[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 V50.0 – 2016/09/26.

Errata Published*	Description
2017/02/20	In Section 2.2.24.1, Oplock Break Acknowledgment. Section 2.2.24.2, Lease Break Acknowledgment, and Section 2.2.26, SMB2 LOCK Request, the following was added to the first paragraph: This message is composed of an SMB2 header, as specified in section 2.2.1, followed by this acknowledgement structure.
2017/01/09	In Section 3.2.4.24, Application Requests Canceling an Operation, the SessionId field initialization was changed to: The SessionId field MUST be set to the session identifier that is previously used for the request being canceled. If the session identified by SessionId has Session.SigningRequired equal to TRUE, the client sets SMB2_FLAGS_SIGNED to TRUE in the Flags field. The SMB2 CANCEL Request MUST be initialized to the default values, as specified in 2.2.30. In Section 3.3.5.16, Receiving an SMB2 CANCEL Request, the following paragraph was added: If SMB2_FLAGS_SIGNED bit is set in the Flags field of the SMB2 header of the cancel request, the server MUST verify the session, as specified in section 3.3.5.2.9. In Section 6, Appendix A Product Behavior, Product Behavior Note 145 associated with the change in Section 3.2.4.24 was removed.
2017/01/09	In Section 3.2.5.1.1, Decrypting the Message, the third bullet point was changed from: The client MUST look up the session in the Connection.SessionTable using the SessionId in the SMB2 TRANSFORM_HEADER of the response. If the session is not found, the response MUST be discarded as invalid. Changed to: The client MUST look up the session in the Connection.SessionTable using the SessionId in the SMB2 TRANSFORM_HEADER of the response. If the session is not found, the response MUST be discarded. A new bullet point was added: If the NextCommand field in the first SMB2 header of the message is equal to 0 and SessionId of the first SMB2 header is not equal to the SessionId field in SMB2 TRANSFORM_HEADER of response, the client MUST discard the message.

Errata Published*	Description
	In Section 3.2.5.1.9, Handling Compounded Responses, the following additions were made. For the first response: • If SMB2_FLAGS_RELATED_OPERATIONS is set in the Flags field of the SMB2 header of response, the client SHOULD<152> discard the message. • If the SessionId field of SMB2 header is not equal to the SessionId field in SMB2 TRANSFORM_HEADER of response, the client MUST discard the message. For each subsequent response: • If SMB2_FLAGS_RELATED_OPERATIONS is not set in the Flags field of the SMB2 header of the response, the client SHOULD<153> discard the message. • If the SessionId field of SMB2 header is not equal to the SessionId field in the SMB2 TRANSFORM_HEADER of response, the client MUST discard the message. <152> Section 3.2.5.1.9: Windows 8, Windows Server 2012, Windows 8.1 and Windows Server 2012 R2 ignore this flag. <153> Section 3.2.5.1.9: Windows 8, Windows Server 2012, Windows 8.1 and Windows Server 2012 R2 discard the message if SMB2_FLAGS_RELATED_OPERATIONS is set in the Flags field of the SMB2 header of the response. In Section 3.3.5.2.7.2, Handling Compounded Related Requests, the last paragraph was changed from: When all operations are complete, the responses SHOULD be compounded into a single response to return to the client. If the responses are compounded, the server MUST set SMB2_FLAGS_RELATED_OPERATIONS in the Flags field of the SMB2 header of all responses except the first one. This indicates that the response was part of a compounded chain. Changed to: When all operations are complete, the responses SHOULD be compounded into a single response to return to the client. If the responses are compounded, the server SHOULD set SMB2_FLAGS_RELATED_OPERATIONS in the Flags field of the SMB2 header of all responses except the first one. This indicates that the response was part of a compounded chain.
2016/12/19	In Section 3.3.5.15.6, Handling a Server-Side Data Copy Request, changed the field name OutputCount to MaxOutputResponse. Changed from: … If the OutputCount value in the SMB2 IOCTL Request is less than the size of the SRV_COPYCHUNK_RESPONSE structure, the server MUST fail the SMB2 IOCTL Request with STATUS_INVALID_PARAMETER. If the OutputCount value in the SMB2 IOCTL Request is greater than or equal to the size of the SRV_COPYCHUNK_RESPONSE structure and any of the following are true, the server MUST send an SMB2 IOCTL Response as specified in section 3.3.5.15.6.2: … Changed to: …

