[MS-FRS2]: Distributed File System Replication Protocol

This topic lists the Errata found in the MS-FRS2 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.

RSS
Atom

Errata below are for Protocol Document Version V28.0 – 2018/09/12.

Errata Published*	Description
2019/02/19	In Section 1.2.1, Normative References, the following reference has been added: [MS-XCA] Microsoft Corporation, "Xpress Compression Algorithm". In Section 2.2.1.4.15, XPRESS Block, the Block Data field has been changed from: If the value of the Block Compressed Size field is less than the value of the Block Uncompressed Size field, then the data has been compressed. For more information about decompressing compressed data, see section 3.1.1.1.3.9. Changed to: If the value of the Block Compressed Size field is less than the value of the Block Uncompressed Size field, then the data has been compressed. For more information about decompressing compressed data, see section 3.1.1.2. In Section 3.1.1.1, Compression, the following was changed from: Many of the FrsTransport methods use compression to reduce the amount of data that is returned to the client. This section describes algorithms and a conceptual model of possible data organization that an implementation maintains in order to decompress compressed data. The described organization is provided to facilitate the explanation of how the algorithm behaves. Error checking and handling has been omitted from all algorithms in the interests of clarity. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with what is described in this document. Changed to: Many of the FrsTransport methods use the LZ77+Huffman Compression algorithm, specified in [MS-XCA] section 2.1, to compress data. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with what is described in this document. The following sections have been removed and replaced with links to MS-XCA: 3.1.1.1.1 Pseudocode Conventions 3.1.1.1.2 Data Structures 3.1.1.1.2.1 PREFIX_CODE_NODE 3.1.1.1.2.2 PREFIX_CODE_SYMBOL 3.1.1.1.2.3 BITSTRING

Errata Published*	Description
	3.1.1.1.3 Procedures 3.1.1.1.3.1 PrefixCodeTreeRebuild 3.1.1.1.3.2 PrefixCodeTreeAddLeaf 3.1.1.1.3.3 SortSymbols 3.1.1.1.3.4 CompareSymbols 3.1.1.1.3.5 BitstringInit 3.1.1.1.3.6 BitstringLookup 3.1.1.1.3.7 BitstreamSkip 3.1.1.1.3.8 PrefixCodeTreeDecodeSymbol A new section, 3.1.1.2, Decompression, has been added: FrsTransport methods that compress data will always return information specifying the size of the original data. It is the caller's responsibility to determine whether the returned data is compressed. If the size of the compressed data buffer that is returned by the server in bytes is equal to the size in bytes of the original uncompressed data, then the buffer returned by the server contains uncompressed data. In Section 3.2.4.1.7, RequestRecords (Opnum 6), the description of the compressedRecords field has been changed from: compressedRecords: The data records, compressed using the DFS-R compression algorithm specified in section 3.1.1.1. The compressedRecords bytes correspond to an array of FRS_ID_GVSN entries. DFS-R uses custom marshaling in this RPC call to compress the set of transmitted records. The size of the FRS_ID_GVSN array is given by the numRecords parameter. The decompression algorithm specified in section 3.1.1.1.3.9 can be used to decompress the received data into a buffer of sizeof(FRS_ID_GVSN)*numRecords bytes, which can be re-interpreted as an array of FRS_ID_GVSN entries. Changed to: compressedRecords: The data records, compressed using the algorithm specified in section 3.1.1.1. The compressedRecords bytes correspond to an array of FRS_ID_GVSN entries. DFS-R uses custom marshaling in this RPC call to compress the set of transmitted records. The size of the FRS_ID_GVSN array is given by the numRecords parameter. The decompression algorithm specified in section 3.1.1.1 can be used to decompress the received data into a buffer of sizeof(FRS_ID_GVSN)*numRecords bytes, which can be re-interpreted as an array of FRS_ID_GVSN entries. In Section 3.2.4.1.14, InitializeFileTransferAsync (Opnum 13), changed from: 2. An encapsulation of the marshaled file data stream using the compressed data format (as specified in section 3.2.4.1.14.2) generated by the DFS-R compression algorithm specified in section 3.1.1.1. Even if the marshaled file data stream is not compressed by the server, it is still encapsulated using the compressed data format. Changed to: 2. An encapsulation of the marshaled file data stream using the compressed data format (as specified in section 3.2.4.1.14.2) generated by the compression algorithm

Errata Published*	Description
	specified in section 3.1.1.1. Even if the marshaled file data stream is not compressed by the server, it is still encapsulated using the compressed data format.

*Date format: YYYY/MM/DD