

FTP-FTAM Gateway Specification

Status of the Memo

This RFC specifies an IAB standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "IAB Official Protocol Standards" for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Abstract

This memo describes a dual protocol stack application layer gateway that performs protocol translation, in an interactive environment, between the FTP and FTAM file transfer protocols.

Two key assumptions are made: 1) POSIX file naming conventions and hierarchical organization, rather than proprietary conventions are in use; and 2) X.500 Directory Services are available.

Acknowledgments

The authors of this RFC would like to express their appreciation to the individuals and organizations that participated in the implementation of the FTP-FTAM Application Layer Gateway and its fielding on the MILNET. Implementation credits go to Mr. John Scott, formerly of the MITRE Corporation, while fielding credits are extended to James Graham and R. Greg Lavender of Open Networks, Inc. (formerly NetWorks One) and Robert Cooney of the Naval Computer and Telecommunications Station (NCTS) Washington. Dr. Marshall Rose is to be commended for recognizing the importance of the FTP-FTAM gateway and promulgating it as a part of the ISO Development Environment (ISODE). The following individuals have provided valuable editorial comments: Larry Friedman, Donna Vincent and Michael Resnick of Digital Equipment Corporation; Robert Cooney of NCTS; and S.E. Hardcastle-Kille of University College London. Funding of the FTP-FTAM Gateway Request for Comments effort was provided by Open Networks Inc. and the Defense Information Systems Agency (DISA), formerly the Defense Communications Agency. DISA sponsors include Len Tabacchi, George Bradshaw, Tom Clarke, and Betsy Turner.

Table of Contents

1.	Introduction.....	2
1.1.	Relationship to Other Work	3
1.2.	Overview of Gateway Operation	4
2.	Gateway Architecture.....	6
3.	Network Naming and Addressing.....	8
4.	Use of the Gateway Services.....	9
4.1.	FTP-Initiated Gateway Service	9
4.2.	FTAM-Initiated Gateway Service	11
4.3.	Summary of Usage	12
5.	Gateway State Variables and Transitions.....	13
5.1.	FTP-Initiated Gateway Service	14
5.2.	FTAM-Initiated Gateway Service	16
6.	Document Type Support.....	18
6.1.	Notes on NBS-9	18
7.	Functional Comparison of FTP and FTAM.....	19
7.1.	Loss of Functionality	20
8.	Mapping of Protocol Functions and Representations.....	20
8.1.	FTP-Initiated Gateway Service	22
8.2.	FTAM-Initiated Gateway Service	38
9.	Mapping between FTP Reply Codes and FTAM Parameters.....	47
9.1.	FTP Reply Codes to FTAM Parameters	48
9.2.	FTAM Parameters to FTP Reply Codes	50
9.3.	Future Mapping Problem	54
9.4.	Error Handling	54
10.	Implementation and Configuration Guidelines.....	54
10.1.	Robustness	54
10.2.	Well-Known TCP/IP Port	55
10.3.	Gateway Listener Processes	55
10.4.	Implementation Testing	55
10.5.	POSIX File Naming and Organization	55
11.	Security Considerations.....	55
12.	References.....	56
13.	Authors' Addresses.....	58

1. Introduction

The TCP/IP and OSI protocol suites will coexist in the Internet community for several years to come. As more and more OSI hosts are fielded on the Internet, the requirement for gateways between the two protocol suites becomes more pressing.

This specification describes an application layer gateway providing interoperability between the TCP/IP File Transfer Protocol (FTP) and the OSI File Transfer, Access, and Management (FTAM) protocol. The proposed application layer gateway is based on a bi-directional set of mappings between the FTP and FTAM protocols. Since the protocols

have quite different command structures, the mappings between them are not one-to-one. This paper assumes knowledge of the File Transfer Protocol (FTP) [RFC959] and the File Transfer, Access, and Management Protocol (FTAM) [ISO8571-1,2,3,4,5].

Two important goals of the mappings are to:

Provide FTP users with as much emulated FTP capability on an FTAM Responder as possible, and

Provide FTAM users with as much emulated FTAM capability on an FTP Server as possible.

Though it is anticipated that the application layer gateway will be implemented on full protocol suites of both TCP/IP and OSI, at least one implementation of such a gateway (included in the ISO Development Environment) can be configured to operate FTAM over either OSI or TCP/IP lower-layer services.

1.1. Relationship to Other Work

Ideas presented in this specification are based on lessons learned in fielding the gateway on the MILNET, operational at NCTS Washington D.C. since 1989, and on the efforts of M. A. Wallace et al. of the National Institute of Standards and Technology (NIST) [NIST86]. In 1986, NIST published a design document for an FTP-FTAM gateway. Since that time, at least one implementation (for a subset of the FTP and FTAM protocols) of the gateway has been developed [MITRE87] and is included with the ISODE. This implementation is based on the NIST protocol translator gateway design [NIST86].

This document's contribution to the advancement of the FTP-FTAM gateway concept is to:

- * Enhance the user interaction capability provided by the ISODE implementation of the FTP-FTAM application layer gateway.
- * Clarify and enhance the mappings (FTP to FTAM, FTAM to FTP) documented by NIST.
- * Provide guidelines for fielding the FTP-FTAM application layer gateway on the Internet so that it is useful as an Internet resource.
- * Produce a formal specification for the FTP-FTAM gateway suitable for implementors to use in building additional FTP-FTAM gateways.

- * Provide a formal specification for organizations wishing to procure FTP-FTAM gateways.

1.2. Overview of Gateway Operation

The gateway provides a virtual end-to-end application file transfer service. As data is sent via FTP, the gateway immediately maps the requested function to FTAM and passes it to the FTAM host. In a similar fashion, but using a different set of mappings, an FTAM request is sent to the gateway, immediately mapped to an FTP function, and passed along to the FTP host.

In FTP, the two parties involved in a file transfer are the Client and Server. The Client is responsible for initiating a connection to the Server. Once the connection is established, all service requests originate from the Client. The FTP-FTAM gateway does not support the FTP three node model.

In FTAM, the two parties involved in a file transfer are the Initiator and Responder. The Initiator is responsible for initiating a connection to the Responder. Once the connection is established, either the Initiator or Responder may issue service requests to the other.

The FTP-FTAM gateway provides two sets of services:

1. FTP-Initiated Gateway Services

Utilized when an FTP Client contacts the FTP-FTAM gateway to instigate a file transfer with an FTAM Responder.

2. FTAM-Initiated Gateway Services

Utilized when an FTAM Initiator contacts the FTP-FTAM gateway to instigate a file transfer with an FTP Server.

The gateway services' names were selected to identify the roles that the FTP-FTAM gateway plays when performing file transfers. For example, when a file transfer is instigated by an FTP Client, it contacts the FTP Server portion of the gateway, which maps protocol information to the FTAM Initiator portion of the gateway, which in turn contacts the remote FTAM Responder. This example scenario uses the FTP-Initiated Gateway Services.

Figure 1 illustrates the perspective of the application process in the FTP-Initiated service. Figure 2 illustrates that of the FTAM-Initiated service.

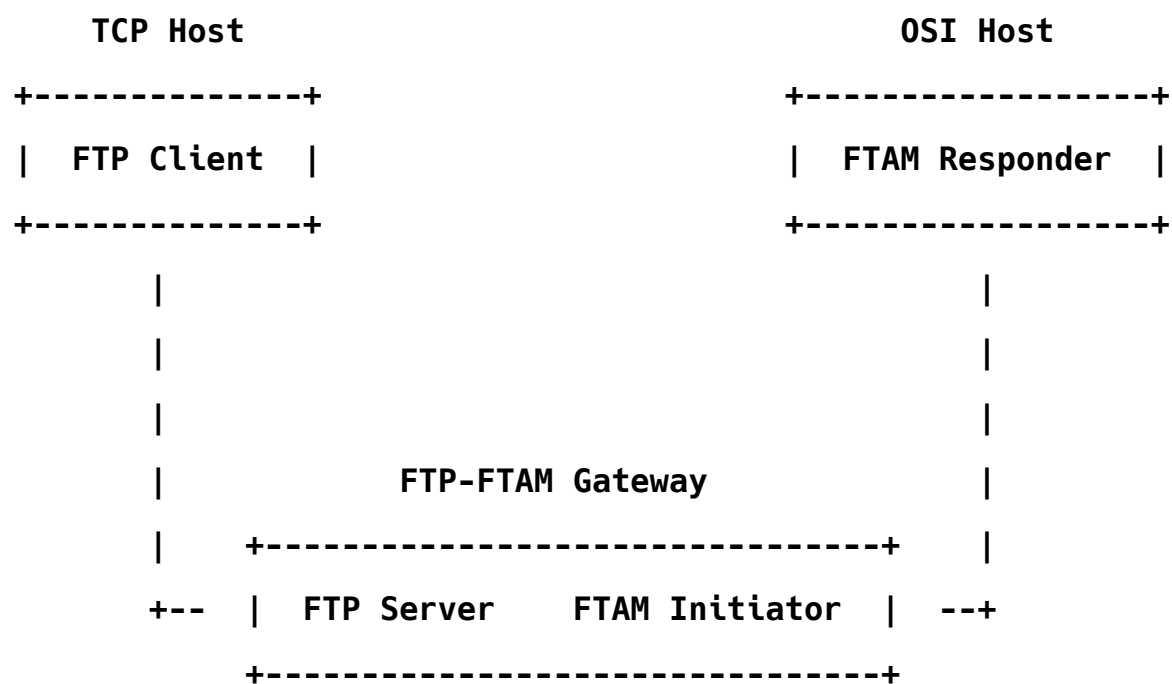


Figure 1 - FTP-Initiated Gateway Service

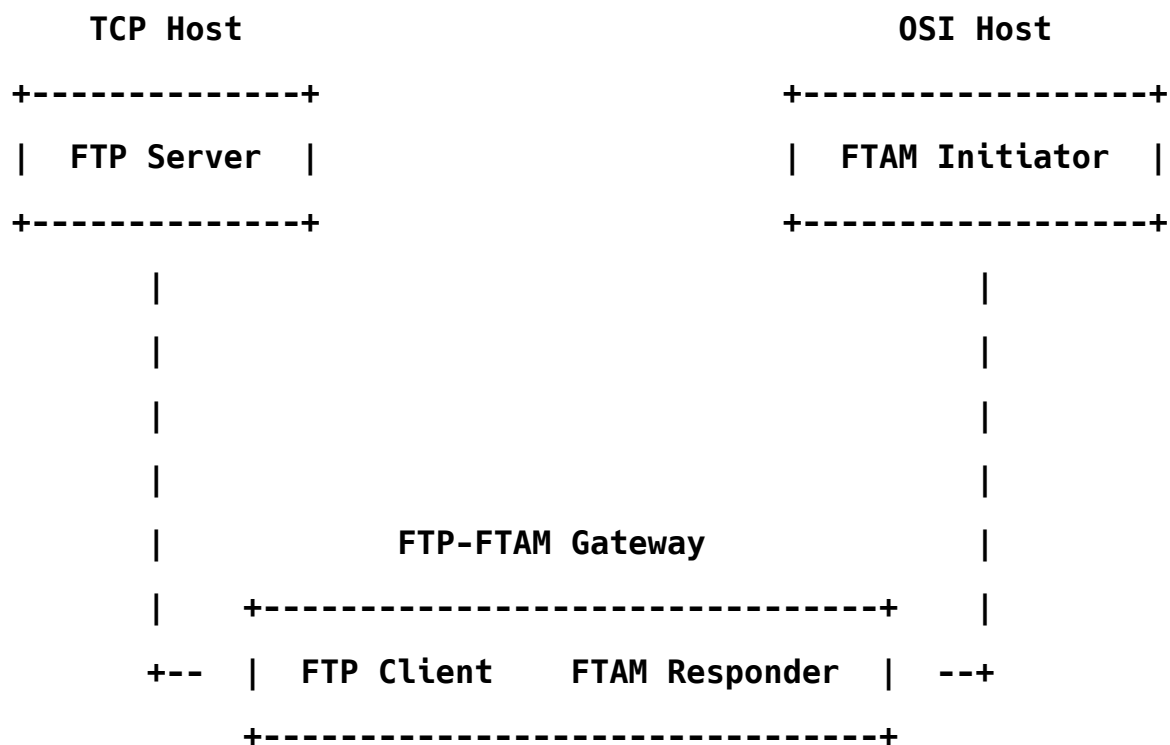


Figure 2 - FTAM-Initiated Gateway Service

2. Gateway Architecture

The gateway architecture, termed a protocol translator [NIST86], is depicted in Figure 3. It implements TCP/IP and OSI protocol stacks with an application level process providing the link between the two. The link between FTP and FTAM is defined by two sets of protocol mappings, one each for the FTP-Initiated and FTAM-Initiated service sets.

