

[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 for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **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 may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft's Open Specification Promise (available here: <http://www.microsoft.com/interop/osp>) or the Community Promise (available here: <http://www.microsoft.com/interop/cp/default.mspx>). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplq@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events 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 specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do 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 are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

Date	Revision History	Revision Class	Comments
03/28/2011	0.1	New	Released new document.
06/10/2011	1.0	Major	Significantly changed the technical content.

Table of Contents

1	Introduction	4
1.1	Glossary	4
1.2	References.....	4
1.2.1	Normative References.....	4
1.2.2	Informative References	4
1.3	Structure Overview (Synopsis)	4
1.3.1	Schema overview	4
1.3.2	Byte Ordering	5
1.4	Relationship to Protocols and Other Structures	5
1.5	Applicability Statement.....	5
1.6	Versioning and Localization	5
1.7	Vendor-Extensible Fields.....	5
2	Structures	6
2.1	Transport.....	6
2.2	Object Definitions	6
2.2.1	Common Node Object Properties	6
2.2.2	Root Node Object	7
2.2.2.1	Root Node Object Data	7
2.2.2.2	Root Node Object References	8
2.2.2.3	Root Node Object Cell References	8
2.2.3	Intermediate Node Object	8
2.2.3.1	Intermediate Node Object Data	8
2.2.3.2	Intermediate Node Object References	9
2.2.3.3	Intermediate Node Object Cell References	9
2.2.4	Data Node Object.....	9
2.2.4.1	Data Node Object Data.....	9
2.2.4.2	Data Node Object References.....	9
2.2.4.3	Data Node Object Cell References.....	9
2.3	Cell Properties	9
2.4	File Chunking	10
2.4.1	Zip Files	10
2.4.2	RDC Analysis	13
2.4.2.1	Generating chunks.....	13
2.4.2.2	Generating signatures	13
2.4.3	Simple Chunking Method	13
3	Structure Examples	15
3.1	Example Put.....	15
3.1.1	Request Header	17
3.1.2	Object Groups.....	19
4	Security Considerations.....	45
5	Appendix A: Product Behavior.....	46
6	Change Tracking.....	47
7	Index	49

1 Introduction

1.1 Glossary

The following terms are defined in [\[MS-GLOS\]](#):

**GUID
little-endian**

The following terms are specific to this document:

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

1.2 References

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. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

[MS-FSSHTTPB] Microsoft Corporation, "[Binary Requests for File Synchronization via SOAP Protocol Specification](#)"

[MS-RDC] Microsoft Corporation, "[Remote Differential Compression Algorithm Specification](#)".

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

1.2.2 Informative References

[MS-FSSHTTP] Microsoft Corporation, "[File Synchronization via SOAP over HTTP Protocol Specification](#)"

[MS-GLOS] Microsoft Corporation, "[Windows Protocols Master Glossary](#)".

1.3 Structure Overview (Synopsis)

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 that offers, 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 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 must be stored/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 sub-tree that further divides the region.

Chunking schemes are defined for files specified in the Zip file format and for files for which no structure is available using the Microsoft FilterMax algorithm.

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 Binary Requests for File Synchronization via SOAP Protocol, as described in [\[MS-FSSHTTPB\]](#).

1.5 Applicability Statement

This protocol is intended for use where incremental updates and efficient transmission of file data are desired features of client server file synchronization. This protocol is designed for use with file formats which are not natively represented in [\[MS-FSSHTTPB\]](#).

1.6 Versioning and Localization

None.

1.7 Vendor-Extensible Fields

None.

2 Structures

2.1 Transport

This protocol uses File Synchronization via SOAP over HTTP protocol as specified in [\[MS-FSSHTTPB\]](#).

2.2 Object Definitions

A Root Node Object, Intermediate Node Object and Data Node Object are using 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 Data Start** field MUST be set as shown in the following table, as specified in [\[MS-FSSHTTPB\]](#) section 2.2.1.12.6.3.

Field	Root	Intermediate	Data
Object Data	As specified in in [MS-FSSHTTPB]	As specified in in [MS-FSSHTTPB]	As specified in in [MS-FSSHTTPB]
Object Extended GUID Array	Specifies an ordered list of the Object Extended GUIDs for each child of the Root Node.	Specifies an ordered list of the Object Extended GUIDs for each child of this node.	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 declaration field of the object group data 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 "1".	A compact unsigned 64-bit integer that MUST be "1".	A compact unsigned 64-bit integer that MUST be "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.

Field	Root	Intermediate	Data
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].

