

## [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.

Errata below are for Protocol Document Version [V60.0 – 2020/03/04](#).

Errata Published*	Description
2020/05/25	<p>In Section 6, Appendix A: Product Behavior, product behavior note &lt;341&gt; for Section 3.3.5.15.8 was changed from:</p> <p>...</p> <p>FSCTL_IS_FILE_ON_CSV_VOLUME (0x0009025C)</p> <p>Windows-based SMB2 servers fail FSCTLs whose transfer type is METHOD_NEITHER with error STATUS_NOT_SUPPORTED except the following ones. For more information about FSCTL transfer type, see [MSDN-IoCtlCodes].</p> <p>FSCTL_PIPE_TRANSCEIVE (0x0011C017)</p> <p>...</p> <p>Changed to:</p> <p>...</p> <p>FSCTL_IS_FILE_ON_CSV_VOLUME (0x0009025C)</p> <p>Windows Vista SP1, Windows 7, Windows Server 2008, and Windows Server 2008 R2 fail FSCTLs whose transfer type is METHOD_NEITHER with error STATUS_NOT_SUPPORTED except the following ones. For more information about FSCTL transfer type, see [MSDN-IoCtlCodes].</p> <p>FSCTL_PIPE_TRANSCEIVE (0x0011C017)</p> <p>...</p> <p>For a Diff of the above changes, see the PDF doc <a href="#">here</a>.</p>
2020/05/25	<p>In Section 3.1.5.3, Decompressing the Chained Message, the following was changed from:</p> <p>If IsCompressionSupported is FALSE, Connection.SupportsChainedCompression is FALSE, or Connection.CompressionIds is empty, the receiver MUST skip the processing in this section.</p> <p>1. The sender MUST initialize RemainingCompressedDataSize with the size of the received compressed SMB2 message and DecompressedMessage with empty buffer.</p>

Errata Published*	Description
	<p>2. The compressed message MUST be decompressed until the size of RemainingCompressedDataSize is greater than the size of SMB2COMPRESSION_PAYLOAD_HEADER:</p> <ul style="list-style-type: none"> <li>• The first 8 bytes of the data MUST be interpreted as SMB2_COMPRESSION_PAYLOAD_HEADER, specified in section 2.2.42.1.</li> <li>• If AlgorithmId in SMB2_COMPRESSION_PAYLOAD_HEADER is not one of the values specified in section 2.2.3.1.3, the connection MUST be disconnected as specified in section 3.2.7.1 or 3.3.7.1.</li> <li>• If AlgorithmId in SMB2_COMPRESSION_PAYLOAD_HEADER is NONE:</li> <li>• If Length is greater than (the size of the received compressed message – 8) or OriginalCompressedSegmentSize in SMB2 COMPRESSION_TRANSFORM_HEADER, the connection MUST be disconnected as specified in section 3.2.7.1 or 3.3.7.1.</li> <li>• Length number of bytes following SMB2_COMPRESSION_PAYLOAD_HEADER MUST be interpreted as uncompressed data and MUST be appended to DecompressedMessage.</li> <li>• Otherwise, the data MUST be decompressed as follows: <ul style="list-style-type: none"> <li>• If AlgorithmId is Pattern_V1, the next 8 bytes MUST be interpreted as SMB2_COMPRESSION_PATTERN_PAYLOAD_V1, specified in section 2.2.42.2.</li> <li>• If Repetitions in SMB2_COMPRESSION_PATTERN_PAYLOAD_V1 is greater than OriginalCompressedSegmentSize in SMB2 COMPRESSION_TRANSFORM_HEADER, the connection MUST be disconnected as specified in section 3.2.7.1 or 3.3.7.1.</li> <li>• Otherwise, DecompressedMessage MUST be appended with Repetitions number of bytes initialized with the character specified in Pattern field.</li> <li>• Otherwise, the data of size specified in Length field MUST be decompressed using the algorithm specified in AlgorithmId field as specified in [MS-XCA] section 2. DecompressedMessage MUST be appended with the decompressed data.</li> <li>• RemainingCompressedDataSize MUST be decremented by the size in Length field.</li> </ul> </li> </ul> <p>3. DecompressedMessage MUST be returned.</p> <p>Changed to:</p> <p>If IsCompressionSupported is FALSE, Connection.SupportsChainedCompression is FALSE, or Connection.CompressionIds is empty, the receiver MUST skip the processing in this section.</p> <p>1. The receiver MUST initialize RemainingCompressedDataSize with the size of the received compressed SMB2 message and DecompressedMessage with empty buffer.</p>

Errata Published*	Description
	<p>2. If the size of RemainingCompressedDataSize is greater than the size of SMB2_COMPRESSION_PAYLOAD_HEADER, the compressed message MUST be decompressed as follows:</p> <ul style="list-style-type: none"> <li>• The first 8 bytes of the data MUST be interpreted as SMB2_COMPRESSION_PAYLOAD_HEADER, specified in section 2.2.42.1.</li> <li>• If AlgorithmId in SMB2_COMPRESSION_PAYLOAD_HEADER is not one of the values specified in section 2.2.3.1.3, the connection MUST be disconnected as specified in section 3.2.7.1 or 3.3.7.1.</li> <li>• If AlgorithmId in SMB2_COMPRESSION_PAYLOAD_HEADER is NONE:</li> <li>• If Length is greater than (the size of the received compressed message – 8) or OriginalCompressedSegmentSize in SMB2_COMPRESSION_TRANSFORM_HEADER, the connection MUST be disconnected as specified in section 3.2.7.1 or 3.3.7.1.</li> <li>• Length number of bytes following SMB2_COMPRESSION_PAYLOAD_HEADER MUST be interpreted as uncompressed data and MUST be appended to DecompressedMessage.</li> <li>• Otherwise, the data MUST be decompressed as follows: <ul style="list-style-type: none"> <li>• If AlgorithmId is Pattern_V1, the next 8 bytes MUST be interpreted as SMB2_COMPRESSION_PATTERN_PAYLOAD_V1, specified in section 2.2.42.2.</li> <li>• If Repetitions in SMB2_COMPRESSION_PATTERN_PAYLOAD_V1 is greater than OriginalCompressedSegmentSize in SMB2_COMPRESSION_TRANSFORM_HEADER, the connection MUST be disconnected as specified in section 3.2.7.1 or 3.3.7.1.</li> <li>• Otherwise, DecompressedMessage MUST be appended with Repetitions number of bytes initialized with the character specified in Pattern field.</li> <li>• Otherwise, the data of size specified in Length field MUST be decompressed using the algorithm specified in AlgorithmId field as specified [MS-XCA] section 2. DecompressedMessage MUST be appended with the decompressed data.</li> <li>• RemainingCompressedDataSize MUST be decremented by the size in Length field.</li> <li>• If the size of RemainingCompressedDataSize is greater than the size of SMB2_COMPRESSION_PAYLOAD_HEADER, the receiver MUST repeat step 2.</li> </ul> </li> <li>3. DecompressedMessage MUST be returned.</li> </ul> <p>For a Diff of the above changes, see the PDF doc <a href="#">here</a>.</p>
2020/05/25	<p>In Section 3.1.4.4, Compressing the Message, the following was changed from:</p> <ol style="list-style-type: none"> <li>1. The sender MUST initialize RemainingUncompressedDataSize with the size of uncompressed SMB2 message, TotalCompressedDataSize with 0, and CompressedMessage with empty buffer.</li> <li>2. The message MUST be compressed until RemainingUncompressedDataSize is greater than zero:</li> </ol>