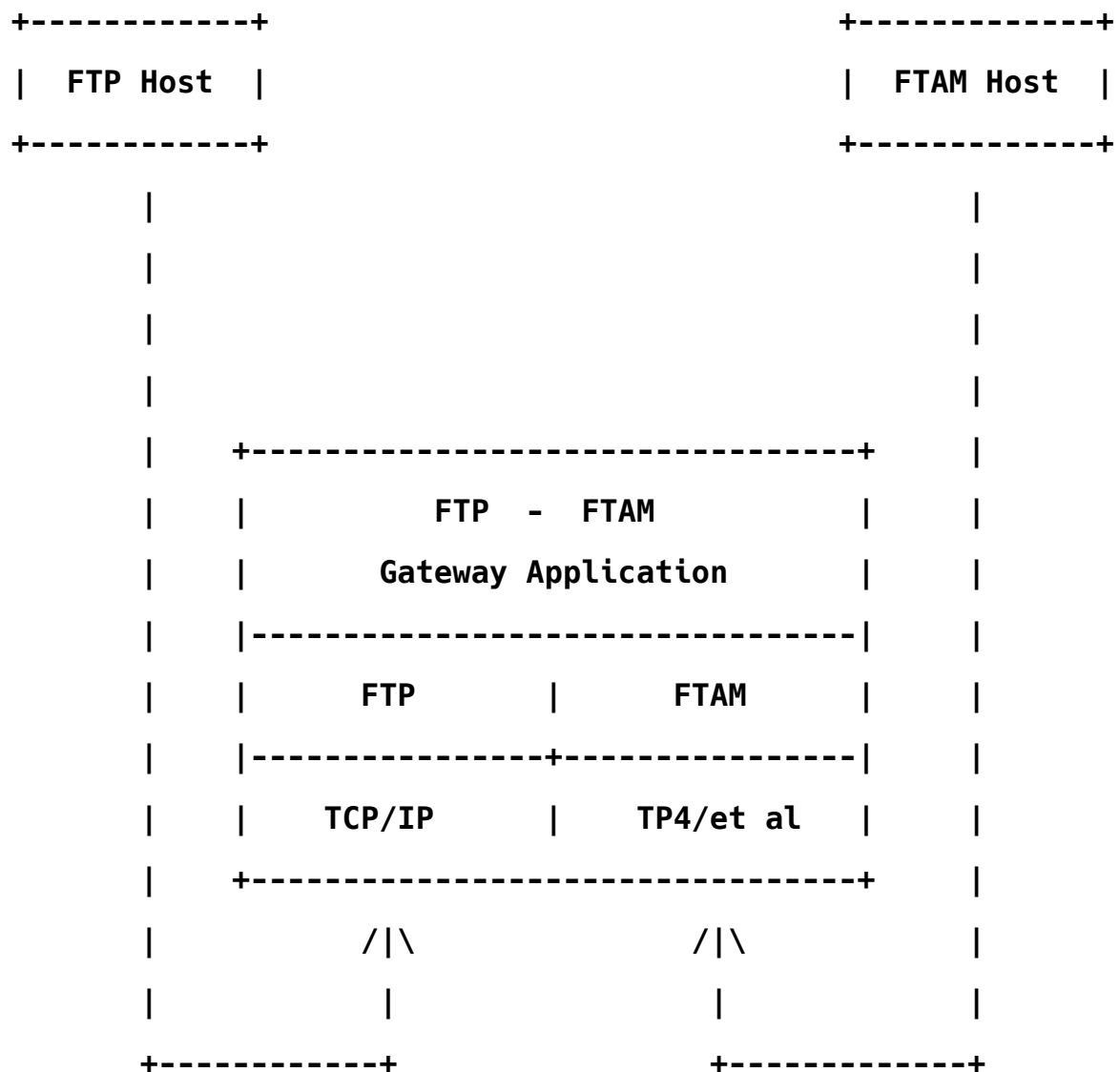


Figure 3 - Gateway Protocol Stack

A fundamental aspect of this gateway architecture is that data is mapped and transmitted immediately; i.e., no transferred file need ever reside on the gateway file system. In the context of this document, the term "filesystem" refers to the file access and maintenance mechanisms provided by the operating system. This lack of gateway filesystem interaction helps speed up the end-to-end data transfer. Another speed-enhancing feature of this architecture is that both the FTP and FTAM network connections can operate

simultaneously. Additional advantages include:

1. FTP and FTAM hosts require no modification to utilize gateway services.
2. Users require no knowledge of the other protocol.
3. Gateway access control is not impaired (since users cannot directly access the gateway filesystem).
4. No additional filesystem space is required on the gateway.
5. Interactive nature of protocols is preserved.
6. Users become aware of fatal errors immediately.

Disadvantages of this design include the initial coding effort required to develop the gateway and the subsequent re-coding efforts required to keep it current.

3. Network Naming and Addressing

The network naming and addressing schemes used by FTP (Domain Names (DN), IP Addresses) and FTAM (Distinguished Names, Presentation Addresses) are quite different. This issue is quite apparent when a user of one protocol needs to identify a destination host of the other protocol.

In the TCP/IP naming and addressing scheme, the identity of the FTP Server is its DN and its IP address [RFC1101]. To initiate a connection to an FTP Server, the FTP Client looks up a DN in either the Domain Name System (DNS) or static host table and obtains an IP address.

In the OSI naming and addressing scheme, the identity of the FTAM Responder service is its Distinguished Name in the OSI Directory (X.500 or static table) and its Presentation address. The Distinguished Name is an authoritative description of the service. A Presentation address consists of a Presentation selector, a session selector, a transport selector, and a network address. To initiate a connection to an FTAM Responder, the FTAM Initiator contacts the OSI Directory, presents the Distinguished Name of the desired FTAM Responder and asks for the Presentation address attribute associated with that name.

An alternative to the direct use of Distinguished Names is to use "User Friendly Naming", as defined in [Kille92]. Gateway support for "User Friendly Naming" is recommended, but not required.

4. Use of the Gateway Services

4.1. FTP-Initiated Gateway Service

The FTP Client uses the FTP-Initiated gateway service to utilize the resources of an FTAM Responder.

To initiate a file transfer from an FTP Client, the Client connects to the FTP-Initiated gateway service via TCP/IP. The gateway then establishes a connection, via OSI, to the FTAM Responder. At this point, the user can initiate file transfer operations.

The FTP Client is responsible for providing the gateway with an authoritative Distinguished Name, or a User Friendly Name, of the desired OSI filestore. It is the responsibility of the gateway to resolve this Distinguished Name, or User Friendly Name, to its corresponding Presentation address.

The logon sequence taken by an FTP Client when initiating a file transfer with an FTAM Responder is given below:

```
% ftp gateway
ftp> site Distinguished-Name-of-FTAM Responder
ftp> user username
ftp> pass password
```

The "ftp gateway" command initiates the connection between the FTP Client and the gateway. Once connected to the gateway, the FTP Client should identify the desired FTAM Responder service via the Responder's Distinguished Name, or User Friendly Name, which is resolved by an algorithm running on the Directory Services provider. This information is sent via a "site Distinguished-Name-of-FTAM Responder" or "site UFN-of-FTAM Responder" command.

Upon receipt of a Distinguished Name or a User Friendly Name, it is the gateway's responsibility to resolve it to the Presentation Address associated with that name. This resolution is done by contacting the OSI Directory (X.500 or local static table) and presenting the Distinguished Name or User Friendly Name. Once the Presentation address is obtained, the gateway can attempt a connection with the ultimate destination file transfer service represented by this Presentation address.

The userid is passed via the "user username" command, and the password is passed via the "pass password". If the FTAM Responder requires a password, a password prompt should appear after issuing the "user username" command. It is anticipated that stronger authentication mechanisms will be required for DoD gateways in the

future.

Using a specific example, suppose an FTAM Responder has the following Distinguished Name:

CountryName	=	"US"
Organization	=	"Open Networks"
OrganizationalUnit	=	"Network Services"
CommonName	=	"netwrx1"
CommonName	=	"FTAM service"

and the FTP-FTAM gateway is available at "washdc1-osigw.navy.mil".

The FTP user action will appear as:

```
% ftp washdc1-osigw.navy.mil
ftp> site "c=US@o=Open Networks@ou=Network Services@cn=netwrx1
      @cn=FTAM service"
ftp> user mindel
ftp> pass *****
```

The "ftp washdc1-osigw.navy.mil" command initiates the connection between the FTP Client and the FTP-FTAM gateway at the Washington Navy Yard, Washington D.C. Once connected, the OSI filestore at Open Networks is identified via its Distinguished Name, "@c=US@o=Open Networks@ou=Network Services@cn=netwrx1@cn=FTAM service".

Alternatively, a User Friendly Name, such as:

"netwrx1, Open Networks, us"

can be specified, enabling the following FTP user action:

```
% ftp washdc1-osigw.navy.mil
ftp> site "netwrx1, Open Networks, us"
ftp> user mindel
ftp> pass *****
```

As this example indicates, use of an intermediate gateway is not transparent. To partially alleviate this awkwardness, the gateway can be made more transparent through the registration of the FTAM host in the DNS using the address of the gateway [RFC1279].

An example will clarify this point. Suppose that the "netwrx1, Open Networks, us" FTAM host is registered in the TCP/IP DNS with the DN of "ftam-service.netwrx1.com" and the IP address of the "washdc1-osigw.navy.mil" gateway. In this example, the following set of user actions is required:

```
% ftp ftam-service.netwx1.com
ftp> user mindel
ftp> pass *****
```

Since the "ftam-service.netwx1.com" really points to the gateway address, the first command will connect the FTP Client to the gateway. The gateway will then use the name (using [RFC1279]) to determine where the actual FTAM host is resident. Gateway support for RFC1279 is recommended, but not required.

4.2. FTAM-Initiated Gateway Service

The FTAM Initiator uses the FTAM-Initiated gateway service to utilize the resources of an FTP Server.

To initiate a file transfer from an FTAM Initiator, the Initiator connects to the FTAM-Initiated gateway service via OSI. The gateway then establishes a connection, via TCP/IP, to the FTP Server. At this point, the user can initiate file transfer operations.

The FTAM Initiator is responsible for providing the gateway with an authoritative DN of the desired TCP/IP filestore. It is the responsibility of the gateway to resolve this DN to its corresponding IP address.

The logon sequence taken by an FTAM Initiator when initiating a file transfer with an FTP Server is given below:

```
% ftam gateway
ftam> user username@DNS-string
ftam> pass password
```

The "ftam gateway" command initiates the connection between the FTAM Initiator and the gateway. Once connected, userid and TCP/IP filestore are identified in the "username@DNS-string" argument to the user command. If the FTP Server requires a password, a password prompt should appear after issuing the user command.

The gateway should incorporate the BIND Resolver functionality so that upon receipt of a Domain Name, the Gateway FTP Client can resolve it via the distributed Domain Name System.

Using a specific example, suppose that a FTP Server has the following Domain Name: "ftp-service.netwx1.com" and an FTP-FTAM gateway is available at:

CountryName	=	"US"
Organization	=	"GOV"
OrganizationalUnit	=	"DOD"
OrganizationalUnit	=	"DISA"
Locality	=	"Washington Navy Yard"
CommonName	=	"wnyosi7"

The FTAM user action will appear as:

```
% ftam @c=US@o=GOV@ou=DOD@ou=DISA@l=Washington Navy Yard
    @cn=wnyosi7
ftam> user mindel@ftp-service.netwrx1.com
ftam> pass *****
```

Alternatively, a User Friendly Name could be used rather than the Distinguished Name.

As mentioned in the previous section, "Use of the FTP-Initiated Gateway Service", use of an intermediate gateway is not transparent. The gateway can be made more transparent through the registration of the FTP host in the X.500 OSI Directory. By querying the X.500 OSI Directory, the gateway can identify where the actual host is resident.

