[MS-ISTD]:
iSCSI Software Target Discovery Protocol Specification

Revision Summary

Date	Revision History	Revision Class	Comments
08/27/2010	0.1	New	Released new document.
10/08/2010	0.1.1	Editorial	Changed language and formatting in the technical content.
11/19/2010	0.1.1	No change	No changes to the meaning, language, or formatting of the technical content.
01/07/2011	0.1.1	No change	No changes to the meaning, language, or formatting of the technical content.
02/11/2011	0.1.1	No change	No changes to the meaning, language, or formatting of the technical content.

Contents

1 Introduction

The iSCSI Software Target Discovery Protocol is specified in this document. It is used to discover **iSCSI Software Targets** running on a network, which can service requests for block-level storage. This protocol uses a **multicast mailslot** protocol message and **unicast** mailslot replies to discover the host names of servers that are running **iSCSI targets**.

The **Internet SCSI (iSCSI)** is an industry-standard protocol defined by [RFC3720]. It provides block-level storage access to the local disks of an iSCSI Software Target computer over a **TCP**/IP network from a remote location. For more information about iSCSI Software Targets, see [MS-ISTM], which describes the support for iSCSI Software Target management from an administrative command line tool or user interface tool.

1.1 Glossary

The following terms are defined in [MS-GLOS]:

broadcast
domain
domain name (2)
Domain Name System (DNS)
fully qualified domain name (FQDN) (1)
Internet SCSI (iSCSI)
little-endian
mailslot (1)
multicast
NetBIOS
NetBIOS name
Transmission Control Protocol (TCP)
unicast
UTF-16LE (Unicode Transformation Format, 16-bits, little-endian)
Virtual Disk Service (VDS) provider

The following terms are specific to this document:

block storage discoverer: The component of the ISTD protocol that initiates the discovery of block storage services on a network.

block storage service: The component of the ISTD protocol that can service requests for block-level storage on a network. Block storage services respond to broadcast messages from a block storage discoverer.

high-availability (HA) cluster: An operational configuration of computers in which a pool of machines, called nodes, each one running an iSCSI Software Target instance, share configuration information, storage and networking resources, and provide seamless services to network-based clients, which are iSCSI initiators in the context of this specification.

iSCSI initiator: A computer that requests access to a remote storage device. An iSCSI initiator issues small computer system interface (SCSI) commands to request services from components, which are logical units of a server known as "targets". For more information concerning iSCSI initiators and targets, see [RFC3720].

iSCSI Software Target: A software implementation of the iSCSI technology specified in [RFC3720], along with supporting functionality.

iSCSI target: The entity that grants and facilitates access to a storage device, as specified in [RFC3720].

ISTD: The iSCSI Software Target Discovery Protocol.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[MS-MAIL] Microsoft Corporation, "Remote Mailslot Protocol Specification", March 2007.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.ietf.org/rfc/rfc2119.txt

1.2.2 Informative References

[MSDN-VDS] Microsoft Corporation, "About VDS", http://msdn.microsoft.com/en-us/library/aa381442(VS.85).aspx

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary", March 2007.

[MS-ISTM] Microsoft Corporation, "iSCSI Software Target Management Protocol Specification", August 2010.

[RFC3720] Satran, J., Meth, K., Sapuntzakis, C., et al., "Internet Small Computer Systems Interface (iSCSI)", RFC 3720, April 2004, http://www.ietf.org/rfc/rfc3720.txt

1.3 Overview

The iSCSI Software Target Discovery Protocol enables the discovery of **block storage services** that are available on a local network to service application requests for block-level storage. These services are often, though not always, implemented as iSCSI targets ([MS-ISTM]). The computer that discovers the block storage services and the computers that provide the block storage services are connected to a local network that shares a common naming resolution infrastructure.

