

# [MS-RSVD]: Remote Shared Virtual Disk Protocol

---

## Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft [Open Specification Promise](#) or the [Community Promise](#). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting [iplg@microsoft.com](mailto:iplg@microsoft.com).
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit [www.microsoft.com/trademarks](http://www.microsoft.com/trademarks).
- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

**Reservation of Rights.** All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

**Tools.** The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

## Revision Summary

Date	Revision History	Revision Class	Comments
08/08/2013	1.0	New	Released new document.
11/14/2013	2.0	Major	Significantly changed the technical content.
02/13/2014	3.0	Major	Significantly changed the technical content.
05/15/2014	4.0	Major	Significantly changed the technical content.

# Contents

<b>1 Introduction</b>	<b>6</b>
1.1 Glossary	6
1.2 References	6
1.2.1 Normative References	6
1.2.2 Informative References	7
1.3 Overview	7
1.4 Relationship to Other Protocols	7
1.5 Prerequisites/Preconditions	7
1.6 Applicability Statement	8
1.7 Versioning and Capability Negotiation	8
1.8 Vendor-Extensible Fields	8
1.9 Standards Assignments	8
<b>2 Messages</b>	<b>9</b>
2.1 Transport	9
2.2 Message Syntax	9
2.2.1 Constants	9
2.2.2 Operation Codes	9
2.2.3 Error Code	10
2.2.4 Structures	11
2.2.4.1 SVHDX_TUNNEL_CHECK_CONNECTION_REQUEST Structure	11
2.2.4.2 SVHDX_TUNNEL_CHECK_CONNECTION_RESPONSE Structure	11
2.2.4.3 SVHDX_TUNNEL_SRB_STATUS_REQUEST Structure	11
2.2.4.4 SVHDX_TUNNEL_SRB_STATUS_RESPONSE Structure	12
2.2.4.5 SVHDX_TUNNEL_DISK_INFO_REQUEST Structure	12
2.2.4.6 SVHDX_TUNNEL_DISK_INFO_RESPONSE Structure	13
2.2.4.7 SVHDX_TUNNEL_SCSI_REQUEST Structure	15
2.2.4.8 SVHDX_TUNNEL_SCSI_RESPONSE Structure	16
2.2.4.9 SVHDX_TUNNEL_VALIDATE_DISK_REQUEST Structure	17
2.2.4.10 SVHDX_TUNNEL_VALIDATE_DISK_RESPONSE Structure	18
2.2.4.11 SVHDX_TUNNEL_OPERATION_HEADER Structure	18
2.2.4.12 SVHDX_OPEN_DEVICE_CONTEXT Structure	19
2.2.4.13 SVHDX_TUNNEL_FILE_INFO_REQUEST Structure	20
2.2.4.14 SVHDX_TUNNEL_FILE_INFO_RESPONSE Structure	20
2.2.4.15 SVHDX_SHARED_VIRTUAL_DISK_SUPPORT_REQUEST Structure	21
2.2.4.16 SVHDX_SHARED_VIRTUAL_DISK_SUPPORT_RESPONSE Structure	21
2.2.5 SRB Status Code	21
<b>3 Protocol Details</b>	<b>23</b>
3.1 Client Details	23
3.1.1 Abstract Data Model	23
3.1.1.1 Global	23
3.1.2 Timers	23
3.1.3 Initialization	23
3.1.4 Higher-Layer Triggered Events	23
3.1.4.1 Sending Any Outgoing Message	23
3.1.4.2 Application Requests Opening a Shared Virtual Disk	23
3.1.4.3 Application Requests Closing a Shared Virtual Disk	24
3.1.4.4 Application Requests Reading From a Shared Virtual Disk	25
3.1.4.5 Application Requests Writing To a Shared Virtual Disk	25

3.1.4.6	Application Requests Virtual Disk File information .....	25
3.1.4.7	Application Requests Connection Status .....	26
3.1.4.8	Application Requests Shared Virtual Disk Information.....	26
3.1.4.9	Application Requests Execution of SCSI Command.....	27
3.1.4.10	Application Requests to Validate a Shared Virtual Disk.....	28
3.1.4.11	Application Requests Querying Shared Virtual Disk Support.....	28
3.1.5	Message Processing Events and Sequencing Rules.....	29
3.1.5.1	Receiving an Open Response.....	29
3.1.5.2	Receiving a Close Response .....	29
3.1.5.3	Receiving a Read Response.....	29
3.1.5.4	Receiving a Write Response .....	29
3.1.5.5	Receiving a Virtual Disk File Information Response .....	30
3.1.5.6	Receiving a Connection Status Response .....	30
3.1.5.7	Receiving a Shared Virtual Disk Information Request .....	30
3.1.5.8	Receiving a SCSI Command Response.....	30
3.1.5.9	Receiving a Validate Disk Response .....	31
3.1.5.10	Receiving a Shared Virtual Disk Support Response .....	31
3.1.6	Timer Events .....	31
3.1.7	Other Local Events .....	31
3.2	Server Details .....	31
3.2.1	Abstract Data Model .....	31
3.2.1.1	Global .....	31
3.2.1.2	Per Open .....	31
3.2.1.3	Per SenseError in SenseErrorDataList.....	31
3.2.2	Timers .....	32
3.2.3	Initialization .....	32
3.2.4	Higher-Layer Triggered Events.....	32
3.2.4.1	Handling a Tunnel Operation Request.....	32
3.2.4.2	Handling a Shared Virtual Disk Support Request.....	32
3.2.5	Message Processing Events and Sequencing Rules.....	32
3.2.5.1	Receiving an Open Request.....	32
3.2.5.2	Receiving a Close Request .....	33
3.2.5.3	Receiving a Read Request.....	33
3.2.5.4	Receiving a Write Request .....	34
3.2.5.5	Receiving a Tunnel Operation Request.....	35
3.2.5.5.1	Receiving a Virtual Disk File Information Request .....	36
3.2.5.5.2	Receiving a Connection Status Request.....	36
3.2.5.5.3	Receiving a Sense Code Request.....	37
3.2.5.5.4	Receiving a Shared Virtual Disk Information Request.....	38
3.2.5.5.5	Receiving a SCSI Command Request .....	38
3.2.5.5.6	Receiving a Validate Disk Request .....	40
3.2.5.6	Receiving a Query Shared Virtual Disk Support Request .....	40
3.2.6	Timer Events .....	40
3.2.7	Other Local Events .....	40
<b>4</b>	<b>Protocol Examples.....</b>	<b>41</b>
4.1	Retrieving Virtual Disk File Information.....	41
4.2	Executing a SCSI Command.....	42
<b>5</b>	<b>Security.....</b>	<b>46</b>
5.1	Security Considerations for Implementers.....	46
5.2	Index of Security Parameters .....	46

**6 Appendix A: Product Behavior ..... 47**  
**7 Change Tracking..... 48**  
**8 Index ..... 58**

# 1 Introduction

The Remote Shared Virtual Disk (RSVD) Protocol is a block-based protocol that is used to access the virtual disk file in the network over SMB3.

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in RFC 2119. Sections 1.5 and 1.9 are also normative but cannot contain those terms. All other sections and examples in this specification are informative.

## 1.1 Glossary

The following terms are defined in [\[MS-GLOS\]](#):

**GUID**  
**SCSI**  
**Unicode**

The following terms are specific to this document:

**persistent reservation:** A SCSI feature supporting control operations for shared devices ([SPC-3] section 5.6).

**SCSI command descriptor block:** A block of information that describes the SCSI command.

**sense data:** Data describing command-completed information that a device server delivers to an application client in the same structure as the status or as parameter data in response to an SCSI command.

**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

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the documents, which are updated frequently. References to other documents include a publishing year when one is available.

### 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](mailto:dochelp@microsoft.com). We will assist you in finding the relevant information.

[MS-SMB2] Microsoft Corporation, "[Server Message Block \(SMB\) Protocol Versions 2 and 3](#)".

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

[UNICODE] The Unicode Consortium, "Unicode Home Page", 2006, <http://www.unicode.org/>

## 1.2.2 Informative References

[MS-GLOS] Microsoft Corporation, "[Windows Protocols Master Glossary](#)".

[SPC-3] International Committee on Information Technology Standards, "SCSI Primary Commands - 3 (SPC-3)", Project T10/1416-D, May 2005, <http://www.t10.org/cgi-bin/ac.pl?t=f&f=/spc3r23.pdf>

## 1.3 Overview

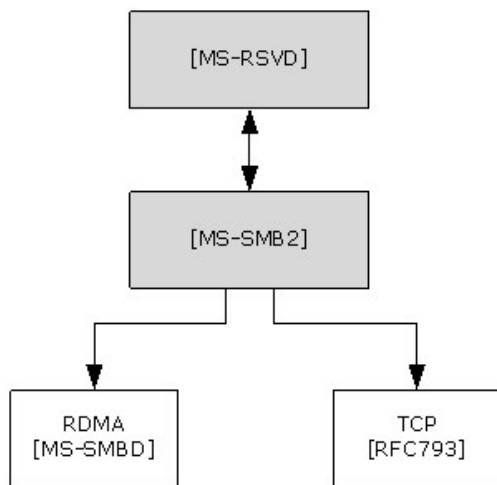
The Remote Shared Virtual Disk Protocol enables a client application to access virtual disk files in a shared fashion on a remote server.

The RSVD protocol supports the following features:

- Allowing a client to open a shared virtual disk on a remote share.
- Reading, writing, or closing shared virtual disk files on the target server.
- Forwarding of raw SCSI commands and receipt of their results.

## 1.4 Relationship to Other Protocols

This protocol depends on Server Message Block (SMB) Protocol version 3 for its transport, as specified in [\[MS-SMB2\]](#).



**Figure 1: Relationship to other protocols**

## 1.5 Prerequisites/Preconditions

The RSVD Protocol has the following preconditions:

- An SMB client has established a connection to an SMB server. This has to be done before a client can issue Remote Shared Virtual Disk Protocol commands.
- The SMB client and server support the SVHDX\_OPEN\_DEVICE\_CONTEXT create context, as specified in section [2.2.4.12](#).

## 1.6 Applicability Statement

The Remote Shared Virtual Disk Protocol is applicable for all scenarios that access a shared virtual disk file between client and server.

## 1.7 Versioning and Capability Negotiation

The Remote Shared Virtual Disk Protocol has a single version. [<1>](#)

Version	Value
RSVD Protocol version 1	0x00000001

## 1.8 Vendor-Extensible Fields

None.

## 1.9 Standards Assignments

This protocol shares the standards assignments of Server Message Block Protocol versions 2 and 3, as specified in [\[MS-SMB2\]](#) section 1.9.



## 2 Messages

### 2.1 Transport

The following sections specify how RSVD Protocol messages are encapsulated on the wire and common protocol data types.

All RSVD Protocol messages begin with a fixed-length RSVD header that is described in section [2.2.4.11](#). The RSVD tunnel header contains an OperationCode field indicating the operation code that is requested by the RSVD client or responded to by the RSVD server.

Unless otherwise specified, multiple-byte fields (16-bit, 32-bit, and 64-bit fields) in the RSVD Protocol message MUST be transmitted in little-endian order (least-significant byte first).

Unless otherwise indicated, numeric fields are integers of the specified byte length.

Unless otherwise specified, all textual strings MUST be in **Unicode** version 5.0 format, as specified in [\[UNICODE\]](#), using the 16-bit Unicode Transformation Format (UTF-16) form of the encoding. Textual strings with separate fields identifying the length of the string MUST NOT be null-terminated unless otherwise specified.

Unless otherwise noted, fields marked as "Reserved" MUST be set to 0 when being sent and MUST be ignored when received. These fields are reserved for future protocol expansion and MUST NOT be used for implementation-specific functionality.

The RSVD Protocol uses the SMB 3.02 dialect in SMB Protocol version 3 as its transport. For more information, see [\[MS-SMB2\]](#) section 2.1.

### 2.2 Message Syntax

#### 2.2.1 Constants

Constant name	Meaning
RSVD_CDB_GENERIC_LENGTH 0x10	Generic length of command descriptor block
RSVD_SCSI_SENSE_BUFFER_SIZE 0x14	SENSE buffer size
RSVD_MAXIMUM_NAME_LENGTH 0x7E	Maximum length of the InitiatorHostName field in section <a href="#">2.2.4.12</a>
FSCTL_SVHDX_SYNC_TUNNEL_REQUEST 0x00090304	Control code for SYNC TUNNEL REQUEST

#### 2.2.2 Operation Codes

The following is a list of all control codes used in shared virtual disk operations.