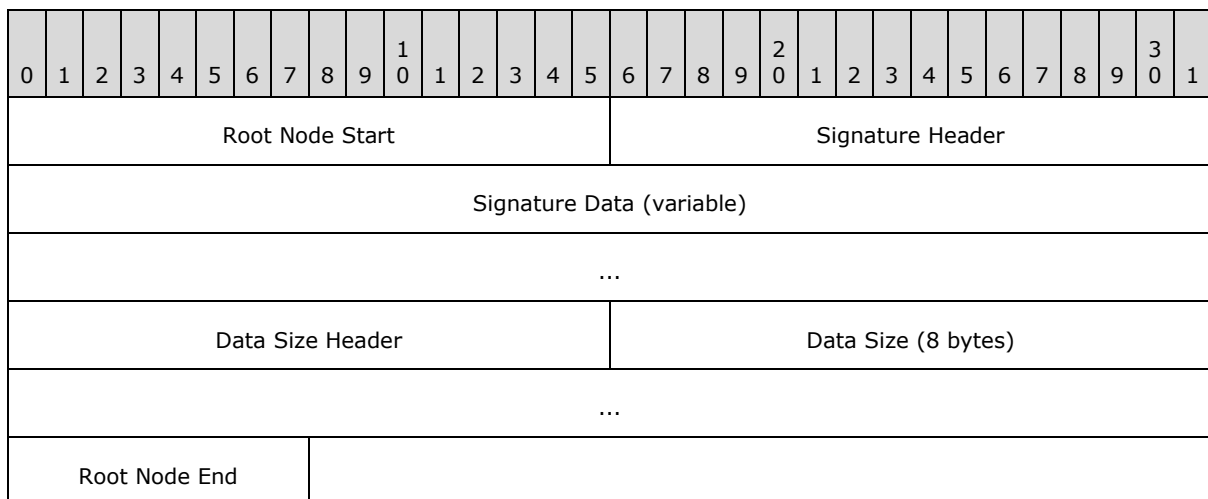
2.2.2 Root Node Object

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

A Root Node Object has the following format for the Root Node Object Data, Root Node Object References and Root Node Object Cell References.

2.2.2.1 Root Node Object Data

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



Root Node Start (2 bytes): A 16-bit stream object header that specifies a compound parse type and a stream object type of 0x20. The value of this field MUST be 0x0104.

Signature Header (2 bytes): A 16-bit stream object header that specifies a single object parse type and a stream object type of 0x21. The value of this field MUST be 0x308.

Signature Data (variable): A binary item as specified in [MS-FSSHTTPB] section 2.2.1.3 that specifies a value that is unique to the file data represented by this root 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 that specifies a single object parse type and a stream object type of 0x22.

Data Size (8 bytes): An unsigned 64-bit integer that specifies the size of the file data represented by this root node object.

Root Node End (1 byte): An 8-bit stream object header end that specifies a stream object of type 0x20. The value of this field MUST be 0x81.

2.2.2.2 Root Node Object References

The Root Node Object’s object reference array MUST specify an ordered set of Object IDs. Each Object ID MUST specify an Intermediate Node Object. The sum of the Data Size values from each of the intermediate node objects MUST equal the Data Size specified in the Object Data of this Root Node Object.

2.2.2.3 Root Node Object Cell References

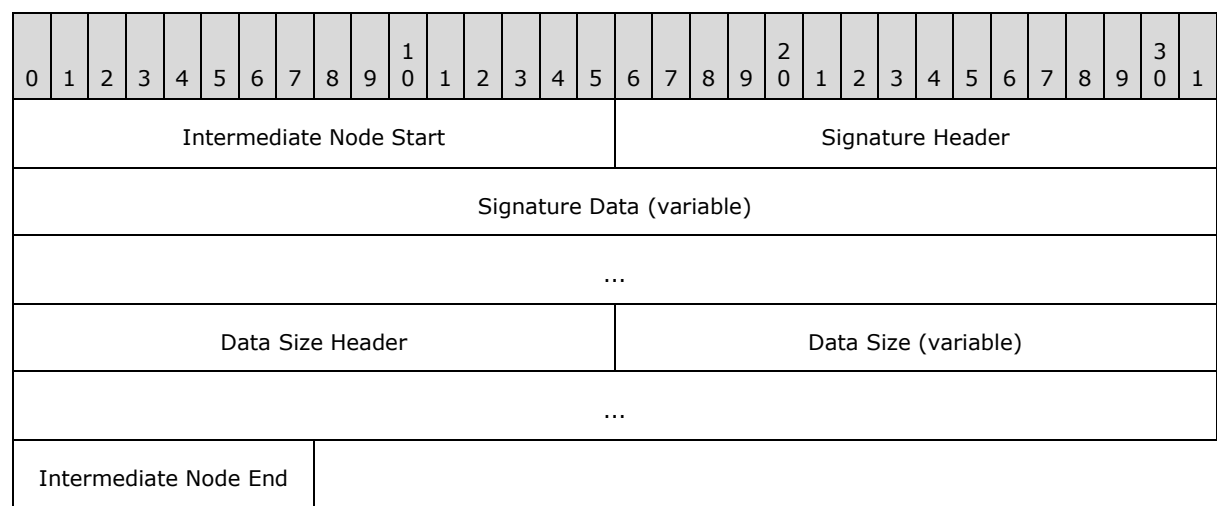
The object’s cell reference array MUST specify an empty array.

