.0

EXAMPLE

In this example, we want to ensure that only JPEG image files are added to a particular file type, MDMS_DEV:jpegs. First, we write a criterion script that accepts a filename and returns true (0) if the file is a JPEG; false (non-zero) otherwise. An extremely simple version of this script (which we will call *isJpg.sh,* located in the directory referenced by environment variable *$CRITHOME*) might be:

```
% cat isJpg.sh
#!/bin/sh
FILENAME=${1}
RESULT_VAL=”1”
echo $FILENAME | grep “.[jJ][pP][gG]$” > /dev/null 2>&1
if [ $? –eq 0 ]
       then RESULT_VAL=”0”
```

```
fi
exit $RESULT_VAL
```

Now, we can use script as follows:

```
$fei5guardian –f foo.jpg –c $CRITHOME/isJpg.sh fei5add MDMS_DEV:jpegs
```

Because *foo.jpg* will pass the criterion script, the following command will be execute:

```
fei5add MDMS_DEV:jpegs foo.jpg
```

If we replaced *foo.jpg* with *bar.img*, then the criterion script would return false and the fei5add command would never execute.

- **fei5list**

DESCRIPTION

This utility gets a list of file names from the FEI5 server. If the file name expression is omitted, all files of the specified type are listed. If a file name expression is used, it is evaluated by the FEI5 server and matching file names are returned. The file name expression is evaluated by the FEI5 server, so it must be surrounded by single/double quotes to protect it from local shell/terminal evaluation.

SYNOPSIS

```
fei5list [<server group>:]<file type> ['<file name expression>']
        {[before|after <date-time>] |
          [between <date-time1> and <date-time2>]}
        [format '<date format>'] {[long | verylong]}
        [query <queryfile>] [help]
```

SINCE

Release 1.4.0.

KEYWORDS

- ***after***: get files with FEI5 modification dates after the specified time value. Default: none.
- ***before***: get files with FEI5 modification dates before the specified time value. Default: none.
- ***between****... and...*: get files with FEI5 modification dates between the specified time range. Default: none.
- ***format***: overrides default date-time format with the specified pattern or name. Default: ccsdsa.
- ***help***: display the command line help message.
- ***long***: lists output with file modification date and file size in bytes. Default: off.
- ***query***: Name of a query file containing a query to a meta-data database. This requires the server to be configured for this capability.
- ***verylong***: lists output with file modification date, file size in bytes, contributor, record creation date, checksum, and any associated comments and archive notes. Note: The FEI5 administrator can add a logDeleteRecord option to a filetype. If this is done, the server will mark the database record as a deleted file and delete the file from disk. If this file is ever added again, the database record marked as deleted will be used and updated for the newly added file. The **fei5list** with the **verylong** option will show the creation date of the original file in this listing. Default: off.

EXAMPLE

```
% fei5list image1
file5
file6
file1
file0
```

```
% fei5list image1 "file*"
file5
file6
file1
file0

% fei5list image1 "file*" long

    1. 2004-03-16T14:52:32.793,         4310, file5
    2. 2004-03-16T14:52:33.586,         4310, file6
    3. 2004-03-16T14:52:54.020,         4310, file1
    4. 2004-03-16T14:53:04.423,         4310, file0

% fei5list image1 "file*" after 2004-03-16T14:52:33 long

    1. 2004-107T14:52:33.586,         4310, file6
    2. 2004-107T14:52:54.020,         4310, file1
    3. 2004-107T14:53:04.423,         4310, file0
```

- **fei5locktype**

DESCRIPTION

This utility allows the user to lock a filetype for read only. The lock can be done at the owner level or a group level. Locktype capability is required.

SYNOPSIS

***fei5locktype* [<server group>:]<file type> [owner | group] [help]**

SINCE

Release 2.1.4

KEYWORDS

- ***help***: display the command line help message.

EXAMPLE Change password

```
% fei5locktype
Changing password for user 'username'
Enter new password (or type "abort" to quit)>>
Re-enter new password>>
```

- **fei5makeclean**

DESCRIPTION

This utility is similar to **fei5delete** but without file name expression support. Its purpose is to quickly remove all files registered to an existing file type.

SYNOPSIS

***fei5makeclean* [<server group>:]<file type> [*help*]**

SINCE

Release 1.4.0.

KEYWORDS

- ***help***: display the command line help message.

EXAMPLE

```
% fei5makeclean image1
     1. Deleted "file6.jpg".
     2. Deleted "file7.jpg".
     3. Deleted "file1.jpg".
     4. Deleted "file2.jpg".
     5. Deleted "file3.jpg".
     6. Deleted "file4.jpg".
     7. Deleted "file5.jpg".
```

- **fei5notify**

DESCRIPTION

This utility is a file availability metadata subscriber. The user begins a notification subscription session to a file type. When the FEI5 server has a file added/replaced to that file type, the client will be notified with metadata on the new/updated file. The notification events can be routed to standard-out, standard-error, log file, and email. These events can also invoke/trigger an external system process. The external system process is expected to return 0 (zero) as its success exit status or return 1 if an error occurs. Each notification session will create a restart cache file (named '<sg>.<ft>.notify', where <sg> and <ft> are the server group and file type names, respectively) in a hidden subdirectory, called '.shadow', in the output directory, or the current working directory. The restart cache data is used to assist in restarting a previous notification session. An options file is typically provided with the **using** parameter and contains the options for controlling the subscription. Note that some parameters can be specified either on the command line or in the options file.

A notification subscription can be submitted as a background process on a UNIX system. If the user logs out of their current session window, the subscription will continue to run until it is killed with a UNIX kill command.

WARNING: Do not start multiple notification sessions for the same file type from the same directory. This can corrupt the restart file and possibly miss notification of files. Also, each notification session can consume up to 30 MB (40 MB for a push subscription) of memory on the host machine. Ensure the host machine contains ample memory for the number of subscriptions you will be starting.

NOTE: The fei5notify command is a launcher script that invokes a Java Virtual Machine (JVM) process. When running a subscription as a background process, there will be two process ids associated with the session (one for the launcher, one for the JVM). The recommended method of killing a session is to kill the process associated with the JVM, which will then allow the launcher process to exit. The *pgrep* or *ps* Unix command can be used to assist in locating the process id for the JVM process. See Appendix F for a sample script on one way to keep subscriptions alive and how to stop subscriptions easily.

SYNOPSIS

```
fei5notify [<server group:]<file type>
                  {[output <path>] [restart] [using <option file>]}
                  [pull|push] [format '<date format>'] [query <queryfile>]
                  [help]
Option File Format:
        invoke          <command>
        invokeExitOnError
        invokeAsync
```

```
logFile          <file name>
logFileRolling {monthly|weekly|daily|hourly|minutely|halfdaily}
mailMessageFrom <email address>
mailMessageTo    <email address, email address, ...>
mailReportAt     <hh:mm am|pm, hh:mm am|pm, ...>
mailReportTo     <email address, email address, ...>
mailSMTPHost     <host name>
```

SINCE

Release 1.7.0.

KEYWORDS

- ***format:***overrides the default date-time format with the specified pattern or name. Default: ccsdsa.
- ***help***: display the command line help message.
- ***invoke***: invokes a command each time a new file is written to the local system. The command might do something like filtering file names. A command can be any executable from the user system's command processor, or shell, so Java, C, C++ or Perl programs are appropriate. This utility expects the invoked process to return a 0 (zero) as its success exit status, otherwise an error message will be logged. The command must be enclosed in double quotes. The command can contain six variables that **fei5notify** will replace before the command in invoked: Default: none.
  - *$fileName*: the fully qualified name of the file just arrived. **Since Release 2.0,** it uses current working directory or output for path.
  - *$fileNameNoPath*: the name of the file just arrived without the path. **Since: Release 1.7.2.**
  - *$filePath*: the path to the output directory. See ***output*** keyword for detail.
  - *$fileType*: The subscription session's file type.
  - *$serverGroup*: the server group name. **Since: Release 1.7.2.**
  - *$comment*: the attached comment (if any) associated to the file. If none, it will be replaced with the word NULL. **Since: Release 1.7.5. Note:** When using the $comment field in a script where $comment is more than one word, make sure the whole string is ingested into a variable as it is usually viewed by a script as multiple parameters
- ***invokeExitOnError***: if this keyword is specified, the invocation expects the command invoked to return an exit value of 0 for success. If the exit value is not zero and this keyword is specified, then an error is logged and the notification session terminates without updating the restart information. This allows the user to correct the problem and run **fei5notify** with **restart** without losing any data. Without this option, the client will continue processing the next file whether a success or failure occurs.
- ***invokeAsync:*** provides a mechanism for an invoked command to continue with the next result without waiting for the invoke to finish. For example, an invoked script takes a bit of time and user doesn't want to be held back too long. There is a max number of invocations that can occur asynchronously. Once this limit is reached, the serverproxy waits until a thread becomes available. This option is incompatible with the **invokeExitOnError** option, which only works when serially processing results are one at a time.
- ***logFile***: error and other messages are normally sent to standard error and standard out. Supply this keyword along with the name of a file to have messages written to a log file instead. It is recommended to place the log file on local disk or a different disk from where the subscription was started. Default: none.
- ***logFileRolling***: long subscription session with ***logFile*** specified can generate large log file that can be difficult to navigate. Use this keyword to tell the session to create new log file periodically. To prevent this utility from creating zero-length log files. This utility only creates a rolling file when there are messages logged since the last rolling period. Default: none.
  - *daily*: creates a new log file at midnight of each day. The old log file will be renamed with '.yyyy-MM-dd' extension.
  - *halfdaily*: creates a new log file at midnight and midday of each day. The old log file will be renamed with '.yyyy-MM-dd-a' or '.yyyy-MM-dd-p' extension.
  - *hourly*: creates a new log file at the top of each hour. The old log file will be renamed with '.yyyy-MM-dd-HH' extension.
  - *monthly*: creates a new log file at the beginning of each month. The old log file will be renamed with '.yyyy-MM' extension.
  - *minutely*: creates a new log file at the beginning of every minute. The old log file will be renamed with '.yyyy-MM-dd-HH-mm' extension.

    - *weekly*: creates a new log file at the first day of each week. The old log file will be renamed with '.yyyy-ww' extension.
- ***mailMessageFrom***: the sender email address used to send email messages and reports. Should also provide mailMessageTo and mailSMTPHost. Default: none.
- ***mailMessageTo***: list of receiver(s) email address. Email addresses are separated by comma ','. Email a message when the subscription starts, fails, aborts, or restarts. Requires the options mailMessageFrom and mailSMTPHost. Default: none.
- ***mailReportAt***: a single time or list of times separated by commas in the day to send a notification report. If no new files are received since the last report, then no email report will be sent. Format is <hh:mm am | pm>. Default: none.
- ***mailReportTo***: list of receiver(s) email address. Email addresses are separated by comma ','. Send a report of available files and if an invoke script was processed on the file. Any errors from the invoke command are are also sent. Requires the options mailMessageFrom, mailSMTPHost and mailReportAt. Default: none.
- ***mailSMTPHost***: the email SMTP host name. It uses port 25 to send email. Default: none.
- ***output***: the output directory for restart information to be kept. Default: current directory. **Since: Release 1.7.2.**
- ***pull***: start the session where the client periodically queries the server, default is every minute, to find out if new files are available. Default: on.
- ***push***: start the session where the server notifies the client when new files are available. Requires 'push' capabilities due to additional memory used by the server. Contact the FEI5 administrator for more details. Default: off. **Since: Release 2.0.0**
- ***query:*** query to a meta-data database. If the server is configured to communicate with a meta-data database, this file will contain the query to search on. Default: none
- ***restart***: to restart from the previous subscription session and get all files added since the previous subscription session. If a restart file does not exist, one will be created with the current time. If the restart file is corrupted, the program will attempt to restart from a backup restart file, if one exists or else a new restart file with the current time will be created. If a restart file exists, but the restart options is not specified, the subscription will reset the time in the file to the current reconnect time. Default: off.
- ***using***: to supply an input options file. Default: none.

EXAMPLE Run in the current session window with a pull subscription

```
% cat notify.opt
logFile /tmp/image1_notify.log
logFileRolling daily
mailMessageFrom ops@jpl.nasa.gov
mailMessageTo ops@jpl.nasa.gov
mailSMTPHost mymail.jpl.nasa.gov

% fei5notify image1 output jpegs restart using notify.opt




FEI5 Information on 2005-02-24T17:23:09.037
Subscribing to [MDMS-DEV:image1] file type.



FEI5 Information on 2005-02-24T17:23:09.037

File type [MDMS_DEV:image1]: available "file6.jpg" 4485 bytes 2005-02-
24T17:22:58.263
...
```

A cntrl-c will stop the subscription.

