The 'MFxx' Command Extensions for FTP
draft-somers-ftp-mfxx-00

Status of this Memo

   This document is an Internet-Draft and is in full conformance with
   all provisions of Section 10 of RFC2026.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF), its areas, and its working groups. Note that
   other groups may also distribute working documents as Internet-
   Drafts.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   The list of current Internet-Drafts can be accessed at http://
   www.ietf.org/ietf/1id-abstracts.txt.

   The list of Internet-Draft Shadow Directories can be accessed at
   http://www.ietf.org/shadow.html.

   This Internet-Draft will expire on January 17, 2003.

Copyright Notice

Abstract

   This document defines extensions to the FTP specification STD9,
   RFC959, "FILE TRANSFER PROTOCOL (FTP)". These extensions provide the
   ability for a FTP Client to modify the last modification time, the
   creation time, or multiple facts (last modification time, creation
   time, operating system permissions, etc.) of a file in the server-FTP
   process NVFS. These extensions are implemented by three new optional
   commands: "MFMT" (Modify File Modification Time), "MFCT" (Modify File
   Creation Time), and "MFF" (Modify File Facts).

Comments

   Comments should be directed to David Somers (dsomers@trevezel.com).

Table of Contents

[1](#). **Introduction**

   The File Transfer Protocol (FTP) currently defined in STD 9, [RFC 959](#)
   [[1](#)], and in place on the Internet allows files to be transferred
   between a server-FTP and a user-FTP, and vice versa. When a file is
   transferred from the user-FTP to the server-FTP, the creation time,
   last modification time, and operating system facts (for example, file
   permissions) can not be specified. The NVFS typically sets these to
   default values.

   This document defines extensions to the File Transfer Protocol.
   Three new optional commands are added: "MFTM", "MFTC", "MFF". These
   commands allow the last modification time, creation time, or a list
   of facts to be modified for an object in the NVFS.

2. Document Conventions

   This document makes use of the conventions in [2]. That provides the
   interpretation of capitalized imperative words like MUST, SHOULD,
   etc.

   This document also uses notation defined in [1]. In particular, the
   terms "reply", "user", "NVFS", "file", "pathname", "FTP commands",
   "DTP", "user-FTP process", "user-PI", "user-DTP", "server-FTP
   process", "server-PI", "server-DTP", "mode", "type", "NVT", "control
   connection", "data connection", and "ASCII", are all used here as
   defined there.

   Syntax required is defined using the Augmented BNF defined in [3].
   Some general ABNF definitions are required throughout the document,
   those will be defined later in this section. At first reading, it
   may be wise to simply recall that these definitions exist here, and
   skip to the next section.

2.1 Basic Tokens

   This document defines basic tokens as specified in section 2.1 of
   [5].

2.2 Pathnames

   This document defines pathnames as specified in section 2.2 of [5].

2.3 Times

   This document defines times as specified in section 2.3 of [5].

2.4 Server Replies

   This document defines server replies as specified in section 2.4 of
   [5].

2.5 Interpreting Examples

   This document defines server replies as specified in section 2.5 of
   [5].

3. Modify File Modification Time (MFMT)

   The FTP command, MODIFY FILE MODIFICATION TIME (MFMT), is used to
   modify the last modification time of a file.

   It should be noted that similar functionality has been implemented by
   some server-PIs as the command MDTM. However, the use of MDTM to
   modify the last modification time of a file conflicts with the use of
   the MDTM command to retrieve the last modification time of a file as
   defined in [5]. It is proposed that the MFMT command be used instead
   of abusing MDTM for such purposes.

3.1 Syntax

   The syntax of the MFMT command is:
   mfmt = "MFMT" SP time-val SP pathname CRLF

   As with all FTP commands, the "MFMT" command label is interpreted in
   a case insensitive manner.

   The "time-val" specifies the last modification time to be applied to
   the file.

   The "pathname" specifies an object in the NVFS.

