

## [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 [V54.0 – 2017/12/01](#).

Errata Published*	Description												
2018/02/26	<p>In Section 2.2.9.2, SMB2 TREE_CONNECT_CONTEXT Request Values, the valid values for ContextType have been changed from:</p> <table border="1" data-bbox="506 709 1414 1024"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>SMB2_RESERVED_TREE_CONNECT_CONTEXT_ID 0x00000000</td> <td>This value is reserved.</td> </tr> <tr> <td>SMB2_REMOTED_IDENTITY_TREE_CONNECT_CONTEXT_ID 0x00000001</td> <td>The Data field contains remotied identity tree connect context data as specified in section 2.2.9.2.1.</td> </tr> </tbody> </table> <p>Changed to:</p> <table border="1" data-bbox="506 1136 1414 1451"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>SMB2_RESERVED_TREE_CONNECT_CONTEXT_ID 0x0000</td> <td>This value is reserved.</td> </tr> <tr> <td>SMB2_REMOTED_IDENTITY_TREE_CONNECT_CONTEXT_ID 0x0001</td> <td>The Data field contains remotied identity tree connect context data as specified in section 2.2.9.2.1.</td> </tr> </tbody> </table>	Value	Meaning	SMB2_RESERVED_TREE_CONNECT_CONTEXT_ID 0x00000000	This value is reserved.	SMB2_REMOTED_IDENTITY_TREE_CONNECT_CONTEXT_ID 0x00000001	The Data field contains remotied identity tree connect context data as specified in section 2.2.9.2.1.	Value	Meaning	SMB2_RESERVED_TREE_CONNECT_CONTEXT_ID 0x0000	This value is reserved.	SMB2_REMOTED_IDENTITY_TREE_CONNECT_CONTEXT_ID 0x0001	The Data field contains remotied identity tree connect context data as specified in section 2.2.9.2.1.
Value	Meaning												
SMB2_RESERVED_TREE_CONNECT_CONTEXT_ID 0x00000000	This value is reserved.												
SMB2_REMOTED_IDENTITY_TREE_CONNECT_CONTEXT_ID 0x00000001	The Data field contains remotied identity tree connect context data as specified in section 2.2.9.2.1.												
Value	Meaning												
SMB2_RESERVED_TREE_CONNECT_CONTEXT_ID 0x0000	This value is reserved.												
SMB2_REMOTED_IDENTITY_TREE_CONNECT_CONTEXT_ID 0x0001	The Data field contains remotied identity tree connect context data as specified in section 2.2.9.2.1.												
2018/02/26	<p>In Section 3.3.5.2.9, Verifying the Session, the following has been changed from:</p> <p>If Connection.Dialect belongs to the SMB 3.x dialect family, and Session.EncryptData is TRUE, the server MUST locate the Request in Connection.RequestList for which Request.MessageId matches the MessageId value in the SMB2 header of the request. If Request.IsEncrypted is FALSE, the server MUST fail the request with STATUS_ACCESS_DENIED.</p> <p>Changed to:</p> <p>If Connection.Dialect belongs to the SMB 3.x dialect family, and Session.EncryptData is TRUE, the server MUST do the following:</p>												

