



@server

iSeries

TFTP server

Version 5 Release 3





@server

iSeries

TFTP server

Version 5 Release 3

Note

Before using this information and the product it supports, be sure to read the information in "Notices," on page 9.

Fourth Edition (August 2005)

This edition applies to version 5, release 3, modification 0 of IBM Operating System/400[®] (product number 5722-SS1) and to all subsequent releases and modifications until otherwise indicated in new editions. This version does not run on all reduced instruction set computer (RISC) models nor does it run on CISC models.

© Copyright International Business Machines Corporation 2000, 2005. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

TFTP	1
Print this topic.	1
Configuring TFTP for Clients	1
Changing TFTP Attributes	2
Server Ports and Client Ports	3
TFTP Transfer Size Option.	3
TFTP Subnet Broadcast Option	3
Client to Server TFTP Read Request (RRQ) Options	5
Server to Client TFTP Option Acknowledgment (OACK)	6

Server to Client Broadcast Data (BDATA) Packets	7
Exit points for controlling TFTP server.	7

Appendix. Notices	9
Trademarks	10
Terms and conditions for downloading and printing publications	10

TFTP

Trivial File Transfer Protocol, or TFTP, is a simple protocol that provides basic file transfer function with no user authentication. TFTP is intended for applications that do not need the sophisticated interactions that FTP provides. Together, TFTP and Bootstrap Protocol (BOOTP) provide support for clients of an iSeries 400 system. They also provide support for other clients that use the TFTP and BOOTP protocols.

You can work with TFTP server properties through iSeries Navigator, the graphical user interface (GUI) for OS/400®.


Use Print this topic to print out the TFTP articles.

Print this topic

To view or download the PDF version, select TFTP (about 187 KB).

To save a PDF on your workstation for viewing or printing:

1. Open the PDF in your browser (click the link above).
2. In the menu of your browser, click **File**.
3. Click **Save As...**
4. Navigate to the directory in which you would like to save the PDF.
5. Click **Save**.

If you need Adobe Acrobat Reader to view or print these PDFs, you can download a copy from the Adobe Web site (www.adobe.com/prodindex/acrobat/readstep.html)  .

Configuring TFTP for Clients

To allow clients to use the TFTP server, you must ensure that the QTFTP profile has authority to access the directories and files that the clients access through the TFTP server. You also need to set the TFTP server attributes to allow the desired client requests.

When configuring TFTP for use by clients, first determine the directories and files that the clients are using. For this example, the clients use the TFTP server to read files from the directory `/netpc/bin/system`.

1. Use the MKDIR command with an argument of `/netpc` to create the directory `/netpc`, as follows:
`MKDIR (netpc)`
2. Specify the WRKLNK command with an argument of `/netpc`, as follows:
`WRKLNK (netpc)`
3. Specify option **9** to display the current authorities.
4. For the *PUBLIC user, specify option **2**, Change user authority, and specify *NONE for New data authorities. This ensures that the file is not open to the public.
5. To add a user on the Work with Authority menu, specify the following on the first line: 1 for Opt, QTFTP for User, and *RX for Data Authority. Press Enter.
6. Press the **PF5** key to refresh the menu. You see the userid *PUBLIC with a data authority of *EXCLUDE, the userid QTFTP with a data authority of *RX, and your own userid with a data authority of *RWX.

Use the MKDIR command to create the following directories:

```
/netpc/bin
/netpc/bin/system
```

Each directory inherits the authority of the parent directory and has the owner added implicitly as a user with *RWX authority. Copy any files that the client requests to the netpc/bin/system subdirectory. You can copy the files in various ways, such as using the COPY command, File Transfer Protocol (FTP), or iSeries Access. You must ensure that the QTFTP profile has *R authority to each file that the client requests. To set the authorities for the files, use the WRKLNK command and option 9, Work with Authority.

7. Specify the CHGTFTP command and press the PF4 key.
8. Change the Alternate source directory to /netpc/bin/system and press Enter. This allows the TFTP server to request any file with the appropriate authority settings, including the directory /netpc/bin/system in its path.
9. To have the changes take effect, stop the TFTP server with ENDTCPVR *TFTP and restart it by using STRTCPSVR *TFTP.

Changing TFTP Attributes

Use the Change TCP/IP TFTP Attributes (CHGTFTP) command to change the TFTP server attributes. The following are two different ways to get to this command prompt:

- Specify the CHGTFTP command.
- Select option 3 on the Configure TCP/IP Applications (CFGTCPAPP) display.