For example, suppose that the FTP Server in the previous example ("ftp-service.netwrx1.com") is registered in the X.500 Directory with the following Distinguished Name:

CountryName	=	"US"
Organization	=	"Open Networks"
OrganizationalUnit	=	"Network Services"
CommonName	=	"netwrx1"
CommonName	=	"FTP service"

and the Presentation Address of the FTP-FTAM gateway. This approach, described in [RFC1279], would permit the following user interactions:

```
% ftam @c=US@o=Open Networks@ou=Network Services
    @cn=netwrx1@cn=FTP Service"
ftam> user mindel
ftam> pass *****
```

4.3. Summary of Usage

As shown in the discussions of the FTP-Initiated and FTAM-Initiated Gateway Services, the gateway user does not have access to the gateway filesystem; he merely makes use of the gateway logon procedure to specify the ultimate destination userid and password.

Two methods of interaction with the gateway were described. In the former, the user must:

1. Be aware that a gateway is required to reach the destination FTP or FTAM host.
2. Determine which gateway is most appropriate for their respective source-destination pair.
3. Explicitly connect to the gateway host prior to connecting to the destination host.

Needless to say, the exchange of files between FTP and FTAM hosts requires more effort than that required for the exchange of files between a pair of hosts utilizing the same file transfer protocol.

The latter, more transparent method does not necessarily require that the user determine which gateway is most appropriate for their respective source-destination pair. In fact, filestore service providers are registered using the address of a predetermined gateway. With this approach, the user:

1. Must be aware that a gateway is required to reach the destination FTP or FTAM host.
2. Need not determine which gateway is most appropriate to access their ultimate destination host.
3. Need not explicitly connect to the gateway prior to connecting to the destination FTP or FTAM host.

5. Gateway State Variables and Transitions

As described, the FTP-FTAM gateway provides two sets of services: FTP-Initiated and FTAM-Initiated. Each service has its own mutually exclusive set of state variables and transitions that deterministically define the actions of the gateway. Gateway support for these state variables and transitions is required.

For conciseness in this discussion, FTP-Initiated will be abbreviated with "FTP-I", and FTAM-Initiated will be abbreviated with "FTAM-I".

Concerning error conditions, if a connection is dropped when the gateway is in any state other than FTP-I:Initial-State or FTAM-I:Initial-State, then the gateway will issue a fatal error message to the host with the remaining connection, and then drop that connection. If the remaining host is an FTP Client, then the gateway will send an ABOR, QUIT, and 426 reply code (Connection closed,

transfer aborted). If it is an FTAM Initiator, then the gateway will send an F-P-ABORT with a <Diagnostic> value with identifier 1011 (Lower layer failure), as well as any known <Further Details>.

Other error conditions are not addressed in this discussion.

5.1. FTP-Initiated Gateway Service

The set of state variables for the FTP-Initiated Gateway service follow:

State Variable	State Definition
FTP-I:Initial-State	Initial state of FTP-Initiated Gateway service. Gateway is waiting for an FTP Client to issue a USER command in order to proceed with connection establishment with remote FTAM Responder. If SITE or ACCT commands are sent while waiting for USER command, save arguments for subsequent use.
FTP-I:Wait-for-PASS	Gateway has already received USER command from FTP Client, as well as userid and destination host DN. Gateway is waiting for the FTAM Responder logon password.
FTP-I:Wait-for-PAddress	Gateway has already received PASS command from FTP Client. Gateway is resolving the provided FTAM Responder's address to a Presentation Address. The provided address may be a Distinguished Name, User Friendly Name, or Domain Name. Resolution will typically be done using X.500 directory services.
FTP-I:Wait-for-Connection	Gateway has initiated a connection to the FTAM Responder and is waiting for notification as to whether or not the logon is successful.
FTP-I:Wait-for-ClientCmd	Connection exists between FTP Client and FTAM Responder. Gateway is waiting for next command or response from FTP

Client. Commands and responses are mapped as they are received.

FTP-I:Wait-for-ResponDrCmd Connection exists between FTP Client and FTAM Responder. Gateway is waiting for next command or response from FTAM Responder. Commands and responses are mapped as they are received.

Each of the possible state transitions is provided in the remainder of Section 5.1. For each state transition, the actions causing the transition are listed.

5.1.1. FTP-I:Initial-State --> FTP-I:Initial-State

1. Gateway receives SITE or ACCT command from FTP Client. SITE argument includes Distinguish Name of FTAM Responder.

5.1.2. FTP-I:Initial-State --> FTP-I:Wait-for-PASS

1. Gateway receives USER command from FTP Client. Arguments include Distinguished Name of FTAM Responder and userid on FTAM responder.

5.1.3. FTP-I:Wait-for-PASS --> FTP-I:Wait-for-PAddress

1. Gateway receives PASS command from FTP Client.

5.1.4. FTP-I:Wait-for-PAddress --> FTP-I:Wait-for-Connection

1. Gateway resolves received Distinguished Name, User Friendly Name, or Domain Name of FTAM Responder to OSI Presentation address.
2. Gateway sends F-INITIALIZE to FTAM Responder with Presentation Address in <Called Presentation Address>, userid in <Initiator Identity>, and password in <Filestore Password>.

5.1.5. FTP-I:Wait-for-Connection --> FTP-I:Wait-for-NextMapping

1. Gateway receives <State Result> of "Success" .
2. Gateway sends 230 reply code (User Logged In) to FTP Client.

5.1.6. FTP-I:Wait-for-ClientCmd --> FTP-I:Wait-for-ResponDrCmd

1. Gateway receives command or response from FTP Client and maps it to FTAM protocol, as defined in section 8.1.

5.1.7. FTP-I:Wait-for-ResponDrCmd --> FTP-I:Wait-for-ClientCmd

1. Gateway receives command or response from FTAM Responder and maps it to FTP protocol, as defined in section 8.1.

5.1.8. FTP-I:Wait-for-ClientCmd --> FTP-I:Wait-for-USER

1. Gateway receives QUIT command from FTP Client; maps QUIT as per Section 8.1.

5.2. FTAM-Initiated Gateway Service

The set of state variables for the FTAM-Initiated Gateway service follow:

State Variable	State Definition
FTAM-I:Initial-State	Initial state of FTAM-Initiated Gateway Service. Gateway is waiting for an FTAM Initiator to issue an F-INITIALIZE command in order to proceed with connection establishment with remote FTP Server.
FTAM-I:Wait-for-IPAddress	Gateway has already received F-INITIALIZE from FTAM Initiator. Gateway is resolving the provided FTP Server's address to an IP address. The provided address may be a Domain Name, Distinguished Name, or User Friendly Name.
FTAM-I:Wait-for-Connection	Gateway has initiated a connection to the FTP Server and is waiting for notification as to whether or not the logon is successful.
FTAM-I:Wait-for-InitiatrCmd	Connection exists between FTAM Initiator and FTP Server. Gateway is waiting for next command or response from FTAM Initiator. Commands and responses are mapped as they are received.

FTP-I:Wait-for-ServerCmd Connection exists between FTAM Initiator and FTP Server. Gateway is waiting for next command or response from FTP Server. Commands and responses are mapped as they are received.

Each of the possible state transitions is provided in the remainder of Section 5.2. For each state transition, the actions causing the transition are listed.

5.2.1. FTAM-I:Initial-State --> FTAM-I:Wait-for-IPAddress

1. Gateway receives F-INITIALIZE from FTAM Initiator. Domain Name of FTP Server is either in <Responding Presentation Address> or in the "@host" portion of the <Initiator Identity> parameter. The userid is in <Initiator Identity>, and password is in <Filestore Password> parameter.

5.2.2. FTAM-I:Wait-for-IPAddress --> FTAM-I:Wait-for-Connection

1. Gateway resolves received Domain Name, Distinguished Name, or User Friendly Name of FTP Server to IP address.
2. Gateway sends USER to FTP Server.
3. Gateway sends PASS to FTP Server.

5.2.3. FTAM-I:Wait-for-Connection --> FTAM-I:Wait-for-NextMapping

1. Gateway receives 230 reply code (User Logged In) from FTP Server.
2. Gateway sends <State Result> of "Success" to FTAM Initiator.

5.2.4 FTAM-I:Wait-for-InitiatrCmd --> FTAM-I:Wait-for-ServerCmd

1. Gateway receives command or response from FTAM Initiator and maps it to FTP protocol, as defined in section 8.2.

5.2.5. FTAM-I:Wait-for-ServerCmd --> FTAM-I:Wait-for-InitiatrCmd

1. Gateway receives command or response from FTP Server and maps it to FTAM protocol, as defined in section 8.2.

5.2.6. FTAM-I:Wait-for-InitiatrCmd --> FTAM-I:Wait-for-INITIALIZE

1. Gateway receives F-CLOSE primitive from FTAM Initiator; maps F-CLOSE as per Section 8.2.

6. Document Type Support

The set of FTAM document types supported by the FTP-FTAM gateway is a subset of the document types identified in the Stable Implementation Agreements for Open Systems Interconnection Protocols: Part 9 - FTAM Phase 2, produced by the March 1992 Open Systems Environment Implementors' Workshop [NIST92]. This subset was chosen for its equivalence to those document types supported by FTP. The set includes:

FTAM-1	"ISO FTAM Unstructured text file"
FTAM-3	"ISO FTAM Unstructured binary file"
NBS-9	"NBS-9 FTAM File directory file"

FTAM document types map to FTP document types as follows:

FTAM	<-->	FTP

FTAM-1	<-->	ASCII
FTAM-3	<-->	8 bit binary
NBS-9	<-->	Directory

Gateway support for FTAM-1 and FTAM-2 is required, whereas support for NBS-9 is recommended.

6.1. Notes on NBS-9

NBS-9 is optional in GOSIP versions 1 and 2 [NIST91]. NBS-9 will be superseded by its replacement when ISO/IEC ISP 10607-2 and ISO/IEC ISP 10607-2/Amendment 1 are published [NIST92].

For conformance to NBS-9, an FTAM Responder is only required to return the <Filename> file attribute, subject to local security and access control. All other requested attributes need not be returned.

Systems supporting the NBS-9 document type shall make available an NBS-9 document called 'DIRLIS'. This document can be used to obtain a listing of files and their associated attributes from a remote Filestore.

7. Functional Comparison of FTP and FTAM

A comprehensive comparison of the services offered by FTP and FTAM is beyond the scope of this specification. What follows is an analysis of several key points. Refer to [NIST 86a] and [ROSE90] for a more complete discourse on this topic.

FTAM is not a superset of FTP; each protocol has functions that only it performs. The set of FTAM functions is, however, larger than the set of FTP functions.

FTP combines file management and file transfer into one protocol engine, whereas FTAM separates management and transfer as they relate to files.

The file transfer services of both FTP and FTAM expect a reliable underlying end-to-end service. At a minimum, this service includes the capability to transfer entire files between remote hosts and to display remote filenames.

In addition to this basic file transfer service, FTAM supports the capability to: access a few records from a file server, create a network file system (similar to Sun's Network File System), handle printing and spooling, and access remote database records. FTP does not support these additional capabilities.

FTP uses TELNET services to set up a connection between the FTP Client and FTP Server. A three-digit reply code followed by explanatory text indicates the status of the preceding request and provides diagnostic information explaining each transaction.

FTAM relies on the Association Control Service Element (ACSE) to start and stop the network for network file interaction. Generally, the ASCE establishes the application association and related application context needed to support the FTAM protocol.

The FTAM protocol is modularized so as to keep the allowable number of actions in any particular state relatively small. There are many more possible sequences of FTP operations than possible sequences of FTAM operations [NIST86].

Because FTAM is more robust than FTP, FTAM allows greater flexibility for conveying information about files. FTAM deals only with aspects of application processes, and leaves data representation and data management facilities to other OSI service elements.

In contrast to the Client/Server model present in the FTP scheme, FTAM is based on the Initiator/Responder model. The key distinction

is that once the FTAM Initiator has established a connection with a remote host, either the Initiator or Responder can request services of the other. In the FTP realm, the Client both initiates a connection and requests all services.

The FTP Client knows the real properties of the remote host filesystem. FTAM, in contrast, embraces a conceptual model of a filesystem, labeled a virtual filestore model. The virtual filestore is a collection of files, each of which has a name that uniquely identifies it. Each file has a set of attributes, such as ownership information and contents, which is the data associated with the file. One file attribute is the <Contents Type> of the file, typically of value "FTAM-1", "FTAM-3", or "NBS-9". The FTAM Initiator only knows the properties of the corresponding Responder and virtual filestore, not the real properties of the filesystem on the remote host.

7.1. Loss of Functionality

As happens whenever two dissimilar protocols, or languages for that matter, are translated, some loss of functionality is inevitable. With reference to the FTP-FTAM gateway, several of the most blatant losses of functionality are:

1. Diagnostics passed between protocols may not be precisely translated.
2. The FTAM partial file (record) transfer may not be supported.
3. Some FTAM attributes are not supported by FTP.