Errata Published*	Description
	<p>If the server supports the 3.1.1 dialect, locate the Request in Connection.RequestList for which Request.MessageId matches the MessageId value in the SMB2 header of the request.</p> <p>Otherwise, if the server supports 3.0 or 3.0.2 dialect, and RejectUnencryptedAccess is TRUE, locate the Request in Connection.RequestList for which Request.MessageId matches the MessageId value in the SMB2 header of the request.</p> <p>If Request.IsEncrypted is FALSE, the server MUST fail the request with STATUS_ACCESS_DENIED.</p> <p>In Section 3.3.5.2.11, Verifying the Tree Connect, the following has been changed from:</p> <p>The server MUST look up the TreeConnect in Session.TreeConnectTable by using the TreeId in the SMB2 header of the request. If no tree connect is found, the request MUST be failed with STATUS_NETWORK_NAME_DELETED.</p> <p>If Connection.Dialect belongs to the SMB 3.x dialect family, the server MUST fail the request with STATUS_ACCESS_DENIED in the following cases:</p> <p>TreeConnect.Share.EncryptData is TRUE, Connection.ServerCapabilities includes SMB2_GLOBAL_CAP_ENCRYPTION, and Request.IsEncrypted is FALSE.</p> <p>EncryptData or TreeConnect.Share.EncryptData or Request.IsEncrypted is TRUE, RejectUnencryptedAccess is TRUE, and Connection.ServerCapabilities does not include SMB2_GLOBAL_CAP_ENCRYPTION.</p> <p>Changed to:</p> <p>The server MUST look up the TreeConnect in Session.TreeConnectTable by using the TreeId in the SMB2 header of the request. If no tree connect is found, the request MUST be failed with STATUS_NETWORK_NAME_DELETED.</p> <p>If Connection.Dialect belongs to the SMB 3.x dialect family, the server MUST fail the request with STATUS_ACCESS_DENIED in the following cases:</p> <p>If the server supports the 3.1.1 dialect, TreeConnect.Share.EncryptData is TRUE, Connection.ServerCapabilities includes SMB2_GLOBAL_CAP_ENCRYPTION, and Request.IsEncrypted is FALSE.</p> <p>Otherwise, if the server supports 3.0 or 3.0.2 dialect, EncryptData or TreeConnect.Share.EncryptData is TRUE, Connection.ServerCapabilities includes SMB2_GLOBAL_CAP_ENCRYPTION, RejectUnencryptedAccess is TRUE, and Request.IsEncrypted is FALSE.</p> <p>EncryptData or TreeConnect.Share.EncryptData or Request.IsEncrypted is TRUE, RejectUnencryptedAccess is TRUE, and Connection.ServerCapabilities does not include SMB2_GLOBAL_CAP_ENCRYPTION.</p>
2018/02/26	<p>In Section 3.3.1.12, Per Lease, the following has been added:</p> <ul style="list-style-type: none"> <li>Lease.Held: A Boolean, when set to TRUE, indicates that at least one Open is associated with this lease.</li> </ul>

Errata Published*	Description
	<p>In Section 3.3.4.7, Object Store Indicates a Lease Break, the following has been changed from:</p> <p>If a lease entry is found, the server MUST check the state of Open.Connection for all Opens in Lease.LeaseOpens. If Open.Session.Connection.Dialect belongs to the SMB 3.x dialect family and Open.Connection is NULL, the server MUST select an alternate connection in Open.Session.ChannelList and update Open.Connection.</p> <p>...</p> <ul style="list-style-type: none"> <li>• Otherwise, the server MUST set Open.Lease.Breaking to FALSE and MUST complete the lease break call from the underlying object store with "NONE" as the new lease state.</li> </ul> <p>Changed to:</p> <p>If a Lease entry is found, the server MUST check the state of Open.Connection for all Opens in Lease.LeaseOpens. If Open.Session.Connection.Dialect belongs to the SMB 3.x dialect family and Open.Connection is NULL, the server MUST select an alternate connection in Open.Session.ChannelList and update Open.Connection.</p> <p>...</p> <ul style="list-style-type: none"> <li>• Otherwise, the server MUST set Open.Lease.Breaking to FALSE and MUST complete the lease break call from the underlying object store with "NONE" as the new lease state.</li> </ul> <p>In Section 3.3.4.17, Server Application Requests Closing an Open, the following has been changed from:</p> <ul style="list-style-type: none"> <li>• The server MUST then remove the Open from Lease.LeaseOpens.</li> <li>• If Lease.LeaseOpens is now empty:</li> <li>• If Lease.Breaking is TRUE, the server MUST complete the lease break to the underlying object store with NONE as the new lease state. &lt;203&gt;</li> <li>• The server MUST remove the Lease from the LeaseTable.LeaseList and free the Lease.</li> </ul> <p>Changed to:</p> <ul style="list-style-type: none"> <li>• The server MUST then remove the Open from Open.Lease.LeaseOpens. If this Open is the last open in Open.Lease.LeaseOpens, the server MUST set Open.Lease.Held to FALSE.</li> <li>• If Open.Lease.Held is FALSE:</li> <li>• If Open.Lease.Breaking is TRUE, the server MUST complete the lease break to the underlying object store with NONE as the new lease state. &lt;203&gt;</li> <li>• The server MUST remove the Open.Lease from the LeaseTable.LeaseList and free the Open.Lease.</li> </ul> <p>In Section 3.3.5.9.8, Handling the SMB2_CREATE_REQUEST_LEASE Create Context, the following has been changed from:</p> <p>The server MUST set Open.OplockState to Held, set Open.Lease to a reference to lease, set Open.OplockLevel to SMB2_OPLOCK_LEVEL_LEASE, and add open to Lease.LeaseOpens. The remainder of open response construction continues as described in "Response Construction".</p> <p>Changed to:</p>