The following diagram shows a typical configuration for **ISTD**:

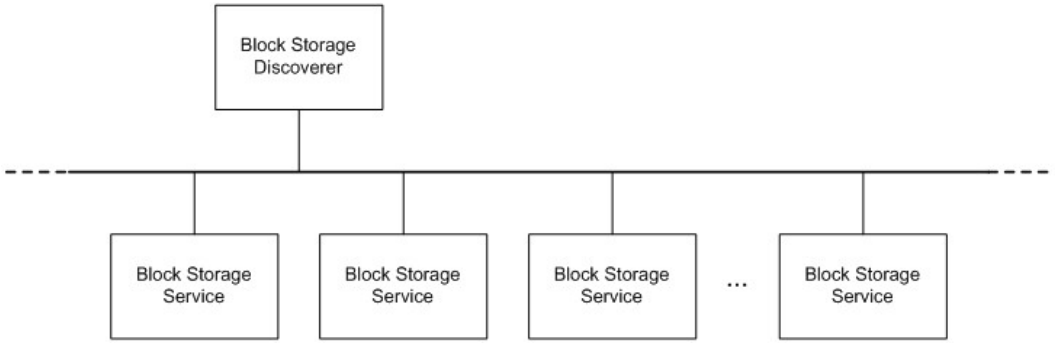


Figure 1: A local network environment for block storage discovery

The entity that discovers block storage services on a network is called the **block storage discoverer**. While the block storage discoverer can run on any computer on a network, it usually runs on the computer supporting applications that require information about block storage services. An example of such a computer is an **iSCSI initiator** running a **Virtual Disk Service (VDS) provider** similar to the one described in [MSDN-VDS].

The block storage discoverer sends a broadcast mailslot message onto a network, addressed to a well-known mailslot, as specified in section 3.2.5.1. Every block storage service that receives this broadcast message responds with another mailslot message that contains information about the computer that is running the block storage service, as specified in sections 3.3.5.1 and 3.2.5.2. The received information can be saved by the block storage discoverer for future use, as specified in section 3.3.5.2.

For details on the ISTD message exchange, see section 3.1.

1.4 Relationship to Other Protocols

The iSCSI Software Target Discovery Protocol depends on the Remote Mailslot Protocol [MS-MAIL]. The transport layers for this protocol are shown in the following diagram:

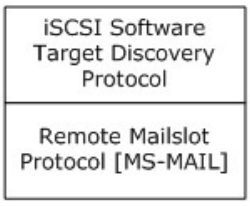


Figure 2: ISTD protocol layering diagram

No protocol depends on ISTD.

1.5 Prerequisites/Preconditions

The iSCSI Software Target Discovery Protocol is implemented over the Remote Mailslot Protocol and has the prerequisites described in [MS-MAIL] section 1.5.

1.6 Applicability Statement

The iSCSI Software Target Discovery Protocol is applicable to scenarios that require the discovery of block storage services on a network. An example of such a scenario is a VDS provider similar to that described in [MSDN-VDS].

1.7 Versioning and Capability Negotiation

The iSCSI Software Target Discovery Protocol does not perform versioning and capability negotiation.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

The iSCSI Software Target Discovery Protocol uses the standard assignments shown in the following table.

Parameter	Value	Reference
The mailslot used by block storage services to listen for the block storage discoverer **broadcast**.	\MAILSLOT\WINTARGET	[MS-MAIL] section 3.1.4.1
The mailslot used by the block storage discoverer to listen for service location replies from block storage services.	\MAILSLOT\WTVDSPROV	[MS-MAIL] section 3.1.4.1

2 Messages

2.1 Transport

The iSCSI Software Target Discovery Protocol MUST use the Remote Mailslot Protocol transfer service, as specified in [MS-MAIL].

The ISTD block storage discoverer MUST create a mailslot named "\MAILSLOT\WTVDSPROV" on the network adapters that are configured to run the **NetBIOS** services required for the mailslot. This mailslot is used to receive WT_VDSPROV_MAILSLOT_MESSAGE messages section 2.2.2).

Each block storage service on the network MUST create a mailslot named "\MAILSLOT\WINTARGET" on the network adapters that are configured to run the NetBIOS services required for the mailslot. This mailslot is used to receive WT_MAILSLOT_INFO messages (section 2.2.1).

