TFTP Protocol

Teldat-Dm 765-I

Legal Notice

Warranty

This publication is subject to change.

Teldat offers no warranty whatsoever for information contained in this manual.

Teldat is not liable for any direct, indirect, collateral, consequential or any other damage connected to the delivery, supply or use of this manual.

Table of Contents

Chapter 1 Introduction

1.1 TFTP Protocol

The TFTP (Trivial File Transfer Protocol) is a protocol used to transfer files between various machines connected through a communications network.

TFTP is a very simple protocol, similar to a basic version of FTP (File Transfer Protocol), which is defined in RFC 1350. TFTP is implemented over a non-reliable, non-orientated to connection communications service. TFTP functionality mainly consists in a client reading or writing a file (occasionally an electronic mail) from a remote server.

The main characteristics distinguishing this from FTP are as follows:

- Uses UDP (port 69) as the transport protocol (unlike FTP which uses TCP port 21).
- It cannot list the content of the remote directories, and therefore needs the client to know the complete route in the server and the name of the files in order to transfer them.
- There are no authentication or encryption mechanisms, so it should be used over secure network paths (private networks, local networks, etc.).
- It supports three different transfer modes, "netascii", "octet" and "mail". The first two correspond to the “ascii” and “image” (binary) from the FTP protocol.

This protocol is often used to transfer small files between PCs in a network, such as when an X Window terminal or any other lite client boots from a network server. A common use of this protocol is the transfer of booting programs.

1.2 Detailing a TFTP Session

As TFTP uses UDP, there is no formal definition for session, client and server. However each file transferred through TFTP constitutes an independent exchange of packets, and there is an informal client-server relationship between the machine initiating the communication and that responding.

1.2.1 Fundamental Characteristics

Any file transfer begins with a client requesting a file reading or writing.

If the server accepts the request, the file is sent in fixed length blocks of 512 bytes. Each data packet contains one of these blocks and has a block number, beginning the transfer with block number 1. The data packets must be acknowledged, so in the absence of machine failure, the file is correctly transferred. The acknowledgment packet indicates the block number it’s acknowledging. Until acknowledgement for a block has been received, the next one is not sent.

A data packet containing fewer than 512 bytes indicates the end of the transmission.

If a packet is lost during transmission, the receiver will timeout and must retransmit the last transmitted packet (data or acknowledgement). This causes the sender of the lost packet to retransmit this said packet. Please note that both the client as well as the server use timeouts.

Three types of events cause errors:

- When it is not possible to accept a transfer request (file not found, permit violation).
- When an incorrectly formatted packet is received.
- When access to a resource is lost (the disk is full) in the middle of a transfer.

Errors cause the termination of the transfer. Sending an error packet indicates an error. This packet is neither acknowledged nor retransmitted, so the other end of the connection may not ever receive this.

1.2.2 Initializing a transfer

A transfer is initialized by a client sending a request (WRQ to request to write or RRQ to request to read) and a positive reply from the server received by the said client. This positive reply can be an acknowledgement packet – for write – or the first data packet –for read--. The acknowledgement packet for write takes block number 0. If the response to a request is an error packet, the request has been denied.

1.2.3 Types of TFTP Packets

TFTP uses 5 types of packets:

Type	Acronym	Operation
1	RRQ	Read request
2	WRQ	Write request
3	DATA	Data packet
4	ACK	Acknowledgement packet
5	ERR	Error packet

1.2.3.1 RRQ and WRQ Packets

These take the following format:

Type	File name

Type: 1 for RRQ and 2 for WRQ.

File name: Arbitrary length string.

1.2.3.2 DATA Packets

These take the following format:

Type	Block number	Data

Type: 3

Block number: 1 for the first block and this progressively increases up to 65535 and consequently zeroizes again.

Data: from 0 to 512 bytes with a file block that is read or write. If this is precisely 512 bytes, it is NOT the last data packet. If it is from 0 to 511 bytes then this indicates the end of the transfer.

1.2.3.3 ACK Packet

These take the following format:

Type	Block number

Type: 4

Block number: Indicates the block number being acknowledged. If this is zero is this an acknowledgement for a write request.

All packets that are not ERR must be acknowledged. Sending a data packet is an ACK acknowledgement for the previous block.

Therefore WRQ or DATA packets are acknowledged by ACK or ERROR packets, while RRQ and ACK packets are acknowledged by DATA or ERR packets.