EXAMPLE Run as a background process on a UNIX system (csh) with a push subscription

```
% cat notify.opt
logFile /tmp/image1_notify.log
logFileRolling daily
mailMessageFrom ops@jpl.nasa.gov
mailMessageTo ops@jpl.nasa.gov
mailSMTPHost mymail.jpl.nasa.gov
invoke "/home/usr/getFileIfNeeded.csh $serverGroup $fileType $fileNameNoPath"

% fei5notify image1 output jpegs push restart using notify.opt >& /dev/null &


% tail /tmp/image1_notify.log


FEI5 Information on 2005-02-24T17:23:09.037


File type [MDMS_DEV:image1]: available "file6.jpg" 4485 bytes 2005-02-
24T17:22:58.263
...

% pgrep -l -f image1
5930 /usr/bin/perl -w /usr/fei/fei5/bin/fei5notify image1 . . .
% ps -ef | grep 5930
UID        PID  PPID C STIME  TTY      TIME     CMD
user      5930 7164 0 14:38  pts/1  00:00:02  /usr/bin/perl -w . . .
user      5935 5930 1 14:38  pts/1  00:00:02  /usr/java . . .
% kill 5935
```

- **fei5register**

DESCRIPTION

This utility registers or creates a record in the FEI5 database for a file that is accessible by the FEI5 server. This avoids doing an additional copy of the data. The command requires the register capability. NOTE: Deletion of a file should be done with the fei5delete command to remove both the file and the record. A fei5unregister will just remove the record from the database. Use of this command requires register capabilities and is restricted to only the batch commands.

SYNOPSIS

```
fei5register [<server group>:]<file type> <file name expression>
   [replace] [force] [comment '<comment test>']
   {[before|after <date-time>] | [between <date-time1> and <date-time2>]}
   [format '<date format>'] [help]

fei5register using <option file>
Option File Format (per line):
    [<server group>:]<file type> <file name>. . . .
    [replace] [force] {[before|after <date-time>] |
    [between <date-time1> and <date-time2>]} [format '<date format>']
    [comment '<comment test>']
```

SINCE

Release 2.0.7b

KEYWORDS

- ***after***: add files with creation times after the specified time value. Default: none.
- ***before***: add files with creation times before the specified time value. Default: none.
- ***between*... *and*...**: add files with creation times between the specified time range values. Default: none.
- ***format:*** overrides default date-time format with the specified pattern or name. Default: ccsdsa.
- ***force:*** overrides the default of checking that the path is the same as the previous registration when used in conjunction with the **replace** option.
- ***comment***: add a description to the file. A comment longer than 255 characters will be truncated. Default: none.
- ***replace***: Replace the fei5 record with the provided information. The file must be in the same directory as the previous registration unless the **force** option is provided. Updates the modified time. Default: off.
- ***help***: display the command line help message.
- ***using***: keyword to supply a list of commands in an options file. Default: none.

EXAMPLE Simple register operation (file exists in the local directory)

```
% fei5register image1 file1
     1. Registered "file1.
```

EXAMPLE Using an options file to register files

```
% cat register.txt
image1 file1
image1 file2

% fei5register using register.txt




1. Registered "file1".
   2. Registered "file2".
```

2.

```
% fei5list image1
file1
file2
```

- **fei5rename**

DESCRIPTION

This utility renames a file on the FEI5 server. An options file can be used to rename multiple files on multiple file types. This command is intended for file type maintenance use. When a new file is published (via add or replace operations) to a file type, it is available for immediate retrieval for users. If the file is renamed after it has been add or replace, it might confuse those who may have retrieved the file under its original name. NOTE: The fei5rename command is not recommended to perform when subscriptions are active on the file type, instead rename with the Unix 'mv' command before adding to FEI.

SYNOPSIS

```
fei5rename [<server group>:]<file type>
           <old file name> <new file name> [help]

fei5rename using <option file>
Option File Format (per line):
```

```
[<server group>:]<file type> <old file name> <new file name>
```

SINCE

Release 1.4.0.

KEYWORDS

- ***help***: display the command line help message.
- ***using***: keyword to supply a list of commands in an options file. Default: none.

EXAMPLE Simple rename operation

```
% fei5rename image1 file0 fido_0
     1. Renamed "file0" to "fido_0".
```

EXAMPLE Using an options file to rename

```
% cat rename.txt
image1 file1 fido_1

% fei5rename using rename.txt



Renaming in file type "image1":


     1. Renamed "file1" to "fido_1".


% fei5list image1
fido_0
fido_1
```

- **fei5replace**

DESCRIPTION

This utility replaces existing files on the FEI5 server. It works just like **fei5add** when the file does not already exist on the FEI5 server. The fei5replace command requires replace capabilities. Contact the FEI5 administrator for more information.

SYNOPSIS

```
fei5replace [<server group>:]<file type> <file name expression>
            {[before|after <date-time>] |
             [between <date-time1> and <date-time2>]}
            [format '<date format>'] [receipt] [diff]
            [comment '<comment text>'] [crc] [autodelete] [help]

fei5replace using <option file>
Option File Format (per line):
        [<server group>:]<file type> <file name>...
        {[before|after <date-time>] |
         [between <date-time1> and <date-time2>]}
        [format '<date format>'] [receipt]
        [comment '<comment text>'] [autodelete] [crc]
```

SINCE

Release 1.4.0.

KEYWORDS

- ***after***: get files with FEI5 modification dates after the specified time value. Default: none.
- ***autodelete***: automatically delete local file after successfully added to the server. Default: off. **Since: Release 1.7.0.**
- ***before***: get files with FEI5 modification dates before the specified time value. Default: none.
- ***between*... *and*...**: get files with FEI5 modification dates between the specified time range value. Default: none.
- ***comment***: any associated comment for the file. A comment longer than 255 characters will be truncated. Default: none.
- ***crc***: calculate the CRC value for each file. Default: off.
- ***diff***: replace the file only if the file in the output directory is different than the one on the server.
- ***format:*** overrides default date-time format with the specified pattern or name. Default: ccsdsa.
- ***help***: display the command line help message.
- ***receipt***: file delivery status is logged in FEI5 server and a tracking receipt ID is sent to the user. Checksum validation is enabled. Requires 'receipt' capability. Contact the FEI5 administrator. Default: off. **Since: Release 1.7.1.**
- ***using***: to supply a list of commands in an options file. Default: none.

EXAMPLE

```
% fei5replace MDMS_DEV:image1 file0
     1. Replaced "file0".

% fei5replace MDMS_DEV:image1 file0 file1


     1. Replaced "file0".


     2. Replaced "file1".

% fei5replace MDMS_DEV:image1 file*
     1. Replaced "file0".
     2. Replaced "file1".
```

- **fei5subscribe**

DESCRIPTION

This utility is a file subscriber. The user begins a subscription session to a file type. When the FEI5 server has a file added/replaced to the file type, the new/updated file will be automatically delivered to the subscription client. The subscription event messages can be routed to standard-out, standard-error, log file, and email. These events can also invoke/trigger an external system process. The external system process is expected to return 0 (zero) as its success exit status and 1 as its error exist status. Each subscription session will create a restart cache file (named '<sg>.<ft>.restart', where <sg> and <ft> are the server group and file type names, respectively) in a hidden subdirectory, called '.shadow', in the specified output directory, or current working directory. The restart cache data is used to assist in restarting a previous subscription session. An options file is typically provided with the **using** parameter and contains the options for controlling the subscription. Note that some parameters can be specified either on the command line or in the options file.

A file subscription can be submitted as a background process on a UNIX system. If the user logs out of their current session window, the subscription will continue to run until it is killed with a UNIX kill command.

WARNING: Do not start multiple subscription sessions for the same file type from the same directory. This can corrupt the restart file and possibly miss files. Also, each subscription session can consume up to 60 MB of memory (80 MB for a push subscription) on the host machine. Ensure the host machine contains ample memory for the number of subscriptions you will be starting.

NOTE: The fei5subscribe command is a launcher script that invokes a Java Virtual Machine (JVM) process. When running a subscription on a UNIX host as a background process, there will be two process ids associated with the subscription (one for the launcher, one for the JVM). The recommended method of killing a subscription is to kill the process associated with the JVM, which will then allow the launcher process to exit. The *pgrep* or *ps* Unix command can be used to assist in locating the process id for the JVM process. See Appendix F for a sample script on one way to keep subscriptions alive and how to stop subscriptions easily.

SYNOPSIS

```
fei5subscribe [<server group:]<file type>
              {[output <path>] [restart] [using <option file>]} [pull|push]
              [replace|version] [format '<date format>']
              [query <queryfile>] [replicate] [replicateroot <rootdir>]
              [filehandler] [diff] [help]
Option File Format:
        crc
        diff
        filehandler
        format         '<date format>'
        invoke          <command>
        invokeExitOnError
        invokeAsync
        logFile          <file name>
        logFileRolling {monthly|weekly|daily|hourly|minutely|halfdaily}
        mailMessageFrom <email address>
        mailMessageTo   <email address, email address, ...>
        mailReportAt    <hh:mm am|pm, hh:mm am|pm, ...>
        mailReportTo    <email address, email address, ...>
        mailSMTPHost    <host name>
        output          <path>
        pull or push
        receipt
        replace [diff] or version
        replicate
        replicateroot
        saferead
        diff
```

SINCE

Release 1.7.0.

KEYWORDS

- ***crc***: calculate the CRC value for each file. This automatically turns on resume file transfer. The file will be transferred and the CRC value will be compared to the value returned by the server. If the two CRC values do not match, another attempt to copy the file will be made. The command will attempt a retry up to 3 times. Default: off.
- ***diff***: used in conjunction with the **replace or version** option. Enables a get on replace of a file only if the file is different from an existing file in the output directory otherwise the file is skipped. Default: off.

- ***filehandler:*** Enables file management when using replication where older files will be deleted when a specified threshold is reached. Requires a configuration file. Default: off.
- ***format:*** overrides default date-time format with the specified pattern or name. Default: ccsdsa.
- ***help***: display the command line help message.
- ***invoke***: 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 the user system's command processor, or shell, so Java, C, C++ or Perl programs are appropriate. The command must be enclosed in double quotes. The command can contain six variables that **fei5subscribe** will replace before the command in invoked: Default: none.
  - *$fileName*: the fully qualified name of the file just arrived. **Since Release 2.0,** it uses current working directory or output for path.
  - *$fileNameNoPath*: the name of the file just arrived without the path. **Since: Release 1.7.2.**
  - *$filePath*: the path to the output directory. See ***output*** keyword for detail.
  - *$fileType*: The subscription session's file type.
  - *$serverGroup*: the server group name. **Since: Release 1.7.2.**
  - *$comment*: the attached comment (if any) associated to the file. If none, it will be replaced with the word NULL. **Since: Release 1.7.5. Note:** When using the $comment field in a script where $comment is more than one word, make sure the whole string is ingested into a variable as it is usually viewed by a script as multiple parameters.
- ***invokeExitOnError***: if this keyword is specified, the invocation expects the command invoked to return an exit value of 0 for success. If exit value is not zero and this keyword is specified, then an error is logged and the subscription session terminates without updating the restart information. This allows the user to correct the problem and run **fei5subscribe** with restart without losing any data. Without this option, the client will continue processing the next file whether a success or failure occurs.
- ***invokeAsync:*** provides a mechanism for an invoked command to continue with the next result without waiting for the invoke to finish. For example, an invoked script takes a bit of time and user doesn't want to be held back too long. There is a max number of invocations that can occur asynchronously. Once this limit is reached, the serverproxy waits until a thread becomes available. This option is incompatible with the **invokeExitOnError** option, which only works when serially processing results are one at a time.
- ***logFile***: error and other messages are normally sent to standard-error and standard-out. Supply this keyword along with the name of a file to have messages written to a log file instead. It is recommended to place the log file on local disk or a different disk from where the subscription was started. Default: none.
- ***logFileRolling***: long subscription session with ***logFile*** specified can generate large log file that can be difficult to navigate. Use this keyword to tell the session to create new log file periodically. To prevent this utility from creating zero-length log files. This utility only creates a rolling file when there are messages logged since the last rolling period. Default: none.
  - *daily*: creates a new log file at midnight of each day. The old log file will be renamed with '.yyyy-MM-dd' extension.
  - *halfdaily*: creates a new log file at midnight and midday of each day. The old log file will be renamed with '.yyyy-MM-dd-a' or '.yyyy-MM-dd-p' extension.
  - *hourly*: creates a new log file at the top of each hour. The old log file will be renamed with '.yyyy-MM-dd-HH' extension.
  - *monthly*: creates a new log file at the beginning of each month. The old log file will be renamed with '.yyyy-MM' extension.
  - *minutely*: creates a new log file at the beginning of every minute. The old log file will be renamed with '.yyyy-MM-dd-HH-mm' extension.
  - *weekly*: creates a new log file at the first day of each week. The old log file will be renamed with '.yyyy-ww' extension.
- ***mailMessageFrom***: the sender email address used to send email messages and reports. Should also provide mailMessageTo and mailSMTPHost. Default: none.
- ***mailMessageTo***: list of receiver(s) email address. Email addresses are separated by comma ','. Email a message when the subscription starts, fails, aborts, or restarts. Requires the options mailMessageFrom and mailSMTPHost. Default: none.
- ***mailReportAt***: a single time or list of times separated by commas in the day to send a notification report. If no new files are received since the last report, then no email report will be sent. Format is <hh:mm am | pm>. Default: none.
- ***mailReportTo***: list of receiver(s) email address. Email addresses are separated by comma ','. Send a report of available files and if an invoke script was processed on the file. Any errors

from the invoke command are also sent. Requires the options mailMessageFrom, mailSMTPHost and mailReportAt. Default: none.

