

[MS-FSCC]: File System Control Codes

This topic lists the Errata found in the MS-FSCC 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 [V47.0 – 2020/03/04](#).

Errata Published*	Description
2020/04/27	<p>In Section 2.7.1, FILE_NOTIFY_INFORMATION, the following was changed from:</p> <p>FILE_ACTION_REMOVED</p> <p>0x00000002 The file was removed from the directory.</p> <p>FILE_ACTION_MODIFIED</p> <p>0x00000003 The file was modified. This can be a change to the data or attributes of the file.</p> <p>FILE_ACTION_RENAMED_OLD_NAME</p> <p>0x00000004 The file was renamed, and this is the old name. If the new name resides within the directory being monitored, the client also receives the FILE_ACTION_RENAMED_NEW_NAME bit value as described in the next list item. If the new name resides outside of the directory being monitored, the client will not receive the FILE_ACTION_RENAMED_NEW_NAME bit value.</p> <p>FILE_ACTION_RENAMED_NEW_NAME</p> <p>0x00000005 The file was renamed, and this is the new name. If the old name resides within the directory being monitored, the client will also receive the FILE_ACTION_RENAME_OLD_NAME bit value. If the old name resides outside of the directory being monitored, the client will not receive the FILE_ACTION_RENAME_OLD_NAME bit value.</p> <p>Changed to:</p> <p>FILE_ACTION_REMOVED</p> <p>0x00000002 The file was removed from the directory. When a file is renamed to a different directory the client will receive this notification along with FILE_ACTION_MODIFIED.</p> <p>FILE_ACTION_MODIFIED</p> <p>0x00000003 The file was modified. This can be a change to the data or attributes of the file. When a file is renamed to a different directory the client will receive this notification along with FILE_ACTION_REMOVED.</p> <p>FILE_ACTION_RENAMED_OLD_NAME</p>

Errata Published*	Description
	<p>0x00000004 The file was renamed, and this is the old name. This notification is only sent when the rename operation does not change the directory the file resides in. The client will also receive a FILE_ACTION_RENAMED_NEW_NAME notification. This notification will not be received if the file is renamed to a different directory.</p> <p>FILE_ACTION_RENAMED_NEW_NAME</p> <p>0x00000005 The file was renamed, and this is the new name. This notification is only sent when the rename operation does not change the directory the file resides in. The client will also receive a FILE_ACTION_RENAME_OLD_NAME notification. This notification will not be received if the file is renamed to a different directory.</p>
2020/04/27	<p>In Section 6, Appendix B: Product Behavior, the following was changed in behavior note <10> Section 2.1.9 from:</p> <p>NTFS computes the 64-bit file ID as follows: 48 bits are the index of the file's primary record in the master file table (MFT), and the other 16 bits are a sequence number. Therefore, it is possible that a different file can have the same 64-bit file ID as a file on that volume had in the past.</p> <p>Changed to</p> <p>NTFS computes the 64-bit file ID as follows: the low 48 bits are the index of the file's primary record in the master file table (MFT); the remaining 16 bits are a sequence number. Therefore, it is possible, though rare, that a different file can have the same 64-bit file ID as a file on that volume had in the past.</p> <p>ReFS maps a subset of the possible 128-bit file ID values to a 64-bit value using a reversible algorithm; for values outside of this subset, ReFS sets the 64-bit file ID to -1.</p> <p>The following was added to behavior note <11> in Section 2.1.10:</p> <p>NTFS computes the 128-bit file ID as follows: the low 48 bits are the index of the file's primary record in the master file table (MFT), the next 16 bits are a sequence number, and the high 64 bits MUST be zero. Therefore, it is possible, though rare, that a different file can have the same 128-bit file ID as a file on that volume had in the past.</p> <p>ReFS computes the 128-bit file ID as follows: the low 64 bits consists of an index uniquely identifying the file's parent directory on the volume. The high 64-bits consists of an index uniquely identifying the file within that directory.</p>

*Date format: YYYY/MM/DD