[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 V66.0 – 2022/04/29.

Errata Published*	Description
2022/11/15	In Section 3.3.5.2.7, "Handling Compounded Requests," clarified applicable versions by revising a product behavior note: Changed from: <249> Section 3.3.5.2.7: In Windows Vista and Windows Server 2008, when an operation in a compound request requires asynchronous processing, Windows-based servers fail them with STATUS_INTERNAL_ERROR except for the following two cases: when a create request in the compound request triggers an oplock break, or when the operation is last in the compound request. Changed to: <249> Section 3.3.5.2.7: In Windows Vista and later, and Windows Server 2008 and later, when an operation in a compound request requires asynchronous processing, Windows-based servers fail them with STATUS_INTERNAL_ERROR except for the following two cases: when a create request in the compound request triggers an oplock break, or when the operation is last in the compound request.
2022/09/20	In Section 3.1.4.4, Compressing the Message, made the description generic because different implementations can make different criteria to determein when to compress on not to compress the data: Changed from: • 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 SMB2_COMPRESSION_CHAINED_PAYLOAD_HEADER. CompressionAlgorithm 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. Changed to: • Otherwise, if an implementation decides that the cost of remaining operations that might require copying the data is worth the encryption savings, then CompressedMessage MUST be appended with newly constructed SMB2_COMPRESSION_CHAINED_PAYLOAD_HEADER. CompressionAlgorithm 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.

<table>
<tr><th>Errata Published*</th><th>Description</th></tr>
<tr><td>2022/09/03</td><td>
In section 3.2.4.3, Application Requests Opening a File, added product behavior notes to clarify how leases are handled:

Changed from:

If an entry is not found, a new File entry MUST be created and added to the GlobalFileTable and a File.LeaseKey,<131> as specified in section 3.2.1.5, MUST be assigned to the entry. File.OpenTable MUST be initialized to an empty table and File.LeaseState MUST be initialized to SMB2_LEASE_NONE.

...
Otherwise, if Connection.SupportsFileLeasing is TRUE, the client SHOULD set RequestedOplockLevel field to SMB2_OPLOCK_LEVEL_LEASE.

Changed to:

If an entry is not found, a new File entry MUST be created and added to the GlobalFileTable and a File.LeaseKey,<131> as specified in section 3.2.1.5, MUST be assigned to the entry.<132> File.OpenTable MUST be initialized to an empty table and File.LeaseState MUST be initialized to SMB2_LEASE_NONE.

If an entry is found, the client MUST include a lease context with the existing lease key, lease state, and epoch.<133>

...

● Otherwise, if Connection.SupportsFileLeasing is TRUE, the client SHOULD<135> set RequestedOplockLevel field to SMB2_OPLOCK_LEVEL_LEASE.

<132> Section 3.2.4.3: On Windows 7 operating system and Windows Server 2008 R2, a 128-bit ClientLeaseId is generated by an arithmetic combination of LeaseKey and ClientGuid, which is passed to the object store at open/create time. On Windows 8 operating system and later and Windows Server 2012 operating system and later, the LeaseKey in the request is used as the ClientLeaseId.
<133> Section 3.2.4.3: On Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2, the Lease.ClientLeaseId and Lease.ParentLeaseKey are passed to the object store in the form of TargetOplockKey and ParentOplockKey. A new or existing lease is thereby associated with the resulting open.
To acquire or promote the lease as dictated by the SMB2_CREATE_REQUEST_LEASE_V2 Create Context, a subsequent object store call is invoked as described in. [MS-FSA] section 2.1.5.18 Server Requests an Oplock. The Open parameter passed is the Open result from the above operation, and the Type parameter is LEVEL_GRANULAR to indicate a Lease request. The RequestedOplockLevel field is constructed to include zero or more bits as follows.
<table>
<tr><th>Object Store RequestedOplockLevel bit to be set</th><th>SMB2 Lease.LeaseState bit requested</th></tr>
<tr><td>READ_CACHING</td><td>SMB2_LEASE_READ_CACHING</td></tr>
<tr><td>WRITE_CACHING</td><td>SMB2_LEASE_WRITE_CACHING</td></tr>
<tr><td>HANDLE_CACHING</td><td>SMB2_LEASE_HANDLE_CACHING</td></tr>
</table>