Errata Published*	Description
	If the MaxOutputResponse value in the SMB2 IOCTL Request is less than the size of the SRV_COPYCHUNK_RESPONSE structure, the server MUST fail the SMB2 IOCTL Request with STATUS_INVALID_PARAMETER. If the MaxOutputResponse value in the SMB2 IOCTL Request is greater than or equal to the size of the SRV_COPYCHUNK_RESPONSE structure and any of the following are true, the server MUST send an SMB2 IOCTL Response as specified in section 3.3.5.15.6.2: …
2016/11/21	In two sections, modified the processing rules for change notifications. In Section 3.3.1.3, Algorithm for Change Notifications in an Object Store, changed from: … • If a change notification request is pending on a directory AND a change occurs to the directory contents matching the events to be monitored as specified in CompletionFilter, the server MUST copy the results into the buffer of the Change Notification response. The server MAY choose to aggregate one or more changes indicated by the underlying object store into a single response. The server MUST construct a SMB2 CHANGE_NOTIFY Response as specified in section 2.2.36. The server MUST then return the results to the client. • The server SHOULD try to fit in the maximum number of events that match the CompletionFilter of the request before completing the request. • If a client issues multiple change notification requests on the same open to a directory, the server MUST queue the requests and complete them on a First In, First Out (FIFO) basis when changes are indicated by the underlying object store. • If the client requested that an entire tree be watched, the server MUST monitor all objects beneath the directory on which the operation was issued, instead of simply the immediate children of that directory. • Change notification information that is returned to the user MUST conform to the syntax specified in section 2.4.42. Changed to: … • The algorithm MUST perform the change notification processing based on the CompletionFilter and SMB2_WATCH_TREE flag in the Flags field of the first CHANGE_NOTIFY request on an Open.LocalOpen. The algorithm MUST ignore the CompletionFilter and SMB2_WATCH_TREE flag in all further requests on the same open. • If the client sets the SMB2_WATCH_TREE flag in the Flags field of the first request on an Open.LocalOpen, indicating that an entire tree is being watched, the algorithm MUST monitor all objects beneath the directory on which the operation was issued, instead of simply the immediate children objects of that directory. • If a client issues multiple change notification requests on the same open to a directory, the server MUST queue the requests and complete them on a First In, First Out (FIFO) basis when changes are indicated by the underlying object store. • If a change notification request is pending on a directory and a change occurs to the directory contents matching the events to be monitored as specified by the CompletionFilter,

<table>
<tr><th>Errata Published*</th><th>Description</th></tr>
<tr><td></td><td>the server MUST copy the results into the Buffer field of the CHANGE_NOTIFY response. The server SHOULD send the maximum number of events that match the CompletionFilter of the first CHANGE_NOTIFY request indicated by the underlying object store into a single response up to the maximum of the OutputBufferLength field. The server MUST construct the response in the format specified in section 2.2.36 and the change notification information in the format specified in [MS-FSCC] section 2.4.42. The server MUST then return the results to the client.

In Section 3.3.5.19, Receiving an SMB2 CHANGE_NOTIFY Request, changed from:

Change notification processing in the object store MUST be handled as specified in section 3.3.1.3. It is also outlined in [MS-CIFS] section 3.3.5.59.4.

Changed to:

The server MUST process a change notification request in the object store as specified by the algorithm in section 3.3.1.3.</td></tr>
<tr><td>2016/11/07</td><td>In Section 2.2.33, SMB2 QUERY_DIRECTORY Request, the values in the Flags field have been changed from:

<table><tr><th>Value</th><th>Meaning</th></tr><tr><td>SMB2_RESTART_SCANS
0x01</td><td>The server MUST restart the enumeration from the beginning, but the search pattern is not changed.</td></tr><tr><td>SMB2_RETURN_SINGLE_ENTRY
0x02</td><td>The server MUST only return the first entry of the search results.</td></tr><tr><td>SMB2_INDEX_SPECIFIED
0x04</td><td>The server SHOULD<64> return entries beginning at the byte number specified by FileIndex.</td></tr><tr><td>SMB2_REOPEN
0x10</td><td>The server MUST restart the enumeration from the beginning, and the search pattern MUST be changed to the provided value. This often involves silently closing and reopening the directory on the server side. SMB2_REOPEN implies SMB2_RESTART_SCANS as well.</td></tr></table>

Changed to:</td></tr>
</table>

Value	Meaning
SMB2_RESTART_SCANS 0x01	The server MUST restart the enumeration from the beginning as specified in section 3.3.5.18.
SMB2_RETURN_SINGLE_ENTRY 0x02	The server MUST only return the first entry of the search results.
SMB2_INDEX_SPECIFIED 0x04	The server SHOULD<64> return entries beginning at the byte number specified by FileIndex.
SMB2_REOPEN 0x10	The server MUST restart the enumeration from the beginning, and the search pattern MUST be changed to the provided value.

In Section 3.3.1.10, Per Open, the following was removed:

• Open.EnumerationLocation: For directories, this value indicates the current location in a directory enumeration and allows for the continuing of an enumeration across multiple requests. For files, this value is unused.