Name	Meaning
RSVD_TUNNEL_GET_FILE_INFO_OPERATION 0x001	Query shared virtual disk file and server information

Name	Meaning
RSVD_TUNNEL_SCSI_OPERATION 0x002	Perform SCSI operation
RSVD_TUNNEL_CHECK_CONNECTION_STATUS_OPERATION 0x003	Query shared virtual disk connection status
RSVD_TUNNEL_SRB_STATUS_OPERATION 0x004	Query sense error code of previously failed request
RSVD_TUNNEL_GET_DISK_INFO_OPERATION 0x005	Query disk information
RSVD_TUNNEL_VALIDATE_DISK_OPERATION 0x006	Perform shared virtual disk validation

### 2.2.3 Error Code

The following is a list of possible RSVD error codes that can be returned by the server.

Name	Meaning
STATUS_SVHDX_ERROR_STORED 0xC05C0000	Sense error data was stored on server
STATUS_SVHDX_ERROR_NOT_AVAILABLE 0xC05CFF00	Sense error data is not available
STATUS_SVHDX_UNIT_ATTENTION_AVAILABLE 0xC05CFF01	Unit Attention data is available for the initiator to query
STATUS_SVHDX_UNIT_ATTENTION_CAPACITY_DATA_CHANGED 0xC05CFF02	The data capacity of the device has changed, resulting in a Unit Attention condition
STATUS_SVHDX_UNIT_ATTENTION_RESERVATIONS_PREEMPTED 0xC05CFF03	A previous operation resulted in this initiator's reservations being preempted, resulting in a Unit Attention condition
STATUS_SVHDX_UNIT_ATTENTION_RESERVATIONS_RELEASED 0xC05CFF04	A previous operation resulted in this initiator's reservations being released, resulting in a Unit Attention condition
STATUS_SVHDX_UNIT_ATTENTION_REGISTRATIONS_PREEMPTED 0xC05CFF05	A previous operation resulted in this initiator's registrations being preempted, resulting in a Unit Attention condition
STATUS_SVHDX_UNIT_ATTENTION_OPERATING_DEFINITION_CHANGED 0xC05CFF06	Represents the data storage format of the device has changed, resulting in a Unit Attention condition

Name	Meaning
STATUS_SVHDX_RESERVATION_CONFLICT 0xC05CFF07	The current initiator is not allowed to perform the SCSI command because of a reservation conflict
STATUS_SVHDX_WRONG_FILE_TYPE 0xC05CFF08	File on which open is performed is of wrong type
STATUS_SVHDX_VERSION_MISMATCH 0xC05CFF09	Protocol version in request is not equal to 1

## 2.2.4 Structures

### 2.2.4.1 SVHDX\_TUNNEL\_CHECK\_CONNECTION\_REQUEST Structure

The SVHDX\_TUNNEL\_CHECK\_CONNECTION\_REQUEST packet is sent by the client to check the connection status to the shared virtual disk. The request MUST contain only SVHDX\_TUNNEL\_OPERATION\_HEADER, and MUST NOT contain any payload.

### 2.2.4.2 SVHDX\_TUNNEL\_CHECK\_CONNECTION\_RESPONSE Structure

The SVHDX\_TUNNEL\_CHECK\_CONNECTION\_RESPONSE packet is sent by the server in response to the operation RSVD\_TUNNEL\_CHECK\_CONNECTION\_STATUS\_OPERATION. The response MUST contain only SVHDX\_TUNNEL\_OPERATION\_HEADER, and MUST NOT contain any payload.

### 2.2.4.3 SVHDX\_TUNNEL\_SRB\_STATUS\_REQUEST Structure

The SVHDX\_TUNNEL\_SRB\_STATUS\_REQUEST packet is sent by the client to get the sense error code of a previously failed request.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31					
StatusKey										Reserved																										
...																																				
...																																				
...																																				
...																																				
...																																				
...																																				

**StatusKey (1 byte):** The client MUST set this field to the least significant byte of the error code of a previously failed request.

**Reserved (27 bytes):** This field MUST be set to zero, and MUST be ignored on receipt.

#### 2.2.4.4 SVHDX\_TUNNEL\_SRB\_STATUS\_RESPONSE Structure

The SVHDX\_TUNNEL\_SRB\_STATUS\_RESPONSE packet is sent by the server in a response to the RSVD\_TUNNEL\_SRB\_STATUS\_OPERATION.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
StatusKey									A	SrbStatus							ScsiStatus				SenseInfoExLength										
SenseDataEx (variable)																															
...																															
...																															
...																															
...																															

**StatusKey (1 byte):** The server MUST set this field to the status key value received in the request.

**A - SenseInfoAutoGenerated (1 bit):** A 1-bit field used to indicate that **sense data** was automatically generated by the virtual **SCSI** disk.

**SrbStatus (7 bits):** A 7-bit field used to communicate error messages from the server to the client. This field MUST contain one of the values in section [2.2.5](#).

**ScsiStatus (1 byte):** An 8-bit field used to communicate the SCSI status that was returned by the shared virtual disk.

**SenseInfoExLength (1 byte):** The length, in bytes, of the request sense data buffer.

**SenseDataEx (variable):** A buffer of maximum size 20 bytes that contains the sense data.

#### 2.2.4.5 SVHDX\_TUNNEL\_DISK\_INFO\_REQUEST Structure

The SVHDX\_TUNNEL\_DISK\_INFO\_REQUEST packet is sent by the client to get shared virtual disk information.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Reserved1																															
...																															
BlockSize																															

LinkageID		
...		
...		
...		
IsMounted	Is4kAligned	Reserved2
FileSize		
...		
VirtualDiskId		
...		
...		
...		

**Reserved1 (8 byte):** The client MUST set this field to zero, and the server MUST ignore it on receipt.

**BlockSize (4 bytes):** The client MUST set this field to zero, and the server MUST ignore it on receipt.

**LinkageID (16 bytes):** The client MUST set this field to zero, and the server MUST ignore it on receipt.

**IsMounted (1 byte):** The client MUST set this field to zero, and the server MUST ignore it on receipt.

**Is4kAligned (1 byte):** The client MUST set this field to zero, and the server MUST ignore it on receipt.

**Reserved2 (2 bytes):** The client MUST set this field to zero, and the server MUST ignore it on receipt.

**FileSize (8 bytes):** The client MUST set this field to zero, and the server MUST ignore it on receipt.

**VirtualDiskId (16 bytes):** This field MUST NOT be used and MUST be reserved. The client MUST set this field to zero, and the server MUST ignore it on receipt.

#### 2.2.4.6 SVHDX\_TUNNEL\_DISK\_INFO\_RESPONSE Structure

The SVHDX\_TUNNEL\_DISK\_INFO\_RESPONSE packet is sent by the server in response to an RSVD\_TUNNEL\_VALIDATE\_DISK\_OPERATION.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
DiskType																															
DiskFormat																															
BlockSize																															
LinkageID																															
...																															
...																															
...																															
IsMounted										Is4kAligned										Reserved											
FileSize																															
...																															
VirtualDiskId																															
...																															
...																															
...																															

**DiskType (4 bytes):** Indicates the type of disk. This field MUST contain exactly one of the following values.

Value	Meaning
VHD_TYPE_FIXED 0x00000002	Indicates that the type of the disk is fixed.
VHD_TYPE_DYNAMIC 0x00000003	Indicates that the type of the disk is dynamic.

**DiskFormat (4 bytes):** Indicates the format of the disk. This field MUST contain exactly one of the following values.

Value	Meaning
VIRTUAL_STORAGE_TYPE_DEVICE_VHDX	Indicates that the type of the disk is shared virtual disk.

Value	Meaning
0x00000003	

**BlockSize (4 bytes):** Specifies the disk block size in bytes.

**LinkageID (16 bytes):** A **GUID** that specifies the linkage identification of the disk.

**IsMounted (1 byte):** A Boolean value. Zero represents FALSE (0x00), indicating that the disk is not ready for read or write operations. One represents TRUE (0x01), indicating that the disk is mounted and ready for read or write operations.

**Is4kAligned (1 byte):** A Boolean value. Zero represents FALSE, indicating disk sectors are not aligned to 4 kilobytes. One represents TRUE, indicating disk sectors are aligned to 4 kilobytes.

**Reserved (2 bytes):** This field **MUST NOT** be used and **MUST** be reserved. The client **SHOULD** set this field to zero, and the server **MUST** ignore it on receipt.

**FileSize (8 bytes):** The size of shared virtual disk, in bytes.

**VirtualDiskId (16 bytes):** A GUID that specifies the identification of the disk.

### 2.2.4.7 SVHDX\_TUNNEL\_SCSI\_REQUEST Structure

The SVHDX\_TUNNEL\_SCSI\_REQUEST packet is sent by the client to process the SCSI request.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Length																Reserved1															
CDBLength								SenseInfoExLength								DataIn								Reserved2							
SrbFlags																															
DataTransferLength																															
CDBBuffer																															
...																															
...																															
...																															
Reserved3																															
DataBuffer (variable)																															
...																															

**Length (2 bytes):** Specifies the size, in bytes, of the SVHDX\_TUNNEL\_SCSI\_REQUEST structure excluding the **DataBuffer** field. The client **MUST** set this field to 36.

**Reserved1 (2 byte):** The client **MUST** set this field to zero.

**CDBLength (1 byte):** The length, in bytes, of the **SCSI command descriptor block**. This value **MUST** be less than or equal to RSVD\_CDB\_GENERIC\_LENGTH.

**SenseInfoExLength (1 byte):** The length, in bytes, of the request sense data buffer. This value **MUST** be less than or equal to RSVD\_SCSI\_SENSE\_BUFFER\_SIZE.

**DataIn (1 byte):** A Boolean, indicating the SCSI command descriptor block transfer type. The value TRUE (0x01) indicates that the operation is to store the data onto the disk. The value FALSE (0x00) indicates that the operation is to retrieve the data from the disk.

**Reserved2 (1 byte):** This field **MUST** be set to 0x00.

**SrbFlags (4 bytes):** An optional, application-provided flag to indicate the options of the SCSI request.

**DataTransferLength (4 bytes):** The length, in bytes, of the additional data placed in the **DataBuffer** field.

**CDBBuffer (16 bytes):** A buffer that contains the SCSI command descriptor block.

**Reserved3 (4 bytes):** This field **MUST** be set to 0x00000000.

**DataBuffer (variable):** A variable-length buffer that contains the additional buffer, as described by the **DataTransferLength** field.

#### 2.2.4.8 SVHDX\_TUNNEL\_SCSI\_RESPONSE Structure

The SVHDX\_TUNNEL\_SCSI\_RESPONSE packet is sent by the server in response to the operation RSVD\_TUNNEL\_SCSI\_OPERATION.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Length																A	SrbStatus						ScsiStatus								
CDBLength						SenseInfoExLength						DataIn						Reserved													
SrbFlags																															
DataTransferLength																															
SenseDataEx (variable)																															
...																															
...																															
...																															



DataBuffer (variable)
...

**Length (2 bytes):** Specifies the size, in bytes, of the SVHDX\_TUNNEL\_SCSI\_RESPONSE structure.

**A - SenseInfoAutoGenerated (1 bit):** A 1-bit field used to indicate that sense data was automatically generated by the virtual SCSI disk.

**SrbStatus (1 byte):** A 7-bit field used to communicate error messages from the server to the client. This field MUST contain one of the values in section [2.2.5](#).

**ScsiStatus (1 byte):** An 8-bit field used to communicate the SCSI status that was returned by the target device.

**CDBLength (1 byte):** The length, in bytes, of the SCSI command descriptor block.

**SenseInfoExLength (1 byte):** The length, in bytes, of the request sense data buffer.

**DataIn (1 byte):** A Boolean used to indicate the command descriptor block data-transfer type. TRUE (0x01), indicates that the operation is to store the data onto the disk. The value FALSE (0x00) indicates that the operation is to retrieve the data from the disk.

**Reserved (1 byte):** This field MUST be set to zero, and MUST be ignored on receipt.

**SrbFlags (4 bytes):** Special flags to indicate options of the SCSI response.

**DataTransferLength (4 bytes):** The length, in bytes, of the additional data placed in the **DataBuffer** field.

**SenseDataEx (variable):** A buffer of maximum size 20 bytes that contains the command descriptor buffer.

**DataBuffer (variable):** A variable-length buffer that contains the additional buffer, as described by the **DataTransferLength** field.

#### 2.2.4.9 SVHDX\_TUNNEL\_VALIDATE\_DISK\_REQUEST Structure

The **SVHDX\_TUNNEL\_VALIDATE\_DISK\_REQUEST** packet is sent by the client to validate the shared virtual disk. This request contains 56 bytes. The client MUST set all 56 bytes to zero and the server MUST ignore them on receipt.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Reserved																															
...																															
...																															
...																															

...
...
...
...
...
...
...
...
...
...

#### 2.2.4.10 SVHDX\_TUNNEL\_VALIDATE\_DISK\_RESPONSE Structure

The **SVHDX\_TUNNEL\_VALIDATE\_DISK\_RESPONSE** packet is sent by the server in a response to the operation **RSVD\_TUNNEL\_VALIDATE\_DISK\_OPERATION**.

