**File Exchange Interface
Users Guide
Version 5.1.1, VICAR 31**

Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, CA 91109
U.S.A.

Thomas Huang
Science Instrument Services

Maintenance Release, December 2003

Table of Contents

Table of Figures

1. Introduction

The MIPL File Exchange Interface (FEI) service offers secure file transaction, store, transport, and management services. The latest distribution, FEI5 software code name Komodo, is a complete redesign from its predecessors by adopting the latest computing technologies and standards. This document explains the FEI client installation and configuration, and the command utilities to interact with the FEI servers.

1.1 FEI Architecture

FEI is a client-server application that is driven by the backend data store and file system. We use the term data store and not database here because the architecture enables the service to be decoupled from the specified backend 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 pluggable security selection, transparent location of servers and clients, automatic file archiving, and high-speed file transfer. The diagram also illustrated the portability of the FEI software that is being supported on all current popular operating systems.
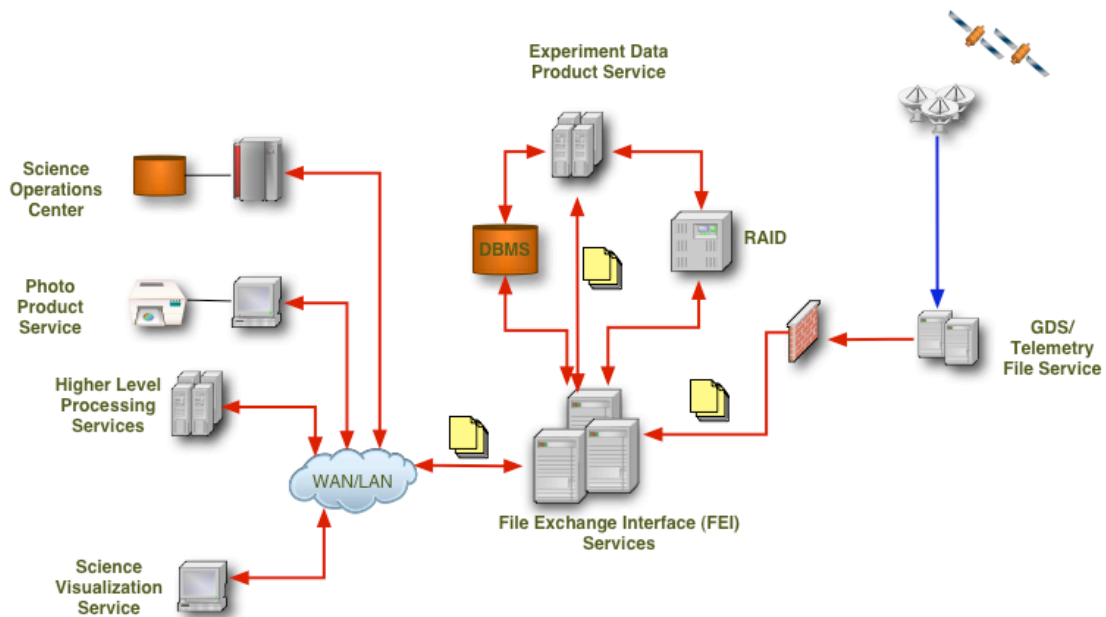
Figure 1 The Science Data Processing Pipeline with FEI

2. Application Programs

The FEI software distribution is bundled with both interactive and batch application programs.

Program	Interactive	Batch	Description
`fei5`	X	X	The general FEI user session application.
`fei5add`		X	Command to add/register file(s) to the FEI server
`fei5admin`	X		The FEI administrator session application.
`fei5comment`		X	Command to add comment to a file in the FEI server
`fei5delete`		X	Command to delete file(s) from the FEI server
`fei5encrypt`	X		Utility application to generate a one-way hash of any input message string.
`fei5get`		X	Command to get file(s) from the FEI server
`fei5kdestroy`		X	Command to destroy the persisted 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 FEI server file type
`fei5makeclean`		X	Command to cleanup a file type on the FEI server
`fei5rename`		X	Command to rename a file on the FEI server
`fei5replace`		X	Command to replace a file on the FEI server

3. Using the fei5 Command

After successfully installing the FEI client distribution, there are a few command line programs to interact with the software. This section discusses how to start and end an FEI section, how to get help, and how to access the current file types. The interactive client session of FEI 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 FEI client session. On the command line prompt, type the command **fei5** launch the application

```
% fei5
```

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

```
Domain file: /project/ops/servers/fei5/domain.fei
FEI Version 5.1.1 November, 2003
Komodo API version 2.7.2 November, 2003
Copyright 2002-2003 Jet Propulsion Laboratory, Caltech, NASA
```

The software begins with command prompt. The login procedure begins by user issuing the 'login' command. The software requests for user login information for the FEI server. This information will be cached and will be used when the user initiates connection to the FEI server.