Errata Published*	Description
	<p>The server MUST set Open.OplockState to Held, set Open.Lease to a reference to Lease, set Open.OplockLevel to SMB2_OPLOCK_LEVEL_LEASE, and add Open to Lease.LeaseOpens. If this Open is the first open in Lease.LeaseOpens, the server MUST set Lease.Held to TRUE. The remainder of open response construction continues as described in "Response Construction".</p> <p>In Section 3.3.5.9.11, Handling the SMB2_CREATE_REQUEST_LEASE_V2 Create Context, the following has been changed from:</p> <p>The server MUST set Open.OplockState to Held, set Open.Lease to a reference to lease, set Open.OplockLevel to SMB2_OPLOCK_LEVEL_LEASE, and add open to Lease.LeaseOpens. The remainder of open response construction continues as described in the "Response Construction" phase.</p> <p>Changed to:</p> <p>The server MUST set Open.OplockState to Held, set Open.Lease to a reference to Lease, set Open.OplockLevel to SMB2_OPLOCK_LEVEL_LEASE, and add Open to Lease.LeaseOpens. If this Open is the first open in Lease.LeaseOpens, the server MUST set Lease.Held to TRUE. The remainder of open response construction continues as described in the "Response Construction" phase.</p>
2018/02/12	<p>In Section 2.2.1, SMB2 Packet Header, the following has been changed from:</p> <p>If the SMB2_FLAGS_ASYNC_COMMAND bit is set in Flags, the header takes the form SMB2 Packet Header – ASYNC (section 2.2.1.1). This header format is used for responses to requests processed asynchronously by the server, as specified in sections 3.3.4.2, 3.3.4.3, 3.3.4.4, and 3.2.5.1.5. This header format MAY be used for any request, and the SMB2 CANCEL Request MUST use this format for canceling requests that have received an interim response, as specified in sections 3.2.4.24 and 3.3.5.16.</p> <p>If the SMB2_FLAGS_ASYNC_COMMAND bit is not set in Flags, the header takes the form SMB2 Packet Header – SYNC (section 2.2.1.2). This format can be used for all requests and responses.</p> <p>Changed to:</p> <p>If the SMB2_FLAGS_ASYNC_COMMAND bit is set in Flags, the header takes the form SMB2 Packet Header – ASYNC (section 2.2.1.1). This header format is used for responses to requests processed asynchronously by the server, as specified in sections 3.3.4.2, 3.3.4.3, 3.3.4.4, and 3.2.5.1.5. The SMB2 CANCEL Request MUST use this format for canceling requests that have received an interim response, as specified in sections 3.2.4.24 and 3.3.5.16.</p> <p>If the SMB2_FLAGS_ASYNC_COMMAND bit is not set in Flags, the header takes the form SMB2 Packet Header – SYNC (section 2.2.1.2).</p>
2018/02/12	<p>In Section 3.2.4.6, Application Requests Reading from a File or Named Pipe, the following has been changed from:</p> <ul style="list-style-type: none"> <li>• The returned list of SMB_DIRECT_BUFFER_DESCRIPTOR_1 structures MUST be appended to the SMB2 header.</li> <li>• The ReadChannelInfoOffset MUST be set to the offset of the appended list from the beginning of the SMB2 header.</li> <li>• The ReadChannelInfoLength MUST be set to the length of the appended list.</li> </ul>