Note: You must have *IOSYSCFG special authority to make changes to the TFTP attributes with the CHGTFTP command.

```

Change TFTP Attributes (CHGTFTP)

Type choices, press Enter.

Autostart server . . . . . *NO          *YES, *NO, *SAME
Enable subnet broadcast . . . . *YES      *YES, *NO, *SAME
Number of server jobs:
  Minimum . . . . . 2          1-20, *SAME, *DFT
  Maximum . . . . . 6          1-250, *SAME, *DFT
Server inactivity timer . . . . 30        1-1440, *SAME, *DFT
ASCII single byte CCSID:
  Coded character set identifier 00819      1-65532, *SAME, *DFT
Maximum block size . . . . . 1024        512-65464, *SAME, *DFT
Connection response timeout . . 60        1-600, *SAME, *DFT
Allow file writes . . . . . *NONE       *DFT, *NONE, *CREATE...
Alternate source directory . . . '*NONE'

More...

F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
  
```

Figure 1. Change TFTP Attributes (CHGTFTP) – Display 1


```
Change TFTP Attributes (CHGTFTP)

Type choices, press Enter.

Alternate target directory . . . '*NONE'
```

Figure 2. Change TFTP Attributes (CHGTFTP) – Display 2

Server Ports and Client Ports

The TFTP server uses a subnet-directed broadcast address as the destination address. It also uses a well-known port as the port of datagrams sent to clients that have requested the subnet broadcast option. The clients listen for and receive datagrams on the well-known port. The keyword for the well-known port is **subntbcst_tftp**, and its decimal value is **247**.

The TFTP server sends subnet-directed broadcast datagrams to clients that request the subnet broadcast option. The source ports from which the TFTP server sends these datagrams do not have to be unique. They can be arbitrarily allocated.

Some routers filter or block subnet-directed broadcast datagrams. In support of router filters, you can define restricted ports for the QFTP profile. If you define restricted ports for the QFTP profile, the TFTP server uses only the defined restricted ports as the source ports for the subnet-directed broadcast datagrams. Network administrators define router filtering rules to allow subnet-directed broadcast datagrams to pass through router filters based on the source port of subnet-directed datagrams being one of the restricted ports defined for the QFTP profile.

TFTP Transfer Size Option

The Transfer Size option allows the client to determine how much data is transferred on a read request (RRQ). This is useful for requesting a subnet broadcast of a file. The client finds the size of the buffer it needs in order to store the file in memory. Drawing from this block size, the client determines the number of blocks for the transfer. The number of blocks is helpful information for tracking the blocks that have been received. You can also use it for the last block acknowledgment (ACK), which must be sent to terminate a transfer normally. Without the Transfer Size option, determining the size and the last block of the transfer requires the client to wait for a block to be received that is smaller than the block size of the transfer.

Note: For files transferred in *netascii* mode, this option might not be as useful if you are converting the data during the transfer in a way that changes its size. Also, the server might require additional processing time to determine the transfer size due to conversion of the file to the appropriate CCSID.

TFTP Subnet Broadcast Option

| Broadcast storms are a performance problem which may occur when there are large numbers of systems
| that boot from the network. These storms occur when large numbers of clients request their boot code at
| the same time. When hundreds of systems are involved in booting, the same data must be routed
| through each hop in the network between each system and the server.

| The TFTP Subnet Broadcast option provides a solution to this problem. It allows the server to broadcast
| the boot code to systems on a subnet basis. Using subnet-directed broadcast, Subnet Broadcast data

| packets are unicast between routers until they reach the subnet on which the systems reside. At this
| point, the router at the destination subnet broadcasts the data packets to the systems on the subnet.
| Disinterested hosts on the subnet throw the data packets away. The packets are usually thrown away by
| the host's IP layer after it determines that no applications are interested in receiving data on the port to
| which the broadcast was directed. See Figure 3 on page 5 for an illustration of a subnet-directed
| broadcast. This solution can drastically reduce the network traffic as well as the time that it takes many
| systems to boot (when booting simultaneously).

| The TFTP Subnet Broadcast option enables clients to join a broadcasting filegroup. It also allows clients to
| receive all subsequent blocks for a file until the client becomes the master client. A client becomes the
| master client when it receives an Option Acknowledge (OACK) packet from the TFTP server that
| indicates that it is the master client. A client must keep track of blocks that it receives. After a client
| becomes the master client, it can request the blocks that it has not received. The master client requests
| blocks by sending ACK packets that include the block number of the block prior to the block that the
| master client requires. For example, if the client wants block 5, it sends an ACK with a block number of
| 4.

