[MS-FSA]: File System Algorithms

Errata below are for Protocol Document Version V34.0 – 2021/06/25.

Errata Published*	Description
2021/09/07	In Section 2.1.5.9.20 FSCTL_OFFLOAD_READ the alignment rules have been updated. Changed from: … ● Set InputBuffer.CopyLength to BlockAlignTruncate(InputBuffer.CopyLength, Open.File.Volume.LogicalBytesPerSector). … Changed to: … ● Set InputBuffer.CopyLength to BlockAlign(InputBuffer.CopyLength, Open.File.Volume.LogicalBytesPerSector). …
2021/08/17	In Section 2.1.1.1 Per Volume a new abstract data model element was added: ● IsHardLinksSupported: A Boolean that is TRUE if the physical media format for this volume supports HardLinks. <1> <1> Section 2.1.1.1: Hard links are supported on NTFS volumes, UDFS volumes, and ReFS volumes formatted version 3.5 or later (Windows Server 2022 and later). In Section 2.1.5.14.6 FileLinkInformation the pseudocode was updated for the IsHardLinksSupported abstract data model element. Changed from: … ● If Open.File.FileType is DirectoryFile, the operation MUST be failed with STATUS_FILE_IS_A_DIRECTORY. ● If Open.Link.IsDeleted is TRUE, the operation MUST be failed with STATUS_ACCESS_DENIED. … Changed to: … ● If Open.File.FileType is DirectoryFile, the operation MUST be failed with STATUS_FILE_IS_A_DIRECTORY.

Errata Published*	Description
	● If Open.File.Volume.IsHardLinksSupported is FALSE, the operation MUST be failed with STATUS_NOT_SUPPORTED. ● If Open.Link.IsDeleted is TRUE, the operation MUST be failed with STATUS_ACCESS_DENIED. … In Section 5 Appendix A: Product Behavior product behavior note 154, the description for the FILE_SUPPORTS_HARD_LINKS row with ReFS was added. Set if 3.5 format or higher volume (formatted using Windows Server 2022 or later)
2021/09/07	In Section 2.1.5.5.3.4 FileIdBothDirectoryInformation the following was updated. Changed from: … ● Entry.FileID set to Link.File.FileId64 … Changed to: … ● If Link.Name == "." (entry for the directory being queried): ● Entry.FileID set to Open.File.FileId64 ● Else if Link.Name == ".." (entry for the parent of the directory being queried): ● Entry.FileID SHOULD<65> be set to Open.Link.ParentFile.FileId64, otherwise MUST be set to zero ● Else: ● Entry.FileID set to Link.File.FileId64 ● EndIf <65> Section 2.1.5.5.3.4: The NTFS file system on versions prior to Windows 11 and Windows Server 2022, and non-NTFS file systems on all versions of Windows, always set the FileID field to zero in the ".." entry. … In MS-FSA section 2.1.5.5.3.5 FileIdExtdDirectoryInformation the following was updated. Changed from: … ● Entry.FileID set to Link.File.FileId128 … Changed to: … ● If Link.Name == "." (entry for the directory being queried): ● Entry.FileID set to Open.File.FileId128 ● Else if Link.Name == ".." (entry for the parent of the directory being queried): ● Entry.FileID SHOULD<67> be set to Open.Link.ParentFile.FileId128, otherwise MUST be set to zero

Errata Published*	Description
	• Else:
	• Entry.FileID set to Link.File.FileId128
	• EndIf
	<67> Section 2.1.5.5.3.5: The NTFS file system on versions prior to Windows 11 and Windows Server 2022, and non-NTFS file systems on all versions of Windows, always set the FileID field to zero in the ".." entry.
	...
	In MS-FSA section 2.1.5.5.3.6 FileIdFullDirectoryInformation the following was updated.
	Changed from:
	...
	• Entry.FileID set to Link.File.FileId64
	...
	Changed to:
	...
	• If Link.Name == "." (entry for the directory being queried):
	• Entry.FileID set to Open.File.FileId64
	• Else if Link.Name == ".." (entry for the parent of the directory being queried):
	• Entry.FileID SHOULD<69> be set to Open.Link.ParentFile.FileId64, otherwise MUST be set to zero
	• Else:
	• Entry.FileID set to Link.File.FileId64
	• EndIf
	<69> Section 2.1.5.5.3.6: The NTFS file system on versions prior to Windows 11 and Windows Server 2022, and non-NTFS file systems on all versions of Windows, always set the FileID field to zero in the ".." entry.
	...
2021/08/17	In Section 2.1.5.9.19 FSCTL_MARK_HANDLE, updated the conditions when STATUS_REDISENT_FILE_NOT_SUPPORTED is returned.
	Changed from:
	...
	Pseudocode for the operation is as follows:
	• If InputBufferSize is less than the size of the MARK_HANDLE_INFO structure, the operation MUST be failed with STATUS_BUFFER_TOO_SMALL.
	• If Open.Stream.StreamType == DirectoryStream, the operation MUST be failed with STATUS_DIRECTORY_NOT_SUPPORTED.<96>
	• If a file is resident the operation MUST be failed with STATUS_RESIDENT_FILE_NOT_SUPPORTED.<97>
	• STATUS_INVALID_PARAMETER is returned if:

Errata Published*	Description
	... Changed To: ... ● STATUS_INVALID_PARAMETER is returned if: ● InputBuffer.HandleInfo contains any flag other than one and only one of either MARK_HANDLE_READ_COPY or MARK_HANDLE_NOT_READ_COPY. ● Open.Mode.FILE_NO_INTERMEDIATE_BUFFERING was not specified at open time, meaning the file was opened for cached IO operations. ● If InputBuffer.CopyNumber > (Open.File.Volume.NumberOfDataCopies – 1). ● If Open.Stream.StreamType != DataStream. ● If InputBuffer.HandleInfo has MARK_HANDLE_READ_COPY set: ● If Open.File.Volume.NumberOfDataCopies < 2, the operation MUST be failed with STATUS_NOT_REDUNDANT_STORAGE. ● If Open.Stream.IsCompressed is TRUE, the operation MUST be failed with STATUS_COMPRESSED_FILE_NOT_SUPPORTED. ● If a file is resident the operation MUST be failed with STATUS_RESIDENT_FILE_NOT_SUPPORTED.<97> ● Set Open.ReadCopyNumber = InputBuffer.CopyNumber. ...

*Date format: YYYY/MM/DD