0	1	2	3	4	5	6	7
IsValidDisk							

**IsValidDisk** (1 byte): A Boolean value that, if set, indicates that the disk is valid.

#### 2.2.4.11 SVHDX\_TUNNEL\_OPERATION\_HEADER Structure

The RSVD tunnel header is the header of RSVD Protocol operations specified in section [2.2.2](#).

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
ProtocolId										ProtocolVersion										OperationCode											
Status																															
RequestId																															
...																															

**ProtocolId (1 byte):** The protocol identifier. The client and server MUST set this value to 2.

**ProtocolVersion(12 bits):** The protocol version. The client and server MUST set this value to 1.

**OperationCode (12 bits):** The command code of this packet. This field MUST contain one of values specified in section [2.2.2](#).

**Status (4 bytes):** The client SHOULD set this field to zero, and the server MUST ignore it on receipt.

**RequestId (8 bytes):** A value that uniquely identifies an operation request and response for all requests sent by this client.

### 2.2.4.12 SVHDX\_OPEN\_DEVICE\_CONTEXT Structure

The SVHDX\_OPEN\_DEVICE\_CONTEXT packet is sent by the client to open the shared virtual disk.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Version																															
HasInitiatorId										Reserved																					
InitiatorId																															
...																															
...																															
...																															
Flags																															
OriginatorFlags																															
OpenRequestId																															
...																															
InitiatorHostNameLength																InitiatorHostName (126 bytes)															
...																															

**Version (4 bytes):** The version of the create context. It MUST be set to the highest supported version of the protocol, as specified in section [1.7](#).

**HasInitiatorId (1 bytes):** A Boolean value, where zero represents FALSE and nonzero represents TRUE.

**Reserved (3 bytes):** This field MUST be set to zero when sent and MUST be ignored on receipt.

**InitiatorId (16 bytes):** A GUID that optionally identifies the initiator of the open request.

**Flags (4 bytes):** Reserved. The client SHOULD set this field to 0x00000000, and the server MUST ignore it on receipt.

**OriginatorFlags (4 bytes):** This field is used to indicate which component has originated or issued the operation. This field MUST be set to one of the following values:

Name	Meaning
SVHDX_ORIGINATOR_PVHDPARSER 0x00000001	Shared virtual disk file to be opened as a virtual SCSI disk device
SVHDX_ORIGINATOR_VHDMP 0x00000004	Shared virtual disk file to be opened in underlying object store

**OpenRequestId (8 bytes):** A 64-bit value assigned by the client for an outgoing request. The server MUST ignore it on receipt.

**InitiatorHostNameLength (2 bytes):** The length, in bytes, of the **InitiatorHostName**. This value MUST be less than or equal to RSVD\_MAXIMUM\_NAME\_LENGTH.

**InitiatorHostName (126 bytes):** A 126-byte buffer containing a null-terminated Unicode UTF-16 string that specifies the computer name on which the initiator resides.

### 2.2.4.13 SVHDX\_TUNNEL\_FILE\_INFO\_REQUEST Structure

The SVHDX\_TUNNEL\_FILE\_INFO\_REQUEST packet is sent by the client to get the shared virtual disk file information. The request MUST contain only SVHDX\_TUNNEL\_OPERATION\_HEADER, and MUST NOT contain any payload.

### 2.2.4.14 SVHDX\_TUNNEL\_FILE\_INFO\_RESPONSE Structure

The SVHDX\_TUNNEL\_INITIAL\_INFO\_RESPONSE packet is sent by the server in response to an RSVD\_TUNNEL\_GET\_FILE\_INFO\_OPERATION packet.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
ServerVersion																															
SectorSize																															
PhysicalSectorSize																															
Reserved																															
VirtualSize																															
...																															

**ServerVersion (4 bytes):** The current version of the protocol running on the server.

**SectorSize (4 bytes):** A 32-bit unsigned integer that indicates the sector size, in bytes, of the shared virtual disk.

**PhysicalSectorSize (4 bytes):** A 32-bit unsigned integer that indicates the physical sector size, in bytes, of the shared virtual disk.

**Reserved (4 bytes):** This field MUST be set to zero, and the client MUST ignore it on receipt.

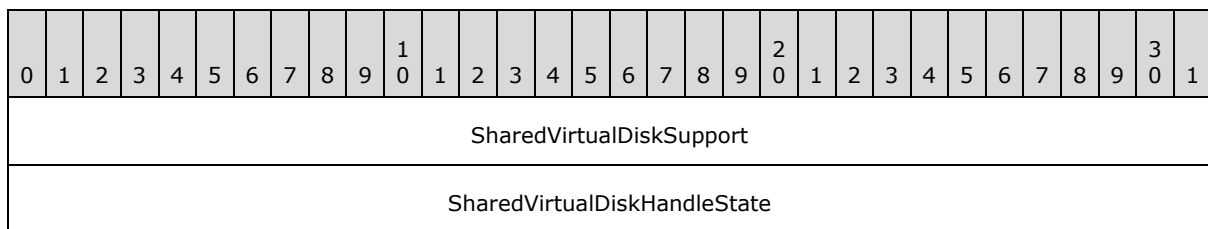
**VirtualSize (8 bytes):** A 64-bit unsigned integer that indicates the virtual size, in bytes, of the shared virtual disk.

#### 2.2.4.15 SVHDX\_SHARED\_VIRTUAL\_DISK\_SUPPORT\_REQUEST Structure

The **SVHDX\_SHARED\_VIRTUAL\_DISK\_SUPPORT\_REQUEST** packet is sent by the client to verify the status of shared virtual disk support on the Open. The request MUST NOT contain any payload.

#### 2.2.4.16 SVHDX\_SHARED\_VIRTUAL\_DISK\_SUPPORT\_RESPONSE Structure

The **SVHDX\_SHARED\_VIRTUAL\_DISK\_SUPPORT\_RESPONSE** packet is sent by the server in a response to an **FSCTL\_QUERY\_SHARED\_VIRTUAL\_DISK\_SUPPORT** request.



**SharedVirtualDiskSupport (4 bytes):** This field is used to indicate whether the server supports shared virtual disks. This field MUST be 0x00000001.

**SharedVirtualDiskHandleState (4 bytes):** This field is used to indicate the state of the shared virtual disk Open. This field MUST contain one of the following values.

Value	Meaning
HandleStateNone 0x00000000	The Open is not opened as a shared virtual disk.
HandleStateFileShared 0x00000001	The shared virtual disk file is opened as a shared virtual disk by another Open.
HandleStateShared 0x00000003	The shared virtual disk file is opened as a shared virtual disk by this Open.

#### 2.2.5 SRB Status Code

The following is a list of possible SRB error messages used to communicate from server to client.

Value	Meaning
0x01	The request was completed successfully.

<b>Value</b>	<b>Meaning</b>
0x02	The request was aborted.
0x06	The Shared Virtual Disk does not support the given request.
0x08	The Shared Virtual Disk device is no longer available.
0x0A	The SCSI device selection timed out.
0x12	A data overrun or underrun error occurred.
0x04	The request completed with any other error.

## 3 Protocol Details

### 3.1 Client Details

#### 3.1.1 Abstract Data Model

##### 3.1.1.1 Global

The client MUST implement the following:

**ClientServiceVersion:** The highest protocol version supported by the client.

**RequestIdentifier:** An unsigned 64-bit value assigned by the client for an outgoing request.

##### 3.1.2 Timers

None.

##### 3.1.3 Initialization

The client MUST initialize **ClientServiceVersion** to the highest protocol version supported by the client. [<2>](#)

**RequestIdentifier:** SHOULD [<3>](#) be initialized to an implementation-specific value.

##### 3.1.4 Higher-Layer Triggered Events

###### 3.1.4.1 Sending Any Outgoing Message

The Client MUST increment **RequestIdentifier** by 1, for every outgoing tunnel operation specified in section [2.2.2](#).

###### 3.1.4.2 Application Requests Opening a Shared Virtual Disk

The application provides the following:

- The name of the server to connect to.
- The name of the share to connect to.
- InitiatorId to uniquely identify the initiator (optional).
- User credentials, an opaque implementation-specific entity that identifies the credentials to be used when authenticating to the remote server.
- The shared virtual disk file name to open.
- The flags for open command (optional).
- The Initiator host name.
- A Boolean that, if set, requires the shared virtual disk file to be opened as virtual SCSI disk.

Upon successful completion, the client MUST return a handle to the application.

The client MUST construct an SVHDX\_OPEN\_DEVICE\_CONTEXT as specified in section [2.2.4.12](#) and MUST be initialized as follows:

- The **Version** field is set to **ClientServiceVersion**.
- The **InitiatorId** field is set to the application-provided InitiatorId, if any. Otherwise, **InitiatorId** is set to zero.
- The **HasInitiatorId** field is set to TRUE if the application provides initiator Id; otherwise set to FALSE.
- The **Flags** field is set to application-provided flags.
- If the application requires the shared virtual disk file to be opened as a virtual SCSI disk device, the client MUST set **OriginatorFlags** to SVHDX\_ORIGINATOR\_PVHDPARSER. Otherwise, it MUST be set to SVHDX\_ORIGINATOR\_VHDMP.
- The **OpenRequestId** is set to **RequestIdentifier**.
- The **InitiatorHostNameLength** is set to **InitiatorHostName** length, in bytes.
- The **InitiatorHostName** is set to the application-provided initiator name. If **InitiatorHostNameLength** is less than 126 bytes, the remaining bytes in the buffer MUST be set to zero.

The client MUST append ":SharedVirtualDisk" at the end of the file name.

The client MUST establish a connection to the server by calling the interface specified in [\[MS-SMB2\]](#) section 3.2.4.2 and providing the following input parameters:

- The application-provided server name.
- The application-provided share name.
- The application-provided user credentials.

If the connection is successfully established, the client MUST open the shared virtual disk file by calling the interface specified in [\[MS-SMB2\]](#) section 3.2.4.3 and provide the following input parameters:

- File name
- The SVHDX\_OPEN\_DEVICE\_CONTEXT Create context.
- **CreateOptions** as specified in [\[MS-SMB2\]](#) section 2.2.13 with the FILE\_NO\_INTERMEDIATE\_BUFFERING bit set.

If there are any errors from the preceding call, return the error to the caller.

### 3.1.4.3 Application Requests Closing a Shared Virtual Disk

The application provides:

- A handle to the **Open** identifying the shared virtual disk file.

The client MUST call the interface specified in [\[MS-SMB2\]](#) section 3.2.4.5 to close the open on the shared virtual disk file, supplying the application-provided handle.



### 3.1.4.4 Application Requests Reading From a Shared Virtual Disk

The application provides:

- A handle to the **Open** identifying the shared virtual disk file.
- The offset, in bytes, from the beginning of the virtual disk from which to read data.
- The number of bytes to read.
- The minimum number of bytes to be read (optional).
- The buffer to receive the data.

The client MUST call the interface specified in [\[MS-SMB2\]](#) section 3.2.4.6, supplying the application-provided parameters

### 3.1.4.5 Application Requests Writing To a Shared Virtual Disk

The application provides:

- A handle to the **Open** identifying a shared virtual disk file.
- The offset, in bytes, from the beginning of the virtual disk where data should be written.
- The number of bytes to write.
- A buffer containing the bytes to be written.

The client MUST call the interface specified in [\[MS-SMB2\]](#) section 3.2.4.7, supplying the application-provided parameters.

### 3.1.4.6 Application Requests Virtual Disk File information

The application provides:

- A handle to the **Open** identifying a shared virtual disk file.
- The maximum output buffer size that it will accept.

The client MUST construct an SVHDX\_TUNNEL\_FILE\_INFO\_REQUEST structure, as specified in section [2.2.4.13](#) as follows:

The SVHDX\_TUNNEL\_OPERATION\_HEADER MUST be initialized as follows:

- The **OperationCode** MUST be set to RSVD\_TUNNEL\_GET\_INITIAL\_INFO\_OPERATION.
- The **Status** field MUST be set to zero.
- The **RequestId** field MUST be set to **RequestIdentifier**.

The client MUST call the interface specified in [\[MS-SMB2\]](#) section 3.2.4.20.11, supplying the following input parameters:

- Application-provided handle to identify the **Open**.
- Control code: FSCTL\_SVHDX\_SYNC\_TUNNEL\_REQUEST.

- SVHDX\_TUNNEL\_OPERATION\_HEADER structure as payload.
- The maximum output buffer size that it will accept.

#### 3.1.4.7 Application Requests Connection Status

The application provides:

- A handle to the **Open** identifying a shared virtual disk file.

The client MUST construct an SVHDX\_TUNNEL\_CHECK\_CONNECTION\_REQUEST structure, as specified in section [2.2.4.1](#) as follows:

The SVHDX\_TUNNEL\_OPERATION\_HEADER MUST be initialized as follows:

- The **OperationCode** MUST be set to RSVD\_TUNNEL\_CHECK\_CONNECTION\_STATUS\_OPERATION.
- The **Status** field MUST be set to zero.
- The **RequestId** field MUST be set to **RequestIdentifier**.