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

Again, the user login information is cached for future connections to any FEI file types. At this point, the user is not yet connected to the FEI server; it is only an indication that the user has successfully started the FEI client application. To establish connection to a given file type, for example image1, the user should used the 'change type' command, use.

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

The **use** command initiates connection to the FEI server that hosts the file type **image1**. There are several criteria need to be verified by the server before authorize the connection request.
1.   The file type exists in the client domain file.
2.   The client distribution has the valid SSL certificate to communicate with the server.
3.   The user exists in the FEI data store.
4.   The cached user password matches the password registered with the FEI.
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. The user is now able to perform the common FEI operations, **add, delete, get, replace, rename**, etc.

Example 1 – to add a file:
```
image1>> add jupiter.jpg
File added: jupiter.jpg
image1>>
```

Example 2 – to get a file:
```
image1>> get jupiter.jpg
Got: jupiter.jpg
image1>>
```

3.1 Getting Help

Type '`help`', or '`?`', at the FEI prompt to list all available FEI 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).
```
>> help
 * Settings commands *
computechecksum  log              verbose
echo             autodelete       receipt
set              timer            test
restart          abort            replacefile
saferead         veryverbose
 * Utility commands *
exit             version          cd
help             pause            showdomainfile
quit             pwd              login
changepassword   bye              batch
ls               logfile          logcmds
 * Filetype commands *
showcapabilities  show             use
unlockfiletype    showsince        add
getsince          showlatest       comment
lockfiletype      showbetween      getbetween
rename            checksum         replace
archive          makedomainfile    get
addandref         delete           getlatest
showafter         showtypes        getafter
 * VFT commands *
addvft           showvftreaders   showvft
delreference     getvft           updatevft
delvftreader     delvft           addreference
cancelreference  setreference     addvftreader
```

More information about a specific command is available by specifying the command name with help.
```
>> help use
Command:      use
Description: Change file type
Usage:        use <file type>
Type:         filetype
```

3.2   Capabilities

A capability specifies the action(s) the user can perform on a given file type and the files within. A user can view their capabilities by typing the **'showCapabilities'** command.

```
>> showCapabilities
image1: add, get, replace
image2: add, delete, get, rename, replace
image3: get
image4: get
```

In this example, 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	Allows user to…
add	add, get, show, comment
archive	archive
delete	delete
get	get, show
locktype	lockFileType, unlockFileType
receipt	set receipt
rename	rename
replace	add, replace, get, show, comment
vft	virtual file type operations

3.3   Working with File Types

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

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

```
>> showTypes
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 FEI administrator will send a new domain file to users of the software.

3.3.1   Making Connection

Users can use the change type command to connect to a file type.

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

FEI displays the name of the file type behind the prompt to inform the user of the file type they are currently connected to. All file type commands (i.e. **add**, **get**, **delete**) will be within the context of this file type.

3.3.2   Adding, Replacing, Deleting and Renaming Files

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

```
image1>> add jupiter.jpg
File added: jupiter.jpg
```

FEI 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 **jupiter.jpg** again, the command will fail. FEI will not add a file that is already exists in the system, but we can replace it using the command:

```
image1>> replace jupiter.jpg
File replaced: jupiter.jpg
```

The **replace** command can also add a file if it is not already exist in the FEI 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. **jupiter.jpg**) from a file type is:

```
image1>> delete jupiter.jpg
File deleted.
```

The command to rename a file is:

```
image1>> rename jupiter.jpg io.jpg
File renamed.
```

3.3.3   Using Wildcards with File Names

FEI 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 FEI file type, issue the following command:

```
image1>> add *
```

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

```
image1>> show jupiter*
```

3.3.4   Showing Lists of Files

FEI provides commands to show files in a file type. These operations are carried out by two collections of commands. The commands that show what is in a file type begin with '**show**'. FEI has commands to show a single file, files using wildcard, and files that fall into sometime frame reference.

