# [MS-SMB2]: Server Message Block (SMB) Protocol Versions 2 and 3

**This topic lists the Errata found in [MS-SMB2] since it was last published. Since this topic is updated frequently, we recommend that you subscribe to these RSS or Atom feeds to receive update notifications.**

**Errata are subject to the same terms as the Open Specifications documentation referenced.**

RSS

Atom

Errata below are for Protocol Document Version V65.0 – 2021/10/06.

| **Errata Published*** | **Description** |
|---|---|
| 2022/01/25 | In Section 3.3.5.2.10, Verifying the Channel Sequence Number, reordered the text to reflect the behavior for windows 8 and others.<br>Changed from:<br>If the SMB2_FLAGS_REPLAY_OPERATION bit is not set in the Flags field of the SMB2 Header:<br>• If ChannelSequence in the SMB2 Header is equal to Open.ChannelSequence, the server MUST increment Open.OutstandingRequestCount by 1.<br>• Otherwise, if the unsigned difference using 16-bit arithmetic between ChannelSequence and Open.ChannelSequence is less than or equal to 0x7FFF, the server MUST increment Open.OutstandingPreRequestCount by Open.OutstandingRequestCount, and MUST set Open.OutstandingRequestCount to 1. The server MUST set Open.ChannelSequence to ChannelSequence in the SMB2 Header.<br>• Otherwise, the server MUST fail SMB2 WRITE, SET_INFO, and IOCTL requests with STATUS_FILE_NOT_AVAILABLE.<br>If the SMB2_FLAGS_REPLAY_OPERATION bit is set in the Flags field of the SMB2 Header:<br>• If ChannelSequence in the SMB2 Header is equal to Open.ChannelSequence and the following:<br>• If ChannelSequence in the SMB2 Header is equal to Open.ChannelSequence and Open.OutstandingPreRequestCount is equal to zero, the server MUST increment Open.OutstandingRequestCount by 1.<br>• Otherwise, if the unsigned difference using 16-bit arithmetic between ChannelSequence and Open.ChannelSequence is less than or equal to 0x7FFF and Open.OutstandingPreRequestCount is equal to zero, the server MUST increment Open.OutstandingPreRequestCount by Open.OutstandingRequestCount and MUST set Open.OutstandingRequestCount to 1. The server MUST set Open.ChannelSequence to ChannelSequence in the SMB2 Header.<br>• Otherwise, the server MUST fail SMB2 WRITE, SET_INFO, and IOCTL requests with STATUS_FILE_NOT_AVAILABLE.<br>Changed to:<br>If the SMB2_FLAGS_REPLAY_OPERATION bit is not set in the Flags field of the SMB2 Header, the server MUST do the following:<br>• If ChannelSequence in the SMB2 Header is equal to Open.ChannelSequence, the server MUST increment Open.OutstandingRequestCount by 1.<br>• Otherwise, if the unsigned difference using 16-bit arithmetic between ChannelSequence in the SMB2 header and Open.ChannelSequence is less than or equal to 0x7FFF, the server MUST perform the following: increment Open.OutstandingPreRequestCount by |