The client MUST call the interface specified in [\[MS-SMB2\]](#) section 3.2.4.20.11, supplying the following input parameters:

- Application-provided handle to identify the **Open**.
- Control code: FSCTL\_SVHDX\_SYNC\_TUNNEL\_REQUEST.
- SVHDX\_TUNNEL\_OPERATION\_HEADER as payload.

#### 3.1.4.8 Application Requests Shared Virtual Disk Information

The application provides:

- A handle to the **Open** identifying a shared virtual disk file.

The client MUST construct an SVHDX\_TUNNEL\_DISK\_INFO\_REQUEST structure, as specified in section [2.2.4.5](#), as follows:

The SVHDX\_TUNNEL\_OPERATION\_HEADER MUST be initialized as follows:

- The **OperationCode** MUST be set to RSVD\_TUNNEL\_GET\_DISK\_INFO\_OPERATION.
- The **Status** field MUST be set to zero.
- The **RequestId** field MUST be set to **RequestIdentifier**.

The SVHDX\_TUNNEL\_DISK\_INFO\_REQUEST Request MUST be initialized as follows:

- The **Reserved** field MUST be set to zero.
- The **BlockSize** field MUST be set to zero.
- The **LinkageId** field MUST be set to zero.
- The **IsMounted** field MUST be set to zero.
- The **Is4kAligned** field MUST be set to zero.

- The **FileSize** field MUST be set to zero.
- The **VirtualDiskId** field MUST be set to zero.

The client MUST call the interface specified in [\[MS-SMB2\]](#) section 3.2.4.20.11, supplying the following input parameters:

- Application-provided handle to identify the **Open**.
- Control code: FSCTL\_SVHDX\_SYNC\_TUNNEL\_REQUEST.
- SVHDX\_TUNNEL\_DISK\_INFO\_REQUEST packet, as payload.

### 3.1.4.9 Application Requests Execution of SCSI Command

The application provides:

- A handle to the **Open** identifying a shared virtual disk file.
- The ID of the initiator.
- The length of the SCSI Command Descriptor Block (CDB) buffer, in bytes.
- The length of the sense information, in bytes.
- A Boolean value indicating the SCSI CDB transfer type, as specified in section [2.2.4.3](#).
- SCSI command flags that indicate options about the SCSI request.
- The SCSI CDB buffer.
- The length of any additional data (optional).
- Any additional data buffer (optional).

The client MUST construct an SVHDX\_TUNNEL\_SCSI\_REQUEST structure, as specified in section [2.2.4.7](#) as follows:

The SVHDX\_TUNNEL\_OPERATION\_HEADER MUST be initialized as follows:

- The **OperationCode** field is set to RSVD\_TUNNEL\_SCSI\_OPERATION.
- The **Status** is set to zero.
- The **RequestId** field MUST be set to RequestIdentifier.

The SVHDX\_TUNNEL\_SCSI\_REQUEST MUST be initialized as follows:

- The **Length** field is set to 36.
- The **Reserved1**, **Reserved2**, and **Reserved3** fields are set to zero.
- The **CDBLength** is set to the application-provided CDB length value.
- The **SenseInfoExLength** set to the application-provided sense info length value.
- The **DataIn** value is set to the application-provided Boolean value.
- The **SrbFlags** field MUST be set to the application-provided special flag values.

- The **DataTransferLength** field MUST be set to the application-provided value for the **Length** of the additional value. If the application doesn't provide a data buffer length, the client MUST set this field to zero.
- The **CDBBuffer** field MUST be set to the application-provided CDB buffer.
- The **DataBuffer** field MUST be set to the application-provided additional data buffer. If the application doesn't provide the buffer, the client MUST set this field to empty.

The client MUST call the interface specified in [\[MS-SMB2\]](#) section 3.2.4.20.11, supplying the following input parameters:

- Application-provided handle to identify the **Open**.
- Control code: FSCTL\_SVHDX\_SYNC\_TUNNEL\_REQUEST.
- SVHDX\_TUNNEL\_SCSI\_REQUEST structure as payload.

### 3.1.4.10 Application Requests to Validate a Shared Virtual Disk

The application provides:

- A handle to the **Open** identifying a shared virtual disk file.

The client MUST construct an SVHDX\_TUNNEL\_VALIDATE\_DISK\_REQUEST structure, as specified in section [2.2.4.9](#), as follows:

The SVHDX\_TUNNEL\_OPERATION\_HEADER MUST be initialized as follows:

- The **OperationCode** MUST be set to RSVD\_TUNNEL\_VALIDATE\_DISK\_OPERATION.
- The **Status** field MUST be set to zero.
- The **RequestId** field MUST be set to **RequestIdentifier**.

The client MUST call the interface specified in [\[MS-SMB2\]](#) section 3.2.4.20.11, supplying the following input parameters:

- Application-provided handle to identify the **Open**.
- Control code: FSCTL\_SVHDX\_SYNC\_TUNNEL\_REQUEST.
- SVHDX\_TUNNEL\_OPERATION\_HEADER structure as payload.

### 3.1.4.11 Application Requests Querying Shared Virtual Disk Support

The application provides:

- A handle to the **Open** identifying a shared virtual disk file.

The client MUST call the interface specified in [\[MS-SMB2\]](#) section 3.2.4.20.11 supplying the following input parameters:

- Application-provided handle to identify the **Open**.
- Control code: FSCTL\_QUERY\_SHARED\_VIRTUAL\_DISK\_SUPPORT

## 3.1.5 Message Processing Events and Sequencing Rules

### 3.1.5.1 Receiving an Open Response

If the response returned by the SMB server indicates an error, the client MUST return the received error code to the calling application.

Otherwise, the client MUST return success and the **Open.ContextHandle** to the calling application.

### 3.1.5.2 Receiving a Close Response

The client MUST return the received response to the calling application.

### 3.1.5.3 Receiving a Read Response

If the response returned by the SMB server indicates success, the client MUST return success and the data buffer that contains the data read for the response.

If the response indicates an error, and the received error code is specified in the section [2.2.3](#), the server MUST return the error code.

If the received error code is not specified in the section [2.2.3](#), the client MUST construct an SVHDX\_TUNNEL\_SRB\_STATUS\_REQUEST structure as follows:

The SVHDX\_TUNNEL\_OPERATION\_HEADER MUST be initialized as follows:

- The **OperationCode** MUST be set to RSVD\_TUNNEL\_SRB\_STATUS\_OPERATION.
- The **Status** field MUST be set to zero.
- The **RequestId** field MUST be set to **RequestIdentifier**.

The SVHDX\_TUNNEL\_SRB\_STATUS\_REQUEST structure MUST be initialized as follows:

- The **StatusKey** field MUST be set to the least significant byte of the received error code.
- The **Reserved** field MUST be set to zero.

The client MUST call the interface specified in [\[MS-SMB2\]](#) section 3.2.4.20.11, supplying the following input parameters:

- Application-provided handle used for Read request.
- Control code: FSCTL\_SVHDX\_SYNC\_TUNNEL\_REQUEST.
- SVHDX\_TUNNEL\_SRB\_STATUS\_REQUEST structure as payload.

The client MUST return the received sense error response to the calling application.

### 3.1.5.4 Receiving a Write Response

If the response returned by the SMB server indicates success, the client MUST return success to the calling application.

If the response indicates an error, and the received error code is specified in the section [2.2.3](#), the server MUST return the error code.

If the received error code is not specified in the section [2.2.3](#), the client MUST construct an SVHDX\_TUNNEL\_SRB\_STATUS\_REQUEST structure as follows:

The SVHDX\_TUNNEL\_OPERATION\_HEADER MUST be initialized as follows:

- The **OperationCode** MUST be set to RSVD\_TUNNEL\_SRB\_STATUS\_OPERATION.
- The **Status** field MUST be set to zero.
- The **RequestId** field MUST be set to **RequestIdentifier**.

The SVHDX\_TUNNEL\_SRB\_STATUS\_REQUEST structure MUST be initialized as follows:

- The **StatusKey** field MUST be set to the least significant byte of the received error code.
- The **Reserved** field MUST be set to zero.

The client MUST call the interface specified in [\[MS-SMB2\]](#) section 3.2.4.20.11, supplying the following input parameters:

- Application-provided handle used for Write request.
- Control code: FSCTL\_SVHDX\_SYNC\_TUNNEL\_REQUEST.
- SVHDX\_TUNNEL\_SRB\_STATUS\_REQUEST structure as payload.

The client MUST return the received sense error response to the calling application.

### **3.1.5.5 Receiving a Virtual Disk File Information Response**

If the response returned by the SMB server indicates an error, the client MUST return the received error code to the calling application.

If the response indicates success, the client MUST return success and the disk file information to the calling application.

### **3.1.5.6 Receiving a Connection Status Response**

The client MUST return the received status code to the calling application.

### **3.1.5.7 Receiving a Shared Virtual Disk Information Request**

If the response returned by the SMB server indicates an error, the client MUST return the received status code to the calling application.

If the response indicates success, the client MUST return success and the virtual disk information to the calling application.

### **3.1.5.8 Receiving a SCSI Command Response**

If the response returned by the SMB server indicates an error, the client MUST return the received error code to the calling application.

If the response indicates success, the client MUST return success and response data to the calling application.

### 3.1.5.9 Receiving a Validate Disk Response

The client MUST return the received status to the calling application.

### 3.1.5.10 Receiving a Shared Virtual Disk Support Response

The client MUST return the received status and response data to the calling application.

### 3.1.6 Timer Events

None.

### 3.1.7 Other Local Events

None.

## 3.2 Server Details

### 3.2.1 Abstract Data Model

#### 3.2.1.1 Global

The server MUST implement the following:

**OpenTable:** A table of opened shared virtual disk files, as specified in section [3.2.1.2](#), indexed by InitiatorId.

**ServerServiceVersion:** The highest protocol version supported by the server.

#### 3.2.1.2 Per Open

**InitiatorId:** A GUID that identifies the initiator of the open request.

**Open.LocalOpen:** An **Open** of a shared virtual disk file in the local resource that is used to perform the local operations, such as reading or writing, to the underlying object.

**Open.FileName:** A variable-length string that contains the Unicode file name supplied by the client for opening the shared virtual disk.

**Open.SenseErrorSequence:** An unsigned 8-bit identifier indicating the sense error sequence which counts from 0 through 255.

**Open.SenseErrorDataList:** A list of sense errors indexed by the **Open.SenseErrorSequence**, as defined in section [3.2.1.3](#).

**Open.IsVirtualSCSIDisk:** A Boolean that, if set indicates that the file is a virtual SCSI disk.

**Open.CreateOptions:** The create options, in the format specified in [\[MS-SMB2\]](#) section 2.2.13.

#### 3.2.1.3 Per SenseError in SenseErrorDataList

**SenseError.StatusKey:** A 32-bit value assigned by the server for this sense error.

**SenseError.SrbStatus:** A 7-bit value indicating the status of the SCSI request block as specified in section [2.2.5](#).

**SenseError.ScsiStatus:** An 8-bit value indicating the SCSI status that was returned by the shared virtual disk.

**SenseError.SenseData:** A pointer to the sense data.

### 3.2.2 Timers

None.

### 3.2.3 Initialization

When the server is started:

- It MUST notify the SMB server that it is ready to process client requests, as specified in [MS-SMB2] section 3.3.4.25.
- The server MUST initialize **ServerServiceVersion** to the highest protocol version supported by the server. <4>

### 3.2.4 Higher-Layer Triggered Events

#### 3.2.4.1 Handling a Tunnel Operation Request

The calling application provides:

- A handle identifying the **Open**.
- Tunnel Operation request.
- The maximum size of the response that will be accepted by the client, indicated by **MaxOutputResponse**.

The server MUST process this request as specified in section [3.2.5.5](#).

#### 3.2.4.2 Handling a Shared Virtual Disk Support Request

The calling application provides:

- A handle identifying the **Open**.
- File name to identify the shared virtual disk file.
- The maximum size of the response data buffer that will be accepted by the client, indicated by **MaxOutputResponse**.

The server MUST process this request as specified in section [3.2.5.6](#).

### 3.2.5 Message Processing Events and Sequencing Rules

#### 3.2.5.1 Receiving an Open Request

When the server receives a request to open a shared virtual disk file, by providing the file name and SVHDX\_OPEN\_DEVICE\_CONTEXT, the server MUST verify the following:

If the file name does not contain ":SharedVirtualDisk" at the end, the server MUST fail the request with error STATUS\_INVALID\_PARAMETER.



If **HasInitiatorId** is neither FALSE (0x00) nor TRUE (0x01), the server MUST fail the request with STATUS\_INVALID\_PARAMETER.

If the **OriginatorFlags** in the request is set to SVHDX\_ORIGINATOR\_VHDMP, the server MUST pass the request to the underlying object store to open the file with read and write access permissions. If the underlying object store fails with STATUS\_SHARING\_VIOLATION, the server MUST fail the request with STATUS\_VHD\_SHARED.

