[MS-FSSHTTPD]:

Binary Data Format for File Synchronization via SOAP

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **Copyrights**. This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
- **Patents**. Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **License Programs**. To see all of the protocols in scope under a specific license program and the associated patents, visit the Patent Map.
- **Trademarks**. The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names**. The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than as specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications documentation does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments, you are free to take advantage of them. Certain Open Specifications documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Support. For questions and support, please contact dochelp@microsoft.com.

Revision Summary

Date	Revision History	Revision Class	Comments
3/28/2011	0.1	New	Released new document.
6/10/2011	1.0	Major	Significantly changed the technical content.
1/20/2012	2.0	Major	Significantly changed the technical content.
4/11/2012	2.0	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	2.1	Minor	Clarified the meaning of the technical content.
9/12/2012	2.1	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	2.1	None	No changes to the meaning, language, or formatting of the technical content.
2/11/2013	2.1	None	No changes to the meaning, language, or formatting of the technical content.
7/30/2013	2.2	Minor	Clarified the meaning of the technical content.
11/18/2013	2.2	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	2.2	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	2.3	Minor	Clarified the meaning of the technical content.
7/31/2014	2.4	Minor	Clarified the meaning of the technical content.
10/30/2014	3.0	Major	Significantly changed the technical content.
3/16/2015	4.0	Major	Significantly changed the technical content.
6/30/2015	5.0	Major	Significantly changed the technical content.
2/26/2016	6.0	Major	Significantly changed the technical content.
4/14/2016	7.0	Major	Significantly changed the technical content.
7/15/2016	7.0	None	No changes to the meaning, language, or formatting of the technical content.
9/14/2016	7.0	None	No changes to the meaning, language, or formatting of the technical content.
6/20/2017	8.0	Major	Significantly changed the technical content.
7/24/2018	9.0	Major	Significantly changed the technical content.
10/1/2018	10.0	Major	Significantly changed the technical content.
12/11/2018	11.0	Major	Significantly changed the technical content.
4/22/2021	12.0	Major	Significantly changed the technical content.
6/25/2021	13.0	Major	Significantly changed the technical content.

Date	Revision History	Revision Class	Comments
7/20/2021	14.0	Major	Significantly changed the technical content.
8/17/2021	15.0	Major	Significantly changed the technical content.
10/5/2021	15.0	None	No changes to the meaning, language, or formatting of the technical content.
11/15/2022	15.1	Minor	Clarified the meaning of the technical content.
2/21/2023	15.2	Minor	Clarified the meaning of the technical content.
5/16/2023	15.3	Minor	Clarified the meaning of the technical content.
8/15/2023	15.4	Minor	Clarified the meaning of the technical content.
11/13/2023	15.5	Minor	Clarified the meaning of the technical content.
2/20/2024	15.6	Minor	Clarified the meaning of the technical content.
4/16/2024	16.0	Major	Significantly changed the technical content.
5/21/2024	16.1	Minor	Clarified the meaning of the technical content.

Table of Contents

1 Introduction

The Binary Data Format for File Synchronization via SOAP provides a schema for representing traditional file data (a stream of bytes) efficiently in the storage model described in [MS-FSSHTTP].

Sections 1.7 and 2 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

globally unique identifier (GUID): A term used interchangeably with universally unique identifier (UUID) in Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] have to be used for generating the GUID. See also universally unique identifier (UUID).

little-endian: Multiple-byte values that are byte-ordered with the least significant byte stored in the memory location with the lowest address.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as defined in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

[MS-FSSHTTPB] Microsoft Corporation, "Binary Requests for File Synchronization via SOAP Protocol".

[MS-FSSHTTP] Microsoft Corporation, "File Synchronization via SOAP over HTTP Protocol".

[MS-RDC] Microsoft Corporation, "Remote Differential Compression Algorithm".

[PKWARE-Zip] PKWARE Inc., ".Zip File Format Specification", 2006, http://www.pkware.com/documents/APPNOTE/APPNOTE-6.3.0.TXT

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, https://www.rfc-editor.org/info/rfc2119

1.2.2 Informative References

None.

1.3 Overview

1.3.1 Schema Overview

This structure describes a schema for representing traditional file data (a stream of bytes) efficiently in the storage model described in [MS-FSSHTTP]. It can be used when no native model, as described in [MS-FSSHTTP], exists for the data, yet the benefits of incremental sync and storage, as described in [MS-FSSHTTP], are required.

Because the schema presumes no knowledge of the semantics of the file data, all data needs to be represented within a single cell, as described in [MS-FSSHTTPB] section 3.1.1, to maintain file consistency. This promotes all concurrent edits to full file conflicts and precludes incremental load. It does, however, offer simplicity and, because of object usage, incremental sync and storage.

File data is split into chunks. These chunks are mapped to objects in the model of this protocol and therefore need to align as closely as possible with the expected change profile of the data. Any expected edit needs to intersect as closely as possible with a whole number of chunks because the entire chunk needs to be stored and synchronized after an edit.

The file data is represented in a hierarchical tree of nodes; each node is an object described in [MS-FSSHTTP]. Each node represents a sequential portion of the file stream, the position of which is inferred by adding the lengths of the preceding sibling node's lengths. The node can either directly contain the region's stream data or be the root of a subtree that further divides the region.

Chunking schemes are defined for files specified in the Zip file format, as described in [PKWARE-Zip], and for files for which no structure is available using the RDC FilterMax algorithm, as described in [MS-RDC] section 3.1.5.1.

1.3.2 Byte Ordering

All data and structures in this document are assumed to be in **little-endian** format.

1.4 Relationship to Protocols and Other Structures

This protocol is embedded with the protocol described in [MS-FSSHTTPB].

1.5 Applicability Statement

This protocol is intended for use where incremental updates and the efficient transmission of file data are desired features of client/server file synchronization. This protocol is designed for use with file formats that are not natively represented in the protocol described in [MS-FSSHTTPB].

1.6 Versioning and Localization

None.

1.7 Vendor-Extensible Fields

None.

2 Structures

2.1 Transport

This protocol uses the protocol specified in [MS-FSSHTTPB].

2.2 Object Definitions

An **Intermediate Node Object**, **Leaf Node Object**, and **Data Node Object** use Objects, as specified in [MS-FSSHTTPB] section 3.1.1.

2.2.1 Common Node Object Properties

A **Node Object** is contained within an object group data element, as specified in [MS-FSSHTTPB] section 2.2.1.12.6. The **Object Group Object Data** field MUST be set as shown in the following table, as specified in [MS-FSSHTTPB] section 2.2.1.12.6.4

Field	Root	Intermediate	Data
Object Group Object Data	As specified in [MS-FSSHTTPB] section 2.2.1.12.6.4.	As specified in in [MS-FSSHTTPB] section 2.2.1.12.6.4.	As specified in in [MS-FSSHTTPB] section 2.2.1.12.6.4.
Object Extended GUID Array	Specifies an ordered list of the Object Extended GUIDs for each child of the root Intermediate Node. Object Extended GUID Array entries MUST be ordered based on the sequential file bytes represented by each Node Object.	Specifies an ordered list of the Object Extended GUIDs for each child of this node. Object Extended GUID Array entries MUST be ordered based on the sequential file bytes represented by each Node Object.	Specifies an empty list of Object Extended GUIDs.
Cell ID Array	Specifies an empty list of Cell IDs.	Specifies an empty list of Cell IDs.	Specifies an empty list of Cell IDs.
Data	As specified in section 2.2.2.	As specified in section 2.2.3.	As specified in section 2.2.4.

The **Object Declaration** field of the object group, as specified in [MS-FSSHTTPB] section 2.2.1.12.6.1, MUST be set as shown in the following table.

Field	Root	Intermediate	Data
Object Group Object Declaration	As specified in in [MS-FSSHTTPB].	As specified in in [MS-FSSHTTPB].	As specified in in [MS-FSSHTTPB].
Object Extended GUID	An extended **GUID**, as specified in [MS-FSSHTTPB] section 2.2.1.7, that specifies an identifier for this object. This GUID MUST be unique within this file.	An extended GUID that specifies an identifier for this object. This GUID MUST be unique within this file.	An extended GUID that specifies an identifier for this object. This GUID MUST be unique within this file.
Object Partition ID	A compact unsigned 64-bit integer that MUST be	A compact unsigned 64-bit integer that MUST be	A compact unsigned 64-bit integer that MUST be

Field	Root	Intermediate	Data
	"1".	"1".	"1".
Object Data Size	A compact unsigned 64-bit integer that MUST be the size of the **Object Data** field.	A compact unsigned 64-bit integer that MUST be the size of the **Object Data** field.	A compact unsigned 64-bit integer that MUST be the size of the **Object Data** field.
Object References Count	A compact unsigned 64-bit integer that specifies the number of object references.	A compact unsigned 64-bit integer that specifies the number of object references.	A compact unsigned 64-bit integer that specifies the number of object references.
Cell References Count	A compact unsigned 64-bit integer that MUST be zero.	A compact unsigned 64-bit integer that MUST be zero.	A compact unsigned 64-bit integer that MUST be zero.

All fields in the parent structure that are not specified in the preceding table are specified in [MS-FSSHTTPB].

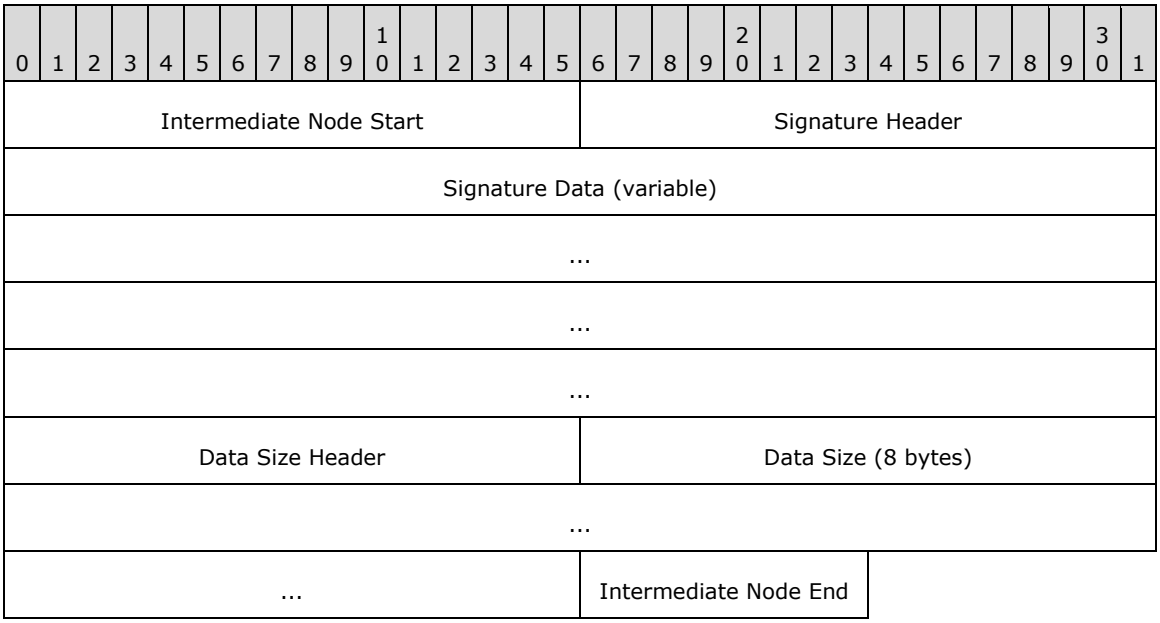
2.2.2 Intermediate Node Object

The root of the directed graph of Objects contained in a cell as specified in [MS-FSSHTTPB] section 3.1.1 is an **Intermediate Node Object.** File data is represented as a hierarchical tree of nodes contained in this graph schema. The root node is the root of this tree and where any traversal begins.<1>

An **Intermediate Node Object** has the format that is specified in the following subsections for the **Intermediate Node Object** data, **Intermediate Node Object** references, and **Intermediate Node Object** cell references.

2.2.2.1 Intermediate Node Object Data

The **Data** field for the **Intermediate Node Object** is specified in the following diagram.



Intermediate Node Start (2 bytes): A 16-bit stream object header, as specified in [MS-FSSHTTPB] section 2.2.1.5.1, with a **Header Type** of 0x00, **Compound** of 0x1, **Type** of 0x20, and **Length** of 0x00. The value of this field MUST be 0x0104.

Signature Header (2 bytes): A 16-bit stream object header, as specified in [MS-FSSHTTPB] section 2.2.1.5.1, with a **Header Type** of 0x00, **Compound** of 0x0, **Type** of 0x21, and **Length** equal to the size of **Signature Data**.

Signature Data (variable): A binary item, as specified in [MS-FSSHTTPB] section 2.2.1.3, specifying a value that is unique to the file data represented by this **Intermediate Node Object**. The value of this item depends on the file chunking algorithm used, as specified in section 2.4. If the length of this binary item is 0 this item SHOULD be ignored.

Data Size Header (2 bytes): A 16-bit stream object header, as specified in [MS-FSSHTTPB] section 2.2.1.5.1, with a **Header Type** of 0x00, **Compound** of 0x0, **Type** of 0x22, and **Length** of 0x08 (the size, in bytes, of **Data Size).** The value of this field MUST be 0x1110.

Data Size (8 bytes): An unsigned 64-bit integer that specifies the size of the file data represented by this **Intermediate Node Object**.

Intermediate Node End (1 byte): An 8-bit stream object header end, as specified in [MS-FSSHTTPB] section 2.2.1.5.3, that specifies a stream object of type 0x20. The value of this field MUST be 0x81.

2.2.2.2 Intermediate Node Object References

The **Object Extended GUID Array**, as specified in [MS-FSSHTTPB] section 2.2.1.12.6.4, of the **Intermediate Node Object** MUST specify an ordered set of **Object Extended GUIDs**. Each **Object Extended GUID** MUST specify a **Leaf Node Object** or another **Intermediate Node Object**. **Object Extended GUID Array** entries MUST be ordered based on the sequential file bytes represented by each descendent **Leaf Node Object**. The sum of the **Data Size** values from all of the **Leaf Node Objects** MUST equal the **Data Size** specified in the **Object Data** of this **Intermediate Node Object**.

2.2.2.3 Intermediate Node Object Cell References

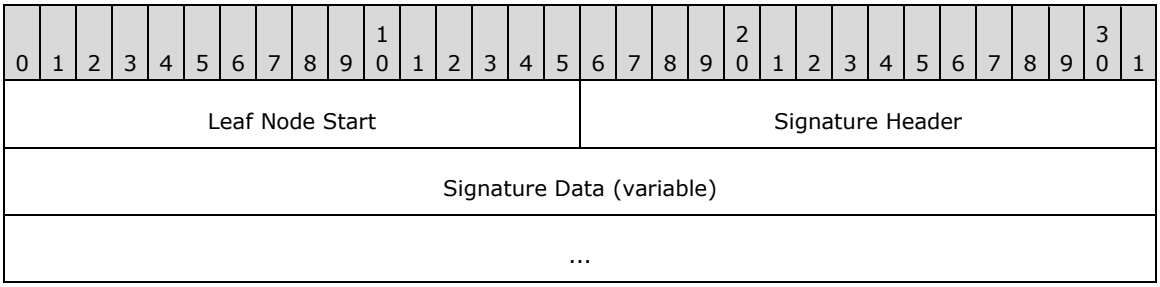
The **Cell ID Array**, as specified in [MS-FSSHTTPB] section 2.2.1.12.6.4, of the **Intermediate Node Object** MUST specify an empty array.

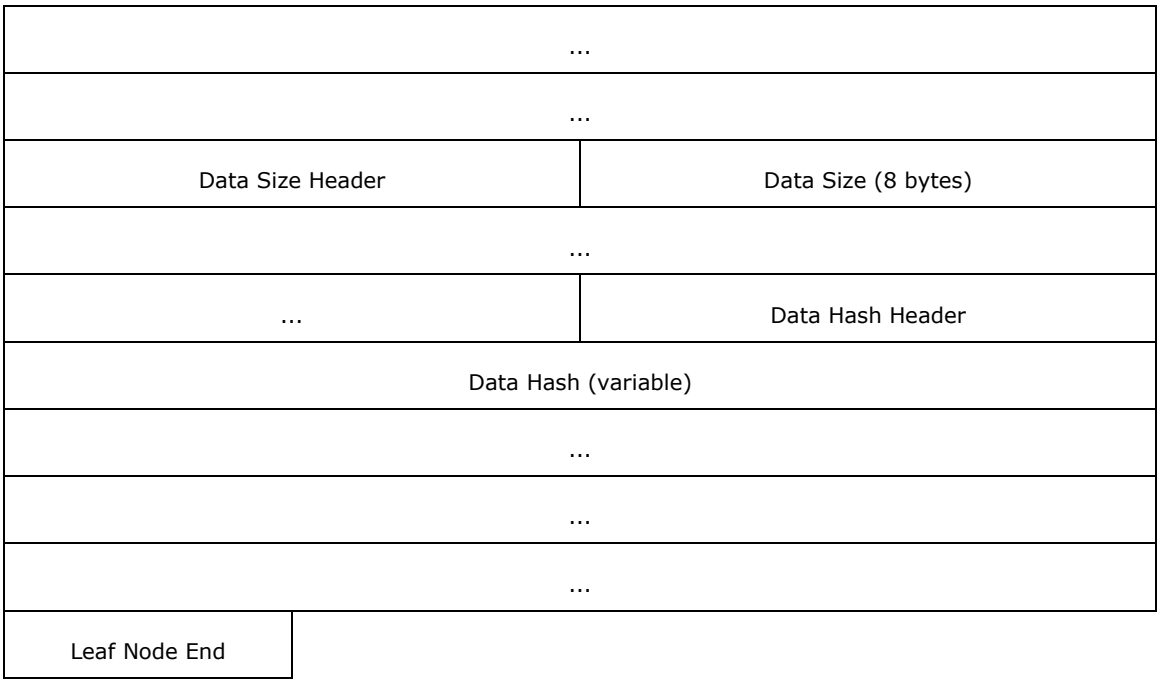
2.2.3 Leaf Node Object

A **Leaf Node Object** has the format that is specified in the following subsections for the object data, object references, and object cell references.

2.2.3.1 Leaf Node Object Data

The **Data** field for the **Leaf Node Object** is specified in the following diagram.





Leaf Node Start (2 bytes): A 16-bit stream object header, as specified in [MS-FSSHTTPB] section 2.2.1.5.1, with a **Header Type** of 0x00, **Compound** of 0x1, **Type** of 0x1F, and **Length** of 0x00. The value of this field MUST be 0x00FC.

Signature Header (2 bytes): A 16-bit stream object header, as specified in [MS-FSSHTTPB] section 2.2.1.5.1, with a **Header Type** of 0x00, **Compound** of 0x0, **Type** of 0x21, and **Length** equal to the size of **Signature Data**.

Signature Data (variable): A binary item, as specified in [MS-FSSHTTPB] section 2.2.1.3, specifying a value that is unique to the file data represented by this **Leaf Node Object**. The value of this item depends on the file chunking algorithm used, as specified in section 2.4.

Data Size Header (2 bytes): A 16-bit stream object header, as specified in [MS-FSSHTTPB] section 2.2.1.5.1, with a **Header Type** of 0x00, **Compound** of 0x0, **Type** of 0x22, and **Length** of 0x08 (the size, in bytes, of **Data Size).** The value of this field MUST be 0x1110.

Data Size (8 bytes): An unsigned 64-bit **integer** that specifies the size of the file data represented by this **Leaf Node Object**.