- ***mailSMTPHost***: the email SMTP host name. It uses port 25 to send email. Default: none.
- ***output***: the output directory for restart information to be kept and where received files are to be placed. Default: directory where the subscription is started.
- ***pull***: start the subscription where the client periodically queries the server, typically every minute, to find out if new files are available. Default: on.
- ***push***: start the notify subscription where the server notifies the client when new files are available. Requires 'push' capabilities due to additional memory used by the server. Contact the FEI5 administrator for more details. Default: off. **Since: Release 2.0.0**
- ***query:*** query to a meta-data database. If the server is configured to communicate with a meta-data database, this file will contain the query to search on. Default: none.
- ***receipt***: file delivery status is logged in FEI5 server and a tracking receipt ID is sent to the subscriber. Checksum validation is enabled. Requires 'receipt' capabilities. Contact the FEI5 administrator. Default: off. **Since: Release 1.7.1.**
- ***restart***: to restart from the previous subscription session and get all files added since the previous subscription session. If a restart file does not exist, one will be created with the current time. If the restart file is corrupted, the program will attempt to restart from a backup restart file, if one exists or else a new restart file with the current time will be created. If a restart file exists, but the restart options is not specified, the subscription will reset the time in the file to the current reconnect time. Default: off.
- ***replace [diff]***: replace existing files within the output directory. The optional diff parameter will replace files only if they differ from ones found in the output directory. Default: off.
- ***replicate***: place the copied file in the same location as the server location. Assumes the user is replicating the server directory structure and that the user has permissions to create directories and files. Default is to copy the file into current working directory or the **output** location.
- ***replicateroot***: prefix to append to the server path name.
- ***saferead***: files are only appear to in the output directory until it has been fully received. Default: off.
- ***version***: create versioning of existing files within the specified output directory. If a newly retrieved file already exists within user's output directory, the current copy will be renamed by attaching a time value to the end of the file before writing the newly retrieved file to the output directory. Default: off.
- ***using***: to supply an input options file. Default: none.

EXAMPLE Run in the current session window

```
% cat subscribe.opt
replace
logFile /usr/logs/image1_subscribe.log
logFileRolling daily
mailMessageFrom ops@jpl.nasa.gov
mailMessageTo ops@jpl.nasa.gov
mailSMTPHost mymail.jpl.nasa.gov
invokeExitOnError
invoke "/home/bin/createNextOrderProduct.csh $fileName"

% fei5subscribe image1 output jpegs using subscribe.opt


FEI5 Information on 2005-02-24T17:49:52.982
Subscribing to [MDMS_DEV:image1] file type.


FEI5 Information on 2005-02-24T17:50:54.473
```

```
File type [MDMS_DEV:image1]: received "file7.jpg" 4485 bytes 2005-02-
24T17:50:16.523

...
```

A cntrl-c will stop the subscription.

EXAMPLE Run as a background process on a UNIX system (csh) with a push subscription

```
% fei5subscribe image1 output jpegs push restart using subscribe.opt >& /dev/null &

FEI5 Information on 2005-02-24T17:49:52.982
Subscribing to [MDMS_DEV:image1] file type.



FEI5 Information on 2005-02-24T17:50:54.473


File type [MDMS_DEV:image1]: received "file7.jpg" 4485 bytes 2005-02-
24T17:50:16.523


...


% tail -f /tmp/image1_subscribe.log
FEI5 Information on 2005-02-24T17:50:54.473


File type [MDMS_DEV:image1]: received "file7.jpg" 4485 bytes 2005-02-
24T17:50:16.523

...
```

EXAMPLE Kill the subscription by killing the JAVA process. Linux example.

```
% pgrep -l -f image1
5930 /usr/bin/perl -w /usr/fei/fei5/bin/fei5subscribe image1 . . .
% ps -ef | grep 5930
UID        PID  PPID C STIME  TTY      TIME     CMD
user      5930 7164 0 14:38  pts/1  00:00:02  /usr/bin/perl -w . . .
user      5935 5930 1 14:38  pts/1  00:00:02  /usr/java . . .
% kill 5935

or pkill can be used if pgreg shows the whole java command.


% pkill -l -f "Client image1"
```

EXAMPLE Kill the subscription by killing the JAVA process. MAC example.

```
% ps -xjww | grep image1
UID   PID  PPID  PGID SESS JOBC STAT TT      TIME      CMD
user  322     1   322 6afd368    0 S     ??    0:00.05 /usr/bin/perl -w . . .
user  324   322   322 6afd368    0 S     ??   11:18.45 /System/Library/Framewor . . .
% kill 324
```

- **fei5unlocktype**

DESCRIPTION

This utility allows the user to unlock a filetype that was locked. The unlock should be done at the owner level or a group level. Locktype capability is required.

SYNOPSIS

fei5unlocktype **[<server group>:]<file type> [owner | group] [help]**

SINCE

Release 2.1.4

KEYWORDS

- ***help***: display the command line help message.

EXAMPLE Change password

```
% fei5unlocktype
Changing password for user 'username'
Enter new password (or type "abort" to quit)>>
Re-enter new password>>
```

- **fei5unregister**

DESCRIPTION

This utility unregisters or deletes a record in the FEI5 database for a file, but does not delete the file. The command requires the register capability. NOTE: Deletion of a file should be done with the fei5delete command to remove both the file and the record.

SYNOPSIS

```
fei5unregister [<server group>:]<file type> '<file name expression>'
    [help]

fei5unregister using <option file>
Option File Format (per line):
    [<server group>:]<file type> '<file name expression>'
```

SINCE

Release 2.0.7b

KEYWORDS

- ***help***: display the command line help message.
- ***using***: keyword to supply a list of commands in an options file. Default: none.

EXAMPLE Simple unregister operation (file exists in the local directory)

```
% fei5unregister image1 file1
1. Unregistered "file1.

% fei5list image1 file1
No files found.
```

4. Using `fei5` – Interactive Utility

This section discusses how to start and end a FEI5 interactive session, how to get help, and how to access the current file types. The interactive client session of FEI5 is equipped with a well-documented user help facility to provide online help information to each of the interactive commands within the client software.

The command line utility **`fei5`** is the launcher for the FEI5 client session. On the command line prompt, type the command **`fei5`** to launch the application

```
% fei5
```

The FEI5 session begins by displaying the current domain file reference and the current FEI5 version information.

```
Domain file: /home/ops/fei5/config/domain.fei
FEI5 release 2.0.0a, April 2006
Komodo API release 2.9.1, April 2006
Copyright 2002-2006 Jet Propulsion Laboratory, Caltech, NASA
```

The software begins with the command prompt. The login procedure begins by the user typing the **`login`** command. The software requests the user login information for the FEI5 server. This information will be cached for the session only and will be used when the user initiates a connection to the FEI5 server. The user may choose to run **fei5kinit** prior to entering **fei5**. **`Fei5kinit`** will create a cached login file that will be used by **fei5** to connect to the FEI5 server.
The user will not have to issue the login command for the session as long as a cached login file exists. See section 3 for more information.

```
>> login
User name>> ops
Password>> ********
```

Again, the user login information is cached for future connections to any FEI5 file types for the session. If the domain file has a default server group entry, the server group name will be displayed in the prompt. The example below shows `MDMS_DEV` as the default server group. At this point, the user is not yet connected to the FEI5 server; it is only an indication that the user has successfully started the FEI5 client application. To establish connection to a given file type, for example `image1`, the user should use the '`use`' command, use.

```
MDMS_DEV:>> use image1
Using file type : image1
MDMS_DEV:image1>>
```

For example, assume that there is another file type named `image1`, but it is registered under the server group `MDMS_OPS`. To connect to that file type, the user must specify the server group name in the use command.

```
MDMS_DEV:image1>> use MDMS_OPS:image1
Using file type MDMS_OPS:image1
MDMS_OPS:image1>>
```

`or alternatively:`

```
MDMS_DEV:image1>> setdefaultgroup MDMS_OPS
Default server group changed to MDMS_OPS
```

```
Select a file type in the group with the use command.

MDMS_OPS:>> use image1
Using file type : image1
```

The **use** command initiates a connection to the FEI5 server that hosts the file type `image1`. There are several criteria that must be verified by the server before the server will authorize the connection request. The server checks that:

1. The file type is defined in the client domain file.
2. The client distribution has the valid SSL certificate to communicate with the server.
3. The user is defined in the FEI5 server registry.
4. The cached user password matches the password registered with the FEI5 server.
5. The user is authorized to connect to the specified file type.

Upon successful authentication, the client session command prompt is changed to display the connected file type name. Based on the user's assigned capabilities by the FEI5 administrator, the user is now able to perform some or all of the common FEI5 operations, **add**, **delete**, **get**, **replace**, **rename**, etc..

EXAMPLE 1 To add a file

```
MDMS_DEV:image1>> add ilenam.jpg
File added: ilenam.jpg
MDMS_DEV:image1>>
```

EXAMPLE 2 To get a file

```
MDMS_DEV:image1>> get ilenam.jpg
Got: ilenam.jpg
MDMS_DEV:image1>>
```

Getting help

Type **help**, **h**, or **?**, at the FEI5 prompt to list all available FEI5 commands. The listing organizes the commands into four categories: Set (environment), Utility, File Type, and Virtual File Type. Commands are case-insensitive (e.g. **showlatest** and **showLatest** are the same).

```
MDMS_DEV:image1>> help
•   Settings commands *
abort            autodelete       computechecksum
diff             echo             log
preserve         receipt          replacefile
replicate        restart          saferead
test             timer            verbose
versionfile      veryverbose
•   Utility commands *
batch           bye             cd
changepassword  dateformat      exit
help            history         logcmds
logfile         login           ls
pause           pwd             quit
set             showdomainfile  version
•   Filetype commands *
add               addandref         archive
checksum          comment           delete
get               getafter          getbetween
getlatest         getsince          lockfiletype
```

```
makedomainfile    rename            replace
setdefaultgroup   show              showafter
showbetween       showcapabilities  showlatest
showsince         showtypes         unlockfiletype
use
```

- **VFT commands ***

```
addreference     addvft           addvftreader
cancelreference  delreference     delvft
delvftreader     getvft           setreference
showvft          showvftreaders   updatevft
```

More information about a specific command is available by specifying the command name with help.

```
MDMS_DEV:>> help use
Command:     use
Description: Change current file type in session. The server group
             parameter is optional, defaultGroup is specified in
             domain file.
Usage:       use [<server group>:]<file type>
Type:        filetype
```

Capabilities

A capability specifies the action(s) the user can perform on a given file type and the files within. Capabilities are set on a per-user basis by the FEI5 server administrator at the server level or at the filetype level. A user can view their capabilities by typing the **showcapabilities** command. The **showcapabilities** command only shows the highest access level.

```
MDMS_DEV:>> showcapabilities
User access: write_all, VFT
File type capabilities:
image1: add, get, replace
image2: add, delete, get, rename, replace
image3: get
image4: get
```

In this example, the user has both user access and file type capabilities. The user capabilities set at the server level can be one of three values: read_all, write_all, or admin. VFT is shown if set. The file type capabilities have more values. The file type name is left of the colon in the output. The capabilities associated with that file type are on the right of the colon in the output. The following table shows the available capabilities.