If the **OriginatorFlags** in the request is not set to SVHDX\_ORIGINATOR\_VHDMP, the server SHOULD<5> pass the request to the virtual SCSI disk.

If the underlying object store or virtual SCSI disk returns a failure indicating that the attempted open operation failed, the server MUST return the received error code.

If the underlying object store or virtual SCSI disk returns success, the server MUST initialize the **Open** as follows, and return STATUS\_SUCCESS to the client.

- **Open.LocalOpen** is set to the **Open** of the object in the local resource received as part of the local creates operation.
- If **HasInitiatorId** is 0x00, **Open.InitiatorId** is set to zero. Otherwise it is set to **InitiatorId**.
- **Open.FileName** MUST be set to the application-provided file name.
- **Open.SenseErrorDataList** MUST be set to empty.
- **Open.SenseErrorSequence** MUST be set to 0x00.
- **Open.IsVirtualSCSIDisk** SHOULD<6> be set to TRUE if **OriginatorFlags** is not set to SVHDX\_ORIGINATOR\_VHDMP. Otherwise, it MUST be set to FALSE.
- **Open.CreateOptions** MUST be set to the **CreateOptions** received as part of the local create operation.

### 3.2.5.2 Receiving a Close Request

When the server receives a request to close a shared virtual disk file, the server MUST verify the following:

The server MUST locate the **Open** in the **OpenTable** where **Open.LocalOpen** matches the **Open** provided by the SMB server, as specified in [\[MS-SMB2\]](#) section 3.3.4.17.

The server MUST close the underlying object store open.

If under the underlying object store return success, the server MUST remove the **Open** from the **OpenTable**, free the **Open** object, and MUST return STATUS\_SUCCESS to the client.

Otherwise, the server MUST return the received error code to the client.

### 3.2.5.3 Receiving a Read Request

When the server receives a read request, the server MUST locate the **Open** in the **OpenTable**, where **Open.LocalOpen** matches the **Open** provided by the SMB2 server, as specified in [\[MS-SMB2\]](#) section 3.2.5.12.

If no **Open** is found, the server MUST fail the request with the STATUS\_INVALID\_PARAMETER code.

If the **Open** is found and **Open.InitiatorId** is zero, the server MUST fail the request with the STATUS\_INVALID\_HANDLE code.

If the **Open** is found and the FILE\_NO\_INTERMEDIATE\_BUFFERING bit is not set in **Open.CreateOptions**, the server MUST fail the request with the STATUS\_NOT\_SUPPORTED code.

If the **Open** is found and **Open.IsVirtualSCSIDisk** is FALSE, the server MUST issue a read to the underlying object store; otherwise the server MUST pass the request to the virtual SCSI disk in an implementation-specific manner.

If the underlying object store or virtual SCSI disk indicates the read is successful, the server MUST return the read data buffer and success to the client.

If the underlying object store or virtual SCSI disk indicates that the read failed, and the received error code is specified in the section [2.2.3](#), the server MUST return the error code.

If the underlying object store is the virtual SCSI disk and the error code is not specified in section [2.2.3](#), the server MUST store the received sense data as follows:

- If **Open.SenseErrorSequence** is 0xFF, the server MUST reset **Open.SenseErrorSequence** to 0x00. Otherwise, the server MUST increment the **Open.SenseErrorSequence** by 0x01.
- The server MUST update **SenseError** in **Open.SenseErrorDataList** at index **Open.SenseErrorSequence** as follows:
  - **SenseError.StatusKey** MUST be set to **Open.SenseErrorSequence**.
  - **SenseError.SrbStatus** MUST be set to the value provided by the virtual SCSI disk.
  - **SenseError.ScsiStatus** MUST be set to the value provided by the virtual SCSI disk.
  - **SenseError.SenseData** MUST be set to the data provided by the virtual SCSI disk.

The server MUST return the error (STATUS\_SVHDX\_ERROR\_STORED | **SenseError.StatusKey**) to the client.

### 3.2.5.4 Receiving a Write Request

When the server receives a write request, the server MUST locate the **Open** in the **OpenTable** where **Open.LocalOpen** matches the **Open** provided by the SMB2 server, as specified in [\[MS-SMB2\]](#) section 3.3.5.13.

If no **Open** is found, the server MUST fail the request with the STATUS\_INVALID\_PARAMETER error code.

If the **Open** is found and **Open.InitiatorId** is zero, the server MUST fail the request with the STATUS\_INVALID\_HANDLE code.

If the **Open** is found and the FILE\_NO\_INTERMEDIATE\_BUFFERING bit is not set in **Open.CreateOptions**, the server MUST fail the request with the STATUS\_NOT\_SUPPORTED code.

If the **Open** is found and **Open.IsVirtualSCSIDisk** is FALSE, the server MUST issue a write to the underlying object store; otherwise the server MUST pass the request to the virtual SCSI disk in an implementation-specific manner.

If the underlying object store or virtual SCSI disk indicates that the write was successful, the server MUST return success to the client.

If the underlying object store or virtual SCSI disk indicates a **persistent reservation** conflict, as described [SPC-3] section 5.6, denied the write, the server MUST fail the request with error STATUS\_SVHDX\_RESERVATION\_CONFLICT.

Otherwise, if the received error code is specified in the section 2.2.3, the server MUST return the error code.

If the underlying object store is the virtual SCSI disk and the error code is not specified in section 2.2.3, the server MUST store the received sense data as follows:

- If **Open.SenseErrorSequence** is 0xFF, the server MUST reset **Open.SenseErrorSequence** to 0x00. Otherwise, the server MUST increment the **Open.SenseErrorSequence** by 0x01.
- The server MUST update **SenseError** in **Open.SenseErrorDataList** at index **Open.SenseErrorSequence** as follows:
  - **SenseError.StatusKey** MUST be set to **Open.SenseErrorSequence**.
  - **SenseError.SrbStatus** MUST be set to the value provided by the virtual SCSI disk.
  - **SenseError.ScsiStatus** MUST be set to the value provided by the virtual SCSI disk.
  - **SenseError.SenseData** MUST be set to the data provided by the virtual SCSI disk.

The server MUST return the error (STATUS\_SVHDX\_ERROR\_STORED | **SenseError.StatusKey**) to the client.

### 3.2.5.5 Receiving a Tunnel Operation Request

When the server receives a tunnel operation request, the server MUST locate the **Open** in the **OpenTable** where **Open.LocalOpen** matches the **Open** provided by the SMB2 server, as specified in [MS-SMB2] section 3.3.5.13.

If no **Open** is found, the server MUST fail the request and return STATUS\_INVALID\_PARAMETER.

If an **Open** is found and if ProtocolId is equal to 0, the server MUST fail the request with STATUS\_INVALID\_DEVICE\_REQUEST.

If any of the following conditions are TRUE, server MUST construct the SVHDX\_TUNNEL\_OPERATION\_HEADER with the ProtocolId, ProtocolVersion, OperationCode, and RequestId fields set to the value received in the request, and with the Status field set as follows.

- If the **ProtocolId** value is not equal to 0x01 or 0x02, the **Status** field MUST be set to STATUS\_NOT\_IMPLEMENTED.
- If the **ProtocolId** value is equal to 0x01, the **Status** field MUST be set to STATUS\_INVALID\_DEVICE\_REQUEST.
- If the **ProtocolVersion** value is not equal to 1, the **Status** field MUST be set to STATUS\_SVHDX\_VERSION\_MISMATCH.
- If the **OperationCode** value does not exist in the tunnel operation list as specified in section 2.2.2, the **Status** field MUST be set to STATUS\_INVALID\_PARAMETER.
- If the **OperationCode** is equal to RSVD\_TUNNEL\_SCSI\_OPERATION and **Open.InitiatorId** is zero, the **Status** field MUST be set to STATUS\_INVALID\_HANDLE.

The server MUST return SVHDX\_TUNNEL\_OPERATION\_HEADER to the client.

If the size of the tunnel operation request including header is less than 16, the server MUST fail the request with STATUS\_BUFFER\_TOO\_SMALL.

Processing for a specific **OperationCode** is as specified in subsequent sections.

### 3.2.5.5.1 Receiving a Virtual Disk File Information Request

When the server receives a request **OperationCode** equal to RSVD\_TUNNEL\_GET\_INITIAL\_INFO\_OPERATION, the request handling proceeds as follows:

If **MaxOutputResponse** is less than 40 (size of SVHDX\_TUNNEL\_OPERATION\_HEADER + size of SVHDX\_TUNNEL\_FILE\_INFO\_RESPONSE), the server MUST fail the request with STATUS\_BUFFER\_TOO\_SMALL.

The server MUST issue a file information request to the virtual SCSI disk in an implementations-specific manner, and update the received response to the client.

If the virtual SCSI disk indicates an error, the server MUST fail the request with the error received by placing it into the **Status** field of the SVHDX\_TUNNEL\_OPERATION\_HEADER, and return the header to the client.

Otherwise, the server MUST construct an SVHDX\_TUNNEL\_FILE\_INFO\_RESPONSE structure as specified in section [2.2.4.14](#) with the following values:

The SVHDX\_TUNNEL\_OPERATION\_HEADER MUST be initialized as follows:

- **OperationCode** MUST be set to the OperationCode value of the request.
- **Status** MUST be set to STATUS\_SUCCESS
- The **RequestId** field MUST be set to the value received in the request.

The SVHDX\_TUNNEL\_FILE\_INFO\_RESPONSE structure MUST be initialized as follows:

- **ServerVersion** MUST be set to the **ServerServiceVersion**.
- **SectorSize** MUST set to the sector size of the shared virtual disk received from the virtual SCSI disk.
- **PhysicalSectorSize** MUST set to the physical sector size of the shared virtual disk received from the virtual SCSI disk.
- **VirtualSize** MUST set to the virtual size of the shared virtual disk received from the virtual SCSI disk.

The response MUST be sent to the client.

### 3.2.5.5.2 Receiving a Connection Status Request

When the server receives a request with an **OperationCode** equal to RSVD\_TUNNEL\_CHECK\_CONNECTION\_STATUS\_OPERATION, the request handling proceeds as follows:

If **MaxOutputResponse** is less than 16 (size of SVHDX\_TUNNEL\_OPERATION\_HEADER), the server MUST fail the request with STATUS\_BUFFER\_OVERFLOW.

The server MUST construct an SVHDX\_TUNNEL\_CHECK\_CONNECTION\_RESPONSE structure as specified in section [2.2.4.2](#) with the following values:

The SVHDX\_TUNNEL\_OPERATION\_HEADER MUST be initialized as follows:

- **OperationCode** MUST be set to the OperationCode value of the request.
- **Status** MUST be set to STATUS\_SUCCESS.
- The **RequestId** field MUST be set to the value received in the request.

The response MUST be sent to the client.

### 3.2.5.5.3 Receiving a Sense Code Request

When the server receives a request with **OperationCode** equal to RSVD\_TUNNEL\_SRB\_STATUS\_OPERATION, the request handling proceeds as follows:

If **MaxOutputResponse** is less than 40 (size of SVHDX\_TUNNEL\_OPERATION\_HEADER + size of SVHDX\_TUNNEL\_SRB\_STATUS\_RESPONSE), the server MUST fail the request with STATUS\_INVALID\_PARAMETER.

If **SenseInfoExLength** field of the request is not equal to the size of the **SenseDataEx**, the server MUST fail the request with STATUS\_INVALID\_PARAMETER.

The server MUST locate the **SenseError** in **Open.SenseErrorDataList** where **SenseError.StatusKey** matches the **StatusKey** of the request.

If **SenseError** is not found, the server MUST return STATUS\_SVHDX\_ERROR\_NOT\_AVAILABLE.

If **SenseError** is found, the server MUST construct the SVHDX\_TUNNEL\_SRB\_STATUS\_RESPONSE structure as specified in section [2.2.4.4](#) with the following values:

The SVHDX\_TUNNEL\_OPERATION\_HEADER MUST be initialized as follows:

- The **OperationCode** field MUST be set to the OperationCode value of the request.
- The **Status** field MUST be set to STATUS\_SUCCESS.
- The **RequestId** field MUST be set to the value received in the request.

The SVHDX\_TUNNEL\_SRB\_STATUS\_RESPONSE packet MUST be initialized as follows:

- The **StatusKey** field MUST be set to the value received in the request.
- The **SenseInfoAutoGenerated** field MUST be set to the value received from the virtual SCSI disk.
- The **SrbStatus** field MUST be set to **SenseError.SrbStatus**.
- The **ScsiStatus** field MUST be set to **SenseError.ScsiStatus**.
- The **SenseInfoExLength** field MUST be set to the length of the **SenseError.SenseData**, in bytes.
- The **SenseDataEx** field MUST be set to **SenseError.SenseData**.

The response MUST be sent to the client.