| Errata Published* | Description |
|---|---|
| | Open.OutstandingRequestCount, and MUST set Open.OutstandingRequestCount to 1. The server MUST set Open.ChannelSequence to ChannelSequence in the SMB2 Header.<br>● Increment Open.OutstandingPreRequestCount by Open.OutstandingRequestCount.<br>● Set Open.OutstandingRequestCount to 1.<br>● Set Open.ChannelSequence to ChannelSequence in the SMB2 Header.<br>● Otherwise, the server MUST fail SMB2 WRITE, SET_INFO, and IOCTL requests with STATUS_FILE_NOT_AVAILABLE.<br>If the SMB2_FLAGS_REPLAY_OPERATION bit is set in the Flags field of the SMB2 Header, the server MUST do the following:<br>● If ChannelSequence in the SMB2 Header is equal to Open.ChannelSequence,<br>● If Open.OutstandingPreRequestCount is equal to zero, the server MUST increment Open.OutstandingRequestCount by 1. Otherwise, the server MUST fail the SMB2 WRITE, SET_INFO, and IOCTL requests with STATUS_FILE_NOT_AVAILABLE.<br>● Otherwise, if the unsigned difference using 16-bit arithmetic between ChannelSequence in the SMB2 header and Open.ChannelSequence is less than or equal to 0x7FFF, the server SHOULD<WBN> do the following:<br>● Increment Open.OutstandingPreRequestCount by Open.OutstandingRequestCount.<br>● Set Open.ChannelSequence to ChannelSequence in the SMB2 Header.<br>● If Open.OutstandingPreRequestCount is equal to zero, set Open.OutstandingRequestCount to 1. Otherwise, set Open.OutstandingRequestCount to 0 and the server MUST fail the SMB2 WRITE, SET_INFO, and IOCTL requests with STATUS_FILE_NOT_AVAILABLE.<br><WBN> Windows 8 and Windows Server 2012 perform the following:<br>If Open.OutstandingPreRequestCount is equal to zero,<br>• Set Open.ChannelSequence to ChannelSequence in the SMB2 Header.<br>• Increment Open.OutstandingPreRequestCount by Open.OutstandingRequestCount<br>• Set Open.OutstandingRequestCount to 1.<br>Otherwise, fail the request with STATUS_FILE_NOT_AVAILABLE.<br>● Otherwise, the server MUST fail SMB2 WRITE, SET_INFO, and IOCTL requests with STATUS_FILE_NOT_AVAILABLE. |
| 2022/01/25 | In section 3.3.5.4, Receiving an SMB2 NEGOTIATE Request, updated the SKU to reflect the behavior for 21H1.<br>Changed from:<br>…<br>● Building an SMB2_COMPRESSION_CAPABILITIES negotiate response context:<br>…<br>● If Connection.CompressionIds is empty, |

| Errata Published* | Description |
|---|---|
| | ● The server SHOULD<265> set CompressionAlgorithmCount to 1.<br>● The server SHOULD<266> set CompressionAlgorithms to "NONE".<br>● <265> Section 3.3.5.4: Windows 10 v2004, Windows Server v2004, Windows 10 v20H2, and Windows Server v20H2 operating systems without [MSKB-5001391] set CompressionAlgorithmCount to 0.<br>● <266> Section 3.3.5.4: Windows 10 v2004, Windows Server v2004, Windows 10 v20H2, and Windows Server v20H2operating systems without [MSKB-5001391] set CompressionAlgorithms to empty.<br>Changed to:<br>…<br>● Building an SMB2_COMPRESSION_CAPABILITIES negotiate response context:<br>…<br>● If Connection.CompressionIds is empty,<br>● The server SHOULD<265> set CompressionAlgorithmCount to 1.<br>● The server SHOULD<266> set CompressionAlgorithms to "NONE".<br>● <265> Section 3.3.5.4: Windows 10 v2004, Windows Server v2004, Windows 10 v20H2, Windows Server v20H2, and Windows 10 v21H1 operating systems without [MSKB-5001391] set CompressionAlgorithmCount to 0.<br>● <266> Section 3.3.5.4: Windows 10 v2004, Windows Server v2004, Windows 10 v20H2, Windows Server v20H2, and Windows 10 v21H1 operating systems without [MSKB-5001391] set CompressionAlgorithms to empty. |
| 2022/01/25 | In Section 3.3.5.2.1.1 3.3.5.2.1.1 Decrypting the Message, updated processing of decrypting the message for both non-anonymous user is not authenitcated and authenticated cases.<br>Changed From:<br>If Session.IsAnonymous or Session.IsGuest is set to TRUE and the request is encrypted, then the server MUST disconnect the connection as specified in section 3.3.7.1.<br>Changed To:<br>If Connection.ConstrainedConnection is set to TRUE and the request is encrypted, then the server MUST disconnect the connection as specified in section 3.3.7.1.<br>If Connection.ConstrainedConnection is set to FALSE, Session.IsAnonymous or Session.IsGuest is set to TRUE and the request is encrypted, then the server SHOULD<WBN> disconnect the connection as specified in section 3.3.7.1.<br><WBN> Windows-based servers will not disconnect the connection. |
| 2021/12/14 | The following sections were changed. Please see the diff document for the details. |

