

[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 [36.0 – 2022/04/29](#).

| Errata Published* | Description |
|-------------------|--|
| 2022/08/09 | <p>In section 2.1.5.15.11, FileRenameInformation, revised renaming processing.</p> <p>Changed from:</p> <ul style="list-style-type: none"> ▪ If <i>RemoveSourceLink</i> is TRUE: <ul style="list-style-type: none"> ▪ If Open.File.FileType is DirectoryFile <ul style="list-style-type: none"> ▪ <i>FilterMatch</i> = FILE_NOTIFY_CHANGE_DIR_NAME ▪ Else <ul style="list-style-type: none"> ▪ <i>FilterMatch</i> = FILE_NOTIFY_CHANGE_FILE_NAME ▪ EndIf ▪ If <i>MoveToNewDir</i> is TRUE or <i>AddTargetLink</i> is FALSE or <i>RemoveTargetLink</i> and <i>ExactCaseMatch</i> are TRUE: <i>Action</i> = FILE_ACTION_REMOVED ▪ Else <ul style="list-style-type: none"> ▪ <i>Action</i> = FILE_ACTION_REMOVED_OLD_NAME ▪ EndIf <p>Changed to:</p> <ul style="list-style-type: none"> ▪ If <i>RemoveSourceLink</i> is TRUE: <ul style="list-style-type: none"> ▪ If Open.File.FileType is DirectoryFile <ul style="list-style-type: none"> ▪ <i>FilterMatch</i> = FILE_NOTIFY_CHANGE_DIR_NAME ▪ Else <ul style="list-style-type: none"> ▪ <i>FilterMatch</i> = FILE_NOTIFY_CHANGE_FILE_NAME ▪ EndIf ▪ If <i>MoveToNewDir</i> is TRUE or <i>AddTargetLink</i> is FALSE or <i>RemoveTargetLink</i> and <i>ExactCaseMatch</i> are TRUE: <i>Action</i> = FILE_ACTION_REMOVED ▪ Else <ul style="list-style-type: none"> ▪ <i>Action</i> = FILE_ACTION_RENAMED_OLD_NAME ▪ EndIf |
| 2022/07/26 | <p>Added revisions to section 2.1.5.2, Server Requests an Open of a Named Pipe. Please see the diff file.</p> |
| 2022/06/01 | <p>Added new section, 2.1.5.2, Server Requests an Open of a Named Pipe. Please see the diff file.</p> |
| 2022/06/01 | <p>In section 2.1.5.15.11, FileRenameInformation, added information about how NTFS prevents a race condition during renaming.</p> <p>Changed from:</p> <p>If Open.File contains open files as specified in section 2.1.4.2, the operation MUST be failed with STATUS_ACCESS_DENIED.</p> <p>Changed to:</p> <p>If Open.File contains open files as specified in section 2.1.4.2, the operation MUST be failed with STATUS_ACCESS_DENIED.<174></p> <p><174> On Windows NTFS, NTFS checks for open files beneath the directory being renamed (performs section 2.1.4.2), it records the count of open files. If there is a lease to break, NTFS</p> |