2.2.3 Intermediate Node Object

An intermediate node object has the following format for the object data, object references and object cell references.

2.2.3.1 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 that specifies a compound parse type and either a stream object type of 0x1F. The value of this field MUST be 0xFC.

Signature Header (2 bytes): A 16-bit stream object header that specifies a single object parse type and a stream object type of 0x21. The value of this field MUST be 0x308

Signature Data (variable): A binary item as specified in [\[MS-FSSHTTPB\]](#) section 2.2.1.3 that specifies 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](#).

Data Size Header (2 bytes): A 16-bit stream object header that specifies a single object parse type and a stream object type of 0x22.

Data Size (variable): A compact 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 that specifies a stream object type of 0x1F.

2.2.3.2 Intermediate Node Object References

The Intermediate Node Object's object reference array **MUST** specify an ordered set of Object IDs. The ordered set of Object IDs **MUST** contain the Object ID of a single Data Node Object, or an ordered list of Intermediate Node Object IDs. The size of the Data Node Object or the sum of the Data Size values from each of the Intermediate Node Objects **MUST** equal the Data Size specified in the Object Data of this Intermediate Node Object.

2.2.3.3 Intermediate Node Object Cell References

The object's cell reference array **MUST** specify an empty array.

2.2.4 Data Node Object

A data node object has the following format 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 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's object reference array **MUST** specify an empty array.

2.2.4.3 Data Node Object Cell References

The object's cell reference array **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 EB93394-571D-41E9-AAD3880D92D31955.

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, as listed in the following table.

Type	Value	GUID
4	1	84DEFAB9-AAA3-4A0D-A3A8520C77AC7073
4	1	6F2A4665-42C8-46C7-BAB4E28FDCE1E32B

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, as shown in the following table.

Type	Value	GUID
4	2	84DEFAB9-AAA3-4A0D-A3A8520C77AC7073

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 representing the primary content stream as shown in the following table.

Type	Value	GUID
4	2	84DEFAB9-AAA3-4A0D-A3A8520C77AC7073

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 **Root Node** MUST be specified 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 which are sequential and adjacent, and which reference the entire contents of the file.

Each chunk contains unique Signature Data as specified by section [2.2.3.1](#).

File data is passed to the following chunking methods:

1. .ZIP algorithm
2. RDC Analysis
3. Simple algorithm

2.4.1 Zip Files

If the first 4 bytes of the file match the **local file header signature** as specified in section V, subsection A of [appnote.txt], then the .ZIP analysis is used if the file is a valid ZIP file as specified in [appnote.txt]. If the file does not comply with the ZIP format, then the RDC analysis as specified in section [2.4.2](#) or the Simple Chunking method as specified in section [2.4.3](#) are used.

.ZIP files are split into chunks based on information in each **local file header** as specified in [appnote.txt]. Analysis of **local file headers** produces file chunk boundaries at the start of the **local file header** as specified in [appnote.txt], the start of the **data file** as specified in the [appnote.txt] and after the **data file** as specified in [appnote.txt], producing 2 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 following structure:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Local File Header Hash (20 bytes)																															

...

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

The signature for the data file chunk has the following structure:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
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 [appnote.txt].

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 [appnote.txt] unless the **local file header extra field** as specified in [appnote.txt] includes a **Zip64 Extended Information Extra Field** as specified in [appnote.txt] in which case it MUST be the value of the **compressed size** field in the **Zip64 Extended Information Extra Field** as specified in [appnote.txt].

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 [appnote.txt] unless the **local file header extra field** as specified in [appnote.txt] includes a **Zip64 Extended Information Extra Field** as specified in [appnote.txt] in which case it MUST be the value of the **uncompressed size** field in the **Zip64 Extended Information Extra Field** as specified in [appnote.txt].

If the combined size in bytes of the **local file header chunk** and **data file chunk** is less than or equal to 4096, then a single chunk is produced with a signature that is the local file header chunk signature followed by the data file chunk signature.

Analysis of chunks into "local file header", "data file" chunks continues at the file location after the current data file until one of the follow conditions occurs:

1. The extent of the data file as specified in the local file header would extend past the end of the file.
2. A sequence other than a local file header signature as specified in [appnote.txt] is found.

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

Once 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:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Small Final Chunk Signature (20 bytes)																															
...																															