| Errata Published* | Description |
|---|---|
| | In Section 2.2.3.1, SMB2 NEGOTIATE_CONTEXT Request Values, updated behavior notes to represent the correct SKUs:<br>Changed from:<br>SMB2_RDMA_TRANSFORM_CAPABILITIES<br>0x0007   The Data field contains a list of RDMA transforms, as specified in section 2.2.3.1.6.<16><br><16> Section 2.2.3.1:  Windows 10 v2004 operating system and prior and Windows Server v2004 v20H2 operating system and prior do not send or process SMB2_RDMA_TRANSFORM_CAPABILITIES.<br>Changed to:<br>0x0007   The Data field contains a list of RDMA transforms, as specified in section 2.2.3.1.6.<16><br><16> Section 2.2.3.1:  Windows 10 operating system and prior and Windows Server v20H2 operating system and prior do not send or process SMB2_RDMA_TRANSFORM_CAPABILITIES.<br>In Section 2.2.3.1, SMB2 NEGOTIATE_CONTEXT Request Values, updated behavior notes to represent the correct SKUs:<br>Changed from:<br>SMB2_SIGNING_CAPABILITIES<br>0x0008   The Data field contains a list of signing algorithms, as specified in section 2.2.3.1.7.<17><br><17> Section 2.2.3.1:  Windows 10 v20H2 operating system and prior and Windows Server v20H2 operating system and prior do not send or process SMB2_SIGNING_CAPABILITIES.<br>Changed to:<br>SMB2_SIGNING_CAPABILITIES<br>0x0008   The Data field contains a list of signing algorithms, as specified in section 2.2.3.1.7.<17><br><17> Section 2.2.3.1:  Windows 10 operating system and prior and Windows Server v20H2 operating system and prior do not send or process SMB2_SIGNING_CAPABILITIES.<br>In Section 2.2.10, SMB2 TREE_CONNECT Response, udpated behavior notes to represent the correct SKUs:<br>Changed from: |

| Errata Published* | Description |
|---|---|
|  | SMB2_SHAREFLAG_COMPRESS_DATA<br>0x00100000   The server supports compression of read/write messages on this share. This flag is only valid for the SMB 3.1.1 dialect.<33><br><33> Section 2.2.10: Windows 10 v20H2 and prior and Windows Server v20H2 and prior do not send or process this flag.<br>Changed to:<br>SMB2_SHAREFLAG_COMPRESS_DATA<br>0x00100000   The server supports compression of read/write messages on this share. This flag is only valid for the SMB 3.1.1 dialect.<33><br><33> Section 2.2.10: Windows 10 operating system and prior and Windows Server v20H2 operating system and prior do not send or process this flag.<br>In Section 2.2.20, SMB2 READ Response, udpated behavior notes to represent the correct SKUs:<br>Changed from:<br>SMB2_READFLAG_RESPONSE_RDMA_TRANSFORM<br>0x00000001   The Buffer field in the response contains SMB2_RDMA_TRANSFORM specified in section 2.2.43.<59><br><59> Section 2.2.20:  Windows 10 v2004 and prior and Windows Server v2004 and prior do not send or process SMB2_READFLAG_RESPONSE_RDMA_TRANSFORM flag.<br>Changed to:<br>SMB2_READFLAG_RESPONSE_RDMA_TRANSFORM<br>0x00000001   The Buffer field in the response contains SMB2_RDMA_TRANSFORM specified in section 2.2.43.<59><br><59> Section 2.2.20:  Windows 10 operating system and prior and Windows Server v20H2 operating system and prior do not send or process SMB2_READFLAG_RESPONSE_RDMA_TRANSFORM flag.<br>In Section 2.2.21 SMB2 WRITE Request, udpated behavior notes to represent the correct SKUs:<br>Changed from:<br>SMB2_CHANNEL_RDMA_TRANSFORM<br>0x00000003   This flag is not valid for SMB 3.0 and 3.0.2 dialects. When connection supports RDMA transform, SMB2_RDMA_TRANSFORM structure is present in the channel information |