| Errata Published* | Description | | | | | | |
|-------------------|---|-------|---------|------------|---------------------------------|------------|---|
| | <p>requests the break and then goes back to the start of performing 2.1.5.15.11. NTFS waits for the lease break acknowledgment and restarts the rename operation. When NTFS performs section 2.1.4.2 again, it again records how many open files there are beneath the directory and compares that to the previous count. If the current count is greater than or equal to the previous count, NTFS fails the rename and prevents a possible race condition.</p> | | | | | | |
| 2022/05/27 | <p>In section 2.1.5.10.34, FSCTL_SET_INTEGRITY_INFORMATION_EX, updated list of applicable updates.</p> <p>Changed from:</p> <p><127> Section 2.1.5.10.34: The FSCTL_SET_INTEGRITY_INFORMATION_EX operation is supported only by the ReFS file system v3.2 or higher (Windows 10 v1507 operating system or higher). FSCTL_SET_INTEGRITY_INFORMATION_EX is handled following the process in this section on systems updated with [MSKB-5014019], [MSKB-5014021], [MSKB-5014022], or [MSKB-5014023].</p> <p>Changed to:</p> <p><127> Section 2.1.5.10.34: The FSCTL_SET_INTEGRITY_INFORMATION_EX operation is supported only by the ReFS file system v3.2 or higher (Windows 10 v1507 operating system or higher). FSCTL_SET_INTEGRITY_INFORMATION_EX is handled following the process in this section on systems updated with [MSKB-5014019], [MSKB-5014021], [MSKB-5014022], [MSKB-5014023], [MSKB-5014701], [MSKB-5014702], or [MSKB-5014710].</p> | | | | | | |
| 2022/05/18 | <p>The following sections were changed. Please see the diff document for the details.</p> <p>In Section 2.1.1.3, Per File, updated a product behavior about how registry entries affect the handling of LastAccessTime:</p> <p>Changed from:</p> <p><17> Section 2.1.1.3: In Windows Vista and subsequent, LastAccessTime updates are disabled by default in the ReFS and NTFS file systems. It is only updated when the file is closed. This behavior is controlled by the following registry key: HKLM\System\CurrentControlSet\Control\FileSystem\NtfsDisableLastAccessUpdate. A nonzero value means LastAccessTime updates are disabled. A value of zero means they are enabled.</p> <p>Changed to:</p> <p><17> Section 2.1.1.3: In Windows Vista and subsequent, LastAccessTime updates are disabled by default in the ReFS and NTFS file systems. It is only updated when the file is closed. This behavior is controlled by the following registry values (respectively): HKLM\System\CurrentControlSet\Control\FileSystem\RefsDisableLastAccessUpdate, and HKLM\System\CurrentControlSet\Control\FileSystem\NtfsDisableLastAccessUpdate. A value of 1 means LastAccessTime updates are disabled. Any other value (or undefined) means they are enabled.</p> <p>In Windows 10 v1803 operating system and subsequent, NTFS has two registry values controlling LastAccessTime updates: HKLM\System\CurrentControlSet\Control\FileSystem\NtfsDisableLastAccessUpdate and HKLM\System\CurrentControlSet\Control\FileSystem\NtfsLastAccessUpdatePolicyVolumeSizeThreshold. The NtfsDisableLastAccessUpdate value is now treated as a bitfield as follows:</p> <table border="1" data-bbox="399 1551 1414 1843"> <thead> <tr> <th data-bbox="399 1551 737 1604">Value</th> <th data-bbox="737 1551 1414 1604">Meaning</th> </tr> </thead> <tbody> <tr> <td data-bbox="399 1604 737 1656">0x00000001</td> <td data-bbox="737 1604 1414 1656">Disable LastAccessTime updates.</td> </tr> <tr> <td data-bbox="399 1656 737 1843">0x00000002</td> <td data-bbox="737 1656 1414 1843"> System managed. Indicates that NTFS uses its own policy for updating LastAccessTime as follows: <ul style="list-style-type: none"> On client systems, LastAccessTime updates are enabled if any of the following conditions are true: <ul style="list-style-type: none"> • NtfsLastAccessUpdatePolicyVolumeSizeThreshold is 0. </td> </tr> </tbody> </table> | Value | Meaning | 0x00000001 | Disable LastAccessTime updates. | 0x00000002 | System managed. Indicates that NTFS uses its own policy for updating LastAccessTime as follows: <ul style="list-style-type: none"> On client systems, LastAccessTime updates are enabled if any of the following conditions are true: <ul style="list-style-type: none"> • NtfsLastAccessUpdatePolicyVolumeSizeThreshold is 0. |
| Value | Meaning | | | | | | |
| 0x00000001 | Disable LastAccessTime updates. | | | | | | |
| 0x00000002 | System managed. Indicates that NTFS uses its own policy for updating LastAccessTime as follows: <ul style="list-style-type: none"> On client systems, LastAccessTime updates are enabled if any of the following conditions are true: <ul style="list-style-type: none"> • NtfsLastAccessUpdatePolicyVolumeSizeThreshold is 0. | | | | | | |