Errata Published*	Description
	<p>Changed to:</p> <ul style="list-style-type: none"> <li>• The returned list of SMB_DIRECT_BUFFER_DESCRIPTOR_V1 structures MUST be added to the Buffer field of the request.</li> <li>• The ReadChannelInfoOffset MUST be set to the offset of the added list from the beginning of the SMB2 header.</li> <li>• The ReadChannelInfoLength MUST be set to the length of the added list.</li> </ul> <p>Otherwise, the following fields of the request MUST be initialized as follows:</p> <ul style="list-style-type: none"> <li>• The Channel field MUST be set to 0.</li> <li>• The first byte of the Buffer field MUST be set to 0.</li> <li>• The ReadChannelInfoOffset field MUST be set to 0.</li> <li>• The ReadChannelInfoLength field MUST be set to 0.</li> </ul> <p>In Section 3.2.4.7, Application Requests Writing to a File or Named Pipe, the following has been changed from:</p> <ul style="list-style-type: none"> <li>• The returned list of SMB_DIRECT_BUFFER_DESCRIPTOR_1 structures MUST be appended to the SMB2 header.</li> <li>• The WriteChannelInfoOffset MUST be set to the offset of the appended list from the beginning of the SMB2 header.</li> <li>• The WriteChannelInfoLength MUST be set to the length of the appended list.</li> </ul> <p>Changed to:</p> <ul style="list-style-type: none"> <li>• The returned list of SMB_DIRECT_BUFFER_DESCRIPTOR_V1 structures MUST be added to the Buffer field of the request.</li> <li>• The WriteChannelInfoOffset MUST be set to the offset of the added list from the beginning of the SMB2 header.</li> <li>• The WriteChannelInfoLength MUST be set to the length of the added list.</li> </ul>
2018/01/16	<p>In Section 2.2.41, SMB2 TRANSFORM_HEADER, the first paragraph has been changed from:</p> <p>The SMB2 Transform Header is used by the client or server when sending encrypted messages. The SMB2 TRANSFORM_HEADER is only valid for the SMB 3.x dialect family.</p> <p>Changed to:</p> <p>The SMB2 TRANSFORM_HEADER is used by the client or server when sending encrypted messages. The SMB2 TRANSFORM_HEADER is only valid for the SMB 3.x dialect family.</p> <p>In Section 3.2.5.1.1, Decrypting the Message, the fourth bullet point has been changed from:</p> <ul style="list-style-type: none"> <li>• The client MUST decrypt the message using Session.DecryptionKey. If Connection.Dialect is "3.1.1", the algorithm specified by Connection.CipherId is used. Otherwise, the AES-128-CCM algorithm is used. The client passes in the TRANSFORM_HEADER, excluding the Signature and ProtocolId fields, and the encrypted SMB2 message as the Optional Authenticated Data input for the algorithm. If decryption succeeds, the client MUST compare the signature in the transform header with the signature returned by the decryption algorithm. If signature</li> </ul>

Errata Published*	Description
	<p>verification succeeds, the client MUST then continue processing the decrypted packet, as specified in subsequent sections. If signature verification fails, the client MUST fail the application request with an implementation-specific error.</p> <p>Changed to:</p> <ul style="list-style-type: none"> <li>The client MUST decrypt the message using Session.DecryptionKey. If Connection.Dialect is "3.1.1", the algorithm specified by Connection.CipherId is used. Otherwise, the AES-128-CCM algorithm is used. The client passes in the Nonce, OriginalMessageSize, Flags/EncryptionAlgorithm and SessionId fields of the SMB2 TRANSFORM_HEADER and the encrypted SMB2 message as the Optional Authenticated Data input for the algorithm. If decryption succeeds, the client MUST compare the signature in the SMB2 TRANSFORM_HEADER with the signature returned by the decryption algorithm. If signature verification succeeds, the client MUST then continue processing the decrypted packet, as specified in subsequent sections. If signature verification fails, the client MUST fail the application request with an implementation-specific error.</li> </ul> <p>In Section 3.3.5.2.1, Decrypting the Message, the second bullet point has been changed from:</p> <ul style="list-style-type: none"> <li>If OriginalMessageSize value received in the TRANSFORM_HEADER is greater than the implementation-specific limit&lt;210&gt; or if it is less than the size of the SMB2 Header, the server MUST disconnect the connection as specified in section 3.3.7.1.</li> </ul> <p>Changed to:</p> <ul style="list-style-type: none"> <li>If OriginalMessageSize value received in the SMB2 TRANSFORM_HEADER is greater than the implementation-specific limit&lt;210&gt; or if it is less than the size of the SMB2 Header, the server MUST disconnect the connection as specified in section 3.3.7.1.</li> </ul> <p>The fifth bullet point has been changed from:</p> <ul style="list-style-type: none"> <li>The server MUST decrypt the message using Session.DecryptionKey. If Connection.Dialect is less than "3.1.1", then AES-128-CCM MUST be used, as specified in [RFC4309]. Otherwise, the algorithm specified by the Connection.CipherId MUST be used. The server passes in the TRANSFORM_HEADER, excluding the Signature and ProtocolId fields, as the Optional Authenticated Data input for the algorithm. If decryption succeeds, the server MUST compare the signature in the transform header with the signature returned by the decryption algorithm. If the signature verification fails, the server MUST disconnect the connection as specified in section 3.3.7.1. If the signature verification succeeds, the server MUST continue processing the decrypted packet, as specified in subsequent sections.</li> </ul> <p>Changed to:</p> <ul style="list-style-type: none"> <li>The server MUST decrypt the message using Session.DecryptionKey. If Connection.Dialect is less than "3.1.1", then AES-128-CCM MUST be used, as specified in [RFC4309]. Otherwise, the algorithm specified by the Connection.CipherId MUST be used. The server passes in the Nonce, OriginalMessageSize, Flags/EncryptionAlgorithm and SessionId fields of the SMB2 TRANSFORM_HEADER as the Optional Authenticated Data input for the algorithm. If decryption succeeds, the server MUST compare the signature in the SMB2 TRANSFORM_HEADER with the signature returned by the decryption algorithm. If the signature verification fails, the server MUST disconnect the connection as specified in</li> </ul>