   The server-PI will respond to the MFMT command with a 213 reply, or
   an error response if the object does not exist, the last modification
   time could not be modified, or some other error has occurred.

   mfmt-response = "213" SP "ModifyTime=" time-val ";" SP pathname CRLF /
                   error-response

   The "time-val" in the response MUST be the modified last modification
   time of the file. This value MAY not be the same as that requested
   due to constraints of the NVFS to store the last modification time
   (for example, it may only have sufficient resolution to store the
   last modification time to the nearest minute instead of to the
   thousandths of a second that "time-val" MAY be specified to).

3.2 Error responses

   Where the command is correctly parsed, but the pathname identifies no
   existing entity, then a 550 reply SHOULD be sent. Where the command
   can not be correctly parsed, a 500 or 501 reply SHOULD be sent.
   Various 4xy replies are also possible in appropriate circumstances.

3.3 FEAT response for MFMT

Where a server-FTP process supports MFMT, as specified here, it MUST
include the response to the FEAT command [4]:

mfmt-feat = SP "MFMT" CRLF

The initial space shown in the mfmt-feat response is that required by
the FEAT command.

This string "MFMT" is not case sensitive, but SHOULD be transmitted
in upper case. Where MFMT is not supported, the MFMT line MUST NOT
be included in the FEAT response.

3.3.1 Example FEAT response

```
C> feat
S> 211- <any descriptive text>
S>  ...
S>  MFMT
S>  ...
S> 211 end
```

The ellipses indicate place holders where other features may be
included, and are not required. The one space indentation of the
feature lines is mandatory [4].

3.4 MFMT Examples

To modify the last modification time of a file called "Fred.txt" to
July 17, 2002 21:07:15,

```
C> MFMT 20020717210715 Fred.txt
S> 213 ModifyTime=20020717210715; Fred.txt
```

4. Modify File Creation Time (MFCT)

The FTP command, MODIFY FILE CREATION TIME (MFCT), is used to modify the last modification time of a file.

Implementers of this command on UNIX(TM) systems should note that the unix "stat" "st_ctime" field does not give creation time, and that unix file systems do not record creation time at all. Unix (and POSIX) implementations will normally not support this command.

4.1 Syntax

The syntax of the MFCT command is:
mfmt = "MFCT" SP time-val SP pathname CRLF

As with all FTP commands, the "MFCT" command label is interpreted in a case insensitive manner.

The "time-val" specifies the creation time to be applied to the file.

The "pathname" specifies an object in the NVFS.

The server-PI will respond to the MFCT command with a 213 reply, or an error response if the object does not exist, the creation time could not be modified, or some other error has occurred.

mfct-response = "213" SP "CreateTime=" time-val ";" SP pathname CRLF /
                error-response

The "time-val" in the response MUST be the modified creation time of the file. This value MAY not be the same as that requested due to constraints of the NVFS to store the creation time (for example, it may only have sufficient resolution to store the creation time to the nearest minute instead of to the thousandths of a second that "time-val" MAY be specified to).

4.2 Error responses

Where the command is correctly parsed, but the pathname identifies no existing entity, then a 550 reply SHOULD be sent. Where the command can not be correctly parsed, a 500 or 501 reply SHOULD be sent. Various 4xy replies are also possible in appropriate circumstances.

4.3 FEAT response for MFCT

Where a server-FTP process supports MFCT, as specified here, it MUST include the response to the FEAT command [4]:

```
mfct-feat = SP "MFCT" CRLF
```

The initial space shown in the mfct-feat response is that required by
the FEAT command.

This string "MFCT" is not case sensitive, but SHOULD be transmitted
in upper case. Where MFCT is not supported, the MFCT line MUST NOT
be included in the FEAT response.

4.3.1 Example FEAT response

```
C> feat
S> 211- <any descriptive text>
S>  ...
S>  MFCT
S>  ...
S> 211 end
```

The ellipses indicate place holders where other features may be
included, and are not required. The one space indentation of the
feature lines is mandatory [4].

