

[MS-FSA]: File System Algorithms

This topic lists the Errata found in the MS-FSA document 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.

Errata below are for Protocol Document Version [V21.0 – 2016/03/02](#).

| Errata Published* | Description |
|-------------------|--|
| 2016/06/27 | <p>In Section 2.1.5.1.2.1, Algorithm to Check Access to an Existing File, changed the pseudocode to clarify GrantedAccess.DELETE and STATUS_SHARING_VIOLATION during access check.</p> <p>Changed from:</p> <ul style="list-style-type: none">• If Open.SharingMode.FILE_SHARE_DELETE is FALSE and Open.GrantedAccess contains any one or more of (FILE_EXECUTE FILE_READ_DATA FILE_WRITE_DATA FILE_APPEND_DATA):• For each ExistingOpen in Open.File.OpenList:• If ExistingOpen.Mode.FILE_DELETE_ON_CLOSE is TRUE and (ExistingOpen.Stream.StreamType is DirectoryStream or ExistingOpen.Stream.Name is empty), then return STATUS_SHARING_VIOLATION.• EndFor• EndIf• If Open.GrantedAccess.DELETE is TRUE and (Open.Stream.StreamType is DirectoryStream or Open.Stream.Name is empty):• For each ExistingOpen in Open.File.OpenList:• If ExistingOpen.SharingMode.FILE_SHARE_DELETE is FALSE, then return STATUS_SHARING_VIOLATION.• EndFor• EndIf• Return STATUS_SUCCESS. <p>Changed to:</p> <ul style="list-style-type: none">• If Open.SharingMode.FILE_SHARE_DELETE is FALSE and Open.GrantedAccess contains any one or more of (FILE_EXECUTE FILE_READ_DATA FILE_WRITE_DATA FILE_APPEND_DATA DELETE):• For each ExistingOpen in Open.File.OpenList:• If ExistingOpen.GrantedAccess.DELETE is TRUE and (ExistingOpen.Stream.StreamType is DirectoryStream or ExistingOpen.Stream.Name is empty), then return STATUS_SHARING_VIOLATION.• EndFor• EndIf• If Open.GrantedAccess.DELETE is TRUE and (Open.Stream.StreamType is DirectoryStream or Open.Stream.Name is empty):• For each ExistingOpen in Open.File.OpenList: |

| Errata Published* | Description |
|-------------------|--|
| | <ul style="list-style-type: none"> • If ExistingOpen.SharingMode.FILE_SHARE_DELETE is FALSE and ExistingOpen.GrantedAccess contains one or more of (FILE_EXECUTE FILE_READ_DATA FILE_WRITE_DATA FILE_APPEND_DATA DELETE), then return STATUS_SHARING_VIOLATION. • EndFor • EndIf • Return STATUS_SUCCESS |
| 2016/06/27 | <p>In Section 2.1.5.1.2.2, Algorithm to Check Sharing Access to an Existing Stream or Directory, the following bullet points have been changed from:</p> <p>If ExistingOpen.SharingMode.FILE_SHARE_READ is FALSE and DesiredAccess contains either FILE_READ_DATA or FILE_EXECUTE If ExistingOpen.SharingMode.FILE_SHARE_WRITE is FALSE and DesiredAccess contains either FILE_WRITE_DATA or FILE_APPEND_DATA If ExistingOpen.SharingMode.FILE_SHARE_DELETE is FALSE and ExistingOpen contains DELETE</p> <p>Changed to:</p> <p>If ExistingOpen.SharingMode.FILE_SHARE_READ is FALSE and Open.GrantedAccess contains either FILE_READ_DATA or FILE_EXECUTE If ExistingOpen.SharingMode.FILE_SHARE_WRITE is FALSE and Open.GrantedAccess contains either FILE_WRITE_DATA or FILE_APPEND_DATA If ExistingOpen.SharingMode.FILE_SHARE_DELETE is FALSE and Open.GrantedAccess contains DELETE</p> |
| 2016/03/21 | <p>In Section 2.1.5.14.1, FileAllocationInformation, added text and a product behavior note to clarify that the FAT/FAT32/exFAT/UDFS file allocation behavior is different from NTFS.</p> <p>Changed from:</p> <ul style="list-style-type: none"> ▪ If NewAllocationSize is less than Open.Stream.Size: <ul style="list-style-type: none"> ▪ The object store MUST set Open.Stream.Size to NewAllocationSize, truncating the Stream. ▪ ... <p>Changed to:</p> <ul style="list-style-type: none"> ▪ If InputBuffer.AllocationSize is less than Open.Stream.Size: <ul style="list-style-type: none"> ▪ Set NewFileSize to min(Open.Stream.Size, NewAllocationSize<127>). ▪ If NewFileSize is less than Open.Stream.Size: <ul style="list-style-type: none"> ▪ The object store MUST set Open.Stream.Size to NewFileSize, truncating the stream. ▪ ... <p><127> Section 2.1.5.14.1: The FAT, FAT32, exFAT, and UDFS file systems instead set NewFileSize to min(Open.Stream.Size, InputBuffer.AllocationSize).</p> |
| 2016/03/21 | <p>In Section 2.1.5.14.11, FileRenameInformation, added Unicode Strings for the root path name and the destination full link name. Also, clarified the behavior for renaming file path names for files at the root directory and for files at a linked destination.</p> <p>Changed from:</p> |