2.2 Message Syntax

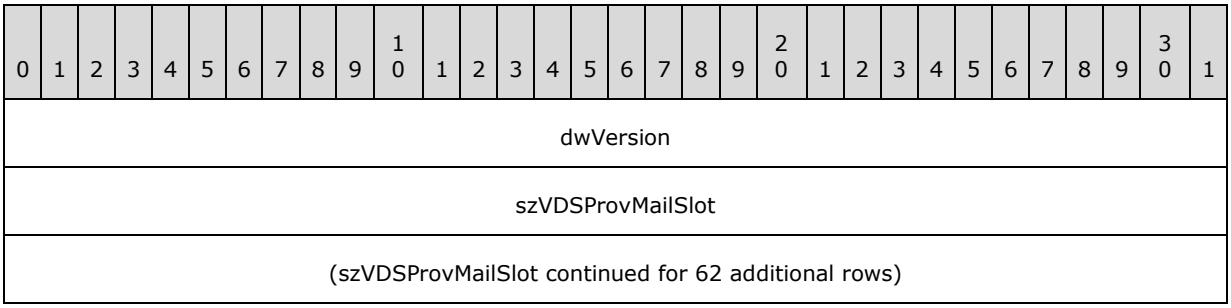
The following statements apply to messages of iSCSI Software Target Discovery Protocol unless otherwise specified:

- Strings use **UTF-16LE (Unicode Transformation Format, 16-bits, little-endian)** encoding.
- Messages are transmitted in **little-endian** byte order.

2.2.1 WT_MAILSLOT_INFO

The **WT_MAILSLOT_INFO** message is sent as a broadcast message from the block storage discoverer.

The following packet diagram shows the format of the WT_MAILSLOT_INFO message.



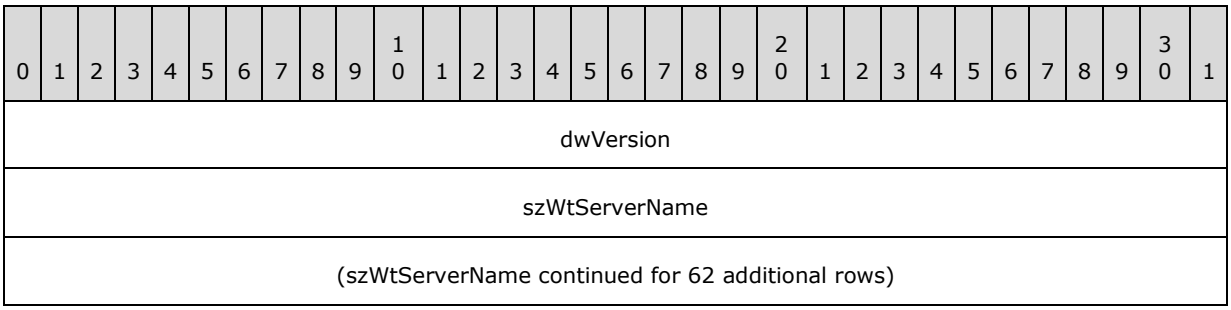
dwVersion (4 bytes): The version of the WT_MAILSLOT_INFO message. This value MUST be 0x00000001.

szVDSProvMailSlot (256 bytes): A null-terminated string that specifies the mailslot destination of the block storage discoverer. The value of this field is in the form "*computer_name*\MAILSLOT\WTVDSPROV", where *computer_name* is a computer name in a given network name resolution **domain** of the computer running the block storage discoverer. The *computer_name* SHOULD be the **DNS** name of the computer.

Note Unused bytes in this field MUST be set to zero.

2.2.2 WT_VDSPROV_MAILSLOT_MESSAGE

The WT_VDSPROV_MAILSLOT_MESSAGE message is a response to the WT_MAILSLOT_INFO message (section 2.2.1). Each block storage service on the network that receives a WT_MAILSLOT_INFO message, and that can service block-level storage requests, SHOULD send a WT_VDSPROV_MAILSLOT_MESSAGE message to the mailslot identified in the WT_MAILSLOT_INFO message.



dwVersion (4 bytes): The version of the WT_MAILSLOT_INFO message. This value MUST be 1.

szWtServerName (256 bytes): A null-terminated string that specifies a computer name in a given network name resolution domain. The computer name SHOULD be the **fully qualified domain name (FQDN)** that uniquely identifies the computer of the block storage service**.**

If the local computer is a node in a **high-availability (HA) cluster**, the value in this field is the FQDN of the local computer, not the name of the HA cluster virtual server. The FQDN is a combination of the DNS host name and the DNS **domain name**, using the form *HostName*.*DomainName*.

Note Unused bytes in this field MUST be set to zero.

3 Protocol Details

3.1 Common Details

This section specifies details that are common to the client and server roles of iSCSI Software Target Discovery Protocol.