1.2.3.4 ERR Packets

These take the following format:

Type	Error code	Message

Type: 5

Error Code: Indicates the nature of the error. The following table explains the meaning of each value:

Value	Meaning
0	Not defined
1	File not found
2	Access violation
3	Disk full
4	Illegal TFTP operation

5	Unknown transfer mode
6	File already exists
7	No such user

Message: **Undetermined length string to extend information on an error should this be necessary.**

1.2.4 Normal Termination

The end of a transfer is indicated with a DATA packet of 0 to 511 bytes. This packet must be acknowledged with an ACK packet (like all other DATA packets).

The machine acknowledging the last DATA packet can close its side of the communication after sending the ACK, however it is more normal to wait a while in order to retransmit the last ACK should it get lost. This situation is detected if the machine receives the last DATA packet again.

The machine sending the last DATA packet must wait until it has received the ACK or stop after a determined period of time. If it stops, it will terminate its side of the communication without knowing if the transfer has been completed or not (depending if it was the last ACK or the last DATA packet respectively, which was lost).

1.2.5 Premature Termination

If a request could not be accepted or an error has occurred in the middle of a transfer, then an error packet is sent and the communication terminated. This packet is not acknowledged nor retransmitted which means it may never reach its destination.

Chapter 2 TFTP Configuration

2.1 Accessing the TFTP Configuration Menu

In order to access the TFTP configuration menu, use the **FEATURE TFTP** command found in the main configuration menu.

Example:

```
*config
Config>feature tftp
-- TFTP user configuration --
TFTP config>
```

2.2 Configuration menu commands

This section describes the TFTP configuration menu commands.

The following table summarizes the commands available in the TFTP configuration menu. These commands are more fully explained further on.

Command	**Function**
? (HELP)	Displays the configuration commands or their options.
NO	Configures the parameters with their default values.
SERVER	Configures the TFTP server parameters.
EXIT	Exits the TFTP configuration menu.

2.2.1 ? (HELP)

Displays the available configuration commands or their options.

Syntax:

```
TFTP config>?
```

Example:

```
TFTP config>?
  no         Negate a command or set its defaults
  server     Configures TFTP server parameters
  exit       Exit to parent menu
TFTP config>
```

2.2.2 NO

Configures the parameters with their default values.

Syntax:

```
TFTP config>no ?
  server     Configures TFTP server parameters
        enable          Enables TFTP server
        file-system     File system to use for TFTP
        port            Port used to receive TFTP requests
```

Example:

Disables the TFTP server.

```
TFTP config>no server enable
TFTP config>
```

Example:

Sets the file system used by the TFTP server to its default value.

```
TFTP config>no server file-system
TFTP config>
```

2.2.3 SERVER

Configures the TFTP server parameters.

Syntax:

```
TFTP config>server ?
  enable         Enables TFTP server
  file-system    File system to use for TFTP
  port           Port used to receive TFTP requests
```

2.2.3.1 SERVER ENABLE

Enables the TFTP server. Once enabled, the device responds to file read and write requests.

Example:

```
TFTP config>server enable
TFTP config>
```

2.2.3.2 SERVER FILE-SYSTEM

Configures the file system used by the TFTP server to its default value. The TFTP server reads or writes files in the configured file system.

Syntax:

```
TFTP config>server file-system ?
  NUL    Default file system when a system has not been loaded
  TST    Checks that the FTP is operating correctly
  MEM    Temporary buffer
  DSK    Handles the disk.The code and configurations are stored here
  FCO    Handles the code Flash memory
  FDA    Handles the data Flash memory
  BIO    Handles the BIOS zone.Only the BIOS code files are saved here
  SMC    Handles the Smart Memory Card
TFTP config>
```

The available file systems are as follows:

NUL: default file system.

TST: used to check the performance of the protocol.

MEM: temporary buffer.

DSK: disk system. The code and configurations are saved here.

FCO: Flash memory used for the code.

FDA: Flash memory used for data.

BIO: BIOS area. To store BIOS files.

SMC: Smart Card.

Default value: the NUL system is used by default.

2.2.3.3 SERVER-PORT

Configures the port used by the TFTP server to listen to requests.