```
>> help show
Command:      show
Description: Show registered files in current file type
Usage:       show <file name expression>
Type:        filetype
```

```
image1>> show
jovianMoons.jpg
jovianRings.jpg
jupiter.jpg
io.jpg
```

3.3.5   Getting Files

FEI provides commands to get files from a file type. These operations are carried out by two collections of commands. The commands that get what is in a file type begin with '**get**'. FEI has commands to get a single file, files using wildcard, and files that fall into sometime frame reference.

```
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

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

3.3.5.1   Process Invocation

FEI user may perform additional processing to the file it has just acquired from FEI. A special keyword '**invoke**' is built into all FEI **get** operations to allow user to invoke external process. There are two predefined parameters to the invoke string to serve as variable to the invoke command. Variable **$filename** represents the name of the file just acquired. Variable **$fileType** represents the file type name.

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

3.3.6   Date/Time Specific Commands

FEI supports commands for adding and showing files based on a specific time or date inputs. The date and time used by FEI is the time at which the file entered the FEI file system.

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

The date input can be specified without a time, which implies midnight for that date.

3.3.6.1   Date Formats

FEI accepts date input in the CCSDS ASCII calendar segmented time code format:
YYYY-MM-DDThh:mm:ss.ddd

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
mm	Minute in two-character subfield with values 00-59
ss	Second in two-character subfield with values 00-59
ddd[1]	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 **'.'** are used as specified subfield separators. All subfields must include leading zeros if the input date does not use all available characters. Times can be truncated, but truncated parts are given a value of zero. That is, a date value of **2003-09-14** gets translated into **2003-09-14T00:00:00.000**.

Examples of date/time input
```
image1>> showAfter 2003-09-14T14:23:00.384
image1>> showAfter 2003-09-14T14:23
image1>> showAfter 2003-09-14
```

3.3.7   The Latest File

Sometimes the user is only interested in the latest files
```
image1>> showLatest
io.jpg
image1>> getLatest
Got file: io.jpg
```

File name expression can also be used when querying for the latest files
```
image1>> showLatest jupiter*
jupiter.jpg
image1>> getLatest jupiter*
Got file: jupiter.jpg
```

3.3.8   FEI Environment Options

The FEI client supports user options that change the expected output of FEI commands. FEI environment options use the following syntax:

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

3.3.8.1   Verbosity Modes

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

---

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

- 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:
```
image1>> set verbose on
image1>> set veryVerbose on
```

Users can disable verbosity by issuing the following command (setting **verbose off** will set **verVerbose off** as well):
```
image1>> set verbose off
```

3.3.8.2   Insuring File Integrity

FEI uses file checksums to ensure a files integrity. Checksums are not enabled by default in FEI due to overhead required to compute the checksum. If enabled, a checksum is computed for a file as it is being transmitted. The receiver also calculates the checksum as the data is received. The checksum is saved and transmitted to the receiver along with the files meta-data. After transmission, 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. The automatic retry feature is being considered in future releases of the client software.

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

```
image1>> set computeChecksum on
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.

```
image1>> set veryverbose on
image1>> show
jovianMoons.jpg      5422bytes        2003-04-24T10:40:04.876
Checksum: "fa5b4314a6d71614c4ca65b224cbdfe4434d4992"
Comment: "Image of Ganymede, Io, Europa and Callisto"
```

3.3.8.3   Resume Transfer

FEI has the capability to allow clients to resume the acquisition 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 FEI and connects to image file type.
```
image1>> set restart on
image1>> set computeChecksum on
```

```
image1>> get largefile.jpg

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

User logs on to FEI again and connects to image file type.
image1>> set restart on
image1>> set computeChecksum on
image1>> get largefile.jpg

*** file transfer resumed from the point where communication was
severed ***
Transfer was resumed for file largefile.jpg at byte 3789568.
```

3.3.8.4   Session Logging

Users can create a log of their FEI 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.

Create a log file (implicitly enables logging):
```
        image1>> log /tmp/log_fille
```

Once the log file has been created, users can turn logging on or off with the following commands:
```
        image1>> set log off
        image1>> set log on
```

Setting logging off leaves the log file open, but FEI 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.

3.4   *Using the Batch Command and Settings*

The batch command reads a file containing a sequence of FEI command 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
log /tmp/image2.log
set log on
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