Data Hash Header (2 bytes, optional): An optional 16-bit stream object header, with a **Header Type** of 0x00, **Compound** of 0x0, **Type** of 0x2F, and **Length** equal to the size of **Data Hash**.<2>

Data Hash (variable): A binary item, as specified in [MS-FSSHTTPB] section 2.2.1.3, that specifies the data hash. The complete hash algorithm is specified in <3>.

Leaf Node End (1 byte): An 8-bit stream object header end, as specified in [MS-FSSHTTPB] section 2.2.1.5.3, that specifies a stream object of type 0x1F. The value of this field MUST be 0x7D.

2.2.3.2 Leaf Node Object References

The **Object Extended GUID Array**, as specified in [MS-FSSHTTPB] section 2.2.1.12.6.4, of the **Leaf Node Object** MUST specify an ordered set of **Object Extended GUIDs**. The ordered set of **Object Extended GUIDs** MUST contain the **Object Extended GUID** of a single **Data Node Object** or an ordered list of **Extended GUIDs** for the **Leaf Node Object**. **Object Extended GUID Array** entries

MUST be ordered based on the sequential file bytes represented by each **Node Object**. The size of the **Data Node Object** or the sum of the **Data Size** values from all of the **Leaf Node Objects** MUST equal the **Data Size** specified in the **Object Data** of this **Leaf Node Object**.

2.2.3.3 Leaf Node Object Cell References

The **Cell Reference Array** of the **Object** MUST specify an empty array.

2.2.4 Data Node Object

A **Data Node Object** has the format that is described in the following subsections for the object data, object references, and object cell references.

2.2.4.1 Data Node Object Data

Binary data that specifies the contents of the chunk of the file that is represented by this **Data Node**, as specified by the file chunking algorithm in section 2.4.

2.2.4.2 Data Node Object References

The **Object Extended GUID Array**, as specified in [MS-FSSHTTPB] section 2.2.1.12.6.4, of the **Data Node Object** MUST specify an empty array.

2.2.4.3 Data Node Object Cell References

The **Object Extended GUID Array**, as specified in [MS-FSSHTTPB] section 2.2.1.12.6.4, of the **Data Node Object** MUST specify an empty array.

2.3 Cell Properties

The **storage manifest data** element, as specified in [MS-FSSHTTPB] section 2.2.1.12.3, MUST have the **Storage Manifest Schema GUID** field set to 0EB93394-571D-41E9-AAD3-880D92D31955.

The **storage manifest data** element, as specified in [MS-FSSHTTPB] section 2.2.1.12.3, MUST have the **Cell ID** field set to extended GUID 5-bit **Uint** values, as specified in [MS-FSSHTTPB] section 2.2.1.7.2 and as listed in the following table.

Type	Value	GUID
4	1	84DEFAB9-AAA3-4A0D-A3A8-520C77AC7073
4	1	6F2A4665-42C8-46C7-BAB4-E28FDCE1E32B

The **storage manifest data element**, as specified in [MS-FSSHTTPB] section 2.2.1.12.3, MUST have the **Root Extended GUID** field set to an extended GUID 5-bit **Uint** value, as specified in [MS-FSSHTTPB] section 2.2.1.7.2 and as shown in the following table.

Type	Value	GUID
4	2	84DEFAB9-AAA3-4A0D-A3A8-520C77AC7073

The **revision manifest data** element, as specified in [MS-FSSHTTPB] section 2.2.1.12.5, MUST have a **Root Extended GUID** field set to an extended GUID 5-bit **Uint** value that represents the primary content stream, as shown in the following table.

Type	Value	GUID
4	2	84DEFAB9-AAA3-4A0D-A3A8-520C77AC7073

If other streams are present, additional roots MUST be specified by the protocol client and are opaque to this protocol. For each stream, a single **Intermediate Node** MUST be specified by using a unique root identifier.

2.4 File Chunking

A file chunk represents a range of data within a file. File chunking produces a list of chunks that are sequential and adjacent and that reference the entire contents of the file.

Each chunk contains unique **Signature Data**, as specified in section 2.2.3.1.

File data is passed to the following chunking methods:

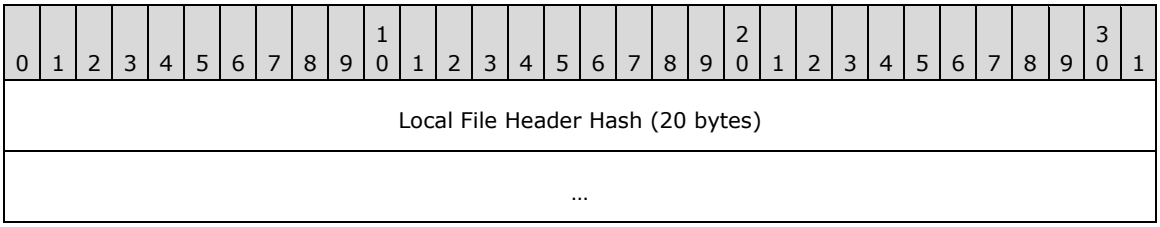
1. .ZIP analysis
2. RDC analysis
3. Simple chunking method

2.4.1 Zip Files

If the first 4 bytes of the file match the **local file header** signature, as specified in [PKWARE-Zip] section V, subsection A, then .ZIP analysis is used if the file is a valid ZIP file as specified in [PKWARE-Zip]. If the file does not comply with the ZIP format, RDC analysis as specified in section 2.4.2 or the simple chunking method as specified in section 2.4.3 is used.

.ZIP files are split into chunks based on information in each **local file header**, as specified in [PKWARE-Zip]. The analysis of **local file headers** produces file chunk boundaries at the start of the **local file header** as specified in [PKWARE-Zip], the start of the **data file** as specified in [PKWARE-Zip], and after the **data file** as specified in [PKWARE-Zip], producing two file chunks for each .ZIP item: the **local file header** chunk and the **data file** chunk.

The signature for the **local file header** chunk has the structure that is specified in the following diagram.<4>



Local File Header Hash: A 20-byte sequence that specifies the SHA-1 hash code of the file bytes represented by the local file header chunk.

The signature for the **data file** chunk has the structure that is specified in the following diagram.<5>

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
CRC																															
Compressed Size (8 bytes)																															
…																															
Uncompressed Size (8 bytes)																															
…																															

CRC (4 bytes): An unsigned 32-bit integer that specifies the value of the **local file header crc-32** field, as specified in [PKWARE-Zip].<6>

Compressed Size (8 bytes): An unsigned 64-bit integer that specifies the size, in bytes, of the **data file** chunk. It MUST be the value of the **local file header compressed size** field, as specified in [PKWARE-Zip], unless the **local file header extra field**, as specified in [PKWARE-Zip], includes a **Zip64 Extended Information Extra Field**, as specified in [PKWARE-Zip], in which case it MUST be the value of the **compressed size** field in the **Zip64 Extended Information Extra Field**, as specified in [PKWARE-Zip].

Uncompressed Size (8 bytes): An unsigned 64-bit integer that specifies the size, in bytes, of the uncompressed data represented by the bytes of the **data file** chunk. It MUST be the value of the **local file header uncompressed size** field, as specified in [PKWARE-Zip], unless the **local file header extra field**, as specified in [PKWARE-Zip], includes a **Zip64 Extended Information Extra Field**, as specified in [PKWARE-Zip], in which case it MUST be the value of the **uncompressed size** field in the **Zip64 Extended Information Extra Field**, as specified in [PKWARE-Zip].

If the combined size, in bytes, of the **local file header** chunk and the **data file** chunk is less than or equal to 4,096, a single chunk is produced with a signature that is the **local file header** chunk signature followed by the **data file** chunk signature. For protocol clients and servers with **VersionNumberType**, as specified in [MS-FSSHTTP] section 2.2.5.13, greater than or equal to 2 and **MinorVersionNumberType**, as specified in [MS-FSSHTTP] section 2.2.5.10, greater than or equal to 2, the signature for the single chunk is a bitwise exclusive OR of the signature bytes of the **local file header** chunk and **the data file** chunk. If the signatures are not of equal length, the extra bytes of the longer signature are appended to the end of the exclusive ORed bytes.

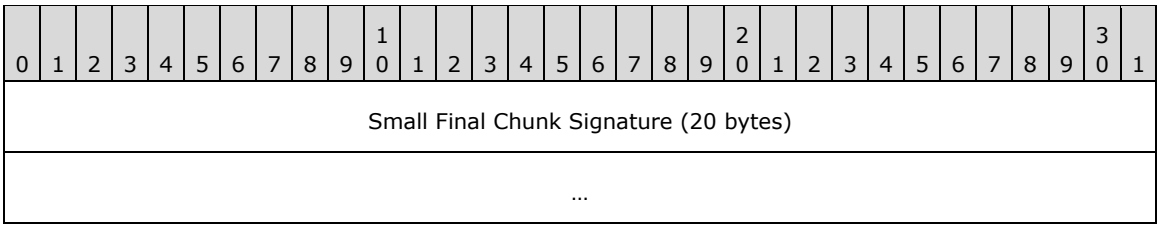
The analysis of chunks into **local file header** and **data file** chunks continues at the file location after the current data file until one of the following conditions occurs:

- The extent of the data file, as specified in the **local file header**, would extend past the end of the file.
- A sequence other than a **local file header** signature, as specified in [PKWARE-Zip], is found.

If the analysis of ZIP local headers terminates without creating any chunks, the .ZIP analysis MUST NOT be used.

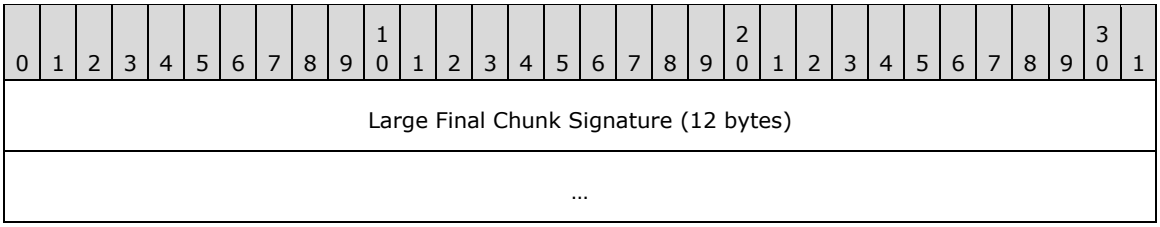
After the analysis of **local file headers** terminates, the remaining bytes in the file are represented by a final chunk.

If the total size, in bytes, of the final chunk is less than or equal to 1 megabyte, the signature for the final chunk has the structure that is shown in the following diagram.



Small Final Chunk Signature: A 20-byte sequence that specifies the SHA-1 hash code of the file bytes represented by the final chunk.

If the total size, in bytes, of the final chunk is greater than 1 megabyte, the signature for the final chunk has the structure that is shown in the following diagram.

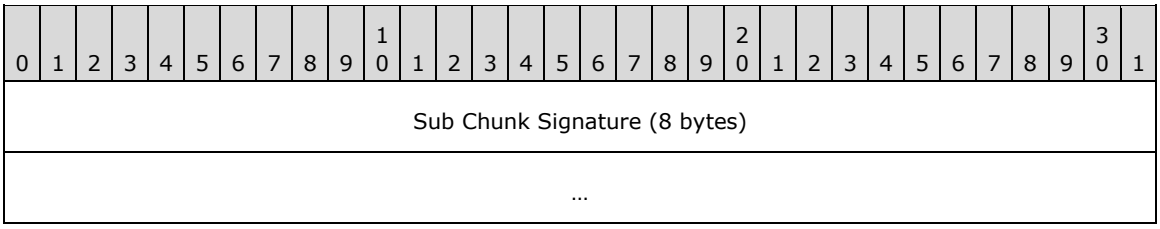


Large Final Chunk Signature: A 12-byte sequence of bytes that specifies the chunk signature.<7> This sequence of bytes MUST be unique.

For each chunk in the chunk list, a **Leaf Node Object**, as specified in section 2.2.3, is created. The **Data Size** of the **Leaf Node Object** MUST be the total number of bytes represented by the chunk. The **Signature Data** of the **Leaf Node Object** MUST be the chunk's signature. The **Leaf Node** is referenced by its parent node.

If the number of .ZIP file bytes represented by a chunk is greater than 3 megabytes, a list of subchunks is generated. Each subchunk represents a sequential chunk of the .ZIP file data. The size of each subchunk is at most 3 megabytes. All but the last subchunk SHOULD be 3 megabytes in size<8>. The total size of all the subchunks MUST equal the **Data Size** of the parent **Intermediate Node Object.**

The signature for these subchunks has the structure that is shown in the following diagram.



Sub Chunk Signature: An 8-byte sequence of bytes that specifies the subchunk signature.<9> This sequence of bytes MUST be unique.

For each subchunk, a **Leaf Node Object**, as specified in section 2.2.3, is created. The parent **Intermediate Node Object** of the subchunk MUST have its **Object Reference Array** include one **Object ID** entry for each subchunk, and these **Object ID** entries MUST be ordered based on the sequential .ZIP file bytes represented by each chunk.

For every **Leaf Node Object** that has a **Data Size** less than or equal to 1 megabyte, a **Data Node Object** MUST be created. The **Object Data** of the **Data Node Object** MUST be the byte sequence from the .ZIP file tracked by the chunk. The **Object Reference Array** and the **Cell Reference Array** of the **Data Node Object** MUST be empty. The **Object References Array** of the **Leaf Node Object**

associated with this **Data Node Object** MUST have a single entry, which MUST be the **Object ID** of the **Data Node Object**.

2.4.2 RDC Analysis

RDC analysis is performed on files for which the first 4 bytes of the file do not match the **local file header** signature, as specified in [PKWARE-Zip]. RDC analysis is performed only if the file size is less than 104,857,600 bytes.<10>

2.4.2.1 Generating Chunks

Files are split into chunks by using the RDC FilterMax algorithm, as specified in [MS-RDC] section 3.1.5.1, using a hash window of 48 and a horizon of 16,384, with the following exceptions:

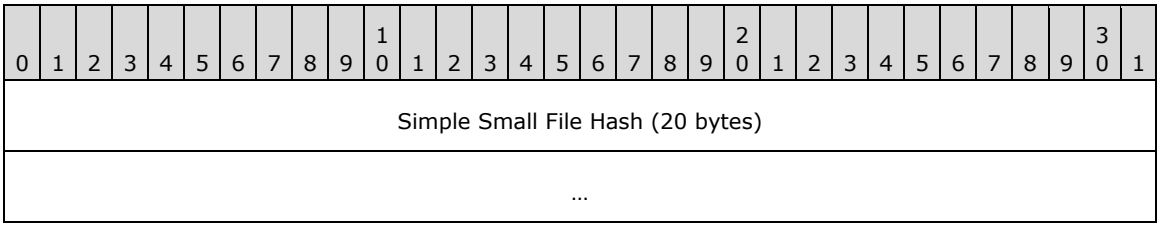
- Given a file F of length n consisting of bytes b_0 .. b_{n-1}, byte b_i is a local maximum only if i > horizon and for all j ≠ i where i − horizon < j < i + horizon, the H3 hash value after adding b_i is greater than that after adding any b_j.
- Local maximums found within the last (n modulus 16,384) + 16,384 bytes of data MUST be ignored and not treated as chunk boundaries.
- If n is evenly divisible by 16,384 and FilterMax finds two chunk boundaries in the last 32,768 bytes of data, the second chunk boundary MUST be ignored.

2.4.2.2 Generating Signatures

Signatures are generated as specified in [MS-RDC] section 3.1.5.2.

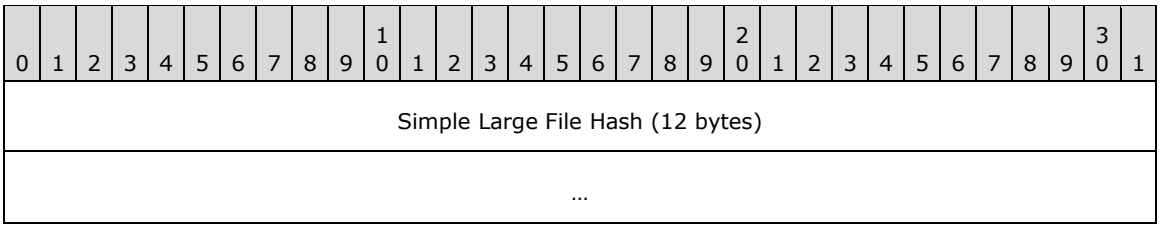
2.4.3 Simple Chunking Method

Files are split into chunks that are each 1 megabyte in size. If the total size, in bytes, of the file is less than or equal to 250 megabytes, the signature for each chunk has the structure that is shown in the following diagram.



Simple Small File Hash: A 20-byte sequence that specifies the SHA-1 hash code of the file bytes represented by the chunk.

If the total size, in bytes, is greater than 250 megabytes, the signature for each chunk has the structure that is shown in the following diagram.



Simple Large File Hash: A 12-byte sequence that specifies the chunk signature. This sequence of bytes MUST be unique.

For each chunk in the chunk list, an **Leaf Node Object**, as specified in section 2.2.3, is created. The **Data Size** of the **Leaf Node Object** MUST be the total number of bytes represented by the chunk. The **Signature Data** of the **Leaf Node Object** MUST be the chunk's signature. The **Leaf Node** is referenced by its parent node.