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 1Mb the signature for the final chunk has the structure:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Large Final Chunk Signature (12 bytes)																															
...																															

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

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

If the number of .ZIP file bytes represented by a chunk is greater than 1 megabyte, a list of sub chunks is generated. Each sub chunk represents a sequential chunk of the .ZIP file data. The size of each sub chunk is at most 1 megabyte. All but the last sub chunk **MUST** be 1Mb in size. The total size of all sub chunks **MUST** equal the **Data Size** of the parent **Intermediate Node Object**.

The signature for these sub chunks has the following structure:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Sub Chunk Signature (8 bytes)																															
...																															

Sub Chunk Signature: An 8 byte sequence of bytes that specifies the sub chunk signature. This sequence of bytes **MUST** be unique.

For each sub chunk an **Intermediate Node Object** is created as specified in section [2.2.3](#). The parent **Intermediate Node Object** of the sub chunk MUST have its **Object Reference Array** include one **Object ID** entry for each sub chunk and these **Object ID** entries MUST be order based on the sequential .ZIP file bytes represented by each chunk.

For all **Intermediate Node Objects** with **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 **Intermediate 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 where the first 4 bytes of the file do not match the local file header signature as specified in [apnote.txt]. It is only performed if the file size is at least 32,768 bytes and less than 262,144,000 bytes.

2.4.2.1 Generating chunks

Files are split into chunks using the FilterMax algorithm described in [\[MS-RDC\]](#) section 3.1.5.1, using a hash window of 48 and a horizon of 16384, with the following exceptions:

Given a file F of length n consisting of bytes $b_0 \dots b_{n-1}$, byte b_i is a local maximum only if $i > \text{horizon}$ and for all $j \neq i$ where $i - \text{horizon} < j < i + \text{horizon}$, the H3 hash value after adding b_i is greater than that after adding any b_j .

Local maximums found within the last $(n \text{ modulus } 16,384) + 16,384$ 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 described in [\[MS-RDC\]](#) section 3.1.5.2

2.4.3 Simple Chunking Method

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

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Simple Small File Hash (20 bytes)																															
...																															

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:

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Simple Large File Hash (12 bytes)																															
...																															

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

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

For all **Intermediate 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 **Intermediate 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 Example Put

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 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 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
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 is the request header for a put.

```

Header:
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

```

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Protocol Version																Minimum Version															
Signature																															
...																															
Cell Request Start																															
User Agent Start																															
User Agent GUID																															
GUID																															
...																															
User Agent Version																															
Version																															
User Agent End																Sub-request Start															
...																Request ID								Request Type							
Priority								Put Changes Request																							
...								Storage Index EXGUID																							

...		
A	Put Changes Flags	Sub-request End
Data Element Package Start		Reserved

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 has a type of 0x40, length 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, length 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, length 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 user agent version. Decoded this has a type of 0x2F, length 4.

Version (4 bytes): 0x2EE0A395 specifies the version of the 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 sub-request start. Decoded this has a type of 0x42, length 3.

Request Id (2 bytes): 0x03 specifies the request number as a compact unsigned 64-bit **integer** for this request. Decoded this represents a value of 0x1.

Request Type (2 bytes): 0x0B specifies the request type as a compact unsigned 64-bit **integer**. Decoded this represents a value of 0x05.

Priority (2 bytes): 0x00 specifies the priority of this sub-request as a compact unsigned 64-bit **integer**.

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

Storage Index EXGUID: {"1EBFDDF8-64FA-4EE7-A5DB-61447E8A8CC1"} 0x01 specifies the Storage Index extended GUID, as specified 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 specified 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 sub-request 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, length 1, compound.

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

3.1.2 Object Groups

The following example is the object groups for a put.

```

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
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

```

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31				
Data Element Start																Data Element EXGUID																			
...																																			
SN																																			
...																																			
Data Element Type				Object Group Declarations Start																Object Declaration															
...				Object EXGUID																															
...																																			
Object Partition ID				Object Data Size								A								Cell References Count															
B				C								Reserved								D															
				E								Object Reference EXGUID 1																							
...																																			
Object Reference EXGUID 2																																			

...			
Object Reference EXGUID 3			
...			
F	G	Root Node Start	
Signature Header		H	I
...	Data Size		
...			
...	J	Object Group Data End	Data Element End

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"} "0x01" specifies the Data element extended GUID, as specified 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 object group declaration start. Decoded, this has a type of "0x1D", length 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", length 25. This specifies the start of the root node object, as specified in section [2.2.2](#).

Object EXGUID: {"4D97BCEC-28DC-41C5-9274-26CB57966F17"} "0x11000001" specifies the object extended GUID, as specified 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 of 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 (1 byte): "0x00F4" specifies the stream object header for a cell object group data header. Decoded, this has a type of "0x0E", length 0x01.