4.4 MFCT Examples

To modify the creation time of a file called "Jim.txt" in the current
directory to July 17, 2002 21:22:30,

```
C> MFCT 20020717212230 Jim.txt
S> 213 CreateTime=20020717212230; Jim.txt
```

[5](). Modify File Facts (MFF)

The FTP command, MODIFY FILE FACTS (MFF), is used to modify one or
more facts of a file. These facts are attributes such as creation
time, last modification time, or operating system specific attributes
such as file permissions.

This command is complimentary to the MLSx commands as detailed in
[[5]()]. The MLSx commands are used to standardize retrieving the file
and directory information returned by the Server-FTP process; the MFF
command aims to standardize modifying the facts for the file and
directory objects in the NVFS.

[5.1]() Syntax

The syntax of the MFF command is:

```
mff                 = "MFF" [ mff-facts ] SP pathname CRLF
mff-facts           = 1*( mff-fact ";" )
mff-fact            = mff-standardfact / mff-osfact / mff-localfact
mff-standardfact    = mff-createtimefact / mff-modifytimefact
mff-createtimefact  = "CreateTime" "=" time-val
mff-modifytimefact  = "ModifyTime" "=" time-val
mff-osfact          = <IANA assigned OS name> "." token "=" *SCHAR
mff-localfact       = "X." token "=" *SCHAR
```

As with all FTP commands, the "MFF" command label is interpreted in a
case insensitive manner.

The "mff-facts" are a series of facts as keyword=value pairs each
followed by a semi-colon (";") character. Fact keyword names are
case-insensitive.

The "pathname" specifies an object in the NVFS.

The server-PI will respond to the MFF command with a 213 reply
containing a list of the facts that MUST detail the modified values,
or an error response if the object does not exist, a fact could not
be modified, or some other error has occurred.

```
mff-response = "213" SP 1*( mff-fact ";") SP pathname CRLF /
               error-response
```

The order of the "mff-fact" keyword=value pairs returned in the
response MAY be in any order.

5.1.1 CreateTime fact

   The CreateTime fact is used to modify the creation time of the object
   specified by "pathname".

   Implementers of this fact on UNIX(TM) systems should note that the
   unix "stat" "st_ctime" field does not give creation time, and that
   unix file systems do not record creation time at all. Unix (and
   POSIX) implementations will normally not support this fact.

5.1.2 ModifyTime fact

   The ModifyTime fact is used to modify the last modification time of
   the object specified by "pathname".

5.2 Operating System specific facts

   Facts that are specific to an operating system, or file system,
   SHOULD be specified by keywords that are prefixed by an IANA
   operating system name ([6]).

   Implementation Note: It is envisioned that the operating system
   specific facts will be identical to those used by the MLSx command as
   detailed in [5].

5.2.1 Example Operating System specific facts

   UNIX.mode   -- Unix file modes (permissions)
   WINDOWS-NT.SIS.Author -- Windows NT,
                            Summary Information Stream, Author property

5.3 Local/Experimental "X." facts

   Implementations may define keywords for experimental, or private,
   use. All such keywords MUST begin with the two character sequence
   "X.". As type names are case-insensitive, "X." and "x." are
   equivalent.

5.4 Error responses

   Where the command is correctly parsed, but the pathname identifies no
   existing entity, then a 550 reply SHOULD be sent. Where the command
   can not be correctly parsed, a 500 or 501 reply SHOULD be sent. If
   an unknown fact is provided, a 504 reply SHOULD be sent. Various 4xy
   replies are also possible in appropriate circumstances.

5.5 FEAT response for MFF

   Where a server-FTP process supports MFF, as specified here, it MUST
   include in the response to the FEAT command [4], a feature line
   containing the string "MFF". This string is not case sensitive, but
   SHOULD be transmitted in upper case. As well as indicating MFF
   support, the MFF feature line indicates which MFF facts are available
   for modification by the server-FTP process. Where MFF is not
   supported, the MFF line MUST NOT be included in the FEAT response.

   mff-feat = SP "MFF" SP factlist CRLF
   factlist = 1*(factname ";")

   The initial space shown in the mff-feat response is that required by
   the FEAT command.