| Errata Published* | Description |
|---|---|
| | specified by the RemainingBytes, WriteChannelInfoOffset, and WriteChannelInfoLength fields.<60><br><60> Section 2.2.21:  Windows 10 v2004 and prior and Windows Server v2004 and prior do not send or process SMB2_CHANNEL_RDMA_TRANSFORM flag.<br>Changed to:<br>SMB2_CHANNEL_RDMA_TRANSFORM<br>0x00000003   This flag is not valid for SMB 3.0 and 3.0.2 dialects. When connection supports RDMA transform, SMB2_RDMA_TRANSFORM structure is present in the channel information specified by the RemainingBytes, WriteChannelInfoOffset, and WriteChannelInfoLength fields.<60><br><60> Section 2.2.21:  Windows 10 operating system and prior and Windows Server v20H2 operating system and prior do not send or process SMB2_CHANNEL_RDMA_TRANSFORM flag.<br>In Section 2.2.43, SMB2_RDMA_TRANSFORM, udpated behavior notes to represent the correct SKUs:<br>Changed from:<br>The SMB2_RDMA_TRANSFORM is used by the client or server to send/receive transformed RDMA payload in READ/WRITE operations. The SMB2_RDMA_TRANSFORM is optional and only valid for the SMB 3.1.1 dialect when connection supports RDMA transform.<82><br><82> Section 2.2.43:  Windows 10 v2004 operating system and prior and Windows Server v2004 operating system and prior do not send or process RDMA transforms.<br>Changed to:<br>The SMB2_RDMA_TRANSFORM is used by the client or server to send/receive transformed RDMA payload in READ/WRITE operations. The SMB2_RDMA_TRANSFORM is optional and only valid for the SMB 3.1.1 dialect when connection supports RDMA transform.<82><br><82> Section 2.2.43:  Windows 10 operating system and prior and Windows Server v20H2 operating system and prior do not send or process RDMA transforms.<br>In Section 2.2.43.1, SMB2_RDMA_CRYPTO_TRANSFORM, udpated behavior notes to represent the correct SKUs:<br>Changed from:<br>The SMB2_RDMA_CRYPTO_TRANSFORM is used by the client or server to send/receive encrypted or signed RDMA payload in READ/WRITE operations. The SMB2_RDMA_CRYPTO_TRANSFORM is optional and only valid for the SMB 3.1.1 dialect.<83><br><83> Section 2.2.43.1:  Windows 10 v2004 and prior and Windows Server v2004 operating system and prior do not send or process RDMA transforms.<br>Changed to: |

| Errata Published* | Description |
|---|---|
| | The SMB2_RDMA_CRYPTO_TRANSFORM is used by the client or server to send/receive encrypted or signed RDMA payload in READ/WRITE operations. The SMB2_RDMA_CRYPTO_TRANSFORM is optional and only valid for the SMB 3.1.1 dialect.<83><br><83> Section 2.2.43.1: Windows 10 operating system and prior and Windows Server v20H2 operating system and prior do not send or process RDMA transforms.<br>In Section 3.1.3, Initialization, udpated behavior notes to represent the correct SKUs:<br>Changed from:<br>IsRDMATransformSupported MUST be set in an implementation-specific manner.<88><br>DisableEncryptionOverSecureTransport MUST be set in an implementation-specific manner.<89><br><88> Section 3.1.3: Windows 10 v20H2 and later and Windows Server v20H2 and later set IsRDMATransformSupported to TRUE.<br><89> Section 3.1.3: Windows 10 v21H1 operating system and Windows Server 2022 set this to TRUE.<br>Changed to:<br>IsRDMATransformSupported MUST be set in an implementation-specific manner.<88><br>DisableEncryptionOverSecureTransport MUST be set in an implementation-specific manner.<89><br><88> Section 3.1.3: Windows 11 operating system and later and Windows Server 2022 operating system and later set IsRDMATransformSupported to TRUE.<br><89> Section 3.1.3: Windows 11 operating system and later and Windows Server 2022 operating system and later set this to TRUE.<br>In Section 3.2.4.2.2.2, SMB2-Only Negotiate, udpated behavior notes to represent the correct SKUs:<br>Changed from:<br>● If an alternate connection is being established to an already connected Server, set Ciphers to Server.CipherId and CipherCount to 1. Otherwise, set Ciphers with the ciphers supported by the client, if any, in the order of preference and CipherCount to number of ciphers in Ciphers field.<121><br><121> Section 3.2.4.2.2.2: Windows 10 v1507 operating system through Windows 10 v20H2 and Windows Server 2016 through Windows Server v20H2 initialize with AES-128-GCM(0x0002), followed by AES-128-CCM(0x0001). |