Reserved (1 byte): Set to "0x00".

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", length 82.

E - Object Extended GUID Array Count (1 byte): "0x07" specifies the number of Extended GUIDs, as specified 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 specified 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. This references the first intermediate 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 specified 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. This references the second intermediate node object.

Object Reference EXGUID 3: {"4D97BCEC-28DC-41C5-9274-26CB57966F17"} "0x12000004" specifies the object extended GUID, as specified 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. This references the third intermediate 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 count of bytes of **Object Data**, as specified in [\[MS-FSSHTTPB\]](#) section 2.2.1.12.6.3, as a compact unsigned 64-bit **integer** with a decoded value of "0x10".

Root 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](#).

H - Signature Data (1 byte): "0x00" specifies a binary item representing the **Signature Data** for this node. Decode this as length zero, content {}.

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", length 0x08.

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

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

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

Data Element End (1 byte): "0x05" specifies the stream object header for 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

```

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Data Element Start																Data Element EXGUID															
...																															
SN																															
...																															
Data Element Type										Object Group Declarations Start																Object Declaration					
...										Object EXGUID																					
...																															
Object Partition ID										Object Data Size										A						Cell References Count					
B										C										Reserved						D					
										E										Object Reference EXGUID											
...																															
F										G										Root Node Start											
Signature Header																Signature Data															
...																															

Data Size Header		Data Size	
...			
...		H	Object Group Data End
Data Element End			

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"} "0x02" specifies the Data element extended GUID, as specified 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 intermediate node object, as specified in section [2.2.3](#).

Object EXGUID: {"4D97BCEC-28DC-41C5-9274-26CB57966F17"} "0x12000002" specifies the object extended GUID, as specified 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 (1 byte): "0x00F4" specifies the stream object header for a cell object group data header. Decoded, this has a type of "0x0E", length 0x01.

Reserved (1 byte): Set to "0x00".

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 specified 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 specified 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".

Intermediate 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 type "0x21", length 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 intermediate node object.

H - Intermediate 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 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 10 11 2C
00000230: 00 00 00 00 00 00 00 7D 79 05
```


0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Data Element Start																Data Element EXGUID															
...																															
SN																															
...																															
Data Element Type								Object Group Declarations Start																Object Declaration							
...								Object EXGUID																							
...																															
Object Partition ID								Object Data Size								A								Cell References Count							
B								C								Reserved								D							
								E								Object Reference EXGUID															
...																															
F								G								Root Node Start															
Signature Header																Signature Data															
...																															
Data Size Header																Data Size															
...																															
...																H								Object Group Data End							
Data Element End																															

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.

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 **Intermediate Node Object**, as specified in section [2.2.3](#).

Object EXGUID: {"4D97BCEC-28DC-41C5-9274-26CB57966F17"} "0x12000003" specifies the object extended GUID, as specified 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 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 (1 byte): "0x00F4" specifies the stream object header for a cell object group data header. Decoded this has a type of "0x0E", length 0x01.

Reserved (1 byte): Set to "0x00".

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 specified 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 specified 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 count of bytes of **Object Data** as a compact unsigned 64-bit **integer** with a decoded value of "0x38".

Intermediate 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 type "0x21", length 0x29

Signature Data (41 bytes): "0x51" specifies a binary item representing the **Signature Data** for this node. Decoded, this has length 0x28, content {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}.

Data Size Header (2 bytes): "0x1110" specifies a 16-bit stream object header, as specified in section [2.2.3](#), that, decoded, specifies a single object parse type with a stream object type of "0x22", length 0x08.

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

H - Intermediate 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 object group data end.

Data Element End (1 byte): "0x05" specifies the stream object header for 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
```

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
Data Element Start																Data Element EXGUID																	
...																																	
SN																																	
...																																	
Data Element Type								Object Group Declarations Start																Object Declaration									
...								Object EXGUID																									
...																																	

Object Partition ID	Object Data Size	A	Cell References Count
B	C	Reserved	D
	E	Object Reference EXGUID	
...			
F	G	Root Node Start	
Signature Header		Signature Data	
...			
Data Size Header		Data Size	
...			
...		H	Object Group Data End
Data Element End			

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"} "0x04" specifies the Data element extended GUID, as specified 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 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 intermediate node object, as specified in section [2.2.3](#).

Object EXGUID: {"4D97BCEC-28DC-41C5-9274-26CB57966F17"} "0x12000004" specifies the object extended GUID, as specified 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 of 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 (1 byte): "0x00F4" specifies the stream object header for a cell object group data header. Decoded, this has a type of "0x0E", length 0x01.

Reserved (1 byte): Set to "0x00".

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", length 60.

E - Object Extended GUID Array Count (1 byte): "0x03" specifies the number of Extended GUIDs, as specified 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 specified 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 count of bytes of **Object Data** as a compact unsigned 64-bit **integer** with a decoded value of "0x24".

Intermediate 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 type "0x21", length 0x15.

Signature Data (21 bytes): "0x29" specifies a binary item representing the signature data for this node. Decoded, this has length 0x14, content {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](#), that, decoded, specifies a single object parse type with a stream object type of "0x22", length 0x08.

Data Size (8 bytes): "0x0000000000000084" specifies an unsigned 64-bit **integer** that specifies the size of the file data represented by this intermediate node object.

H - Intermediate 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 object group data end.

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