For all **Leaf Node Objects**, a **Data Node Object** MUST be created. The **Object Data** of the **Data Node Object** MUST be the byte sequence from the file tracked by the chunk. The **Object Reference Array** and the **Cell Reference Array** of the **Data Node Object** MUST be empty. The **Object References Array** of the **Leaf Node Object** associated with this **Data Node Object** MUST have a single entry, which MUST be the **Object ID** of the **Data Node Object.**

3 Structure Examples

3.1 Put Changes Request

This section provides an example of a **Put Changes** request saving a .ZIP file through the protocol.

```
00000000: 0C 00 0B 00 9C CF 29 F3  39 94 06 9B 06 02 00 00
00000010: EE 02 00 00 AA 02 20 00  7E B8 31 E7 45 DD AA 44
00000020: AB 80 0C 75 FB D1 53 0E  7A 02 08 00 95 A3 E0 2E
00000030: 77 01 16 02 06 00 03 0B  00 D2 02 26 00 0C F8 DD
00000040: BF 1E FA 64 E7 4E A5 DB  61 44 7E 8A 8C C1 00 48
00000050: 0B 01 AC 02 00 0C 56 0C  2F 16 61 BB 32 55 D4 4B
00000060: 98 8B C6 87 B9 A9 85 8D  80 37 2D 91 05 80 B3 D4
00000070: 4A 8E BE 9D EA 85 0F D5  C3 01 00 00 00 00 00 00
00000080: 00 0B EC 00 C0 32 80 EC  BC 97 4D DC 28 C5 41 92
00000090: 74 26 CB 57 96 6F 17 01  00 00 11 03 21 07 00 75
000000A0: F4 00 B0 A4 07 80 EC BC  97 4D DC 28 C5 41 92 74
000000B0: 26 CB 57 96 6F 17 02 00  00 12 80 EC BC 97 4D DC
000000C0: 28 C5 41 92 74 26 CB 57  96 6F 17 03 00 00 12 80
000000D0: EC BC 97 4D DC 28 C5 41  92 74 26 CB 57 96 6F 17
000000E0: 04 00 00 12 00 21 04 01  08 03 00 10 11 DC 00 00
000000F0: 00 00 00 00 00 81 79 05  0C 56 14 2F 16 61 BB 32
00000100: 55 D4 4B 98 8B C6 87 B9  A9 85 8D 80 37 2D 91 05
00000110: 80 B3 D4 4A 8E BE 9D EA  85 0F D5 C3 02 00 00 00
00000120: 00 00 00 00 0B EC 00 C0  32 80 EC BC 97 4D DC 28
00000130: C5 41 92 74 26 CB 57 96  6F 17 02 00 00 12 03 71
00000140: 03 00 75 F4 00 B0 A0 03  80 EC BC 97 4D DC 28 C5
00000150: 41 92 74 26 CB 57 96 6F  17 05 00 00 12 00 71 FC
00000160: 00 08 53 51 F3 33 D2 A6  BB 6F 43 C9 81 7A AB 3A
00000170: 62 9D 3C 8A 39 5F 10 9D  82 89 D1 F7 05 00 00 00
00000180: 00 00 00 00 05 00 00 00  00 00 00 00 10 11 2C 00
00000190: 00 00 00 00 00 00 7D 79  05 0C 56 1C 2F 16 61 BB
000001A0: 32 55 D4 4B 98 8B C6 87  B9 A9 85 8D 80 37 2D 91
000001B0: 05 80 B3 D4 4A 8E BE 9D  EA 85 0F D5 C3 03 00 00
000001C0: 00 00 00 00 00 0B EC 00  C0 32 80 EC BC 97 4D DC
000001D0: 28 C5 41 92 74 26 CB 57  96 6F 17 03 00 00 12 03
000001E0: 71 03 00 75 F4 00 B0 A0  03 80 EC BC 97 4D DC 28
000001F0: C5 41 92 74 26 CB 57 96  6F 17 06 00 00 12 00 71
00000200: FC 00 08 53 51 91 2F 5F  63 5F 88 C7 02 5E D9 BD
00000210: 48 96 F4 1A 62 D3 BC BE  B4 47 3E B6 FB 05 00 00
00000220: 00 00 00 00 00 05 00 00  00 00 00 00 00 10 11 2C
00000230: 00 00 00 00 00 00 00 7D  79 05 0C 56 24 2F 16 61
00000240: BB 32 55 D4 4B 98 8B C6  87 B9 A9 85 8D 80 37 2D
00000250: 91 05 80 B3 D4 4A 8E BE  9D EA 85 0F D5 C3 04 00
00000260: 00 00 00 00 00 00 0B EC  00 C0 32 80 EC BC 97 4D
00000270: DC 28 C5 41 92 74 26 CB  57 96 6F 17 04 00 00 12
00000280: 03 49 03 00 75 F4 00 B0  78 03 80 EC BC 97 4D DC
00000290: 28 C5 41 92 74 26 CB 57  96 6F 17 07 00 00 12 00
000002A0: 49 FC 00 08 2B 29 49 B5  3C 0E 99 CA 71 E4 D9 53
000002B0: 71 A6 6D 00 6E 60 EA 8F  A6 C6 10 11 84 00 00 00
000002C0: 00 00 00 00 7D 79 05 0C  56 2C 2F 16 61 BB 32 55
000002D0: D4 4B 98 8B C6 87 B9 A9  85 8D 80 37 2D 91 05 80
000002E0: B3 D4 4A 8E BE 9D EA 85  0F D5 C3 05 00 00 00 00
000002F0: 00 00 00 0B EC 00 C0 32  80 EC BC 97 4D DC 28 C5
00000300: 41 92 74 26 CB 57 96 6F  17 05 00 00 12 03 59 00
00000310: 00 75 F4 00 B0 5E 00 00  59 50 4B 03 04 14 00 00
00000320: 00 00 00 E5 AC 66 3E 82  89 D1 F7 05 00 00 00 05
00000330: 00 00 00 09 00 00 00 48  65 6C 6C 6F 2E 74 78 74
00000340: 48 65 6C 6C 6F 79 05 0C  56 34 2F 16 61 BB 32 55
00000350: D4 4B 98 8B C6 87 B9 A9  85 8D 80 37 2D 91 05 80
00000360: B3 D4 4A 8E BE 9D EA 85  0F D5 C3 06 00 00 00 00
00000370: 00 00 00 0B EC 00 C0 32  80 EC BC 97 4D DC 28 C5
00000380: 41 92 74 26 CB 57 96 6F  17 06 00 00 12 03 59 00
00000390: 00 75 F4 00 B0 5E 00 00  59 50 4B 03 04 14 00 00
000003A0: 00 00 00 F0 AC 66 3E 47  3E B6 FB 05 00 00 00 05
000003B0: 00 00 00 09 00 00 00 57  6F 72 6C 64 2E 74 78 74
```

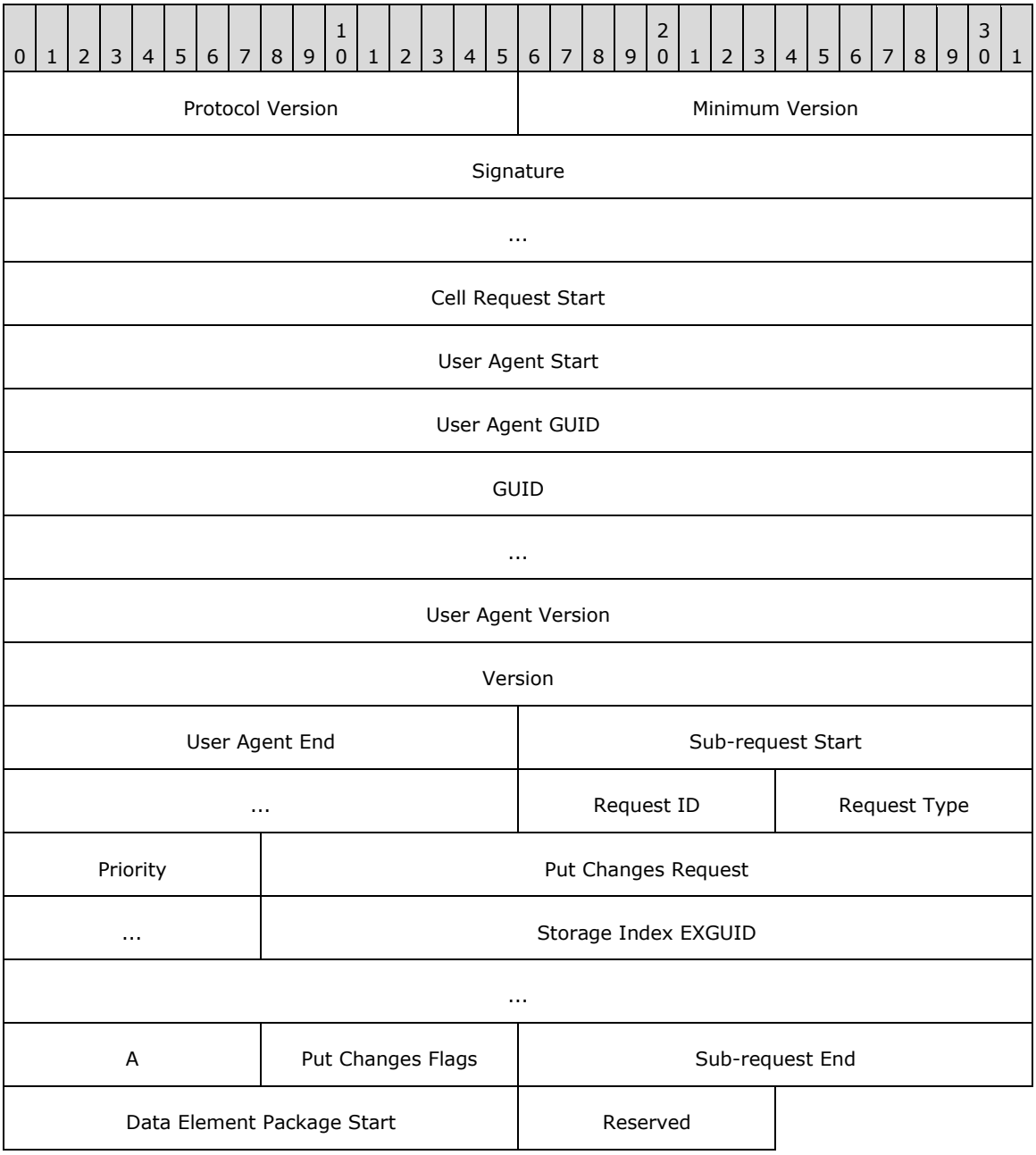
```
000003C0: 57 6F 72 6C 64 79 05 0C  56 3C 2F 16 61 BB 32 55
000003D0: D4 4B 98 8B C6 87 B9 A9  85 8D 80 37 2D 91 05 80
000003E0: B3 D4 4A 8E BE 9D EA 85  0F D5 C3 07 00 00 00 00
000003F0: 00 00 00 0B EC 00 C0 34  80 EC BC 97 4D DC 28 C5
00000400: 41 92 74 26 CB 57 96 6F  17 07 00 00 12 03 12 02
00000410: 00 00 75 F4 00 B2 00 10  01 00 00 12 02 50 4B 01
00000420: 02 14 00 14 00 00 00 00  00 E5 AC 66 3E 82 89 D1
00000430: F7 05 00 00 00 05 00 00  00 09 00 00 00 00 00 00
00000440: 00 01 00 20 00 00 00 00  00 00 00 48 65 6C 6C 6F
00000450: 2E 74 78 74 50 4B 01 02  14 00 14 00 00 00 00 00
00000460: F0 AC 66 3E 47 3E B6 FB  05 00 00 00 05 00 00 00
00000470: 09 00 00 00 00 00 00 00  01 00 20 00 00 00 2C 00
00000480: 00 00 57 6F 72 6C 64 2E  74 78 74 50 4B 05 06 00
00000490: 00 00 00 02 00 02 00 6E  00 00 00 58 00 00 00 00
000004A0: 00 79 05 0C 56 0C A0 93  65 66 4D 17 12 4F B0 45
000004B0: 83 1C 6A 44 BE 35 80 37  2D 91 05 80 B3 D4 4A 8E
000004C0: BE 9D EA 85 0F D5 C3 0A  00 00 00 00 00 00 00 05
000004D0: 60 20 94 33 B9 0E 1D 57  E9 41 AA D3 88 0D 92 D3
000004E0: 19 55 38 66 14 B9 FA DE  84 A3 AA 0D 4A A3 A8 52
000004F0: 0C 77 AC 70 73 0C B9 FA  DE 84 A3 AA 0D 4A A3 A8
00000500: 52 0C 77 AC 70 73 0C 65  46 2A 6F C8 42 C7 46 BA
00000510: B4 E2 8F DC E1 E3 2B 05  0C 56 4C 2F 16 61 BB 32
00000520: 55 D4 4B 98 8B C6 87 B9  A9 85 8D 80 37 2D 91 05
00000530: 80 B3 D4 4A 8E BE 9D EA  85 0F D5 C3 0B 00 00 00
00000540: 00 00 00 00 07 58 22 0C  89 C3 0D 4D 66 5E 6E 4D
00000550: 88 C4 52 71 D5 B4 80 28  05 0C 56 0C 39 04 FD BE
00000560: 69 4B B0 4A 8D F9 A4 B5  EA 91 D5 B9 80 37 2D 91
00000570: 05 80 B3 D4 4A 8E BE 9D  EA 85 0F D5 C3 0C 00 00
00000580: 00 00 00 00 00 09 D0 24  0C 89 C3 0D 4D 66 5E 6E
00000590: 4D 88 C4 52 71 D5 B4 80  28 00 50 4C 14 B9 FA DE
000005A0: 84 A3 AA 0D 4A A3 A8 52  0C 77 AC 70 73 80 EC BC
000005B0: 97 4D DC 28 C5 41 92 74  26 CB 57 96 6F 17 01 00
000005C0: 00 11 C8 22 0C 2F 16 61  BB 32 55 D4 4B 98 8B C6
000005D0: 87 B9 A9 85 8D C8 22 14  2F 16 61 BB 32 55 D4 4B
000005E0: 98 8B C6 87 B9 A9 85 8D  C8 22 1C 2F 16 61 BB 32
000005F0: 55 D4 4B 98 8B C6 87 B9  A9 85 8D C8 22 24 2F 16
00000600: 61 BB 32 55 D4 4B 98 8B  C6 87 B9 A9 85 8D C8 22
00000610: 2C 2F 16 61 BB 32 55 D4  4B 98 8B C6 87 B9 A9 85
00000620: 8D C8 22 34 2F 16 61 BB  32 55 D4 4B 98 8B C6 87
00000630: B9 A9 85 8D C8 22 3C 2F  16 61 BB 32 55 D4 4B 98
00000640: 8B C6 87 B9 A9 85 8D 05  0C 56 0C F8 DD BF 1E FA
00000650: 64 E7 4E A5 DB 61 44 7E  8A 8C C1 80 DB 35 CE 41
00000660: 06 A3 76 4D BA 08 A2 15  B4 A8 EA 05 01 00 00 00
00000670: 00 00 00 00 03 88 54 0C  A0 93 65 66 4D 17 12 4F
00000680: B0 45 83 1C 6A 44 BE 35  80 C8 D2 6E FA 7F 4C 2B
00000690: B5 8E BE 9D EA 85 0F D5  C3 19 00 00 00 00 00 00
000006A0: 00 70 98 0C B9 FA DE 84  A3 AA 0D 4A A3 A8 52 0C
000006B0: 77 AC 70 73 0C 65 46 2A  6F C8 42 C7 46 BA B4 E2
000006C0: 8F DC E1 E3 2B 4C 2F 16  61 BB 32 55 D4 4B 98 8B
000006D0: C6 87 B9 A9 85 8D 80 C8  D2 6E FA 7F 4C 2B B5 8E
000006E0: BE 9D EA 85 0F D5 C3 18  00 00 00 00 00 00 00 68
000006F0: 76 0C 89 C3 0D 4D 66 5E  6E 4D 88 C4 52 71 D5 B4
00000700: 80 28 0C 39 04 FD BE 69  4B B0 4A 8D F9 A4 B5 EA
00000710: 91 D5 B9 80 C8 D2 6E FA  7F 4C 2B B5 8E BE 9D EA
00000720: 85 0F D5 C3 17 00 00 00  00 00 00 00 05 55 03 01
```

3.1.1 Request Header

The following example consists of the request header for a **Put Changes** request.

```
00000000: 0C 00 0B 00 9C CF 29 F3  39 94 06 9B 06 02 00 00
00000010: EE 02 00 00 AA 02 20 00  7E B8 31 E7 45 DD AA 44
00000020: AB 80 0C 75 FB D1 53 0E  7A 02 08 00 95 A3 E0 2E
00000030: 77 01 16 02 06 00 03 0B  00 D2 02 26 00 0C F8 DD
00000040: BF 1E FA 64 E7 4E A5 DB  61 44 7E 8A 8C C1 00 48
```

```
00000050: 0B 01 AC 02 00
```



Protocol Version (2 bytes): 0x000C specifies the protocol version of this request.

Minimum Version (2 bytes): 0x000B specifies the minimum version of the protocol schema with which this request is compatible.

Signature: 0x9B069439F329CF9C specifies the signature of this request.

Cell Request Start (4 bytes): 0x00000206 specifies a stream object header for a cell request start. Decoded, this has a type of 0x40, a length of zero and is compound.

User Agent Start (4 bytes): 0x000002EE specifies a stream object header for user agent start. Decoded, this has a type of 0x5D and a length of zero and is compound.

User Agent GUID (4 bytes): 0x002002AA specifies a stream object header for a user agent GUID. Decoded, this has a type of 0x55 and a length of 16.

GUID: {E731B87E-DD45-44AA-80AB80-0C75FBD1530E} is the GUID of the user agent.

User Agent Version (4 bytes): 0x0008027A specifies a stream object header for the user agent version. Decoded, this has a type of 0x2F and a length of 4.

Version (4 bytes): 0x2EE0A395 specifies the version of the protocol client.

User Agent End (2 bytes): 0x0177 specifies a stream object header for user agent end.

Sub-request Start (4 bytes): 0x00060216 specifies a stream object header for the subrequest start. Decoded, this has a type of 0x42 and a length of 3.

Request Id (1 byte): 0x03 specifies the request number as a compact unsigned 64-bit integer for this request. Decoded, this represents a value of 0x1.

Request Type (1 byte): 0x0B specifies the request type as a compact unsigned 64-bit integer. Decoded, this represents a value of 0x05.

Priority (1 byte): 0x00 specifies the priority of this subrequest as a compact unsigned 64-bit integer.

Put Changes Request (4 bytes): 0x002602D2 specifies a stream object header for the **Put Changes** request. Decoded, this has a type of 0x5A and a length of 9.

Storage Index EXGUID: {1EBFDDF8-64FA-4EE7-A5DB-61447E8A8CC1} 0x01 specifies the storage index extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 0C F8 DD BF 1E FA 64 E7 4E A5 DB 61 44 7E 8A 8C C1.

A - Expected Storage Index EXGUID (1 byte): {000000-0000-0000-0000-00000000} 0x00 specifies the expected storage index extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 0x00.

Put Changes Flags (1 byte): 0x48 specifies the flags on the **Put Changes** request.

Sub-Request End (2 bytes): 0x010B specifies the stream object header for the subrequest end. Decoded, this has a type of 0x21.

Data Element Package Start (2 bytes): 0x02AC specifies the stream object header for a data element package start. Decoded, this has a type of 0x15 and a length of 1 and is compound.

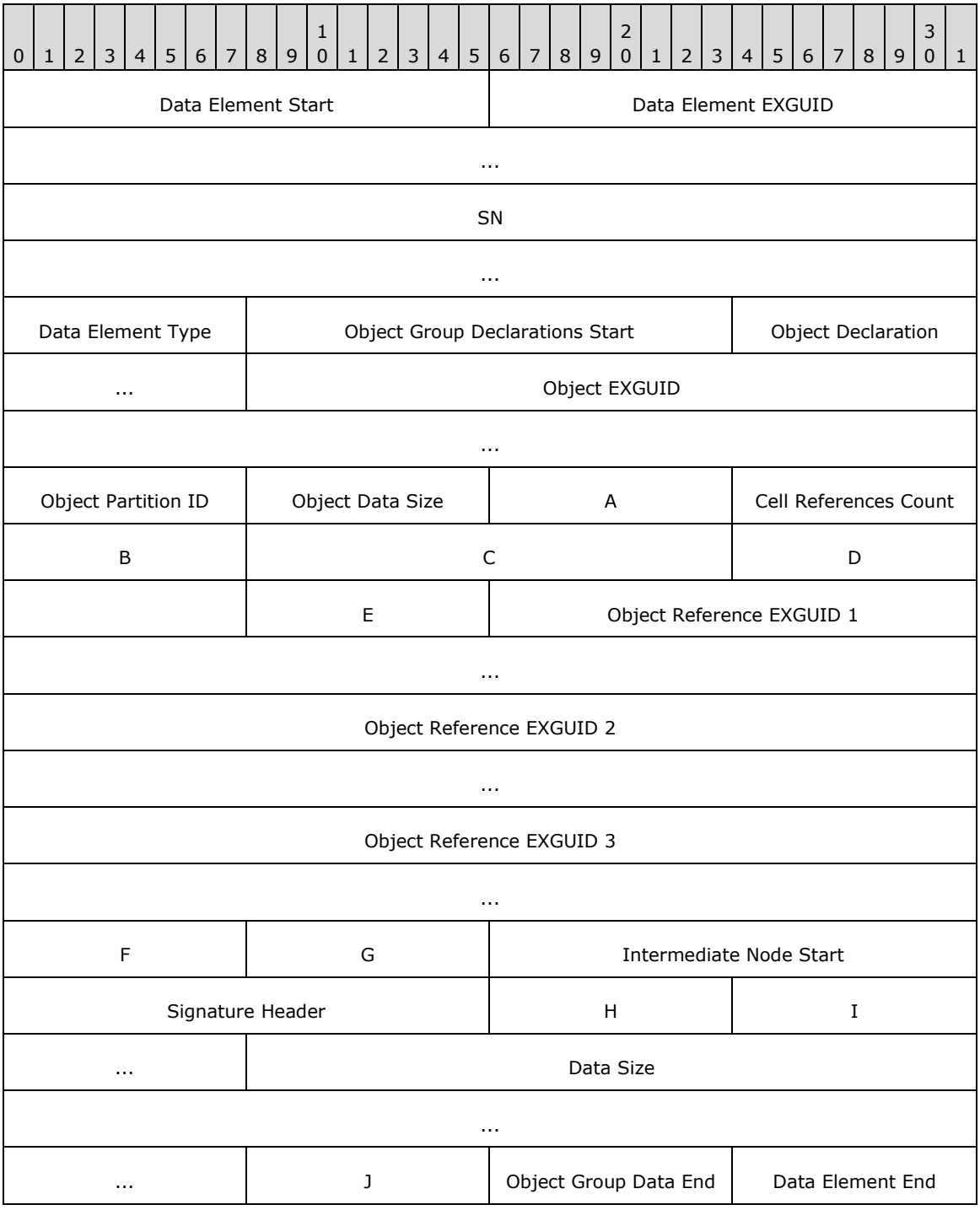
Reserved (1 byte): 0x00 specifies a reserved byte.