The primary goal of the gateway protocol mappings are to minimize this loss of functionality. As this gateway specification and subsequent implementations evolve, means to partially overcome loss of functionality may become more obvious. For example, the gateway may be able to emulate file record transfers between FTAM Initiators and FTP Servers.

8. Mapping of Protocol Functions and Representations

The mappings presented are based upon the FTAM protocol implementation as defined in Stable Implementation Agreements for Open Systems Interconnection Protocols: Part 9 - FTAM Phase 2, produced by the March 1992 Open Systems Environment Implementors' Workshop [NIST92], and in [IS08571-1], [IS08571-2], [IS08571-3], [IS08571-4], and [IS08571-5]. The FTP protocol as defined in Request for Comments [RFC959]. The mappings are strongly influenced by the work of M. A. Wallace et. al. at NIST [NIST86] and John Scott

at MITRE [MITRE87].

A key goal of the mappings presented in this document is to minimize the loss of functionality between the two protocols. The specific approach taken to implement the mappings is left to the discretion of the gateway implementor. The focus of the protocol function and representation mappings is on non-error encumbered processing. The mapping of diagnostic and error messages is treated separately in section 9.

At a minimum, the FTAM implementation in the FTP-FTAM gateway support for Implementation Profiles T1 (Simple File Transfer) and M1 (Management), as defined in [NIST92], is required. These Implementation Profiles correspond to the A/111 and A/13 Profiles of Standards Promotion and Application Group in Europe, respectively [NIST92].

At a minimum, the gateway support for the following is required:

ASCII and 8 bit binary file types. It should also support FTP File Stream Mode.

The following FTAM document types: FTAM-1 (unstructured text file), FTAM-3 (unstructured binary file), and NBS-9 (set of directory entries).

POSIX file naming and organization conventions are assumed in these mappings; i.e., files in the systems are assumed to be organized in a hierarchical structure in which all of the non-terminal nodes are directories and all of the terminal nodes are any other type of file.

The following terminology is used in the mapping specifications:

argumentFTP Service Command argument, as used in [RFC959].
parameterFTAM Service Primitive parameters and attributes,
as enumerated in Tables 6, 50, and 51 of [ISO8571-3].

The following notation is used in the mapping specifications:

Arguments and parameters are enclosed in angle brackets; e.g., <Action Result>

Values of arguments and parameters are enclosed in quotation marks; e.g., "Success"

FTP Service Commands and FTAM Primitives are in uppercase; e.g., F-INITIALIZE

8.1. FTP-Initiated Gateway Service

The protocol mapping between FTP and FTAM may be one-to-zero (i.e., not mappable), one-to-one, or one-to-many.

The general steps taken by the FTP-FTAM gateway to provide the FTP-Initiated service are:

1. Accept an FTP Client request at the FTP Server side of the gateway service.
2. Map the request to the (set of) corresponding FTAM Initiator function(s).
3. Acting as an FTAM Initiator, send the FTAM Initiator function(s) to the FTAM Responder.
4. Accept information returned to the FTAM Initiator side of the gateway. This information originated at the FTAM Responder.
5. Map this returned information to the protocol form understood by the FTP Server side of the gateway.
6. Send this returned information from the FTP Server side of the gateway to the FTP Client.

For each FTP protocol function, the FTAM protocol functions required to map it are identified:

FTP	FTAM

ABOR	F-BEGIN-GROUP, F-CANCEL, F-CLOSE, F-DESELECT, F-END-GROUP
ACCT	F-INITIALIZE,
ALLO	none
APPE	F-BEGIN-GROUP, F-CLOSE, F-CREATE, F-DATA, F-DATA-END, F-DESELECT, F-END-GROUP, F-OPEN, F-READ-ATTRIBUTES, F-SELECT, F-TRANSFER-END, F-WRITE
CDUP	F-BEGIN-GROUP, F-DESELECT, F-END-GROUP, F-SELECT

CWD	F-BEGIN-GROUP, F-END-GROUP, F-DESELECT, F-SELECT
DELE	F-BEGIN-GROUP, F-DELETE, F-END-GROUP, F-SELECT
HELP	none
LIST	F-BEGIN-GROUP, F-CLOSE, F-DATA, F-DATA-END, F-DESELECT, F-END-GROUP, F-OPEN, F-READ, F-READ-ATTRIBUTES, F-SELECT, F-TRANSFER-END
MKD	none
MODE	none
NLST	F-BEGIN-GROUP, F-CLOSE, F-DATA, F-DATA-END, F-DESELECT, F-END-GROUP, F-OPEN, F-READ, F-SELECT, F-TRANSFER-END
NOOP	none
PASS	F-INITIALIZE
PASV	none
PORT	none
PWD	F-BEGIN-GROUP, F-DESELECT, F-END-GROUP, F-READ-ATTRIBUTES, F-SELECT
QUIT	F-P-ABORT or F-U-ABORT, F-TERMINATE
REIN	F-BEGIN-GROUP, F-CANCEL, F-CLOSE, F-DESELECT, F-END-GROUP
REST	F-CHECK, F-RESTART
RETR	F-BEGIN-GROUP, F-CLOSE, F-DATA, F-DATA-END, F-DESELECT, F-END-GROUP, F-OPEN, F-READ, F-SELECT, F-TRANSFER-END
RMD	none
RNFR	F-BEGIN-GROUP, F-DESELECT, F-END-GROUP, F-SELECT
RNT0	F-BEGIN-GROUP, F-CHANGE-ATTRIBUTES, F-DESELECT, F-END-GROUP, F-SELECT
SITE	F-INITIALIZE
SMNT	none

STAT	none
STOR	F-BEGIN-GROUP, F-CLOSE, F-CREATE, F-DATA, F-DATA-END, F-DESELECT, F-END-GROUP, F-OPEN, F-READ-ATTRIBUTES, F-SELECT, F-TRANSFER-END, F-WRITE
STOU	F-BEGIN-GROUP, F-CLOSE, F-CREATE, F-DATA, F-DATA-END, F-DESELECT, F-END-GROUP, F-OPEN, F-READ-ATTRIBUTES, F-SELECT, F-TRANSFER-END, F-WRITE
STRU	none
TYPE	none
USER	F-INITIALIZE

The remainder of this section presents detailed mapping procedures for each of the FTP protocol functions. Gateway support for these mappings is required.

8.1.1. ABOR

1. Send F-CANCEL to FTAM Responder.
2. Send the following grouped request to the FTAM Responder.
F-BEGIN-GROUP
F-CLOSE
F-DESELECT
F-END-GROUP
3. Translate FTAM Responder <Action Result> and <Diagnostic> parameters to equivalent FTP reply code(s) and send reply codes to FTP Client.
4. Translate FTP Client reply codes to equivalent FTAM <Action Result> and <Diagnostic> parameters and send parameters to FTAM Responder.

8.1.2. ACCT

1. Set <Account> parameter value for issuing F-INITIALIZE to FTAM Responder.
2. If <Called Presentation Address>, <Initiator Identity>, and <Filestore Password> parameters are available, attempt connection with FTAM Responder; Otherwise wait for additional ACCT commands.
3. Translate FTAM Responder <Action Result> and <Diagnostic> parameters to equivalent FTP reply code(s) and send reply codes to FTP Client.
4. Translate FTP Client reply codes to equivalent FTAM <Action Result> and <Diagnostic> parameters and send parameters to

FTAM Responder.

Note:

- a. The ACCT command will be effective with the next PASS command.

8.1.3. ALLO

1. Return a 200 reply code to FTP Client.

8.1.4. APPE

1. Save current pathname by appending saved CWD string with <pathname> argument. If no saved CWD string, proceed to step 12.
2. Send the following grouped request to FTAM Responder.
 - F-BEGIN-GROUP
 - F-SELECT
 - F-READ-ATTRIBUTES
 - Save <Contents Type> parameter value
 - F-DESELECT
 - F-END-GROUP
3. If the <Contents Type> parameter value returned with the F-READ-ATTRIBUTES has a value of "NBS-9", proceed to step 12.
4. Send the following grouped request to the FTAM responder.
 - F-BEGIN-GROUP
 - F-CREATE
 - Set the <Override> parameter in the F-CREATE to "Select Old File".
 - F-OPEN
 - F-END-GROUP
5. If the file existed, set the <Contents Type> parameter in the F-CREATE to match that returned by the F-READ-ATTRIBUTES.
6. If the file did not exist and no previous FTP TYPE "Image" command was issued, then set the <Contents Type> parameter to "FTAM-1"; Otherwise, set the <Contents Type> parameter to "FTAM-3".
7. Send F-WRITE, with <Bulk Data Transfer Specification, FADU Operation> parameter set to "File Extend", to FTAM Responder.
8. Loop reading data from FTP data connection, sending the data in F-DATA PDUs until end-of-file on the FTP connection.
9. Send F-DATA-END to FTAM Responder.
10. Send F-TRANSFER-END to FTAM Responder.
11. Send the following grouped request to the FTAM Responder.

F-BEGIN-GROUP

F-CLOSE

F-DESELECT

F-END-GROUP

12. Translate FTAM Responder <Action Result> and <Diagnostic> parameters to equivalent FTP reply code(s) and send reply code(s) to FTP Client.
13. Translate FTP Client reply codes to equivalent FTAM <Action Result> and <Diagnostic> parameters and send parameters to FTAM Responder.

Note:

- a. <pathname> argument is assumed to be a filename, relative to the currently saved CWD.
- b. CWD of the FTAM system must be defined prior to issuance of APPE.

8.1.5. CDUP

1. Determine parent directory from saved CWD string. If no saved CWD string, proceed to step 4.
2. Set <Contents Type> parameter to "NBS-9".
3. Send the following grouped request to FTAM Responder.
F-BEGIN-GROUP
F-SELECT
F-DESELECT
F-END-GROUP
4. Translate FTAM Responder <Action Result> and <Diagnostic> parameters to equivalent FTP reply code(s) and send reply code(s) to FTP Client.
5. Translate FTP Client reply codes to equivalent FTAM <Action Result> and <Diagnostic> parameters and send parameters to FTAM Responder.

Note:

- a. A POSIX file organization is assumed; i.e., files in the systems are organized in a hierarchical structure in which all of the non-terminal nodes are directories and all of the terminal nodes are any other type of file.
- b. If the parent directory does not exist, the current working directory remains unchanged.
- c. CWD of the FTAM system must be defined prior to issuance of CDUP.

8.1.6. CWD

1. Save <pathname> argument as CWD string.
2. Set <Contents Type> parameter to "NBS-9".

3. Send the following grouped request to FTAM Responder.
F-BEGIN-GROUP
F-SELECT
F-DESELECT
F-END-GROUP
4. Translate FTAM Responder <Action Result> and <Diagnostic> parameters to equivalent FTP reply code(s) and send reply code(s) to FTP Client.
5. Translate FTP Client reply codes to equivalent FTAM <Action Result> and <Diagnostic> parameters and send parameters to FTAM Responder.

Note:

- a. The <pathname> argument is assumed to be an absolute directory specification.
- b. If the specified directory does not exist, the current working directory remains unchanged.
- c. Saved CWD string is used in other FTP-to-FTAM mappings, such as APPE.

8.1.7. DELE

1. Save current pathname by appending saved CWD string with <pathname> argument. If no saved CWD string, proceed to step 3.
2. Send the following grouped request to FTAM Responder.
F-BEGIN-GROUP
F-SELECT
F-DELETE
F-END-GROUP
3. Translate FTAM Responder <Action Result> and <Diagnostic> parameters to equivalent FTP reply code(s) and send reply code(s) to FTP Client.
4. Translate FTP Client reply codes to equivalent FTAM parameters and send parameters to FTAM Responder.

Note:

- a. <pathname> argument is assumed to be a filename, relative to the currently saved CWD.
- b. CWD of the FTAM system must be defined prior to issuance of DELE.

8.1.8. HELP

1. If no <string> argument is provided, send helpful information about the implementation of the gateway to the FTP Client. If an argument is provided, send more specific information.