| Errata Published* | Description |
|-------------------|---|
| | <p>This routine uses the following local variables:</p> <ul style="list-style-type: none"> ▪ Unicode strings: PathName, NewLinkName, PrevFullLinkName, SourceFullLinkName ▪ ... <p>Changed to:</p> <p>This routine uses the following local variables:</p> <ul style="list-style-type: none"> ▪ Unicode strings: PathName, RootPathName, NewLinkName, PrevFullLinkName, SourceFullLinkName, DestFullLinkName ▪ ... <p>Changed from:</p> <p>Pseudocode for the operation is as follows:</p> <p>...</p> <p>The operation MUST be failed with STATUS_INVALID_PARAMETER under any of the following conditions:</p> <p>...</p> <ul style="list-style-type: none"> ▪ Split InputBuffer.FileName into PathName and NewLinkName per section 2.1.5.1. ▪ If the first character of InputBuffer.FileName is '\\': <p>...</p> <p>Changed to:</p> <p>Pseudocode for the operation is as follows:</p> <p>...</p> <p>The operation MUST be failed with STATUS_INVALID_PARAMETER under any of the following conditions:</p> <p>...</p> <ul style="list-style-type: none"> ▪ If this operation is from a remote client, and either InputBuffer.RootDirectory is nonzero or the first character of InputBuffer.FileName is '\\. ▪ If InputBuffer.RootDirectory is nonzero and the first character of InputBuffer.FileName is '\\. ▪ If InputBuffer.RootDirectory is nonzero: <ul style="list-style-type: none"> ▪ The object store MUST set RootPathName to the full pathname from Open.File.Volume.RootDirectory to the file represented by InputBuffer.RootDirectory, in an implementation-specific manner. ▪ The object store MUST set DestFullLinkName to RootPathName + '\\ + InputBuffer.FileName. ▪ Else: <ul style="list-style-type: none"> ▪ The object store MUST set DestFullLinkName to InputBuffer.FileName. ▪ EndIf ▪ Split DestFullLinkName into PathName and NewLinkName per section 2.1.5.1. ▪ If the first character of InputBuffer.FileName is '\\ or InputBuffer.RootDirectory is nonzero or this operation is from a remote client: ▪ ... |
| 2016/03/21 | <p>In Section 2.1.5.4, Server Requests Closing an Open, corrected the processing rules for Phase1 (Delete on Close behavior).</p> <p>Changed from:</p> <ul style="list-style-type: none"> ▪ Phase 1 - Delete on Close: |

| Errata Published* | Description |
|----------------------|---|
| | <ul style="list-style-type: none"> ▪ If Open.Mode.FILE_DELETE_ON_CLOSE is TRUE: <ul style="list-style-type: none"> ▪ If Open.Stream.StreamType is DirectoryStream or Open.Stream.Name is empty: <ul style="list-style-type: none"> ▪ Open.Link.IsDeleted MUST be set to TRUE. ▪ Else: <ul style="list-style-type: none"> ▪ Open.Stream.IsDeleted MUST be set to TRUE. ▪ EndIf ▪ EndIf <p>Changed to:</p> <ul style="list-style-type: none"> ▪ Phase 1 - Delete on Close: ▪ If Open.Mode.FILE_DELETE_ON_CLOSE is TRUE: <ul style="list-style-type: none"> ▪ If Open.Stream.Name is empty: <ul style="list-style-type: none"> ▪ If (Open.Stream.StreamType is DataStream or Open.File.DirectoryList is empty), then Open.Link.IsDeleted MUST be set to TRUE. ▪ Else: <ul style="list-style-type: none"> ▪ Open.Stream.IsDeleted MUST be set to TRUE. ▪ EndIf ▪ EndIf |

*Date format: YYYY/MM/DD