The ISTD messaging sequence consists of a single broadcast message from the block storage discoverer to a well-known mailslot on the network. All computers on the network that offer block storage services and that are ready to service block-level storage requests SHOULD respond to the broadcast message with another message. The following sequence diagrams capture these messaging sequences.

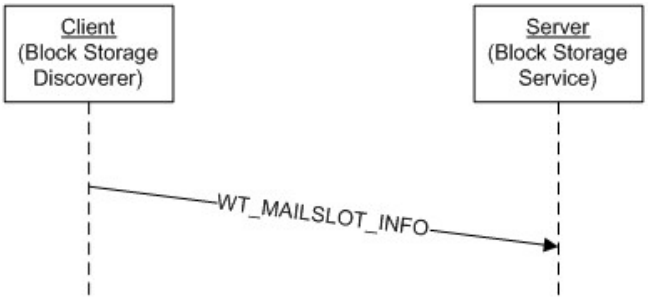


Figure 3: The WT_MAILSLOT_INFO broadcast message

In the preceding message sequence, the block storage discoverer performs the client role and sends the **WT_MAILSLOT_INFO** message (section 2.2.1) as a broadcast message to the network. Each of the block storage service computers on the network that can service block-level storage requests receives and processes the broadcast message. See sections 3.2.5.1 and 3.3.5.1 for processing details.

Figure 4: The WT_VDSPROV_MAILSLOT_MESSAGE response message

In the preceding message sequence, each block storage service computer on the network performs the client role and sends a response to the block storage discoverer in the form of the **WT_VDSPROV_MAILSLOT_MESSAGE** message (section 2.2.2). The block storage discoverer processes each of the response messages as it receives them. See sections 3.2.5.2 and 3.3.5.2 for processing details.

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the

explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

The following non-persisted data element is common to the client and server roles defined in the ISTD protocol:

BlockStorageDiscovererMailslot: A null-terminated string that is at least 128 characters long. This element is used by a block storage service computer to store the mailslot destination of the block storage discoverer, which was specified in the **szVDSProvMailSlot** field of a **WT_MAILSLOT_INFO** message (section 2.2.1).

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

None.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Client Details

This section specifies details of the client roles in iSCSI Software Target Discovery Protocol.

3.2.1 Abstract Data Model

See section 3.1.1 for data used by a block storage service when it is acting in the client role.

3.2.2 Timers

None.

3.2.3 Initialization

On startup, the block storage discoverer configures its host name, using any configured network name resolution domain, including DNS names or **NetBIOS names**, and it makes the host name available to applications and services that are running on the computer. The block storage discoverer uses the host name during **WT_MAILSLOT_INFO** message client processing (section 3.2.5.1).

On startup, each block storage service configures its host name, using any configured network name resolution domain, including DNS names or NetBIOS names, and it makes the host name available to applications and services that are running on the computer. The block storage services use the host name during **WT_VDSPROV_MAILSLOT_MESSAGE** message client processing (section 3.2.5.2).

3.2.4 Higher-Layer Triggered Events

After initialization is complete (sections 3.2.3 and 3.3.3), the block storage discoverer creates and broadcasts a discovery message onto the network (section 3.2.5.1).

During normal operation of the block storage discoverer, a rediscovery of block storage services on the network can be triggered by an external application that is utilizing the services of the block storage discoverer.

3.2.5 Message Processing Events and Sequencing Rules

3.2.5.1 WT_MAILSLOT_INFO Processing

The block storage discoverer performs the client role for the **WT_MAILSLOT_INFO** message (section 2.2.1).

The client creates a WT_MAILSLOT_INFO packet and sends it as a broadcast to the mailslot destination that has been designated for the ISTD broadcast message, "\MAILSLOT\WINTARGET".

3.2.5.2 WT_VDSPROV_MAILSLOT_MESSAGE Processing

Block storage services perform the client role for the **WT_VDSPROV_MAILSLOT_MESSAGE** message (section 2.2.2). This message is sent in response to a **WT_MAILSLOT_INFO** broadcast message (section 2.2.1), as specified in section 3.3.5.1.