### 3.2.5.5.4 Receiving a Shared Virtual Disk Information Request

When the server receives a request with an **OperationCode** equal to `RSVD_TUNNEL_GET_DISK_INFO_OPERATION`, the request handling proceeds as follows:

If **MaxOutputResponse** is less than 72 (size of `SVHDX_TUNNEL_OPERATION_HEADER` + size of `SVHDX_TUNNEL_DISK_INFO_RESPONSE`), the server MUST fail the request with `STATUS_BUFFER_TOO_SMALL`.

The server MUST issue a query disk information request to the virtual SCSI disk in an implementation-specific manner and update the received response to the client.

If the virtual SCSI disk indicates an error, the server MUST fail the request with the error received, placing into the **Status** field of the `SVHDX_TUNNEL_OPERATION_HEADER`, and return the header to the client.

Otherwise, the server MUST construct an `SVHDX_TUNNEL_DISK_INFO_RESPONSE` structure as specified in section [2.2.4.4](#) with the following values:

The `SVHDX_TUNNEL_OPERATION_HEADER` MUST be initialized as follows:

- **OperationCode** MUST be set to the `OperationCode` value of the request.
- **Status** MUST be set to `STATUS_SUCCESS`.
- The **RequestId** field MUST be set to the value received in the request.

The `SVHDX_TUNNEL_DISK_INFO_RESPONSE` structure MUST be initialized as follows:

- **DiskType** MUST be set to the value received from the virtual SCSI disk.
- **DiskFormat** MUST be set to `VIRTUAL_STORAGE_TYPE_DEVICE_VHDX`.
- **BlockSize** MUST be set to the number of bytes received from virtual SCSI disk.
- **LinkageId** MUST be set to the linkage GUID received from virtual SCSI disk.
- **IsMounted** MUST be set to `TRUE` if the virtual SCSI disk indicates that the disk is ready for read or write operations. Otherwise, the server MUST set this field to `FALSE`.
- **Is4kAligned** MUST be set to `TRUE` if the virtual SCSI disk indicates that the disk sectors are aligned to 4 kilobytes. Otherwise, the server MUST set this field to `FALSE`.
- **Reserved** MUST be set to zero.
- **FileSize** MUST be set to the value received from the virtual SCSI disk
- **VirtualDiskId** MUST be set to the virtual disk GUID received from virtual SCSI disk.

The response MUST be sent to the client.

### 3.2.5.5.5 Receiving a SCSI Command Request

When the server receives a request with an **OperationCode** equal to `RSVD_TUNNEL_SCSI_OPERATION`, the request handling proceeds as follows:

If **MaxOutputResponse** is less than 52 (size of SVHDX\_TUNNEL\_OPERATION\_HEADER + size of SVHDX\_TUNNEL\_SCSI\_RESPONSE), the server MUST fail the request with STATUS\_INVALID\_PARAMETER.

If any of the following conditions is TRUE, the server MUST generate an error response as specified below:

- Size of the request is less than 36
- **Length** field of the request is not equal to 36 bytes
- **SenseInfoExLength** is greater than (size of CDBBuffer + 4)
- **CDBLength** is greater than the size of **CDBBuffer**

The SVHDX\_TUNNEL\_OPERATION\_HEADER MUST be initialized. **ProtocolId**, **ProtocolVersion**, **OperationCode** and **RequestId** fields MUST be set to the values received in the request. **Status** field MUST be set to STATUS\_INVALID\_PARAMETER. The SVHDX\_TUNNEL\_SCSI\_RESPONSE MUST be set to the data received in SVHDX\_TUNNEL\_SCSI\_REQUEST structure of the request. The server MUST send SVHDX\_TUNNEL\_OPERATION\_HEADER and SVHDX\_TUNNEL\_SCSI\_RESPONSE to the client and SHOULD NOT [<7>](#) add additional data to the response.

Otherwise, the server MUST issue a CDB command to the virtual SCSI disk in an implementation-specific manner.

The server MUST construct a SVHDX\_TUNNEL\_SCSI\_RESPONSE structure as specified in section [2.2.4.8](#) with the following values:

The SVHDX\_TUNNEL\_OPERATION\_HEADER MUST be initialized as follows:

- The **OperationCode** field MUST be set to the **OperationCode** value of the request.
- The **Status** field MUST be set to the value received from the virtual SCSI disk.
- The **RequestId** field MUST be set to the value received in the request.

The SVHDX\_TUNNEL\_SCSI\_RESPONSE MUST be initialized as follows:

- The **SenseInfoAutoGenerated** field MUST be set to the value received from the virtual SCSI disk.
- The **SrbStatus** field MUST be set to one of the values specified in section [2.2.5](#) reflecting the status indicated by the virtual SCSI disk.
- The **SCSIStatus** field MUST be set to value received from the virtual SCSI disk.
- The **DataIn** field MUST be set to the value received in the request.
- The **Length** field is set to the size, in bytes, of the SVHDX\_TUNNEL\_SCSI\_RESPONSE structure that is constructed following the syntax specified in section [2.2.4.8](#).
- The **SenseInfoExLength** field is set to the length of the sense information received from the virtual SCSI disk, if any.
- The **SenseDataEx** field is set to the sense information received from the virtual SCSI disk, if any.

The response MUST be sent to the client.

### 3.2.5.5.6 Receiving a Validate Disk Request

When the server receives a request with an **OperationCode** equal to `RSVD_TUNNEL_VALIDATE_DISK_OPERATION`, the request handling proceeds as follows:

If **MaxOutputResponse** is less than 17 (size of `SVHDX_TUNNEL_OPERATION_HEADER` + size of `SVHDX_TUNNEL_VALIDATE_DISK_RESPONSE`), the server MUST fail the request with `STATUS_BUFFER_TOO_SMALL`.

The server MUST issue a validate request to the virtual SCSI disk in an implementation-specific manner.

The server MUST construct an `SVHDX_TUNNEL_VALIDATE_DISK_RESPONSE` structure, as specified in section [2.2.4.10](#), with the following values:

The `SVHDX_TUNNEL_OPERATION_HEADER` MUST be initialized as follows:

**OperationCode** MUST be set to the `OperationCode` value of the request.

If the virtual SCSI disk indicates command success, the **Status** MUST be set to `STATUS_SUCCESS`; otherwise, it MUST be set to the error code provided by the virtual SCSI disk.

The **RequestId** field MUST be set to the value received in the request.

The **IsValidDisk** field MUST be set to `TRUE` if the virtual SCSI disk indicates that the disk is valid. Otherwise, the server MUST set this field to `FALSE`.

The response MUST be sent to the client.

### 3.2.5.6 Receiving a Query Shared Virtual Disk Support Request

If **MaxOutputResponse** is less than 8 (size of `SVHDX_SHARED_VIRTUAL_DISK_SUPPORT_RESPONSE`), the server MUST fail the request with `STATUS_BUFFER_TOO_SMALL`.

The server MUST search the **OpenTable** where **Open.FileName** matches the file name.

If no **Open** is found, the server MUST set **SharedVirtualDiskHandleState** to **HandleStateNone**.

If any **Open** is found for which **Open.LocalOpen** matches the application-provided handle, the server MUST set **SharedVirtualDiskHandleState** to **HandleStateShared**.

Otherwise, the server MUST set **SharedVirtualDiskHandleState** to **HandleStateFileShared**.

The response MUST be sent to the client.

## 3.2.6 Timer Events

None.

## 3.2.7 Other Local Events

None.

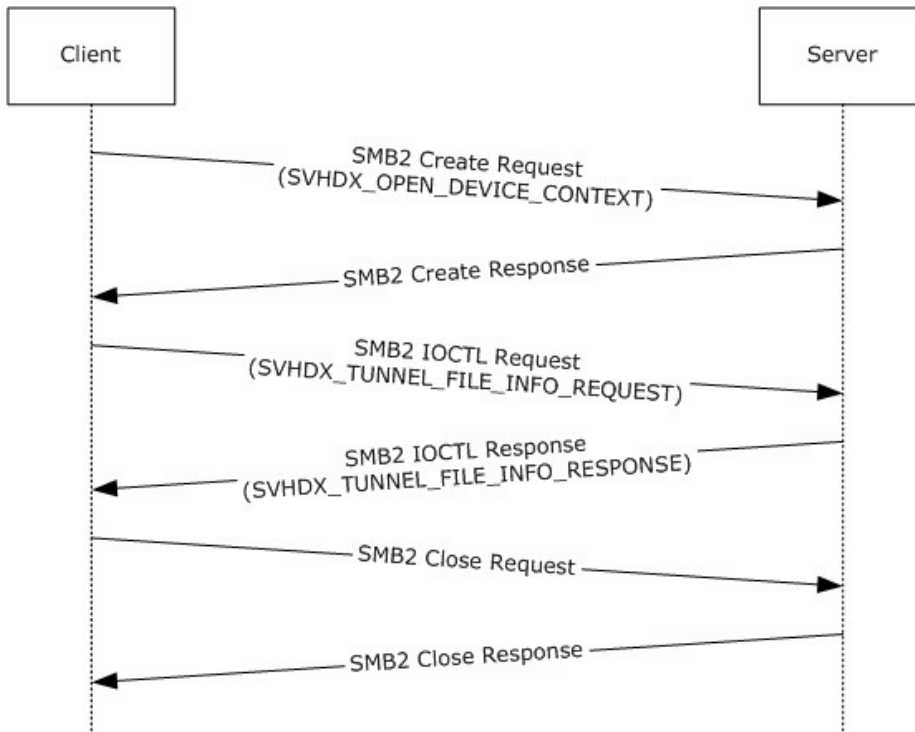


## 4 Protocol Examples

The following section describes common scenarios that indicate normal traffic flow in order to illustrate the function of the Remote Shared Virtual Disk (RSVD) Protocol.

### 4.1 Retrieving Virtual Disk File Information

The following diagram demonstrates the steps taken to open a shared virtual disk file, retrieve virtual disk file information, and close it.



**Figure 2: Retrieving virtual disk file information**

1. The client sends an SMB2 CREATE Request with the SVHDX\_OPEN\_DEVICE\_CONTEXT create context to open a shared virtual disk file.

```
Version: 1(0x00000001)
HasInitiatorId: 1
Reserved: 0(0x000000)
InitiatorId: (0x07770D201F2740834579D46F5AC43B73)
Flags : 0(0x00000000)
OriginatorFlags:1(0x00000001)
OpenRequestId: (0x000000001EC7871E)
InitiatorHostNameLength:16(0x0010)
InitiatorHostName:client01(0x003100300074006E00650069006C0063)
```

2. The server responds with an SMB2 CREATE Response giving the handle to the open identifying shared virtual disk file.

3. The client sends an SMB2 IOCTL Request with SVHDX\_TUNNEL\_FILE\_INFO\_REQUEST to retrieve the virtual disk information.

```
OperationCode: RSVD_TUNNEL_GET_INITIAL_INFO_OPERATION (0x02001001)
Status: 0 (0x00000000)
RequestId: (0x000000001Ec7871E)
ServerVersion: 0 (0x00000000)
SectorSize: 0 (0x00000000)
PhysicalSectorSize: 0 (0x00000000)
VirtualSize: 0 (0x0000000000000000)
```

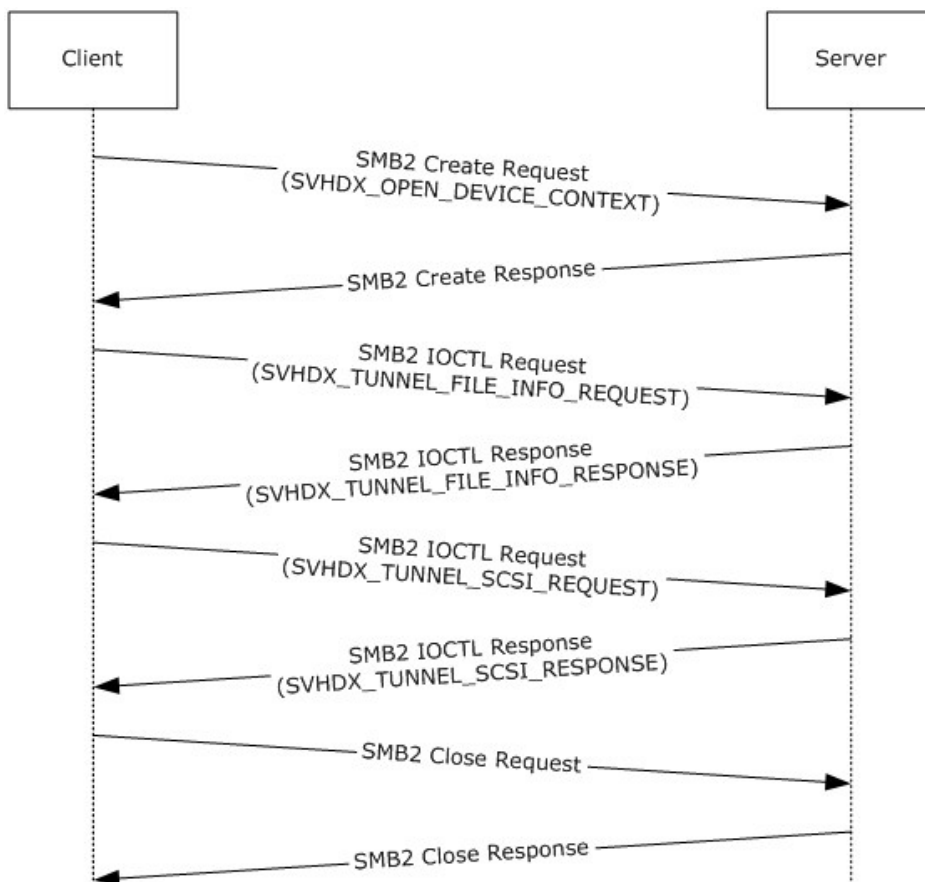
4. The server sends an SMB2 IOCTL Response with SVHDX\_TUNNEL\_FILE\_INFO\_RESPONSE containing the virtual disk information.

```
OperationCode: RSVD_TUNNEL_GET_FILE_INFO_OPERATION (0x02001001)
Status: 0 (0x00000000)
RequestId: (0x000000001Ec7871E)
ServerVersion: 1 (0x00000001)
SectorSize: 512 (0x00000200)
PhysicalSectorSize: 256 (0x00000100)
VirtualSize: (0x4000000000000000)
Reserved: 0 (0x00000000)
```

5. The client sends an SMB2 CLOSE Request to close the shared virtual disk file.
6. The server sends an SMB2 CLOSE Response indicating the close was successful.