```

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Data Element Start																Data Element EXGUID															
...																															
SN																															
...																															
Data Element Type				Object Group Declarations Start																Object Declaration											
...				Object EXGUID																											
...																															
Object Partition ID				Object Data Size												A								Cell References Count							
B				C												Reserved								D							
...				E												F								G							
Object Data																															
...																															
Object Group Data End				Data Element End																											

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"} "0x05" specifies the **Data** element extended GUID, as specified 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 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 intermediate node object, as specified in section [2.2.3](#).

Object EXGUID: {"4D97BCEC-28DC-41C5-9274-26CB57966F17"} "0x12000005" specifies the object extended GUID, as specified 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 of 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 (1 byte): "0x00F4" specifies the stream object header for a cell object group data header. Decoded this has a type of "0x0E", length 0x01.

Reserved (1 byte): Set to "0x00".

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", length 47.

E - Object Extended GUID Array Count (1 byte): "0x00" specifies the number of Extended GUIDs, as specified 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 count of bytes of **Object Data** as a compact unsigned 64-bit **integer** with a decoded value of "0x2C".

Object Data (44 bytes): Content {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 "Hello".

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

Data Element End (1 byte): "0x05" specifies the stream object header for 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

```

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
Data Element Start																Data Element EXGUID															
...																															
SN																															
...																															
Data Element Type				Object Group Declarations Start																Object Declaration											
...				Object EXGUID																											
...																															
Object Partition ID				Object Data Size												A								Cell References Count							
B				C												Reserved								D							
...				E												F								G							
Object Data																															
...																															
Object Group Data End				Data Element End																											

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"} "0x06" specifies the **Data** element extended GUID, as specified 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 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 intermediate node object, as specified in section [2.2.3](#).

Object EXGUID: {"4D97BCEC-28DC-41C5-9274-26CB57966F17"} "0x12000006" specifies the object extended GUID, as specified 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 of 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 (1 byte): "0x00F4" specifies the stream object header for a cell object group data header. Decoded, this has a type of "0x0E", length 0x01.

Reserved (1 byte): Set to "0x00".

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", length 47.

E - Object Extended GUID Array Count (1 byte): "0x00" specifies the number of Extended GUIDs, as specified 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 count of bytes of **Object Data** as a compact unsigned 64-bit **integer** with a decoded value of "0x2C".

Object Data (44 bytes): Content {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 "World".

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

Data Element End (1 byte): "0x05" specifies the stream object header for 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

```

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
Data Element Start																Data Element EXGUID															
...																															
SN																															
...																															
Data Element Type								Object Group Declarations Start																Object Declaration							
...								Object EXGUID																							
...																															
Object Partition ID								Object Data Size								A								Cell References Count							
B								C								Reserved								D							
...								E								F								G							
Object Data																															
...																															
Object Group Data End								Data Element End																							

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"} "0x07" specifies the Data element extended GUID, as specified 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 object group declaration start. Decoded this has a type of "0x1D", length 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", length 26. This specifies the start of the data node object.

Object EXGUID: {"4D97BCEC-28DC-41C5-9274-26CB57966F17"} "0x12000007" specifies the object extended GUID, as specified 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 (1 byte): "0x00F4" specifies the stream object header for a cell object group data header. Decoded, this has a type of "0x0E", length 0x01.

Reserved (1 byte): Set to "0x00".

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", length 136.

E - Object Extended GUID Array Count (1 byte): "0x00" specifies the number of Extended GUIDs, as specified 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 count of bytes of **Object Data** as a compact unsigned 64-bit **integer** with a decoded value of "0x84".

Object Data (132 bytes): Content {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 object group data end.