CAPABILITIES	LEVEL	ALLOWS USERS TO…
add	filetype	add, get, show, comment files
admin	server	administer the server, add, delete, replace, get, show, comment on all file types
archive	filetype	archive

CAPABILITIES	LEVEL	ALLOWS USERS TO...
delete	filetype	delete files
get	filetype	get, show/list files
locktype	filetype	lock a file type from further modification, but allow gets and shows
push	filetype	invoke push subscriptions
read_all	server	get, show on all file types
receipt	filetype	generate a receipt record in the database
register	filetype	allows the users to use the register capability.
rename	filetype	rename a file
replace	filetype	add, replace, get, show, comment
replicate	filetype	Allows the user to replicate the server location on the user's local disk.
vft	filetype	virtual file type operations
VFT	server	virtual file type operations on all file types
write_all	server	add, delete, replace, get, show, comment on all file types
subtype	filetype	create sub-type

Working with file types

FEI5 organizes files by base file types or simply file types, which is a virtual interface to the file system. Users do not have to know the physical location of the file types and the files they manage, because of the FEI5 file type abstraction. FEI5's data store also keeps metadata about each file within a file type.

From within the FEI5 client session, the user can see the list of file types that are registered within the domain file

```
MDMS_DEV:>> showTypes
MDMS_DEV:image1
MDMS_DEV:image2
MDMS_DEV:image3
MDMS_DEV:image4
```

In this example, there are four file types. These file types come from the domain file. If a new file type is added, the FEI5 administrator will notify the user and the user will use the **makedomainfile** to create a new domain.fei file that will need to replace the domain.fei file in the $FEI5/ directory.

MDMS_DEV: image1>> makedomainfile domain.fei

Making connection

Users connect to a file type with the **use** command

```
MDMS_DEV:>> use image1
Using file type : image1
MDMS_DEV:image1>>
```

FEI5 displays the name of the server group and file type behind the prompt to inform the user of the file type to which they are currently connected. All file type commands (i.e. **add**, **get**, **delete**) are within the context of the named file type.

Adding, replacing, deleting and renaming files

The command to add a file (e.g. **filenam.jpg**) to a file type is:

```
MDMS_DEV:image1>> help add
Command:     add
Description: Add files matching file name expression to current file type
Usage:       add <file name expression>[<"comment">]
Type:        filetype

MDMS_DEV:image1>> add filenam.jpg
File added: filenam.jpg
```

FEI5 responds to the command to indicate what occurred. In the example, the file was added successfully. If the user tried to add a file with the name **filenam.jpg** again, the command will fail. FEI5 will not add a file that is already exists in the system, but we can replace it using the command:

```
MDMS_DEV:image1>> help replace
Command:     replace
Description: Replace a registered file
Usage:       replace <file name> [<"comment">]
Type:        filetype

MDMS_DEV:image1>> replace filenam.jpg
File replaced: filenam.jpg
```

The **replace** command can also add a file if it does not already exist in the FEI5 server. Therefore, **replace** is like add without the protection of stopping users from overwriting an existing file.

The command to delete a file (e.g. **filenam.jpg**) from a file type is:

```
MDMS_DEV:image1>> help delete
Command:     delete
Description: Delete registered file(s)
Usage:       delete <file name expression>
Type:        filetype

MDMS_DEV:image1>> delete filenam.jpg
"filenam.jpg" has been deleted from file type "image1".
```

The command to rename a file is:

```
MDMS_DEV:image1>> help rename
Command:     rename
Description: Rename a registered file
Usage:       rename <original file name> <new file name>
Type:        filetype

MDMS_DEV:image1>> rename filenam.jpg io.jpg
File filenam.jpg renamed to io.jpg
```

If the file is renamed after it has been published, it might confuse those who may have retrieved the file under its original name. NOTE: Not recommended to perform when subscriptions are active on the file type, instead rename before adding to FEI.

Using wild-cards with file names

FEI5 supports a simple form of file globing that allows adding or replacing a group of files without typing in each file's name. The asterisk character '*' stands for zero or more characters in a file name. To add all files from a local directory to an FEI5 file type, issue the following command:

```
MDMS_DEV:image1>> add *
```

To show all files in a file type that match an expression (i.e. **ilenam***), issue the following command:

```
MDMS_DEV:image1>> show ilenam*
```

Showing lists of files

FEI5 provides commands to show files in a file type. The commands that show what is in a file type begin with **show**. FEI5 has commands to show a single file, files using wild-card, and files that fall into some time frame reference.

```
MDMS_DEV:image1>> help show
Command:     show
Description: Show registered files in current file type
Usage:       show <file name expression>
                  [invoke "<system command>"]
Type:        filetype

MDMS_DEV:image1>> show
jovianMoons.jpg
jovianRings.jpg
ilenam.jpg
io.jpg
```

Getting files

FEI5 provides commands to get files from a file type into the current directory. These operations are carried out by two collections of commands. The commands that get what is in a file type begin with **get**. FEI5 has commands to get a single file, files using wild-card, and files that fall into some time-frame reference.

```
MDMS_DEV:image1>> help get
Command:     get
Description: Get one or more files from current file type.  If external system
process invocation is specified, then it executes the system command for each
file received.
Usage:     get <file name expression> [invoke "<system command>"]
Type:      filetype

MDMS_DEV:image1>> get jovianMoons.jpg
Got: jovianMoons.jpg
```

By default, the get operation will not retrieve a file that exists in the user's current working directory. To override this behavior, the user can choose to replace the file by using the **set replaceFile on** command or to version the file by using the **set versionFile on** command. The user can use the FEI5 utility command **cd** to change the current working directory.

Process invocation

A FEI5 user may perform additional processing to the file it has just acquired from FEI5. A special keyword **invoke** is built into all FEI5 **get and show (includes getafter, showbetween, etc.)** operations to allow the user to invoke an external process. There are six predefined parameters available to the invoke string to serve as variables to the invoke command. Variable **$fileName** represents the fully-qualified name of the file just acquired. Variable **$fileType** represents the file type name. Variable **$fileNameNoPath** represents the file name without the directory path. Variable **$filePath** represents the file path. Variable **$serverGroup** represents the server group name. Variable **$comment** represents the file's associated comment. If no comment is associated with a file, the variable returns the string NULL.

```
MDMS_DEV:image1>> get mission_images.tar.gz invoke "tar -zxvf $fileName"
Invoke command 'tar -zxvf /home/ops/output/mission_images.tar.gz'

MDMS_DEV:image1>> get image1 invoke "./runthis.csh $fileNameNoPath"
Invoke command './runthis.csh image1'
```

Date/Time specific commands

FEI5 supports commands for adding and showing files based on a specific time or date inputs. The date and time used by FEI5 is the time at which the file entered the FEI5 file system. See section 3 for additional date formats that are allowed by **fei5**.

The date/time based commands are: **getAfter**, **getBetween**, **getLatest**, **showAfter**, **showBetween**, **showLatest**

The date time format is CCSDSA by default. Use the **dateformat** utitlity command to show the current format and to change the format. If the dateformat is changed, the format of the date/time based commands should conform to the new format.

Named formats (i.e. "ccsdsa", "utc") date inputs can be specified without a time, which implies midnight for that date. All other formats must be completely defined. See Section 3 for additional information on date formats.

EXAMPLES Date/time input

```
MDMS_DEV:image1>> help showafter
Command:        showafter
Description: Show files registered after a given date
Usage:       showAfter (yyyy-MM-ddThh:mm:ss.SSS)
             [invoke "<system command>"]
Type:        filetype
MDMS_DEV:image1>> showAfter 2003-09-14T14:23:00.384
MDMS_DEV:image1>> showAfter 2003-09-14T14:23
MDMS_DEV:image1>> showAfter 2003-09-14
```

- **The latest file**

  Sometimes the user is only interested in the latest files

```
MDMS_DEV:image1>> showLatest
io.jpg
MDMS_DEV:image1>> getLatest
Got file: io.jpg
```

  File name expression can also be used when querying for the latest files

```
MDMS_DEV:image1>> showLatest ilenam*
ilenam.jpg
MDMS_DEV:image1>> getLatest ilenam*
Got file: ilenam.jpg
```

- **Environment options**

  The FEI5 client supports user options that change the expected output of FEI5 commands. FEI5 environment options use the following syntax (See appendix A for a description of each setting):

```
set <command> {on, off}
```

  Simply type the **set** command without any arguments to display the current environment settings.

```
MDMS_DEV:>> set
abort                on
autoDelete           off
computeChecksum      off
diff                 off
echo                 on
log                  on
preserve             off
```

```
receipt              off
replaceFile          off
replicate            off
restart              off
safeRead             off
test                 off
timer                off
verbose              on
veryverbose          off
versionFile          off
```

Verbosity modes

Setting the verbosity modes within the FEI5 client determines what information is displayed by the **show** commands (i.e. **show**, **showAfter**, **showBetween**, etc). There are three verbosity modes:

- *default* (Non-Verbose): Displays the file name only. This is the default state of verbosity.
- *verbose*: Displays the file name, file size in bytes and the date and time the file was received.
- *veryVerbose*: Displays verbose output and any comment, archive note and checksum data associated with the file.

Users can enable verbosity by issuing the following commands:

```
MDMS_DEV:image1>> set verbose on
MDMS_DEV:image1>> show
    1. 2004-08-05T12:23:19.373,   1048576000, bigfile.tst

MDMS_DEV:image1>> set veryVerbose on
MDMS_DEV:image1>> show
    1. 2004-08-05T12:23:19.373,   1048576000, bigfile.tst
       [Contributor] opt
       [Created] 2004-08-05T11:32:42.473
       [Comment] "a large file test"
```

Users can disable verbosity by issuing the following command (setting **verbose off** will set **veryVerbose off** as well):

```
MDMS_DEV:image1>> set verbose off
```

Insuring file integrity

FEI5 uses file checksums to ensure a file's integrity when transmitted from the server to a client. Checksums are not enabled by default in FEI5 due to the overhead required to compute them. If enabled, a checksum is computed for the file as it is being transmitted or received, depending on the operation. For 'get' operations, the receiver (client) also calculates the checksum as the data is received. The sender (server) checksum is transmitted to the receiver after the file has been fully transmitted. The sender and receiver checksums are compared. If the file was altered during transmission, the checksum values will not match and the file will be rejected. By default, only the file size is checked. An automatic retry feature is included in the client software as of release 1.8.0. The command **get** will attempt up to 3 times to transfer the file successfully; otherwise an error will be reported.

FEI5 file types can be configured by the FEI5 administrator to automatically compute checksums on all added files. Users can also enable file checksums by setting the **computeChecksum** environment option.

```
MDMS_DEV:image1>> set computeChecksum on
MDMS_DEV:image1>> add jovianMoons.jpg
File added: jovianMoons.jpg Checksum:
"fa5b4314a6d71614c4ca65b224cbdfe4434d4992"
```

To see a file's checksum, enable the **veryVerbose** environment option and use any of the '**show**' commands. The checksum will appear before any file comment or archive note.

```
MDMS_DEV:image1>> set veryverbose on
MDMS_DEV:image1>> show
    1. 2003-04-24T10:40:04.876,         5422, jovianMoons.jpg
       [Contributor] ops
       [Created] 2003-04-23T17:05:14.356
       [Checksum] fa5b4314a6d71614c4ca65b224cbdfe4434d4992
       [Comment] "Image of Ganymede, Io, Europa and Callisto"
```

- **Resume transfer**

  FEI5 has the capability to allow clients to resume the transfer of a file in the event that communication was severed before file transfer completes. Setting the restart environment option enables resume transfer. Resume transfer requires file checksum data to detect file corruption; therefore the **computeChecksum** environment option must be enabled. The resume transfer capability benefits users who must transfer large files and don't have time to retransfer the entire file if communication is severed.

  User logs on to FEI5 and connects to image file type.

```
MDMS_DEV:image1>> set restart on
MDMS_DEV:image1>> set computeChecksum on
MDMS_DEV:image1>> get largefile.jpg

*** communication is severed before transfer completes ***
```

  User logs on to FEI5 again and re-connects to image file type.

```
MDMS_DEV:image1>> set restart on


MDMS_DEV:image1>> set computeChecksum on

MDMS_DEV:image1>> get largefile.jpg

*** file transfer resumed from the point where communication was severed ***
```

```
Got: largegile.jpg Checksum: "a0d271f0248d488c4362578725443b863e0be7e6"
Transfer was resumed for file largefile.jpg at byte 3789568.
```

Session logging

Users can create a log of their FEI5 session, which will capture all inputs and outputs to a file. A log file must be specified and opened before logging can occur. Once the log file is open, logging can be enabled or disabled at any time during the session. By default, the log file is opened in the directory where the fei5 application was started unless a directory path is provided in the log command.