```
>> set test on
>> batch addFile.fei
Executing batch file /home/ops/scripts/addFile.fei
>> # Change local directory and file type
>> cd /home/ops/image1
Directory /home/ops/image1 not found
Batch execution aborted
```

After correcting the error via a simple text editor
```
>> set test on
>> batch addFile.fei
Executing batch file /home/ops/scripts/addFile.fei
…
Batch file test completed
```

To run the batch against the server
```
>> set test off
>> batch addFile.fei
…
```

3.4.1 Repeating Batch Files

A batch file can be scheduled for future execution(s) within the FEI 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.

To schedule a batch file to be executed a 5:00pm
```
>> batch mybatch.fei repeatAt 05:00 pm
```

To schedule a batch file to be executed every 5:00 pm
```
>> batch mybatch.fei repeatEvery 05:00 pm
```

3.4.2 Automatic File Pulling

By combining the FEI **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:

```
>> batch puller.fei repeatEvery 00:10
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
>> set restart on
>> use image1
Using file type : image1
image1>> cd /home/ops/output
Current directory set to /home/ops/output
image11>> getAfter
Got: jupiter.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
image1>> set restart on
image1>> use image1
Using file type : image1
image1>> cd /home/ops/output
Current directory set to /home/ops/output
image1>> getAfter
No files found.
[2003-06-23 12:31:49.524] Scheduled batch execution end
…
```

3.4.3   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 (**addfiles.fei**) with login information.

```
# login first
login ops mypassword
# 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
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
```

3.5   *Virtual File Type (VFT)*

FEI 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 FEI 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 FEI get file time queries.
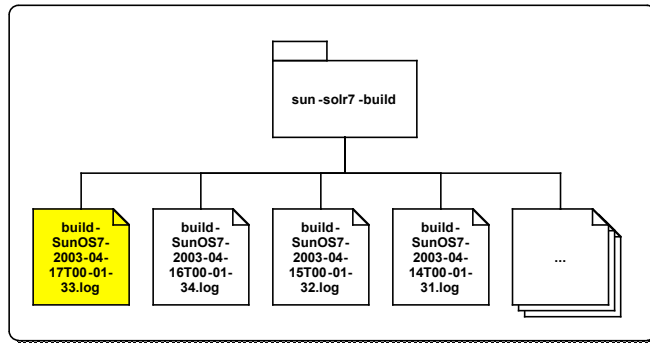
Figure 2 Example build log file type
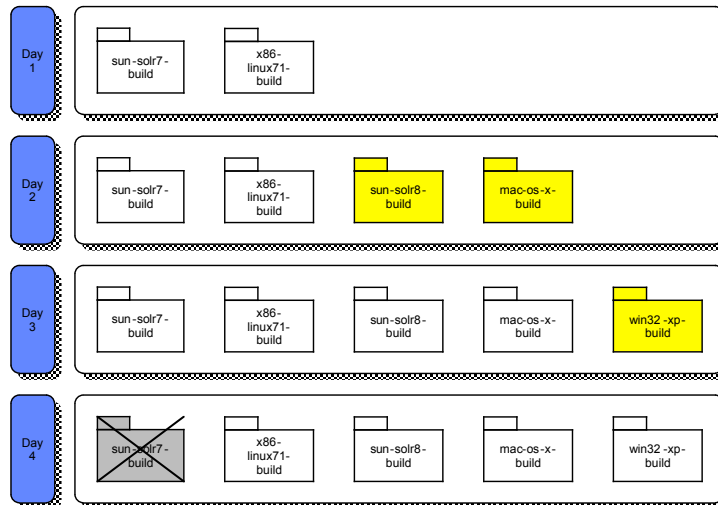
Figure 3 A sample timeline for file type evolution.

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.

Figure 4 Using virtual file type to manage build logs.

3.5.1   Creating VFTs

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

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

3.5.2   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 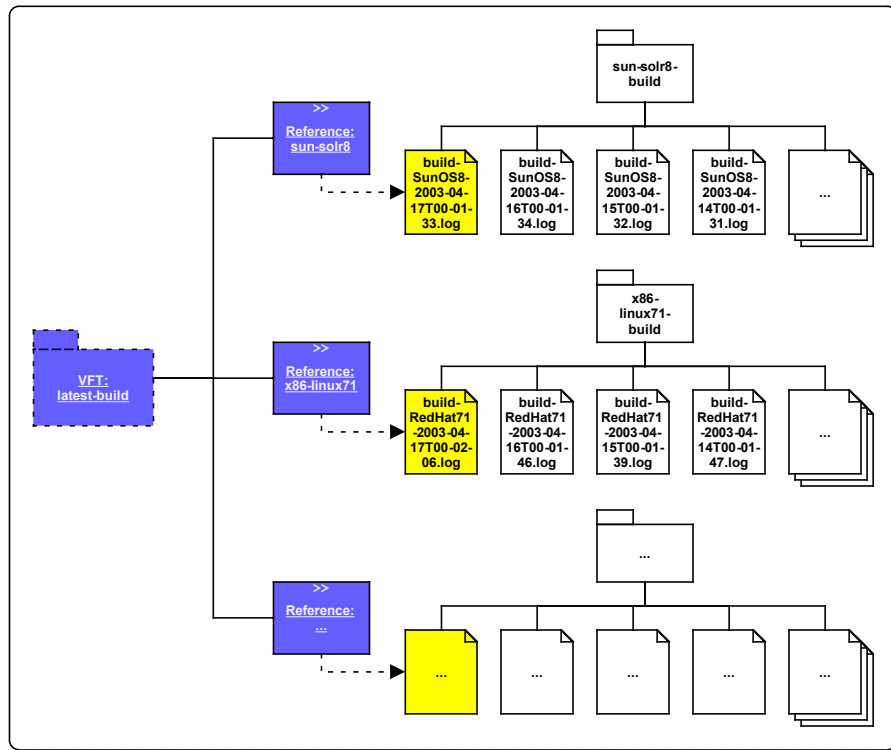the current RedHat Linux 7.1 build log, and so on. If we have decided to rename an existing reference, it can be done at the VFT reference management level that is transparent to the VFT users.