Errata Published*	Description
	<ul style="list-style-type: none"> <li>• If Connection.CompressionIds includes Pattern_V1, message MUST be scanned for data patterns as specified in section 3.1.4.4.1. If the returned FrontDataPattern.Repetitions is greater than zero, CompressedMessage MUST be appended with newly constructed SMB2_COMPRESSION_PAYLOAD_HEADER. AlgorithmId MUST be set to Pattern_V1. Length MUST be set to the size of SMB2_COMPRESSION_PATTERN_PAYLOAD_V1. CompressedMessage MUST be appended with front data pattern returned by section 3.1.4.4.1. RemainingUncompressedDataSize MUST be decremented by FrontDataPattern.Repetitions. TotalCompressedDataSize MUST be incremented by FrontDataPattern.Repetitions.</li> <li>• If RemainingUncompressedDataSize is greater than 1024, CompressedMessage MUST be appended with newly constructed SMB2_COMPRESSION_PAYLOAD_HEADER. AlgorithmId MUST be set to first preferred algorithm in Connection.CompressionIds. The data MUST be compressed using the algorithm specified in AlgorithmId as specified in [MS-XCA] section 2. Length MUST be set to the size of the compressed data. A 4-byte field, indicating the size of the original plain text size of the data compressed, MUST be appended to CompressedMessage. CompressedMessage MUST be appended with the compressed data. RemainingUncompressedDataSize MUST be decremented by the size of data before compression. TotalCompressedDataSize MUST be incremented by the size of compressed data.</li> <li>• Otherwise if RemainingUncompressedDataSize is greater than zero, if (size of the uncompressed SMB2 message / RemainingUncompressedDataSize) is greater than 2, CompressedMessage MUST be appended with newly constructed SMB2_COMPRESSION_PAYLOAD_HEADER. AlgorithmId MUST be set to NONE. Length MUST be set to RemainingUncompressedDataSize. CompressedMessage MUST be appended with the Remaining uncompressed data. RemainingUncompressedDataSize MUST be decremented by the size of data before compression. TotalCompressedDataSize MUST be incremented by the size of compressed data.</li> <li>• If BackDataPattern.Repetitions is greater than zero, CompressedMessage MUST be appended with newly constructed SMB2_COMPRESSION_PAYLOAD_HEADER. AlgorithmId MUST be set to Pattern_V1. Length MUST be set to the size of SMB2_COMPRESSION_PATTERN_PAYLOAD_V1. CompressedMessage MUST be appended with BackDataPattern returned by section 3.1.4.4.1. RemainingUncompressedDataSize MUST be decremented by BackDataPattern.Repetitions. TotalCompressedDataSize MUST be incremented by BackDataPattern.Repetitions.</li> </ul> <p>3. If TotalCompressedDataSize+8 is less than the size of uncompressed SMB2 message, the sender MUST prepend CompressedMessage with first 8 bytes of SMB2_COMPRESSION_TRANSFORM_HEADER. OriginalCompressedSegmentSize MUST be set to the size of uncompressed SMB2 message.</p> <p>4. Otherwise, the uncompressed SMB2 message is sent.</p> <p>Changed to:</p> <ol style="list-style-type: none"> <li>1. The sender MUST initialize RemainingUncompressedDataSize with the size of uncompressed SMB2 message, TotalCompressedDataSize with 0, and CompressedMessage with empty buffer.</li> <li>2. If RemainingUncompressedDataSize is greater than zero, the message MUST be compressed as follows: <ul style="list-style-type: none"> <li>• If Connection.CompressionIds includes Pattern_V1, message MUST be scanned for data patterns as specified in section 3.1.4.4.1. If the returned FrontDataPattern.Repetitions is greater than zero, CompressedMessage MUST be appended with newly constructed SMB2_COMPRESSION_PAYLOAD_HEADER. AlgorithmId MUST be set to Pattern_V1. Length</li> </ul> </li> </ol>

Errata Published*	Description
	<p>MUST be set to the size of SMB2_COMPRESSION_PATTERN_PAYLOAD_V1. CompressedMessage MUST be appended with front data pattern returned by section 3.1.4.4.1. RemainingUncompressedDataSize MUST be decremented by FrontDataPattern.Repetitions. TotalCompressedDataSize MUST be incremented by FrontDataPattern.Repetitions.</p> <ul style="list-style-type: none"> <li>• If RemainingUncompressedDataSize is greater than 1024, CompressedMessage MUST be appended with newly constructed SMB2_COMPRESSION_PAYLOAD_HEADER. AlgorithmId MUST be set to first preferred algorithm in Connection.CompressionIds. The data MUST be compressed using the algorithm specified in AlgorithmId as specified in [MS-XCA] section 2. Length MUST be set to the size of the compressed data. A 4-byte field, indicating the size of the original plain text size of the data compressed, MUST be appended to CompressedMessage. CompressedMessage MUST be appended with the compressed data. RemainingUncompressedDataSize MUST be decremented by the size of data before compression. TotalCompressedDataSize MUST be incremented by the size of compressed data.</li> <li>• Otherwise if RemainingUncompressedDataSize is greater than zero, if (size of the uncompressed SMB2 message / RemainingUncompressedDataSize) is greater than 2, CompressedMessage MUST be appended with newly constructed SMB2_COMPRESSION_PAYLOAD_HEADER. AlgorithmId MUST be set to NONE. Length MUST be set to RemainingUncompressedDataSize. CompressedMessage MUST be appended with the Remaining uncompressed data. RemainingUncompressedDataSize MUST be decremented by the size of data before compression. TotalCompressedDataSize MUST be incremented by the size of compressed data.</li> <li>• If BackDataPattern.Repetitions is greater than zero, CompressedMessage MUST be appended with newly constructed SMB2_COMPRESSION_PAYLOAD_HEADER. AlgorithmId MUST be set to Pattern_V1. Length MUST be set to the size of SMB2_COMPRESSION_PATTERN_PAYLOAD_V1. CompressedMessage MUST be appended with BackDataPattern returned by section 3.1.4.4.1. RemainingUncompressedDataSize MUST be decremented by BackDataPattern.Repetitions. TotalCompressedDataSize MUST be incremented by BackDataPattern.Repetitions.</li> <li>• If RemainingUncompressedDataSize is greater than zero, the sender MUST repeat step 2.</li> </ul> <p>3. If TotalCompressedDataSize+8 is less than the size of uncompressed SMB2 message, the sender MUST prepend CompressedMessage with first 8 bytes of SMB2_COMPRESSION_TRANSFORM_HEADER. OriginalCompressedSegmentSize MUST be set to the size of uncompressed SMB2 message. Otherwise, the uncompressed SMB2 message is sent.</p> <p>For a Diff of the above changes, see the PDF doc <a href="#">here</a>.</p>
2020/05/25	<p>In Section 2.2.42.1, SMB2_COMPRESSION_PAYLOAD_HEADER, the following was changed from:</p> <p>The SMB2_COMPRESSION_PAYLOAD_HEADER is used by the client or server when sending chained compressed payloads. This structure MUST start at an 8-byte aligned boundary relative to the start of the message. This optional structure is only valid for the SMB 3.1.1 dialect&lt;70&gt;.</p> <p>Changed to:</p> <p>The SMB2_COMPRESSION_PAYLOAD_HEADER is used by the client or server when sending chained compressed payloads. This optional structure is only valid for the SMB 3.1.1 dialect&lt;70&gt;.</p> <p>For a Diff of the above changes, see the PDF doc <a href="#">here</a>.</p>

