[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 V47.0 – 2015/06/30.

Errata Published*	**Description**
2015/09/28	In Section 3.2.4.3.5, Application Requests Creating a File Opened for Durable Operation, updated required client operations for applications requesting durability. Changed from: If the application is requesting durability, the client MUST do the following: ▪ If Connection.Dialect belongs to the SMB 3.x dialect family, the client MUST construct a create context by using the syntax specified in section 2.2.13.2.11, with the following values set: ▪ Timeout MUST be set to an implementation-specific value <115>. ▪ If TreeConnect.IsCAShare is TRUE, the client MUST set the SMB2_DHANDLE_FLAG_PERSISTENT bit in the Flags field. ▪ Reserved MUST be set to zero. ▪ CreateGuid MUST be set to a newly generated GUID. ▪ If the SMB2_DHANDLE_FLAG_PERSISTENT bit is not set in the Flags field, the client MUST perform one of the following: ▪ Request a batch oplock by setting RequestedOplockLevel in the create request to SMB2_OPLOCK_LEVEL_BATCH. ▪ Request a handle caching lease by including an SMB2_CREATE_REQUEST_LEASE or SMB2_CREATE_REQUEST_LEASE_V2 Create Context in the create request with a LeaseState that includes SMB2_LEASE_HANDLE_CACHING. ▪ Otherwise, the client MUST construct a create context using the syntax specified in section 2.2.13.2.3. The client MUST also perform one of the following: ▪ Request a batch oplock by setting RequestedOplockLevel in the create request to SMB2_OPLOCK_LEVEL_BATCH. ▪ Request a handle caching lease by including an SMB2_CREATE_REQUEST_LEASE Create Context in the create request with a LeaseState that includes SMB2_LEASE_HANDLE_CACHING. ▪ The client MUST append the newly constructed create context to any other create contexts being issued with this CREATE request. If the application is not requesting durability, the client MUST follow the normal processing, as specified in section 3.2.4.3. Changed to: If the application is requesting durability, the client MUST do the following: ▪ If Connection.Dialect belongs to the SMB 3.x dialect family, the client MUST construct a create context by using the syntax specified in section 2.2.13.2.11, with the following values set: ▪ Timeout MUST be set to an implementation-specific value <115>.

Errata Published*	Description
	▪ If TreeConnect.IsCAShare is TRUE, the client MUST set the SMB2_DHANDLE_FLAG_PERSISTENT bit in the Flags field. Otherwise, the client SHOULD perform one of the following: ▪ Request a batch oplock by setting RequestedOplockLevel in the create request to SMB2_OPLOCK_LEVEL_BATCH. ▪ Request a handle caching lease by including an SMB2_CREATE_REQUEST_LEASE or SMB2_CREATE_REQUEST_LEASE_V2 Create Context in the create request with a LeaseState that includes SMB2_LEASE_HANDLE_CACHING. ▪ Reserved MUST be set to zero. ▪ CreateGuid MUST be set to a newly generated GUID. ▪ Otherwise, the client MUST construct a create context using the syntax specified in section 2.2.13.2.3. The client SHOULD perform one of the following: ▪ Request a batch oplock by setting RequestedOplockLevel in the create request to SMB2_OPLOCK_LEVEL_BATCH. ▪ Request a handle caching lease by including an SMB2_CREATE_REQUEST_LEASE Create Context in the create request with a LeaseState that includes SMB2_LEASE_HANDLE_CACHING. ▪ The client MUST append the newly constructed create context to any other create contexts being issued with this CREATE request. If the application is not requesting durability, the client MUST follow the normal processing, as specified in section 3.2.4.3.
2015/09/28	In Section 3.3.5.9.10, Handling the SMB2_CREATE_DURABLE_HANDLE_REQUEST_V2 Create Context, the 8th paragraph has been changed. Changed from: If an Open is found and the SMB2_FLAGS_REPLAY_OPERATION bit is set in the SMB2 header, the server MUST construct an SMB2_CREATE_DURABLE_HANDLE_RESPONSE_V2 response create context. The Timeout MUST be set to Open.DurableOpenTimeout. If Open.IsPersistent is TRUE, the server MUST set the SMB2_DHANDLE_FLAG_PERSISTENT bit in the Flags field. The Buffer specified by the response MUST include the CreateContextsLength and CreateContextsOffset fields. Changed to: If an Open is found and the SMB2_FLAGS_REPLAY_OPERATION bit is set in the SMB2 header, the server MUST perform the following: ▪ The server MUST set Open.Connection to the connection that received this request. ▪ The server MUST construct an SMB2_CREATE_DURABLE_HANDLE_RESPONSE_V2 create context as follows: ▪ The Timeout field MUST be set to Open.DurableOpenTimeout. ▪ If Open.IsPersistent is TRUE, the server MUST set the SMB2_DHANDLE_FLAG_PERSISTENT bit in the Flags field. ▪ The Buffer specified by the response MUST include the CreateContextsLength and CreateContextsOffset fields.
2015/09/14	In Section 3.3.5.16, Receiving an SMB2 CANCEL Request, corrected the second paragraph. Changed from: An SMB2 CANCEL Request is the only request received by the server that is not signed and does not contain a sequence number that must be checked. Thus, the server MUST NOT process the received packet as specified in sections 3.3.5.2.3 and 3.3.5.2.4.