```
>> addReference latest-build sun-solr8.log sun-solr8.link
>> 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`.

3.5.3   Setting Reference Values

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

```
>> setReference latest-build sun-solr8.log sun-solr8
      build-SunOS8-2003-04-17T00-01-33.log
>> 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 that the references are associated to.

3.5.4   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.

```
>> updateVFT latest-build
```

3.5.5   Acquiring Files Using VFT

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

```
>> 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 VFS 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 (if requested)

3.5.6 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**.

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

3.5.7 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.

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

3.6   Ending an FEI Session

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

```
>> exit
>> bye
>> quit
```

4. Using the fei5admin Command

This release of FEI comes with an interactive service administration application to enabled administrators to perform common administrative operations without directly modifying the service registry. This application allows the administrator to connect to multiple servers at the same time and monitor user connection activities. Like the fei5 application, this interactive administration tool is equipped with a well-documented user help facility to provide online help information to each of the interactive commands.

```
>> help
 * Settings commands *
computechecksum  log              verbose
echo             autodelete       receipt
set              timer            test
restart          abort            replacefile
saferead         veryverbose
 * Utility commands *
exit             version          cd
help             pause            showdomainfile
quit             pwd              login
changepassword   bye              batch
ls               logfile          logcmds
 * Admin commands *
delrole              deluser              connections
showroles            addfiletype          showfiletypes
addfiletypetorole    hotboot              showmemory
showservers          deluserfromrole      fsync
delfiletypefromrole  addusertorole        showconnections
delfiletype          shutdown             showusers
showrolesforuser     addrole              showrolesforfiletype
adduser              showserverparameters connect
focus                dsync
```

5. Using the fei5encrypt Command

This utility command is to help user to generate a one-way hash on any input text message. This is the utility to use to generate encrypted message strings such as passwords. This should be the utility for creating the bootstrap server when creating the initial administrator login password.

```
% fei5encrypt
Message >>
4a8a9fc31dc15a4b87bb145b05db3ae0bf2333e4
```

6. More Command Line Utilities

Long time FEI users have gotten used to the simple command line utilities for interacting with FEI file types, that the new pro-interactive approach of the new client might be less flexible. For this reason, we have introduced a set of command line utilities to mimic the legacy FEI command line programs to simplify legacy application migrations.