| When a client receives an OACK packet that indicates that it is the master client, the client must send an
| ACK that requests the first block it requires. From then on, the client must request blocks in ascending
| but not necessarily consecutive order. A master client continues to send ACK packets to the server to
| indicate the next block that it requires. When the master client receives all of the blocks it requires, it
| sends an ACK with the number of the last block on the file being transferred. Once the server receives an
| ACK with the last block number of the file being transferred, the transfer to the client sending the ACK is
| considered complete. A client can terminate its transfer at any time by sending an ACK for the last block
| or by sending an Error (ERR) packet. A client can terminate this transfer regardless of whether it is the
| master client or not.

Note: This TFTP Subnet Broadcast option is designed to improve simultaneous transfer of large files to
multiple clients on a common subnet. This option does not help with files that require only a few
blocks to transfer or single client transfers.

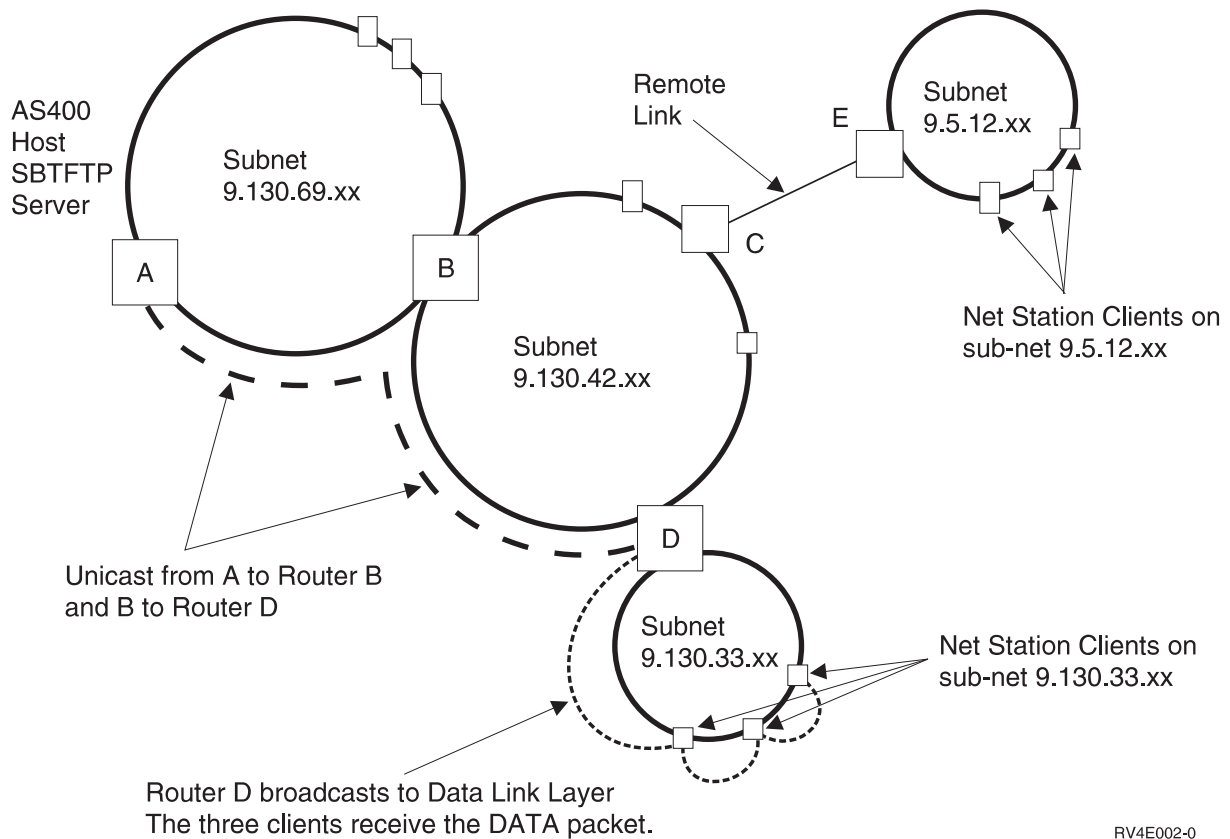


Figure 3. Example of Broadcasting over Subnets

Client to Server TFTP Read Request (RRQ) Options

The information that follows includes the additional TFTP options that are supported and a description of their use. To view the standard TFTP request parameters and their meanings, refer to Internet Request for Comments (RFC) 1350. For more information related to the TFTP options that are described here, see RFCs 1782, 1783, and 1784. Internet RFC 2090 describes the TFTP multicast option, which has some similarities to the Subnet Broadcast option. However, the TFTP Multicast option is not supported at this time. The TFTP Multicast option RFC is mentioned here as a reference to help understand the Subnet Broadcast option.