Syntax:

```
TFTP config>server port <port>
  <0..65535>    Port number
```

Example:

Configuration for port 150 as the listening port.

```
TFTP config>server port 150
```

Default value: port 69 is used by default.

2.2.4 EXIT

Exits the TFTP configuration menu.

Syntax:

```
TFTP config>exit
```

Example:

```
TFTP config>exit
Config>
```

Chapter 3 TFTP Monitoring

3.1 Accessing the TFTP Monitoring Menu

In order to access the TFTP monitoring menu, use the **FEATURE TFTP** command found in the main monitoring menu.

Example:

```
*monitor
Console Operator
+features tftp
TFTP manager
TFTP+
```

3.2 TFTP Monitoring Menu Commands

This section describes the TFTP monitoring menu commands. These commands allow you to use the TFTP client.

The following table summarizes the commands available in the TFTP Monitoring Menu:

Command	Function
del	Deletes the specified file from the device's active file system.
dir	Displays information on the active file system directory.
get	Transfers (read) a file stored in a TFTP server to the TFTP client active file system.
put	Transfers (write) a file from the active file system of the TFTP client to a TFTP server.
syst	Establishes the active file system in the device for the TFTP client.
exit	Exits the TFTP Monitoring menu.

3.2.1 Del

Deletes the specified file from the active file system in the device.

Syntax:

```
TFTP>del <file>
```

- **< file >** the complete name (including the extension) of the file which we wish to delete from the active file system

Example:

```
TFTP>dir
Current file system: DSK
  Name                    Length Attributes
  -----------------------------------------
                               0   ---V--
  ATLAS.BIN              3521280   -----A
  ATLAS.CFG                 2463   -----A
  IGMP.CFG                  1611   -----A
Total: 4 files, 3525354 bytes. 390144 bytes free.
TFTP>del igmp.cfg
File erased
TFTP>
```

In order to check that the file has effectively been deleted, you can list the files currently found in the active file system.

```
TFTP>dir
Current file system: DSK
```

```
  Name                      Length Attributes
  -----------------------------------------
                                 0   ---V--
  ATLAS.BIN                3521280   -----A
  ATLAS.CFG                   2463   -----A
Total: 3 files, 3523743 bytes. 392192 bytes free.
TFTP>
```

3.2.2 Dir

Lists information on the active file system directory on screen.

Syntax:

```
TFTP>dir
```

Example:

```
TFTP>dir
Current file system: DSK
  Name                      Length Attributes
  -----------------------------------------
                                 0   ---V--
  ATLAS.BIN                3521280   -----A
  ATLAS.CFG                   2463   -----A
  IGMP.CFG                    1611   -----A
Total: 4 files, 3525354 bytes. 390144 bytes free.
TFTP>
```

The meaning of the fields that appear is as follows:

- *General Information*

Current file system	Device file system active in the TFTP client.
Total	Total number of files in the active file system, total space in bytes occupied by the said files. The rest is the free space in the active file system.

- *Information on the files*

Name	Complete name of the file.
Length	Space in bytes occupied by the file.
Attributes	File attributes (R: Read only / H: Hidden / S: System / V: Volume / D: Directory / A: Archive).

3.2.3 Get

Transfers (read) a file stored in the TFTP server to the TFTP client active file system.

Syntax:

```
TFTP>get [<vrf>] <server> <source_file> [{<ip> | <ifc>}]
TFTP>get [<vrf>] <server> <source_file> [<destination_file>] [{<ip> | <ifc>}]
```

- **< vrf >** This is the name of the VRF used to communicate with the TFTP server in cases where this is located in a secondary VRF.
- **< server >** this is the TFTP server IP address from where we wish to transfer the source file.
- **< source_file >** this is the complete name of the file (including extension and route) that we wish to transfer (read) from the TFTP server file system.
- **< destination_file >** this is the complete name of the file (including the extension) where we want to write the original file within the TFTP client active file system.
- **< ip >** this is the IP address used as source for the TFTP petition.
- **< ifc >** this is the name of the interface used to get the TFTP petition IP source address.

