

Internet-Draft

Dilip Kumar Nanecha  
Chetan Aswathanarayana

Expires: July 16, 2006

January 16, 2006

**Safe GET & Safe PUT options for FTP**  
<[draft-dilip-ftpext-safe-transfer-00.txt](#)>

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

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

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

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

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

Copyright Notice

Copyright (C) The Internet Society (2006). All Rights Reserved

Abstract

This draft presents a mechanism to overcome one of the cumbersome aspects of file transfer. Ftp fails to transfer the data as there is not enough disk space on the remote machine, but it fails only after it does the substantial data transfer and discovers that there is no more disk space left on the remote machine.

The basic intention is to provide FTP Extension for checking the availability of the disk space before starting the actual file transfer operation on remote machines.



## Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC-2119](#) [5].

## Table of Contents

Status of this Memo.....	<a href="#">1</a>
Abstract.....	<a href="#">1</a>
<a href="#">1.</a> Introduction.....	<a href="#">3</a>
<a href="#">2.</a> Requirement.....	<a href="#">3</a>
<a href="#">3.</a> Concepts.....	<a href="#">5</a>
<a href="#">3.1.</a> File Size.....	<a href="#">5</a>
<a href="#">3.2.</a> RETRIEVE (RETR / GET).....	<a href="#">5</a>
<a href="#">3.3.</a> STORE (STOR / PUT).....	<a href="#">5</a>
<a href="#">3.4.</a> ALLOCATE (ALLO).....	<a href="#">5</a>
<a href="#">4.</a> Framework Specification.....	<a href="#">7</a>
<a href="#">5.</a> Description of New FTP Commands.....	<a href="#">9</a>
<a href="#">5.1.</a> LOCAL DISK SIZE (LDSIZE).....	<a href="#">9</a>
<a href="#">5.2.</a> SAFE PUT (SPUT).....	<a href="#">9</a>
<a href="#">5.3.</a> SAFE GET (SGET).....	<a href="#">9</a>
<a href="#">6.</a> Example.....	<a href="#">9</a>
<a href="#">7.</a> Security Considerations.....	<a href="#">10</a>
<a href="#">8.</a> Appendices.....	<a href="#">10</a>
8.A. References.....	<a href="#">10</a>
8.B. Authors' Addresses.....	<a href="#">11</a>
<a href="#">9.</a> Intellectual Property Statement.....	<a href="#">11</a>



## **1. Introduction**

When a file is to be sent to or received from a remote machine, there is no possible way of ensuring the file would be transferred successfully always.

In Certain Situations FTP transfer fails after performing substantial data transfer. In those cases we are loosing in terms of the time or network bandwidth, and many a times lost of productivity as we have to re-initiate the file transfer. This is specifically applicable when we do the ftp over WAN.

## **2. Requirement**

The requirement is to provide the user with new ftp commands, SGET [Safe Get] & SPUT [Safe Put] using which users can be rest assured that file transfer would take place with out any problem. SGET & SPUT are defined in such a way that users would get errors or exceptions if any before the actual file transfer occurs.

## **3. Concepts**

### **3.1 File SIZE**

FTP uses SIZE(SIZE OF FILE) command to obtain the transfer size of a file from the server-FTP process. The size command returns the exact number of octets (8 bit bytes) of the file which would be transmitted over the connection.

### **3.2 RETRIEVE (RETR / GET)**

This command causes the server-DTP to transfer a copy of the file, specified in the pathname, to the server- or user-DTP at the other end of the data connection. The status and contents of the file at the server site shall be unaffected.

### **3.3 STORE (STOR / PUT)**

This command causes the server-DTP to accept the data transferred via the data connection and to store the data as a file at the server site. If the file specified in the pathname exists at the server site, then its contents shall be replaced by the data being transferred. A new file is created at the server site if the file specified in the pathname does not already exist.

### **3.4 ALLOCATE (ALLO)**

This command may be required by some servers to reserve

sufficient storage to accommodate the new file to be transferred. The argument shall be a decimal integer representing the number of bytes (using the logical byte size) of storage to be reserved for the file. For files sent with record or page structure a maximum record or page

size (in logical bytes) might also be necessary; this is indicated by a decimal integer in a second argument field of the command. This second argument is optional, but when present should be separated from the first by the three Telnet characters <SP> R <SP>. This command shall be followed by a STORE or APPEND command. The ALLO command should be treated as a NOOP (no operation) by those servers which do not require that the maximum size of the file be declared beforehand, and those servers interested in only the maximum record or page size should accept a dummy value in the first argument and ignore it.

#### **4. Framework Specification**

Check the free/available disk space before doing the actual data transfer, whether this free space can accommodate the requested file transfer. If free space is available then reserve the space required for the file to be transferred. This can be accomplished with the help of new commands for FTP like SGET, SPUT & LDSIZE.

In order to implement SGET/SPUT we are going to need a new command LDSIZE (local disk size) which gives the local disk size available for a user on server/client. (This could be determined by using quotas of a user.)