Errata Published*	Description
	<p>section 3.3.7.1. If the signature verification succeeds, the server MUST continue processing the decrypted packet, as specified in subsequent sections.</p>
2018/01/16	<p>In Section 2.2.2.2, ErrorData format, changed from:</p> <p>The ErrorData MUST be formatted based on the error code being returned.</p> <p>If the error code in the header of the response is set to STATUS_STOPPED_ON_SYMLINK, this field MUST contain a Symbolic Link Error Response as specified in section 2.2.2.2.1.</p> <p>If the error code in the header of the response is set to STATUS_BAD_NETWORK_NAME, and the ErrorId in the SMB2 Error Context response is set to SMB2_ERROR_ID_SHARE_REDIRECT, this field MUST contain a Share Redirect Error Response as specified in section 2.2.2.2.2.</p> <p>If the error code in the header of the response is STATUS_BUFFER_TOO_SMALL, this field MUST be set to a 4-byte value indicating the minimum required buffer length.</p> <p>Changed to:</p> <p>The ErrorData MUST be formatted based on the error code being returned in the Status field of the SMB2 Packet header for the SMB2 Error Response (section 2.2.2).</p> <p>If the Status field of the header of the response is set to STATUS_STOPPED_ON_SYMLINK, this field MUST contain a Symbolic Link Error Response as specified in section 2.2.2.2.1.</p> <p>If the Status field of the header of the response is set to STATUS_BAD_NETWORK_NAME, and the ErrorId in the SMB2 Error Context response is set to SMB2_ERROR_ID_SHARE_REDIRECT, this field MUST contain a Share Redirect Error Response as specified in section 2.2.2.2.2.</p> <p>If the Status field of the header of the response is set to STATUS_BUFFER_TOO_SMALL, this field MUST be set to a 4-byte value indicating the minimum required buffer length.</p>
2018/01/16	<p>In Section 2.2.9.2.1.4, LUID_ATTR_DATA, the size of the Luid field has been changed from 4 to 8 bytes and the description changed from:</p> <p>Luid (4 bytes): LUID is a locally unique identifier, as specified in [MS-DTYP] section 2.3.7.</p> <p>Changed to:</p> <p>Luid (8 bytes): Locally unique identifier, as specified in [MS-DTYP] section 2.3.7.</p>
2018/01/16	<p>In Section 3.3.1.12, Per Lease, the description of Lease.Breaking has been changed from:</p> <ul style="list-style-type: none"> <li>• Lease.Breaking: A Boolean that indicates if a lease break is in progress.</li> </ul> <p>Changed to:</p> <ul style="list-style-type: none"> <li>• Lease.Breaking: A Boolean, if set to TRUE, indicating a lease break requiring acknowledgement is in progress.</li> </ul> <p>In Section 3.3.4.7, Object Store Indicates a Lease Break, the following was changed from:</p> <p>If Lease.LeaseOpens is not empty, the server MUST do the following:</p>