Errata Published*	Description
2020/05/25	<p>In Section 2.2.42.1, SMB2_COMPRESSION_PAYLOAD_HEADER, a new field was added:</p> <p>OriginalPayloadSize (4 bytes): This optional field is present only when AlgorithmId is LZNT1, LZ77, or LZ77+Huffman. The size, in bytes, of the uncompressed payload.</p> <p>Section 3.1.4.4 Compressing the Message was changed from:</p> <p>If IsCompressionSupported is FALSE or Connection.CompressionIds is empty, the sender MUST skip the processing in this section.</p> <p>If Connection.SupportsChainedCompression is TRUE, the sender &lt;77&gt;MUST prepare the compressed message2.2.42 as the following:</p> <ol style="list-style-type: none"> <li>1. The sender MUST initialize RemainingUncompressedDataSize with the size of uncompressed SMB2 message, TotalCompressedDataSize with 0, and CompressedMessage with empty buffer.</li> <li>2. If RemainingUncompressedDataSize is greater than zero, the message MUST be compressed as follows: <ul style="list-style-type: none"> <li>• If Connection.CompressionIds includes Pattern_V1, message MUST be scanned for data patterns as specified in section 3.1.4.4.1. If the returned FrontDataPattern.Repetitions is greater than zero, CompressedMessage MUST be appended with newly constructed SMB2_COMPRESSION_PAYLOAD_HEADER. AlgorithmId MUST be set to Pattern_V1. Length MUST be set to the size of SMB2_COMPRESSION_PATTERN_PAYLOAD_V1. CompressedMessage MUST be appended with front data pattern returned by section 3.1.4.4.1. RemainingUncompressedDataSize MUST be decremented by FrontDataPattern.Repetitions. TotalCompressedDataSize MUST be incremented by FrontDataPattern.Repetitions.</li> <li>• If RemainingUncompressedDataSize is greater than 1024, CompressedMessage MUST be appended with newly constructed SMB2_COMPRESSION_PAYLOAD_HEADER. AlgorithmId MUST be set to first preferred algorithm in Connection.CompressionIds. The data MUST be compressed using the algorithm specified in AlgorithmId as specified in section 2. Length MUST be set to the size of the compressed data. A 4-byte field, indicating the size of the original plain text size of the data compressed, MUST be appended to CompressedMessage. CompressedMessage MUST be appended with the compressed data. RemainingUncompressedDataSize MUST be decremented by the size of data before compression. TotalCompressedDataSize MUST be incremented by the size of compressed data.</li> <li>• Otherwise if RemainingUncompressedDataSize is greater than zero, if (size of the uncompressed SMB2 message / RemainingUncompressedDataSize) is greater than 2, CompressedMessage MUST be appended with newly constructed SMB2_COMPRESSION_PAYLOAD_HEADER. AlgorithmId MUST be set to NONE. Length MUST be set to RemainingUncompressedDataSize. CompressedMessage MUST be appended with the Remaining uncompressed data. RemainingUncompressedDataSize MUST be decremented by the size of data before compression. TotalCompressedDataSize MUST be incremented by the size of compressed data.</li> <li>• If BackDataPattern.Repetitions is greater than zero, CompressedMessage MUST be appended with newly constructed SMB2_COMPRESSION_PAYLOAD_HEADER. AlgorithmId MUST be set to Pattern_V1. Length MUST be set to the size of SMB2_COMPRESSION_PATTERN_PAYLOAD_V1. CompressedMessage MUST be appended with BackDataPattern returned by section 3.1.4.4.1. RemainingUncompressedDataSize MUST be decremented by BackDataPattern.Repetitions. TotalCompressedDataSize MUST be incremented by BackDataPattern.Repetitions.</li> </ul> </li> </ol>

Errata Published*	Description
	<ul style="list-style-type: none"> <li>• If RemainingUncompressedDataSize is greater than zero, the sender MUST repeat step 2.</li> </ul> <p>3. If TotalCompressedDataSize+8 is less than the size of uncompressed SMB2 message, the sender MUST prepend CompressedMessage with first 8 bytes of SMB2 COMPRESSION_TRANSFORM_HEADER. OriginalCompressedSegmentSize MUST be set to the size of uncompressed SMB2 message. Otherwise, the uncompressed SMB2 message is sent.</p> <p>Otherwise, the sender SHOULD construct the SMB2 COMPRESSION_TRANSFORM_HEADER specified in section as follows:</p> <ol style="list-style-type: none"> <li>1. CompressionAlgorithm MUST be set to one from Connection.CompressionIds.</li> <li>2. The sender MAY choose to leave the leading portion of the SMB2 message uncompressed and compressing only the trailing portion.</li> <li>3. The sender MUST perform the following: <ul style="list-style-type: none"> <li>• If the entire SMB2 message is being compressed, then set Offset to zero; otherwise, set Offset to the length, in bytes, of the uncompressed part of the message.</li> <li>• Set OriginalCompressedSegmentSize to the uncompressed length, in bytes, of the portion of the message that is being compressed.</li> </ul> </li> </ol> <p>The sender MUST compress the data using the CompressionAlgorithm as specified in [MS-XCA] section 2.</p> <p>If the size of the compressed data is less than OriginalCompressedSegmentSize, the sender MUST perform the following:</p> <ul style="list-style-type: none"> <li>• If Offset is zero, the sender MUST replace the SMB2 message with the SMB2 COMPRESSION_TRANSFORM_HEADER followed by the compressed SMB2 message. Otherwise, the sender MUST replace the portion of the SMB2 message selected for compression with the compressed part and prepend the SMB2 message with the SMB2 COMPRESSION_TRANSFORM_HEADER.</li> </ul> <p>Otherwise, the uncompressed SMB2 message without the SMB2 COMPRESSION_TRANSFORM_HEADER is used.</p> <p>Changed to:</p> <p>If IsCompressionSupported is FALSE or Connection.CompressionIds is empty, the sender MUST skip the processing in this section.</p> <p>If Connection.SupportsChainedCompression is FALSE, the sender SHOULD&lt;77&gt; construct the SMB2 COMPRESSION_TRANSFORM_HEADER specified in section 2.2.42 as follows:</p> <ol style="list-style-type: none"> <li>1. CompressionAlgorithm MUST be set to LZNT1, LZ77, or LZ77+Huffman specified in Connection.CompressionIds.</li> <li>2. The sender MAY choose to leave the leading portion of the SMB2 message uncompressed and compressing only the trailing portion.&lt;78&gt;</li> </ol>

