             The "MFMT", "MFCT", and "MFF" Command Extensions for FTP
                          draft-somers-ftp-mfxx-04

Status of this Memo

Abstract

   This document defines extensions to the FTP specification STD 9, RFC
   959, "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 an object in the server-
   FTP process NVFS. These extensions are implemented by three new
   optional commands: "MFMT" (Modify Fact: Modification Time), "MFCT"
   (Modify Fact: Creation Time), and "MFF" (Modify Fact: Facts).

Comments

   Comments are solicited and should be addressed to David Somers
   (dsomers@omz13.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,
   specifically three new optional commands: "MFMT", "MFCT", and "MFF".

   The "MFMT" command allows the last modification time an object in the
   NVFS to be modified. This is an alternative to abusing "MDTM" (as
   defined in section 3 of [2]), which was only intended to read the
   modification time and not to set it as some implementations do.

   The "MFCT" command allows the creation time an object in the NVFS to
   be modified.

   The "MFF" command allows multiple facts of an object in the NVFS to
   be modified. The MFF command is complimentary to the MLSx commands
   as detailed in [2]. The MLSx commands provides a standardized way of
   retrieving facts for objects in the NVFS; the MFF command aims to
   standardize modifying (or setting) the facts for the objects in the
   NVFS.

2. Document Conventions

   The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
   "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
   document are to be interpreted as described in [3]."

   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 [4].
   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 in the same manner as that as
   specified in section 2.1 of [2].

2.2. Pathnames

   This document defines pathnames in the same manner as that specified
   in section 2.2 of [2].

2.3. Times

   This document defines times in the same manner as that as specified
   in section 2.3 of [2].

2.4. Server Replies

   This document defines server replies in the same manner as that
   specified in section 2.4 of [2].

2.5. Interpreting Examples

   This document presents examples in the same manner as that specified
   in section 2.5 of [2].

3. Modify Fact: Modification Time (MFMT)

   The FTP command, MODIFY FACT: MODIFICATION TIME (MFMT), is used to
   modify the last modification time of an object in the NVFS.

   The ability to modify the last modification time MAY also be
   achieved, if supported by the server-PI, by using the "Modify" fact
   to the MFF command as detailed in Section 5.

   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 an object conflicts with the use
   of the MDTM command to retrieve the last modification time of an
   object as defined in [2]. It is RECOMMENDED that, if possible,
   client-PIs use the MFMT command instead of abusing the MDTM command
   to change the modification time of an object in the NVFS.

   If the client-PI wants to modify both the modification time and the
   creation time, it is RECOMMENDED that the MFF command, if supported
   by the server-PI, be used instead.

   If the NVFS supports the concept of both creation times and
   modification times, it is RECOMMENDED that server-PI give priority to
   setting the modification time, even if it means that to set the
   modification time requested by the client-PI the server-PI must also
   change the creation time. An example of this prioritization is if
   the requested modification time is prior to the current creation time
   and the NVFS does not permit the modification time to be prior to the
   creation time; the only way to set the modification time in such a
   situation is for the server-PI to automagically change the creation
   time to a time before or of the same as the requested modification
   time. The rationale for this prioritization of modification over
   creation time is because, generally speaking, it is more important
   for the modification time to be more valid than the creation time as
   the modification time is typically used to perform object
   synchronization between hosts.

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 object.

The "pathname" specifies an object in the NVFS.

The server-PI MUST 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 "Modify=" time-val ";" SP pathname CRLF /
                error-response

The "time-val" in the response MUST be the modified last modification
time of the object. 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). It is
RECOMMENDED that the client-PI parse the 213 response to determine
what the modification time was actually modified to by the server-PI.

3.2. Error responses

Where the command is correctly parsed, but the pathname identifies no
existing object, then a 550 reply SHOULD be sent. Where the command
can not be correctly parsed, a 500 or 501 reply SHOULD be sent. If
the date or time specified is invalid (for example, February 29 in a
non-leap year), then a 501 reply MUST be sent. Various 4xy replies
are also possible in appropriate circumstances.

3.3. FEAT response for MFMT

Where a server-FTP process supports the MFMT command, as specified
here, it MUST include the response to the FEAT command [5]:

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 [5].