Errata Published*	Description
	<ul style="list-style-type: none"> <li>• If Open.Connection is NULL, Open.IsPersistent is TRUE, and Open.DurableOpenTimeout is not earlier than the current time, the server MUST set Open.Lease.Breaking to FALSE, complete the lease break call from the underlying object store with "NONE" as the new lease state, and take no further action.</li> <li>• If Open.Connection is NULL, the server MUST set Open.Lease.Breaking to TRUE, and take no further action.</li> <li>• Otherwise, construct a Lease Break Notification (section 2.2.23.2) message to send to the client.</li> </ul> <p>The server MUST set the Command in the SMB2 header to SMB2_OPLOCK_BREAK, and the MessageId to 0xFFFFFFFFFFFFFFFF. The server MUST set the SessionId and TreeId in the SMB2 header to 0. If Lease.LeaseState is SMB2_LEASE_READ_CACHING, the server MUST set the Flags field of the message to zero and MUST set Open.OplockState to None for all opens in Lease.LeaseOpens. The server MUST set Lease.Breaking to FALSE, and the LeaseKey field MUST be set to Lease.LeaseKey. Otherwise the server MUST set the Flags field of the message to SMB2_NOTIFY_BREAK_LEASE_FLAG_ACK_REQUIRED, indicating to the client that lease acknowledgment is required. The LeaseKey field MUST be set to Lease.LeaseKey. The server MUST set Open.OplockState to Breaking for all Opens in Lease.LeaseOpens. The server MUST set the CurrentLeaseState field of the message to Lease.LeaseState, set Lease.Breaking to TRUE, set Lease.BreakToLeaseState to the new lease state indicated by the object store, and set Lease.LeaseBreakTimeout to the current time plus an implementation-specific default value in milliseconds.&lt;200&gt; If the server implements the SMB 3.x dialect family and Lease.Version is 2, the server MUST set NewEpoch to Lease.Epoch + 1. Otherwise, NewEpoch MUST be set to zero.</p> <p>The SMB2 Lease Break Notification is sent to the client using the connection specified in Open.Connection of the first Open in Lease.LeaseOpens. The message SHOULD NOT be signed. The server MUST start the oplock break acknowledgment timer as specified in 3.3.2.1. If there was an error in attempting to transmit the message to the client, the server MUST retry the send using the connection specified in Open.Connection of the next Open in Lease.LeaseOpens. If the server fails to send transmit the message on any Open.Connection associated with this lease, the server MUST complete the lease break call from the underlying object store with "NONE" as the new lease state.</p> <p>Changed to:</p> <p>If Lease.LeaseOpens is not empty, the server MUST construct a Lease Break Notification (section 2.2.23.2) message to send to the client.</p> <p>The server MUST set the Command field in the SMB2 header to SMB2_OPLOCK_BREAK, and the MessageId field to 0xFFFFFFFFFFFFFFFF. The server MUST set the SessionId and TreeId fields in the SMB2 header to 0.</p> <p>If Lease.LeaseState is SMB2_LEASE_READ_CACHING, the server MUST set the Flags field of the message to zero and MUST set Open.OplockState to "None" for all opens in Lease.LeaseOpens. The server MUST set Lease.Breaking to FALSE, and the LeaseKey field MUST be set to Lease.LeaseKey.</p> <p>Otherwise, the server MUST set the Flags field of the message to SMB2_NOTIFY_BREAK_LEASE_FLAG_ACK_REQUIRED, indicating to the client that lease acknowledgment is required. The LeaseKey field MUST be set to Lease.LeaseKey. The server MUST set Open.OplockState to "Breaking" for all Opens in Lease.LeaseOpens. The server MUST set the CurrentLeaseState field of the message to Lease.LeaseState, set Lease.Breaking to TRUE, set Lease.BreakToLeaseState to the new lease state indicated by the object store, and</p>

Errata Published*	Description
	<p>set Lease.LeaseBreakTimeout to the current time plus an implementation-specific&lt;200&gt; default value in milliseconds.</p> <p>If the server implements the SMB 3.x dialect family and Lease.Version is 2, the server MUST set NewEpoch to Lease.Epoch + 1. Otherwise, NewEpoch MUST be set to zero.</p> <p>The SMB2 Lease Break Notification is sent to the client using the connection specified in Open.Connection of the first Open in Lease.LeaseOpens. The message SHOULD NOT be signed. If the server fails to send the message to the client, the server MUST retry the send using the connection specified in Open.Connection of the next Open in Lease.LeaseOpens.</p> <p>If the server succeeds in sending the message on any Open.Connection associated with this Lease, the server MUST start the oplock break acknowledgment timer as specified in section 3.3.2.1.</p> <p>Otherwise, the server MUST perform the following steps:</p> <ul style="list-style-type: none"> <li>• If Open.IsPersistent is TRUE, and Lease.LeaseState is not SMB2_LEASE_READ_CACHING, and Open.DurableOpenTimeout is not earlier than the current time, the server MUST take no further action.</li> <li>• Otherwise, the server MUST set Open.Lease.Breaking to FALSE and MUST complete the lease break call from the underlying object store with "NONE" as the new lease state.</li> </ul>

\*Date format: YYYY/MM/DD