## 4.2 Executing a SCSI Command

The following diagram demonstrates the steps taken to open a shared virtual disk file, execute a SCSI command, and close it.



**Figure 3: Executing a SCSI command**

1. The client sends an SMB2 CREATE Request with the SVHDX\_OPEN\_DEVICE\_CONTEXT create context to open a shared virtual disk file.

```

Version: 1 (0x00000001)
HasInitiatorId: 1
Reserved: 0 (0x000000)
InitiatorId: (0x07770D201F2740834579D46F5AC43B73)
Flags : 0 (0x00000000)
OriginatorFlags: 1 (0x00000001)
OpenRequestId: (0x000000001EC7871E)
InitiatorHostNameLength: 16 (0x0010)
InitiatorHostName: client01 (0x003100300074006E00650069006C0063)
  
```

2. The server responds with an SMB2 CREATE Response giving the handle to the open identifying shared virtual disk file.
3. The client sends an SMB2 IOCTL Request with SVHDX\_TUNNEL\_FILE\_INFO\_REQUEST to retrieve the virtual disk information.

```

OperationCode: RSVD_TUNNEL_GET_INITIAL_INFO_OPERATION (0x02001001)
Status: 0 (0x00000000)
  
```

```
RequestId:(0x000000001Ec7871E)
ServerVersion:0 (0x00000000)
SectorSize: 0 (0x00000000)
PhysicalSectorSize:0 (0x00000000)
VirtualSize:0 (0x0000000000000000)
```

4. The server sends an SMB2 IOCTL Response with SVHDX\_TUNNEL\_FILE\_INFO\_RESPONSE containing the virtual disk information.

```
OperationCode: RSVD_TUNNEL_GET_FILE_INFO_OPERATION (0x02001001)
Status: 0 (0x00000000)
RequestId: (0x000000001Ec7871E)
ServerVersion: 1 (0x00000001)
SectorSize: 512(0x00000200)
PhysicalSectorSize: 256(0x00000100)
VirtualSize: (0x4000000000000000)
Reserved: 0 (0x00000000)
```

5. The client sends an SMB2 IOCTL Request with SVHDX\_TUNNEL\_SCSI\_REQUEST to execute a SCSI command.

```
OperationCode: RSVD_TUNNEL_SCSI_OPERATION (0x02001002)
Status: 0 (0x00000000)
RequestId:(0x000000001Ec7871E)
Length:36 (0x0024)
Reserved1: 0 (0x0000)
CDBLength:16 (0x10)
SenseInfoExLength:20(0x14)
DataIn: 1(0x01)
Reserved2:0(0x00)
SrbFlags: (0x0020014A)
DataTransferLength: 8(0x00000008)
CDBBuffer: 37(0x0000000000000000000000000000000000000000000025)
Reserved3: 0(0x00000000)
DataBuffer: 0(0x000000000000000000)
```

6. The server sends an SMB2 IOCTL Response with SVHDX\_TUNNEL\_SCSI\_RESPONSE containing the response.

```
OperationCode: RSVD_TUNNEL_SCSI_OPERATION (0x02001002)
Status: 0 (0x00000000)
RequestId: (0x000000001Ec7871E)
Length: 36 (0x0024)
SrbStatus: 0(0x00)
ScsiStatus: 0(0x00)
CDBLength: 16(0x10)
SenseInfoExLength: 20(0x14)
DataIn: 1(0x01)
Reserved: 0(0x00)
SrbFlags: (0x0020014A)
DataTransferLength: 8(0x00000008)
SenseDataEx:37(0x0000000000000000000000000000000000000000000025)
DataBuffer: 0(0x000000000000000000)
```

7. The client sends an SMB2 CLOSE Request to close the shared virtual disk file.
8. The server sends an SMB2 CLOSE Response indicating the close was successful.

## **5 Security**

### **5.1 Security Considerations for Implementers**

None.

### **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 2012 R2 operating system
- Windows 8.1 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 1.7:](#) The following table illustrates Windows operating system versions that support RSVD clients and RSVD servers.

RSVD client	RSVD server
Windows Server 2012 R2 Windows 8.1	Windows Server 2012 R2

[<2> Section 3.1.3:](#) Windows 8.1 and Windows Server 2012 R2 set **ClientServiceVersion** to RSVD protocol version 1(0x00000001).

[<3> Section 3.1.3:](#) Windows-based RSVD clients initialize to zero.

[<4> Section 3.2.3:](#) Windows Server 2012 R2 sets **ServerServiceVersion** to RSVD Protocol version 1 (0x00000001).

[<5> Section 3.2.5.1:](#) If the **OriginatorFlags** in the request is 0x00000008, Windows Server 2012 R2 will not consider it as a virtual SCSI disk device and will incorrectly pass the request to the underlying object store.

[<6> Section 3.2.5.1:](#) If the **OriginatorFlags** in the request is 0x00000008, Windows Server 2012 R2 will incorrectly set **Open.IsVirtualSCSIDisk** to FALSE.

[<7> Section 3.2.5.5.5:](#) Windows Server 2012 R2 appends [**MaxOutputResponse** – (size of SVHDX\_TUNNEL\_OPERATION\_HEADER + size of SVHDX\_TUNNEL\_SCSI\_RESPONSE)] number of bytes, all set to zeroes, at the end of the response.

## 7 Change Tracking

This section identifies changes that were made to the [MS-RSVD] protocol document between the February 2014 and May 2014 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.
- The removal of a document from the documentation set.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **Editorial** means that the formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the technical content of the document is identical to the last released version.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.
- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.



- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated**.

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

The changes made to this document are listed in the following table. For more information, please contact [dochelp@microsoft.com](mailto:dochelp@microsoft.com).

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
<a href="#">2.2.1 Constants</a>	70982 Changed the value of the RSVD_MAXIMUM_NAME_LENGTH constant.	Y	Content updated.
<a href="#">2.2.1 Constants</a>	70986 Added Description column and description text.	N	Content updated.
<a href="#">2.2.2 Operation Codes</a>	70996 Updated the values of the control codes used in virtual disk operations.	Y	Content updated.
<a href="#">2.2.2 Operation Codes</a>	70986 Added Description column and description text.	N	Content updated.
<a href="#">2.2.3 Error Code</a>	71005 Added new error code STATUS_SVHDX_ERROR_STORED and its value.	Y	New content added.
<a href="#">2.2.3 Error Code</a>	70986 Added Description column and description text.	N	Content updated.
<a href="#">2.2.4.3 SVHDX_TUNNEL_SRB_STATUS_REQUEST Structure</a>	70931 Changed the composition of the SVHDX_TUNNEL_SRB_STATUS_REQUEST packet to include only a one-byte StatusKey field and a 27-byte Reserved field.	Y	Content updated.
<a href="#">2.2.4.4</a>	70931	Y	Content

<b>Section</b>	<b>Tracking number (if applicable) and description</b>	<b>Major change (Y or N)</b>	<b>Change type</b>
<a href="#">SVHDX_TUNNEL_SRB_STATUS_RESPONSE Structure</a>	Modified the structure of the SVHDX_TUNNEL_SRB_STATUS_RESPONSE packet by restricting the StatusKey field to 1 byte and removing the Reserved field.		Updated.
<a href="#">2.2.4.4 SVHDX_TUNNEL_SRB_STATUS_RESPONSE Structure</a>	71093 Changed SenseDataEx data type to variable.	Y	Content updated.
<a href="#">2.2.4.4 SVHDX_TUNNEL_SRB_STATUS_RESPONSE Structure</a>	70984 Added the A bit field to the structure; changed the description of the SrbStatus field to restrict the range of values.	Y	Content updated.
<a href="#">2.2.4.6 SVHDX_TUNNEL_DISK_INFO_RESPONSE Structure</a>	71038 Modified the description of the Disk Type (4 bytes) and Disk Format (4 bytes) fields.	N	Content updated.
<a href="#">2.2.4.7 SVHDX_TUNNEL_SCSI_REQUEST Structure</a>	70975 Modified the description of the Length field.	Y	Content updated.
<a href="#">2.2.4.7 SVHDX_TUNNEL_SCSI_REQUEST Structure</a>	71040 Updated Reserved1 (2 byte), Reserved2 (1 byte), and Reserved3 (4 bytes).	Y	Content updated.
<a href="#">2.2.4.7 SVHDX_TUNNEL_SCSI_REQUEST Structure</a>	70975 Changed the casing of the CdbLength field name to CDBLength.	Y	Content updated.
<a href="#">2.2.4.7 SVHDX_TUNNEL_SCSI_REQUEST Structure</a>	70986 Removed reference to deleted section 2.2.4.	Y	Content updated.
<a href="#">2.2.4.8 SVHDX_TUNNEL_SCSI_RESPONSE Structure</a>	70974 Updated the size of the SrbFlags field from 1 byte to 4.	Y	Content updated.
<a href="#">2.2.4.8 SVHDX_TUNNEL_SCSI_RESPONSE Structure</a>	70984 Modified the SrbStatus field description to reference [SPC-3] section 4.5.	N	Content updated.
<a href="#">2.2.4.8 SVHDX_TUNNEL_SCSI_RESPONSE Structure</a>	71093 Changed SenseDataEx data type to	Y	Content

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
	variable.		updated.
<a href="#">2.2.4.8 SVHDX_TUNNEL_SCSI_RESPONSE Structure</a>	70975 Changed the casing of the CdbLength field name to CDBLength.	Y	Content updated.
<a href="#">2.2.4.8 SVHDX_TUNNEL_SCSI_RESPONSE Structure</a>	70986 Removed reference to deleted section 2.2.4.	N	Content updated.
<a href="#">2.2.4.9 SVHDX_TUNNEL_VALIDATE_DISK_REQUEST Structure</a>	70930 Modified the specification of the SVHDX_TUNNEL_VALIDATE_DISK_REQUEST packet.	Y	Content updated.
<a href="#">2.2.4.10 SVHDX_TUNNEL_VALIDATE_DISK_RESPONSE Structure</a>	70930 Clarified the meaning of the IsValidDisk Boolean value and modified the specification of the SVHDX_TUNNEL_VALIDATE_DISK_RESPONSE structure.	Y	Content updated.
<a href="#">2.2.4.11 SVHDX_TUNNEL_OPERATION_HEADER Structure</a>	70996 Amended the header structure to include the ProtocolId and ProtocolVersion fields.	Y	Content updated.
<a href="#">2.2.4.12 SVHDX_OPEN_DEVICE_CONTEXT Structure</a>	70982 Clarified the sizes of the InitiatorHostNameLength value and the InitiatorHostName field.	Y	Content updated.
<a href="#">2.2.4.12 SVHDX_OPEN_DEVICE_CONTEXT Structure</a>	70986 Added Description column and description text.	N	Content updated.
<a href="#">2.2.4.14 SVHDX_TUNNEL_FILE_INFO_RESPONSE Structure</a>	70972 Amended the SVHDX_TUNNEL_FILE_INFO_RESPONSE structure and changed its field descriptions accordingly.	Y	Content updated.
<a href="#">2.2.4.14 SVHDX_TUNNEL_FILE_INFO_RESPONSE Structure</a>	70986 Changed SVHDX_TUNNEL_FILE_INFO_RESPONSE to SVHDX_TUNNEL_INITIAL_INFO_RESPONSE.	Y	Content updated.
<a href="#">2.2.4.15</a>	71095	Y	Content