Errata Published*	Description
	<p>3. The sender MUST perform the following:</p> <ul style="list-style-type: none"> <li>• If the entire SMB2 message is being compressed, then set Offset to zero; otherwise, set Offset to the length, in bytes, of the uncompressed part of the message.</li> <li>• Set OriginalCompressedSegmentSize to the uncompressed length, in bytes, of the portion of the message that is being compressed.</li> </ul> <p>4. The sender MUST compress the data using the CompressionAlgorithm as specified in [MS-XCA] section 2.</p> <p>5. If the size of the compressed data is less than OriginalCompressedSegmentSize, the sender MUST perform the following:</p> <ul style="list-style-type: none"> <li>• If Offset is zero, the sender MUST replace the SMB2 message with the SMB2 COMPRESSION_TRANSFORM_HEADER followed by the compressed SMB2 message.</li> <li>• Otherwise, the sender MUST replace the portion of the SMB2 message selected for compression with the compressed part and prepend the SMB2 message with the SMB2 COMPRESSION_TRANSFORM_HEADER. The compressed SMB2 message is sent.</li> </ul> <p>6. Otherwise, the original, uncompressed SMB2 message without the SMB2 COMPRESSION_TRANSFORM_HEADER is sent.</p> <p>Otherwise, the sender MUST prepare the compressed message as follows:</p> <ol style="list-style-type: none"> <li>1. The sender MUST initialize RemainingUncompressedDataSize with the size of uncompressed data, TotalCompressedDataSize with 0, and CompressedMessage with empty buffer.</li> <li>2. The uncompressed data MUST be compressed as follows: <ul style="list-style-type: none"> <li>• If Connection.CompressionIds includes Pattern_V1 and RemainingUncompressedDataSize is greater than 32, the uncompressed data MUST be scanned for data patterns as specified in section 3.1.4.4.1. If the returned ForwardDataPattern.Repetitions is greater than zero, CompressedMessage MUST be appended with newly constructed SMB2_COMPRESSION_PAYLOAD_HEADER. AlgorithmId MUST be set to Pattern_V1. Length MUST be set to the size of SMB2_COMPRESSION_PATTERN_PAYLOAD_V1. CompressedMessage MUST be appended with ForwardDataPattern. RemainingUncompressedDataSize MUST be decremented by ForwardDataPattern.Repetitions. If the returned BackwardDataPattern is not NULL and BackwardDataPattern.Repetitions is greater than zero, RemainingUncompressedDataSize MUST be decremented by BackwardDataPattern.Repetitions. TotalCompressedDataSize MUST be incremented by ForwardDataPattern.Repetitions.</li> <li>• If RemainingUncompressedDataSize is greater than 1024, CompressedMessage MUST be appended with newly constructed SMB2_COMPRESSION_PAYLOAD_HEADER. AlgorithmId MUST be set to LZNT1, LZ77, or LZ77+Huffman specified in Connection.CompressionIds. The uncompressed data MUST be compressed using the algorithm specified in AlgorithmId as specified in [MS-XCA] section 2. Length MUST be set to the size of the compressed data. OriginalPayloadSize MUST be set to the size of the uncompressed data. CompressedMessage MUST be appended with the compressed data. RemainingUncompressedDataSize MUST be decremented by the size of data before compression. TotalCompressedDataSize MUST be incremented by the size of compressed data.</li> <li>• Otherwise if RemainingUncompressedDataSize is greater than zero and (size of the uncompressed SMB2 message / RemainingUncompressedDataSize) is greater than 2, CompressedMessage MUST be appended with newly constructed</li> </ul> </li> </ol>

Errata Published*	Description
	<p>SMB2_COMPRESSION_PAYLOAD_HEADER. AlgorithmId MUST be set to NONE. Length MUST be set to RemainingUncompressedDataSize. CompressedMessage MUST be appended with the uncompressed data. RemainingUncompressedDataSize MUST be decremented by the size of data before compression. TotalCompressedDataSize MUST be incremented by the size of compressed data.</p> <ul style="list-style-type: none"> <li>• If BackwardDataPattern is not NULL and BackwardDataPattern.Repetitions is greater than zero, CompressedMessage MUST be appended with newly constructed SMB2_COMPRESSION_PAYLOAD_HEADER. AlgorithmId MUST be set to Pattern_V1. Length MUST be set to the size of SMB2_COMPRESSION_PATTERN_PAYLOAD_V1. CompressedMessage MUST be appended with BackwardDataPattern. TotalCompressedDataSize MUST be incremented by BackwardDataPattern.Repetitions.</li> <li>• If RemainingUncompressedDataSize is greater than zero, the sender MUST repeat step 2.</li> </ul> <p>3. If TotalCompressedDataSize+8 is less than the size of uncompressed SMB2 message, the sender MUST prepend CompressedMessage with first 8 bytes of SMB2 COMPRESSION_TRANSFORM_HEADER. OriginalCompressedSegmentSize MUST be set to the size of uncompressed SMB2 message. The compressed SMB2 message is sent. Otherwise, the original, uncompressed SMB2 message is sent.</p> <p>Section 3.1.4.4.1, Algorithm for Scanning Data Patterns V1, was changed from:</p> <p>The inputs for this algorithm are:</p> <p>InputBuffer: Input data to scan data patterns</p> <p>InputBufferSize: Size of InputBuffer</p> <p>FrontScan: A Boolean value indicating if data is to be scanned forward (TRUE) or backward (FALSE).</p> <p>The output is two DataPatterns of type SMB2_COMPRESSION_PATTERN_PAYLOAD_V1 2.2.42.2.</p> <p>If the length of InputBuffer is less than or equal to 32, no pattern compression processing is performed.</p> <p>Scan for data patterns by setting FrontScan to TRUE as specified in section . Returned DataPattern MUST be interpreted as FrontDataPattern. FrontDataPattern.Pattern MUST be set to the first byte in InputBuffer.</p> <p>If FrontDataPattern.Repetitions is equal to InputBufferSize, return FrontDataPattern. Otherwise, scan for data patterns by setting FrontScan to FALSE as specified in section 3.1.4.4.1.1. Returned DataPattern MUST be interpreted as BackDataPattern. BackDataPattern.Pattern MUST be set to the last byte in InputBuffer.</p> <p>Return FrontDataPattern and BackDataPattern.</p> <p>Changed to:</p> <p>Construct a new SMB2_COMPRESSION_PATTERN_PAYLOAD_V1 structure, specified in section 2.2.42.2 and scan forward in the buffer for a consecutive series of bytes equal to the first byte:</p>