3.1.2 Object Groups

The following example consists of the object groups for a **Put Changes** request.

```
00000050:                0C 56 0C  2F 16 61 BB 32 55 D4 4B
00000060: 98 8B C6 87 B9 A9 85 8D  80 37 2D 91 05 80 B3 D4
00000070: 4A 8E BE 9D EA 85 0F D5  C3 01 00 00 00 00 00 00
00000080: 00 0B EC 00 C0 32 80 EC  BC 97 4D DC 28 C5 41 92
00000090: 74 26 CB 57 96 6F 17 01  00 00 11 03 21 07 00 75
000000A0: F4 00 B0 A4 07 80 02 00  00 12 EC BC 97 4D DC 28
000000B0: C5 41 92 74 26 CB 57 96  6F 17 80 03 00 00 12 EC
000000C0: BC 97 4D DC 28 C5 41 92  74 26 CB 57 96 6F 17 80
000000D0: 04 00 00 12 EC BC 97 4D  DC 28 C5 41 92 74 26 CB
000000E0: 57 96 6F 17 00 21 04 01  08 03 00 10 11 DC 00 00
```

```
000000F0: 00 00 00 00 00 81 79 05
```



Data Element Start (2 bytes): 0x560C specifies the stream object header for a data element start, as described in [MS-FSSHTTPB] section 2.2.1.5.1. Decoded, this has a type of 0x1 and a length of 43 and is compound.

Data Element EXGUID: {BB61162F-5532-4BD4-988B-C687B9A9858D} 0x01 specifies the data element extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 0C 2F 16 61 BB 32 55 D4 4B 98 8B C6 87 B9 A9 85 8D.

SN (25 bytes): {05912D37-B380-4AD4-8EBE-9DEA850FD5C3} 0x01 specifies the serial number decoded from 80 37 2D 91 05 80 B3 D4 4A 8E BE 9D EA 85 0F D5 C3 01 00 00 00 00 00 00 00.

Data Element Type (1 byte): 0x0B specifies the data element type as a compact unsigned 64-bit integer. Decoded, this represents a data element type of 0x5.

Object Group Declarations Start (2 bytes): 0x00EC specifies the stream object header for the object group declarations start. Decoded, this has a type of 0x1D and a length of zero and is compound.

Object Declaration (2 byte): 0x32C0 specifies the stream object header for an object declaration start. Decoded, this has a type of 0x18 and a length of 25. This specifies the start of the **Intermediate Node Object**, as specified in section 2.2.2.

Object EXGUID: {4D97BCEC-28DC-41C5-9274-26CB57966F17} 0x11000001 specifies the object extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 80 EC BC 97 4D DC 28 C5 41 92 74 26 CB 57 96 6F 17 01 00 00 11.

Object Partition ID (1 byte): 0x03 specifies an object partition identifier as a compact unsigned 64-bit integer with a decoded value of 0x01.

Object Data Size (1 byte): 0x21 specifies the size, in bytes, of the object as a compact unsigned 64-bit integer. Decoded, this represents 0x10.

A - Object References Count (1 byte): 0x07 specifies the number of object references as a compact unsigned 64-bit integer with a decoded value of 0x03.

Cell References Count (1 byte): 0x00 specifies the number of cell references as a compact unsigned 64-bit integer with a decoded value of 0x00.

B - Object Group Declaration end (1 byte): 0x75 specifies the stream object header for an object group declaration end.

C - Cell Object Group Data Header (2 bytes): 0x00F4 specifies the stream object header for a cell object group data header. Decoded, this has a type of 0x1E, length 0, and is compound.

D - Cell Object Group Object Data (2 bytes): 0xA4B0 specifies the stream object header for a cell object group object data. Decoded this has a type of 0x16 and a length of 82.

E - Object Extended GUID Array Count (1 byte): 0x07 specifies the number of extended GUIDs, as described in [MS-FSSHTTPB] section 2.2.1.7, as a compact unsigned 64-bit integer with a decoded value of 0x03.

Object Reference EXGUID 1: {4D97BCEC-28DC-41C5-9274-26CB57966F17} 0x12000002 specifies the object extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 80 02 00 00 12 EC BC 97 4D DC 28 C5 41 92 74 26 CB 57 96 6F 17. This references the first **Leaf Node Object**, as specified in section 2.2.3.

Object Reference EXGUID 2: {4D97BCEC-28DC-41C5-9274-26CB57966F17} 0x12000003 specifies the object extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 80 03 00 00 12 EC BC 97 4D DC 28 C5 41 92 74 26 CB 57 96 6F 17. This references the second **Leaf Node Object**.

Object Reference EXGUID 3: {4D97BCEC-28DC-41C5-9274-26CB57966F17} 0x12000004 specifies the object extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 80 04 00 00 12 EC BC 97 4D DC 28 C5 41 92 74 26 CB 57 96 6F 17. This references the third **Leaf Node Object**.

F – Cell ID Array Count (1 byte): 0x00 specifies the number of **Cell IDs** as a compact unsigned 64-bit integer with a decoded value of 0x00.

G - Object Data Length (1 byte): 0x21 specifies the number of bytes in **Object Data**, as described in [MS-FSSHTTPB] section 2.2.1.12.6.4, as a compact unsigned 64-bit integer with a decoded value of 0x10.

Intermediate Node Start (2 bytes): 0x0104 specifies a 16-bit stream object header, as specified in section 2.2.2.1

Signature Header (2 bytes): 0x0308 specifies a 16-bit stream object header, as specified in section 2.2.2. Decoded, this has **Header** equal to 0x00, **Compound** equal to 0x0, **Type** equal to 0x21, and **Length** equal to 0x01.

H - Signature Data (1 byte): 0x00 specifies a binary item representing the **Signature Data** for this node. Decoded, this has a length equal to zero and content equal to {}.

I - Data Size Header (2 bytes): 0x1110 specifies a 16-bit stream object header that, decoded, specifies a single object parse type with a stream object type of 0x22 and a length of 0x08.

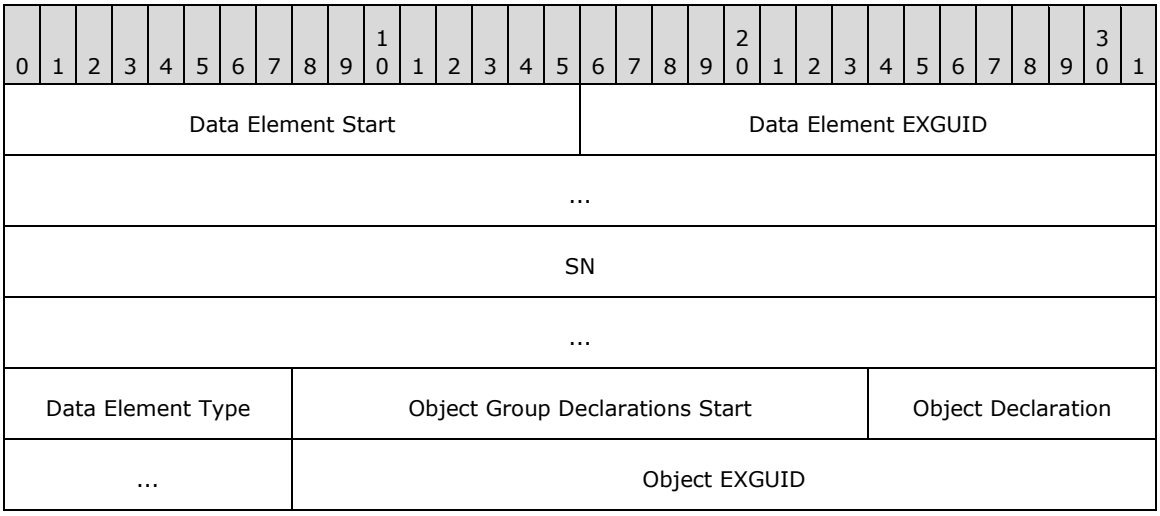
Data Size (8 bytes): 0x00000000000000DC represents an unsigned 64-bit integer that specifies the size of the file data represented by this **Intermediate Node Object**.

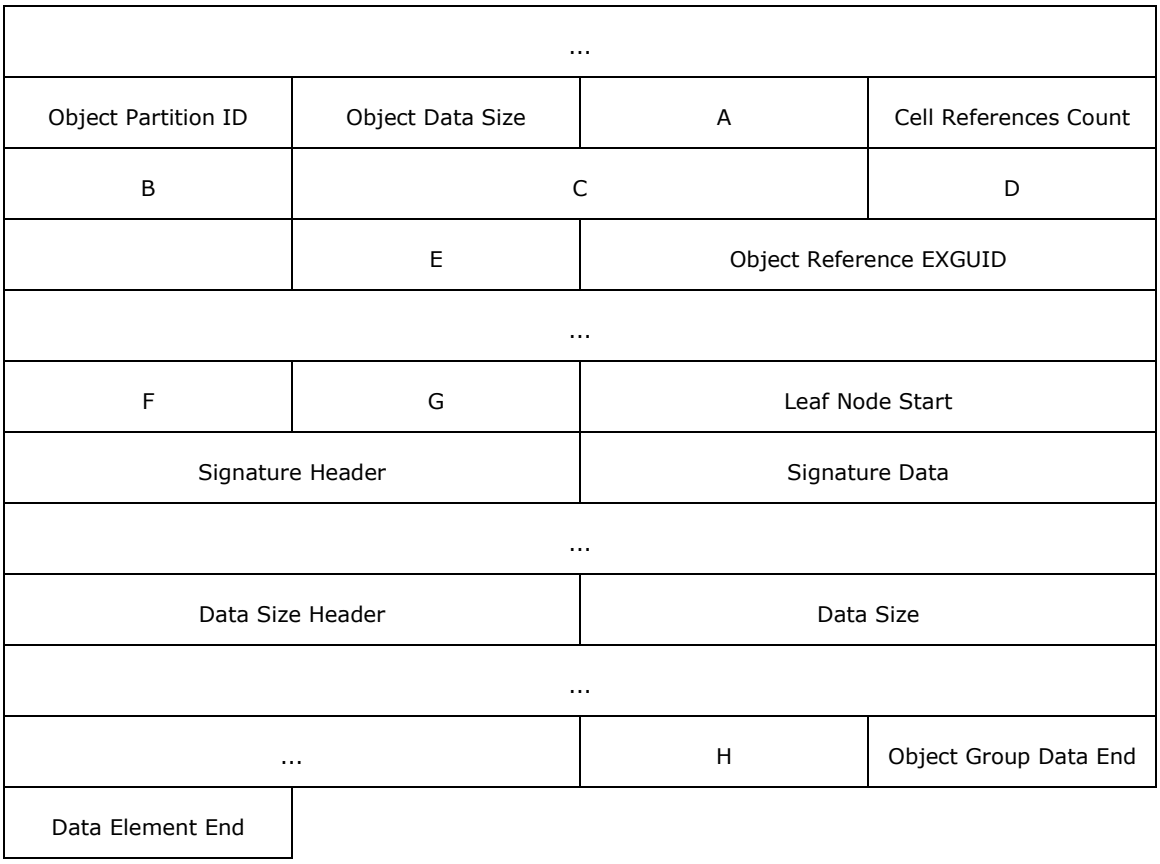
J - Intermediate Node End (1 byte): 0x81 specifies an 8-bit stream object header end decoded as a stream object type of 0x20.

Object Group Data End (1 byte): 0x79 specifies the stream object header for an object group data end.

Data Element End (1 byte): 0x05 specifies the stream object header for a data element end.

```
000000F0:                         0C 56 14 2F 16 61 BB 32
00000100: 55 D4 4B 98 8B C6 87 B9  A9 85 8D 80 37 2D 91 05
00000110: 80 B3 D4 4A 8E BE 9D EA  85 0F D5 C3 02 00 00 00
00000120: 00 00 00 00 0B EC 00 C0  32 80 EC BC 97 4D DC 28
00000130: C5 41 92 74 26 CB 57 96  6F 17 02 00 00 12 03 71
00000140: 03 00 75 F4 00 B0 A0 03  80 EC BC 97 4D DC 28 C5
00000150: 41 92 74 26 CB 57 96 6F  17 05 00 00 12 00 71 FC
00000160: 00 08 53 51 F3 33 D2 A6  BB 6F 43 C9 81 7A AB 3A
00000170: 62 9D 3C 8A 39 5F 10 9D  82 89 D1 F7 05 00 00 00
00000180: 00 00 00 00 05 00 00 00  00 00 00 00 10 11 2C 00
00000190: 00 00 00 00 00 00 7D 79  05
```





Data Element Start (2 bytes): 0x560C specifies the stream object header for a data element start, as described in [MS-FSSHTTPB] section 2.2.1.5.1. Decoded, this has a type of 0x1 and a length of 43 and is compound.

Data Element EXGUID: {BB61162F-5532-4BD4-988B-C687B9A9858D} 0x02 specifies the data element extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 14 2F 16 61 BB 32 55 D4 4B 98 8B C6 87 B9 A9 85 8D.

SN (25 bytes): {05912D37-B380-4AD4-8EBE-9DEA850FD5C3} 0x02 specifies the serial number decoded from 80 37 2D 91 05 80 B3 D4 4A 8E BE 9D EA 85 0F D5 C3 02 00 00 00 00 00 00 00.

Data Element Type (1 byte): 0x0B specifies the data element type as a compact unsigned 64-bit **integer**. Decoded, this represents a data element type of 0x5.

Object Group Declarations Start (2 bytes): 0x00EC specifies the stream object header for object group declaration start. Decoded this has a type of 0x1D, length zero and is compound.

Object Declaration (2 bytes): 0x32C0 specifies the stream object header for an object declaration start. Decoded this has a type of 0x18, length 25. This specifies the start of the leaf node object, as specified in section 2.2.3.

Object EXGUID: {4D97BCEC-28DC-41C5-9274-26CB57966F17} 0x12000002 specifies the object extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 80 EC BC 97 4D DC 28 C5 41 92 74 26 CB 57 96 6F 17 02 00 00 12.

Object Partition ID (1 byte): 0x03 specifies an object partition identifier as a compact unsigned 64-bit **integer** with a decoded value of 0x01.

Object Data Size (1 byte): 0x71 specifies the size of bytes of the object as a compact unsigned 64-bit **integer**. Decoded, this represents 0x38.

A - Object References Count (1 byte): 0x03 specifies the number of object references as a compact unsigned 64-bit **integer** with a decoded value of 0x01.

Cell References Count (1 byte): 0x00 specifies the number of cell references as a compact unsigned 64-bit **integer** with a decoded value of 0x00.

B - Object Group Declaration end (1 byte): 0x75 specifies the stream object header for an object group declaration end.

C - Cell Object Group Data Header (2 bytes): 0x00F4 specifies the stream object header for a cell object group data header. Decoded, this has a type of 0x1E, length 0, and is compound.

D - Cell Object Group Object Data (2 bytes): 0xA0B0 specifies the stream object header for a cell object group object data. Decoded, this has a type of 0x16, length 80.

E - Object Extended GUID Array Count (1 byte): 0x03 specifies the number of Extended GUIDs, as described in [MS-FSSHTTPB] section 2.2.1.7, as a compact unsigned 64-bit **integer** with a decoded value of 0x01.

Object Reference EXGUID : {4D97BCEC-28DC-41C5-9274-26CB57966F17} 0x12000005 specifies the object extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 80 EC BC 97 4D DC 28 C5 41 92 74 26 CB 57 96 6F 17 05 00 00 12. This references the first data node object, as specified in section 2.2.4.

F – Cell ID Array Count (1 byte): 0x00 specifies the number of **Cell IDs** as a compact unsigned 64-bit **integer** with a decoded value of 0x00.

G - Object Data Length (1 byte): 0x71 specifies the count of bytes of **Object Data** as a compact unsigned 64-bit **integer** with a decoded value of 0x38.

Leaf Node Start (2 bytes): 0x00FC specifies a 16-bit stream object header, as specified in section 2.2.3

Signature Header (2 bytes): 0x5308 specifies a 16-bit stream object header, as specified in section 2.2.3. Decoded this has Header of 0x00, Compound of 0x0, Type of 0x21 and Length of 0x29.

Signature Data (41 bytes): 0x51 specifies a binary item representing the **Signature Data** for this node. Decoded, this has length 0x28, content {F3 33 D2 A6 BB 6F 43 C9 81 7A AB 3A 62 9D 3C 8A 39 5F 10 9D 82 89 D1 F7 05 00 00 00 00 00 00 00 05 00 00 00 00 00 00 00}.

Data Size Header (2 bytes): 0x1110 specifies a 16-bit stream object header that, decoded, specifies a single object parse type with a stream object type of 0x22, length 0x08.

Data Size (8 bytes): 0x000000000000002C is an unsigned 64-bit **integer** that specifies the size of the file data represented by this leaf node object.