6.1 User Login Utilities

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

6.1.1 fei5Kinit

```
fei5Kinit [<user name>] [help]
```

This utility is used to create a user login cache file that will be accessed by other general FEI batch utilities. The following example creates a login cache file for the user ops.

```
% fei5Kinit ops
Password>> ****
```

Here is another method

```
% fei5Kinit
User name>> ops
Password>> *****
```

6.1.2 fei5Klist

```
fei5Klist
```

This utility allows the user to view information on their login cache file.

```
% fei5Klist
Copyright 2002-2003 Jet Propulsion Laboratory, Caltech, NASA
FEI Version 5.1.1 November, 2003
Komodo API version 2.7.2 November, 2003

Credential cache file: /home/ops/.komodo/.komodologin
File modified on: Tue Jan 06 17:02:30 PST 2004
Default principal: ops
```

6.1.3 fei5Kdestroy

```
fei5Kdestroy
```

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

```
% fei5Kdestroy

% fei5Klist
Login Error! Please acquire credentials with login utility.
```

6.2 General Utilities

6.2.1 fei5Add

```
Usage:  fei5add <file type> <file name expression>...
    {[before|after <datetime>] | [between <datetime1> and <datetime2>]
    [comment "<comment text>"] [crc] [help]}

Usage:  fei5add using <option file>
Option File Format (per line):
    <file type> <file name>...
    {[before|after <datetime>] | [between <datetime1> and <datetime2>]
    [comment "<comment text>"] [crc]}
```

Keywords
- **before**: add file with creation times before the specified time value. Default: none.
- **after**: add files with creation times after the specified time value. Default: none.
- **between**… **and**…: add files with creation times between the specified time range. Default: none.
- **comment**: add description to the file. Default: none.
- **crc**: calculate the CRC value for each file. Default: none.
- **using**: keyword to supply a list of commands in an options file. Default: none.

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

A file is not added if it already exists in FEI. Use fei5Replace to do that.

Example 1:
```
% 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 2:
```
% cat addlist.txt
image1 /home/ops/data/file0
image1 file1 file2  # list of file names
image1 file* between 2003-255 and 2003-256
```

```
% fei5Add using addlist.txt
```

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

6.2.2 fei5Delete

```
Usage:  fei5Delete <file type> "<file name expression>" [help]

Usage:  fei5Delete using <option file>
Option File Format (per line):
        <file type> <file name>
```

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

fei5Delete takes the file type and file name or expression uses the information to return a list of files.

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

% fei5Delete using delList.txt
```

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

```
% fei5Delete image1 "file*"
"file0" has been deleted from file type "image1".
"file7" has been deleted from file type "image1".
"file8" has been deleted from file type "image1".
%
```


6.2.3 fei5Get

```
Usage:  fei5Get <file type> "<file name expression>"
        {[output <path>] [crc] [before|after <datetime>] |
        [between <datetime1> and <datetime2>] [replace] [help]}

Usage:  fei5Get using <option file>
Option File Format (per line):
        <file type> "<file name expression>"
        {[output <path>] [crc] [before|after <datetime>] |
        [between <datetime1> and <datetime2>] [replace]}
```

Keywords
- **output**: output path. Default: current directory.
- **before**: get files with FEI modification dates before the specified time value. Default: none.

- **after:** get files with FEI modification dates after the specified time value. Default: none.
- **between**… **and**…: get files with FEI modification dates between the specified time range. Default: none.
- **crc**: calculate the CRC value for each file. Default: none.
- **replace**: replace existing files within the specified output directory. Default: none.
- **using**: keyword to supply a list of commands in an options file. Default: none.

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

Note: A file glob is evaluated by FEI, so it must be surrounded by single/double quotes to protect it from shell evaluation. Get will not replace a file that already exists in your local directory unless you supply the keyword "replace". "Replace" replaces the current local file.