Create a log file (implicitly enables logging because **set log on** is the default):

```
MDMS_DEV:image1>> logfile /tmp/log_file
```

Once the log file has been created, users can turn logging off or on with the following commands:

```
MDMS_DEV:image1>> set log off
MDMS_DEV:image1>> set log on
```

Setting logging off leaves the log file open, but FEI5 does not write to it. Creating a new log file closes the existing log file if one exists. In addition, exiting the session also closes the session log file.

Using the batch command and settings

The batch command reads a file containing a sequence of FEI5 commands and executes them. For example, the contents of file `addFiles.fei` looks like this:

```
# Change local directory and file type
cd /home/ops/image1
use image1
# Add three files
add Jupiter.jpg
add io.jpg
add jovianMoons.jpg
# Change the local directory and file type
cd /home/ops/image2
use image2
add *
# Open a log file, show all the files and then close
# the log file.  This gives us a record of all image2
# files
logfile /tmp/image2.log
# set log on is the default value so session logging will start
show
```

Lines that begin with the # character are comments. Often a user would like to test a set of batch commands before actually executing them against the server. There is a **test** configuration option to do just that

```
MDMS_DEV:>> set test on
MDMS_DEV:>> batch addFiles.fei
Executing batch file /home/ops/scripts/addFiles.fei
```

```
MDMS_DEV:>> # Change local directory and file type
MDMS_DEV:>> cd /home/ops/image1
Directory /home/ops/image1 not found
Batch execution aborted
```

After correcting the error via a simple text editor

```
MDMS_DEV:>> set test on
MDMS_DEV:>> batch addFiles.fei
Executing batch file /home/ops/scripts/addFiles.fei
…
Batch file test completed
```

To run the batch against the server

```
MDMS_DEV:>> set test off
MDMS_DEV:>> batch addFiles.fei
…
```

A batch file containing an error will exit the fei5 application. It is best to run in test mode and correct errors before running against the server.

- **Repeating batch files**

  A batch file can be scheduled for future execution(s) within the FEI5 session. The **repeatAt** keyword is used to schedule a batch file to be executed at the specified time. The **repeatEvery** keyword is used to schedule a batch file to be executed at the specified time interval.

```
MDMS_DEV:>> help batch
Command:
Description: Read a sequence of FEI commands from a file.
Usage: batch <filename> [repeatAt hh:mm AM|PM]  [repeatEvery hh:mm  [hh:mm AM|PM]]
Usage: utitlity
```

To schedule a batch file to be executed at 5:00 pm every day while the session is open

```
MDMS_DEV:>> batch mybatch.fei repeatAt 05:00 pm
```

To schedule a batch file to be started at 5:00 pm and repeat every 5 minutes afterwards while the session is open

```
MDMS_DEV:>> batch mybatch.fei repeatEvery 00:05 05:00 pm
```

To schedule a batch file to be executed every 5 minutes while the session is open

```
MDMS_DEV:>> batch mybatch.fei repeatEvery 00:05
```

To kill the batch command, the user should issue a <control-C>, but this will also kill the **fei5** session.

- **Automatic file pulling**

  By combining the FEI5 batch and restart capabilities and the **getAfter** command, we can construct a simple automatic file-pulling task that is similar to a subscriber. A file-pulling task is essentially a **getAfter** command within a batch file that is scheduled for a specified time interval. Here is a simple `puller.fei` batch

```
# Enable session restart
      set restart on
      # Connect to file type
```

```
use image1
# Redirect output directory
cd home/ops/output
# Query the server for new files
getAfter
```

When executing in a restart batch mode, the **getAfter** command does not require an input time value. The time value is being stored in a hidden directory in the user's local machine. Now to schedule this batch file to be executed every 1 minute:

```
MDMS_DEV:>> batch puller.fei repeatEvery 00:01
Batch file will be executed every 600000 (ms)
[2003-06-23 12:21:49.545] Scheduled batch execution start
Executing batch file /home/ops/scripts/puller.fei
MDMS_DEV:>> set restart on
MDMS_DEV:>> use image1
Using file type MDMS_DEV:image1
MDMS_DEV:image1>> cd /home/ops/output
Current directory set to /home/ops/output
MDMS_DEV:image11>> getAfter
Got: □ilenam.jpg
Got: io.jpg
Got: jovianMoon.jpg
[2003-06-23 12:21:49.777] Scheduled batch execution end

[2003-06-23 12:31:49.513] Scheduled batch execution start



Executing batch file /home/ops/scripts/puller.fei

MDMS_DEV:image1>> set restart on

MDMS_DEV:image1>> use image1
Using file type MDMS_DEV:image1
MDMS_DEV:image1>> cd /home/ops/output
Current directory set to /home/ops/output
MDMS_DEV:image1>> getAfter
No files found.
[2003-06-23 12:31:49.524] Scheduled batch execution end
…
```

To kill the batch command, the user should issue a <control-C>, but this will also kill the **fei5** session.

- **Executing a batch file from command line**

  So far we have been describing the **fei5** command as an interactive application. There are times when we need to execute a batch file directly from the command line without having to interactively login and issue the batch command. Here is an example of a simple batch file (`add-files.fei`) with login information.

```
# login first if a login cache file is not available
login ops mypassword
# Change local directory and file type
cd /home/ops/image1
use image1
# Add three files
add □ilenam.jpg
add io.jpg
add jovianMoons.jpg
# Change the local directory and file type
cd /home/ops/image2
```

```
use image2
add *
# Open a log file, show all the files and then close
# the log file.  This gives us a record of all image2
# files
log /tmp/image2.log
set log on
show
```

Use the **-b** command line option of **fei5** to execute a batch file from the command line.

```
% fei5 -b addfiles.fei

or to place the job in the background

% fei5 -b addfiles.fei &
```

Ending a session

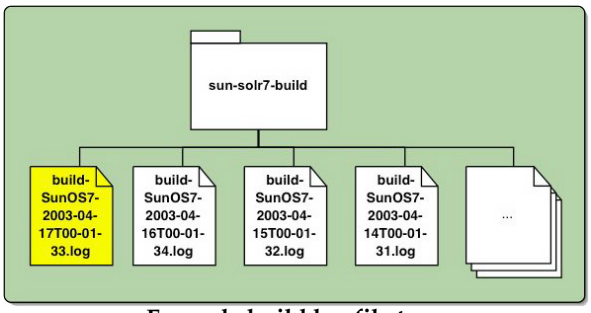
The user can end an FEI5 session by typing any of the following commands

```
MDMS_DEV:>> exit
MDMS_DEV:>> bye
MDMS_DEV:>> quit
```

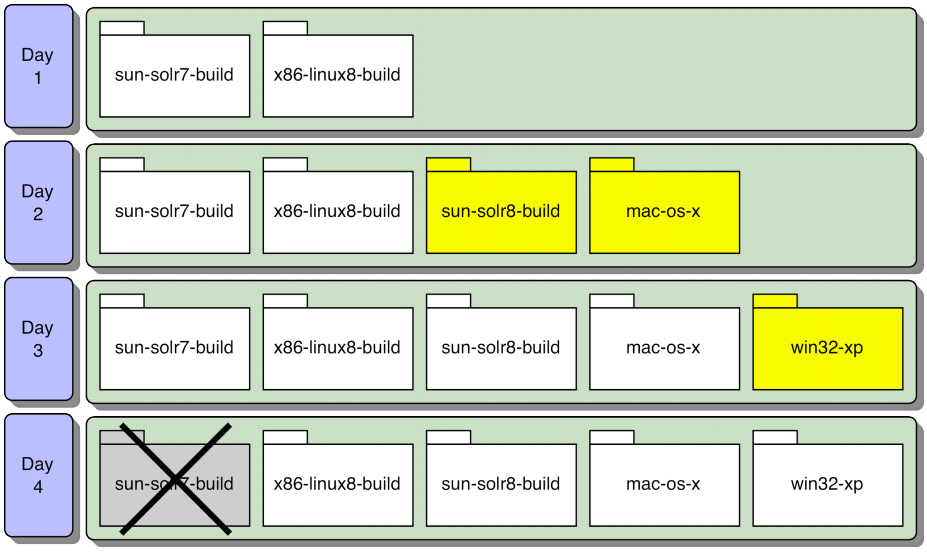
5. Working with Virtual File Types (VFT)

FEI5 offers a second method in organizing files. A virtual file type (VFT), as its name has indicated, is not a base file type but a file type that is derived from existing file(s) within existing file type(s). A virtual file type is a collection of references, or alias, to files in the FEI5 server repository. A virtual file type can be derived from files from various base file types. In addition, a virtual file type also keeps a history of its past references to enable query and acquisition of virtual file type data at a particular moment in time.

MDMS is using VFT to organize our nightly build logs for web presentation and query. In configuration and management, we are often only interested in the latest build results for all current supported platforms. We might look at past build logs when the latest build fails to see what has changed. In addition, the number of support platforms can also vary due to revisions to project requirements. With such dynamic environment, organizing files with base file types may solve the problem but with the overhead of domain file distribution, querying each latest supported build log file types with static FEI5 get file time queries.

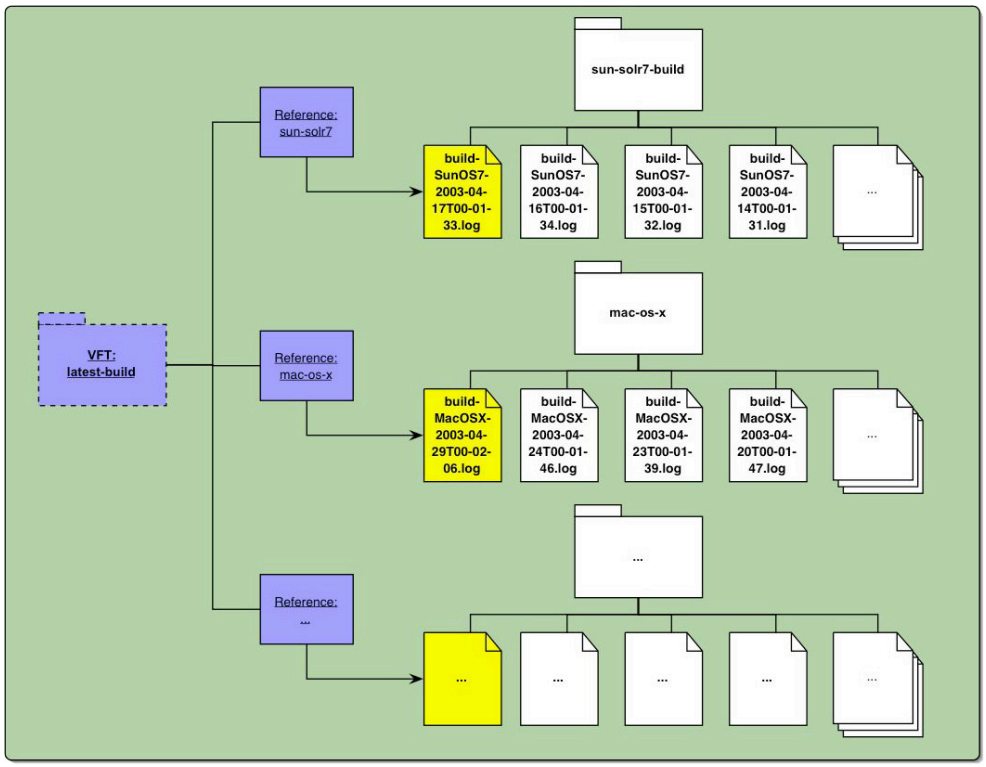


Example build log file type



A sample timeline for file type evolution.

VFT manages a collection of file references and enable users to access a collection of related files via a single file type name. The references it manages can be updated over time, but the updates are transparent to the users.



Using virtual file type to manage build logs.

Creating VFTs

Creating VFTs requires special user capability that is not enabled by default for FEI5 users.

```
MDMS_DEV:>> addVFT latest-build "The latest build logs."
```

Creating references

Once the VFT is defined, it is time to associate file references to it. The reference name is the nominal name by which the file will be known. For example, we can use the name **`sun-solr8.log`** to refer to the current Solaris 8 build log, and we can use the name **`x86-linux71.log`** to refer to the current RedHat Linux 7.1 build log, and so on. If we have decided to rename an existing reference, it can be done at the VFT reference management level that is transparent to the VFT users.

```
MDMS_DEV:>> addReference latest-build sun-solr8.log sun-solr8.link
MDMS_DEV:>> addReference latest-build x86-linux71.log x86-linux71.link
```

The commands above have just created two virtual reference names. The `*.link` files are created at the server-side as symbolic links to the physical files when the physical files are assigned. At this point, these references have not been associated to any physical files in the base file types, so their reference value is `NULL`.