Data Element End (1 byte): "0x05" specifies the stream object header for 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 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

```

0	1	2	3	4	5	6	7	8	9	0 ¹	1	2	3	4	5	6	7	8	9	0 ²	1	2	3	4	5	6	7	8	9	0 ³	1
Data Element Start																Data Element EXGUID															
...																															
SN																															
...																															
Data Element Type								Cell Storage Manifest Schema GUID Start																GUID							
...																															
Cell Storage Manifest Root Declare Start																Root EXGUID															
...																															
Cell ID																															
...																															
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", length 43, compound.

Data Element EXGUID: {"666593A0-174D-4F12-B045-831C6A44BE35"} "0x01" specifies a string representation of the **Data** element extended GUID, as specified 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.

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", Length 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", length 51.

Root EXGUID: {"84DEFAB9-AAA3-4A0D-A3A8-520C77AC7073"} "0x02" specifies a string representation of the root storage manifest extended GUID, as specified 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 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
```

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
Data Element Start																Data Element EXGUID															
...																															
SN																															
...																															
Data Element Type								Cell Manifest Current Revision Start																A							
...																															
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", Length 43, compound.

Data Element EXGUID: {"BB61162F-5532-4BD4-988B-C687B9A9858D"} "0x09" specifies a string representation of the Data element Extended GUID, as specified 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.

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", length 17.

A - Cell Manifest Current Revision EXGUID: {"4D0DC389-5E66-4D6E-88C4-5271D5B48028"} "0x01" specifies a string representation of the current revision extended GUID, as specified 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 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

```

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
Data Element Start																Data Element EXGUID															
...																															
SN																															
...																															
Data Element Type								Revision Manifest Start																Revision ID							
...																															

Base Revision ID	
...	
Revision Manifest Root Declare	Root Extended GUID
...	
Object Extended GUID	
...	
A	Object Group 1 EXGUID
...	
B	Object Group 2 EXGUID
...	
C	Object Group 3 EXGUID
...	
D	Object Group 4 EXGUID
...	
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", length 43, compound.

Data Element EXGUID: {"BEFD0439-4B69-4AB0-8DF9-A4B5EA91D5B9"} "0x01" specifies the Data element extended GUID, as specified 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.

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 specified 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 specified in [\[MS-FSSHTTPB\]](#) section 2.2.1.7, decoded from 00.

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

Root Extended GUID: {"84DEFAB9-AAA3-4A0D-A3A8-520C77AC7073"} "0x02" specifies the root extended GUID in the form of an extended GUID, as specified 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 specified 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 Revision Manifest Object Group Reference Start. Decoded, this has a type of "0x19", length 17.

Object Group 1 EXGUID: {"BB61162F-5532-4BD4-988B-C687B9A9858D"} "0x01" specifies the object group extended GUID, as specified 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 Revision Manifest Object Group Reference Start. Decoded, this has a type of "0x19", length 17.

Object Group 2 EXGUID: {"BB61162F-5532-4BD4-988B-C687B9A9858D"} "0x02" specifies the object group extended GUID, as specified 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 Revision Manifest Object Group Reference Start. Decoded, this has a type of "0x19", length 17.

Object Group 3 EXGUID: {"BB61162F-5532-4BD4-988B-C687B9A9858D"} "0x03" specifies the object group 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.

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

Object Group 4 EXGUID: {"BB61162F-5532-4BD4-988B-C687B9A9858D"} "0x04" specifies the object group extended GUID, as specified in [\[MS-FSSHTTPB\]](#) section 2.2.1.7, decoded from 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 Revision Manifest Object Group Reference Start. Decoded, this has a type of "0x19", length 17.

Object Group 5 EXGUID: {"BB61162F-5532-4BD4-988B-C687B9A9858D"} "0x05" specifies the object group extended GUID, as specified 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 Revision Manifest Object Group Reference Start. Decoded, this has a type of "0x19", length 17.

Object Group 6 EXGUID: {"BB61162F-5532-4BD4-988B-C687B9A9858D"} "0x06" specifies the object group extended GUID, as specified 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 Revision Manifest Object Group Reference Start. Decoded, this has a type of "0x19", length 17.

Object Group 7 EXGUID: {"BB61162F-5532-4BD4-988B-C687B9A9858D"} "0x07" specifies the object group extended GUID, as specified 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 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 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
```

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Data Element Header																Data Element EXGUID															

...		
Serial Number		
...		
Data Element Type	Storage Index Manifest Mapping Start	A
...		
Manifest Mapping SN		
...		
Storage Index Cell Mapping Start	Cell Id	
...		
Cell Mapping EXGUID		
...		
Cell Mapping SN		
Cell Storage Index Revision Mapping Start	Revision EXGUID	
...		
Revision Mapping EXGUID		
...		
Revision Mapping SN		
...		
B		

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

Data Element EXGUID: {"1EBFDDF8-64FA-4EE7-A5DB-61447E8A8CC1"} "0x01" specifies the **Data** element extended GUID, as specified 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.

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 storage index cell mapping. Decoded, this represents a type of "0x11", length 42.

A - Manifest Mapping EGUID: {"666593A0-174D-4F12-B045-831C6A44BE35"} "0x01" specifies the manifest mapping extended GUID, as specified 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.