Example 1:

```
TFTP>get 10.1.1.1 cfgfiles\atlasTeldat.cfg atlas.cfg
Receiving atlas.cfg from 10.1.1.1\cfgfiles\atlasTeldat.cfg...
```

```
     2463
File successfully received
TFTP>
```

In order to check that the file has effectively been transferred from the server, you can list the files currently found in the active disk.

```
TFTP>dir
Current file system: DSK
  Name                    Length Attributes
  -----------------------------------------
                               0   ---V--
  ATLAS.BIN              3521280   -----A
  ATLAS.CFG                 2463   -----A
Total: 3 files, 3523743 bytes. 392192 bytes free.
TFTP>
```

Example 2:

```
TFTP>get vrf-3 10.1.1.1 atlas.cfg 10.1.1.51
Receiving atlas.cfg from 10.1.1.1...
     2463
File successfully received
TFTP>
```

3.2.4 Put

Transfers (write) a file from the active file system of the TFTP client to the TFTP server.

Syntax:

```
TFTP>put [<vrf>] <server> <source_file> [{<ip> | <ifc>}]
TFTP>put [<vrf>] <server> <source_file> [<destination_file>] [{<ip> | <ifc>}]
```

- **< vrf >** This is the name of the VRF used to communicate with the TFTP server in cases where this is located in a secondary VRF.
- **< server >** TFTP server IP address to which you wish to transfer the source file.
- **< source_file >** this is the complete name (including the extension) of the file that you want to transfer from the active file system.
- **< destination_file >** this is the complete file name (including extension and route) where you want to write the source file within the TFTP server file system.
- **< ip >** this is the IP address used as source for the TFTP petition.
- **< ifc >** this is the name of the interface used to get the TFTP petition IP source address.

Example 1:

```
TFTP>put 10.1.1.1 atlas.cfg \cfgfiles\atlasTeldat.cfg
Sending atlas.cfg to 10.1.1.1\\cfgfiles\atlasTeldat.cfg...
Sending 2463 bytes
     2463
File successfully sent
TFTP>
```

Example 2:

```
TFTP>put vrf-2 10.1.1.1 atlas.cfg ethernet0/1.11
Sending atlas.cfg to 10.1.1.1...
Sending 2463 bytes
     2463
File successfully sent
TFTP>
```

3.2.5 Syst

Establishes the active file system in the device for the TFTP client.

Syntax:

```
TFTP >syst <file system>
```

- **< file system >** this is the code for the file system name we want to activate.

In order to find out what file systems are available in the device, activate the root file system (NUL) and subsequently a list of the existing files will appear. An example of this is given below:

```
TFTP>syst nul
Current file system set to NUL
TFTP>dir
Current file system: NUL
  Name                    Length Attributes
  -----------------------------------------
  TST                          0   ----D-
  MEM                          0   ----D-
  DSK                          0   ----D-
  BIO                          0   ----D-
Total: 4 files, 0 bytes.
TFTP>
```

The list of directories that appear corresponds to the manageable file systems in the device. A list showing the most common file systems in the Teldat devices is shown below together with their applications:

BIO	BIOS zone. Only the BIOS code files are saved here
DSK	Disk. The code and configurations are stored here
FCO	Code Flash memory
FDA	Data Flash memory
MEM	Temporary buffer
NUL	Default file system when a system has not been loaded
SMC	Smart Memory Card
TST	FTP Checking zone

Example:

```
TFTP>syst fda
Current file system set to FDA
TFTP>
```

By default the active file system is DSK.

3.2.6 Exit

Exits the TFTP Monitoring menu.

Syntax:

```
TFTP>exit
```

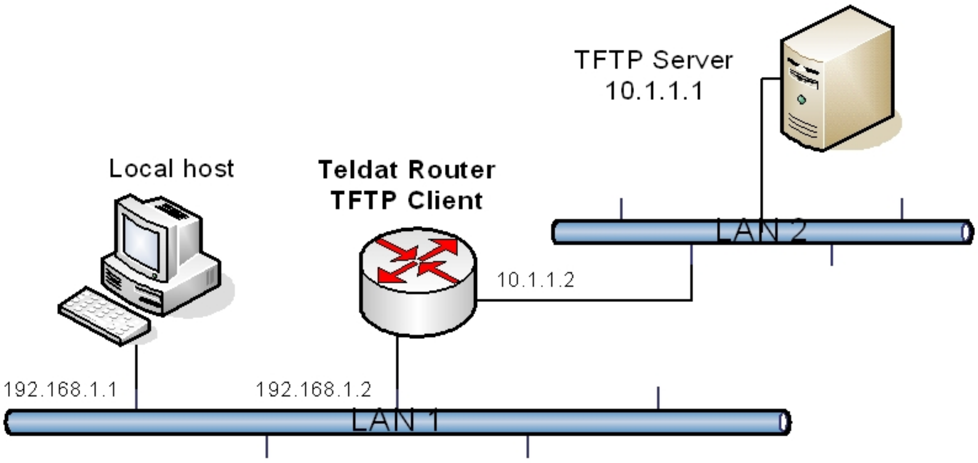
Example:

```
TFTP>exit
+
```

Chapter 4 Example

4.1 Loading the code and the configuration file

Supposing we have the following situation:



