Network Working Group
Request for Commments: 2347
Updates: 1350
Obsoletes: 1782
Category: Standards Track

G. Malkin
Bay Networks
A. Harkin
Hewlett Packard Co.

May 1998

TFTP Option Extension

Status of this Memo

Copyright Notice

Abstract

The Trivial File Transfer Protocol [1] is a simple, lock-step, file transfer protocol which allows a client to get or put a file onto a remote host. This document describes a simple extension to TFTP to allow option negotiation prior to the file transfer.

Introduction

The option negotiation mechanism proposed in this document is a backward-compatible extension to the TFTP protocol. It allows file transfer options to be negotiated prior to the transfer using a mechanism which is consistent with TFTP's Request Packet format. The mechanism is kept simple by enforcing a request-respond-acknowledge sequence, similar to the lock-step approach taken by TFTP itself.

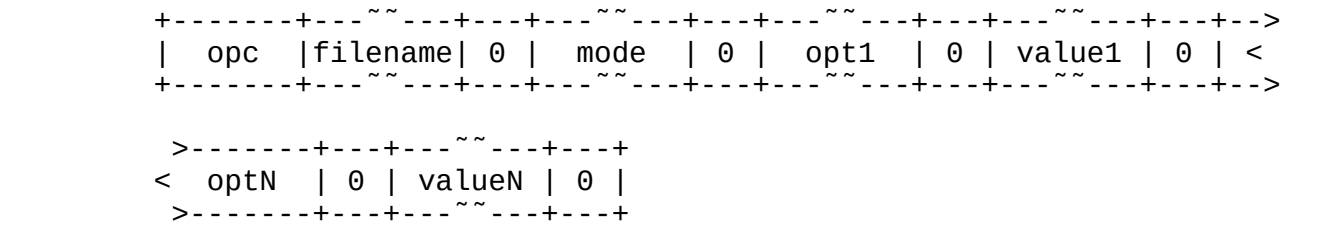
While the option negotiation mechanism is general purpose, in that many types of options may be negotiated, it was created to support the Blocksize option defined in [2]. Additional options are defined in [3].

Packet Formats

TFTP options are appended to the Read Request and Write Request packets. A new type of TFTP packet, the Option Acknowledgment (OACK), is used to acknowledge a client's option negotiation request. A new error code, 8, is hereby defined to indicate that a transfer

should be terminated due to option negotiation.

Options are appended to a TFTP Read Request or Write Request packet as follows:



opc
: The opcode field contains either a 1, for Read Requests, or 2, for Write Requests, as defined in [1].

filename
: The name of the file to be read or written, as defined in [1]. This is a NULL-terminated field.

mode
: The mode of the file transfer: "netascii", "octet", or "mail", as defined in [1]. This is a NULL-terminated field.

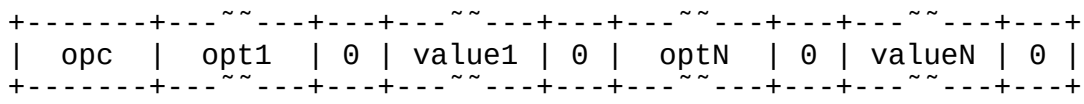
opt1
: The first option, in case-insensitive ASCII (e.g., blksize). This is a NULL-terminated field.

value1
: The value associated with the first option, in case-insensitive ASCII. This is a NULL-terminated field.

optN, valueN
: The final option/value pair. Each NULL-terminated field is specified in case-insensitive ASCII.

The options and values are all NULL-terminated, in keeping with the original request format. If multiple options are to be negotiated, they are appended to each other. The order in which options are specified is not significant. The maximum size of a request packet is 512 octets.

The OACK packet has the following format:



opc
The opcode field contains a 6, for Option Acknowledgment.

opt1
The first option acknowledgment, copied from the original request.

value1
The acknowledged value associated with the first option. If and how this value may differ from the original request is detailed in the specification for the option.

optN, valueN
The final option/value acknowledgment pair.

Negotiation Protocol

The client appends options at the end of the Read Request or Write request packet, as shown above. Any number of options may be specified; however, an option may only be specified once. The order of the options is not significant.

If the server supports option negotiation, and it recognizes one or more of the options specified in the request packet, the server may respond with an Options Acknowledgment (OACK). Each option the server recognizes, and accepts the value for, is included in the OACK. Some options may allow alternate values to be proposed, but this is an option specific feature. The server must not include in the OACK any option which had not been specifically requested by the client; that is, only the client may initiate option negotiation. Options which the server does not support should be omitted from the OACK; they should not cause an ERROR packet to be generated. If the value of a supported option is invalid, the specification for that option will indicate whether the server should simply omit the option from the OACK, respond with an alternate value, or send an ERROR packet, with error code 8, to terminate the transfer.

An option not acknowledged by the server must be ignored by the client and server as if it were never requested. If multiple options were requested, the client must use those options which were acknowledged by the server and must not use those options which were not acknowledged by the server.