| Errata Published* | Description |
|---|---|
| | Windows 10 v21H1 and later and Windows Server 2022 and later initialize with AES-128-GCM(0x0002), followed by AES-128-CCM(0x0001), followed by AES-256-GCM(0x0004), followed by AES-256-CCM(0x0003).<br>● CompressionAlgorithms SHOULD<122> be set to the algorithms supported by the client in the order of preference.<br><122> Section 3.2.4.2.2.2: Windows 10 v1903, Windows 10 v1909, Windows Server v1903 and Windows Server v1909 operating systems initialize with LZ77(0x0002) followed by LZ77+Huffman(0x0003) followed by LZNT1(0x0001).<br>Windows 10 v2004 and later and Windows Server v2004 operating systems initialize with Pattern_V1(0x0004) followed by LZ77(0x0002) followed by LZ77+Huffman(0x0003) followed by LZNT1(0x0001).<br>● If an alternate connection is being established to an already connected Server, set RDMATransformIds to Server.RDMATransformIds. Otherwise, set RDMATransformIds to the RDMA transforms in an implementation-defined manner.<123><br><123> Section 3.2.4.2.2.2: Windows 10 v20H2 and Windows Server v20H2 set RDMATransformIds to SMB2_RDMA_TRANSFORM_ENCRYPTION (0x0001).<br>Windows 10 v21H1 and later and Windows Server 2022 and later set RDMATransformIds to SMB2_RDMA_TRANSFORM_ENCRYPTION (0x0001) and SMB2_RDMA_TRANSFORM_SIGNING (0x0002).<br>● If the client implements the SMB 3.1.1 dialect, the client SHOULD<125> add an SMB2 NEGOTIATE_CONTEXT with ContextType as SMB2_SIGNING_CAPABILITIES to the negotiate request as specified in section 2.2.3.1:<br><125> Section 3.2.4.2.2.2: Windows 10 v20H2 and prior and Windows Server v20H2 and prior do not send or process SMB2_SIGNING_CAPABILITIES negotiate context.<br>● If an alternate connection is being established to an already connected Server, set SigningAlgorithms to Server.SigningAlgorithmId and set SigningAlgorithmCount to 1. Otherwise, set SigningAlgorithms to the signing algorithms supported by the client, if any, in the order of preference, and set SigningAlgorithmCount to the number of elements in the SigningAlgorithms field.<126><br><126> Section 3.2.4.2.2.2: Windows 10 v21H1 and Windows Server 2022 initialize with AES-GMAC(0x0002) followed by AES-CMAC(0x0001) followed by HMAC-SHA256(0x0000).<br>Changed to:<br>● If an alternate connection is being established to an already connected Server, set Ciphers to Server.CipherId and CipherCount to 1. Otherwise, set Ciphers with the ciphers supported by the client, if any, in the order of preference and CipherCount to number of ciphers in Ciphers field.<121><br><121> Section 3.2.4.2.2.2: Windows 10 operating system and Windows Server 2016 through Windows Server v20H2 initialize with AES-128-GCM(0x0002), followed by AES-128-CCM(0x0001). |

| Errata Published* | Description |
|---|---|
| | Windows 11 operating system and later and Windows Server 2022 operating system and later initialize with AES-128-GCM(0x0002), followed by AES-128-CCM(0x0001), followed by AES-256-GCM(0x0004), followed by AES-256-CCM(0x0003).<br>● CompressionAlgorithms SHOULD<122> be set to the algorithms supported by the client in the order of preference.<br><122> Section 3.2.4.2.2.2: Windows 10 v1903, Windows 10 v1909, Windows Server v1903 and Windows Server v1909 operating systems initialize with LZ77(0x0002) followed by LZ77+Huffman(0x0003) followed by LZNT1(0x0001).<br>Windows 10 v2004 operating system and later and Windows Server v2004 operating system and later initialize with Pattern_V1(0x0004) followed by LZ77(0x0002) followed by LZ77+Huffman(0x0003) followed by LZNT1(0x0001).<br>● If an alternate connection is being established to an already connected Server, set RDMATransformIds to Server.RDMATransformIds. Otherwise, set RDMATransformIds to the RDMA transforms in an implementation-defined manner.<123><br><123> Windows 10 11 operating system and later and Windows Server 2022 operating system and later set RDMATransformIds to SMB2_RDMA_TRANSFORM_ENCRYPTION (0x0001) and SMB2_RDMA_TRANSFORM_SIGNING (0x0002).<br>● If the client implements the SMB 3.1.1 dialect, the client SHOULD<125> add an SMB2 NEGOTIATE_CONTEXT with ContextType as SMB2_SIGNING_CAPABILITIES to the negotiate request as specified in section 2.2.3.1:<br><125> Section 3.2.4.2.2.2: Windows 10 operating system and prior and Windows Server v20H2 operating system and prior do not send or process SMB2_SIGNING_CAPABILITIES negotiate context.<br>● If an alternate connection is being established to an already connected Server, set SigningAlgorithms to Server.SigningAlgorithmId and set SigningAlgorithmCount to 1. Otherwise, set SigningAlgorithms to the signing algorithms supported by the client, if any, in the order of preference, and set SigningAlgorithmCount to the number of elements in the SigningAlgorithms field.<126><br><126> Section 3.2.4.2.2.2: Windows 10 11 operating system and later and Windows Server 2022 operating system and later initialize with AES-GMAC(0x0002) followed by AES-CMAC(0x0001) followed by HMAC-SHA256(0x0000).<br>In Section 3.3.3, Initialization, udpated behavior notes to represent the correct SKUs:<br>Changed from:<br>● AllowNamedPipeAccessOverQUIC MUST be set in an implementation-specific<212> manner.<br><212> Section 3.3.3: By default, Windows 10 v21H1 and Windows Server 2022 set AllowNamedPipeAccessOverQUIC to FALSE.<br>Changed to: |