Setting reference values

The command below associates a physical file to an existing reference.

```
MDMS_DEV:>> setReference latest-build sun-solr8.log sun-solr8 build-SunOS8-2003-
04-17T00-01-33.log
MDMS_DEV:>> setReference latest-build x86-linux71.log x86-linux71 build-
RedHat71-2003-04-17T00-02-06.log
```

The **setReference** command also creates physical symbolic links to the physical files to which references are associated.

Committing VFT updates

In order for a VFT to recognize the update references, the user must commit the changes. This is required to ensure consistency in VFT values.

```
MDMS_DEV:>> updateVFT latest-build
```

Acquiring files using VFT

Once files are bounded to references, the complete set can be gotten using the **getVFT** command.

```
MDMS_DEV:>> getVFT latest-build
got: sun-solr8.log =>
      sun-solr8/build-SunOS8-2003-04-17T00-01-33.log
got:x86-linux71.log =>
      x86-linux71/build-RedHat71-2003-04-17T00-02-06.log
…
```

While the **getVFT** command is being executed, the **updateVFT** command cannot be run on the specific VFT. This prevents users from getting an inconsistent set of referenced files.

The **getVFT** command creates one additional file in the user's local directory, the VFT manifest. This file's name is the name of the VFT followed by `.vft` extension. This file gets over-written each time the **getVFT** command is executed on the specific VFT. The file contains:

- The name of the VFT and any associated descriptive information.
- The date and time at which the **getVFT** command was executed.
- If the **getVFT** command was provided a historical time, that time is included as **vftHistoricalTime**, otherwise this line does not appear.
- The time the **updateVFT** command was executed to create this VFS.
- If a receipt was requested, the receipt number is included.
  - A list file names received. Each file description includes:
  - Reference name
  - Actual file type and file name referenced
  - File's checksum

Changing references

Users can update a reference to VFT to a different file. Use the **setReference** command to add a new reference to an existing VFT. Use **cancelReference** command to remove any out-of-date reference. The final step, of course, is to commit any changes to the VFT by using the **updateVFT**.

```
MDMS_DEV:win32-xp>> add build-Win32XP-2003-04-30T00-02-10.log
MDMS_DEV:win32-xp>> addReference latest-build win32-xp.log win32-xp.link
MDMS_DEV:win32-xp>> setReference latest-build win32-xp.log win32-xp
      build-Win32XP-2003-04-30T00-02-10.log
MDMS_DEV:win32-xp>> updateVFT latest-build "Added latest Win32 XP build"
```

- **Historical records**

  Users can query for all past VFT information by using the **showVFT** command with a specified time value. In addition, users can also acquire historical VFT using **getVFT** command with the similar interface for date/time query.

```
MDMS_DEV:>> showVFT latest-build 2003-05-01
MDMS_DEV:>> getVFT latest-build 2003-05-01
```

Appendix A – fei5 Tool Session Setting Parameters

PARAMETER	DESCRIPTION	DEFAULT
`abort`	During processing of batch file abort when encountered an error.	On
`autoDelete` or `autodelete`	Deletes the local copy of a file after adding or replacing it in the FEI5 data store.	Off
`computeChecksum` or `computechecksum`	Compute a checksum before and after files are transmitted. If the checksum values do not match, then the transaction will be aborted and issue and error message.	Off
`diff`	Enables replace of a file in the server if the files are different on a replace command or replace of a file in the user's local disk if the files are different on a get command.	Off
`echo`	Enables command output to terminal and log files.	On
`log`	When enabled, FEI5 commands and responses are logged to a file. Default is on, but if log file is not assigned, there is no logging.	On
`preserve`	This is a fei5admin parameter. When enabled, preserves date-time meta-data values during an intra-server group file move. Subscriptions running on the target file type will not be notified about moved file. If disabled, then the modification time of the moved file will be updated, alerting active subscription sessions of the file. Requires admin capabilities.	Off
`receipt`	When getting a file, a confirmation of receipt is logged in the data store. Upon completion of the transfer a receipt ID is returned. This identified the record kept within the data store. When this is enabled, the FEI5 client always computes a checksum on the transferred files contents and places it in the receipt record stored in the data store. This requires the user to have file registration capability.	Off
`replaceFile` or `replacefile`	By default FEI5 won't overwrite a file in current working directory during a `get` operation. To have existing files replaced during `get`, this parameter must be enabled.	Off

PARAMETER	DESCRIPTION	DEFAULT
`replicate`	Allows the user to output the obtained file in the same location as the stored server path of the file. Requires valid disk permissions to create directories.	Off
`restart`	When on, `getAfter` command will track date of last download for given file type. When use with `computeChecksum` on, it allows transfer to resume.	Off
`safeRead` or `saferead`	During get operations, the files are written to a shadow directory while being transferred. Upon completion, the files are moved to final destination.	Off
`test`	When executing a batch file, check commands syntax but do not execute.	Off
`timer`	When turned on each command is timed.	Off
`verbose`	Include files' size in bytes and date and time file was received along with the name.	off
`veryVerbose` or `veryverbose`	Along with verbose information show any comment, archive note or checksum associated with a file.	Off
`versionFile` or `versionfile`	Creates versioning of existing files by appending filename with version information.	Off

Appendix B – fei5 Tool Interactive Utility Commands

COMMAND	DESCRIPTION
`batch <file name> [{repeatAt hh:mm {am|pm}, repeatEvery hh:mm [hh:mm {am|pm}}]`	Reads a sequence of FEI5 commands from a file and execute them as though they were typed in. `repeatAt`: the batch file executes at the specified time of day. `repeatEvery`: the batch file executes repeatedly at the specified time interval until the session exits or is killed. The first batch executes immediately unless a start time is specified (time value with am/pm specified).
`bye`	Exits FEI5 client.
`cd [<local directory>]`	Change local working directory.
`changePassword or changepassword`	Change user password in the FEI5 server.
`dateformat`	Sets/shows the current date format. If no argument is provided, the current format is displayed. Otherwise, command expects either a named format (i.e. 'utc') or a date format string following SimpleDataFormat pattern. (http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html)
`exit`	Exits FEI5 client.
`help [{<command name>,<command type}>]`	Outputs all available commands. Help `<command name>` returns info on specific command.
`history <number>`	Displays history of recent commands and, when number supplied, executes command at that index of the history. If number is negative, then the most recent starting from the number is executed.
`logCmds or logcmds <file>`	Log commands to a file.
`logFile or logfile <file>`	Assign name of a log file.

COMMAND	DESCRIPTION
`login [<name>] [<password>]`	Enter user name and password, will prompt interactively if not in batch mode.
`ls`	List contents of local directory.
`pause`	Stops executing a batch script until the `<return>` key is pressed.
`pwd`	Show local working directory.
`quit`	Exits FEI5 client.
`set <parameter> {on,off}`	Sets session settings. Parameters: abort, `autoDelete`, `computeChecksum`, `log`, `receipt`, `replaceFile`, `restart`, `safeRead`, `test`, `timer`, `verbose`, `veryVerbose`, …
`showDomainFile or showdomainfile`	Echoes current domain file to screen.
`version`	Show version of FEI5 client.

Appendix C – Interactive File Type Commands

COMMAND	DESCRIPTION
`add <file name expression> [<"comment">]`	Add files matching file name expression to current file type.
`addAndRef|addandref <file name expression> <vft> <link directory>`	Add files matching file name expression to current file type. Each file's name is made unique, and a vftReference, using the source file name is added to the given VFT.
`archive <file name>`	Update archive information for the specified file.
`checksum <local file name>`	Computes checksum on a local file.
`comment <file name> <"comment">`	Update comment for the specified file.
`delete <file name expression>`	Delete a registered file.
`get <file name expression> [invoke "<system command>"]`	Get one or more files from current file type. If external system process invocation is specified, then it executes the system command for each file received. Use `$fileName` within the invoke string to refer to the fully qualified name of the file just received. Use `$fileType` within the invoke string to refer to the name of the current file type. Use `$fileNameNoPath` within the invoke string to refer to the name of the file just received without the directory path. Use `$filePath` within the invoke string to refer to the file directory path. Use `$serverGroup` within the invoke string to refer to the name of the server group. Use `$comment` within the invoke string to refer to file's associated comment.
`getAfter|getafter [<yyyy-MM-ddThh:mm:ss.SSS>] [invoke "<system command>"]`	Get files registered after date given. If no date is given and setting "restart" is on, use date in restart file. If external system process invocation is specified, then it executes the system command for each file received. Use `$fileName` within the invoke string to refer to the fully qualified name of the file just received. Use `$fileType` within the invoke string to refer to the name of the current file type. Use `$fileNameNoPath` within the invoke string to refer to the name of the file just received without the directory path. Use `$filePath` within the invoke string to refer to the file directory path. Use `$serverGroup` within the invoke string to refer to the name of the server group. Use `$comment` within the invoke string to refer to file's associated comment.

COMMAND	DESCRIPTION
`getBetween|getbetween <yyyy-MM-ddThh:mm:ss.SSS> and <yyyy-MM-ddThh:mm:ss.SSS> [invoke "<system command>"]`	Get files added to current file type during specified time period. If external system process invocation is specified, then it executes the system command for each file received. Use `$fileName` within the invoke string to refer to the fully qualified name of the file just received. Use `$fileType` within the invoke string to refer to the name of the current file type. Use `$fileNameNoPath` within the invoke string to refer to the name of the file just received without the directory path. Use `$filePath` within the invoke string to refer to the file directory path. Use `$serverGroup` within the invoke string to refer to the name of the server group. Use `$comment` within the invoke string to refer to file's associated comment.
`getLatest|getlatest [<file name expression>] [invoke "<system command>"]`	Get latest file for current file type. If external system process invocation is specified, then it executes the system command for each file received. Use `$fileName` within the invoke string to refer to the fully qualified name of the file just received. Use `$fileType` within the invoke string to refer to the name of the current file type. Use `$fileNameNoPath` within the invoke string to refer to the name of the file just received without the directory path. Use `$filePath` within the invoke string to refer to the file directory path. Use `$serverGroup` within the invoke string to refer to the name of the server group. Use `$comment` within the invoke string to refer to file's associated comment.
`lockFileType|lockfiletype [{group,owner}]`	Lock current file type to prevent modification (from add, replace, rename, delete). User's can still read and get files. The filetype can be locked at the group or owner level. Requires locktype capabilities.
`makeDomainFile|makedomainfile <file name>`	Make new domain file from database information.
`rename <original file name> <new file name>`	Rename a registered file.
`replace <file name> [<"comment">]`	Replace registered files.
`setDefaultGroup|setdefaultgroup <group name>`	Set the default server group for this session.

COMMAND	DESCRIPTION
`show <file name expression>` `[invoke "<system command>"]`	Show registered files in current file type. If external system process invocation is specified, then it executes the system command for each file received. Use `$fileName` within the invoke string to refer to the name of the file just received. Use `$fileType` within the invoke string to refer to the name of the current file type. Use `$fileNameNoPath` within the invoke string to refer to the name of the file. Use `$serverGroup` within the invoke string to refer to the name of the server group. Use `$comment` within the invoke string to refer to file's associated comment.

`showAfter|showafter`
`<yyyy-MM-ddThh:mm:ss.SSS>`
`[invoke "<system command>"]`
	Show files registered after given date.
	
	If external system process invocation is specified, then it executes the system command for each file received. Use `$fileName` within the invoke string to refer to the name of the file just received. Use `$fileType` within the invoke string to refer to the name of the current file type. Use `$fileNameNoPath` within the invoke string to refer to the name of the file. Use `$serverGroup` within the invoke string to refer to the name of the server group. Use `$comment` within the invoke string to refer to file's associated comment.

`showBetween|showbetween`
`<yyyy-MM-ddThh:mm:ss.SSS>`
`and <yyyy-MM-ddThh:mm:ss.SSS>`
`[invoke "<system command>"]`
	Show files added to current file type during specified time range.
	
	If external system process invocation is specified, then it executes the system command for each file received. Use `$fileName` within the invoke string to refer to the name of the file just received. Use `$fileType` within the invoke string to refer to the name of the current file type. Use `$fileNameNoPath` within the invoke string to refer to the name of the file. Use `$serverGroup` within the invoke string to refer to the name of the server group. Use `$comment` within the invoke string to refer to file's associated comment.

`showCapabilities|showcapabilities` `<file type>`	Show user's capabilities for given file type.