| Errata Published* | Description | | | | | | | |
|--|--|--|-------|---------|------------|---------------------------------|------------|--|
| | | <ul style="list-style-type: none"> • The size of the boot volume is <= NtfsLastAccessUpdatePolicyVolumeSizeThreshold in GB. • NtfsLastAccessUpdatePolicyVolumeSizeThreshold is undefined and (prior to Windows 10 v2004) the size of the boot volume is <= 128GB. <p>On server systems, or client systems where the above conditions do not apply, LastAccessTime updates are always disabled.</p> <p>At system startup, after evaluating the above policy, NTFS will set/clear flag 0x00000001 accordingly to reflect that LastAccessTime updates are disabled/enabled.</p> | | | | | | |
| | 0x80000000 | <p>Flags initialized. Indicates NTFS recognizes flags other than 0x00000001. At system startup, if flag 0x80000000 is not set, the system will automatically set flag 0x80000000 and will additionally set flag 0x00000002 (becoming system managed) if flag 0x00000001 was set.</p> | | | | | | |
| <p>If the value of NtfsDisableLastAccessUpdate is controlled by group policy, then only flag 0x00000001 is recognized.</p> | | | | | | | | |
| <p>In Section 2.1.1.4, Per Link, updated a product behavior about how registry entries affect the handling of LastAccessTime:</p> | | | | | | | | |
| <p>Changed from:</p> | | | | | | | | |
| <p><31> Section 2.1.1.4: In Windows Vista and subsequent LastAccessTime updates are disabled by default in the ReFS and NTFS file systems. It is only updated when the file is closed. This behavior is controlled by the following registry key: HKLM\System\CurrentControlSet\Control\FileSystem\NtfsDisableLastAccessUpdate. A nonzero value means LastAccessTime updates are disabled. A value of zero means they are enabled.</p> | | | | | | | | |
| <p>Changed to:</p> | | | | | | | | |
| <p><31> Section 2.1.1.4: In Windows Vista and subsequent, LastAccessTime updates are disabled by default in the ReFS and NTFS file systems. It is updated only when the file is closed. This behavior is controlled by the following registry values (respectively): HKLM\System\CurrentControlSet\Control\FileSystem\RefsDisableLastAccessUpdate, and HKLM\System\CurrentControlSet\Control\FileSystem\NtfsDisableLastAccessUpdate. A value of 1 means LastAccessTime updates are disabled. Any other value (or undefined) means they are enabled.</p> | | | | | | | | |
| <p>In Windows 10 v1803 and subsequent, NTFS has two registry values controlling LastAccessTime updates: HKLM\System\CurrentControlSet\Control\FileSystem\NtfsDisableLastAccessUpdate and HKLM\System\CurrentControlSet\Control\FileSystem\NtfsLastAccessUpdatePolicyVolumeSizeThreshold. The NtfsDisableLastAccessUpdate value is now treated as a bitfield as follows:</p> | | | | | | | | |
| <table border="1"> <thead> <tr> <th data-bbox="399 1602 745 1654">Value</th> <th data-bbox="745 1602 1414 1654">Meaning</th> </tr> </thead> <tbody> <tr> <td data-bbox="399 1654 745 1703">0x00000001</td> <td data-bbox="745 1654 1414 1703">Disable LastAccessTime updates.</td> </tr> <tr> <td data-bbox="399 1703 745 1898">0x00000002</td> <td data-bbox="745 1703 1414 1898"> <p>System managed. Indicates that NTFS uses its own policy for updating LastAccessTime as follows:</p> <p>On client systems, LastAccessTime updates are enabled if any of the following conditions are true:</p> <ul style="list-style-type: none"> • NtfsLastAccessUpdatePolicyVolumeSizeThreshold is 0. </td> </tr> </tbody> </table> | | | Value | Meaning | 0x00000001 | Disable LastAccessTime updates. | 0x00000002 | <p>System managed. Indicates that NTFS uses its own policy for updating LastAccessTime as follows:</p> <p>On client systems, LastAccessTime updates are enabled if any of the following conditions are true:</p> <ul style="list-style-type: none"> • NtfsLastAccessUpdatePolicyVolumeSizeThreshold is 0. |
| Value | Meaning | | | | | | | |
| 0x00000001 | Disable LastAccessTime updates. | | | | | | | |
| 0x00000002 | <p>System managed. Indicates that NTFS uses its own policy for updating LastAccessTime as follows:</p> <p>On client systems, LastAccessTime updates are enabled if any of the following conditions are true:</p> <ul style="list-style-type: none"> • NtfsLastAccessUpdatePolicyVolumeSizeThreshold is 0. | | | | | | | |

| Errata Published* | Description | |
|-------------------|--|--|
| | | <ul style="list-style-type: none"> • The size of the boot volume is less than or equal to NtfsLastAccessUpdatePolicyVolumeSizeThreshold in GB. • NtfsLastAccessUpdatePolicyVolumeSizeThreshold is undefined and (prior to Windows 10 v2004) the size of the boot volume is <= 128GB. <p>On server systems, or client systems where the above conditions do not apply, LastAccessTime updates are always disabled.</p> <p>At system startup, after evaluating the above policy, NTFS will set/clear flag 0x00000001 accordingly to reflect that LastAccessTime updates are disabled/enabled.</p> |
| | 0x80000000 | <p>Flags initialized. Indicates NTFS recognizes flags other than 0x00000001. At system startup, if flag 0x80000000 is not set, the system will automatically set flag 0x80000000 and will additionally set flag 0x00000002 (becoming system managed) if flag 0x00000001 was set.</p> |
| | <p>If the value of NtfsDisableLastAccessUpdate is controlled by group policy, then only flag 0x00000001 is recognized.</p> | |
| 2022/05/02 | <p>In Section 2.1.5.9.34, FSCTL_SET_INTEGRITY_INFORMATION_EX, updated processing rules for system versions.</p> <p>Changed from: The server provides:<127></p> <p><127> Section 2.1.5.9.34: The FSCTL_SET_INTEGRITY_INFORMATION_EX operation is supported only by the ReFS file system v3.2 or higher (Windows 10 v1507 operating system or higher).</p> <p>Changed to: The server provides:<127></p> <p><127> Section 2.1.5.9.34: The FSCTL_SET_INTEGRITY_INFORMATION_EX operation is supported only by the ReFS file system v3.2 or higher (Windows 10 v1507 operating system or higher). FSCTL_SET_INTEGRITY_INFORMATION_EX is handled following the process in this section on systems updated with [MSKB-5014019], [MSKB-5014021], [MSKB-5014022], or [MSKB-5014023].</p> | |