| Errata Published* | Description |
|---|---|
| | ● AllowNamedPipeAccessOverQUIC MUST be set in an implementation-specific<212> manner.<br><212> Section 3.3.3: By default, Windows 10 11 operating system and later and Windows Server 2022 operating system and later set AllowNamedPipeAccessOverQUIC to FALSE.<br>In Section 3.3.5.15, Receiving an SMB2 IOCTL Request, udpated behavior notes to represent the correct SKUs:<br>Changed from:<br>The server SHOULD<353> fail the request with STATUS_NOT_SUPPORTED when an FSCTL is not allowed on the server, and SHOULD<354> fail the request with STATUS_INVALID_DEVICE_REQUEST when the FSCTL is allowed, but is not supported on the file system on which the file or directory handle specified by the FSCTL exists, as specified in [MS-FSCC] section 2.2.<br><353> Section 3.3.5.15: Windows 8 and later and Windows Server 2012 and later allow only the CtlCode values, as specified in section 2.2.31, and the following CtlCode values, as specified in [MS-FSCC] section 2.3.<br>Windows 10 v21H1 operating system and later and Windows Server 2022 and later allow the additional CtlCode value, as specified in [MS-FSCC].<br>FSCTL name FSCTL function number<br>FSCTL_GET_RETRIEVAL_POINTERS_AND_REFCOUNT 0x903D3<br>FSCTL_GET_RETRIEVAL_POINTER_COUNT 0x9042B<br>FSCTL_REFS_STREAM_SNAPSHOT_MANAGEMENT 0x90440<br>Changed to:<br>Changed from:<br>The server SHOULD<353> fail the request with STATUS_NOT_SUPPORTED when an FSCTL is not allowed on the server, and SHOULD<354> fail the request with STATUS_INVALID_DEVICE_REQUEST when the FSCTL is allowed, but is not supported on the file system on which the file or directory handle specified by the FSCTL exists, as specified in [MS-FSCC] section 2.2.<br><353> Section 3.3.5.15: Windows 8 and later and Windows Server 2012 and later allow only the CtlCode values, as specified in section 2.2.31, and the following CtlCode values, as specified in [MS-FSCC] section 2.3.<br>Windows 10 11 operating system and later and Windows Server 2022 and later allow the additional CtlCode value, as specified in [MS-FSCC].<br>FSCTL name FSCTL function number<br>FSCTL_GET_RETRIEVAL_POINTERS_AND_REFCOUNT 0x903D3 |