2. Return the FTP reply code 214 to the FTP Client.

8.1.9. LIST

1. If <pathname> argument is provided, proceed to step 3.
2. Save current pathname by appending saved CWD string with <pathname> argument; If no saved CWD string, proceed to step 11.
3. Send the following grouped request to the FTAM Responder.
F-BEGIN-GROUP
F-SELECT
F-READ-ATTRIBUTES
 Save <Filename>, <Contents Type>, <Data/Time of Last Modification>, and <Filesize> parameters
F-DESELECT
F-END-GROUP
4. If the <Contents Type> parameter of the F-READ-ATTRIBUTES is not "NBS-9", then return the <Filename>, <Contents Type>, <Date/Time of Last Modification>, and <Filesize> parameter values, obtained with the previous F-READ-ATTRIBUTES, to the FTP data connection; and proceed to step 8.
5. Send the following grouped request to the FTAM Responder.
F-BEGIN-GROUP
F-SELECT
F-OPEN
F-END-GROUP
6. Send F-READ to FTAM Responder.
7. Loop reading F-DATA until F-DATA-END. As data is received, write the <Filename>, <Permitted Actions>, <Contents Type>, and <Date/Time of Last Modification> parameter values from the PDU to the FTP data connection.
8. Send F-TRANSFER-END to FTAM Responder.
9. Send the following grouped request to the FTAM responder.
F-BEGIN-GROUP
F-CLOSE
F-DESELECT
F-END-GROUP
10. Translate FTAM Responder <Action Result> and <Diagnostic> parameters to equivalent FTP reply code(s) and send reply code(s) to FTP Client.
11. Translate FTP Client reply codes to equivalent FTAM <Action Result> and <Diagnostic> parameters and send parameters to FTAM Responder.

Note:

- a. Assume the <pathname> argument is relative to the saved CWD, whether filename or directory specification.

- b. CWD of the FTAM system must be defined prior to issuance of LIST.
- c. Transfers over data connection should be in ASCII.
- e. If list of files with full directory/file specification is received from FTAM Responder, then gateway should parse list to strip off directory portion.

8.1.10. MKD

1. Return a 502 reply code (Command not implemented) to the FTP Client.

Note:

- a. As indicated in the NIST Stable Implementation Agreements for FTAM [NIST92], creation or deletion of NBS-9 files is outside the scope of the agreements.

8.1.11. MODE

1. If <argument> is "Stream", return 200 reply code to FTP Client; Otherwise return a 504 reply code (Command not implemented for that parameter).

8.1.12. NLST

1. If <pathname> argument is provided, use <pathname> argument as <Filename> parameter value in F-SELECT issued in step 3.
2. If no argument is provided, use saved CWD value as <Filename> parameter value in F-SELECT issued in step 3; If no CWD string is saved and no argument is provided, proceed to step 9.
3. Set <Contents Type> parameter to "NBS-9".
4. Send the following grouped request to the FTAM Responder.
 - F-BEGIN-GROUP
 - F-SELECT
 - F-OPEN
 - F-END-GROUP
5. Send F-READ to FTAM Responder.
6. Loop reading F-DATA until F-DATA-END. As data is received, write the filenames and other useful information from the PDU to the FTP data connection.
7. Send F-TRANSFER-END to FTAM Responder.
8. Send the following grouped request to the FTAM responder.
 - F-BEGIN-GROUP
 - F-CLOSE
 - F-DESELECT
 - F-END-GROUP
9. Translate FTAM Responder <Action Result> and <Diagnostic>

- parameters to equivalent FTP reply code(s) and send reply code(s) to FTP Client.
10. Translate FTP Client reply codes to equivalent FTAM <Action Result> and <Diagnostic> parameters and send parameters to FTAM Responder.

Note:

- a. As per RFC 959 (FTP), the NLST <pathname> argument is a directory.
- b. Assume the argument is relative to the saved CWD, whether filename or directory specification.
- c. CWD of the FTAM system must be defined prior to issuance of NLST.
- d. Transfers over data connection should be in ASCII.
- e. Gateway should parse full directory/file specifications received from FTAM Responder to strip off directory portion. This is required to support the "FTP multiple get" function that pipes NLST output to the STOR command.

8.1.13. N00P

1. Return a 200 reply code to FTP Client.

8.1.14. PASS

1. Set <Filestore Password> parameter for F-INITIALIZE.
2. If <Called Presentation Address>, <User Identity>, and <Filestore Password> are available, issue F- INITIALIZE to FTAM Responder.
3. Translate FTAM Responder <Action Result> and <Diagnostic> parameters to equivalent FTP reply code(s) and send reply code(s) to FTP Client.
4. Translate FTP Client reply codes to equivalent FTAM <Action Result> and <Diagnostic> parameters and send parameters to FTAM Responder.

8.1.15. PASV

1. Wait for data transfer on default data port or data port specified by PORT command.
2. Return a 200 reply code to FTP Client.

8.1.16. PORT

1. Return a 200 reply code to FTP Client.

8.1.17. PWD

1. If there is a saved CWD string, return it to the FTP client and proceed to step 4.
2. Set <Contents Type> attribute to "NBS-9".
3. Send the following grouped request to FTAM Responder.
F-BEGIN-GROUP
F-SELECT
F-READ-ATTRIBUTES
F-DESELECT
F-END-GROUP
4. Return the current directory name to the FTP client.
5. Translate FTAM Responder <Action Result> and <Diagnostic> parameters to equivalent FTP reply code(s) and send reply code(s) to FTP Client.
6. Translate FTP Client reply codes to equivalent FTAM <Action Result> and <Diagnostic> parameters and send parameters to FTAM Responder.

8.1.18. QUIT

1. If user is not logged in, proceed to step 5.
2. If file transfer is in progress, send F-P-ABORT or F-U-ABORT to FTAM Responder.
3. If file transfer is not in progress, send and F-TERMINATE to FTAM Responder.
4. Return charge information to FTP Client.
5. Translate FTAM Responder <Action Result> and <Diagnostic> parameters to equivalent FTP reply code(s) and send reply code(s) to FTP Client.
6. Translate FTP Client reply codes to equivalent FTAM <Action Result> and <Diagnostic> parameters and send parameters to FTAM Responder.

8.1.19. REIN

1. Flush all I/O and account information.
2. Allow all transfers in progress to be completed.
3. Set all parameters to default values.
4. Send F-CANCEL to FTAM Responder.
5. Send the following grouped request to FTAM Responder.
F-BEGIN-GROUP
F-CLOSE
F-DESELECT
F-END-GROUP
6. Leave the control connection open.
7. Translate FTAM Responder <Action Result> and <Diagnostic> parameters to equivalent FTP reply code(s) and send reply

- code(s) to FTP Client.
8. Translate FTP Client reply codes to equivalent FTAM <Action Result> and <Diagnostic> parameters and send parameters to FTAM Responder.

Note:

- a. Typically followed by a USER command.

8.1.20. REST

1. Send F-CHECK to FTAM Responder.
2. Send F-RESTART to FTAM Responder.
3. Translate FTAM Responder <Action Result> and <Diagnostic> parameters to equivalent FTP reply code(s) and send reply code(s) to FTP Client.
4. Translate FTP Client reply codes to equivalent FTAM <Action Result> and <Diagnostic> parameters and send parameters to FTAM Responder.

Notes:

- a. Will only have affect on FTAM Responder if the restart functional unit is negotiated on F-INITIALIZE.
- b. Refer to ISO 8571-3 for additional subtleties of FTAM checkpoint and restart.

8.1.21. RETR

1. Save current pathname by appending saved CWD string with <pathname> argument. If no saved CWD string, proceed to step 9.
2. Set <Contents Type> parameter to appropriate type of file.
3. Send the following grouped request to the FTAM Responder.
F-BEGIN-GROUP
F-SELECT
F-OPEN
F-END-GROUP
4. If file does not exist, proceed to step 9.
5. Send F-READ to FTAM Responder.
6. Loop reading F-DATA until F-DATA-END. As data is received, write it to the FTP data connection.
7. Send F-TRANSFER-END to FTAM Responder.
8. Send the following grouped request to the FTAM Responder.
F-BEGIN-GROUP
F-CLOSE
F-DESELECT
F-END-GROUP
9. Translate FTAM Responder <Action Result> and <Diagnostic> parameters to equivalent FTP reply code(s) and send reply

- code(s) to FTP Client.
10. Translate FTP Client reply codes to equivalent FTAM <Action Result> and <Diagnostic> parameters and send parameters to FTAM Responder.

Note:

- a. <pathname> argument is assumed to be a filename, relative to the currently saved CWD.
- b. CWD of the FTAM system must be defined prior to issuance of RETR.

8.1.22. RMD

1. Return a 502 reply code (Command not implemented) to the FTP Client.

Note:

- a. As indicated in the NIST Stable Implementation Agreements for FTAM [NIST92], creation or deletion of NBS-9 files is outside the scope of the agreements.

8.1.23. RNFR

1. Save current pathname by appending saved CWD string with <pathname> argument. If no saved CWD string, proceed to step 3.
2. Send the following grouped request to the FTAM Responder.
F-BEGIN-GROUP
F-SELECT
 Get <Filename> parameter value from RNFR <pathname> argument.
F-DESELECT
F-END-GROUP
3. Translate FTAM Responder <Action Result> and <Diagnostic> parameters to equivalent FTP reply code(s) and send reply code(s) to FTP Client.
4. Translate FTP Client reply codes to equivalent FTAM <Action Result> and <Diagnostic> parameters and send parameters to FTAM Responder.

Note:

- a. <pathname> argument is assumed to be a filename, relative to the currently saved CWD.
- b. Together with RNT0, this command causes a file to be renamed.
- c. CWD of the FTAM system must be defined prior to issuance of RNFR.

8.1.24. RNT0

1. Save current pathname by appending saved CWD string with <pathname> argument. If no saved CWD string, proceed to step 3.
2. Send the following grouped request to the FTAM Responder.
F-BEGIN-GROUP
F-SELECT
F-CHANGE-ATTRIBUTES
 Get <Filename> parameter from arguments provided by RNT0 and previous RNFR.
F-DESELECT
F-END-GROUP
3. Translate FTAM Responder <Action Result> and <Diagnostic> parameters to equivalent FTP reply code(s) and send reply code(s) to FTP Client.
4. Translate FTP Client reply codes to equivalent FTAM <Action Result> and <Diagnostic> parameters and send parameters to FTAM Responder.

Note:

- a. <pathname> argument is assumed to be a filename, relative to the currently saved CWD.
- b. Together with RNFR, this command causes a file to be renamed.
- c. CWD of the FTAM system must be defined prior to issuance of RNT0.

8.1.25. SITE

1. Save the specified destination address information.
2. Set the <Called Presentation Address> parameter value equal to the <string> argument. This parameter will be used when the F-INITIALIZE is sent to the FTAM Responder.
3. Translate FTAM Responder <Action Result> and <Diagnostic> parameters to equivalent FTP reply code(s) and send reply code(s) to FTP Client.
4. Translate FTP Client reply codes to equivalent FTAM <Action Result> and <Diagnostic> parameters and send parameters to FTAM Responder.

Note:

- a. The <string> argument to the SITE command may include a Distinguished Name or a User Friendly Name.

8.1.26. SMNT

1. Return a 502 reply code to FTP Client.

Note:

- a. Argument is ignored.

8.1.27. STAT

1. Provide the gateway session status to the FTP Client.
2. Return a 211 reply code to FTP Client.

Note:

- a. Argument is ignored.

8.1.28. STOR

1. Save current pathname by appending saved CWD string with <pathname> argument. If no saved CWD string, proceed to step 11.
2. Send the following grouped request to FTAM Responder.
F-BEGIN-GROUP
F-SELECT
F-READ-ATTRIBUTES
 Save <Contents Type> parameter value
F-DESELECT
F-END-GROUP
3. If the <Contents Type> parameter returned with the F-READ-ATTRIBUTES indicates a directory, proceed to step 11.
4. Send the following grouped request to the FTAM responder.
F-BEGIN-GROUP
F-CREATE
 Set the <Override> parameter in the F-CREATE to "Delete and create with new attributes."
F-OPEN
F-END-GROUP
5. If the file existed, set the <Contents Type> parameter in the F-CREATE to match the F-READ-ATTRIBUTES. If the file did not exist, set the <Contents Type> parameter to "FTAM-1". If TYPE "Image" was previously requested, set the <Contents Type> parameter to "FTAM-3".
6. Send F-WRITE, with <Bulk Data Transfer Specification, FADU Operation> parameter set to "File Extend", to FTAM Responder.
7. Loop reading data from FTP data connection, sending the data in F-DATA PDUs until end-of-file on the FTP connection.
8. Send F-DATA-END to FTAM Responder.
9. Send F-TRANSFER-END to FTAM Responder.

10. Send the following grouped request to the FTAM Responder.
F-BEGIN-GROUP
F-CLOSE
F-DESELECT
F-END-GROUP
11. Translate FTAM Responder <Action Result> and <Diagnostic> parameters to equivalent FTP reply code(s) and send reply code(s) to FTP Client.
12. Translate FTP Client reply codes to equivalent FTAM <Action Result> and <Diagnostic> parameters and send parameters to FTAM Responder.