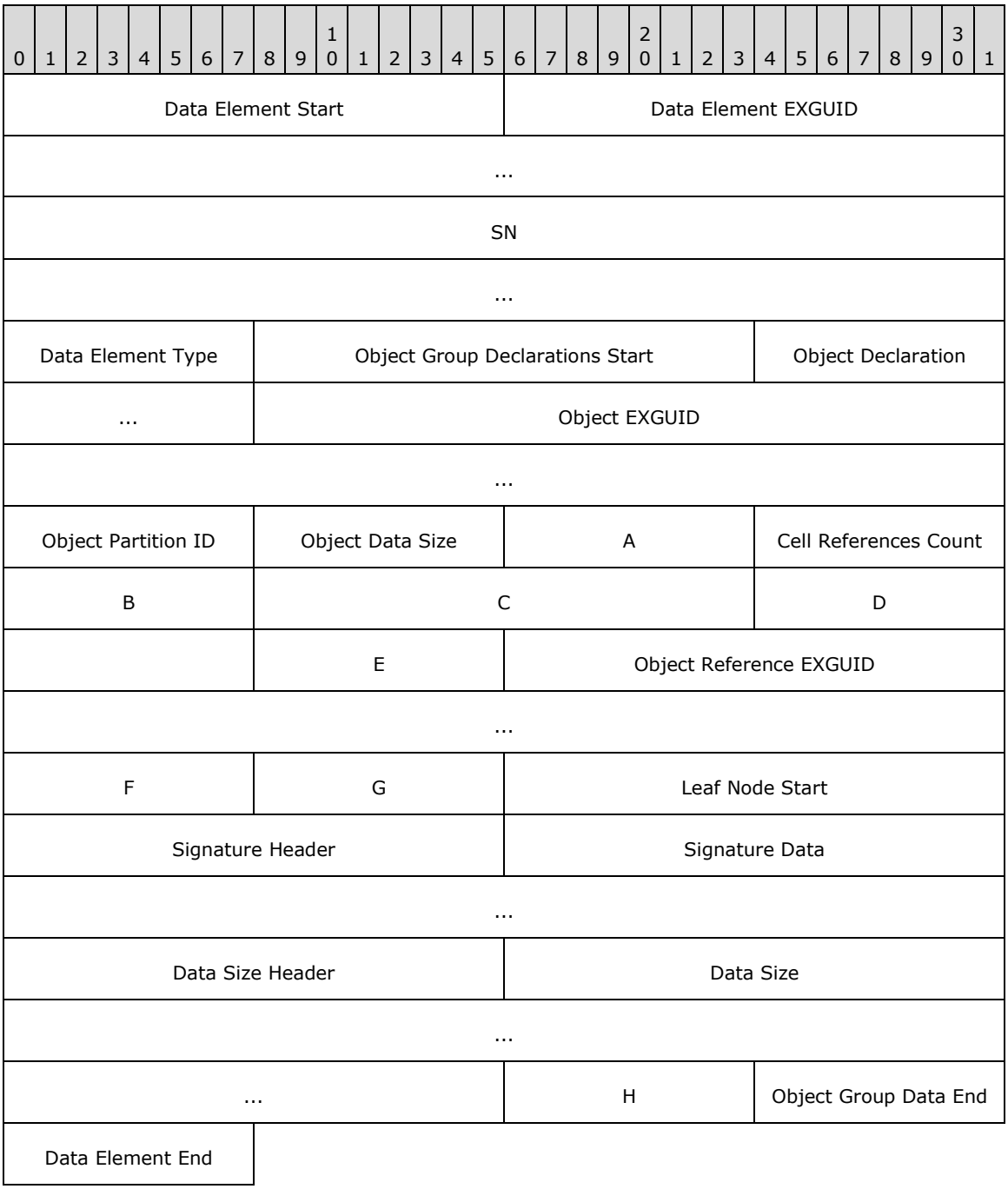
H - Leaf Node End (1 byte): 0x7D specifies an 8-bit stream object header end decoded as stream object type 0x1F.

Object Group Data End (1 byte): 0x79 specifies the stream object header for an object group data end.

Data Element End (1 byte): 0x05 specifies the stream object header for data element end.

```
00000190:                           0C 56 1C 2F 16 61 BB
000001A0: 32 55 D4 4B 98 8B C6 87  B9 A9 85 8D 80 37 2D 91
000001B0: 05 80 B3 D4 4A 8E BE 9D  EA 85 0F D5 C3 03 00 00
```

```
000001C0: 00 00 00 00 00 0B EC 00  C0 32 80 EC BC 97 4D DC
000001D0: 28 C5 41 92 74 26 CB 57  96 6F 17 03 00 00 12 03
000001E0: 71 03 00 75 F4 00 B0 A0  03 80 EC BC 97 4D DC 28
000001F0: C5 41 92 74 26 CB 57 96  6F 17 06 00 00 12 00 71
00000200: FC 00 08 53 51 91 2F 5F  63 5F 88 C7 02 5E D9 BD
00000210: 48 96 F4 1A 62 D3 BC BE  B4 47 3E B6 FB 05 00 00
00000220: 00 00 00 00 00 05 00 00  00 00 00 00 00 10 11 2C
00000230: 00 00 00 00 00 00 00 7D  79 05
```



Data Element Start (2 bytes): 0x560C specifies the stream object header for a data element start, as specified in [MS-FSSHTTPB] section 2.2.1.5.1. Decoded, this has a type of 0x1, length 43, Compound.

Data Element EXGUID: {BB61162F-5532-4BD4-988B-C687B9A9858D} 0x03 specifies the **Data** element extended GUID, as specified in [MS-FSSHTTPB] section 2.2.1.7, decoded from 1C 2F 16 61 BB 32 55 D4 4B 98 8B C6 87 B9 A9 85 8D.

SN (25 bytes): {05912D37-B380-4AD4-8EBE-9DEA850FD5C3} 0x03 specifies the serial number decoded from 80 37 2D 91 05 80 B3 D4 4A 8E BE 9D EA 85 0F D5 C3 03 00 00 00 00 00 00 00.

Data Element Type (1 byte): 0x0B specifies the data element type as a compact unsigned 64-bit integer. Decoded, this represents a data element type of 0x5.

Object Group Declarations Start (2 bytes): 0x00EC specifies the stream object header for and object group declarations start. Decoded, this has a type of 0x1D and a length of zero and is compound.

Object Declaration (2 bytes): 0x32C0 specifies the stream object header for an object declaration start. Decoded, this has a type of 0x18 and a length of 25. This specifies the start of the **Leaf Node Object**, as specified in section 2.2.3.

Object EXGUID: {4D97BCEC-28DC-41C5-9274-26CB57966F17} 0x12000003 specifies the object extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 80 EC BC 97 4D DC 28 C5 41 92 74 26 CB 57 96 6F 17 03 00 00 12.

Object Partition ID (1 byte): 0x03 specifies an object partition identifier as a compact unsigned 64-bit integer with a decoded value of 0x01.

Object Data Size (1 byte): 0x71 specifies the size, in bytes, of the object as a compact unsigned 64-bit integer. Decoded, this represents 0x38.

A - Object References Count (1 byte): 0x03 specifies the number of object references as a compact unsigned 64-bit integer with a decoded value of 0x01.

Cell References Count (1 byte): 0x00 specifies the number of cell references as a compact unsigned 64-bit integer with a decoded value of 0x00.

B - Object Group Declaration end (1 byte): 0x75 specifies the stream object header for an object group declaration end.

C - Cell Object Group Data Header (2 bytes): 0x00F4 specifies the stream object header for a cell object group data header. Decoded, this has a type of 0x1E, length 0, and is compound.

D - Cell Object Group Object Data (2 bytes): 0xA0B0 specifies the stream object header for a cell object group object data. Decoded, this has a type of 0x16 and a length of 80.

E - Object Extended GUID Array Count (1 byte): 0x03 specifies the number of extended GUIDs, as described in [MS-FSSHTTPB] section 2.2.1.7, as a compact unsigned 64-bit integer with a decoded value of 0x01.

Object Reference EXGUID : {4D97BCEC-28DC-41C5-9274-26CB57966F17} 0x12000006 specifies the object extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 80 EC BC 97 4D DC 28 C5 41 92 74 26 CB 57 96 6F 17 06 00 00 12. This references the second **Data Node Object**, as specified in section 2.2.4.

F – Cell ID Array Count (1 byte): 0x00 specifies the number of **Cell IDs** as a compact unsigned 64-bit integer with a decoded value of 0x00.

G - Object Data Length (1 byte): 0x71 specifies the number of bytes in **Object Data** as a compact unsigned 64-bit integer with a decoded value of 0x38.

Leaf Node Start (2 bytes): 0x00FC specifies a 16-bit stream object header, as specified in section 2.2.3.

Signature Header (2 bytes): 0x5308 specifies a 16-bit stream object header, as specified in section 2.2.3. Decoded this has **Header** equal to 0x00, **Compound** equal to 0x0, **Type** equal to 0x21, and **Length** equal to 0x29.

Signature Data (41 bytes): 0x51 specifies a binary item representing the **Signature Data** for this node. Decoded, this has a length of 0x28 and content equal to {91 2F 5F 63 5F 88 C7 02 5E D9 BD 48 96 F4 1A 62 D3 BC BE B4 47 3E B6 FB 05 00 00 00 00 00 00 00 05 00 00 00 00 00 00 00}.

Data Size Header (2 bytes): 0x1110 specifies a 16-bit stream object header, as specified in section 2.2.3. Decoded, this specifies a single object parse type with a stream object type of 0x22 and a length of 0x08.

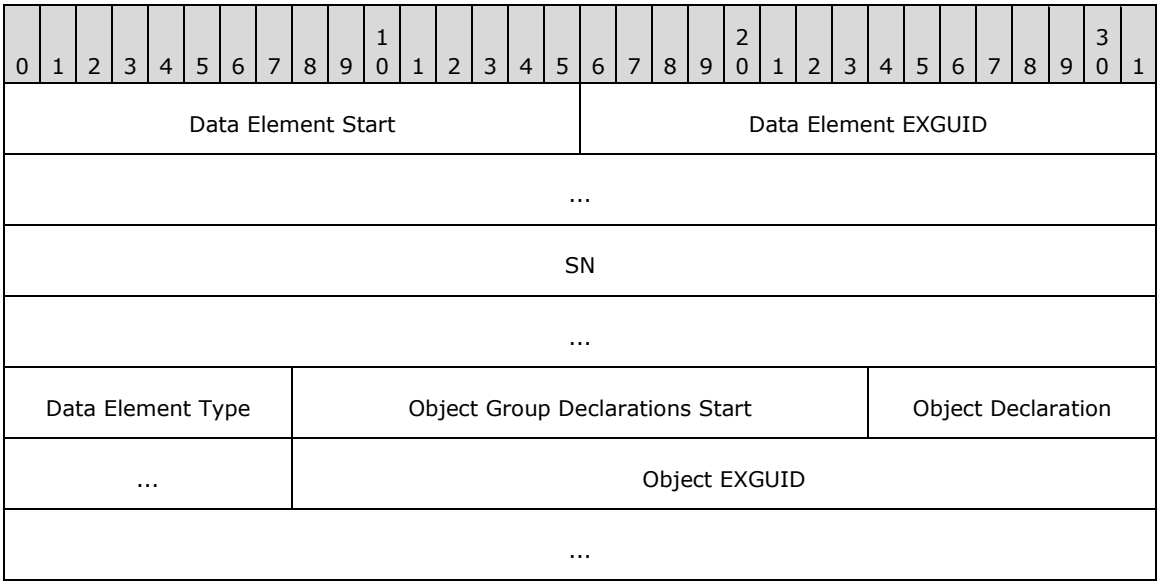
Data Size (8 bytes): 0x000000000000002C specifies an unsigned 64-bit integer that specifies the size of the file data represented by this **Leaf Node Object**.

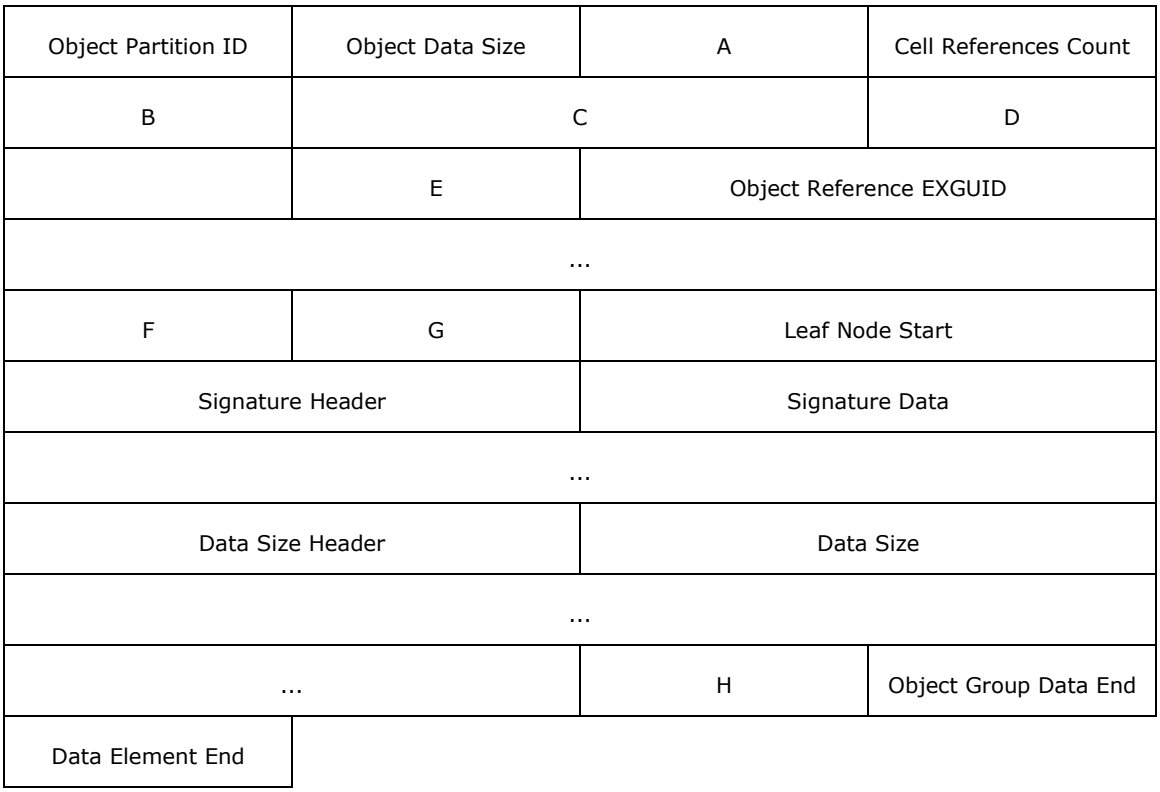
H - Leaf Node End (1 byte): 0x7D specifies an 8-bit stream object header end decoded as a stream object type of 0x1F.

Object Group Data End (1 byte): 0x79 specifies the stream object header for an object group data end.

Data Element End (1 byte): 0x05 specifies the stream object header for a data element end.

```
00000230:                                  0C 56 24 2F 16 61
00000240: BB 32 55 D4 4B 98 8B C6  87 B9 A9 85 8D 80 37 2D
00000250: 91 05 80 B3 D4 4A 8E BE  9D EA 85 0F D5 C3 04 00
00000260: 00 00 00 00 00 00 0B EC  00 C0 32 80 EC BC 97 4D
00000270: DC 28 C5 41 92 74 26 CB  57 96 6F 17 04 00 00 12
00000280: 03 49 03 00 75 F4 00 B0  78 03 80 EC BC 97 4D DC
00000290: 28 C5 41 92 74 26 CB 57  96 6F 17 07 00 00 12 00
000002A0: 49 FC 00 08 2B 29 49 B5  3C 0E 99 CA 71 E4 D9 53
000002B0: 71 A6 6D 00 6E 60 EA 8F  A6 C6 10 11 84 00 00 00
000002C0: 00 00 00 00 7D 79 05
```





Data Element Start (2 bytes): 0x560C specifies the stream object header for a data element start, as described in [MS-FSSHTTPB] section 2.2.1.5.1. Decoded, this has a type of 0x1 and a length of 43 and is compound.

Data Element EXGUID: {BB61162F-5532-4BD4-988B-C687B9A9858D} 0x04 specifies the data element extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 24 2F 16 61 BB 32 55 D4 4B 98 8B C6 87 B9 A9 85 8D.

SN (25 bytes): {05912D37-B380-4AD4-8EBE-9DEA850FD5C3} 0x04 specifies the serial number decoded from 80 37 2D 91 05 80 B3 D4 4A 8E BE 9D EA 85 0F D5 C3 04 00 00 00 00 00 00 00.

Data Element Type (1 byte): 0x0B specifies the data element type as a compact unsigned 64-bit integer. Decoded, this represents a data element type of 0x5.

Object Group Declarations Start (2 bytes): 0x00EC specifies the stream object header for an object group declarations start. Decoded, this has a type of 0x1D and a length of zero and is compound.

Object Declaration (2 bytes): 0x32C0 specifies the stream object header for an object declaration start. Decoded, this has a type of 0x18 and a length of 25. This specifies the start of the **Leaf Node Object**, as specified in section 2.2.3.

Object EXGUID: {4D97BCEC-28DC-41C5-9274-26CB57966F17} 0x12000004 specifies the object extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 80 EC BC 97 4D DC 28 C5 41 92 74 26 CB 57 96 6F 17 04 00 00 12.

Object Partition ID (1 byte): 0x03 specifies an object partition identifier as a compact unsigned 64-bit integer with a decoded value of 0x01.

Object Data Size (1 byte): 0x49 specifies the size, in bytes, of the object as a compact unsigned 64-bit integer. Decoded, this represents 0x24.

A - Object References Count (1 byte): 0x03 specifies the number of object references as a compact unsigned 64-bit integer with a decoded value of 0x01.

Cell References Count (1 byte): 0x00 specifies the number of cell references as a compact unsigned 64-bit integer with a decoded value of 0x00.

B - Object Group Declaration end (1 byte): 0x75 specifies the stream object header for an object group declaration end.

C - Cell Object Group Data Header (2 bytes): 0x00F4 specifies the stream object header for a cell object group data header. Decoded, this has a type of 0x1E, length 0, and is compound.

D - Cell Object Group Object Data (2 bytes): 0x78B0 specifies the stream object header for a cell object group object data. Decoded, this has a type of 0x16 and a of length 60.

E - Object Extended GUID Array Count (1 byte): 0x03 specifies the number of extended GUIDs, as described in [MS-FSSHTTPB] section 2.2.1.7, as a compact unsigned 64-bit integer with a decoded value of 0x01.

Object Reference EXGUID : {4D97BCEC-28DC-41C5-9274-26CB57966F17} 0x12000007 specifies the object extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 80 EC BC 97 4D DC 28 C5 41 92 74 26 CB 57 96 6F 17 07 00 00 12. This references the third **Data Node Object**, as specified in section 2.2.4.

F – Cell ID Array Count (1 byte): 0x00 specifies the number of **Cell IDs** as a compact unsigned 64-bit integer with a decoded value of 0x00.

G - Object Data Length (1 byte): 0x49 specifies the number of bytes in **Object Data** as a compact unsigned 64-bit integer with a decoded value of 0x24.

Leaf Node Start (2 bytes): 0x00FC specifies a 16-bit stream object header, as specified in section 2.2.3

Signature Header (2 bytes): 0x2B08 specifies a 16-bit stream object header, as specified in section 2.2.3. Decoded this has **Header** equal to 0x00, **Compound** equal to 0x0, **Type** equal to 0x21, and **Length** equal to 0x15.

Signature Data (21 bytes): 0x29 specifies a binary item representing the signature data for this node. Decoded, this has a length of 0x14 and content equal to {49 B5 3C 0E 99 CA 71 E4 D9 53 71 A6 6D 00 6E 60 EA 8F A6 C6}.

Data Size Header (2 bytes): 0x1110 specifies a 16-bit stream object header, as specified in section 2.2.3. Decoded, this specifies a single object parse type with a stream object type of 0x22 and a length of 0x08.

Data Size (8 bytes): 0x0000000000000084 specifies an unsigned 64-bit integer that specifies the size of the file data represented by this **Leaf Node Object**.