COMMAND	DESCRIPTION
`showLatest|showlatest [<file name expression>] [invoke "<system command>"]`	Show latest files added to current file type. If external system process invocation is specified, then it executes the system command for each file received. Use `$fileName` within the invoke string to refer to the name of the file just received. Use `$fileType` within the invoke string to refer to the name of the current file type. Use `$fileNameNoPath` within the invoke string to refer to the name of the file. Use `$serverGroup` within the invoke string to refer to the name of the server group. Use `$comment` within the invoke string to refer to file's associated comment.
`showTypes|showtypes`	Show available file types.
`unlockFileType|unlockfiletype {group,owner}`	Unlock the current file type for a given group or owner. Requires locktype capabilities.
`Use [<server group>:]<file type>`	Change current file type in session. The server group parameter is optional, `defaultGroup` is specified in domain file.

Appendix D – Interactive VFT Commands

COMMAND	DESCRIPTION
`addReference <vft name> <reference name> <link> [<"comment">]`	Add a reference to a VFT, with optional file system link.
`addVFT <name> [<"comment">]`	Create a new VFT.
`addVFTReader <virtual file type> <file system user>`	Allow file system user to read VFT.
`cancelReference <vft name> <reference name>`	Cancel a VFT reference change.
`delReference <vft name> <reference name>`	Delete reference file name from VFT.
`delVFT <vft name>`	Delete current VFT.
`delVFTReader <virtual file type> <file system user>`	Disallow file system user from reading VFT.
`getVFT <vft name> [<yyyy-MM-ddThh:mm:ss.SSS>]`	Get files referenced within VFT. Optionally restrict to specified date.
`setReference <vft name> <reference name> [<file type> <file name>]`	Reference a file in the FEI5 registry.
`showVFT [<vft name>] [<yyyy-MM-ddThh:mm:ss.SSS>]`	Show available VFTs. If VFT is specified, show details.
`showVFTReaders <virtual file type>`	Show list of file system users allowed to read VFT. If user specified, show details
`updateVFT <vft name> ["<comment>"]`	Update VFT by setting references to point to new files.

Appendix E – Troubleshooting

MESSAGE	DETAILS
`Access denied. (for add, replace, delete)`	The user does not have write access to the targeted file type.
`Access denied. (for vft)`	VFT requires special user capability. Please contact the FEI5 administrator on VFT access.
`Access to "null" denied.`	Attempting to access a filetype that does not exist in the server. Check the spelling of the filetype name.
`Attempt to initiate a push subscription in "<filetype>" denied.`	The user needs push capabilities to use this option. Contact the FEI5 administrator.
`Cannot add "<file name>":… addFile: File already exists.`	The file is already registered under the current file type. Use the **fei5eplace** command to replace any existing files in the server.
`Cannot add "<file name>":… registerFile: File already exists.`	The file is already registered under the current file type. Use the **replace** parameter to replace any existing records in the server database.
`Cannot add "<filename>": Duplicate entry addfile. 'is locked for write.`	Contact the FEI5 administrator if the problem persists. A record in the database exist with the currentLock set to write. The administrator will need to reset.
`Cannot add "<filename>": The space required to store file exceeds the reserved space for filetype (ftname).`	Contact the FEI5 administrator. The allocated disk space for the filetype has been exceeded.
`Cannot get file <filename>.`	The client is unable to copy the file to disk. Check that there aren't any hardware disk or network issues. Check the filename requested.
`Cannot get file… 'lockfile', File does not exist.`	The file you are trying to get does not exist in the FEI5 server. Check the requested filename.
`Cannot get file <filename>, file not found on server.`	The file was removed from the server prior to the user receiving it.
`Cannot get file… IO Exception: "permission denied"`	The user does not have the correct Unix permissions to write to the output location.

MESSAGE	DETAILS
Cannot get file… IO Exceptions: "sync failed"	You may have run out of disk space or you may have exceeded you disk quota.
Cannot read options file…	The options file used by the fei5notify and fie5subscribe command is not readable. Make sure the file exist and is readable by the user.
Can't get file… IO exception: Connection has been shutdown: javax.net.ssl.SSLException: bad record MAC.	There were problems (most likely network) with the file transfer. Retry the command. If this was during a get command, delete the partial file and restart the get or use the **replace** parameter.
Can't get file… IO exception: "Read timed out"	There were problems (most likely network) with the file transfer. Retry the command. If this was during a get command, delete the partial file and restart the get or use the **replace** parameter.
Can't get file… IO exception: "Unexpected EOF from network peer."	The user is unable to get the file due to network issues. Resubmit the command.
Can't make file reference: File exists Unexpected EOF from network peer.	The FEI5 server is unable to create the reference link, because the link already exists in the server file system.
Can't move file from shadow directory.	The client is unable to write to the .shadow directory located in the current working directory or as specified with the 'output' parameter. Use ls –al to check the directory privileges and fix.
Connection attempt failed: <host name>	The server is offline or the SSL certificate is corrupt or expired. Please contact FEI5 administrator on server status.
Connection attempt failed: java.security.cert.CertificateException: Couldn't find trusted certificate	The user has an invalid SSL certificate. Please make sure the certificate is installed under the $FEI5/ directory. Contact the FEI5 administrator.
Error occurred while attempting to restore restart cache from exiting file <filename>… Unable to restore from existing cache file. Or Unable to restore from existing backup cache file.	If you ran out of disk space you may have corrupted your restart file. You will be unable to restart from the point of error. Restart your subscription and use fei5get to retrieve any missing files. Suggest placing your restart files and your data on different disks.

MESSAGE	DETAILS
`Error occurred while loading encryptor.`	Windows install. Contact the FEI administrator. There is a parsing error in the code due to a space in the path.
`Error while writing file. <filename> (No such file or directory)`	The server is unable to write the file to disk. Will also get this if a file was registered and the local file does not exist. Notify the FEI5 administrator immediately.
`Error while writing file. Cannot save file in shadow directory for replace.`	The disk where FEI5 is writing to may have run out of disk space. Contact the FEI5 administrator immediately.
`Error while writing file <filename> (No space left on device)`	The disk where FEI5 is writing to may have run out of disk space. Contact the FEI5 administrator immediately.
`File deletion from filesystem failed.`	Attempt to delete the file from the underlying filesystem failed. If the file was registered, use the unregister command to remove from the filetype.
`File is not readable.`	The fei5register command cannot register the requested file because the server does not have the system privileges to read the file or the file is on a disk that the server cannot see.
`File: "<file name>" already exits. (for get…)`	The file already exists under the user's current working directory. Use '**set replaceFile on**' or '**set versionFile on**' in the interactive client if the user is intended to have the file replaced. Specify the '**replace**' or '**version**' option for command line client.
`File "<file name> does not exist.`	The file you are trying to add does not exist in the user's current directory or specified directory. Check the filename.
`File "<file name>" has an inconsistent size than what is recorded. File may have been compromised.`	Report the error to the FEI Administrator. The server file is corrupt or the server disk may have run out of space.
`File: "<file name>" not found on server disk.`	Make sure the file is registered in the file type using the show or fei5list command, and try again. If problem persists, contact the FEI5 administrator, it could indicate a server disk problem.

MESSAGE	DETAILS
`<filename> File skipped.`	The file exists in the local directory and the 'replace' parameter was not specified. Delete the file and try again or use the replace parameter.
`<filename> File transfer verification failed.`	Corruption in the file was detected when an fei5get with crc was issued and file transfer is resumed and a disconnection occurred during the resumed transfer. Delete the corrupted file and get the file again.
`File type not selected.`	The user is not connected to any file type at this point. Use the '**use**' command to connect to a file type.
`File type "<file type>" not found in domain.`	The file type is not listed in the FEI5 domain file. Use **showDomainFile** and **showTypes** for the list of file types in the domain file.
`getVFTInfo - can't find vft '<vft name>'`	The user has supplied an invalid VFT name. Use '**showVFT**' command to list all existing VFTs.
`Invalid login`	User has supplied invalid login information (user name and/or password) or your password has been recently changed. Type **login** if in the fei5 program or **fei5kinit** if using batch commands to input correct login information.
`Invalid input values. No files found`	The FEI5 server is unable to retrieve data from its repository or it cannot connect to the FEI5 database. Contact the FEI5 administrator.
`Invoke process "<invoke call>" failed.`	Invoke of a script from a subscription failed. Verify script runs properly as a standalone script with the same parameters as the failed attempt.
`IO exception while …`	The client has lost its communication with the FEI5 server. Please contact FEI5 administrator on server status.
`IO exception while adding file`	Make sure the user has privilege to read the file they are adding.
`IO exception while getting filenames matching "<file name>"`	The client is unable to connect to the FEI5 server. Contact the FEI5 administrator
`IO exception while showing files: Read time out.`	A windows client can produce this error if single quotes are being used instead of double quotes. If the client is extremely slow and you get this message, contact the FEI5 administrator. The JAVA virtual machine for the server may have exceeded its memory allocation.

MESSAGE	DETAILS
`IO exception: <filename> (No such file or directory)`	The FEI5 server is unable to read the requested file from the server disk file system. Contact the FEI5 administrator.
`'lock file', unable to lock file`	File is temporarily locked by another user session. If the problem persists, contact the FEI5 administrator.
`Lost connection to [server-group:filetype]. Attempting restart`	The client has lost connection to the server from a subscription usually due to network issues or the server is down. The client subscription will attempt to reconnect to the server and will continue the attempts until the reconnection is re-established.
`Missing user login information.`	No cache file found. Use the fei5kinit command to create.
`No files found.`	If you believe there should be files returned, check the file name or file name expression entered.
`Please acquire credentials with login utility.`	Use **fei5kinit** to set your username and password
`This operation is not supported by this A&A registry.`	This message will appear if you are trying to change an LDAP password. You should follow the instructions provided by the LDAP service on how to change the password. After the LDAP password is change the user should perform a fei5kinit command.
`Transfer with receipt request denied.`	The user does not have 'receipt' capabilities to perform the requested operation. Contact the FEI5 administrator.
`Unable to determine JRE version. Current JRE location /usr/java.`	The software is unable to find JAVA. Set the JAVA_HOME environment variable. Refer to the installation guide.
`Unable to find file type <server group>:<file type name> in domain`	Not a valid file type. Check the file type name used. Use **fei5filetypes** to list the filetypes recognized by the server.
`Unable to connect to file type <filetype> server(s).`	The server is down or the FEI5 server may have reached the max number of user connections or an invalid SSL certificate is being used or check that a firewall is not preventing the connection. A firewall exception may be needed; consult the Installation Guide. If the problem persists, contact the FEI5 administrator.

MESSAGE	DETAILS
`Unable to persist session restart info to <filename>`	Client is unable to write to the subscription restart file. Verify the disk and or file exist and that the user has privileges to write to the file. If this error is proceeded with a "Too many open files" message, stop and restart your subscriptions to clear resource usage on the host machine. If you recently changed your password and did not reissue a "fei5kinit", your subscription cannot reconnect.
`Unable to process file "<filename>". Java.ioException: File reading error.`	Contact the FEI5 administrator. The server may need to be restarted.
`Unable to restart session to <servergroup:filetype>. Next attempt in …`	Client subscription is unable to communicate with the server and will continue to attempt reconnection until re-established. The cause is usually due to a network issue or the server is down.
`updateComment, File does not exist.`	Check the spelling of the name of the file.
`Verification error (CRC) occurred for <filename> on attempt 1. Trying again…`	The CRC (checksum) computed for the file transfer did not match. Try retrieving the file again. Should the message persist, contact the FEI administrator.
`"" was unexpected at this time.`	A windows client will get this message if the CLASSPATH is defined and it ends with a ';'. Check the system environment variables and remove the ';'.

Appendix F – Tips and Tricks

Tip 1 – fei5accept

When working with a set of files on the same file type, the 'fei5accept' command can be more efficient than invoking 'fei5get using <filename>'. However, if the number of files in the fei5accept list is too large (dependent on the amount of memory used to start the server) a Java *OutOfMemoryError* might occur. (i.e. for a 1GB size of memory, a list with 162,020 files caused this error). Recommend a list less than 50,000.

Tip 2 – Subscription logging

For fei5notify and fei5subscribe subscriptions, use the logFileRolling option. This allows the user to delete older log files easily. Some editors will disconnect the log writing capability if you prune a fei5 log. With the logFileRolling option, the user on a UNIX system can delete files with older dates. A cronjob to do the deleting can be done:

```
* 0 * * * find /home/fei/subscriptionLogs/ -name "*log.200*" –mtime +14 –exec rm –f {} \; > /dev/null 2>&1
```

Tip 3 – Subscription $comment variable

For fei5notify and fei5subscribe subscriptions, when using the $comment variable, if the comment is a string of words, it will get passed into scripts as multiple parameters. The user should make sure they ingest the string properly.