Errata Published*	Description
	<ul style="list-style-type: none"> <li>• For each consecutive byte matched, SMB2_COMPRESSION_PATTERN_PAYLOAD_V1.Repetitions MUST be incremented by 1.</li> <li>• If none, stop scan.</li> </ul> <p>If SMB2_COMPRESSION_PATTERN_PAYLOAD_V1.Repetitions is less than 64, SMB2_COMPRESSION_PATTERN_PAYLOAD_V1.Repetitions MUST be set to 0. SMB2_COMPRESSION_PATTERN_PAYLOAD_V1.Pattern MUST be set to the first byte in the buffer.</p> <p>If SMB2_COMPRESSION_PATTERN_PAYLOAD_V1.Repetitions is equal to the size of the buffer, the processing MUST return the SMB2_COMPRESSION_PATTERN_PAYLOAD_V1 as ForwardDataPattern, and BackwardDataPattern set to NULL.</p> <p>Otherwise, construct a new SMB2_COMPRESSION_PATTERN_PAYLOAD_V1 structure, specified in section 2.2.42.2 and scan backward in the buffer for a consecutive series of bytes equal to the last byte:</p> <ul style="list-style-type: none"> <li>• For each consecutive byte matched, SMB2_COMPRESSION_PATTERN_PAYLOAD_V1.Repetitions MUST be incremented by 1.</li> <li>• If none, stop scan.</li> </ul> <p>If SMB2_COMPRESSION_PATTERN_PAYLOAD_V1.Repetitions is less than 64, SMB2_COMPRESSION_PATTERN_PAYLOAD_V1.Repetitions MUST be set to 0. SMB2_COMPRESSION_PATTERN_PAYLOAD_V1.Pattern MUST be set to the last byte in the buffer.</p> <p>The processing MUST return both SMB2_COMPRESSION_PATTERN_PAYLOAD_V1 structures respectively as ForwardDataPattern and BackwardDataPattern.</p> <p>Section 3.1.4.4.1.1, Scan for Data Patterns, was removed.</p> <p>In Section 3.1.5.3, Decompressing the Chained Message, the following was changed from:</p> <ol style="list-style-type: none"> <li>1. The receiver MUST initialize RemainingCompressedDataSize with the size of the received compressed SMB2 message and DecompressedMessage with empty buffer.</li> <li>2. If the size of RemainingCompressedDataSize is greater than the size of SMB2_COMPRESSION_PAYLOAD_HEADER, the compressed message MUST be decompressed as follows:       <p style="text-align: center;">...</p> <ul style="list-style-type: none"> <li>• Otherwise, the data of size specified in Length field MUST be decompressed using the algorithm specified in AlgorithmId field as specified in [MS-XCA] section 2. DecompressedMessage MUST be appended with the decompressed data.</li> </ul> </li> </ol> <p>Changed to:</p>

Errata Published*	Description
	<p>1. The receiver MUST initialize RemainingCompressedDataSize with the size of the received compressed data and DecompressedMessage with empty buffer.</p> <p>2. The compressed data MUST be decompressed as follows:</p> <p>...</p> <ul style="list-style-type: none"> <li>Otherwise, the data of size specified in Length field MUST be decompressed using the algorithm specified in AlgorithmId field as specified in [MS-XCA] section 2. If the size of the decompressed data is not equal to OriginalPayloadSize, the connection MUST be disconnected as specified in section 3.2.7.1 or section 3.3.7.1. DecompressedMessage MUST be appended with the decompressed data.</li> </ul> <p>For a Diff of the above changes, see the PDF doc <a href="#">here</a>.</p>
2020/05/25	<p>In Section 6, Appendix A: Product Behavior, product behavior note &lt;235&gt; was changed from:</p> <p>&lt;235&gt; Section 3.3.5.4: Windows 10 v1903 and later and Windows Server v1903 and later only set CompressionAlgorithms to the first common algorithm supported by the client and server.</p> <p>Changed to:</p> <p>&lt;235&gt; Section 3.3.5.4: Windows 10 v1903, Windows 10 v1909, Windows Server v1903, and Windows Server v1909 only set CompressionAlgorithms to the first common algorithm supported by the client and server.</p> <p>Windows 10 v2004 and Windows Server v2004 select a common pattern scanning algorithm and the first common compression algorithm, specified in section 2.2.3.1.3, supported by the client and server.</p> <p>For a Diff of the above changes, see the PDF doc <a href="#">here</a>.</p>
2020/05/11	<p>In Section 3.3.5.9.12, Handling the SMB2_CREATE_DURABLE_HANDLE_RECONNECT_V2 Create Context, the following was changed from:</p> <ul style="list-style-type: none"> <li>If the request includes the SMB2_DHANDLE_FLAG_PERSISTENT bit in the Flags field of the SMB2_CREATE_DURABLE_HANDLE_RECONNECT_V2 create context, the server MUST look up an existing Open in the GlobalOpenTable by doing a lookup with the CreateGuid of the create context. If the lookup fails, the server SHOULD&lt;292&gt; fail the request with STATUS_OBJECT_NAME_NOT_FOUND and proceed as specified in "Failed Open Handling" in section 3.3.5.9.</li> </ul> <p>Changed to:</p> <ul style="list-style-type: none"> <li>If the request includes the SMB2_DHANDLE_FLAG_PERSISTENT bit in the Flags field of the SMB2_CREATE_DURABLE_HANDLE_RECONNECT_V2 create context, TreeConnect.Share.IsCA is TRUE, and Connection.ServerCapabilities includes SMB2_GLOBAL_CAP_PERSISTENT_HANDLES, the server MUST look up an existing Open in the GlobalOpenTable by doing a lookup with the CreateGuid of the create context. If the lookup fails, the server SHOULD&lt;292&gt; fail the request with STATUS_OBJECT_NAME_NOT_FOUND and proceed as specified in "Failed Open Handling" in section 3.3.5.9.</li> </ul>
2020/05/11	<p>In Section 3.2.5.5, Receiving an SMB2 TREE_CONNECT Response, step 11 in Product Behavior Note &lt;156&gt; was changed from:</p>