The client SHOULD process the message as follows:

- Saves the mailslot destination that was obtained from processing the WT_MAILSLOT_INFO message in a **BlockStorageDiscovererMailslot** data element (section 3.1.1).
- Creates a WT_VDSPROV_MAILSLOT_MESSAGE packet and sends it to the mailslot destination.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

3.3 Server Details

This section specifies details of the server roles in iSCSI Software Target Discovery Protocol.

3.3.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations

adhere to this model as long as their external behavior is consistent with that described in this document.

The following non-persisted data is defined in the server abstract data model:

BlockStorageServerNameCollection: A collection of **BlockStorageServerName** elements that have been discovered by the block storage discoverer. This collection can be made available to external applications to find block storage services on the network.

BlockStorageServerName: A null-terminated string that is at least 128 characters long. This element is used by the block storage discoverer to store the host name that is specified in the **szWtServerName** field of a **WT_VDSPROV_MAILSLOT_MESSAGE** message (section 2.2.2). The host name SHOULD be the fully qualified domain name (FQDN) of the block storage service computer that sent the message.

3.3.2 Timers

The block storage discoverer MAY<1> maintain a timer that controls how long it waits for responses to the **WT_MAILSLOT_INFO** broadcast message (section 2.2.1); that is, the length of time spent listening for **WT_VDSPROV_MAILSLOT_MESSAGE** messages (section 2.2.2).

3.3.3 Initialization

On startup, the block storage discoverer creates a mailslot (section 2.1) in which to receive **WT_VDSPROV_MAILSLOT_MESSAGE** messages (section 2.2.2).

On startup, each block storage service creates a mailslot (section 2.1) in which to receive **WT_MAILSLOT_INFO** messages (section 2.2.1).

3.3.4 Higher-Layer Triggered Events

None.

3.3.5 Message Processing Events and Sequencing Rules

3.3.5.1 WT_MAILSLOT_INFO Processing

Block storage services perform the server role for the **WT_MAILSLOT_INFO** message (section 2.2.1).

The server SHOULD listen on the mailslot that it created for ISTD messages (section 3.3.3). The server SHOULD process each message as follows:

- Verifies that the size of the message is correct, according to the packet definition.
- Verifies that the version of the protocol that sent the message is compatible with the version of the protocol on the server, using the **dwVersion** field of the message.
- Extracts the mailslot for the block storage discoverer from the **szVDSProvMailSlot** field and stores it in a **BlockStorageDiscovererMailslot** data element (section 3.1.1).
- Begins the process of informing the block storage discoverer that the server exists on the network and is available to service block-level storage requests (section 3.2.5.2).

3.3.5.2 WT_VDSPROV_MAILSLOT_MESSAGE Processing

The block storage discoverer performs the server role for the **WT_VDSPROV_MAILSLOT_MESSAGE** message (section 2.2.2).

The server SHOULD listen on the mailslot that it created for ISTD messages (section 3.3.3). The server SHOULD process each message that is received as follows:

- Verifies that the size of the message is correct, according to the packet definition.
- Verifies that the version of the protocol that sent the message is compatible with the version of the protocol on the server, using the **dwVersion** field of the message.
- Extracts the host name of the block storage service computer that sent the message from the **szWtServerName** field and stores it in a **BlockStorageServerName** data element (section 3.3.1).
- Adds the **BlockStorageServerName** element to the collection that is represented by the **BlockStorageServerNameCollection** data object.

3.3.6 Timer Events

If the block storage discoverer maintains a listening timer, as specified in section 3.3.2, the expiration of that timer MAY<2> trigger the closing of the mailslot that was created to listen for **WT_VDSPROV_MAILSLOT_MESSAGE** messages (section 2.2.2).

3.3.7 Other Local Events

None.

4 Protocol Examples

This section describes a typical message sequence of block storage discovery by iSCSI Software Target Discovery Protocol. The following diagram illustrates a possible message sequence:

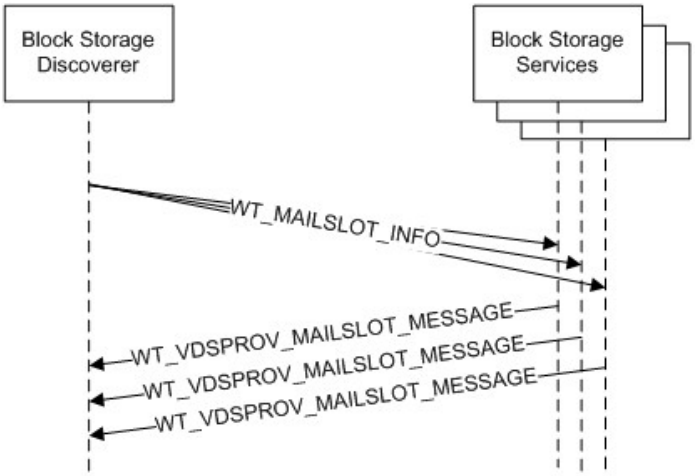


Figure 5: An ISTD protocol message sequence

The sections that follow show examples of network traffic from an actual ISTD protocol message sequence, with a broadcast by a block storage discoverer and a response by a block storage service represented.

4.1 The WT_MAILSLOT_INFO Broadcast

The following example of network traffic captures an ISTD block storage discoverer broadcast of a **WT_MAILSLOT_INFO** message (section 2.2.1) to the well-known mailslot name "\MAILSLOT\WINTARGET". The name of the computer running the block storage discoverer is "client-computer".

```
Smb: C; Transaction, Mail Slots, Write Mail Slot, FileName = \MAILSLOT\WINTARGET
   MailSlotsSetupWords:
   ByteCount: 283 (0x11B)
   MailSlotsBuffer:
      FileName: \MAILSLOT\WINTARGET
      Pad2: Binary Large Object (3 Bytes)
      MailSlotData: Binary Large Object (260 Bytes)
```

dwVersion: 01 00 00 00

szVDSProvMailSlot:

```
5C 00 5C 00 63 00 6C 00 69 00 65 00 6E 00 74 00   \.\.c.l.i.e.n.t.
2D 00 63 00 6F 00 6D 00 70 00 75 00 74 00 65 00   -.c.o.m.p.u.t.e.
72 00 5C 00 4D 00 41 00 49 00 4C 00 53 00 4C 00   r.\.M.A.I.L.S.L.
4F 00 54 00 5C 00 57 00 54 00 56 00 44 00 53 00   O.T.\.W.T.V.D.S.
50 00 52 00 4F 00 56 00 00 00 00 00 00 00 00 00   P.R.O.V.........
```

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
```

4.2 The WT_VDSPROV_MAILSLOT_MESSAGE Response

The following example of network traffic captures an ISTD block storage service response of a **WT_VDSPROV_MAILSLOT_MESSAGE** message (section 2.2.2) to the mailslot name that was specified in the broadcast message (section 4.1). The name of the computer running the block storage service is "SERVER-012345".

```
Smb: C; Transaction, Mail Slots, Write Mail Slot, FileName = \MAILSLOT\WTVDSPROV
   MailSlotsSetupWords:
   ByteCount: 283 (0x11B)
   MailSlotsBuffer:
      FileName: \MAILSLOT\WTVDSPROV
      Pad2: Binary Large Object (3 Bytes)
      MailSlotData: Binary Large Object (260 Bytes)
```

dwVersion: 01 00 00 00

szWtServerName:

```
53 00 45 00 52 00 56 00 45 00 52 00 2D 00 30 00   S.E.R.V.E.R.-.0.
31 00 32 00 33 00 34 00 35 00 2E 00 77 00 69 00   1.2.3.4.5...w.i.
6E 00 67 00 74 00 69 00 70 00 2E 00 74 00 6F 00   n.g.t.i.p...t.o.
79 00 73 00 2E 00 6D 00 69 00 63 00 72 00 6F 00   y.s...m.i.c.r.o.
73 00 6F 00 66 00 74 00 2E 00 63 00 6F 00 6D 00   s.o.f.t...c.o.m.
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
```

5 Security

5.1 Security Considerations for Implementers

The iSCSI Software Target Discovery Protocol is not a secure protocol. This protocol is not appropriate for applications that require secure communication between client and server.

5.2 Index of Security Parameters

None.

6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Windows Server® 2008 R2 operating system

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

<1> Section 3.3.2: The Windows implementation of the block storage discoverer does not maintain a timer; instead, it listens for WT_VDSPROV_MAILSLOT_MESSAGE messages as long as it is running.

<2> Section 3.3.6: The Windows implementation of the block storage discoverer does not maintain a timer; instead, it listens for WT_VDSPROV_MAILSLOT_MESSAGE messages as long as it is running.

7 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

8 Index

W