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 Modify=20020717210715; Fred.txt
```

4. Modify Fact: Creation Time (MFCT)

   The FTP command, MODIFY FACT: CREATION TIME (MFCT), is used to modify
   the creation time of an object in the NVFS.

   The ability to modify the creation time MAY also be achieved, if
   supported by the server-PI, by using the "Create" fact to the MFF
   command as detailed in Section 5.

   If the client-PI wants to modify both the modification time and the
   creation time, it is RECOMMENDED that the MFF command, if supported
   by the server-PI, be used instead.

4.1. Syntax

   The syntax of the MFCT command is:
   mfct = "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
   object.

   The "pathname" specifies an object in the NVFS.

   The server-PI MUST 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 "Create=" time-val ";" SP pathname CRLF /
                   error-response

   The "time-val" in the response MUST be the modified creation time of
   the object. 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). It is RECOMMENDED that the client-PI
   parse the 213 response to determine what the creation time was
   actually modified to by the server-PI.

4.2. Error responses

   Where the command is correctly parsed, but the pathname identifies no
   existing object, then a 550 reply SHOULD be sent. Where the command
   can not be correctly parsed, a 500 or 501 reply SHOULD be sent. If
   the date or time specified is invalid (for example, February 29 in a

   non-leap year), then a 501 reply MUST be sent. Various 4xy replies
   are also possible in appropriate circumstances.

4.3. FEAT response for MFCT

   Where a server-FTP process supports the MFCT command, as specified
   here, it MUST include the response to the FEAT command [5]:

   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 [5].

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 Create=20020717212230; Jim.txt

5. Modify Fact: Facts (MFF)

   The FTP command, MODIFY FACT: FACTS (MFF), is used to modify one or
   more facts of an object in the NVFS. These facts are attributes such
   as creation time, last modification time, or operating system
   specific attributes such as access permissions.

   The MFF command is complimentary to the MLSx commands as detailed in
   [2]. The MLSx commands provides a standardized way of retrieving
   facts for objects in the NVFS; the MFF command aims to standardize
   modifying (or setting) the facts for the objects in the NVFS. The
   format of facts is identical to that specified in [2].

   If the client-PI wants to modify only the modification time or only
   the creation time, it is RECOMMENDED that the MFCT (see Section 4) or
   MFMT (see Section 3) commands, if supported by the server-PI, be used
   respectively instead of the MFF command.

   It MAY be possible for a client to attempt to set the modification
   time prior to the creation time; this situation may be nonsensical,
   but it may be necessary, and it is RECOMMENDED that it not be
   considered an error by the server-PI. If necessary, the server-PI
   SHOULD prioritize modifying the modification time, even if that means
   automagically changing the creation time (due to, for example, NVFS
   restrictions).

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-create-fact / mff-modify-fact
mff-create-fact    = "Create" "=" time-val
mff-modify-fact    = "Modify" "=" 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
   separated by a semi-colon (";") character. Fact keyword names are
   case-insensitive.

   The "pathname" specifies an object in the NVFS.

The server-PI MUST respond to the MFF command with a 213 reply
containing a list of the facts that MUST detail the values actually
set for all the facts specified in the request (which MAY differ from
that requested), or an error response.

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.

The "time-val" in the response MAY not be the same as that requested
due to constraints of the NVFS to store such times (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).

The error response MUST be returned if any of the following
conditions happen:

o   The object does not exist.

o   A fact could not be modified, for example due to insufficient
    access control permissions.

o   An unknown fact was specified.

When an error response is returned, the client-PI MUST make no
assumptions about which, if any, facts have been modified. In other
words, modifying facts is not an atomic operation. The client-PI can
issue an MLST (if the server-PI supports the MLST command) to
determine what attributes have, in fact, been modified.

5.2. Standard facts

This document defines the following standard facts for use by MFF:

o   Create

o   Modify

5.2.1. Create fact

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

Note that this is the same fact that can be read using the MLST and
MLSD commands in [2], and is detailed in see section 7.5.4 of the

aforementioned document.

5.2.2. Modify fact

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

Note that this is the same fact that can be read using the MLST and
MLSD commands in [2], and is detailed in see section 7.5.3 of the
aforementioned document.

5.3. 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 [7].

Implementation Note: It is envisioned that the operating system
specific facts will be identical to those used by the MLSx command as
detailed in [2]; implementers can then use the same logic to process
facts whether for MLSx or MFF.

The specification of Operating Systems specific facts is explicitly
outside the scope of this document. Such specifications SHOULD be
documented elsewhere (that is, in an internet draft, RFC, etc.)

5.3.1. Example Operating System specific facts

The following examples are only indicative of how it is anticipated
that some Operating System specific facts could be implemented.