Errata Published*	**Description**
	Changed to: An SMB2 CANCEL Request does not contain a sequence number that must be checked. Thus, the server MUST NOT process the received packet as specified in section 3.3.5.2.3.
2015/08/17	In Section 3.3.5.5, Receiving an SMB2 SESSION_SETUP Request, added information about session binding from a different client Guid. Changed from: ▪ … ▪ If the SMB2_FLAGS_SIGNED bit is not set in the Flags field in the header, the server MUST fail the request with error STATUS_INVALID_PARAMETER. ▪ If Session.State is InProgress, the server MUST fail the request with STATUS_REQUEST_NOT_ACCEPTED. ▪ … Changed to: ▪ … ▪ If the SMB2_FLAGS_SIGNED bit is not set in the Flags field in the header, the server MUST fail the request with error STATUS_INVALID_PARAMETER. ▪ If Session.Connection.ClientGuid is not the same as Connection.ClientGuid, the server MAY fail the request with STATUS_USER_SESSION_DELETED. ▪ If Session.State is InProgress, the server MUST fail the request with STATUS_REQUEST_NOT_ACCEPTED. ▪ …
2015/08/03	In Section 3.3.5.9, Receiving an SMB2 CREATE Request, corrected the list of Windows product versions that check Treeconnect.MaximalAccess when deleting a file. Changed from: If the FILE_DELETE_ON_CLOSE flag is set in CreateOptions and any of the following conditions is TRUE, the server SHOULD<249> fail the request with STATUS_ACCESS_DENIED. <249> Section 3.3.5.9: Windows Vista SP1, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, and Windows Server 2012 do not perform this verification. Changed to: If the FILE_DELETE_ON_CLOSE flag is set in CreateOptions and any of the following conditions is TRUE, the server SHOULD<249> fail the request with STATUS_ACCESS_DENIED. <249> Section 3.3.5.9: Windows Vista SP1, Windows Server 2008, Windows 7, and Windows Server 2008 R2 do not perform this verification.
2015/08/03	In several sections, changed the reference that specifies the HMAC-SHA256 algorithms. In sections:

<table>
<tr><th>Errata Published*</th><th>Description</th></tr>
<tr><td></td><td>
<p>1.6, Applicability Statement</p>
<p>3.1.4.1, Signing An Outgoing Message</p>
<p>3.1.5.1, Verifying an Incoming Message</p>
<p>Changed from:</p>
<p>[FIPS180-2] National Institute of Standards and Technology, "Secure Hash Standard", FIPS PUB 180-2, August 2002, http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf</p>
<p>Changed to:</p>
<p>[FIPS180-4] FIPS PUBS, "Secure Hash Standards (SHS)", March 2012, http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf</p>
<p>In Section 2.2.3.1.1, SMB2_PREAUTH_INTEGRITY_CAPABILITIES, added this updated reference to the HashAlgorithms table.</p>
<p>Changed from:</p>
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0x0001</td><td>SHA-512</td></tr>
</table>
<p>Changed to:</p>
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0x0001</td><td>SHA-512 as specified in [FIPS180-4]</td></tr>
</table>
</td></tr>
<tr><td>2015/08/03</td><td>
<p>In Section 2.2.21, SMB2 WRITE Request, corrected that SMB2_CHANNEL_RDMA_V1_INVALIDATE should reference WriteChannelInfoOffset and WriteChannelInfoLength fields.</p>
<p>Changed from:</p>
<p>Channel (4 bytes): For the SMB 2.0.2 and 2.1 dialects, this field MUST NOT be used and MUST be reserved. The client MUST set this field to 0, and the server MUST ignore it on receipt. For the SMB 3.x dialect family, this field MUST contain exactly one of the following values:</p>
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>SMB2_CHANNEL_RDMA_V1_INVALIDATE
0x00000002</td><td>This flag is not valid for the SMB 2.0.2, 2.1, and 3.0 dialects. One or more SMB_DIRECT_BUFFER_DESCRIPTOR_V1 structures as specified in [MS-SMBD] section 2.2.3.1 are present in the channel information specified by the <b>ReadChannelInfoOffset</b> and <b>ReadChannelInfoLength</b> fields. The server is requested to perform remote invalidation when responding to the request as specified in [MS-SMBD] section 3.1.4.2.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Errata Published*</th><th>Description</th></tr>
<tr><td></td><td>
<p>Changed to:</p>
<p>Channel (4 bytes): For the SMB 2.0.2 and 2.1 dialects, this field MUST NOT be used and MUST be reserved. The client MUST set this field to 0, and the server MUST ignore it on receipt. For the SMB 3.x dialect family, this field MUST contain exactly one of the following values:</p>
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>SMB2_CHANNEL_RDMA_V1_INVALIDATE