The following is a list of supported options and their descriptions:

blksize

Null (0h) terminated keyword *blksize* that is followed by the requested block size and represented as a null-terminated ASCII string. This option requests a block size for the requested file transfer instead of using the default of 512.

sbroadcast

Null-terminated keyword *sbroadcast* that is followed by the subnet mask of the subnet to which the client is connected. This option indicates that the client wants to participate in a subnet-directed broadcast group. The subnet mask that is included with this option is used with the client's IP address to determine the client's subnet address.

tsize

Null-terminated keyword *tsize* that is followed by a null-terminated ASCII representation of 0 (30h). This option is a request for the server to return the file size in an Option Acknowledgment (OACK).

Server to Client TFTP Option Acknowledgment (OACK)

The TFTP server sends an Option Acknowledgment (OACK) to a client in response to either a read request or a write request that includes additional TFTP options as described in “Client to Server TFTP Read Request (RRQ) Options” on page 5. An OACK that the servers sends in response to a transfer request includes only responses to requested options that the server supports. The server can also send an OACK to a client subsequently to the start of a subnet broadcast transfer. This is done to indicate to the client whether it is the master client in a subnet broadcast file group. An OACK packet that the server sends subsequently to the start of a subnet broadcast transfer includes the sbroadcast option.

The following is a list of supported options and their descriptions:

blksize

Null (0h) terminated *blksize* keyword that is followed by the block size that is used for this file transfer. It is represented as a null-terminated ASCII string. This is the response to a requested block size, and the value returned here can be less than the requested block size. The server determines the block size for the transfer based on the requested block size, the maximum configured block size, and possibly the subnet broadcast transfers that are already in progress.

sbroadcast

Null-terminated *sbroadcast* keyword that is followed by a null-terminated ASCII string that includes the following fields separated by commas:

port

The ASCII representation of the port to which the subnet-directed broadcast datagrams are broadcast. This is the well-known port that is registered with the Internet Assigned Number Authority (IANA) with the keyword of *subntbcst_ftp* and a decimal value of 247. This field might be empty in OACK packets that the server sends subsequently to the start of a subnet broadcast transfer.

sbid

The ASCII representation of a decimal number that is called the *subnet broadcast identifier*. Possible values are 0 through 4,294,967,295 (FFFFFFFFh). This is used along with the server source port to determine if a subnet-directed broadcast datagram is part of a requested transfer. This field can be empty in OACK packets that the server sends subsequently to the start of a subnet-based broadcast transfer.

mc

This is either an ASCII (31h) **1** or ASCII (32h) **0** to indicate to the client whether it is currently the master client. A value of **1** indicates that the client is the master client, and a value of **0** indicates that the client is not the master client.

In response to an OACK, the master client must send an ACK to the server. The master client sets the block number in this ACK to the number of the block prior to the first block that is required by the master client.

The master client acknowledges subnet broadcast data (BDATA) packets by sending an ACK to the server. The master client sets the block number in this ACK to the block prior to the current block that the master client requires.

Clients that are not indicated as being the master client respond to an OACK packet with an ACK that has the block number set to zero.

Note: The block number in ACK packets is the 2-byte binary representation of the number in network byte order.

tsize

The null-terminated *tsize* keyword that is followed by the null-terminated ASCII representation of the decimal number that represents the file size of the requested file. The client uses this information to ensure that it has enough space to store the file and to determine the last block number of the file.

Note: The client can also determine the file size and last block of a transfer when it receives a block that contains less data than the block size.