• Open.EnumerationSearchPattern: For directories, this value holds the search pattern that is used in directory enumeration and allows for the continuing of an enumeration across multiple requests. For files, this value is unused.

In Section 3.3.5.9, Receiving an SMB2 CREATE Request, the following was removed:

• Open.EnumerationLocation is set to 0.

• Open.EnumerationSearchPattern is set to an empty string.

In Section 3.3.5.18, Receiving an SMB2 QUERY_DIRECTORY Request, the following was changed from:

If any other information class is specified in the FileInformationClass field of the SMB2 QUERY_DIRECTORY Request, the server MUST fail the operation with STATUS_INVALID_INFO_CLASS. If the information class requested is not supported by the server, the server MUST fail the request with STATUS_NOT_SUPPORTED.

If SMB2_REOPEN is set in the Flags field of the SMB2 QUERY_DIRECTORY Request, the server SHOULD<343> set Open.EnumerationLocation to 0 and Open.EnumerationSearchPattern to an empty string.

If SMB2_RESTART_SCANS is set in the Flags field of the SMB2 QUERY_DIRECTORY Request, the server MUST set Open.EnumerationLocation to 0.

If Open.EnumerationLocation is 0 and Open.EnumerationSearchPattern is an empty string, then Open.EnumerationSearchPattern MUST be set to the search pattern specified in the SMB2 QUERY_DIRECTORY by FileNameOffset and FileNameLength. If FileNameLength is 0, the server SHOULD<344> set Open.EnumerationSearchPattern as "*" to search all entries.

If SMB2_INDEX_SPECIFIED is set in the Flags field of the SMB2 QUERY_DIRECTORY Request and the underlying object store supports resuming enumerations by index number, the server MUST set Open.EnumerationLocation to the FileIndex received in the SMB2 QUERY_DIRECTORY Request. An underlying store MAY<345> choose to support resuming enumerations by index number.

Errata Published*	Description
	If SMB2_INDEX_SPECIFIED is set and FileNameLength is not zero, the server MUST set Open.EnumerationSearchPattern to the search pattern specified in the request by FileNameOffset and FileNameLength. The server MUST now enumerate the files and directories that are contained within the directory specified by Open.LocalOpen, starting at the index Open.EnumerationLocation.<346> Each entry MUST be formed as specified in [MS-FSCC] section 2.4. The server MUST fill in entries up to the OutputBufferLength received in the client request. The server MUST only include entries that match Open.EnumerationSearchPattern. For an explanation of wildcard evaluation for search patterns, see [MS-CIFS] section 2.2.1.1.3. If SMB2_RETURN_SINGLE_ENTRY is set in the Flags field of the request, the server MUST return only a single entry. If TreeConnect.Share.DoAccessBasedDirectoryEnumeration is TRUE and the object store supports security, the server MUST also exclude entries for which the user represented by Session.SecurityContext does not have FILE_READ_DATA or FILE_LIST_DIRECTORY access. After populating the buffer, the server MUST set Open.EnumerationLocation to the location of the next enumeration entry after the last one that was returned in the buffer. If there are no remaining entries, the server MUST set Open.EnumerationLocation to an invalid value indicating that the enumeration is complete. If an error is encountered, the server MUST fail the request with the error code received from the underlying object store by sending an error response as specified in section 2.2.2. If there are no entries to return and this was the initial query (Open.EnumerationLocation was zero before querying the object store), the server MUST fail the request with STATUS_NO_SUCH_FILE. If there are no entries to return and this was not the initial query (Open.EnumerationLocation was not zero before querying the object store), the server MUST fail the request with STATUS_NO_MORE_FILE S. <341> Section 3.3.5.18: The Windows SMB2 server implementation closes and reopens the directory handle in order to "reset" the enumeration state. So any outstanding operations on the directory handle will be failed with a STATUS_FILE_CLOSED error. <342> Section 3.3.5.18: If the length of the received data is less than the size of SMB2 header (0x40) plus size of SMB2 QUERY_DIRECTORY request (0x21), Windows servers fail the request with STATUS_INVALID_PARAMETER. Otherwise, if FileNameLength is 0 and the underlying file system is NTFS, Windows servers fail the request with STATUS_OBJECT_NAME_INVALID. <343> Section 3.3.5.18: Windows-based servers do not support resuming an enumeration at a specified FileIndex. The server will ignore this flag. <346> Section 3.3.5.18: Windows performs directory query information requests via the corresponding interfaces in [MS-FSA] section 2.1.5.5.3: • FileBothDirectoryInformation: [MS-FSA] section 2.1.5.5.3.1. • FileDirectoryInformation: [MS-FSA] section 2.1.5.5.3.2. • FileFullDirectoryInformation: [MS-FSA] section 2.1.5.5.3.3. • FileIdBothDirectoryInformation: [MS-FSA] section 2.1.5.5.3.4. • FileIdFullDirectoryInformation: [MS-FSA] section 2.1.5.5.3.5. • FileNamesInformation: [MS-FSA] section 2.1.5.5.3.6. <347> Section 3.3.5.19: Windows Vista SP1 and Windows Server 2008 limit OutputBufferLength size to 256 KB. Changed to: If any other information class is specified in the FileInformationClass field of the SMB2 QUERY_DIRECTORY Request, the server MUST fail the operation with STATUS_INVALID_INFO_CLASS. If the information class requested is not supported by the server, the server MUST fail the request with STATUS_NOT_SUPPORTED.