0x00000002</td><td>This flag is not valid for the SMB 2.0.2, 2.1, and 3.0 dialects. One or more SMB_DIRECT_BUFFER_DESCRIPTOR_V1 structures as specified in [MS-SMBD] section 2.2.3.1 are present in the channel information specified by the <b>WriteChannelInfoOffset</b> and <b>WriteChannelInfoLength</b> fields. The server is requested to perform remote invalidation when responding to the request as specified in [MS-SMBD] section 3.1.4.2.</td></tr>
</table>
</td></tr>
<tr><td>2015/08/03</td><td>
<p>In two sections, clarified that SMB2_GLOBAL_CAP_ENCRYPTION applies to SMB 3.0 and 3.0.2 dialects.</p>
<p>In Section 2.2.4,SMB2 NEGOTIATE Response, changed from:</p>
<p>Capabilities (4 bytes): The Capabilities field specifies protocol capabilities for the server. This field MUST be constructed using a combination of zero or more of the following values.</p>
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>SMB2_GLOBAL_CAP_ENCRYPTION

0x00000040</td><td>When set, indicates that the server supports encryption. This flag is not valid for the SMB 2.0.2 and SMB 2.1 dialects.</td></tr>
</table>
<p>Changed to:</p>
<p>Capabilities (4 bytes): The Capabilities field specifies protocol capabilities for the server. This field MUST be constructed using a combination of zero or more of the following values.</p>
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>SMB2_GLOBAL_CAP_ENCRYPTION

0x00000040</td><td>When set, indicates that the server supports encryption. This flag is valid for the SMB 3.0 and 3.0.2 dialects.</td></tr>
</table>
<p>In Section 3.2.5.2, Receiving an SMB2 NEGOTIATE Response, changed from:</p>
<p>If the client implements SMB 2.1 or SMB 3.x dialect family, the client MUST perform the following:</p>
<ul><li>The client MUST store the returned dialect in Connection.Dialect.</li></ul>
<p>…</p>
</td></tr>
</table>

Errata Published*	Description
	If Connection.Dialect belongs to the SMB 3.x dialect family, the client MUST perform the following: - If SMB2_GLOBAL_CAP_DIRECTORY_LEASING is set in the Capabilities field of the SMB2 NEGOTIATE Response, the client MUST set Connection.SupportsDirectoryLeasing to TRUE. Otherwise, it MUST be set to FALSE. Changed to: If the client implements SMB 2.1 or SMB 3.x dialect family, the client MUST perform the following: - The client MUST set Connection.Dialect to DialectRevision in the SMB2 NEGOTIATE Response. … If Connection.Dialect belongs to the SMB 3.x dialect family, the client MUST perform the following: - If SMB2_GLOBAL_CAP_ENCRYPTION is set in the Capabilities field of the SMB2 NEGOTIATE Response and Connection.Dialect is "3.0" or "3.0.2", the client MUST set Connection.SupportsEncryption to TRUE. Otherwise, it MUST be set to FALSE.
2015/08/03	In 5 sections, corrected that Open.LockSequenceArray[] must be initialized to 0xFF, not 0. In Section 3.3.1.10, Per Open, changed from: If the server implements the SMB 2.1 or SMB 3.x dialect family and supports leasing, it MUST implement the following: - … - Open.ResilientOpenTimeout: A time value that indicates when a handle that has been preserved for resiliency will be closed by the system if a client has not reclaimed it. - Open.LockSequenceArray: An array of 64 entries used to maintain lock sequences for resilient Opens. Each entry value MUST be empty, or MUST be 4-bit integer modulo 16. Each entry MUST be assigned an index from the range of 1 to 64. Changed to: If the server implements the SMB 2.1 or SMB 3.x dialect family and supports leasing, it MUST implement the following: - … - Open.ResilientOpenTimeout: A time-out value that indicates when a handle that has been preserved for resiliency will be closed by the system if a client has not reclaimed it. - Open.LockSequenceArray: An array of 64 entries used to maintain lock sequences for resilient opens. Each entry MUST be assigned an index from the range of 1 to 64. Each entry is a structure with the following elements:

Errata Published*	Description
	▪ SequenceNumber: A 4-bit integer modulo 16. ▪ Valid: A Boolean, if set to TRUE, indicates that the SequenceNumber element is valid. In Section 3.3.5.9, Receiving an SMB2 CREATE Request, changed from: If Connection.Dialect is not "2.0.2" and the server supports leasing, the server MUST initialize the following: ▪ … ▪ Open.ResilientOpenTimeout MUST be set to 0. ▪ Open.LockSequenceArray: Each element of Open.LockSequenceArray MUST be initialized to empty. Changed to: If Connection.Dialect is not "2.0.2" and the server supports leasing, the server MUST initialize the following: ▪ … ▪ Open.ResilientOpenTimeout MUST be set to 0. ▪ Each entry of Open.LockSequenceArray MUST be initialized as follows: ▪ Set Valid to FALSE. In Section 3.3.5.14, Receiving an SMB2 LOCK Request, changed from: The server verifies the LockSequence by performing the following steps: ▪ The server MUST use LockSequenceIndex as an index into Open.LockSequenceArray in order to locate the sequence number entry. If the index exceeds the maximum extent of the Open.LockSequenceArray, or LockSequenceIndex is 0, or if the sequence number entry is empty, the server MUST skip step 2 and continue lock/unlock processing. ▪ The server MUST compare LockSequenceNumber to the SequenceNumber of the entry located in step 1. If the sequence numbers are equal, the server MUST complete the lock/unlock request with success. Otherwise, the server MUST reset the entry value to empty and continue lock/unlock processing. Changed to: The server verifies the LockSequence by performing the following steps:

Errata Published*	Description
	▪ The server MUST use LockSequenceIndex as an index into Open.LockSequenceArray in order to locate the sequence number entry. If the index exceeds the maximum extent of the Open.LockSequenceArray, or LockSequenceIndex is 0, or if the Open.LockSequenceArray.Valid is FALSE, the server MUST skip step 2 and continue lock/unlock processing. ▪ The server MUST compare LockSequenceNumber to the SequenceNumber of the entry located in step 1. If the sequence numbers are equal, the server MUST complete the lock/unlock request with success. Otherwise, the server MUST reset the entry by setting Valid to FALSE and continue lock/unlock processing. In Section 3.3.5.14.1, Processing Unlocks, changed from: If the unlock operation succeeds and there are no remaining entries in the Locks array, Connection.Dialect is "2.1" or belongs to the SMB 3.x dialect family, the server supports leasing, and Open.IsResilient is TRUE, the server MUST set the lock sequence number in Open.LockSequenceArray through the following step to indicate that the unlock request with LockSequence has been successfully processed by the server: ▪ If an entry is found via the lock request process described in the numbered list in section 3.3.5.14, the server MUST save LockSequenceNumber into the corresponding entry. Changed to: If the unlock operation succeeds and there are no remaining entries in the Locks array, Connection.Dialect is "2.1" or belongs to the SMB 3.x dialect family, the server supports leasing, and Open.IsResilient is TRUE, the server MUST set the lock sequence number in Open.LockSequenceArray through the following step to indicate that the unlock request with LockSequence has been successfully processed by the server: ▪ If an entry is found via the lock request process described in the numbered list in section 3.3.5.14, the server MUST set Valid to TRUE and save LockSequenceNumber into SequenceNumber of the corresponding entry. In Section 3.3.5.14.2, Processing Locks, changed from: If the lock operation succeeds and there are no remaining entries in the Locks array, Connection.Dialect is "2.1" or belongs to the SMB 3.x dialect family, the server supports leasing, and Open.IsResilient is TRUE, the server MUST set the lock sequence number in Open.LockSequenceArray through the following step to indicate that the lock request with LockSequence has been successfully processed by the server: ▪ If an entry is found via the lock request process described in the numbered list in section 3.3.5.14, the server MUST save LockSequenceNumber into the corresponding entry. Changed to:

Errata Published*	**Description**
	If the lock operation succeeds and there are no remaining entries in the Locks array, Connection.Dialect is "2.1" or belongs to the SMB 3.x dialect family, the server supports leasing, and Open.IsResilient is TRUE, the server MUST set the lock sequence number in Open.LockSequenceArray through the following step to indicate that the lock request with LockSequence has been successfully processed by the server: ▪ If an entry is found via the lock request process described in the numbered list in section 3.3.5.14, the server MUST set Valid to TRUE and save LockSequenceNumber into SequenceNumber of the corresponding entry

*Date format: YYYY/MM/DD