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

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

Revision EXGUID: {"4D0DC389-5E66-4D6E-88C4-5271D5B48028"} "0x01" specifies the revision extended GUID, as specified 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 specified 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.

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

00000720:

55 03 01

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
A										Cell Request End																					

A - Data Element Package End (1 byte): "0x55" specifies the stream object header for data element package end. This stream object was started in section [3.1.1](#).

Cell Request End (2 bytes): "0x0103" specifies the stream object header for cell request end. This stream object was started in section [3.1.1](#).

4 Security Considerations

This protocol does not introduce any additional security considerations beyond those that apply to its containing protocol [\[MS-FSSHTTPB\]](#).

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 released service packs:

- Microsoft® Office 2010 suites
- Microsoft® SharePoint® Server 2010
- Microsoft® SharePoint® Workspace 2010

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product 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.

6 Change Tracking

This section identifies changes that were made to the [MS-FSSHTTPD] protocol document between the March 2011 and June 2011 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

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 or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- The removal of a document from the documentation set.
- Changes made for template compliance.

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 **Editorial** means that the language and formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated**.

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
5 Appendix A: Product Behavior	Updated the list of applicable product versions.	N	Content updated.

7 Index

A

[Applicability](#) 5

B

[Byte ordering](#) 5

C

[Change tracking](#) 47

Common data types and fields

[cell properties](#) 9

[file chunking](#) 10

[RDC analysis](#) 13

[generating chunks](#) 13

[generating signatures](#) 13

[simple chunking method](#) 13

[zip files](#) 10

[object definitions](#) 6

[common node object properties](#) 6

[data node object](#) 9

[cell references](#) 9

[data](#) 9

[references](#) 9

[intermediate node object](#) 8

[cell references](#) 9

[data](#) 8

[references](#) 9

[root node object](#) 7

[cell references](#) 8

[data](#) 7

[references](#) 8

[transport](#) 6

D

[Data node object](#) 9

[cell references](#) 9

[data](#) 9

[references](#) 9

Data types and fields – common

[transport](#) 6

Details

common data types and fields

[cell properties](#) 9

[file chunking](#) 10

[RDC analysis](#) 13

[generating chunks](#) 13

[generating signatures](#) 13

[simple chunking method](#) 13

[zip files](#) 10

[object definitions](#) 6

[common node object properties](#) 6

[data node object](#) 9

[cell references](#) 9

[data](#) 9

[references](#) 9

[intermediate node object](#) 8

[cell references](#) 9

[data](#) 8

[references](#) 9

[root node object](#) 7

[cell references](#) 8

[data](#) 7

[references](#) 8

[transport](#) 6

E

[Example Put example](#) 15

[object groups](#) 19

[request header](#) 17

Examples

[Example Put](#) 15

[object groups](#) 19

[request header](#) 17

F

[Fields - vendor-extensible](#) 5

[File chunking](#) 10

[RDC analysis](#) 13

[generating chunks](#) 13

[generating signatures](#) 13

[simple chunking method](#) 13

[zip files](#) 10

G

[Glossary](#) 4

I

[Implementer - security considerations](#) 45

[Informative references](#) 4

[Intermediate node object](#) 8

[cell references](#) 9

[data](#) 8

[references](#) 9

L

[Localization](#) 5

N

[Normative references](#) 4

P

[Product behavior](#) 46

R

[RDC analysis](#) 13

- [generating chunks](#) 13
- [generating signatures](#) 13
- References
 - [informative](#) 4
 - [normative](#) 4
- [Relationship to protocols and other structures](#) 5
- [Root node object](#) 7
 - [cell references](#) 8
 - [data](#) 7
 - [references](#) 8

S

- [Schema overview](#) 4
- [Security - implementer considerations](#) 45
- [Simple chunking method](#) 13
- Structures
 - [cell properties](#) 9
 - [file chunking](#) 10
 - [RDC analysis](#) 13
 - [generating chunks](#) 13
 - [generating signatures](#) 13
 - [simple chunking method](#) 13
 - [zip files](#) 10
 - [object definitions](#) 6
 - [common node object properties](#) 6
 - [data node object](#) 9
 - [cell references](#) 9
 - [data](#) 9
 - [references](#) 9
 - [intermediate node object](#) 8
 - [cell references](#) 9
 - [data](#) 8
 - [references](#) 9
 - [root node object](#) 7
 - [cell references](#) 8
 - [data](#) 7
 - [references](#) 8
 - [transport](#) 6
- Synopsis
 - [byte ordering](#) 5
 - [schema overview](#) 4

T

- [Tracking changes](#) 47

V

- [Vendor-extensible fields](#) 5
- [Versioning](#) 5

Z

- [Zip files](#) 10