We want to remotely download a code release (*newatlas.bin*) as well as a new configuration file (*newatlas.cfg*) for the router. Both files are stored in a TFTP server, accessible through the device.

The first task is to connect to the router from the local host (e.g. via telnet) and subsequently use the TFTP client the router has to transfer the required files from the TFTP server.

Once connected to the router (through the console or LAN), access the TFTP clients interface and select the file system where we want to load the files, in this case, DSK[1]:

```
Teldat               (c)2001-2005
Router model ATLAS 2 8 CPU MPC860      S/N: 403/00122
1 LAN, 3 WAN Lines, 1 ISDN Line
CIT software version: 10.5.4-Alfa Feb 16 2005 13:30:43
*monitor
Console Operator
+tftp
TFTP manager
TFTP>syst dsk
Current file system set to DSK
TFTP>
```

To check we are in the selected file system, execute a listing:

```
TFTP>dir
Current file system: DSK
  Name                    Length Attributes
  -----------------------------------------
                               0   ---V--
  ATLAS.BIN              3521280   -----A
  ATLAS.CFG                 2463   -----A
  IGMP.CFG                  1611   -----A
Total: 4 files, 3525354 bytes. 390144 bytes free.
TFTP>
```

Once that is done, begin to download the new code release –*newatlas.bin*– found in the TFTP server "*code*" directory. To do this, execute the following command:

```
TFTP>get 10.1.1.1 \code\newatlas.bin atlas.bin
Receiving newatlas.bin from 10.1.1.1\\code\newatlas.bin...
   3521536
```

[1] Given that DSK is the default file system – where the code and the configuration files for the devices are normally loaded – it won't be necessary, in this particular case, to execute a selection operation.

```
File successfully received
TFTP>
```

As you can see, we have given the destination file the same name – *atlas.bin*- as the code file is already in the device (displayed on executing the previous listing). This means the said file will be overwritten. However if the device has sufficient free space in the file system, we recommend you use a different destination file name and subsequently activate this new code release selecting the said file through the *set application-active* configuration command. This is a precaution means against situations such as a power cut occurring during the download, the device in this case would be out of service as the code file will be incomplete.

Note

Whenever possible, we strongly recommend that the new file is saved under a different name, establishing it as active, and subsequently deleting the old file.

Next we need to download from the configuration file – *newatlas.cfg*- which can be found in the TFTP server *"cfgfiles"* directory. To do this, execute the following command:

```
TFTP>get 10.1.1.1 \cfgfiles\newatlas.cfg atlas2.cfg

Receiving atlas2.cfg from 10.1.1.1\\cfgfiles\newatlas.cfg...
     2463
File successfully received
TFTP>
```

In this case, we have saved the new configuration file as atlas2.cfg in the active file system (DSK).

We can check that both files have been correctly written to DSK by listing the active file system and checking that the name and length coincide with the transmitted files:

```
TFTP>dir
Current file system: DSK
  Name                    Length Attributes
  -----------------------------------------
                             0   ---V--
  ATLAS.CFG                2463   -----A
  IGMP.CFG                 1611   -----A
  ATLAS.BIN             3521536   -----A
  ATLAS2.CFG               2463   -----A
Total: 5 files, 3528073 bytes. 387072 bytes free.
TFTP>
```

Please note that in order to activate the new configuration file in the device, you need to inform [2] the device through the configuration menu:

```
TFTP>exit
+ctrl+p
*config
Config>set file-cfg atlas2.cfg
Config>
```

The last action to carry out is to reboot the device so the new code release and the new configuration activate.

```
Config>ctrl+p
*load immediate
Are you sure to reload the device(Yes/No)? y
  Done
```

[2] Provided that the name of the active configuration file currently in the device has not been used for the destination file.