```
UNIX.mode              -- Unix file mode (in octal)
UNIX.owner             -- Unix owner (as a decimal Uid or a username)
UNIX.group             -- Unix group (as a decimal Gid or a groupname)
WINDOWS-NT.SIS.Author  -- Windows NT,
                          Summary Information Stream, Author property
```

5.4. 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 fact names are case-insensitive, "X." and "x." are
equivalent.

5.5. Error responses

Where the command is correctly parsed, but the pathname identifies no
existing object, then a 550 reply SHOULD be sent. Where the command

   can not be correctly parsed, a 500 or 501 reply SHOULD be sent. If
   the date or time specified for a Create or Modify fact is invalid
   (for example, February 29 in a non-leap year), then a 501 reply MUST
   be sent. If an unknown fact is provided, a 504 reply SHOULD be sent,
   and it is RECOMMENDED that the 504 reply indicate the name(s) of the
   unknown fact(s). Various 4xy replies are also possible in
   appropriate circumstances.

5.6. FEAT response for MFF

   Where a server-FTP process supports the MFF command, as specified
   here, it MUST include in the response to the FEAT command [5], 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, as a semi-
   colon delimited list, 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.

5.6.1. Example FEAT responses

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

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

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

   This server-FTP process indicates that it supports the MFF command,
   and supports modification of the create fact, modify fact, and an
   operating system specific fact called "WINDOWS-NT.SIS.Author" of an

   object in the NVFS. Note that the "WINDOWS-NT.SIS.Author" fact is an
   example of what could be possible, not what is possible; such a fact
   may exist, but its definition is outside the scope of this document.

   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 [5].

5.7. MFF Examples

   Note that some of these examples refer to the UNIX.mode fact; whether
   such a fact exists or not is outside the scope of this document, and
   is used below only to show what could be possible and not what is
   possible.

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

   C> MFF Create=20020717212230; Sheila.txt
   S> 213 Create=20020717212230; Sheila.txt

   Note that the above could also be achieved using the MFCT command,
   thus:

   C> MFCT 20020717212230 Sheila.txt
   S> 213 Create=20020717212230; Sheila.txt

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

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

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

   C> MFF UNIX.mode=777;Create=20020718012845; Fred.txt
   S> 213 Create=20020718012845;UNIX.mode=777; Fred.txt

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

   C> MFF UNIX.mode=777;Create=20020718012845; Fred.txt
   S> 504 Parameter Not Implemented (UNIX.mode)

   The creation time may or may not have been modified by the server-PI
   before the server-PI determined that the UNIX.mode fact was not

implemented.

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 [2]

   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.

   A general discussion of issues related to the security of FTP can be
   found in [6].

8. References

8.1. Normative References

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

   [2] Hethmon, P., "Extensions to FTP", RFC 3659, March 2007.

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

   [4] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax
        Specifications: ABNF", RFC 4234, October 2005.

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

8.2. Informative References

   [6] Allman, M. and S. Ostermann, "FTP Security Considerations",
        RFC 2577, May 1999.

URIs

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

   [8]   <http://xml.resource.org>

Appendix A. Acknowledgements

   Thank you to the authors and editors of [2] for the facts in their
   MLSx command which have been hijacked (in the nicest possible way) by
   the MFF command herein.

   A big thanks to xml2rfc [8] which greatly aided in the production of
   this document.

   Finally, many thanks to all who commented on drafts of this document
   and helped clarify and improve it, and an even bigger thanks to those
   who have implemented or who intend to implement the FTP commands
   presented in this document.

Author's Address

    David M. P. Somers
    Tunnelstrooss 36
    Lipperscheid 9164
    LU

    Email: dsomers@omz13.com
    URI: http://www.omz13.com/ftp-mfxx