Note:

- a. <pathname> argument is assumed to be a filename, relative to the currently saved CWD.
- b. CWD of the FTAM system must be defined prior to issuance of STOR.

8.1.29. STOU

1. Save current pathname by appending saved CWD string with <pathname> argument. If no saved CWD string, proceed to step 11.
2. Send the following grouped request to FTAM Responder.
F-BEGIN-GROUP
F-SELECT
F-READ-ATTRIBUTES
 Save <Contents Type> parameter value
F-DESELECT
F-END-GROUP
3. If the file already exists, proceed to step 12.
4. If the <Contents Type> parameter returned with the F-READ-ATTRIBUTES indicates a directory, proceed to step 11.
5. Send the following grouped request to the FTAM responder.
F-BEGIN-GROUP
F-CREATE
 Set the <Override> parameter in the F-CREATE to "Delete and create with new attributes."
F-OPEN
F-END-GROUP
6. If the file existed, set the <Contents Type> parameter in the F-CREATE to match the F-READ-ATTRIBUTES. If the file did not exist, set the <Contents Type> parameter to "FTAM-1". If TYPE "Image" was previously requested, set the <Contents Type> parameter to "FTAM-3".
7. Send F-WRITE, with <Bulk Data Transfer Specification, FADU Operation> parameter set to "File Extend", to FTAM Responder.
8. Loop reading data from FTP data connection, sending the

- data in F-DATA PDUs until end-of-file on the FTP connection.
9. Send F-DATA-END to FTAM Responder.
 10. Send F-TRANSFER-END to FTAM Responder.
 11. Send the following grouped request to the FTAM Responder.
F-BEGIN-GROUP
F-CLOSE
F-DESELECT
F-END-GROUP
 12. Translate FTAM Responder <Action Result> and <Diagnostic> parameters to equivalent FTP reply code(s) and send reply code(s) to FTP Client.
 13. Translate FTP Client reply codes to equivalent FTAM <Action Result> and <Diagnostic> parameters and send parameters to FTAM Responder.

Note:

- a. <pathname> argument is assumed to be a filename, relative to the currently saved CWD.
- b. Same as STOR, except the name of the created file must be unique in that directory.
- c. CWD of the FTAM system must be defined prior to issuance of STOU.

8.1.30. STRU

1. If <structure code> argument is not "File", return 504 reply code to FTP Client; Otherwise return 200 reply code to FTP Client.

8.1.31. SYST

1. Return 502 reply code to FTP client.

8.1.32. TYPE

1. If no <type code> argument is provided, set <Contents Type> parameter equal to "FTAM-1".
2. If argument is provided, and equal to "ASCII", set <Contents Type> parameter to "FTAM-1".
3. If argument is provided, and equal to "Image", set <Contents Type> parameter to "FTAM-3".
4. Translate FTAM Responder <Action Result> and <Diagnostic> parameters to equivalent FTP reply code(s) and send reply code(s) to FTP Client.
5. Translate FTP Client reply codes to equivalent FTAM <Action Result> and <Diagnostic> parameters and send parameters to FTAM Responder.

Note:

- a. Default to ASCII if no <type code> argument is provided.

8.1.33. USER

1. Set <Initiator Identity> parameter for issuing F-INITIALIZE to FTAM Responder.
2. If the destination address was specified in the Domain Name used to attach to the gateway, use it to set the value of the <Called Presentation Address> parameter of the to-be-issued F-INITIALIZE command.
3. If the destination address is not known, check if it was specified in a previously issued SITE command. If available, set <Called Presentation Address> parameter for issuing F-INITIALIZE to FTAM Responder.
4. If the destination address is still not available, check if it is encoded in the user identity (e.g., user@host). If encoded, set <Called Presentation Address> parameter for issuing F-INITIALIZE to FTAM Responder using the "host" portion.
5. If no destination address is available, proceed to step 7.
6. Prompt user for password.
7. Translate FTAM Responder <Action Result> and <Diagnostic> parameters to equivalent FTP reply code(s) and send reply code(s) to FTP Client.
8. Translate FTP Client reply codes to equivalent FTAM <Action Result> and <Diagnostic> parameters and send parameters to FTAM Responder.

Note:

- a. A USER command should be acceptable in any state.
- b. Multiple mechanisms are available for specifying the destination address: 1) Domain Name used in connecting to gateway (see section 4, Use of Gateway Services); 2) SITE command argument; and 3) user@host format.

8.2. FTAM-Initiated Gateway Service

The protocol mapping between FTP and FTAM may be one-to-zero (i.e., not mappable), one-to-one, or one-to-many.

The general steps taken by the FTP-FTAM gateway to provide the FTAM-Initiated service are:

1. Accept an FTAM Initiator request at the FTAM Responder side of the gateway.
2. Map the request to the (set of) corresponding FTP Client

function(s).

3. Acting as an FTP Client, send the FTP Client function(s) to the FTP Server.
4. Accept information returned to the FTP Client side of the gateway. This information originated at the FTP Server.
5. Map this returned information to a form understood by the FTAM Responder side of the gateway.
6. Send this returned information from the FTAM Responder side of the gateway to the FTAM Initiator.

For each FTAM protocol function, the FTP protocol functions required to map it are identified:

FTAM	FTP

F-BEGIN-GROUP	none
F-CANCEL	ABOR
F-CHANGE-ATTRIBUTE	RNFR, RNT0
F-CHECK	none
F-CLOSE	none
F-CREATE	STOR
F-DATA	ALLO, STOR or RETR or APPE
F-DATA-END	none
F-DELETE	DELE
F-DESELECT	none
F-END-GROUP	STAT
F-ERASE	DELE
F-INITIALIZE	ACCT, PASS, USER
F-LOCATE	none

F-OPEN	MODE, STRU, TYPE
F-READ	MODE, NLST, RETR, TYPE
F-READ-ATTRIBUTE	LIST
F-RECOVER	REST
F-RESTART	ABOR, REST
F-SELECT	LIST
F-TERMINATE	QUIT
F-TRANSFER	none
F-P-ABORT	QUIT
F-U-ABORT	QUIT
F-WRITE	APPE or STOR, NOOP

The remainder of this section presents detailed mapping procedures for each of the FTAM protocol functions. Where appropriate, each FTAM service primitive is followed by those parameters that are relevant to the mapping. Gateway support for these mappings is required.

8.2.1. F-BEGIN-GROUP REQ

1. Send F-BEGIN-GROUP RESP PDU to FTAM Initiator signifying that processes are available to handle concatenated requests.

8.2.2. F-CANCEL REQ

1. Close FTP data connection.
2. Send ABOR to FTP Server.
3. Translate FTP Server reply code to equivalent FTAM Responder action and diagnostic parameters and send parameters to FTAM Initiator via F-CANCEL RESP PDU.
4. Translate FTAM Initiator action and diagnostic parameters to equivalent FTP reply codes and send reply codes to FTP Server.

Note:

- a. F-U-ABORT REQ is a viable alternative to F-CANCEL REQ.
- b. Note that since ABOR is not implemented by all FTP Servers,

the remote file may be corrupted, though accessible.

8.2.3. F-CHANGE-ATTRIBUTE REQ

1. Get original filename from <Filename> parameter and send it with an RNFR to the FTP Server.
2. Get new filename from <Filename> parameter and send it with an RNT0 to the FTP Server.
3. Translate FTP Server reply code to equivalent FTAM Responder action and diagnostic parameters and send parameters to FTAM Initiator via F-CHANGE-ATTRIBUTE RESP PDU.
4. Translate FTAM Initiator action and diagnostic parameters to equivalent FTP reply codes and send reply codes to FTP Server.

Note:

- a. Allow for processing an arbitrary number attributes at one time.
- b. Allow for responses of "Attribute currently unavailable for change" and "Attribute not currently supported".
- c. At a minimum, support the <Filename>, <Permitted Actions>, and <Contents Type> parameters.

8.2.4. F-CHECK REQ

1. Send an F-CHECK RESP PDU to the FTAM Initiator.

8.2.5. F-CLOSE REQ

1. Send F-CLOSE RESP PDU , with <Action Result> parameter value of "Success", to FTAM Initiator.

Note:

- a. If an error had occurred during transfer, it would have been noted before the F-CLOSE REQ.

8.2.6. F-CREATE REQ

1. Send STOR and zero data bytes to FTP Server.
2. Translate FTP Server reply code to equivalent FTAM Responder <Action Result> and <Diagnostic> parameters and send parameters to FTAM Initiator.
3. Translate FTAM Initiator <Action Result> and <Diagnostic> parameters to equivalent FTP reply codes and send reply codes to FTP Server.

8.2.7. F-DATA PDU

1. If necessary, send ALLO command to FTP Server.
2. Depending on whether reading or writing, send STOR, RETR, or APPE command to FTP Server.
3. Translate FTP Server reply code to equivalent FTAM Responder <Action Result> and <Diagnostic> parameters and send parameters to FTAM Initiator.
4. Translate FTAM Initiator <Action Result> and <Diagnostic> parameters to equivalent FTP reply codes and send reply codes to FTP Server.

Note:

- a. The use of an FTP command may be unnecessary. Sending the data on the data connection may be adequate.

8.2.8. F-DATA-END REQ

1. Close the data connection.
2. Save mandatory Diagnostic parameter for later use.
3. Translate FTP Server reply code to equivalent FTAM Responder <Action Result> and <Diagnostic> parameters and send parameters to FTAM Initiator.
4. Translate FTAM Initiator <Action Result> and <Diagnostic> parameters to equivalent FTP reply codes and send reply codes to FTP Server.

8.2.9. F-DELETE REQ

1. Send DELE to FTP server.
2. Translate FTP Server reply code to equivalent FTAM Responder <Action Result> and <Diagnostic> parameters and send parameters to FTAM Initiator via F-DELETE RESP PDU.
3. Translate FTAM Initiator <Action Result> and <Diagnostic> parameters to equivalent FTP reply codes and send reply codes to FTP Server.

8.2.10. F-DESELECT REQ

1. Return F-DESELECT RESP PDU, with <Action Result> parameter value of "Success", to FTAM Initiator.

8.2.11. F-END-GROUP REQ

1. Send STAT command sequence to FTP Server.
2. Translate FTP Server reply code to equivalent FTAM Responder <Action Result> and <Diagnostic> parameters and send parameters to FTAM Initiator via F-END

GROUP RESP.

3. Translate FTAM Initiator <Action Result> and <Diagnostic> parameters to equivalent FTP reply codes and send reply codes to FTP Server.

8.2.12. F-ERASE REQ

1. Send DELE to FTP Server.
2. Translate FTP Server reply code to equivalent FTAM Responder <Action Result> and <Diagnostic> parameters and send parameters to FTAM Initiator via F-ERASE RESP PDU.
3. Translate FTAM Initiator <Action Result> and <Diagnostic> parameters to equivalent FTP reply codes and send reply codes to FTP Server.

8.2.13. F-INITIALIZE REQ

1. Establish initial area for activity attributes.
2. Save <Responding Presentation Address>, <Initiator Identity>, and <Filestore Password> parameter values received from FTAM Initiator.
3. If the destination address was specified in the Distinguished Name (or User Friendly Name) used to attach to the gateway, save it as the ultimate destination address.
4. If the ultimate destination address is not yet known, look at the "@host" portion of the <Initiator Identity> parameter for the ultimate destination parameter.
5. If the ultimate destination address is still not known, check if it is available in the <Responding Presentation Address> parameter.
6. Get userid from <Initiator Identity> and send it with USER command to FTP Server.
7. Get password from <Filestore Password> and send it with PASS command to FTP Server.
8. If necessary, send ACCT command to FTP Server.
9. Negotiate acceptance of mandatory functional units, service classes, service types, presentation contexts, and attribute groups.
10. Accept context management functional unit passed by Presentation service provider.
11. Translate FTP Server reply code to equivalent FTAM Responder <Action Result> and <Diagnostic> parameters and send parameters to FTAM Initiator via F-INIT RESP PDU.
12. Translate FTAM Initiator <Action Result> and <Diagnostic> parameters to equivalent FTP reply codes and send reply codes to FTP Server.

Note:

- a. Multiple mechanisms are available for specifying the destination address: 1) Distinguished Name, or User Friendly Name, used in connecting to the gateway (see section 4, Use of Gateway Services); 2) user@host format; and 3) Inclusion as <Responding Presentation Address> parameter value.