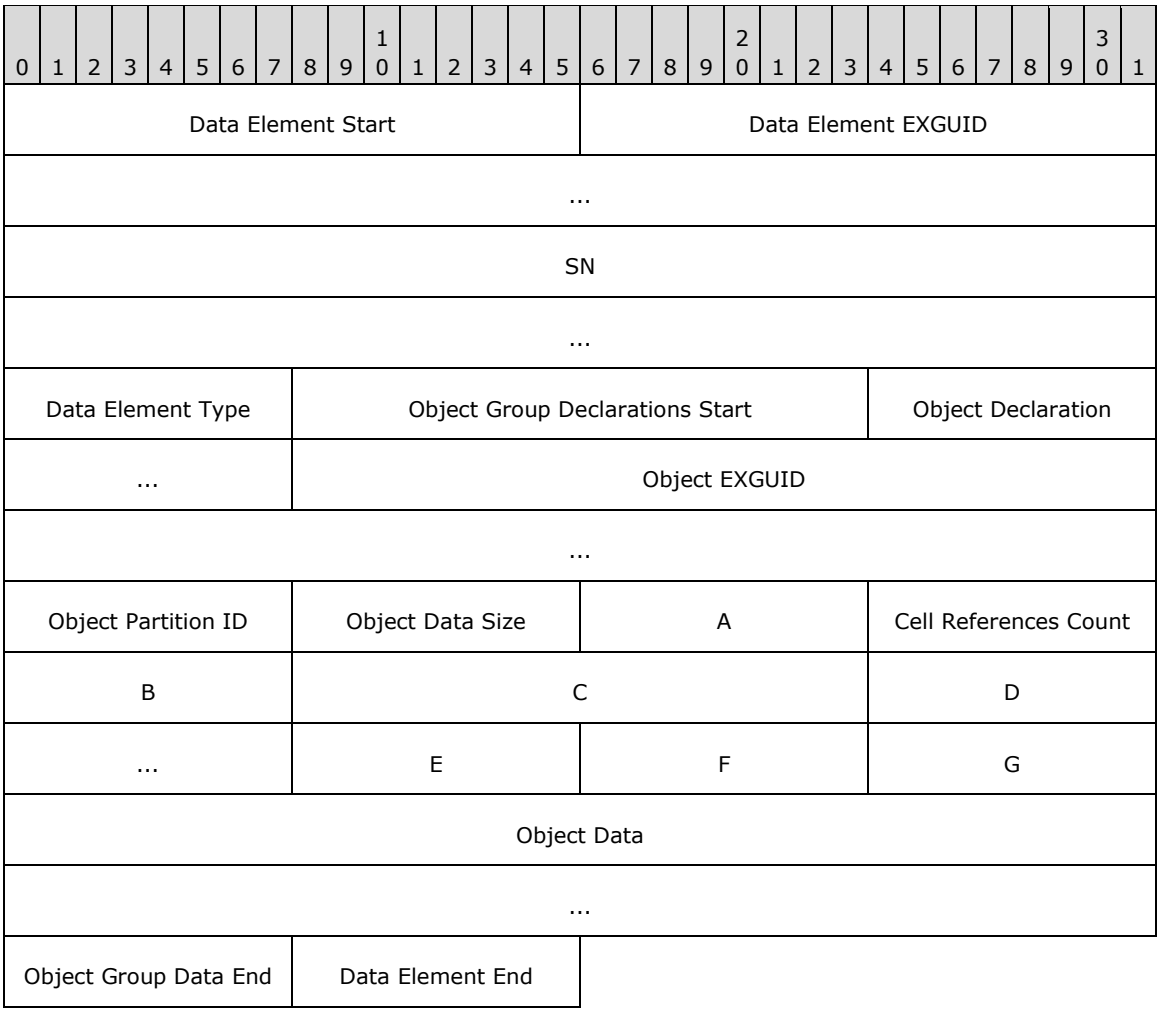
H - Leaf Node End (1 byte): 0x7D specifies an 8-bit stream object header end decoded as a stream object type of 0x1F.

Object Group Data End (1 byte): 0x79 specifies the stream object header for an object group data end.

Data Element End (1 byte): 0x05 specifies the stream object header for a data element end.

```
000002C0:                      0C  56 2C 2F 16 61 BB 32 55
000002D0: D4 4B 98 8B C6 87 B9 A9  85 8D 80 37 2D 91 05 80
000002E0: B3 D4 4A 8E BE 9D EA 85  0F D5 C3 05 00 00 00 00
000002F0: 00 00 00 0B EC 00 C0 32  80 EC BC 97 4D DC 28 C5
```

```
00000300: 41 92 74 26 CB 57 96 6F  17 05 00 00 12 03 59 00
00000310: 00 75 F4 00 B0 5E 00 00  59 50 4B 03 04 14 00 00
00000320: 00 00 00 E5 AC 66 3E 82  89 D1 F7 05 00 00 00 05
00000330: 00 00 00 09 00 00 00 48  65 6C 6C 6F 2E 74 78 74
00000340: 48 65 6C 6C 6F 79 05
```



Data Element Start (2 bytes): 0x560C specifies the stream object header for a data element start, as described in [MS-FSSHTTPB] section 2.2.1.5.1. Decoded, this has a type of 0x1 and a length of 43 and is compound.

Data Element EXGUID: {BB61162F-5532-4BD4-988B-C687B9A9858D} 0x05 specifies the data element extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 2C 2F 16 61 BB 32 55 D4 4B 98 8B C6 87 B9 A9 85 8D.

SN (25 bytes): {05912D37-B380-4AD4-8EBE-9DEA850FD5C3} 0x05 specifies the serial number decoded from 80 37 2D 91 05 80 B3 D4 4A 8E BE 9D EA 85 0F D5 C3 05 00 00 00 00 00 00 00.

Data Element Type (1 byte): 0x0B specifies the data element type as a compact unsigned 64-bit integer. Decoded, this represents a data element type of 0x5.

Object Group Declarations Start (2 bytes): 0x00EC specifies the stream object header for an object group declaration start. Decoded, this has a type of 0x1D and a length of zero and is compound.

Object Declaration (2 bytes): 0x32C0 specifies the stream object header for an object declaration start. Decoded, this has a type of 0x18 and a length of 25. This specifies the start of the **Leaf Node Object**, as specified in section 2.2.3.

Object EXGUID: {4D97BCEC-28DC-41C5-9274-26CB57966F17} 0x12000005 specifies the object extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 80 EC BC 97 4D DC 28 C5 41 92 74 26 CB 57 96 6F 17 05 00 00 12.

Object Partition ID (1 byte): 0x03 specifies an object partition identifier as a compact unsigned 64-bit integer with a decoded value of 0x01.

Object Data Size (1 byte): 0x59 specifies the size, in bytes, of the object as a compact unsigned 64-bit integer. Decoded, this represents 0x2C.

A - Object References Count (1 byte): 0x00 specifies the number of object references as a compact unsigned 64-bit integer with a decoded value of 0x00.

Cell References Count (1 byte): 0x00 specifies the number of cell references as a compact unsigned 64-bit integer with a decoded value of 0x00.

B - Object Group Declaration end (1 byte): 0x75 specifies the stream object header for an object group declaration end.

C - Cell Object Group Data Header (2 bytes): 0x00F4 specifies the stream object header for a cell object group data header. Decoded, this has a type of 0x1E, length 0, and is compound.

D - Cell Object Group Object Data (2 bytes): 0x5EB0 specifies the stream object header for a cell object group object data. Decoded, this has a type of 0x16 and a of length 47.

E - Object Extended GUID Array Count (1 byte): 0x00 specifies the number of extended GUIDs, as described in [MS-FSSHTTPB] section 2.2.1.7, as a compact unsigned 64-bit integer with a decoded value of 0x00.

F – Cell ID Array Count (1 byte): 0x00 specifies the number of **Cell IDs** as a compact unsigned 64-bit integer with a decoded value of 0x00.

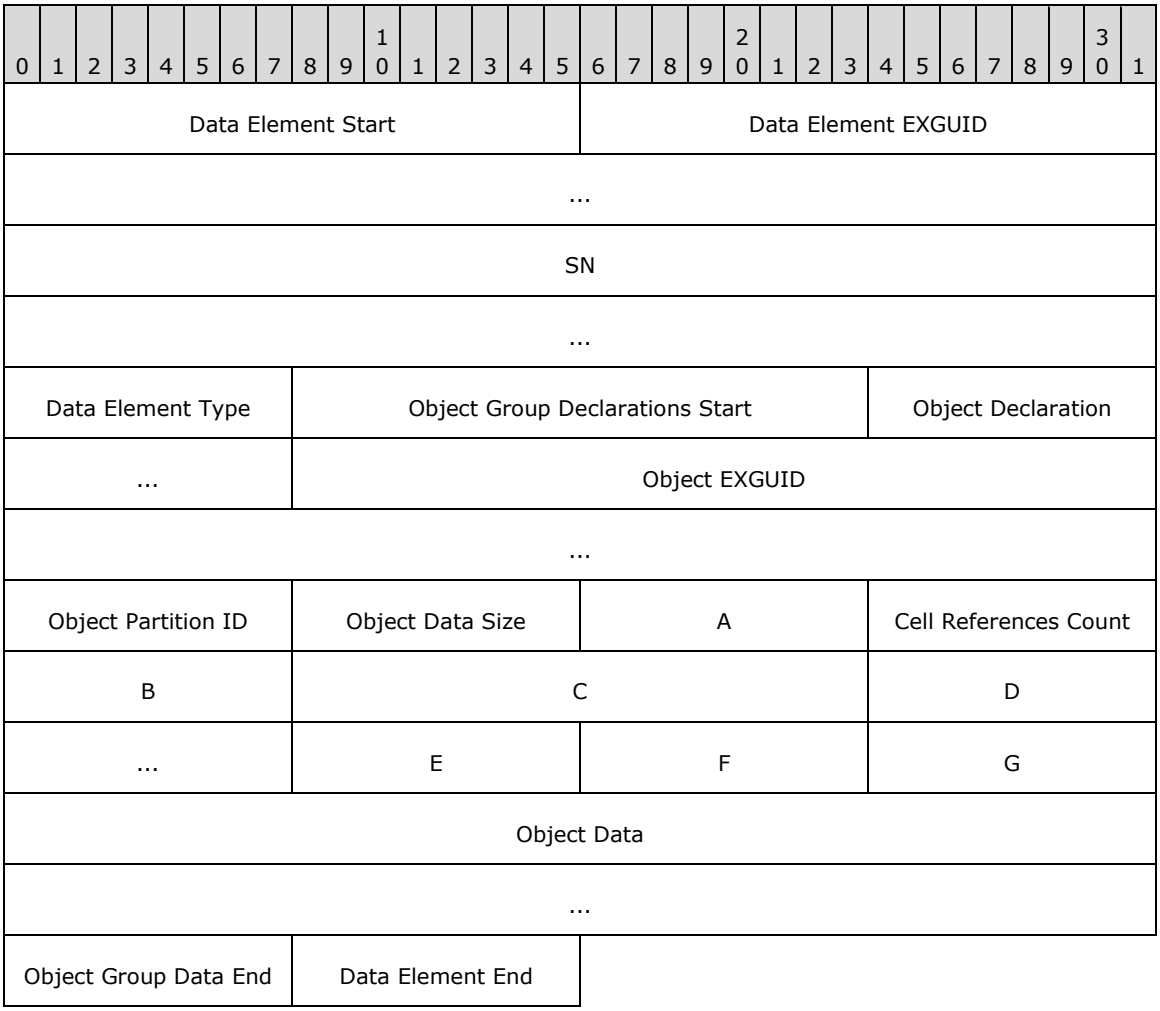
G - Object Data Length (1 byte): 0x59 specifies the number of bytes in **Object Data** as a compact unsigned 64-bit integer with a decoded value of 0x2C.

Object Data (44 bytes): Content of {50 4B 03 04 14 00 00 00 00 00 E5 AC 66 3E 82 89 D1 F7 05 00 00 00 05 00 00 00 09 00 00 00 48 65 6C 6C 6F 2E 74 78 74 48 65 6C 6C 6F} with the first 39 bytes being a zip local header and the last 5 bytes being the data file named Hello.

Object Group Data End (1 byte): 0x79 specifies the stream object header for an object group data end.

Data Element End (1 byte): 0x05 specifies the stream object header for a data element end.

```
00000340:                      0C  56 34 2F 16 61 BB 32 55
00000350: D4 4B 98 8B C6 87 B9 A9  85 8D 80 37 2D 91 05 80
00000360: B3 D4 4A 8E BE 9D EA 85  0F D5 C3 06 00 00 00 00
00000370: 00 00 00 0B EC 00 C0 32  80 EC BC 97 4D DC 28 C5
00000380: 41 92 74 26 CB 57 96 6F  17 06 00 00 12 03 59 00
00000390: 00 75 F4 00 B0 5E 00 00  59 50 4B 03 04 14 00 00
000003A0: 00 00 00 F0 AC 66 3E 47  3E B6 FB 05 00 00 00 05
000003B0: 00 00 00 09 00 00 00 57  6F 72 6C 64 2E 74 78 74
000003C0: 57 6F 72 6C 64 79 05
```



Data Element Start (2 bytes): 0x560C specifies the stream object header for a data element start, as described in [MS-FSSHTTPB] section 2.2.1.5.1. Decoded, this has a type of 0x1 and a length of 43 and is compound.

Data Element EXGUID: {BB61162F-5532-4BD4-988B-C687B9A9858D} 0x06 specifies the data element extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 34 2F 16 61 BB 32 55 D4 4B 98 8B C6 87 B9 A9 85 8D.

SN (25 bytes): {05912D37-B380-4AD4-8EBE-9DEA850FD5C3} 0x06 specifies the serial number decoded from 80 37 2D 91 05 80 B3 D4 4A 8E BE 9D EA 85 0F D5 C3 06 00 00 00 00 00 00 00.

Data Element Type (1 byte): 0x0B specifies the data element type as a compact unsigned 64-bit integer. Decoded, this represents a data element type of 0x5.

Object Group Declarations Start (2 bytes): 0x00EC specifies the stream object header for an object group declaration start. Decoded, this has a type of 0x1D and a length of zero and is compound.

Object Declaration (2 bytes): 0x32C0 specifies the stream object header for an object declaration start. Decoded, this has a type of 0x18 and a length of 25. This specifies the start of the **Leaf Node Object**, as specified in section 2.2.3.

Object EXGUID: {4D97BCEC-28DC-41C5-9274-26CB57966F17} 0x12000006 specifies the object extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 80 EC BC 97 4D DC 28 C5 41 92 74 26 CB 57 96 6F 17 06 00 00 12.

Object Partition ID (1 byte): 0x03 specifies an object partition identifier as a compact unsigned 64-bit integer with a decoded value of 0x01.

Object Data Size (1 byte): 0x59 specifies the size, in bytes, of the object as a compact unsigned 64-bit integer. Decoded, this represents 0x2C.

A - Object References Count (1 byte): 0x00 specifies the number of object references as a compact unsigned 64-bit integer with a decoded value of 0x00.

Cell References Count (1 byte): 0x00 specifies the number of cell references as a compact unsigned 64-bit integer with a decoded value of 0x00.

B - Object Group Declaration end (1 byte): 0x75 specifies the stream object header for an object group declaration end.

C - Cell Object Group Data Header (2 bytes): 0x00F4 specifies the stream object header for a cell object group data header. Decoded, this has a type of 0x1E, length 0, and is compound.

D - Cell Object Group Object Data (2 bytes): 0x5EB0 specifies the stream object header for a cell object group object data. Decoded, this has a type of 0x16 and a length of 47.

E - Object Extended GUID Array Count (1 byte): 0x00 specifies the number of extended GUIDs, as described in [MS-FSSHTTPB] section 2.2.1.7, as a compact unsigned 64-bit integer with a decoded value of 0x00.

F – Cell ID Array Count (1 byte): 0x00 specifies the number of **Cell IDs** as a compact unsigned 64-bit integer with a decoded value of 0x00.

G - Object Data Length (1 byte): 0x59 specifies the number of bytes in **Object Data** as a compact unsigned 64-bit integer with a decoded value of 0x2C.

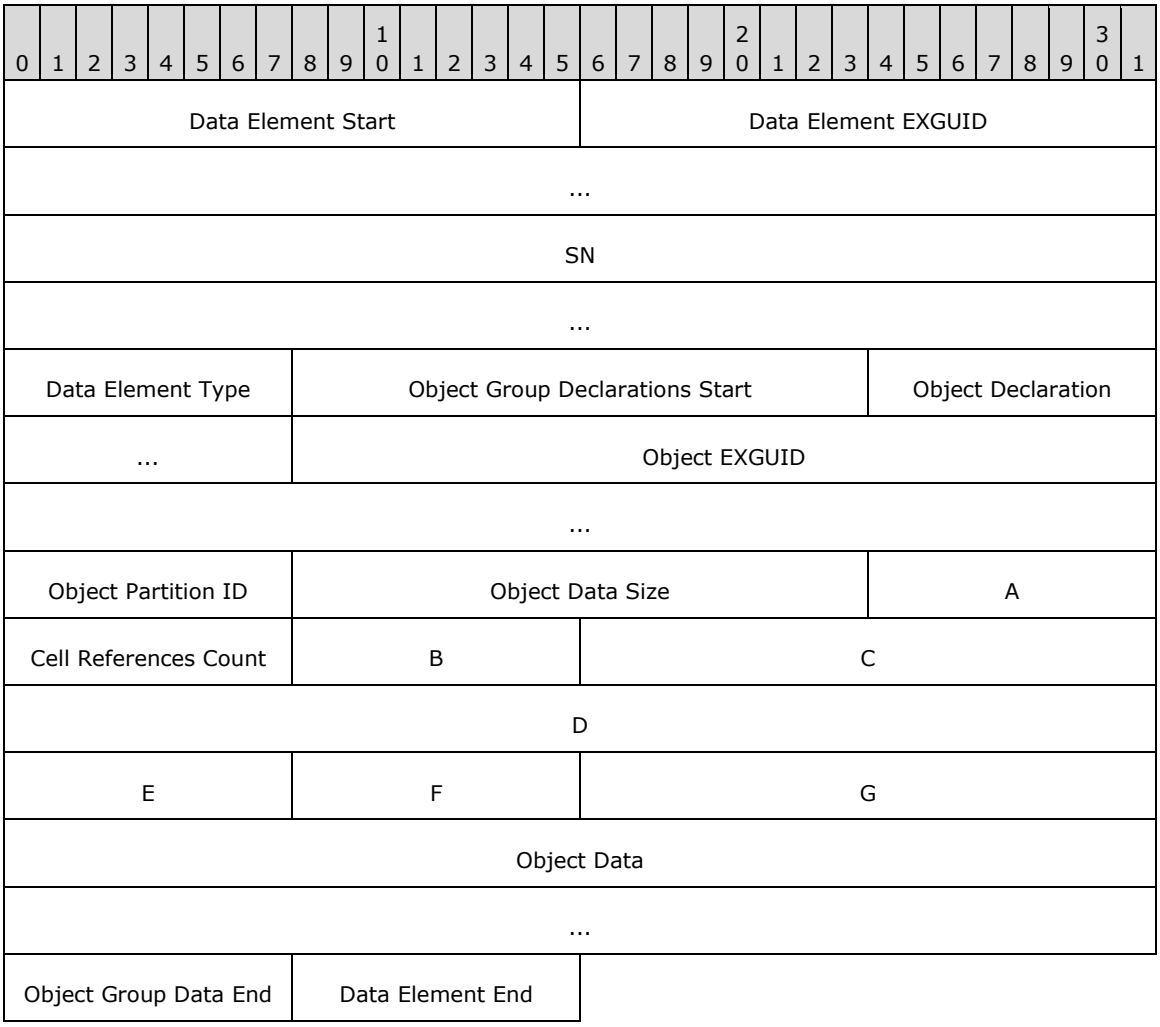
Object Data (44 bytes): Content of {50 4B 03 04 14 00 00 00 00 00 F0 AC 66 3E 47 3E B6 FB 05 00 00 00 05 00 00 00 09 00 00 00 57 6F 72 6C 64 2E 74 78 74 57 6F 72 6C 64} with the first 39 bytes being a zip local header and the last 5 bytes being the data file named World.

Object Group Data End (1 byte): 0x79 specifies the stream object header for an object group data end.