Example 1:
```
% fei5Get image1 file1
% fei5Get image1 file1 replace
% fei5Get image1 'file*'
% fei5Get image1 '*' after 2003-10-24
% fei5Get image1 'file*' version after 2003-298T14:20
% fei5Get image1 'file*' after 2003-298T14:20 replace
% fei5Get image1 'file*' between 2003-298T14:20 and 2003-298T14:50
```

Example 2:
```
% cat getlist.txt
image1 file0 output /home/ops/data        # directing output
image1 file1 replace                      # use replacng
image1 file* output /home/ops/data replace # use wild card

% fei5Get using getlist.txt
Got: file0
Got: file1
Got: "file0"
Got: "file1"
Got: "file2"
Got: "file3"
Got: "file4"
Got: "file5"
Got: "file6"
Got: "file7"
Got: "file8"
Got: "file9"
Got: "file10"
```

6.2.4 fei5List

```
Usage:  fei5List <file type> ["<file name expression>"]
        {[long | verylong] [before|after <datetime>] |
        [between <datetime1> and <datetime2>] [help]}
```

Keywords
- **before**: get files with FEI modification dates before the specified time value. Default: none.
- **after**: get files with FEI modification dates after the specified time value. Default: none.

- **between**… **and**…: get files with FEI modification dates between the specified `time` range. Default: none.
- **long**: lists output with FEI modification date and file size in bytes. Default: none.
- **verylong**: lists output with FEI modification date, file size in bytes, and any associated comments and archive notes. Default: none.

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

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

```
% fei5List image1
file0
file1
file2
file3
file4

% fei5List image1 'file*'
file0
file1
file2
file3
file4

% fei5List image1 "file*" long

    1.   2003-298T14:21:27.256, 5, file0
    2.   2003-298T14:21:27.426, 5, file1
    3.   2003-298T14:21:27.550, 5, file2
    4.   2003-298T14:21:27.686, 5, file3
    5.   2003-298T14:21:27.816, 5, file4
    6.   2003-298T14:55:06.753, 151292, file5

% fei5List image1 "file*" after 2003-298T14:30 long

    1.   2003-298T14:55:06.753, 151292, file5
```

6.2.5 fei5MakeClean

```
Usage:  fei5MakeClean <file type> ["<file name expression>"] [help]
```

This command is similar to fei5Delete with the default behavior for removing all files within the specified file type.

6.2.6 fei5Rename

```
Usage:  fei5Rename <file type> <old file name> <new file name> [help]

Usage:  fei5Rename using <option file>
Option File Format (per line):
```

```
          <file type> <old file name> <new file name>
```

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

Example 1:
```
% fei5Rename image1 file0 fido_0
```

Example 2:
```
% cat rename.txt
image1 file1 fido_1

% fei5Rename using rename.txt

% fei5List image1
fido_0
fido_1
```

6.2.7 fei5Replace

```
Usage:  fei5Replace <file type> <file name expression>...
    {[before|after <datetime>] | [between <datetime1> and <datetime2>]
    [comment "<comment text>"] [crc] [help]}


Usage:  fei5Replace using <option file>
Option File Format (per line):
    <file type> <file name>...
    {[before|after <datetime>] | [between <datetime1> and <datetime2>]
    [comment "<comment text>"] [crc]}
```

Keywords
- **before**: get files with FEI modification dates before the specified time value. Default: none.
- **after**: get files with FEI modification dates after the specified time value. Default: none.
- **between**... **and**...: get files with FEI modification dates between the specified time range. Default: none.
- **crc**: calculate the CRC value for each file. Default: none.
- **comment**: any associated comment for the file. Default: none.
- **using**: to supply a list of commands in an options file. Default: none.

This command works just like **fei5Add** but a file is replaced in FEI if it already exists. Here are some examples showing command line keywords:

Example:
```
% fei5Replace image1 file0
File replaced: file0

% fei5Replace image1 file0 file1
File replaced: file0
File replaced: file1
```

Appendix A – fei5 Interactive Command Reference

Setting Commands		
Command	**Description**	**Default**
abort	During processing of batch file abort when encountered an error.	off
autoDelete	Deletes the local copy of a file after adding or replacing it in the FEI data store.	off
computeChecksum	Compute a checksum before and after files are transmitted. If not the checksum values do not match, then the transaction will be aborted and issue and error message.	off
log	When on, FEI commands and responses are logged to a file. Default is on, but if log file is not assigned, there is no logging.	on
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 FEI 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	By default it won't overwrite a file in current working directory.	off
restart	When on, getAfter command will track date of last download for given file type.	off
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

set <parameter> {on,off}	Sets session settings. Parameters: abort, `autoDelete`, `computeChecksum`, `log`, `receipt`, `replaceFile`, `restart`, `safeRead`, `test`, `timer`, `verbose`, `veryVerbose`, ...	N/A
`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`	Along with verbose information show any comment, archive note or checksum associated with a file.	off