8.2.14. F-LOCATE REQ**Note:**

- a. Not supported since FTAM-1 and FTAM-3 don't support this primitive.

8.2.15. F-OPEN REQ

1. Get <Contents Type> and <Processing Mode> parameter values from FTAM Initiator.
2. Send TYPE command to FTP Server.
3. Send MODE command to FTP Server.
4. Send STRU command to FTP Server.
5. Translate FTP Server reply code to equivalent FTAM Responder <Action Result> and <Diagnostic> parameters and send parameters to FTAM Initiator via F-OPEN RESP PDU.
6. Translate FTAM Initiator <Action Result> and <Diagnostic> parameters to equivalent FTP reply codes and send reply codes to FTP Server.

Note:

- a. Establishes definite value for presentation context name parameter for this data transfer.
- b. Assumes that the <Requested Access> parameter is permitted.

8.2.16. F-READ REQ

1. If requested file type and file mode are different than current settings, send TYPE and MODE to FTP Server.
2. If <Contents Type> is FTAM-1 or FTAM-3, then send RETR to FTP Server.
3. If <Contents Type> is "NBS-9", then send NLST to FTP Server.
4. If reply code from FTP Server is 1xx, open FTP data connection and loop until End-of-File is read on FTP data connection. Inside loop, read block from FTP data connection, format FTAM DATA PDU, and send FTAM PDU to FTAM Initiator. At End-of-File on FTP data connection, send F-DATA-END and return.

5. If reply code from FTP Server is not 1xx, send F-CANCEL REQ to FTAM Initiator.
6. Translate FTP Server reply code to equivalent FTAM Responder <Action Result> and <Diagnostic> parameters and send parameters to FTAM Initiator via F-READ RESP PDU.
7. Translate FTAM Initiator <Action Result> and <Diagnostic> parameters to equivalent FTP reply codes and send reply codes to FTP Server.

Note:

- a. To send NLST response, TYPE must be ASCII.

8.2.17. F-READ-ATTRIBUTE REQ

1. Send LIST to FTP Server.
2. Translate returned information into the <Filename>, <Contents Type>, and <Permitted Actions> parameter values and return them to the FTAM Initiator.
3. Translate FTP Server reply code to equivalent FTAM Responder <Action Result> and <Diagnostic> parameters and send parameters to FTAM Initiator via F-READ-ATTRIBUTE RESP PDU.
4. Translate FTAM Initiator <Action Result> and <Diagnostic> parameters to equivalent FTP reply codes and send reply codes to FTP Server.

8.2.18. F-RECOVER REQ

1. Send REST command to FTP Server.
2. Translate FTP Server reply code to equivalent FTAM Responder <Action Result> and <Diagnostic> parameters and send parameters to FTAM Initiator.
3. Translate FTAM Initiator <Action Result> and <Diagnostic> parameters to equivalent FTP reply codes and send reply codes to FTP Server.

Note:

- a. Regime recovery is only possible if the <Recovery Functional Unit> parameter was negotiated previously by an F-INITIALIZE.

8.2.19. F-RESTART REQ

1. To interrupt any bulk data transfer in progress, send ABOR to FTP Server.
2. To negotiate the point at which data transfer is to be restarted, get <Checkpoint Identifier> parameter from FTAM Initiator and send it with REST to FTP Server.

3. Translate FTP Server reply code to equivalent FTAM Responder <Action Result> and <Diagnostic> parameters and send parameters to FTAM Initiator via F-RESTART RESP PDU.
4. Translate FTAM Initiator <Action Result> and <Diagnostic> parameters to equivalent FTP reply codes and send reply codes to FTP Server.

8.2.20. F-SELECT REQ

1. Get <Filename> parameter and send with LIST command to FTP Server to determine whether or not the file exists.
2. If file exists, compare the POSIX file access rights with the <Requested Access> parameter sent by the FTAM Initiator. If the access rights match, return <Action Result> parameter value of "Success", otherwise return <Action Result> parameter value of "Failure".
3. Translate FTP Server reply code to equivalent FTAM Responder <Action Result> and <Diagnostic> parameters and send parameters to FTAM Initiator via F-SELECT RESP PDU.
4. Translate FTAM Initiator <Action Result> and <Diagnostic> parameters to equivalent FTP reply codes and send reply codes to FTP Server.

Note:

- a. The specified file is binary/text file if one record is received or is a directory file if multiple records are received.

8.2.21. F-TERMINATE REQ

1. Send QUIT to FTP Server.
2. Translate FTP Server reply code to equivalent FTAM Responder <Action Result> and <Diagnostic> parameters and send parameters to FTAM Initiator via F-TERMINATE RESP PDU.
3. Translate FTAM Initiator <Action Result> and <Diagnostic> parameters to equivalent FTP reply codes and send reply codes to FTP Server.

8.2.22. F-TRANSFER-END

1. Get <Action Result> parameter value from last F-DATA-END and return it to FTAM Initiator as <Action Result> parameter of this F-TRANSFER-END.

8.2.23. F-P-ABORT REQ

1. Send QUIT to FTP Server.
2. Return <Action Result> parameter value of "Permanent Error"

- to FTAM Initiator.
3. Translate FTP Server reply code to equivalent FTAM Responder <Action Result> and <Diagnostic> parameters and send parameters to FTAM Initiator.
4. Translate FTAM Initiator <Action Result> and <Diagnostic> parameters to equivalent FTP reply codes and send reply codes to FTP Server.

8.2.24. F-U-ABORT REQ

1. Send QUIT to FTP Server.
2. Return <Action Result> parameter value of "Permanent Error" to FTAM Initiator.
3. Translate FTP Server reply code to equivalent FTAM Responder <Action Result> and <Diagnostic> parameters and send parameters to FTAM Initiator.
4. Translate FTAM Initiator <Action Result> and <Diagnostic> parameters to equivalent FTP reply codes and send reply codes to FTP Server.

8.3. F-WRITE REQ

1. Save bulk transfer specification parameter from PDU.
2. Send NOOP to FTP Server to receive status information.
3. If the <Bulk Data Transfer Specification, FADU Operation> parameter has a value of "File Extend", then send an APPE to the FTP Server, otherwise send a STOR to the FTP Server.
4. If reply code from FTP Server is 200, then accept FTP data connection; otherwise send F-CANCEL REQ to FTAM Initiator.
5. Translate FTP Server reply code to equivalent FTAM Responder <Action Result> and <Diagnostic> parameters and send parameters to FTAM Initiator.
6. Translate FTAM Initiator <Action Result> and <Diagnostic> parameters to equivalent FTP reply codes and send reply codes to FTP Server.

9. Mapping between FTP Reply Codes and FTAM Parameters

The focus of the protocol function and representation mappings, presented in the previous sections, is on non-error encumbered processing. Though appropriate responses are designated in many cases, it is intended that a more thorough use of responses will be incorporated into gateway implementations.

The purpose of this section is to provide a set of mappings between FTAM responses (<Action Result> and <Diagnostic>) and FTP responses (reply codes).

The <Action Result> parameter of the FTAM File Service primitives conveys information which summarizes that available in the <Diagnostic> parameter. The value is never less than the most severe diagnostic value. The valid values of this parameter are "Success", "Transient Error", and "Permanent Error". The FTP response text should be supplied in the <Further Details> field of the <Diagnostics> sequence in the FTAM response and abort messages.

An FTAM <Action Result> "Success" may be accompanied by a <Diagnostic> with value of "Informative Error Type". These "Success" diagnostic messages are associated with error type 0 in the table below (and in [ISO8571-3]). Error type 1 indicates a transient error, while type 2 indicates a permanent error.

An FTP reply consists of a three digit number followed by some text. The number is defined as a 3-digit code, each digit of which has a special significance. The first digit conveys approximately the same information as the FTAM <Action Result> parameter; i.e., positive, transient negative, or permanent negative.

The FTP specification document [RFC959] explicitly states that the list of reply codes should not be expanded beyond that which is presented in [RFC959]. This requirement is adhered to in the mappings presented in this document.

9.1. FTP Reply Codes to FTAM Parameters

This section presents the set of mappings between FTP reply codes and their equivalent FTAM action and diagnostic parameters. Gateway support for these mappings is recommended, but not required. The following abbreviations are used for FTAM action parameter values:

trans = transient error
perman = permanent error

FTP Reply		FTAM Diagnostic		
Code	Text	Result	Type	Id
110	Restart marker reply	success	0	0
120	Service ready in nnn minutes	success	0	0
125	Data connection open, transfer starting	success	0	0
150	File status okay; about to open data connection	success	0	0
200	Command okay	success	0	0
202	Command not implemented;			

	superfluous	success	0	0
211	System status, or system help reply	success	0	0
212	Directory status	success	0	0
213	File status	success	0	0
214	Help message	success	0	0
215	NAME system type	success	0	0
220	Service ready for new user	success	0	0
221	Service closing control connection	success	0	0
225	Data connection; no transfer in progress	success	0	0
226	Closing data connection	success	0	0
227	Entering passive mode (h1,h2,...)	success	0	0
230	User logged in, proceed	success	0	0
250	Requested file action okay, completed	success	0	0
257	"PATHNAME" created	success	0	0
331	User name okay, need password	success	0	0
332	Need account for logon	success	0	0
350	Requested file action pending further information	success	0	0
421	Service not available, closing control connection	trans	1	1
425	Can't open data connection	trans	1	3
426	Connection closed, transfer aborted	trans	1	1014
450	Requested file action not taken, file unavailable (e.g., file busy)	trans	1	5041
451	Requested file action aborted, local error in processing	trans	1	5028
452	Requested action not taken, insufficient storage space	trans	1	9
500	Syntax error, command unrecognized	perman	2	5015
501	Syntax error in parameters or arguments	perman	2	4004
502	Command not implemented	perman	2	5016
503	Bad sequence of commands	perman	2	1015
504	Command not implemented for that parameter	perman	2	4003
530	Not logged in	perman	2	2020
532	Need account for storing files	perman	2	2008
550	Requested action not taken; file unavailable (e.g., file not found, no access)	perman	2	3013
551	Requested action aborted, page type	perman	2	5002
552	Requested file action aborted, exceeded storage allocation	perman	2	9

553 Requested file action not taken, | perman 2 3024
file name not allowed

9.2. FTAM Parameters to FTP Reply Codes

This section presents the set of mappings between FTAM diagnostic parameters and their equivalent FTP reply codes. Gateway support for these mappings is recommended, but not required. As previously mentioned, type 0 is an informative error type that may be returned with a "Success" action result, type 1 is a transient error type, and type 2 is a permanent error type.

FTAM Diagnostic			FTP Reply Code
Type	Id	Reason	

1,2	0	No reason	421
0	1	Responder error	211
1,2	1	Responder error	421
1,2	2	System shutdown	421
0	3	FTAM mgmt problem, unspecific	211
1,2	3	FTAM mgmt problem, unspecific	425
0	4	FTAM mgmt, bad account	221
2	4	FTAM mgmt, bad account	532
0	5	FTAM mgmt, security not passed	211
2	5	FTAM mgmt, security not passed	530
0	6	Delay may be encountered	211
0	7	Initiator error, unspecific	211
1,2	7	Initiator error, unspecific	421
0	8	Subsequent error	211
1,2	8	Subsequent error	421
0	9	Temporal insufficiency of resources	211
1,2	9	Temporal insufficiency of resources	452
1,2	10	Access req. violates VFS security	550
1,2	11	Access req. violates local security	550
2	1000	Conflicting parameter values	504
2	1001	Unsupported parameter values	504
2	1002	Mandatory parameter not set	504
2	1003	Unsupported parameter	504
2	1004	Duplicated parameter	504
2	1005	Illegal parameter type	504
2	1006	Unsupported parameter types	504
2	1007	FTAM protocol err., unspecific	426
2	1008	FTAM protocol err., procedure err	426
2	1009	FTAM protocol err., funct. unit err	426
2	1010	FTAM protocol err., corruption err.	426