The Status code returned indicates whether the requested lease was granted.
</td></tr>
</table>

Errata Published*	Description
	<135> Section 3.2.4.3: Microsoft Windows lease-aware clients always include SMB2_OPLOCK_LEVEL_LEASE if the open can potentially cause a lease break.
2022/07/26	In Section 3.2.4.3 Application Requests Opening a File, updated what file elements client uses when it accesses same path across multiple opens. Changed From: If the client accesses a file through multiple paths, such as using different server names or share names or parent directory names, it will create multiple File elements, and therefore multiple File.LeaseKeys for the same remote file. This loses the performance benefits of sharing cache state across all Opens of the same file and can cause additional lease breaks to be generated, as actions by a client through one path will affect caching by that client through other paths. However, the impact is a matter of performance; cache correctness is preserved. Changed To: If the client accesses a file through multiple paths, such as using different server names or share names or parent directory names, it will create multiple File elements, and therefore multiple File.LeaseKeys for the same remote file. This loses the performance benefits of sharing cache state across all Opens of the same file and can cause additional lease breaks to be generated, as actions by a client through one path will affect caching by that client through other paths. However, the impact is a matter of performance; cache correctness is preserved. If the client accesses same path across multiple opens, the client will use same File element and therefore same File.LeaseKey is used. In Section 3.2.4.3.8 Requesting a Lease on a File or a Directory, updated setting of LeaseKey field for SMB2_CREATE_REQUEST_LEASE_V2 create context Changed From: . LeaseKey obtained from File.LeaseKey of the file or directory being opened. Changed To: . LeaseKey is set to File.LeaseKey obtained from section 3.2.4.3.
2022/07/12	In Section 3.2.4.3 Application Requests Opening a File, updated what file elements client uses when it accesses same path across multiple opens. Changed From: If the client accesses a file through multiple paths, such as using different server names or share names or parent directory names, it will create multiple File elements, and therefore multiple File.LeaseKeys for the same remote file. This loses the performance benefits of sharing cache state across all Opens of the same file and can cause additional lease breaks to be generated, as actions by a client through one path will affect caching by that client through other paths. However, the impact is a matter of performance; cache correctness is preserved. Changed To: If the client accesses a file through multiple paths, such as using different server names or share names or parent directory names, it will create multiple File elements, and therefore multiple File.LeaseKeys for the same remote file. This loses the performance benefits of sharing cache state across all Opens of the same file and can cause additional lease breaks to be generated, as actions by a client through one path will affect caching by that client through other paths. However, the impact is a matter of performance; cache correctness is preserved. If the client accesses same path across multiple opens, the client will use same File element and therefore same File.LeaseKey is used. In Section 3.2.4.3.8 Requesting a Lease on a File or a Directory, updated setting of LeaseKey field for SMB2_CREATE_REQUEST_LEASE_V2 create context Changed From:

Errata Published*	Description
	. LeaseKey obtained from File.LeaseKey of the file or directory being opened. Changed To: . LeaseKey is set to File.LeaseKey obtained from section 3.2.4.3.
2022/06/28	In Section 2.2.41 SMB2 TRANSFORM_HEADER, updated the definition of signature field. Changed from: Signature (16 bytes): The 16-byte signature of the encrypted message generated by using Session.EncryptionKey. Changed to: Signature (16 bytes): The 16-byte signature of the message generated using negotiated encryption algorithm. In Section 2.2.43.1 SMB2_RDMA_CRYPTO_TRANSFORM, updated the definition of signature field. Changed from: Signature (variable): The signature of the encrypted/signed data generated using Session.EncryptionKey. The length of this field MUST be less than or equal to 16 bytes. Changed to: Signature (variable): The signature of the data generated using negotiated encryption/signing algorithm. The length of this field MUST be less than or equal to 16 bytes.
2022/06/28	In section 3.2.5.15, Receiving an SMB2 Query_Directory response, added information about a case where STATUS_BUFFER_OVERFLOW is returned and the buffer content length is zero. Changed from: If the Status field of the SMB2 header of the response indicates success, the client MUST copy the received information in the SMB2 QUERY_DIRECTORY Response following the SMB2 header that is described by the OutputBufferOffset and OutputBufferLength into the buffer that is provided by the calling application. The client MUST return success and the OutputBufferLength to the application. Changed to: If the Status field of the SMB2 header of the response indicates success, the client MUST copy the received information in the SMB2 QUERY_DIRECTORY Response following the SMB2 header that is described by the OutputBufferOffset and OutputBufferLength into the buffer that is provided by the calling application. The client MUST return success and the OutputBufferLength to the application. There can be cases where STATUS_BUFFER_OVERFLOW is returned and the OutputBufferSize is set to zero. See [MSDOCS-ABEConcepts] for an example of such a case where output entries are filtered when the requester does not have the required permissions. [MS-FSA] section 2.1.5.6.3 describes the algorithm.
2022/06/01	In Section 3.3.5.9.12 Handling the SMB2_CREATE_DURABLE_HANDLE_RECONNECT_V2 Create Context, updated setting Epoch field in the case of handling the

Errata Published*	Description
	SMB2_CREATE_DURABLE_HANDLE_RECONNECT_V2 with SMB2_CREATE_REQUEST_LEASE_V2 create context. Changed From: . If Lease.LeaseState includes SMB2_LEASE_WRITE_CACHING, the server MUST set Lease.Epoch to the Epoch field in the Create Context request. Otherwise, the server MUST set Lease.Epoch to the Epoch field in the Create Context request incremented by 1. Epoch MUST be set to Lease.Epoch. Changed To: . Epoch SHOULD<329> be set to Lease.Epoch. <329> When an open, with Open.IsPersistent set to TRUE, is being reconnected due to server failover, Windows Server 2012 operating system and later perform the following: . If Lease.LeaseState includes SMB2_LEASE_WRITE_CACHING, Epoch and Lease.Epoch are set to Epoch field in the Create Context request. . If Lease.LeaseState does not include SMB2_LEASE_WRITE_CACHING, Epoch and Lease.Epoch are set to Epoch field in the Create Context request incremented by 1.
2022/06/01	In Section 3.2.4.4 Re-establishing a Durable Open, updated setting Epoch field in the case of re-establishing a durable open with SMB2_CREATE_REQUEST_LEASE_V2 create context. Changed From: . If Connection.Dialect is not "2.0.2", and the original open was performed by using a lease as described in section 3.2.4.3.8, as indicated by Open.OplockLevel set to SMB2_OPLOCK_LEVEL_LEASE, it MUST also implement the following: . The client MUST re-request the lease as described in section 3.2.4.3.8, and the LeaseState field MUST be set to File.LeaseState of the file being opened. Changed To: . If Connection.Dialect is not "2.0.2", and the original open was performed by using a lease as specified in section 3.2.4.3.8, as indicated by Open.OplockLevel set to SMB2_OPLOCK_LEVEL_LEASE, the client MUST re-request the lease as specified in section 3.2.4.3.8 with the exception of the following values: . The LeaseState field MUST be set to File.LeaseState of the file being opened. . If Connection.Dialect belongs to the SMB 3.x dialect family, the Epoch field MUST be set to File.LeaseEpoch of the file being opened.
2022/06/01	In Section 3.3.4.7, Object Store Indicates an Oplock Break, updated the text to address the Open issues and setting of lease state. Changed from: If a Lease entry is found, the server MUST perform the following: If Lease.LeaseOpens is empty, the server MUST complete the lease break call from the underlying object store with "NONE" as the new lease state, set Lease.LeaseState to "NONE", and take no further action. Otherwise, for the specified Open, the server MUST select the connection as specified in section 3.3.4.1.6.