Errata Published*	Description
	<p>11. The client attempts to establish an alternate channel on each selected interface and address pair. The client will create only a single connection per address pair when the server interface is neither RSS- nor RDMA-capable.</p> <p>Changed to:</p> <p>11. By default, Windows clients create four connections per RSS-capable address pair or two connections per RDMA-capable address pair or only a single connection when the address pair is neither RSS-capable nor RDMA-capable.</p>
2020/05/11	<p>In Section 3.2.5.3.3, Handling Session Binding, the following was added:</p> <p>...</p> <p>If SMB2_SESSION_FLAG_IS_GUEST bit is set in the SessionFlags field of the SMB2 SESSION_SETUP Response, the client SHOULD&lt;153&gt; return STATUS_INVALID_NETWORK_RESPONSE to the caller.</p> <p>&lt;246&gt; Section 3.3.5.6: Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2 servers will not reset ResilientOpenScavengerExpiryTime.</p> <p>...</p> <p>The client MUST ignore the SMB2_SESSION_FLAG_ENCRYPT_DATA bit in the SessionFlags field of the SMB2 SESSION_SETUP Response.</p> <p>...</p> <p>In Section 3.3.5.3, Handling GSS-API Authentication, the following was changed from:</p> <p>10. If global EncryptData is TRUE, the server MUST do the following:</p> <p>If Connection.ServerCapabilities includes SMB2_GLOBAL_CAP_ENCRYPTION or RejectUnencryptedAccess is TRUE,</p> <p>Changed to:</p> <p>10. If global EncryptData is TRUE, Connection.Dialect belongs to the SMB 3.x dialect family, Connection.ServerCapabilities includes SMB2_GLOBAL_CAP_ENCRYPTION, RejectUnencryptedAccess is TRUE, and SMB2_SESSION_FLAG_BINDING is not set in the Flags field of the request, the server MUST do the following:</p>
2020/05/11	<p>In Section 3.3.1.12, Per Lease the following was added:</p> <ul style="list-style-type: none"> <li>• Lease.FileDeleteOnClose: A Boolean, if set to TRUE, indicating that file deletion on close is pending.</li> </ul> <p>In Section 3.3.5.9.7, Handling the SMB2_CREATE_DURABLE_HANDLE_RECONNECT Create Context, step 5 was changed from:</p>

Errata Published*	Description
	<p>5. If Open.Lease is not NULL and Open.FileName does not match the file name specified in the Buffer field of the SMB2 CREATE request, the server MUST fail the request with STATUS_INVALID_PARAMETER.</p> <p>Changed to:</p> <p>5. If Open.Lease is not NULL, Open.Lease.FileDeleteOnClose is FALSE, and Open.Lease.FileName does not match the file name specified in the Buffer field of the SMB2 CREATE request, the server MUST fail the request with STATUS_INVALID_PARAMETER.</p> <p>In Section 3.3.5.9.8, Handling the SMB2_CREATE_REQUESTLEASE Create Context, the following was added:</p> <p>If both SMB2_CREATE_DURABLE_HANDLE_RECONNECT_V2 and SMB2_CREATE_REQUESTLEASE create contexts are present in the request, they are processed as specified in section 3.3.5.9.12, and this section does not apply.</p> <p>...</p> <ul style="list-style-type: none"> <li>● Lease.FileDeleteOnClose is set to FALSE.</li> </ul> <p>...</p> <p>If Open.Lease is not NULL and CreateOptions field in the CREATE request includes FILE_DELETE_ON_CLOSE, the server MUST set Open.Lease.FileDeleteOnClose to TRUE.</p> <p>The following was changed from:</p> <p>The server MUST attempt to locate a Lease by performing a lookup in the LeaseTable.LeaseList using the LeaseKey in the SMB2_CREATE_REQUESTLEASE as the lookup key. If a lease is found but Lease.Filename does not match the file name for the incoming request, the request MUST be failed with STATUS_INVALID_PARAMETER.</p> <p>Changed to:</p> <p>The server MUST attempt to locate a Lease by performing a lookup in the LeaseTable.LeaseList using the LeaseKey in the SMB2_CREATE_REQUESTLEASE as the lookup key. If a lease is found, Lease.FileDeleteOnClose is FALSE, and Lease.Filename does not match the file name for the incoming request, the request MUST be failed with STATUS_INVALID_PARAMETER.</p> <p>In Section 3.3.5.9.11, Handling the SMB2_CREATE_REQUESTLEASE_V2 Create Context, the following was added:</p> <p>If both SMB2_CREATE_DURABLE_HANDLE_RECONNECT_V2 and SMB2_CREATE_REQUESTLEASE_V2 create contexts are present in the request, they are processed as specified in section 3.3.5.9.12, and this section does not apply.</p> <p>...</p> <ul style="list-style-type: none"> <li>● Lease.FileDeleteOnClose is set to FALSE.</li> </ul>

Errata Published*	Description
	<p>...</p> <p>If Open.Lease is not NULL and CreateOptions field in the CREATE request includes FILE_DELETE_ON_CLOSE, the server MUST set Open.Lease.FileDeleteOnClose to TRUE.</p> <p>The following paragraph was changed from:</p> <p>The server MUST attempt to locate a Lease by performing a lookup in the LeaseTable.LeaseList using the LeaseKey in the SMB2_CREATE_REQUESTLEASE_V2 as the lookup key. If a lease is found but Lease.Filename does not match the file name for the incoming request, the request MUST be failed with STATUS_INVALID_PARAMETER.</p> <p>Changed to:</p> <p>The server MUST attempt to locate a Lease by performing a lookup in the LeaseTable.LeaseList using the LeaseKey in the SMB2_CREATE_REQUESTLEASE_V2 as the lookup key. If a lease is found , Lease.FileDeleteOnClose is FALSE, and Lease.Filename does not match the file name for the incoming request, the request MUST be failed with STATUS_INVALID_PARAMETER.</p> <p>In Section 3.3.5.9.12, Handling the SMB2_CREATE_DURABLE_HANDLE_RECONNECT_V2 Create Context, the following was changed from:</p> <ul style="list-style-type: none"> <li>• Open.Lease is not NULL and Open.FileName does not match the file name specified in the Buffer field of the SMB2 CREATE request.</li> </ul> <p>Changed to:</p> <ul style="list-style-type: none"> <li>• Open.Lease is not NULL, Open.Lease.FileDeleteOnClose is FALSE, and Open.Lease.FileName does not match the file name specified in the Buffer field of the SMB2 CREATE request.</li> </ul> <p>In Section 3.3.5.21.1, Handling SMB2_0_INFO_FILE, the following was changed from:</p> <p>If the object store supports security and the information class is FileBasicInformation or FilePipeInformation, and Open.GrantedAccess does not include FILE_WRITE_ATTRIBUTES, the server MUST fail the request with STATUS_ACCESS_DENIED.</p> <p>If the object store supports security and the information class is FileRenameInformation, FileDispositionInformation, or FileShortNameInformation, and Open.GrantedAccess does not include DELETE, the server MUST fail the request with STATUS_ACCESS_DENIED.</p> <p>If the object store supports security and the information class is FileFullEaInformation, and Open.GrantedAccess does not include FILE_WRITE_EA, the server MUST fail the request with STATUS_ACCESS_DENIED.</p> <p>If the object store supports security and the information class is FileFullEaInformation and the EA buffer in the Buffer field is not in a valid format, the server MUST fail the request with STATUS_EA_LIST_INCONSISTENT.</p> <p>If the object store supports security and the information class is FileAllocationInformation, FileEndOfFileInformation, or FileValidDataLengthInformation, and Open.GrantedAccess does not include FILE_WRITE_DATA, the server MUST fail the request with STATUS_ACCESS_DENIED.</p>