SGET is similar to GET except for the fact that it would fetch the size of the remote file to be received. It would also get the local disk space available for the user based on quota. It will compare the sizes, and if file size is greater than disk size it would prompt user to verify the transaction (SGET command).

SPUT is also similar to PUT except for a new command on Server, LDSIZE: This command would enable users to know their disk space on the server. During SPUT it would query LDSIZE and check that against the file size, if file size is greater than LDSIZE the SPUT would prompt user to verify the transaction (SPUT command).

#### **5. Description of New FTP Commands**

##### **5.1 LOCAL DISK SIZE (LDSIZE)**

This command causes the server FTP to transfer the free space available for a user in number of octets (8 bit bytes) to the user FTP at the other end of the data connection.

This command will be useful in case of SPUT. It is not a mandatory to implement this command, since we can also get the free space available using many other ways. This is just one way of getting total available free disk space for a user. This can

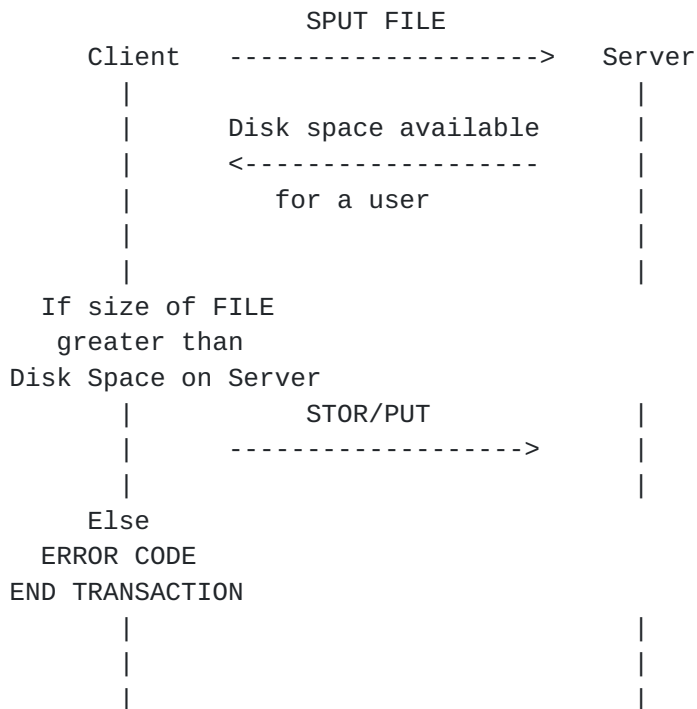
be achieved by using rquota to determine the disk allocation scheme for all users.



## 5.2 SAFE PUT (SPUT)

This command causes the server FTP to send the free disk space available for a user in number of octets (8 bit bytes) to the client, the client in turn would compare the free disk space with the size of the file to be transferred. If there is enough free space available then the data transfer is initiated similar to STOR command.

The interaction is as follows,

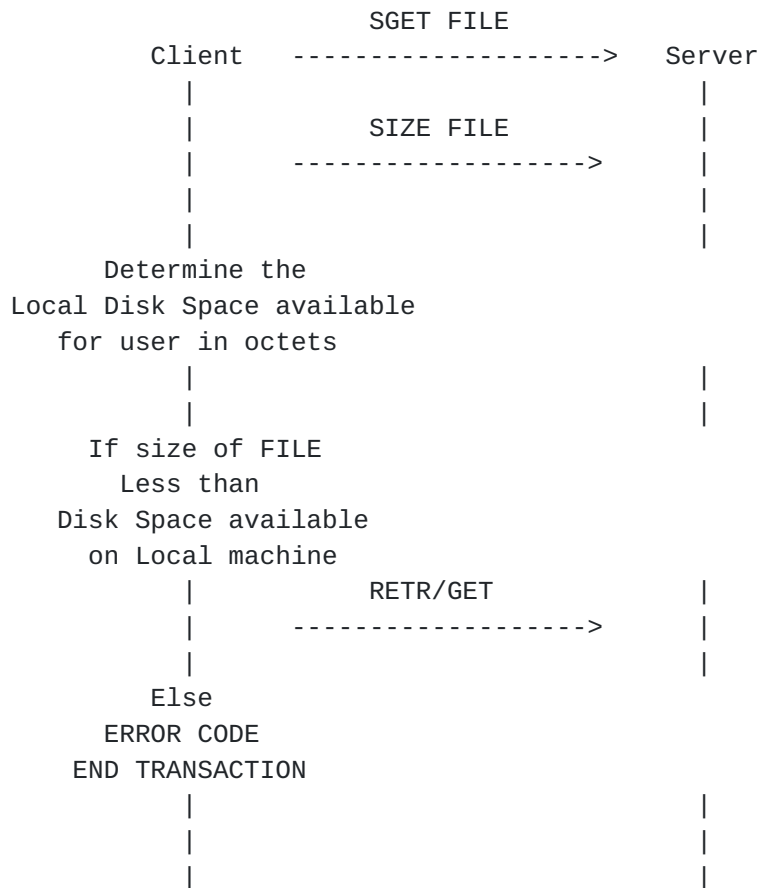


## 5.3 SAFE GET (SGET)

This Command causes the client FTP to determine size of the file which has to be retrieved from the server. This is done by sending size <file> command to the server. It will also determine free disk space available for a user in the local system in terms of number of octets (8 bit bytes). It compares the free space available with file size and if free space is available to hold the file, the server will transfer a copy of the file, specified in the pathname, to the user-DTP at the other end of the data connection.