Tip 4 – Sample scripts for showing and stopping subscriptions

Here is an example perl script for stopping subscriptions on a unix platform. This can also be used to view running subscriptions. The user may need to modify if using a different platform.

```
% cat fei5StopSub.pl
#!/usr/bin/perl
#
#  fei5StopSub.pl - Script to show all running fei5notify and fei5subscribe subscriptions
#       and then prompt the user as to whether the user wants to stop any subscriptions.  A
#       no answer will just return without stopping any processes.  At the prompt the user can
#       stop all subscriptions or be prompted and only stop specific subscriptions.  A fei5
#       subscription is a java process launched by a startup script.  The script will show you
#       both the startup process ids (PID) and the java process ids (PID).  The user will be
#       prompted to stop only the java process because this will stop the startup process.
#
use strict;
##  include Getopt module to make a call for getting command line options
use Getopt::Std;

my $usage = qq{

  fei5StopSub.pl [-d] <[servergroup:]filetype>
            -d - Debug mode (optional)

};

our ($opt_d);

## Get options
```

```
##
if (!getopts('d')) {
   die $usage;
}

if ( !defined $ARGV[0] ) {
   print "ERROR:>>>>> Missing parameter\n";
   die $usage;
}

print "Parm: $ARGV[0]\n" if ($opt_d);

# Find the process id of the FEI5 launcher scripts
#
my $user = $ENV{'USER'};
my (@subs, $numSubs);
findLaunchers();

#  If no processes, then print message saying so and stop
#
if ($numSubs < 1) {
   die "No fei5subscribe or fei5notify processess matching filetype\n";
}
# Get java proc ids for each FEI5 launcher process id
#
my ($pid, $pids, $proc, $javaProc, @javapids, $jpid);
print "-----------------------------------------------------\n";
print "---------- FEI5 subscriptions and java process Ids: ----------\n\n";
findJavaProcs();
print "=====================================================\n";
my ($inp);

# Allow the user to stop all processes, prompt for each process or do not stop processes
#
print ">>>> OK to stop the FEI5 subscriptions? (a: stop all; p: prompt for each java pid, n: no)> ";
chomp ($inp = <STDIN>);
print "inp: $inp|" if ($opt_d);
if ($inp =~ /[aA]/ ) {
  print ">>>> Stopping processes for PID(s): @javapids\n";
  unshift @javapids, 15;
  kill @javapids;
  print ">>>> This can take a while.... Rerun the script again to verify the jobs were stopped.\n";
  sleep $numSubs * 1;
} elsif ($inp =~ /[pP]/) {
   print ">>>> Prompting for PID(s)\n";
   print ">>>> The user might have to rerun the script to verify the jobs were stopped\n";
   foreach my $pid ( @javapids ) {
      next if ($pid == 15);
      print ">>>> OK to stop the java process PID(s) $pid (y/n): ";
```

```
      chomp (my $prompt=<STDIN>);
      if ($prompt =~ /[yY]/ ) {
         print ">>>> Stopping process for PID: $pid\n";
         kill 15, $pid;
         sleep 4
      } else {
         print ">>>> NOT Stopping process for PID: $pid continue to next pid\n";
      }
   }
} else {
   print ">>>> No FEI5 subscriptions stopped.\n";
}

 print "------------------------- CURRENT RUNNING SUBSCRIPTIONS ----------------------------\n";

#sleep 10;

findLaunchers();

if ($numSubs < 1) {
   die "No fei5subscribe or fei5notify processess matching filetype\n";
}
#my ($pid, $pids, $proc, $javaProc, @javapids, $jpid);
#print "-------------------------------------------------\n";
#print "--------------- FEI5 subscriptions and java process Ids: ------------\n\n";
findJavaProcs();

#
# sub findLaunchers - Find the FEI5 perl or Bourne shell launchers
#
sub findLaunchers() {

 my $PSCMD;

 if ($ENV{'OSTYPE'} =~ /solaris/) {
  $PSCMD = "/usr/ucb/ps axww | grep \'bin/[perl|sh].*fei5[sn].*$ARGV[0]\' | grep -v grep";
 } else {
  $PSCMD = "ps axww -U $user -opid,command | grep \'bin/[perl|sh].*fei5[sn].*$ARGV[0]\'";
 }


#  @subs = `pgrep -l -U $user -f 'bin/[perl|sh].*fei5[sn].*$ARGV[0]'`;

 @subs = `$PSCMD`;
 print "subs: @subs" if ($opt_d);

 $numSubs = @subs;
 print "# subs: $numSubs\n" if ($opt_d);
}
```

```
#
# findJavaProcs - find the java process ids associated with the launchers
#
sub findJavaProcs() {
 $pids = "";
 my ($PSCMD,$tmpProc,$junk);

 foreach my $sub ( @subs ) {
    $sub =~ s/^ *//;
    $sub =~ s/ .*[perl|sh].*fei5/ fei5/;
    print "$sub";
    ($pid,$proc) = split / /,$sub;
    if ($ENV{'OSTYPE'} =~ /solaris/) {
      $PSCMD = "/usr/ucb/ps alxww | grep java | grep $pid | grep -v grep | awk \'{print \$3}\'";
    } else {
      $PSCMD = "ps axww -U $user -opid,ppid,command | grep java | grep $pid | grep -v grep | awk
\'{print \$1}\'";
    }
#     print "PSCMD: $PSCMD\n";
    $pids .= $pid.",";
    #$javaProc = `pgrep -U $user -P $pid`;
    $javaProc = `$PSCMD`;
    chomp $javaProc;
    push @javapids, $javaProc;
    print "     JavaProcId: $javaProc\n";
 }
 chop $pids;
 print "pids: $pids|\n" if ($opt_d);
}
```

Tip 5 – Sample script for keeping subscriptions alive.

Here is an example c-shell script for keeping subscriptions alive on a linux/MAC/solaris platform. This script is useful when used in a cronjob. The user may need to modify the script if using a different platform. For Solaris, this script expects a BSD version of ps to be found in /usr/ucb/ps. The **ps** default on Solaris is the not the BSD version.

```
#!/bin/csh
#  script to (re)start FEI5 subscriptions
#
#   syntax:  keepAliveSubs.csh
#
#   Assumes a subscribe.options file exists in the directory from
#   which the subscriptions is started. The FEI5OptionsFileSetup.csh
#   file can assist you in creating options files for filetypes.
#   This script also assumes you are using the BSD version of ps.
#   The user should verify the script is working properly for their
#   platform before using operationally.
#
```

```
#   The user should change the following variables:
#     svrGrp - name of the FEI5 server group
#     DIR - location where the options files exist
#     FEIDIR - location of where the fei5 client resides
#     FEICMD - FEI5 command to use: either fei5subscribe or fei5notify
#              The default is set up for fei5subscribe.
#     filetypes - name of filetypes to create subscriptions for
#
#   This script can be run as a cronjob.
#------------------------------------------------------------------------

# Set the FEI5 server group variable, svrGrp, default is MSL
# Set the directory where the subscription options files reside.
# Expects the filetype to be the last subdirectory
#    i.e. /Users/jsmith/subscriptions/ops_sharad
#      set DIR = '/Users/jsmith/subscriptions'
# This is also the default location for transfered files unless
# the fei5 'output' option is used in the FEI5 Options file setup.

set svrGrp = "MSL"
set DIR = '/Users/jsmith/subscriptions'
set FEIDIR = '/Users/jsmith/fei5/latest'
set FEICMD = 'fei5subscribe'
# Filetypes to check
set filetypes = ( ops_ncam_edr ops_mcam_edr msl_rsvp )

if (${OSTYPE} =~ solaris ) then
  set PSCMD = "/usr/ucb/ps"
else
  set PSCMD = "ps"
endif

set curpath = `pwd`

## Setup FEI5, the $FEIDIR variable points to the location where the fei5 client resides
##
## If you are on an msop machine, you can use the OPGS delivered client.
#source /msop/opgs/software/current/scripts/fei5/fei5setup.csh

cd $FEIDIR
source use_FEI5.csh
cd $curpath

date

umask 002

# This is necessary so if ps returns output with a "?",
# grep will still work.  This is a csh issue.
```

```
set nonomatch

############  start FEI5 subscriptions  ##################
# Foreach filetype, first check if a process is running.
# If no, a subscription will be started
# Is yes, return

foreach ft ($filetypes[*])
### Yes, that is a space at the end of the grep command.....
 set proc = `$PSCMD auxww | grep  "${FEICMD} ${svrGrp}:${ft} " | grep $USER | grep -v grep`
 if ( ! ${#proc} ) then
   cd $DIR/${ft}
   echo "Starting ${ft} subscription"
   ${FEICMD} ${svrGrp}:${ft} push restart using subscribe.options  &
 else
   echo "Subscription for $ft already running."
 endif
end

cd $curpath

exit(0)
```

Tip 6 – Sample script for maintaining subscription options files and directories.

Here is an example c-shell script for creating directories and options files that are similar, and keep the log in a separate directory.

```
#!/bin/csh
#  Script to setup fei5notify or fei5subscribe options files.
#
#   syntax:  FEI5OptionsFileSetup.csh
#
#  This script will create a sub-directory of the filetype name
#  in the $DIR location.  Within the filetype sub-directory, the
#  script will create an options file of the name subscribe.options.
#  If msl_rsvp is one of the filetypes, will need to copy the
#  mslHandler.props file from the FEI5 scripts examples area to the
#  current directory.
#
#
#  Users should change the following variables:
#   DIR - location where to create options files and run subscriptions
#   FROM_EMAIL - value for mailMessageFrom option
#   TO_EMAIL - value for mailMessageTo option
#   SMTP - value for mailSMTPHost value
#   filetypes - names of filetypes to create options files for
#
#  If using fei5notify, comment out the 'replace' echo statement below.
```

```
#  If the user wants to invoke a command, update the invoke statement
#    below and uncomment.
#
#  Refer to the FEI5 User's Guide for more information on the
#  options for the fei5notify and fei5subscribe commands.

date

# Variable setup
#
set DIR = '/Users/jsmith/subscriptions'
set FROM_EMAIL = 'john.Smith@yahoo.com'
set TO_EMAIL = 'jsmith@gmail.com,jane.smith@yahoo.com'
set SMTP = 'smtp.yahoo.com'

set filetypes = ( ops_ncam_edr ops_mcam_edr msl_rsvp )

umask 002

############  Create options files for each filetype  ##################

foreach ft ($filetypes[*])
 if ( $ft =~ msl_rsvp ) then
   if ( ! -e mslHandler.props ) then
     echo "WARNING: File: mslHandler.props not found in current directory."
     echo "Copy file from the FEI5 scripts examples directory to current directory and rerun script"
     exit(1)
   endif
 endif
 set OPT_FILE = "$DIR/${ft}/subscribe.options"
 echo "---- Processing filetype ${ft} ----"
 echo "  Creating directory $DIR/${ft} if it doesn't exist."
 mkdir -p $DIR/${ft}
 echo "  Creating fei5 subscription options file for filetype $ft in:"
 echo "      $OPT_FILE"
 echo "mailMessageFrom $FROM_EMAIL" >&! ${OPT_FILE}
 echo "mailMessageTo $TO_EMAIL" >> ${OPT_FILE}
 echo "mailSMTPHost $SMTP" >>  ${OPT_FILE}
 echo "invokeExitOnError" >> ${OPT_FILE}
 echo "logFile $DIR/${ft}/${ft}.log" >>  ${OPT_FILE}
 echo "logFileRolling daily" >>  ${OPT_FILE}
## Use replace only with fei5subscibe. Comment out for fei5notify.
 echo "replace" >> ${OPT_FILE}
## Use invoke to post process files.  See examples below.
#  echo 'invoke "/Users/jsmith/procs/process.csh $fileNameNoPath" ' >> ${OPT_FILE}
#  echo 'invoke "echo $fileNameNoPath" ' >> ${OPT_FILE}
 if ( $ft =~ msl_rsvp ) then
   echo "  Adding options for replication to $OPT_FILE."
   echo "     Make sure the subscription owner has privs to write files to the diretory tree, /ods."
```

```
   echo "replicate" >> ${OPT_FILE}
## Use replicateRoot to redirect the writing of files from /ods to $replicateRoot/ods"
#    echo "replicateRoot $DIR" >> ${OPT_FILE}
   echo "filehandler" >> ${OPT_FILE}
## Enable automatic disk management to check if disk is becoming full and delete older files to make room.
## If the 'output' option is used, change the output for the cp command to the same location as the output
## option.
#    cp mslHandler.props $DIR/${ft}/komodo.filehandling.mslremotesubscription.props
 endif
end

exit(0)
```