<table>
<tr><th>Errata Published*</th><th>Description</th></tr>
<tr><td></td><td>FSCTL_GET_RETRIEVAL_POINTER_COUNT   0x9042B<br>FSCTL_REFS_STREAM_SNAPSHOT_MANAGEMENT   0x90440</td></tr>
<tr><td>2021/12/14</td><td>In Section 3.3.5.15 Receiving an SMB2 IOCTL Request, the following was updated<br><br>Changed From:<br><br>Windows 10 v21H1 and later and Windows Server 2022 and later allow the additional CtlCode value, as specified in [MS-FSCC].
<table>
<tr><th>FSCTL name</th><th>FSCTL function number</th></tr>
<tr><td>FSCTL_GET_RETRIEVAL_POINTERS_AND_REFCOUNT</td><td>0x903D3</td></tr>
<tr><td>FSCTL_GET_RETRIEVAL_POINTER_COUNT</td><td>0x9042B</td></tr>
<tr><td>FSCTL_REFS_STREAM_SNAPSHOT_MANAGEMENT</td><td>0x90440</td></tr>
<tr><td>FSCTL_SET_INTEGRITY_INFORMATION_EX</td><td>0x90380</td></tr>
</table>
<br>Changed to:<br>Windows 10 v21H1 and later and Windows Server 2022 and later allow the additional CtlCode value, as specified in [MS-FSCC].
<table>
<tr><th>FSCTL name</th><th>FSCTL function number</th></tr>
<tr><td>FSCTL_GET_RETRIEVAL_POINTERS_AND_REFCOUNT</td><td>0x903D3</td></tr>
<tr><td>FSCTL_GET_RETRIEVAL_POINTER_COUNT</td><td>0x9042B</td></tr>
<tr><td>FSCTL_REFS_STREAM_SNAPSHOT_MANAGEMENT</td><td>0x90440</td></tr>
</table>
<br>Windows Server 2022 and later allow the additional CtlCode value, as specified in [MS-FSCC].
<table>
<tr><th>FSCTL name</th><th>FSCTL function number</th></tr>
<tr><td>FSCTL_SET_INTEGRITY_INFORMATION_EX</td><td>0x90380</td></tr>
</table>
</td></tr>
<tr><td>2021/12/14</td><td>In Section 3.2.5.14.12 Handling a Validate Negotiate Info Response, updated processing of validate negotiate info response when neither signed nor encrypted.<br><br>Changed From:<br><br>If the response is not signed or the signature verification in section 3.2.5.1.3 does not succeed, the client MUST terminate the Connection.<br><br>Changed To:<br><br>If the response is neither signed nor encrypted, the client MUST terminate the Connection.</td></tr>
<tr><td>2021/12/14</td><td>In Section 3.3.5.15 Receiving an SMB2 IOCTL Request, the following was updated</td></tr>
</table>

<table>
<tr><th>Errata Published*</th><th>Description</th></tr>
<tr><td></td><td>
Changed From:<br><br>
Windows 10 v21H1 and later and Windows Server 2022 and later allow the additional CtlCode value, as specified in [MS-FSCC].
<table>
<tr><th>FSCTL name</th><th>FSCTL function number</th></tr>
<tr><td>FSCTL_GET_RETRIEVAL_POINTERS_AND_REFCOUNT</td><td>0x903D3</td></tr>
<tr><td>FSCTL_GET_RETRIEVAL_POINTER_COUNT</td><td>0x9042B</td></tr>
<tr><td>FSCTL_REFS_STREAM_SNAPSHOT_MANAGEMENT</td><td>0x90440</td></tr>
<tr><td>FSCTL_SET_INTEGRITY_INFORMATION_EX</td><td>0x90380</td></tr>
</table>
<br>
Changed to:<br>
Windows 10 v21H1 and later and Windows Server 2022 and later allow the additional CtlCode value, as specified in [MS-FSCC].
<table>
<tr><th>FSCTL name</th><th>FSCTL function number</th></tr>
<tr><td>FSCTL_GET_RETRIEVAL_POINTERS_AND_REFCOUNT</td><td>0x903D3</td></tr>
<tr><td>FSCTL_GET_RETRIEVAL_POINTER_COUNT</td><td>0x9042B</td></tr>
<tr><td>FSCTL_REFS_STREAM_SNAPSHOT_MANAGEMENT</td><td>0x90440</td></tr>
</table>
<br>
Windows Server 2022 and later allow the additional CtlCode value, as specified in [MS-FSCC].
<table>
<tr><th>FSCTL name</th><th>FSCTL function number</th></tr>
<tr><td>FSCTL_SET_INTEGRITY_INFORMATION_EX</td><td>0x90380</td></tr>
</table>
</td></tr>
</table>

*Date format: YYYY/MM/DD