5.5.1 Example FEAT responses

   C> feat
   S> 211- <any descriptive text>
   S> ...
   S> MFF ModifyTime;
   S> ...
   S> 211 end

   This server-FTP process indicates that it supports MFF, and only
   supports modification of the last modification time of an object in
   the NVFS.

   C> feat
   S> 211- <any descriptive text>
   S> ...
   S> MFF CreateTime;ModifyTime;WINDOWS-NT.SIS.Author;
   S> ...
   S> 211 end

   This server-FTP process indicates that it supports MFF, and supports
   modification of the creation time, last modification time, and an
   operating system specific fact called "WINDOWS-NT.SIS.Author" of an
   object in the NVFS.

   The ellipses indicate place holders where other features may be
   included, and are not required. The one space indentation of the
   feature lines is mandatory [4].

5.6 MFF Examples

   To modify the creation time of a file called "Sheila.txt" to July 17,

```
   2002 21:22:30,

   C> MFF CreateTime=20020717212230; Sheila.txt
   S> 213 CreateTime=20020717212230; Sheila.txt
```

   To modify the permissions on a Unix-based NVFS for the file called
   "Bob.txt" to 777,

```
   C> MFF UNIX.perms=777; Bob.txt
   S> 213 UNIX.perms=777; Bob.txt
```

   To modify the permissions on a Unix-based NVFS for the file called
   "Fred.txt" to 777, and the creation time to July 18, 2002 01:28:45,

```
   C> MFF UNIX.perms=777;CreateTime=20020718012845; Bob.txt
   S> 213 CreateTime=20020718012845;UNIX.perms=777; Bob.txt
```

   If the same request was made to a server-FTP process that does not
   support the UNIX.perms fact,

```
   C> MFF UNIX.perms=777;CreationTime=20020718012845; Bob.txt
   S> 504 Parameter Not Implemented (UNIX.perms)
```

6. IANA Considerations

   This specification makes use of some lists of values currently
   maintained by the IANA. It does not add any values to any existing
   registries.

6.1 The OS Specific fact registry

   The MFF command reuses the OS Specific fact registry that is used by
   the MLSx commands as detailed in [5]

   The OS names for the OS portion of the fact name must be taken from
   the IANA's list of registered OS names. To add a fact name to this
   OS specific registry of OS specific facts, an applicant must send to
   the IANA a request, in which is specified the OS name, the OS
   specific fact name, a definition of the syntax of the fact value,
   which must conform to the syntax of a token as given in this
   document, and a specification of the semantics to be associated with
   the particular fact and its values. Upon receipt of such an
   application, and if the combination of OS name and OS specific fact
   name has not been previously defined, the IANA will add the
   specification to the registry.

   Any examples of OS specific facts found in this document are to be
   treated as examples of possible OS specific facts, and do not form a
   part of the IANA's registry merely because of being included in this
   document.

7. Security Considerations

No significant security issues, not already present in the FTP
protocol, are believed to have been created by this extension.

References

    [1]   Postel, J. and J. Reynolds, "File Transfer Protocol", STD 9, RFC
          959, October 1985.

    [2]   Bradner, S., "Key words for use in RFCs to Indicate Requirement
          Levels", BCP 14, RFC 2119, March 1997.

    [3]   Crocker, D., "Augmented BNF for Syntax Specifications: ABNF",
          RFC 2234, November 1997.

    [4]   Hethmon, P., "Feature negotiation mechanism for the File
          Transfer Protocol", RFC 2389, August 1998.

    [5]   Elz, R. and P. Hethmon, "Extensions to FTP", draft-ietf-ftpext-
          mlst-15 (work in progress), April 2002.

    [6]   <ftp://ftp.iana.org/assignments/operating-system-names>

Author's Address

    David M. P. Somers
    Trevezel Systems GmbH
    Rambouxstrasse 26
    Trier D-54224
    GERMANY

    EMail: dsomers@trevezel.com

Appendix A. Acknowledgements

Thank you to the authors and editors of [5] for the facts in their MLSx command which have been hijaaked (in the nicest possible way) by the MFF command herein.

A big thanks for xml2rfc which greatly helped in the production of this document.

Full Copyright Statement

Acknowledgement