

Danny

See IEN 100-102

AUTODIN FTP Summary

The primary function of the AUTODIN II File Transfer Protocol (FTP) is to permit the efficient and reliable transfer of data files among a group of heterogeneous computer systems or hosts. Hosts are connected to communication networks which are, in turn, connected by gateways to form an internetwork. FTP is the agreement concerning the type and order of information exchanged in the process of accomplishing a file transfer.

There are several key ideas, features and innovations which make AUTODIN FTP an improvement over former FTPs:

* **The Copy Model.**

The basic underlying model of file transfer assumed for AUTODIN FTP is motivated by the standard single host file copy operation. A user issues the following command:

```
copy SourceFile to DestinationFile
```

The copy operation is performed by the file system.

* **Distribution and the Three Party Model.**

While a number of differences occur when a file is "copied" between distinct hosts, the basic copy model is still valid. The maximum number of hosts involved in the copy operation increases from one to possibly three: the hosts on which the User, SourceFile and DestinationFile reside, known as the Controller, Donor, and Recipient, respectively. The actual work of transferring the file must be split between the file systems of Donor and Recipient, with the Controller providing coordination and user interaction functions.

* **Communication Between Components.**

One of the Donor and Recipient is chosen as the Active component for a file transfer; the other is designated the

Passive component. The Controller interacts with the User and the Active component in initiating and specifying the nature of the file transfer. The Active component drives the Passive component, forwarding messages to and from the Controller as necessary.

* Access Control to Files.

There are provisions in the protocol for flexible access control to files. Commands are provided for exchanging standard user name and password information. Additional, alternative commands are present for exchanging the authenticated identity of the user who initiated the transfer. One use of this information, FTP Access Control Lists (ACL), solves a problem present in most FTPs: the requirement of multiple login. Under the FTP ACL scheme, access to a file is controlled by the Donor and Recipient components checking an ACL associated with each file. Basically, if the name in the ACL matches the authenticated identity of the user, then the access is allowed. Alternative schemes to FTP ACLs may use the authenticated user identity information transmitted by the protocol in any desired way.

* Machine Oriented Commands.

The encoding of commands is oriented towards use by computer programs, rather than by humans. Previous protocols have been hampered by having to express all commands as character strings. AUTODIN FTP uses a coded command and parameter reference structure oriented toward process to process interactions.

* Levels of Implementation.

The protocol has been designed so that there are three levels of implementation: Core, Base and Extensions. The Core is the minimum set of features necessary to engage in file transfer. The Base includes additions to the Core that enhance the operation of file transfer. The Extensions are those features that will be added to the protocol after the specification is made or experimental protocol features. We estimate that the Core implementation will require 6 to 9 person-months to implement on a PDP-11 class machine, with the Base requiring an additional 6 to 9 months.

* Formalization of Data Transfer Protocol.

All aspects of FTP dealing with data have been separated into a separate sub-protocol called the Data Transfer Protocol. The intent is to have DTP serve the data transfer needs of other high level protocols in addition to FTP. DTP is concerned with data typing, compression, record grouping, formatting. In addition it specifies a scheme for sharing a single connection between data and control items.

* Partial Transfers.

A model of the meaning of partial transfer to a file is developed. The protocol contains a mechanism for causing parts of a file to be transferred from one host to another for the most reasonable and useful special cases.

The rest of this summary will discuss the mechanisms of AUTODIN II FTP.

There are three recognizable phases of a file transfer: Component Initiation and Option Negotiation, File Specification, and File Contents Transfer. In the first phase, processes are created at the Donor and Recipient hosts and a negotiation occurs among them and the Controller to determine which of the optional FTP features are going to be in effect during the file transfer. In the second phase, the Controller gathers the file specifications from the user and interacts with the Donor and Recipient to verify, for example, that the user has access to the files specified. In the final phase, the actual contents of the file is transferred from the Donor to the Recipient.

In all of these phases the Controller, Donor and Recipient exchange information in the form of FTP Commands and Parameter References. An example of an FTP Command is the BeginTransfer command which indicates that the actual transfer of the contents of a file is to begin. Both Commands and Parameter References and the contents of a file being transferred travel over the same connection.

From a global viewpoint, there is a single set of values, known as FTP State Parameters, which store the current state of a file transfer operation. Each parameter has a well defined data type. An example of an FTP State Parameter is the SourceFileName parameter whose value is a set of ASCII characters.

The transfer is distributed among as many as three hosts, and each FTP component must maintain an identical set of FTP State Parameters. The work of file transfer is to set and read these parameters and to obey commands which reference these parameters. For example, the Controller gets the value of the SourceFileName parameter from the user and sends a "write" reference to this parameter to the Donor, thus informing the Donor about the name of the SourceFile. Also, upon receiving a BeginTransfer Command, the Donor opens the file whose name is contained in its copy of SourceFileName for transmission to the Recipient. The mechanism for replying about the success or failure of a Command is just another Command. For negative replies, the values of Parameters that caused the reply can be cited using the normal parameter referencing mechanism.

The Data Transfer Protocol (DTP) is used to transfer not only the contents of a file but also the Commands and Parameter References of FTP. Therefore all of the low level data typing and data grouping facilities of DTP may be used in the exchange of Parameter References and Commands occurring in the higher level FTP.

DTP divides the data stream into multiple ControlSegments. Each ControlSegment has a ControlHeader containing an Operation followed by zero or more DataSegments. Each DataSegment has a DataHeader followed by a DataBody. One field in the DataHeader indicates the number of 8-bit bytes in the DataBody.

FTP uses DTP in the following way: Each FTP Command is assigned a unique code which fits into the Operation field of a DTP ControlHeader. A Parameter Reference is encoded and sent as two DTP data items: a ReferenceDescriptor and a Value. Thus a single FTP Command with one Parameter Reference would correspond to a single DTP ControlSegment which encloses two DataSegments. Additional Parameter References would result in additional pairs of DataSegments in the single ControlSegment.