Utility Commands		
Command	**Description**	**Default**

batch <file name>
  [{repeatAt hh:mm {am,pm},
    repeatEvery hh:mm [hh:mm {am,pm}}]
	Reads a sequence of FEI commands from a file and execute them as though they were typed in.
	
	repeaseAt: the batch file executes at the specified time of day.
	
	repeaseEvery: 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).
		N/A

bye	Exits FEI client.	N/A
cd [<local directory>]	Change local working directory.	N/A
changePassword	Change user password in the FEI server.	N/A
exit	Exits FEI client.	N/A
help [<command name>]	Outputs all available commands. Help <command name> returns info on specific command.	N/A
logcmds <fle>	Log commands to a file	N/A
logFile <file>	Assign name of a log file.	N/A
login [<name>] [<password>]	Enter user name and password, will prompt interactively if not in batch mode	N/A
ls	List contents of local directory.	N/A
pause	Stops executing a batch script until the <return> key is pressed.	N/A
pwd	Show local working directory.	N/A
quit	Show local working directory.	N/A
showDomainFile	Echoes current domain file to screen.	N/A
version	Show version of FEI client.	N/A

File Type Commands		
Command	**Description**	**Default**
`add <file name expression>` `   [<"comment">]`	Add files matching file name expression to current file type.	N/A
`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.	N/A
`archive <file name>`	Update archive information for the specified file.	N/A
`checksum <local file name>`	Computes checksum on a local file.	N/A
`comment <file name> <"comment">`	Update comment for the specified file.	N/A
`use <file type>`	Use a file type on the server.	N/A
`delete <file name>`	Delete a registered file.	N/A
`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 `$fileName` within the invoke string to refer to the name of the file just received. Use `$fileType` within the invoke string to refer to the name of the current file type.	N/A
`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 `$fileName` within the invoke string to refer to the name of the file just received. Use `$fileType` within the invoke string to refer to the name of the current file type.	N/A

`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.	N/A

	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.	
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 name of the file just received. Use $fileType within the invoke string to refer to the name of the current file type.	N/A
lockFileType [{group,owner}]	Lock current file type to prevent modification.	N/A
show <file name expression>	Show registered files in current file type.	N/A
showAfter <yyyy-MM-ddThh:mm:ss.SSS>	Show files registered after given date.	N/A
showCapabilities <file type>	Show user's capabilities for given file type	N/A
showBetween <yyyy-MM-ddThh:mm:ss.SSS> and <yyyy-MM-ddThh:mm:ss.SSS>	Show files added to current file type during specified time range.	N/A
showLatest [<file name expression>]	Show latest files added to current file type.	N/A
showTypes	Show available file types.	N/A
unlockFileType {group,owner}	Unlock current file type for given group or owner.	N/A

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

Appendix B - Troubleshooting

Message	Details
Invalid login	User has supplied invalid login information (user name and/or password). Type **login** again to input correct login information.
File type not selected.	The user is not connected to any file type at this point. Use '**ct**' command to connect to a file type.
File type "<file type>" not found in domain.	The file type is not listed in the FEI domain file. Use **showDomainFile** and **showTypes** for the list of file types in the domain file.
Cannot add "<file name>": addFile: File already exists.	The file is already registered under current file type. Use **replace** command to replace any existing files in the server.
Access denied. (for add, replace, delete)	The user does not have write access to the targeted file type.
File: "<file name>" already exits. (for get…)	The file already exists under the user's current working directory. Use '**set replaceFile on**' if the user is intended to have the file replaced.
Connection attempt failed: <host name>	The server is offline. Please contact FEI administrator on server status.
IO exception while …	The client has lost its communication with the FEI server. Please contact FEI 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 FEI/config directory. Contact FEI administrator for SSL certificate update.
Access denied. (for vft)	VFT requires special user capability. Please contact FEI administrator on VFT access.
getVFTInfo – can't find vft '<vft name>'	The user has supplied an invalid VFT name. Use '**showVFT**' command to list all existing VFTs.
Can't make file reference: … : File exists Unexpected EOF from network peer.	The FEI server is unable to create the reference link, because the link already exists in the server file system.