Data Element End (1 byte): 0x05 specifies the stream object header for a data element end.

```
000003C0:                      0C  56 3C 2F 16 61 BB 32 55
000003D0: D4 4B 98 8B C6 87 B9 A9  85 8D 80 37 2D 91 05 80
000003E0: B3 D4 4A 8E BE 9D EA 85  0F D5 C3 07 00 00 00 00
000003F0: 00 00 00 0B EC 00 C0 34  80 EC BC 97 4D DC 28 C5
00000400: 41 92 74 26 CB 57 96 6F  17 07 00 00 12 03 12 02
00000410: 00 00 75 F4 00 B2 00 10  01 00 00 12 02 50 4B 01
00000420: 02 14 00 14 00 00 00 00  00 E5 AC 66 3E 82 89 D1
00000430: F7 05 00 00 00 05 00 00  00 09 00 00 00 00 00 00
00000440: 00 01 00 20 00 00 00 00  00 00 00 48 65 6C 6C 6F
00000450: 2E 74 78 74 50 4B 01 02  14 00 14 00 00 00 00 00
00000460: F0 AC 66 3E 47 3E B6 FB  05 00 00 00 05 00 00 00
00000470: 09 00 00 00 00 00 00 00  01 00 20 00 00 00 2C 00
00000480: 00 00 57 6F 72 6C 64 2E  74 78 74 50 4B 05 06 00
00000490: 00 00 00 02 00 02 00 6E  00 00 00 58 00 00 00 00
```

```
000004A0: 00 79 05
```



Data Element Start (2 bytes): 0x560C specifies the stream object header for a data element start, as described in [MS-FSSHTTPB] section 2.2.1.5.1. Decoded, this has a type of 0x1 and a length of 43 and is compound.

Data Element EXGUID: {BB61162F-5532-4BD4-988B-C687B9A9858D} 0x07 specifies the data element extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 3C 2F 16 61 BB 32 55 D4 4B 98 8B C6 87 B9 A9 85 8D.

SN (25 bytes): {05912D37-B380-4AD4-8EBE-9DEA850FD5C3} 0x07 specifies the serial number decoded from 80 37 2D 91 05 80 B3 D4 4A 8E BE 9D EA 85 0F D5 C3 07 00 00 00 00 00 00 00.

Data Element Type (1 byte): 0x0B specifies the data element type as a compact unsigned 64-bit integer. Decoded this represents a data element type of 0x5.

Object Group Declarations Start (2 bytes): 0x00EC specifies the stream object header for an object group declaration start. Decoded this has a type of 0x1D and a length of zero and is compound.

Object Declaration (2 bytes): 0x34C0 specifies the stream object header for an object declaration start. Decoded, this has a type of 0x18 and a length of 26. This specifies the start of the **Data Node Object**.

Object EXGUID: {4D97BCEC-28DC-41C5-9274-26CB57966F17} 0x12000007 specifies the object extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 80 EC BC 97 4D DC 28 C5 41 92 74 26 CB 57 96 6F 17 07 00 00 12.

Object Partition ID (1 byte): 0x03 specifies an object partition identifier as a compact unsigned 64-bit integer with a decoded value of 0x01.

Object Data Size (2 byte): 0x0212 specifies the size of bytes of the object as a compact unsigned 64-bit integer. Decoded, this represents 0x84.

A - Object References Count (1 byte): 0x00 specifies the number of object references as a compact unsigned 64-bit integer with a decoded value of 0x00.

Cell References Count (1 byte): 0x00 specifies the number of cell references as a compact unsigned 64-bit integer with a decoded value of 0x00.

B - Object Group Declaration end (1 byte): 0x75 specifies the stream object header for an object group declaration end.

C - Cell Object Group Data Header (2 bytes): 0x00F4 specifies the stream object header for a cell object group data header. Decoded, this has a type of 0x1E, length 0, and is compound.

D - Cell Object Group Object Data (4 bytes): 0x011000B2 specifies the stream object header for a cell object group object data. Decoded, this has a type of 0x16 and a length 136.

E - Object Extended GUID Array Count (1 byte): 0x00 specifies the number of extended GUIDs, as described in [MS-FSSHTTPB] section 2.2.1.7, as a compact unsigned 64-bit integer with a decoded value of 0x00.

F – Cell ID Array Count (1 byte): 0x00 specifies the number of **Cell IDs** as a compact unsigned 64-bit integer with a decoded value of 0x00.

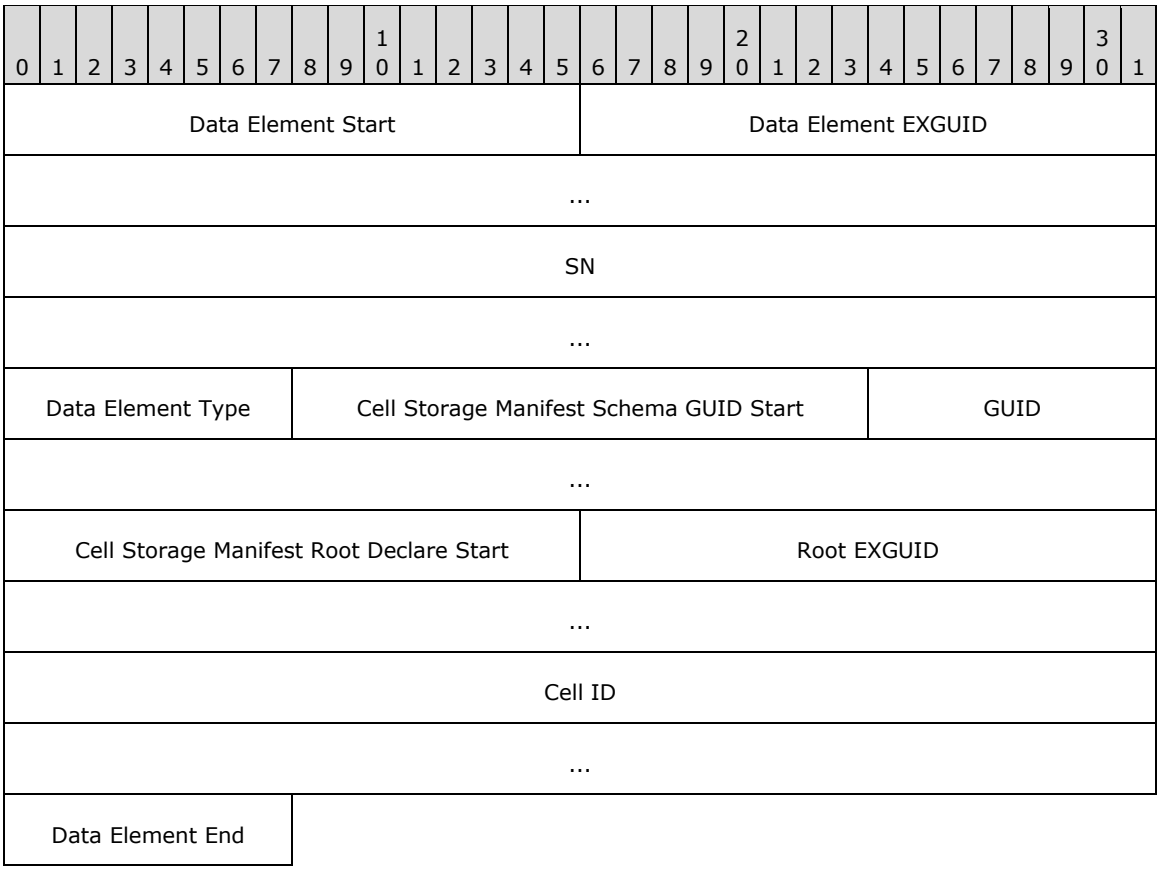
G - Object Data Length (2 byte): 0x0212 specifies the number of bytes in **Object Data** as a compact unsigned 64-bit integer with a decoded value of 0x84.

Object Data (132 bytes): Content of {50 4B 01 02 … 00 00 00 00} representing a zip central directory.

Object Group Data End (1 byte): 0x79 specifies the stream object header for an object group data end.

Data Element End (1 byte): 0x05 specifies the stream object header for a data element end.

```
000004A0:          0C 56 0C A0 93  65 66 4D 17 12 4F B0 45
000004B0: 83 1C 6A 44 BE 35 80 37  2D 91 05 80 B3 D4 4A 8E
000004C0: BE 9D EA 85 0F D5 C3 0A  00 00 00 00 00 00 00 05
000004D0: 60 20 94 33 B9 0E 1D 57  E9 41 AA D3 88 0D 92 D3
000004E0: 19 55 38 66 14 B9 FA DE  84 A3 AA 0D 4A A3 A8 52
000004F0: 0C 77 AC 70 73 0C B9 FA  DE 84 A3 AA 0D 4A A3 A8
00000500: 52 0C 77 AC 70 73 0C 65  46 2A 6F C8 42 C7 46 BA
00000510: B4 E2 8F DC E1 E3 2B 05
```



Data Element Start (2 bytes): 0x560C specifies the stream object header for a data element start. Decoded, this has a type of 0x1 and a length of 43 and is compound.

Data Element EXGUID: {666593A0-174D-4F12-B045-831C6A44BE35} 0x01 specifies a string representation of the data element extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 0C A0 93 65 66 4D 17 12 4F B0 45 83 1C 6A 44 BE 35.

SN: {05912D37-B380-4AD4-8EBE-9DEA850FD5C3} 0x0A specifies a string representation of the serial number decoded from 80 B3 D4 4A 8E BE 9D EA 85 0F D5 C3 0A 00 00 00 00 00 00 00.

Data Element Type (1 byte): 0x05 specifies the data element type as a compact unsigned 64-bit integer. Decoded, this represents a data element type of 0x2.

Cell Storage Manifest Schema GUID Start (2 bytes): 0x2060 specifies the stream object header for a cell storage manifest schema GUID. Decoded, this has a type of 0x0C and a length of 16.

GUID: {0EB93394-571D-41E9-AAD3-880D92D31955} specifies a string representation of the schema GUID.

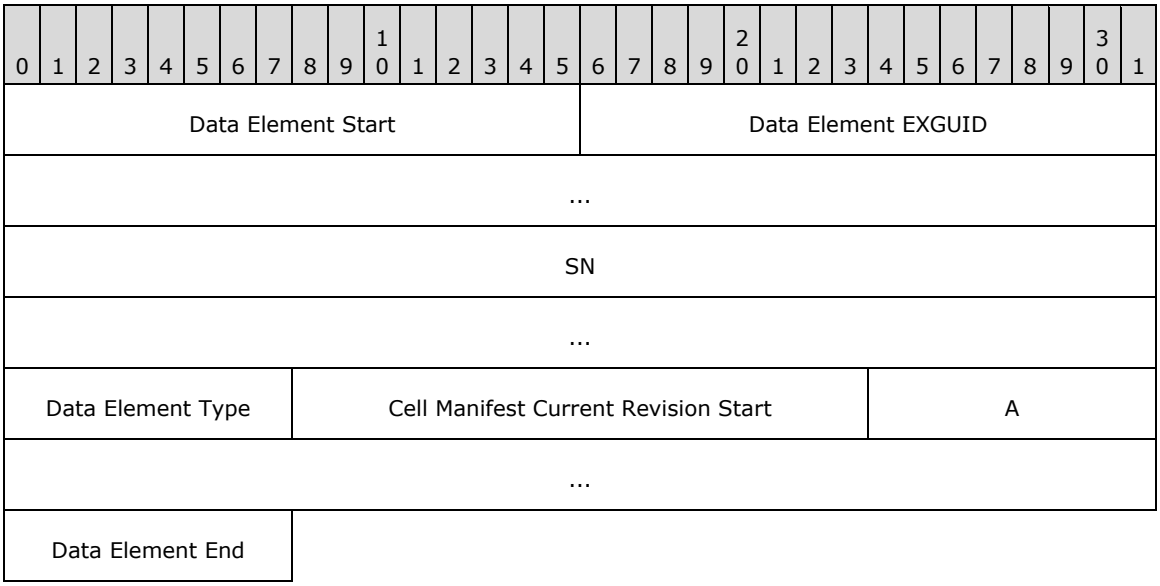
Cell Storage Manifest Root Declare (2 bytes): 0x6638 specifies the stream object header for a cell storage manifest root declare. Decoded, this has a type of 0x07 and a length of 51.

Root EXGUID: {84DEFAB9-AAA3-4A0D-A3A8-520C77AC7073} 0x02 specifies a string representation of the root storage manifest extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 14 B9 FA DE 84 A3 AA 0D 4A A3 A8 52 0C 77 AC 70 73.

Cell ID: {84DEFAB9-AAA3-52A8-0C77-520C77AC7073} 0x01, {6F2A4665-42C8-46C7-BAB4-E28FDCE1E32B} 0x01 specifies a string representation of the cell ID decoded from 0C B9 FA DE 84 A3 AA 0D 4A A3 A8 52 0C 77 AC 70 73 0C 65 46 2A 6F C8 42 C7 46 BA B4 E2 8F DC E1 E3 2B.

Data Element End (1 byte): 0x05 specifies the stream object header for a data element end.

```
00000510:                         0C 56 4C 2F 16 61 BB 32
00000520: 55 D4 4B 98 8B C6 87 B9  A9 85 8D 80 37 2D 91 05
00000530: 80 B3 D4 4A 8E BE 9D EA  85 0F D5 C3 0B 00 00 00
00000540: 00 00 00 00 07 58 22 0C  89 C3 0D 4D 66 5E 6E 4D
00000550: 88 C4 52 71 D5 B4 80 28  05
```



Data Element Start (2 bytes): 0x560C specifies the stream object header for a data element start. Decoded, this has a type of 0x1 and a length of 43 and is compound.

Data Element EXGUID: {BB61162F-5532-4BD4-988B-C687B9A9858D} 0x09 specifies a string representation of the data element Extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 4C 2F 16 61 BB 32 55 D4 4B 98 8B C6 87 B9 A9 85 8D.

SN: 0x80 {05912D37-B380-4AD4-8EBE-9DEA850FD5C3} 0x0B specifies the serial number decoded from 80 37 2D 91 05 80 B3 D4 4A 8E BE 9D EA 85 0F D5 C3 0B 00 00 00 00 00 00 00.

Data Element Type (1 byte): 0x07 specifies the data element type as a compact unsigned 64-bit integer. Decoded, this represents a data element type of 0x3.

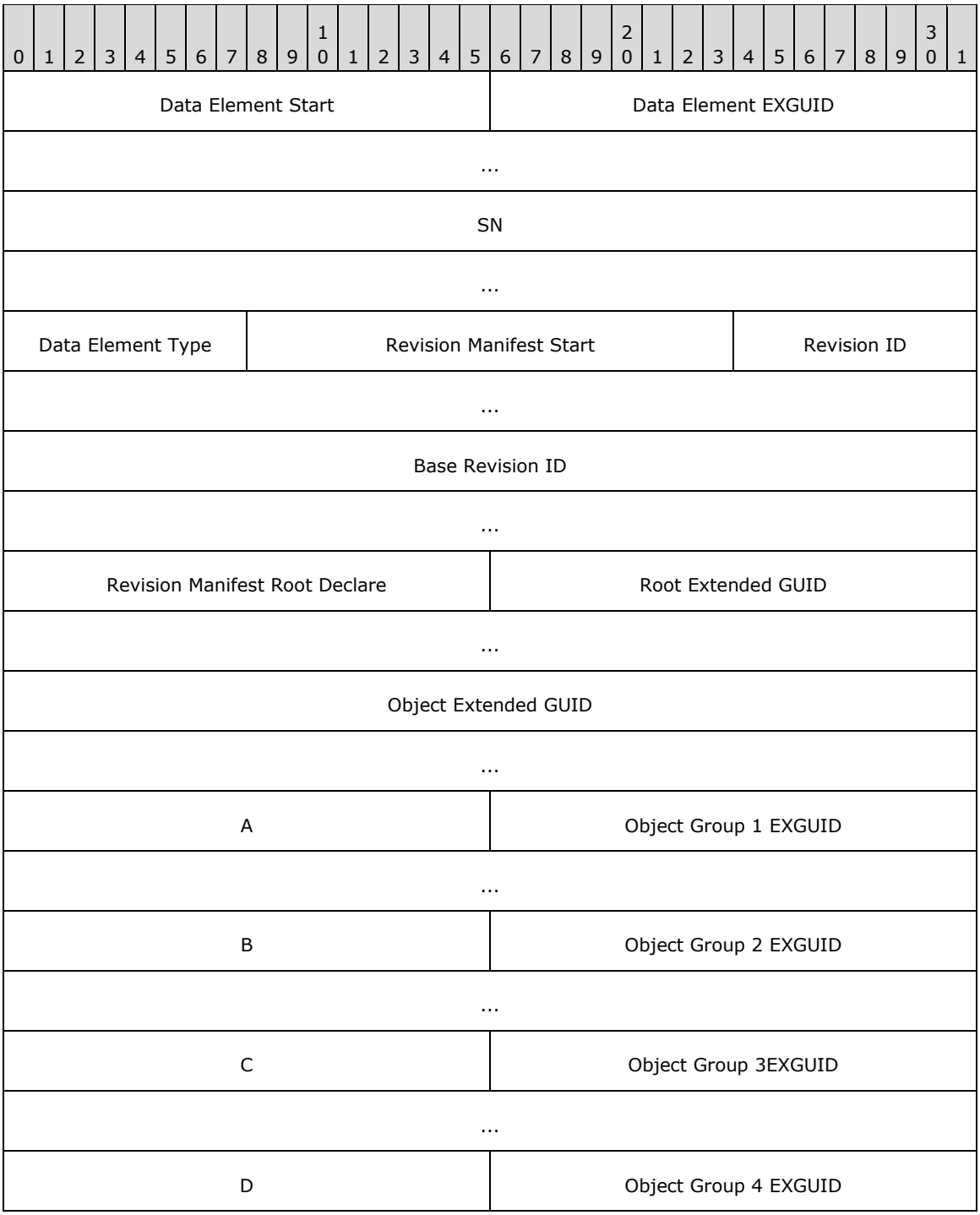
Cell Manifest Current Revision Start (2 bytes): 0x2258 specifies the stream object header for a cell manifest current revision start. Decoded, this has a type of 0x0B and a length of 17.

A - Cell Manifest Current Revision EXGUID: {4D0DC389-5E66-4D6E-88C4-5271D5B48028} 0x01 specifies a string representation of the current revision extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 0C 89 C3 0D 4D 66 5E 6E 4D 88 C4 52 71 D5 B4 80 28 .

Data Element End (1 byte): 0x05 specifies the stream object header for a data element end.

```
00000550:                            0C 56 0C 39 04 FD BE
00000560: 69 4B B0 4A 8D F9 A4 B5  EA 91 D5 B9 80 37 2D 91
00000570: 05 80 B3 D4 4A 8E BE 9D  EA 85 0F D5 C3 0C 00 00
00000580: 00 00 00 00 00 09 D0 24  0C 89 C3 0D 4D 66 5E 6E
00000590: 4D 88 C4 52 71 D5 B4 80  28 00 50 4C 14 B9 FA DE
000005A0: 84 A3 AA 0D 4A A3 A8 52  0C 77 AC 70 73 80 EC BC
000005B0: 97 4D DC 28 C5 41 92 74  26 CB 57 96 6F 17 01 00
000005C0: 00 11 C8 22 0C 2F 16 61  BB 32 55 D4 4B 98 8B C6
```

```
000005D0: 87 B9 A9 85 8D C8 22 14  2F 16 61 BB 32 55 D4 4B
000005E0: 98 8B C6 87 B9 A9 85 8D  C8 22 1C 2F 16 61 BB 32
000005F0: 55 D4 4B 98 8B C6 87 B9  A9 85 8D C8 22 24 2F 16
00000600: 61 BB 32 55 D4 4B 98 8B  C6 87 B9 A9 85 8D C8 22
00000610: 2C 2F 16 61 BB 32 55 D4  4B 98 8B C6 87 B9 A9 85
00000620: 8D C8 22 34 2F 16 61 BB  32 55 D4 4B 98 8B C6 87
00000630: B9 A9 85 8D C8 22 3C 2F  16 61 BB 32 55 D4 4B 98
00000640: 8B C6 87 B9 A9 85 8D 05
```



...	
E	Object Group 5 EXGUID
...	
F	Object Group 6 EXGUID
...	
G	Object Group 7 EXGUID
...	
Data Element End	

Data Element Start (2 bytes): 0x560C specifies the stream object header for a data element start. Decoded, this has a type of 0x1 and a length of 43 and is compound.