2	1011 Lower layer failure	426
1,2	1012 Lower layer addressing error	426
1,2	1013 Timeout	426
1,2	1014 System shutdown	426
2	1015 Illegal grouping sequence	503
2	1016 Grouping threshold violation	503
2	1017 Inconsistent PDU request	503
2	2000 Association with user not allowed	532
2	2002 Unsupported service class	504
0	2003 Unsupported functional unit	211
2	2003 Unsupported functional unit	502
0	2004 Attribute group error, unspecific	211
1,2	2004 Attribute group error, unspecific	504
2	2005 Attribute group not supported	504
0	2006 Attribute group not allowed	211
2	2006 Attribute group not allowed	504
0	2007 Bad account	211
2	2007 Bad account	532
0	2008 Association management, unspecific	211
1,2	2008 Association management, unspecific	532
2	2009 Association management, bad address	532
1,2	2010 Association management, bad account	532
0	2011 Checkpoint window error, too large	211
2	2011 Checkpoint window error, too large	426
0	2012 Checkpoint window error, too small	211
2	2012 Checkpoint window error, too small	426
0	2013 Checkpoint window error, unsupp.	211
2	2013 Checkpoint window error, unsupp.	504
0	2014 Communications QoS not supported	211
1,2	2014 Communications QoS not supported	504
2	2015 Initiator identity unacceptable	532
0	2016 Context management refused	211
0	2017 Rollback not available	211
0	2018 Contents type list cut by responder	211
0	2019 Contents type list by Presentation Service	211
2	2020 Invalid filestore password	530
2	2021 Incompatible service classes	530
1,2	3000 Filename not found	550
1,2	3001 Selection attributes not matched	550
2	3002 Initial attributes not possible	550
2	3003 Bad attribute name	550
1,2	3004 Non-existent file	550
1,2	3005 File already exists	553
1,2	3006 File cannot be created	553
1,2	3007 File cannot be deleted	553
0	3008 Concurrency control not available	211

2	3008	Concurrency control not available	503
0	3009	Concurrency control not supported	211
2	3009	Concurrency control not supported	502
0	3010	Concurrency control not possible	211
2	3010	Concurrency control not possible	503
0	3011	More restrictive lock	211
1	3011	More restrictive lock	450
1,2	3012	File busy	450
1,2	3013	File not available	450
0	3014	Access control not available	211
1,2	3014	Access control not available	503
0	3015	Access control not supported	211
1,2	3015	Access control not supported	502
0	3016	Access control inconsistent	211
1,2	3016	Access control inconsistent	503
0	3017	Filename truncated	211
0	3018	Initial attributes altered	211
1,2	3019	Bad account	532
0	3020	Override selected existing file	211
0	3021	Override deleted and recreated	211
0	3022	Create override deleted and recreate file with new attributes	211
1,2	3023	Create override, not possible	553
1,2	3024	Ambiguous file specification	553
2	3025	Invalid create password	550
2	3026	Invalid delete password on override	550
2	3027	Bad attribute value	550
2	3028	Requested access violation	550
2	3029	Functional unit not available for requested access	550
0	3030	File created but not selected	211
1	3030	Invalid create password	550
0	4000	Attribute non-existent	211
1,2	4000	Attribute non-existent	501
1,2	4001	Attribute cannot be read	504
1,2	4002	Attribute cannot be changed	504
1,2	4003	Attribute not supported	504
2	4004	Bad attribute name	501
2	4005	Bad attribute value	501
0	4006	Attribute partially supported	211
0	4007	Additional set attribute value not distinct	211
1,2	5000	Bad FADU, unspecific	550
2	5001	Bad FADU, size error	501
2	5002	Bad FADU, type error	551
2	5003	Bad FADU, poorly specified	501
2	5004	Bad FADU, bad location	550
0	5005	FADU does not exist	550

1	5005 FADU does not exist	550
0	5006 FADU not available, unspecific	550
1,2	5006 FADU not available, unspecific	550
1,2	5007 FADU not available for reading	550
1,2	5008 FADU not available for writing	550
1,2	5009 FADU not available for location	550
1,2	5010 FADU not available for erasure	550
1,2	5011 FADU cannot be inserted	550
1,2	5012 FADU cannot be replaced	550
0	5013 FADU cannot be located	550
1,2	5013 FADU cannot be located	550
2	5014 Bad data element type	550
1,2	5015 Operation not available	500
1,2	5016 Operation not supported	502
0	5017 Operation inconsistent	211
2	5017 Operation inconsistent	503
0	5018 Concurrency control not available	211
1,2	5018 Concurrency control not available	503
0	5019 Concurrency control not supported	211
2	5019 Concurrency control not supported	502
0	5020 Concurrency control inconsistent	211
2	5020 Concurrency control inconsistent	503
0	5021 Processing mode not available	211
1,2	5021 Processing mode not available	503
0	5022 Processing mode not supported	211
2	5022 Processing mode not supported	504
0	5023 Processing mode inconsistent	211
2	5023 Processing mode inconsistent	503
0	5024 Access context not available	211
2	5024 Access context not available	503
0	5025 Access context not supported	211
2	5025 Access context not supported	504
1,2	5026 Bad write, unspecific	550
1,2	5027 Bad read, unspecific	550
0	5028 Local failure, unspecific	211
1,2	5028 Local failure, unspecific	451
0	5029 Local failure, filespace exhausted	211
1,2	5029 Local failure, filespace exhausted	552
0	5030 Local failure, data corrupted	211
1,2	5030 Local failure, data corrupted	451
0	5031 Local failure, data corrupted	211
1,2	5031 Local failure, data corrupted	451
2	5032 Future file size exceeded	451
0	5034 Future file size increased	211
0	5035 Functional unit invalid in processing mode	211
2	5035 Functional unit invalid in processing mode	503

0	5036	Contents type inconsistent	211
2	5036	Contents type inconsistent	503
0	5037	Contents type simplified	211
0	5038	Duplicate FADU name	211
1,2	5039	Damage to select/open regime	553
1,2	5040	FADU locking not available on file	450
1,2	5041	FADU locked by another user	450

9.3. Future Mapping Problem

At some point in the future, the FTAM <Responding Presentation Address> parameter may be used for purposes other than the current use of passing the final destination address in the FTAM-Initiated gateway service [NIST86]. If this happens, the destination address will have to be passed in another location, such as in the "@host" portion of the <Initiator Identity>. Currently, the FTP-FTAM gateway specification permits either mechanism for storage of the ultimate destination address.

9.4. Error Handling

The minimal acceptable solution for FTAM-Initiated service errors is to map FTP failures to FTAM "Unrecoverable error" and return the FTP diagnostic string in the FTAM <Further Details> field. Similarly for FTP-Initiated service errors, the minimal acceptable solution is to return reply code 221, "Service closing control connection, Logged out if appropriate". While this minimal solution is acceptable, the recommended approach for Gateway developers is to implement the mappings presented in Section 9.1, FTP Reply Codes to FTAM Parameters, and Section 9.2, FTAM Parameters to FTP Reply Codes.

10. Implementation and Configuration Guidelines

The intent of this specification is to specify the required characteristics and functions of an FTP-FTAM gateway. The specific approach taken to realize these specifications in an operational gateway are left to the discretion of the implementor. We do take the liberty, however, of suggesting several ideas concerning the configuration and implementation of such gateways.

10.1. Robustness

The gateway should be robust enough to handle situations where a subset of the FTP and/or FTAM protocols are implemented on a host.

The gateway should support multiple concurrent FTP and FTAM connections.

These are requirements for gateway implementations.

10.2. Well-Known TCP/IP Port

It is recommended that the FTP-Initiated gateway process listen on TCP/IP port 21, the well-known port for FTP listener processes. As the gateway computer is primarily intended to provide gateway services, use of this port will alleviate the need for gateway users to specify the desired port when they connect to the gateway. The standard FTP server listener process can then be moved to another port that is known to those users (e.g., System Administrators) requiring FTP-to-FTP access to the gateway computer.

10.3. Gateway Listener Processes

To simplify the administrative overhead on the gateway computer system, it is recommended that the FTP-Initiated service and FTAM-Initiated gateway listener processes be merged into a single executable module. This single daemon will act as the one and only gateway listener processes. As connections were established with hosts, other processes would be created.

10.4. Implementation Testing

To assist in the development and evaluation of FTP-FTAM gateway prototypes, NIST has developed a test system to evaluate a gateway's conformance to the protocol standards [NIST88].

10.5. POSIX File Naming and Organization

The OSI profiles do not define a standard manner for an FTAM Responder to return file names.

To avoid unnecessary complexity, proprietary file systems are not addressed in these mappings. Gateway support for POSIX file naming and organization conventions is required; i.e., files are assumed to be organized in a hierarchical structure in which all of the non-terminal nodes are directories and all of the terminal nodes are any other type of file.

11. Security Considerations

The gateway system may place the burden of authentication on the destination system. However, the gateway must accommodate the passing through of all authentication parameters. The authentication parameters of each protocol are applied at the destination and no additional parameters are needed for authentication at the gateway. As such, no gateway password file is required to support gateway

functions.

It is anticipated that the requirement for a strong authentication mechanism will soon replace the most currently used, userid and password mechanism. The U.S. National Security Agency (NSA) has already prototyped and has plans field a Message Secure Protocol (MSP) as part of the Defense Message System (DMS) Program which will soon become the Department of Defense (DoD) mandatory messaging system. MSP utilizes a public key encryption-like mechanism which will be used to authenticate users and allow signed operations. The current philosophy is to use this same mechanism for all authentication and access control situations, such as logging onto remote hosts or gateways. Detailed specifications for Pre-MSP, used in the unclassified though sensitive arena, are scheduled to be published in the first quarter of 1993. The requirement for gateways to process PMSP and MSP strong authentication mechanisms will be part of all future DoD procurements.

12. References

- [IS08571-1] Information processing systems - Open Systems Interconnection - File Transfer, Access and Management, Part 1: General Introduction, International Standards Organization for Standards, First Edition, October 1988.
- [IS08571-2] Information processing systems - Open Systems Interconnection - File Transfer, Access and Management, Part 2: Virtual Filestore Definition, International Standards Organization for Standards, First Edition, October 1988.
- [IS08571-3] Information processing systems - Open Systems Interconnection - File Transfer, Access and Management, Part 3: File Service Definition, International Standards Organization for Standards, First Edition, October 1988.
- [IS08571-4] Information processing systems - Open Systems Interconnection - File Transfer, Access and Management, Part 4: File Protocol Specification, International Standards Organization for Standards, First Edition, October 1988.
- [IS08571-5] Information processing systems - Open Systems Interconnection - File Transfer, Access and Management, Part 5: Protocol Implementation Conformance Statement, International Standards Organization for Standards, First Edition.

- [KILLE92] Hardcastle-Kille, S., "Using the OSI Directory to achieve User Friendly Naming", OSI-DS 24 (v1.1), October 1992.
- [MITRE87] Scott, J., "An FTP/FTAM Application Bridge, An FTAM/FTAM (MTR-87W00186)", The MITRE Corporation, July 1987.
- [NETWRX90a] Mindel, J., "Gateway Technical Specification" Open Networks, Inc. (formerly NetWorks One), 28 February 1990.
- [NETWRX90b] Mindel, J., "FTP Gateway User's Guide", Open Networks, Inc. (formerly NetWorks One), 28 February 1990.
- [NIST86] Wallace, M, et. al., "A Gateway Architecture Between FTP and FTAM (ICST/SNA86-6)", National Institute of Standards and Technology, U.S. Department of Commerce, July 1986.
- [NIST88] A Test System for Implementations of FTAM/FTP Gateways: Final Report Part 1, National Institute of Standards and Technology, U.S. Chamber of Commerce, October 1988.
- [NIST91] CSL Bulletin: File Transfer, Access, and Management, National Institute of Standards and Technology, U.S. Chamber of Commerce, July 1991.
- [NIST92] Stable Implementation Agreements for Open Systems Interconnection Protocols: Part 9 - FTAM Phase 2, Output from the March 1992 Open Systems Environment Implementors' Workshop (OIW), March 1992.
- [RFC959] Postel, J., and J. Reynolds, "File Transfer Protocol (FTP)", STD 9, RFC 959, USC/Information Sciences Institute, October 1985.
- [RFC1101] Mockapetris, P., "DNS Encoding of Network Names and other Types", RFC 1101, USC/Information Sciences Institute, April 1989.
- [RFC1279] Hardcastle-Kille, S., "X.500 and Domain", RFC 1279, University College London, November 1991.
- [ROSE90] Rose, M., "The Open Book: A Practical Perspective on OSI", Prentice-Hall Inc., 1990.

13. Authors' Addresses

Joshua L. Mindel
Open Networks, Inc.
11490 Commerce Park Dr., Suite 205
Reston, Virginia 22091 USA

Phone: (703) 648-0013
Email: mindel@netwrx1.nw1.com

Robert L. Slaski
Open Networks, Inc.
11490 Commerce Park Dr., Suite 205
Reston, Virginia 22091 USA

Phone: (703) 648-0013
Email: slaski@netwrx1.nw1.com