Errata Published*	Description
	<p>...</p> <p>Otherwise, the server MUST initialize an SMB2 SET_INFO Response following the syntax given in section 2.2.40.</p> <p>If the underlying object store returns successfully, the information class is FileRenameInformation, Connection.Dialect is "2.1" or belongs to the SMB 3.x dialect family, the server supports leasing, and Open.Lease is not NULL, the server MUST update Open.Lease.Filename to the new name for the file.</p> <p>Changed to:</p> <p>If the object store supports security and FileInfoClass is FileBasicInformation or FilePipeInformation, and Open.GrantedAccess does not include FILE_WRITE_ATTRIBUTES, the server MUST fail the request with STATUS_ACCESS_DENIED.</p> <p>If the object store supports security and FileInfoClass is FileRenameInformation, FileDispositionInformation, or FileShortNameInformation, and Open.GrantedAccess does not include DELETE, the server MUST fail the request with STATUS_ACCESS_DENIED.</p> <p>If the object store supports security and FileInfoClass is FileFullEaInformation, and Open.GrantedAccess does not include FILE_WRITE_EA, the server MUST fail the request with STATUS_ACCESS_DENIED.</p> <p>If the object store supports security and FileInfoClass is FileFullEaInformation and the EA buffer in the Buffer field is not in a valid format, the server MUST fail the request with STATUS_EA_LIST_INCONSISTENT.</p> <p>If the object store supports security and FileInfoClass is FileAllocationInformation, FileEndOfFileInformation, or FileValidDataLengthInformation, and Open.GrantedAccess does not include FILE_WRITE_DATA, the server MUST fail the request with STATUS_ACCESS_DENIED.</p> <p>...</p> <p>If the underlying object store returns successfully, FileInfoClass is FileDispositionInformation, Connection.Dialect is not "2.0.2", and Open.Lease is not NULL, the server MUST set Open.Lease.FileDeleteOnClose to TRUE.</p> <p>If the underlying object store returns successfully, FileInfoClass is FileRenameInformation, Connection.Dialect is not "2.0.2", and Open.Lease is not NULL, the server MUST update Open.Lease.Filename to the new name for the file and Open.Lease.FileDeleteOnClose to FALSE.</p>
2020/05/11	<p>In Section 3.2.3, Initialization, the following was added:</p> <p>If the client implements the SMB 2.1 dialect or SMB 3.x dialect family:</p> <p>The following was changed from:</p> <p>ClientGuid: If implemented, MUST be set to a newly generated GUID.</p> <p>Changed to:</p>

Errata Published*	Description
	<p>ClientGuid: MUST be set to a newly generated GUID.</p> <p>In Section 3.2.4.2.2.2, SMB2-Only Negotiate, the following was changed from:</p> <ul style="list-style-type: none"> <li>• If the client implements the SMB 2.1 or SMB 3.x dialect, ClientGuid SHOULD be set to the Guid provided by the application. Otherwise, it MUST be set to 0. The client MUST set Connection.ClientGuid to the ClientGuid initialized above.</li> </ul> <p>&lt;106&gt; Section 3.2.4.2.2.2: Windows 7 without [MSKB-3002286] sets ClientGuid to the global ClientGuid value.</p> <p>Changed to:</p> <ul style="list-style-type: none"> <li>• If the client implements the SMB 2.1 or SMB 3.x dialect, ClientGuid MUST be set to the global ClientGuid value. Otherwise, it MUST be set to 0. The client MUST set Connection.ClientGuid to the ClientGuid initialized above.</li> </ul> <p>In Section 3.3.1.5, Global, the following was added:</p> <ul style="list-style-type: none"> <li>• GlobalClientTable: A list of clients, indexed by the ClientGuid as specified in section 3.3.1.16.</li> </ul> <p>A new Section 3.3.1.16, Per Client, was added:</p> <p>If the server implements the SMB 3.x dialect family, it implements the following:</p> <p>Client.ClientGuid: An identifier of the client machine.</p> <p>Client.Dialect: The dialect of SMB2 negotiated with the client. This value MUST be either "2.0.2", "2.1", "3.0", "3.0.2", or "3.1.1".</p> <p>In Section 3.3.3, Initialization, the following was added:</p> <ul style="list-style-type: none"> <li>• GlobalClientTable MUST be set to an empty list.</li> </ul> <p>In Section 3.3.5.5.3, Handling GSS-API Authentication, the following was changed from:</p> <p>The server MUST look up all existing connections from the client in the global ConnectionList where Connection.ClientGuid matches Session.Connection.ClientGuid. For any matching Connection, if Connection.Dialect is not the same as Session.Connection.Dialect, the server SHOULD close the newly created Session, as specified in section 3.3.4.12, by providing Session.SessionGlobalId as the input parameter, and fail the session setup request with STATUS_USER_SESSION_DELETED.</p> <p>&lt;243&gt; Section 3.3.5.5.3: Windows Vista SP1, Windows Server 2008, Windows 7, and Windows Server 2008 R2 servers do not fail the request if dialects do not match.</p> <p>Changed to:</p>