Data Element EXGUID: {BEFD0439-4B69-4AB0-8DF9-A4B5EA91D5B9} 0x01 specifies the data element extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 0C 39 04 FD BE 69 4B B0 4A 8D F9 A4 B5 EA 91 D5 B9.

SN: {05912D37-B380-4AD4-8EBE-9DEA850FD5C3} 0x0C specifies a string representation of the serial number decoded from 80 37 2D 91 05 80 B3 D4 4A 8E BE 9D EA 85 0F D5 C3 0C 00 00 00 00 00 00 00.

Data Element Type (1 byte): 0x09 specifies the data element type as a compact unsigned 64-bit integer. Decoded, this represents a data element type of 0x4.

Revision Manifest Start (2 bytes): 0x24D0 specifies the stream object header for revision manifest start. Decoded, this has type 0x1A, length 18.

Revision ID: {4D0DC389-5E66-4D6E-88C4-5271D5B48028} 0x01 specifies the revision identifier, in the form of an extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 0C 89 C3 0D 4D 66 5E 6E 4D 88 C4 52 71 D5 B4 80 28.

Base Revision ID: {00000000-0000-0000-0000-000000000000} 0x0 specifies the base revision identifier in the form of an extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 00.

Revision Manifest Root Declare: 0x4C50 specifies the stream object header for a revision manifest root declare. Decoded, this has a type of 0x0A and a length of 0x25.

Root Extended GUID: {84DEFAB9-AAA3-4A0D-A3A8-520C77AC7073}, 0x02 specifies the root extended GUID in the form of an extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 14 B9 FA DE 84 A3 AA 0D 4A A3 A8 52 0C 77 AC 70 73.

Object Extended GUID: {4D97BCEC-28DC-41C5-9274-26CB57966F17}, 0x11000001 specifies the root extended GUID in the form of an extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 80 EC BC 97 4D DC 28 C5 41 92 74 26 CB 57 96 6F 17 01 00 00 11.

A - Revision Manifest Object Group Reference Start (2 bytes): 0x22C8 specifies the stream object header for a revision manifest object group reference start. Decoded, this has a type of 0x19 and a length of 17.

Object Group 1 EXGUID: {BB61162F-5532-4BD4-988B-C687B9A9858D} 0x01 specifies the object group extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 0C 2F 16 61 BB 32 55 D4 4B 98 8B C6 87 B9 A9 85 8D.

B - Revision Manifest Object Group Reference Start (2 bytes): 0x22C8 specifies the stream object header for a revision manifest object group reference start. Decoded, this has a type of 0x19 and a length of 17.

Object Group 2 EXGUID: {BB61162F-5532-4BD4-988B-C687B9A9858D} 0x02 specifies the object group extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 14 2F 16 61 BB 32 55 D4 4B 98 8B C6 87 B9 A9 85 8D.

C - Revision Manifest Object Group Reference Start (2 bytes): 0x22C8 specifies the stream object header for a revision manifest object group reference start. Decoded, this has a type of 0x19 and a length of 17.

Object Group 3 EXGUID: {BB61162F-5532-4BD4-988B-C687B9A9858D} 0x03 specifies the object group extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 1C 2F 16 61 BB 32 55 D4 4B 98 8B C6 87 B9 A9 85 8D.

D - Revision Manifest Object Group Reference Start (2 bytes): 0x22C8 specifies the stream object header for a revision manifest object group reference start. Decoded, this has a type of 0x19 and a length of 17.

Object Group 4 EXGUID: {BB61162F-5532-4BD4-988B-C687B9A9858D} 0x04 specifies the object group extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 24 2F 16 61 BB 32 55 D4 4B 98 8B C6 87 B9 A9 85 8D.

E - Revision Manifest Object Group Reference Start (2 bytes): 0x22C8 specifies the stream object header for a revision manifest object group reference start. Decoded, this has a type of 0x19 and a length of 17.

Object Group 5 EXGUID: {BB61162F-5532-4BD4-988B-C687B9A9858D} 0x05 specifies the object group extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 2C 2F 16 61 BB 32 55 D4 4B 98 8B C6 87 B9 A9 85 8D.

F - Revision Manifest Object Group Reference Start (2 bytes): 0x22C8 specifies the stream object header for a revision manifest object group reference start. Decoded, this has a type of 0x19 and a length of 1.

Object Group 6 EXGUID: {"BB61162F-5532-4BD4-988B-C687B9A9858D} 0x06 specifies the object group extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 34 2F 16 61 BB 32 55 D4 4B 98 8B C6 87 B9 A9 85 8D.

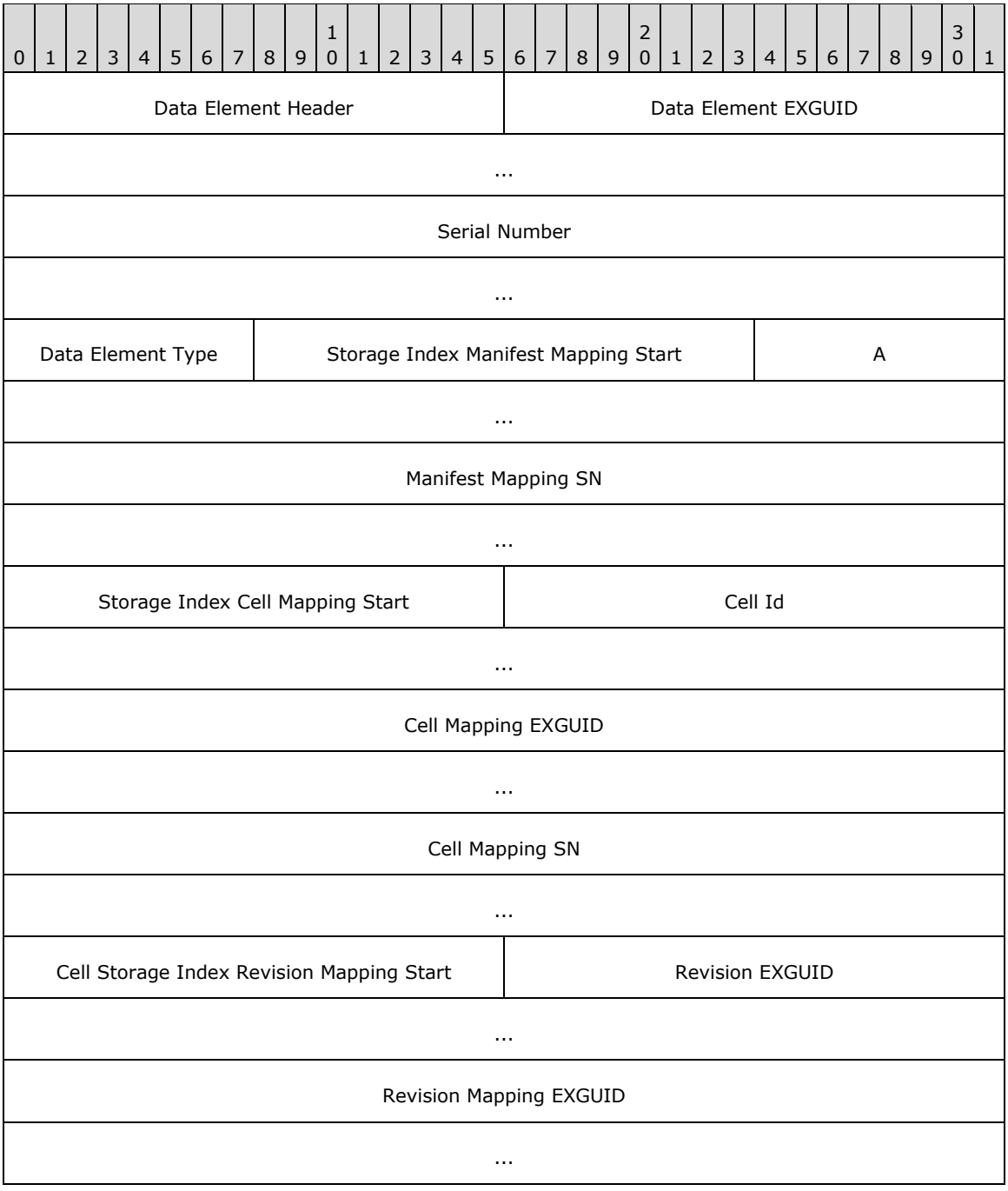
G - Revision Manifest Object Group Reference Start (2 bytes): 0x22C8 specifies the stream object header for a revision manifest object group reference start. Decoded, this has a type of 0x19 and a length of 17.

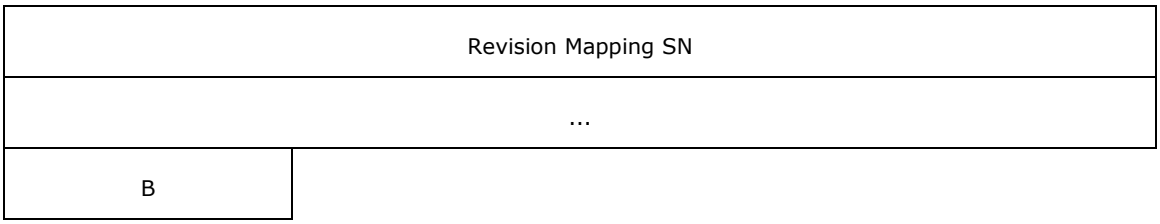
Object Group 7 EXGUID: {BB61162F-5532-4BD4-988B-C687B9A9858D} 0x07 specifies the object group extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 3C 2F 16 61 BB 32 55 D4 4B 98 8B C6 87 B9 A9 85 8D.

Data Element End (1 byte): 0x05 specifies the stream object header for a data element end.

```
00000640:                         0C 56 0C F8 DD BF 1E FA
00000650: 64 E7 4E A5 DB 61 44 7E  8A 8C C1 80 DB 35 CE 41
00000660: 06 A3 76 4D BA 08 A2 15  B4 A8 EA 05 01 00 00 00
00000670: 00 00 00 00 03 88 54 0C  A0 93 65 66 4D 17 12 4F
00000680: B0 45 83 1C 6A 44 BE 35  80 C8 D2 6E FA 7F 4C 2B
00000690: B5 8E BE 9D EA 85 0F D5  C3 19 00 00 00 00 00 00
000006A0: 00 70 98 0C B9 FA DE 84  A3 AA 0D 4A A3 A8 52 0C
```

```
000006B0: 77 AC 70 73 0C 65 46 2A  6F C8 42 C7 46 BA B4 E2
000006C0: 8F DC E1 E3 2B 4C 2F 16  61 BB 32 55 D4 4B 98 8B
000006D0: C6 87 B9 A9 85 8D 80 C8  D2 6E FA 7F 4C 2B B5 8E
000006E0: BE 9D EA 85 0F D5 C3 18  00 00 00 00 00 00 00 68
000006F0: 76 0C 89 C3 0D 4D 66 5E  6E 4D 88 C4 52 71 D5 B4
00000700: 80 28 0C 39 04 FD BE 69  4B B0 4A 8D F9 A4 B5 EA
00000710: 91 D5 B9 80 C8 D2 6E FA  7F 4C 2B B5 8E BE 9D EA
00000720: 85 0F D5 C3 17 00 00 00  00 00 00 00 05
```





Data Element Start (2 bytes): 0x560C specifies the stream object header for a data element start. Decoded, this has a type of 0x1 and a length of 43 and is compound.

Data Element EXGUID: {1EBFDDF8-64FA-4EE7-A5DB-61447E8A8CC1} 0x01 specifies the data element extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 0C F8 DD BF 1E FA 64 E7 4E A5 DB 61 44 7E 8A 8C C1.

SN: {41CE35DB-A306-4D76-BA08-A215B4A8EA05} 0x01 specifies the serial number decoded from 80 DB 35 CE 41 06 A3 76 4D BA 08 A2 15 B4 A8 EA 05 01 00 00 00 00 00 00 00.

Data Element Type (1 byte): 0x03 specifies the data element type as a compact unsigned 64-bit integer. Decoded, this represents a data element type of 0x1.

Storage Index Manifest Mapping Start (2 bytes): 0x5488 specifies the stream object header for a storage index cell mapping. Decoded, this represents a type of 0x11 and a length of 42.

A - Manifest Mapping EGUID: {666593A0-174D-4F12-B045-831C6A44BE35} 0x01 specifies the manifest mapping extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 0C A0 93 65 66 4D 17 12 4F B0 45 83 1C 6A 44 BE 35 80.

Manifest Mapping SN: {FA6ED2C8-4C7F-B52B-8EBE-9DEA850FD5C3} 0x19 specifies the manifest mapping serial number decoded from C8 D2 6E FA 7F 4C 2B B5 8E BE 9D EA 85 0F D5 C3 19 00 00 00 00 00 00 00.

Storage Index Cell Mapping Start (2 bytes): 0x9870 specifies the stream object header for a storage index cell mapping. Decoded, this has a type of 0x0E and a length of 76.

Cell Id: {84DEFAB9-AAA3-4A0D-A3A8-520C77AC7073} 0x01, {6F2A4664-42C8-46C7-BAB4-E28FDCE1E32B} 0x01 specifies the cell identifier decoded from 0C B9 FA DE 84 A3 AA 0D 4A A3 A8 52 0C 77 AC 70 73 0C 65 46 2A 6F C8 42 C7 46 BA B4 E2 8F DC E1 E3 2B.

Cell Mapping EXGUID: {BB61162F-5532-4BD4-988B-C687B9A9858D} 0x09 specifies the cell mapping extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 4C 2F 16 61 BB 32 55 D4 4B 98 8B C6 87 B9 A9 85 8D.

Cell Mapping SN: {FA6ED2C8-4C7F-B52B-8EBE-9DEA850FD5C3} 0x18 specifies the cell mapping serial number decoded from 80 C8 D2 6E FA 7F 4C 2B B5 8E BE 9D EA 85 0F D5 C3 18 00 00 00 00 00 00 00.

Cell Storage Index Revision Mapping Start (2 bytes): 0x7668 specifies the stream object header for a cell storage index revision mapping start. Decoded, this represents a type of 0x0D and a length of 59.

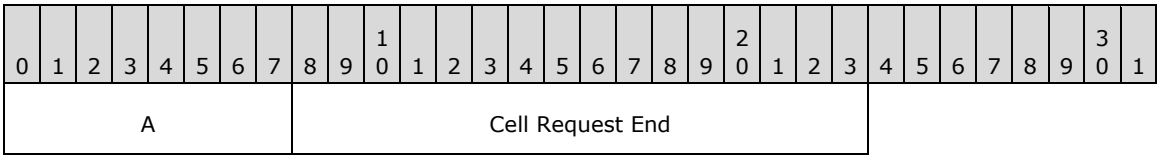
Revision EXGUID: {4D0DC389-5E66-4D6E-88C4-5271D5B48028} 0x01 specifies the revision extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 0C 89 C3 0D 4D 66 5E 6E 4D 88 C4 52 71 D5 B4 80 28.

Revision Mapping EXGUID: {BEFD0439-4B69-4AB0-8DF9-A4B5EA91D5B9} 0x01 specifies the revision mapping extended GUID, as described in [MS-FSSHTTPB] section 2.2.1.7, decoded from 0C 39 04 FD BE 69 4B B0 4A 8D F9 A4 B5 EA 91 D5 B9.

Revision Mapping SN: {FA6ED2C8-4C7F-B52B-8EBE-9DEA850FD5C3} 0x017 specifies the revision mapping serial number decoded from 80 C8 D2 6E FA 7F 4C 2B B5 8E BE 9D EA 85 0F D5 C3 17 00 00 00 00 00 00 00.

B - Data Element End (1 byte): 0x05 specifies the stream object header for a data element end.

```
00000720:                                       55 03 01
```



A - Data Element Package End (1 byte): 0x55 specifies the stream object header for a data element package end. This stream object was started in the request header, as detailed in section 3.1.1.

Cell Request End (2 bytes): 0x0103 specifies the stream object header for a cell request end. This stream object was started in the request header, as detailed in section 3.1.1.

4 Security

4.1 Security Considerations for Implementers

This protocol does not introduce any additional security considerations beyond those that apply to its containing protocol, as described in [MS-FSSHTTPB].

4.2 Index of Security Fields

None.

5 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include updates to those products.

- Microsoft Office 2010 suites
- Microsoft Office 2013
- Microsoft SharePoint Server 2010
- Microsoft SharePoint Server 2013
- Microsoft SharePoint Workspace 2010
- Windows 8.1 Update
- Microsoft Office 2016
- Windows 10 operating system
- Microsoft SharePoint Server 2016
- Microsoft Office 2019
- Microsoft SharePoint Server 2019
- Microsoft Office 2021
- Microsoft SharePoint Server Subscription Edition
- Microsoft Office 2024 Preview

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms "SHOULD" or "SHOULD NOT" implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term "MAY" implies that the product does not follow the prescription.

<1> Section 2.2.2: If there is only one object in the file, Microsoft SharePoint Server 2010 does not return the **Intermediate Node Object**, but a **Leaf Node Object** at the beginning.

<2> Section 2.2.3.1: SharePoint Server 2010 and SharePoint Server 2013 do not support the **Data Hash Header**.

<3> Section 2.2.3.1: The Data Hash computation is as follows:

A quick, simple non-cryptographic hash algorithm that works by XORing the bytes in a circular-shifting fashion.

A high level description of the algorithm without the introduction of the length is as follows:

Let's say a "block" is a 20-byte array.

```
method block zero():
  returns a block with all zero bits.

method block reverse(block b)
  returns a block with all of the bytes reversed (00010203... => ...03020100)
```

```
method block extend8(byte b):
  returns a block with all zero bits except for the lower 8 bits which come from b.

method block extend64(int64 i):
  returns a block of all zero bits except for the lower 64 bits which come from i
  and are in little-endian byte order.

method block rotate(block bl, int n):
  returns bl rotated left by n bits.

method block xor(block bl1, block bl2):
  returns a bitwise xor of bl1 with bl2

method block XorHash0(byte[] rgb):
  block ret = zero()
  for (int i = 0; i < rgb.Length; i ++)
    ret = xor(ret, rotate(extend8(rgb[i]), i * 11))
  returns reverse(ret)
```

<4> Section 2.4.1: The signature is stored in the Leaf node corresponding to the **local file header** chunk.

<5> Section 2.4.1: The signature is stored in the node corresponding to the **data file** chunk which is a child of the root Intermediate node.

<6> Section 2.4.1: Office 2010 March 2015 CU adds an additional unsigned 32-bit integer, which specifies the CRC of the compressed data, after **CRC** field. It is available in Office 2010 March 2015 CU and later.

<7> Section 2.4.1: This sequence of bytes is randomly generated in Microsoft products.

<8> Section 2.4.1: For SharePoint Server 2010, if the number of .ZIP file bytes represented by a chunk is greater than 1 megabyte, a list of subchunks is generated. Each subchunk represents a sequential chunk of the .ZIP file data. The size of each subchunk is at most 1 megabyte. All but the last subchunk is 1 megabyte in size.

<9> Section 2.4.1: This sequence of bytes is randomly generated in Microsoft products.

<10> Section 2.4.2: SharePoint Server 2010 performs the RDC analysis if the file size is more than or equal to 32768 bytes and less than 262,144,000 bytes.

6 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements.
- A document revision that captures changes to protocol functionality.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

Section	Description	Revision class
2.4.1 Zip Files	Updated the product behavior.	Minor

7 Index

S

T

V

Z