Errata Published*	Description
	If no connection is available, for each Open in Lease.LeaseOpens, the server MUST close the Open as specified in section 3.3.4.17 for the following cases: ● Open.IsDurable, Open.IsResilient, and Open.IsPersistent are all FALSE. ● Lease.BreakToLeaseState does not contain SMB2_LEASE_HANDLE_CACHING and Open.IsDurable is TRUE. … 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<227> default value in milliseconds. Changed to: If a Lease entry is found, the server MUST perform the following: If Lease.LeaseOpens is empty, the server MUST complete the lease break call from the underlying object store with "NONE" as the new lease state, set Lease.LeaseState to "NONE", and take no further action. If no connection is available among all Opens in Lease.LeaseOpens, the server MUST close every Open as specified in section 3.3.4.17 in one of the following cases: ● Open.IsDurable, Open.IsResilient, and Open.IsPersistent are all FALSE. ● The new lease state indicated by object store does not contain SMB2_LEASE_HANDLE_CACHING and Open.IsDurable is TRUE. … 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 and NewLeaseState field to the new lease state indicated by the object store, and set Lease.LeaseBreakTimeout to the current time plus an implementation-specific<227> default value in milliseconds.
2022/05/27	In section 3.3.5.15, Receiving an SMB2 IOCTL Request, updated the list of applicable updates. Changed from: Processing of FSCTL_SET_INTEGRITY_INFORMATION_EX is handled as described in [MS-FSA] and [MS-FSCC] when the system is updated with [MSKB-5014019], [MSKB-5014021], [MSKB-5014022], and [MSKB-5014023]. Changed to: Processing of FSCTL_SET_INTEGRITY_INFORMATION_EX is handled as described in [MS-FSA] and [MS-FSCC] when the system is updated with [MSKB-5014019], [MSKB-5014021], [MSKB-5014022], [MSKB-5014023], [MSKB-5014701], [MSKB-5014702], or [MSKB-5014710].
2022/05/18	In Section 3.3.5.22.2, Processing a Lease Acknowledgment, updated the text to remove the symbols: Changed from: If LeaseState is not <= Lease.BreakToLeaseState, the server MUST fail the request with STATUS_REQUEST_NOT_ACCEPTED. Changed to: If LeaseState is not a subset of Lease.BreakToLeaseState, the server MUST fail the request with STATUS_REQUEST_NOT_ACCEPTED.
2022/05/02	In Section 3.3.5.15, Receiving an SMB2 IOCTL Request, updated processing rules for system versions.

Errata Published*	Description
	Changed from: The server SHOULD<355> fail the request with STATUS_NOT_SUPPORTED when an FSCTL is not allowed on the server, and SHOULD<356> fail the request with STATUS_INVALID_DEVICE_REQUEST when the FSCTL is allowed, but is not supported on the file system on which the file or directory handle specified by the FSCTL exists, as specified in [MS-FSCC] section 2.2. Changed to: The server SHOULD<355> fail the request with STATUS_NOT_SUPPORTED when an FSCTL is not allowed on the server, and SHOULD<356> fail the request with STATUS_INVALID_DEVICE_REQUEST when the Processing of FSCTL_SET_INTEGRITY_INFORMATION_EX is handled as described in [MS-FSA] and [MS-FSCC] when the system is updated with [MSKB-5014019], [MSKB-5014021], [MSKB-5014022], and [MSKB-5014023]. FSCTL is allowed, but is not supported on the file system on which the file or directory handle specified by the FSCTL exists, as specified in [MS-FSCC] section 2.2.