[MS-SMB2]: Server Message Block (SMB) Protocol Versions 2 and 3

Errata below are for Protocol Document Version V52.0 – 2017/06/01.

Errata Published*	Description
2017/08/07	In Section 3.3.5.21.1, Handling SMB2_0_INFO_FILE, changed from: If FileInfoClass is FileRenameInformation and the size of the buffer is less than the size of FILE_RENAME_INFORMATION_TYPE_2 as specified in [MS-FSCC] section 2.4.34.2, the server MUST fail the request with STATUS_INFO_LENGTH_MISMATCH. Changed to: If FileInfoClass is FileRenameInformation and, the server does the following: --If the size of the buffer is less than the size of FILE_RENAME_INFORMATION_TYPE_2 as specified in [MS-FSCC] section 2.4.34.2, the server MUST fail the request with STATUS_INFO_LENGTH_MISMATCH. --If the file name pointed to by the FileName parameter of the FILE_RENAME_INFORMATION_TYPE_2, as specified in [MS-FSCC] section 2.4.34.2, contains a separator character, then the server MUST fail the request with STATUS_NOT_SUPPORTED. --If the RootDirectory field of FILE_RENAME_INFORMATION_TYPE_2 as specified in [MS-FSCC] section 2.4.34.2 is zero, the FileName field MUST specify a full pathname as specified in [MS-FSCC] section 2.1.5 to be assigned to the file. If the RootDirectory field is not zero, the server MUST return STATUS_INVALID_PARAMETER.
2017/06/15	In Section 2.2.1.1, SMB2 Packet Header – ASYNC, and Section 2.2.1.2, SMB2 Packet Header – SYNC, the Signature field was changed from: Signature (16 bytes): The 16-byte signature of the message, if SMB2_FLAGS_SIGNED is set in the Flags field of the SMB2 header. If the message is not signed, this field MUST be 0. Changed to: Signature (16 bytes): The 16-byte signature of the message, if SMB2_FLAGS_SIGNED is set in the Flags field of the SMB2 header and the message is not encrypted. If the message is not signed, this field MUST be 0. In Section 3.2.4.1.1, Signing the Message, a new sentence was added to the last paragraph: Changed from: If the client implements the SMB 3.x dialect family, and if the request is for session set up, the client MUST use Session.SigningKey, and for all other requests the client MUST provide Channel.SigningKey by looking up the Channel in Session.ChannelList, where the connection matches the Channel.Connection. Otherwise, the client MUST use Session.SessionKey for signing the request. The client provides the key for signing, the length of the request, and the request itself, and calculates the signature as specified in section 3.1.4.1. If the client signs the request, it MUST set the SMB2_FLAGS_SIGNED bit in the Flags field of the SMB2 header. Changed to:

Errata Published*	Description
	If the client implements the SMB 3.x dialect family, and if the request is for session set up, the client MUST use Session.SigningKey, and for all other requests the client MUST provide Channel.SigningKey by looking up the Channel in Session.ChannelList, where the connection matches the Channel.Connection. Otherwise, the client MUST use Session.SessionKey for signing the request. The client provides the key for signing, the length of the request, and the request itself, and calculates the signature as specified in section 3.1.4.1. If the client signs the request, it MUST set the SMB2_FLAGS_SIGNED bit in the Flags field of the SMB2 header. If the client encrypts the message, as specified in section 3.1.4.3, then the client MUST set the Signature field of the SMB2 header to zero. In Section 3.2.5.1.3, Verifying the Signature, the third paragraph was changed from: If the SMB2 header of the response has SMB2_FLAGS_SIGNED set in the Flags field, the client MUST verify the signature as follows: Changed to: If the SMB2 header of the response has SMB2_FLAGS_SIGNED set in the Flags field and the message is not encrypted, the client MUST verify the signature as follows: In Section 3.3.4.1.1, Signing the Message, a new sentence was added to the last paragraph: Changed from: The server provides the key for signing, the length of the response, and the response itself, and calculates the signature as specified in section 3.1.4.1. If the server signs the message, it MUST set the SMB2_FLAGS_SIGNED bit in the Flags field of the SMB2 header. Changed to: The server provides the key for signing, the length of the response, and the response itself, and calculates the signature as specified in section 3.1.4.1. If the server signs the message, it MUST set the SMB2_FLAGS_SIGNED bit in the Flags field of the SMB2 header. If the server encrypts the message, as specified in section 3.1.4.3, the server MUST set the Signature field of the SMB2 header to zero. In Section 3.3.5.2.4, Verifying the Signature, the first sentence of the third paragraph was changed from: If the SMB2 header of the request has SMB2_FLAGS_SIGNED set in the Flags field, the server MUST verify the signature. Changed to: If the SMB2 header of the request has SMB2_FLAGS_SIGNED set in the Flags field and the message is not encrypted, the server MUST verify the signature.
2017/06/15	In 8 subsections in Section 2, Message Syntax, listed below, processing information was moved to more appropriate subsections in Section 3, Protocol Details. • Section 2.2.10, SMB2 TREE_CONNECT Response • Section 2.2.13.2.1, SMB2_CREATE_EA_BUFFER • Section 2.2.13.2.2, SMB2_CREATE_SD_BUFFER • Section 2.2.14.2.3, SMB2_CREATE_DURABLE_HANDLE_RESPONSE • Section 2.2.14.2.5, SMB2_CREATE_QUERY_MAXIMAL_ACCESS_RESPONSE • Section 2.2.14.2.12, SMB2_CREATE_DURABLE_HANDLE_RESPONSE_V2 • Section 2.2.24.1, Oplock Break Acknowledgment • Section 2.2.36, SMB2 CHANGE_NOTIFY Response For details on the changes, see the PDF Diff file at https://winprotocoldoc.blob.core.windows.net/productionwindowsarchives/MS-SMB2/[MS-SMB2]-170615-diff.pdf.

*Date format: YYYY/MM/DD