Server to Client Broadcast Data (BDATA) Packets

The following is a list of the fields in a Broadcast Data Packet and their descriptions:

block#

2-byte binary number in the network byte order that indicates the number of a particular block of data.

sbid

4-byte binary number in the network byte order that is called the *subnet broadcast identification*. This must be compared with the *sbid* that was returned in the OACK response to a read request (RRQ) with the Subnet Broadcast option. Along with the source port, this uniquely identifies a Subnet Broadcast File Transfer. The source port of the BDATA packet must be compared with the source port of the initial OACK packet that was received for this transfer. Only BDATA packets that match on both the SBID and source ports are considered part of the requested transfer. All other BDATA packets must be ignored.

data

This is the data for this block of the file transfer. With the exception of the last block of the file, the size of the data is equal to the block size for the transfer. The last block of the file must be less than the block size, even if it means that the length of the data in the last block is zero. However, the server might not be done broadcasting blocks after the last block of the file is broadcast. Control can be transferred to another client in the subnet broadcast file group that has not yet received all the blocks in the file.

Exit points for controlling TFTP server

With the use of exit programs, the experienced programmer can create customized processing during an application. If the TFTP server finds a program registered to one of the exit points for the server, it calls that program using parameters that are defined by the exit point.

An exit point is a specific point in the TFTP program where control may pass to an exit program. An exit program is a program to which the exit point passes control.

For each exit point, there is an associated programming interface, called an exit point interface. The exit point uses this interface to pass information between the TFTP application and the exit program. Each exit point has a unique name. Each exit point interface has an exit point format name that defines how information is passed between the TFTP application and the customer-written exit program.

Different exit points may share the same exit point interface. When this is the case, multiple exit points can call a single exit program.

To allow the exit programs to work properly, you must Install and register your exit point programs. If your programs are no longer needed, you must properly Remove the exit point programs to prevent their future functioning. For more information about using exit programs, see Using server exit programs.

Exit point performance

The following table lists exit points that give you additional control over the TFTP server.

Table 1. TFTP exit point and format

TCP/IP exit point	Application	Exit point format	Brief description
-------------------	-------------	-------------------	-------------------

Table 1. TFTP exit point and format (continued)

QIBM_QTOD_SERVER_REQ	TFTP	VLRQ0100	The TCP/IP request validation exit point provides additional control for restricting an operation.
----------------------	------	----------	----------------------------------------------------------------------------------------------------

Note:

- The same interface format is used for request validation for the FTP client, FTP server, REXEC server, and TFTP server. This allows the use of one exit program for request validation of any combination of these applications.
- The same interface format is used for server log-on processing for the FTP server and TFTP server applications. This allows the use of one exit program to process log-on requests for both of these applications.

Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

- | IBM Director of Licensing
- | IBM Corporation
- | North Castle Drive Armonk, NY 10504-1785
- | U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

- | IBM World Trade Asia Corporation
- | Licensing
- | 2-31 Roppongi 3-chome, Minato-ku
- | Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

- | IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

- | IBM Corporation
- | Software Interoperability Coordinator, Department 49XA

| 3605 Highway 52 N
| Rochester, MN 55901
| U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

| The licensed program described in this information and all licensed material available for it are provided
| by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement,
| IBM License Agreement for Machine Code, or any equivalent agreement between us.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

Application System/400
AS/400
e(logo)server
IBM
iSeries
Operating System/400
OS/400
400

Other company, product, and service names may be trademarks or service marks of others.

Terms and conditions for downloading and printing publications

| Permissions for the use of the information you have selected for download are granted subject to the
| following terms and conditions and your indication of acceptance thereof.

| **Personal Use:** You may reproduce this information for your personal, noncommercial use provided that
| all proprietary notices are preserved. You may not distribute, display or make derivative works of this
| information, or any portion thereof, without the express consent of IBM®.

| **Commercial Use:** You may reproduce, distribute and display this information solely within your
| enterprise provided that all proprietary notices are preserved. You may not make derivative works of this
| information, or reproduce, distribute or display this information or any portion thereof outside your
| enterprise, without the express consent of IBM.

| Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either
| express or implied, to the information or any data, software or other intellectual property contained
| therein.

| IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of
| the information is detrimental to its interest or, as determined by IBM, the above instructions are not
| being properly followed.

| You may not download, export or re-export this information except in full compliance with all applicable
| laws and regulations, including all United States export laws and regulations. IBM MAKES NO
| GUARANTEE ABOUT THE CONTENT OF THIS INFORMATION. THE INFORMATION IS PROVIDED

| "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING
| BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT,
| AND FITNESS FOR A PARTICULAR PURPOSE.

All material copyrighted by IBM Corporation.

| By downloading or printing information from this site, you have indicated your agreement with these
| terms and conditions.



Printed in USA