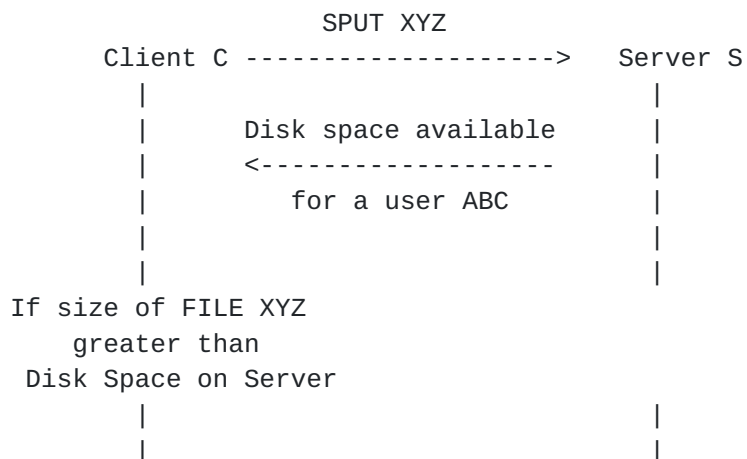


The interaction during safeget is as follows,



## 6. Example

A File named "XYZ" has to be SPUT From Client C to Server S by a user named "ABC".



```

|          STOR/PUT XYZ          |
|----->                         |
|                                |
| Else                           |
| ERROR CODE                     |
| END TRANSACTION                |
|                                |
|                                |
|                                |

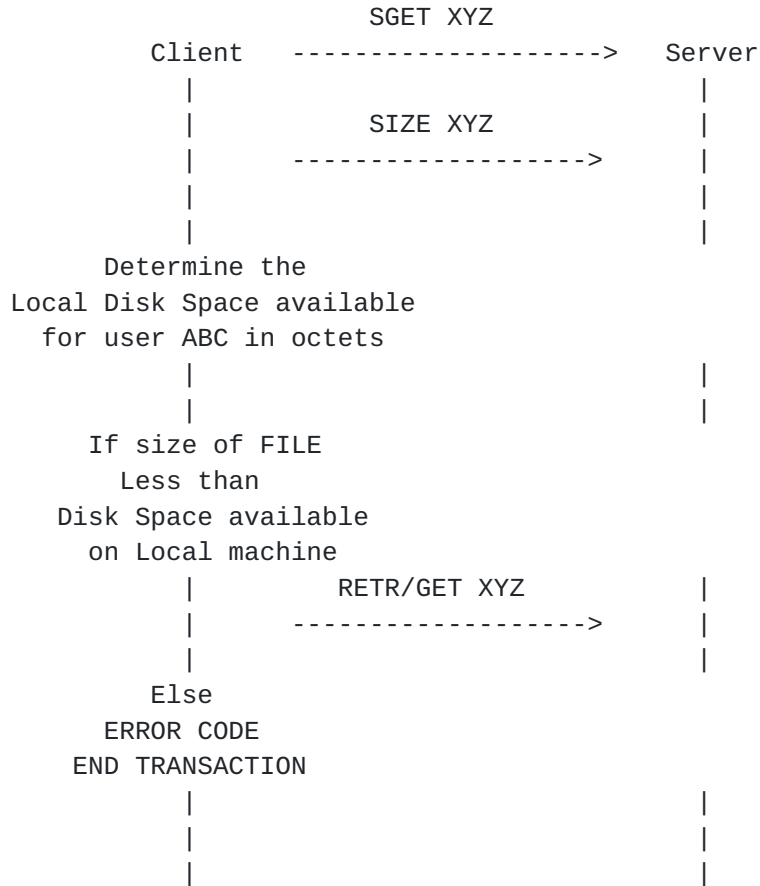
```

Dilip et al.

Expires July 16 2006

[Page 6]

A File named "XYZ" has to be SGET From Client C to Server S by a user named "ABC".



## 7. Security Considerations.

The most important Security point that needs to be considered in this draft is with respect to a new command LDSIZE, which will be able to protect anonymous servers from denial of service attack's. This might also be it's greatest weakness since the attacker know's the disk space available to him.



## **8. Appendices**

### **8.A. References**

The following documents contain definitions or specifications that are necessary to understand this document properly:

### **8.B. Author's Address**

Dilip Kumar Nanecha  
Hewlett-Packard,  
29, Cunningham road,  
Bangalore,  
India 560052  
dilip@hp.com  
+91-80-2205-3038

Chetan Aswathanarayana  
Hewlett-Packard,  
29, Cunningham road,  
Bangalore,  
India 560052  
aswathac@india.hp.com  
+91-80-2205-3103





## **8. Full Copyright Statement**

Copyright (C) The Internet Society (2004). This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

### Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