Errata Published*	Description
	<p>If the server implements the SMB 3.x dialect family and Session.Connection.Dialect is not "2.0.2", the server MUST look up a client entry in GlobalClientTable using Session.Connection.ClientGuid. If no entry is found, the server MUST create a new Client entry by setting Client.ClientGuid to Session.Connection.ClientGuid and Client.Dialect to Session.Connection.Dialect. The server MUST insert the Client entry into GlobalClientTable. If an entry is found and Client.Dialect is not equal to Session.Connection.Dialect, the server MUST close the newly created Session, as specified in section 3.3.4.12, by providing Session.SessionGlobalId as the input parameter, and fail the session setup request with STATUS_USER_SESSION_DELETED.</p> <p>In Section 3.3.7.1, Handling Loss of a Connection, the following was added:</p> <p>If the server implements the SMB 3.x dialect family, the server MUST enumerate all connections in ConnectionList using the removed Connection.ClientGuid where Connection.Dialect is not "2.0.2". If no Connection entry is found, the server MAY remove the Client entry identified by Connection.ClientGuid from GlobalClientTable.</p>
2020/04/13	<p>In Section 3.3.5.6, Receiving an SMB2 LOGOFF Request, the following was changed from:</p> <p>When the server receives a request with an SMB2 header with a Command value equal to SMB2 LOGOFF, message handling MUST proceed as follows.</p> <p>The server MUST locate the session being logged off, as specified in section 3.3.5.2.9.</p> <p>The server MUST remove this session from the GlobalSessionTable and also from the Connection.SessionTable, and deregister the session by invoking the &lt;247&gt;event specified in [MS-SRVS] section 3.1.6.3, providing Session.SessionGlobalId as input parameter. ServerStatistics.sts0_sopens MUST be decreased by 1. The server MUST close every Open in Session.OpenTable of the old session, where Open.IsDurable is FALSE and Open.IsResilient is FALSE, as specified in section 3.3.4.17. For all opens in Session.OpenTable where Open.IsDurable is TRUE or Open.IsResilient is TRUE, the server MUST set Open.Session, Open.Connection, and Open.TreeConnect to NULL. Any tree connects in Session.TreeConnectTable of the old session MUST be deregistered by invoking the event specified in [MS-SRVS] section 3.1.6.7, providing the tuple &lt;TreeConnect.Share.ServerName, TreeConnect.Share.Name&gt; and TreeConnect.TreeGlobalId as input parameters, and each of them MUST be freed. For each deregistered TreeConnect, TreeConnect.Share.CurrentUses MUST be decreased by 1.</p> <p>If Connection.Dialect belongs to the SMB 3.x dialect family, the server MUST remove the session from each Channel.Connection.SessionTable in Session.ChannelList. All channels in Session.ChannelList MUST be removed and freed.</p> <p>The server MUST construct an SMB2 LOGOFF Response with a status code of STATUS_SUCCESS, following the syntax specified in section 2.2.8, and send it to the client. The session itself is then freed.</p> <p>Changed to:</p> <p>When the server receives a request with an SMB2 header with a Command value equal to SMB2 LOGOFF, message handling MUST proceed as follows.</p> <p>The server MUST locate the session being logged off, as specified in section 3.3.5.2.9.</p> <p>For each Open in Session.OpenTable, the server MUST perform the following:</p>

Errata Published*	Description
	<ul style="list-style-type: none"> <li>• If Open.IsResilient is TRUE, the server MUST do the following:           <ul style="list-style-type: none"> <li>• The server MUST set Open.Session, Open.Connection, and Open.TreeConnect to NULL.</li> <li>• The server MUST set Open.ResilientOpenTimeout to the current time plus Open.ResiliencyTimeOut.</li> <li>• The server SHOULD&lt;247&gt; start or reset the Resilient Open Scavenger Timer, as specified in section 3.3.2.4, under the following conditions:               <ul style="list-style-type: none"> <li>• If the Resilient Open Scavenger Timer is not already active.</li> <li>• If the Resilient Open Scavenger Timer is active and ResilientOpenScavengerExpiryTime is greater than Open.ResilientOpenTimeOut.</li> </ul> </li> </ul> </li> </ul> <p>In both of the preceding cases, the server MUST set the timer to expire at Open.ResilientOpenTimeOut and MUST set ResilientOpenScavengerExpiryTime to Open.ResilientOpenTimeOut.</p> <ul style="list-style-type: none"> <li>• If Open.IsDurable is TRUE, the server MUST do the following:       <ul style="list-style-type: none"> <li>• The server MUST set Open.Session, Open.Connection, and Open.TreeConnect to NULL.</li> <li>• The server MUST set Open.DurableOpenScavengerTimeOut to the current time plus Open.DurableOpenTimeOut.</li> <li>• The server MUST start the Durable Open Scavenger Timer, as specified in section 3.3.2.2.</li> <li>• Otherwise the server MUST close the Open as specified in section 3.3.4.17.</li> </ul> </li> </ul> <p>Any tree connects in Session.TreeConnectTable of the old session MUST be deregistered by invoking the event specified in [MS-SRVS] section 3.1.6.7, providing the tuple &lt;TreeConnect.Share.ServerName, TreeConnect.Share.Name&gt; and TreeConnect.TreeGlobalId as input parameters, and each of them MUST be freed. For each deregistered TreeConnect, TreeConnect.Share.CurrentUses MUST be decreased by 1.</p> <p>If Connection.Dialect belongs to the SMB 3.x dialect family, the server MUST remove the session from each Channel.Connection.SessionTable in Session.ChannelList. All channels in Session.ChannelList MUST be removed and freed.</p> <p>The server MUST remove this session from the GlobalSessionTable and also from the Connection.SessionTable, and deregister the session by invoking the event specified in [MS-SRVS] section 3.1.6.3, providing Session.SessionGlobalId as input parameter. ServerStatistics.sts0_sopens MUST be decreased by 1.</p> <p>The server MUST construct an SMB2 LOGOFF Response with a status code of STATUS_SUCCESS, following the syntax specified in section 2.2.8, and send it to the client. The session itself is then freed.</p>

Errata Published*	Description
	<p>&lt;247&gt; Section 3.3.5.6: Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2 servers will not reset ResilientOpenScavengerExpiryTime.</p>
2020/04/13	<p>In Section 3.2.5.14.11, Handling a Network Interfaces Response, the following was changed from:</p> <p>The client MUST extract IPv4Address and IPv6Address addresses from each NETWORK_INTERFACE_INFO structure and MUST set Connection.Server.AddressList to the received values.</p> <p>The client MUST return the list of network interfaces received from the server to the calling application.</p> <p>Changed to:</p> <p>If the Status field of the SMB2 header of the response indicates an error, the client MUST return the received status code to the calling application.</p> <p>If the Status field of the SMB2 header of the response indicates success, the client MUST extract IPv4Address and IPv6Address addresses from each NETWORK_INTERFACE_INFO structure and MUST set Connection.Server.AddressList to the received values.</p> <p>In Section 3.3.5.15.11, Handling a Query Network Interface Request, the following was changed from:</p> <p>The server MUST enumerate the local network interfaces in an implementation-specific manner. For each IP address in each network interface, the server MUST construct a NETWORK_INTERFACE_INFO structure as specified in section 2.2.32.5, with the following values:</p> <p>Changed to:</p> <p>This section applies only to servers that implement the SMB 3.x dialect family.</p> <p>When the server receives a request with an SMB2 header with a Command value equal to SMB2 IOCTL and a CtlCode of FSCTL_QUERY_NETWORK_INTERFACE_INFO, message handling proceeds as follows:</p> <p>If Connection.Dialect does not belong to the SMB 3.x dialect family or Connection.ServerCapabilities does not include SMB2_GLOBAL_CAP_MULTI_CHANNEL, the server MAY fail the request with STATUS_NOT_SUPPORTED.</p> <p>Otherwise, the server MUST enumerate the local network interfaces in an implementation-specific manner. For each IP address in each network interface, the server MUST construct a NETWORK_INTERFACE_INFO structure as specified in section 2.2.32.5, with the following values:</p>

\*Date format: YYYY/MM/DD