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
<a href="#">SVHDX_SHARED_VIRTUAL_DISK_SUPPORT_REQUEST Structure</a>	Removed requirement that the request MUST contain only SVHDX_TUNNEL_OPERATION_HEADER.		Updated.
<a href="#">2.2.4.16 SVHDX_SHARED_VIRTUAL_DISK_SUPPORT_RESPONSE Structure</a>	71036 Amended the size of the SharedVirtualDiskSupport and SharedVirtualDiskHandleState fields to 4 bytes each; amended the values of the SharedVirtualDiskHandleState flags.	Y	Content updated.
<a href="#">2.2.5 SRB Status Code</a>	70984 Added section.	Y	New content added.
<a href="#">3.1.4.2 Application Requests Opening a Shared Virtual Disk</a>	70982 Clarified the settings of the InitiatorHostNameLength and InitiatorHostName fields.	Y	Content updated.
<a href="#">3.1.4.2 Application Requests Opening a Shared Virtual Disk</a>	70998 Added CreateOptions to the list of input parameters to the interface.	Y	Content updated.
<a href="#">3.1.4.2 Application Requests Opening a Shared Virtual Disk</a>	71004 Removed the instructions on constructing and initializing two FILE_FULL_EA_INFORMATION structures and on initializing the traffic EA.	Y	Content updated.
<a href="#">3.1.4.6 Application Requests Virtual Disk File information</a>	70910 Removed initialization instructions for the SVHDX_TUNNEL_FILE_INFO_REQUEST request and replaced its use as a parameter with the SVHDX_TUNNEL_OPERATION_HEADER structure.	Y	Content updated.
<a href="#">3.1.4.6 Application Requests Virtual Disk File information</a>	70986 Changed RSVD_TUNNEL_GET_FILE_INFO_OPERATION to RSVD_TUNNEL_GET_INITIAL_INFO_OPERATION in OperationCode bullet point.	Y	Content updated.
<a href="#">3.1.4.7 Application Requests Connection Status</a>	71010 Removed space from SVHDX_TUNNEL_OPERATION_HEADER name.	N	Content updated.

<b>Section</b>	<b>Tracking number (if applicable) and description</b>	<b>Major change (Y or N)</b>	<b>Change type</b>
<a href="#">3.1.4.9 Application Requests Execution of SCSI Command</a>	70975 Modified the SVHDX_TUNNEL_SCSI_REQUEST initialization instructions to exclude directions for the Length field.	Y	Content updated.
<a href="#">3.1.4.9 Application Requests Execution of SCSI Command</a>	70984 Modified the initialization instructions for the DataIn value of SVHDX_TUNNEL_SCSI_REQUEST to reference [SPC-3] section 3.1.24.	N	Content updated.
<a href="#">3.1.4.9 Application Requests Execution of SCSI Command</a>	70986 Clarified wording and removed reference to deleted section 2.2.4 in SCSI command flags bullet point.	N	Content updated.
<a href="#">3.1.5.3 Receiving a Read Response</a>	70931 Modified the initialization instructions for the SVHDX_TUNNEL_SRB_STATUS_REQUEST structure.	Y	Content updated.
<a href="#">3.1.5.4 Receiving a Write Response</a>	70931 Modified the initialization instructions for the SVHDX_TUNNEL_SRB_STATUS_REQUEST structure.	Y	Content updated.
<a href="#">3.2.1.2 Per Open</a>	70998 Added Open.CreateOptions to the list of Open members.	Y	Content updated.
<a href="#">3.2.1.2 Per Open</a>	70980 Added the field InitiatorId.	Y	Content updated.
<a href="#">3.2.1.3 Per SenseError in SenseErrorDataList</a>	70984 Amended the size of the SenseError.SrbStatus variable and added a reference to the permitted values.	Y	Content updated.
<a href="#">3.2.4.1 Handling a Tunnel Operation Request</a>	71041 Added section.	Y	New content added.
<a href="#">3.2.4.2 Handling a Shared Virtual Disk Support Request</a>	71041 Added section.	Y	New content

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
			added.
<a href="#">3.2.5.1 Receiving an Open Request</a>	70998 Added Open.CreateOptions to the Open initialization instructions for cases where the underlying object store or virtual SCSI disk returns success.	Y	Content updated.
<a href="#">3.2.5.1 Receiving an Open Request</a>	70994 Clarified two product behavior notes to specify Windows Server 2012 R2.	N	Product behavior note updated.
<a href="#">3.2.5.1 Receiving an Open Request</a>	71002 Indicated Windows 8.1 will incorrectly pass the request to the underlying object store and will incorrectly set Open.IsVirtualSCSIDisk to FALSE If the OriginatorFlags in the request is 0x00000000.	Y	Product behavior note updated.
<a href="#">3.2.5.1 Receiving an Open Request</a>	70980 Added the processing rules for Open.InitiatorId when HasInitiatorId is 0x00.	Y	Content updated.
<a href="#">3.2.5.1 Receiving an Open Request</a>	70986 Removed server processing rule for when the Version is not equal to RSVD_ECP_CONTEXT_VERSION_1.	Y	Content updated.
<a href="#">3.2.5.3 Receiving a Read Request</a>	70998 Added processing instructions for the case in which the Open is found and FILE_NO_INTERMEDIATE_BUFFERING is not set in Open.CreateOptions.	Y	Content updated.
<a href="#">3.2.5.3 Receiving a Read Request</a>	70980 Added the error message that is generated when Open is found and Open.InitiatorId is zero.	Y	Content updated.
<a href="#">3.2.5.4 Receiving a Write Request</a>	70998 Added processing instructions for the case in which the Open is found and FILE_NO_INTERMEDIATE_BUFFERING is not set in Open.CreateOptions.	Y	Content updated.
<a href="#">3.2.5.4 Receiving a Write Request</a>	71007 Changed Boolean syntax of STATUS_SVHDX_ERROR_STORED or SenseError.StatusKey to	Y	Content updated.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
	(STATUS_SVHDX_ERROR_STORED   SenseError.StatusKey).		
<a href="#">3.2.5.4 Receiving a Write Request</a>	70980 Added the error message that is generated when Open is found and Open.InitiatorId is zero.	Y	Content updated.
<a href="#">3.2.5.5 Receiving a Tunnel Operation Request</a>	70996 Updated the server processing instructions.	Y	Content updated.
<a href="#">3.2.5.5 Receiving a Tunnel Operation Request</a>	71040 Added error message that is generated when the size of the tunnel operation request including header is less than 16.	Y	New content added.
<a href="#">3.2.5.5 Receiving a Tunnel Operation Request</a>	70980 Added the error message that is generated when OperationCode is equal to RSVD_TUNNEL_SCSI_OPERATION and Open.InitiatorId is zero,	Y	Content updated.
<a href="#">3.2.5.5.1 Receiving a Virtual Disk File Information Request</a>	71041 Described fail code that occurs when MaxOutputResponse is less than 40.	Y	Content updated.
<a href="#">3.2.5.5.1 Receiving a Virtual Disk File Information Request</a>	70986 Changed RSVD_TUNNEL_GET_FILE_INFO_OPERATION to RSVD_TUNNEL_GET_INITIAL_INFO_OPERATION.	Y	Content updated.
<a href="#">3.2.5.5.2 Receiving a Connection Status Request</a>	71041 Described fail code that occurs when MaxOutputResponse is less than 16.	Y	Content updated.
<a href="#">3.2.5.5.3 Receiving a Sense Code Request</a>	70931 Modified the initialization instructions for the SVHDX_TUNNEL_SRB_STATUS_RESPONSE packet to remove the setting for the Reserved field.	Y	Content updated.
<a href="#">3.2.5.5.3 Receiving a Sense Code Request</a>	71041 Described fail code that occurs when MaxOutputResponse is less than 40.	Y	Content updated.
<a href="#">3.2.5.5.3</a>	70984	Y	Content

<b>Section</b>	<b>Tracking number (if applicable) and description</b>	<b>Major change (Y or N)</b>	<b>Change type</b>
<a href="#">Receiving a Sense Code Request</a>	Added initialization instructions for the SenseInfoAutoGenerated field.		t updated.
<a href="#">3.2.5.5.4 Receiving a Shared Virtual Disk Information Request</a>	71041 Described fail code that occurs when MaxOutputResponse is less than 72.	Y	Content updated.
<a href="#">3.2.5.5.5 Receiving a SCSI Command Request</a>	70975 Clarified that the server fails the request with STATUS_INVALID_PARAMETER if the Length field is not equal to 36 bytes.	Y	Content updated.
<a href="#">3.2.5.5.5 Receiving a SCSI Command Request</a>	71040 Clarified how request handling generates an error response.	Y	Content updated.
<a href="#">3.2.5.5.5 Receiving a SCSI Command Request</a>	70975 Changed the casing of the CdbLength field name to CDBLength.	Y	Content updated.
<a href="#">3.2.5.5.5 Receiving a SCSI Command Request</a>	71041 Described fail code that occurs when MaxOutputResponse is less than 52. Indicated that the server SHOULD NOT add additional data to the response, with a related product behavior note.	Y	Content updated.
<a href="#">3.2.5.5.5 Receiving a SCSI Command Request</a>	70984 Amended the server instructions for the construction of the SVHDX_TUNNEL_SCSI_RESPONSE structure.	Y	Content updated.
<a href="#">3.2.5.5.6 Receiving a Validate Disk Request</a>	70930 Updated server processing instructions for requests received with an OperationCode of RSVD_TUNNEL_VALIDATE_DISK_OPERATION.	Y	Content updated.
<a href="#">3.2.5.5.6 Receiving a Validate Disk Request</a>	71041 Described fail code that occurs when MaxOutputResponse is less than 17.	Y	Content updated.
<a href="#">3.2.5.6 Receiving a Query Shared Virtual Disk Support Request</a>	71036 Removed the values referenced after the HandleState constants.	Y	Content updated.



Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
<a href="#">3.2.5.6</a> <a href="#">Receiving a Query Shared Virtual Disk Support Request</a>	71041 Described fail code that occurs when MaxOutputResponse is less than 8. Removed handle and file name information provided by the calling application.	Y	Content updated.
<a href="#">4</a> <a href="#">Protocol Examples</a>	70997 Added new section.	Y	New content added.
<a href="#">4.1</a> <a href="#">Retrieving Virtual Disk File Information</a>	70997 Added section.	Y	New content added.
<a href="#">4.1</a> <a href="#">Retrieving Virtual Disk File Information</a>	70986 Changed RSVD_TUNNEL_GET_FILE_INFO_OPERATION to RSVD_TUNNEL_GET_INITIAL_INFO_OPERATION in step 3 of Retrieving virtual disk file information.	Y	Content updated.
<a href="#">4.2</a> <a href="#">Executing a SCSI Command</a>	70975 Changed the casing of the CdbLength field name to CDBLength.	Y	Content updated.
<a href="#">4.2</a> <a href="#">Executing a SCSI Command</a>	70997 Added section.	Y	New content added.
<a href="#">4.2</a> <a href="#">Executing a SCSI Command</a>	70986 Changed RSVD_TUNNEL_GET_FILE_INFO_OPERATION to RSVD_TUNNEL_GET_INITIAL_INFO_OPERATION in step 3 of Executing a SCSI command.	Y	Content updated.

## 8 Index

### A

[Applicability](#) 8

### C

[Capability negotiation](#) 8

[Change tracking](#) 48

Client

[initialization](#) 23

[other local events](#) 31

[timer events](#) 31

[timers](#) 23

[Constants message](#) 9

### E

[Error Code message](#) 10

### F

[Fields - vendor-extensible](#) 8

### G

[Glossary](#) 6

### I

[Implementer - security considerations](#) 46

[Index of security parameters](#) 46

[Informative references](#) 7

Initialization

[client](#) 23

[server](#) 32

[Introduction](#) 6

### M

Messages

[Constants message](#) 9

[Error Code message](#) 10

[Operation Codes message](#) 9

[SRB Status Code message](#) 21

[transport](#) 9

### N

[Normative references](#) 6

### O

[Operation Codes message](#) 9

Other local events

[client](#) 31

[server](#) 40

[Overview \(synopsis\)](#) 7

### P

[Parameters - security index](#) 46

[Preconditions](#) 7

[Prerequisites](#) 7

[Product behavior](#) 47

### R

References

[informative](#) 7

[normative](#) 6

[Relationship to other protocols](#) 7

### S

Security

[implementer considerations](#) 46

[parameter index](#) 46

Server

[initialization](#) 32

[other local events](#) 40

[timer events](#) 40

[timers](#) 32

[SRB Status Code message](#) 21

[Standards assignments](#) 8

### T

Timer events

[client](#) 31

[server](#) 40

Timers

[client](#) 23

[server](#) 32

[Tracking changes](#) 48

[Transport](#) 9

### V

[Vendor-extensible fields](#) 8

[Versioning](#) 8