Errata Published*	Description
	If SMB2_RESTART_SCANS or SMB2_REOPEN is set in the Flags field of the SMB2 QUERY_DIRECTORY Request, the server MUST restart the scan with the search pattern specified, in an implementation-specific manner<341>. If SMB2_RETURN_SINGLE_ENTRY is set in the Flags field of the request, the server MUST return only a single entry. The server MUST invoke the query directory procedure from the underlying object store in an implementation-specific manner<342>. An Underlying object store MAY<343> choose to support resuming enumerations by index number, if SMB2_INDEX_SPECIFIED is set in the Flags field and an index number is specified in the FileIndex field of the SMB2 QUERY_DIRECTORY Request. If TreeConnect.Share.DoAccessBasedDirectoryEnumeration is TRUE and the object store supports security, the server MUST also exclude entries for which the user represented by Session.SecurityContext does not have FILE_READ_DATA or FILE_LIST_DIRECTORY access. <341> Section 3.3.5.18: Windows Vista SP1, Windows Server 2008, Windows 7, and Windows Server 2008 R2 close and reopen the directory handle prior to processing the request. <342> Section 3.3.5.18: Windows-based servers perform query directory requests, as specified in [MS-FSA] section 2.1.5.5 with the following input parameters: • Open is set to Open.LocalOpen. • FileInformationClass is set to the InformationClass that is received in the SMB2 QUERY_DIRECTORY Request. • OutputBufferSize is set to the OutputBufferLength that is received in the SMB2 QUERY_DIRECTORY Request. • If SMB2_RESTART_SCANS or SMB2_REOPEN is set in the Flags field of the SMB2 QUERY_DIRECTORY Request, RestartScan is set to TRUE. • If SMB2_RETURN_SINGLE_ENTRY is set in the Flags field of the request, ReturnSingleEntry is set to TRUE. • FileIndex is set to FileIndex received in the SMB2 QUERY_DIRECTORY Request. • FileNamePattern is set to the search pattern specified in the SMB2 QUERY_DIRECTORY by FileNameOffset and FileNameLength. <343> Section 3.3.5.18: Windows-based servers do not support resuming an enumeration at a specified FileIndex. The server will ignore this flag. <344> Section 3.3.5.19: Windows Vista SP1 and Windows Server 2008 limit OutputBufferLength size to 256 KB.
2016/10/10	In two sections, revised the description of SessionId. In Section 2.2.1.1, SMB2 Packet Header – ASYNC, changed from: SessionId (8 bytes): Uniquely identifies the established session for the command. This MUST be 0 for requests that do not have an associated user context. This MUST be 0 for the first SMB2 SESSION_SETUP Request for a specified security principal. The following SMB 2 Protocol commands do not require the SessionId to be set to a nonzero value received from a previous SMB2 SESSION_SETUP Response. The client MUST set the SessionId to 0, and the server SHOULD<2> ignore this value for the following commands: • SMB2 NEGOTIATE request • SMB2 NEGOTIATE response Changed to: SessionId (8 bytes): Uniquely identifies the established session for the command. This field MUST be set to 0 for an SMB2_NEGOTIATE request (section 2.2.3) and for an SMB2_NEGOTIATE response (section 2.2.4). In Section 2.2.1.2, SMB2 Packet Header – SYNC, changed from:

Errata Published*	Description
	SessionId (8 bytes): Uniquely identifies the established session for the command. This MUST be 0 for requests that do not have a user context that is associated with them. This MUST be 0 for the first SMB2 SESSION_SETUP Request for a specified security principal. The following SMB 2 Protocol commands do not require the SessionId to be set to a nonzero value received from a previous SMB2 SESSION_SETUP Response. The client MUST set SessionId to 0, and the server SHOULD<5> ignore this value for the following commands: • SMB2 NEGOTIATE Request • SMB2 NEGOTIATE Response Changed to: SessionId (8 bytes): Uniquely identifies the established session for the command. This field MUST be set to 0 for an SMB2_NEGOTIATE request (section 2.2.3) and for an SMB2_NEGOTIATE response (section 2.2.4).

*Date format: YYYY/MM/DD