When the client appends options to the end of a Read Request packet, three possible responses may be returned by the server:

OACK - acknowledge of Read Request and the options;

DATA - acknowledge of Read Request, but not the options;

ERROR - the request has been denied.

When the client appends options to the end of a Write Request packet, three possible responses may be returned by the server:

OACK - acknowledge of Write Request and the options;

ACK - acknowledge of Write Request, but not the options;

ERROR - the request has been denied.

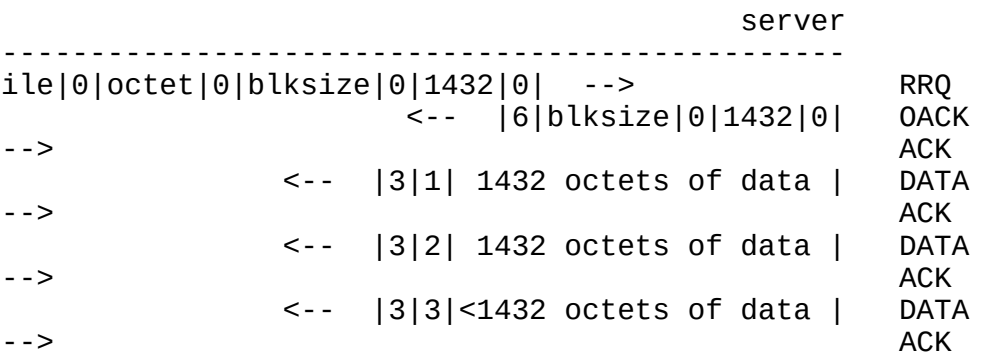
If a server implementation does not support option negotiation, it will likely ignore any options appended to the client's request. In this case, the server will return a DATA packet for a Read Request and an ACK packet for a Write Request establishing normal TFTP data transfer. In the event that a server returns an error for a request which carries an option, the client may attempt to repeat the request without appending any options. This implementation option would handle servers which consider extraneous data in the request packet to be erroneous.

Depending on the original transfer request there are two ways for a client to confirm acceptance of a server's OACK. If the transfer was initiated with a Read Request, then an ACK (with the data block number set to 0) is sent by the client to confirm the values in the server's OACK packet. If the transfer was initiated with a Write Request, then the client begins the transfer with the first DATA packet, using the negotiated values. If the client rejects the OACK, then it sends an ERROR packet, with error code 8, to the server and the transfer is terminated.

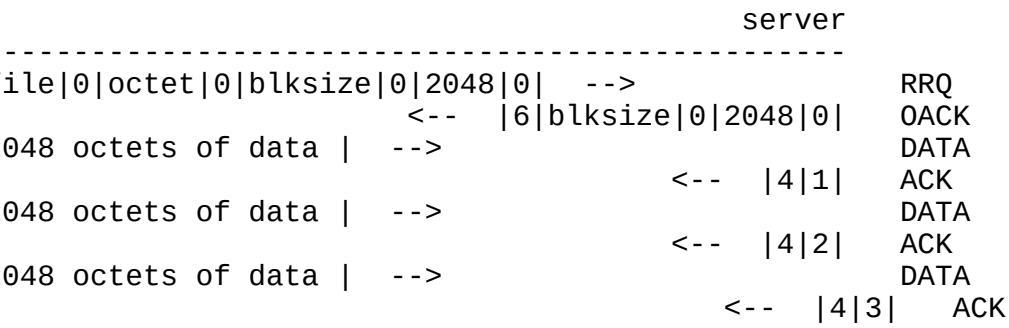
Once a client acknowledges an OACK, with an appropriate non-error response, that client has agreed to use only the options and values returned by the server. Remember that the server cannot request an option; it can only respond to them. If the client receives an OACK containing an unrequested option, it should respond with an ERROR packet, with error code 8, and terminate the transfer.

Examples

Read Request



Write Request



Security Considerations

The basic TFTP protocol has no security mechanism. This is why it has no rename, delete, or file overwrite capabilities. This document does not add any security to TFTP; however, the specified extensions do not add any additional security risks.

References

[1] Sollins, K., "The TFTP Protocol (Revision 2)", STD 33, RFC 1350, October 1992.

[2] Malkin, G., and A. Harkin, "TFTP Blocksize Option", RFC 2348, May 1998.

[3] Malkin, G., and A. Harkin, "TFTP Timeout Interval and Transfer Size Options", RFC 2349, May 1998.

Authors' Addresses

Gary Scott Malkin
Bay Networks
8 Federal Street
Billerica, MA 01821

Phone: (978) 916-4237
EMail: gmalkin@baynetworks.com

Art Harkin
Internet Services Project
Information Networks Division
19420 Homestead Road MS 43LN
Cupertino, CA 95014

Phone: (408) 447-3755
EMail: ash@cup.hp.com

Full Copyright Statement