

# [MS-ECS]: Enterprise Client Synchronization Protocol

---

## 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 [Open Specification Promise](#) or the [Community Promise](#). 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 [iplg@microsoft.com](mailto:iplg@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. For a list of Microsoft trademarks, visit [www.microsoft.com/trademarks](http://www.microsoft.com/trademarks).
- **Fictitious Names.** The example companies, organizations, products, domain names, email 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
08/08/2013	1.0	New	Released new document.
11/14/2013	2.0	Major	Significantly changed the technical content.
02/13/2014	2.0	No change	No change to the meaning, language, or formatting of the technical content.
05/15/2014	3.0	Major	Significantly changed the technical content.

# Contents

<b>1 Introduction</b>	<b>7</b>
1.1 Glossary	7
1.2 References	7
1.2.1 Normative References	7
1.2.2 Informative References	8
1.3 Overview	8
1.4 Relationship to Other Protocols	8
1.5 Prerequisites/Preconditions	9
1.6 Applicability Statement	9
1.7 Versioning and Capability Negotiation	9
1.8 Vendor-Extensible Fields	9
1.9 Standards Assignments	9
<b>2 Messages</b>	<b>10</b>
2.1 Transport	10
2.2 Common Data Types	10
2.2.1 HTTP Headers	10
2.2.1.1 x-ecs-request-error	11
2.2.1.2 x-ecs-devicename	11
2.2.1.3 x-ecs-share-type	12
2.2.1.4 x-ecs-changes-URL	12
2.2.1.5 x-ecs-partnershipID	12
2.2.1.6 x-ecs-admin-contact	12
2.2.1.7 x-ecs-session-location-url	12
2.2.1.8 x-ecs-session-id	12
2.2.1.9 x-ecs-continue	13
2.2.1.10 x-ecs-metadata-version	13
2.2.1.11 x-ecs-domain	13
2.2.2 Common Data Structures	13
2.2.2.1 SYNC_BLOB	15
2.2.2.2 QUOTA_USAGE_ENTRY	15
2.2.2.3 POLICY_ENTRY	16
2.2.2.4 BATCH_LIMITS_ENTRY	16
2.2.2.5 FILE_METADATA_ENTRY	17
2.2.2.6 FILE_INFO_INPUT_ENTRY	19
2.2.2.7 FILE_INFO_ENTRY	20
2.2.2.8 FILE_STATUS_ENTRY	21
2.2.2.9 UPLOAD_ENTRY	21
2.2.2.10 UPLOAD_RESPONSE_ENTRY	23
2.2.2.11 DOWNLOAD_ENTRY	23
2.2.2.12 DOWNLOAD_RESPONSE_ENTRY	24
2.2.2.13 FILE_DOWNLOAD_INFO_ENTRY	25
2.2.2.14 SYNC_CHANGE_BATCH	26
2.2.2.15 SYNC_MD5HASH	26
2.2.2.16 VECTOR_POLICY_ENTRY	27
2.2.2.17 VECTOR_FILE_METADATA_ENTRY	27
2.2.2.18 VECTOR_FILE_INFO_INPUT_ENTRY	27
2.2.2.19 VECTOR_FILE_INFO_ENTRY	28
2.2.2.20 VECTOR_FILE_STATUS_ENTRY	28
2.2.2.21 VECTOR_UPLOAD_ENTRY	28

2.2.2.22	VECTOR_UPLOAD_RESPONSE_ENTRY.....	29
2.2.2.23	VECTOR_DOWNLOAD_ENTRY.....	29
2.2.2.24	VECTOR_DOWNLOAD_RESPONSE_ENTRY.....	30
2.2.2.25	VECTOR_FILE_DOWNLOAD_INFO_ENTRY.....	30
2.2.2.26	VECTOR_STRING.....	30
2.2.2.27	ECS_STRING.....	31
<b>3</b>	<b>Protocol Details.....</b>	<b>32</b>
3.1	ServiceDiscovery Server Details.....	32
3.1.1	Abstract Data Model.....	32
3.1.2	Timers.....	32
3.1.3	Initialization.....	32
3.1.4	Higher-Layer Triggered Events.....	33
3.1.5	Message Processing Events and Sequencing Rules.....	33
3.1.5.1	Server Discovery.....	33
3.1.5.1.1	GET.....	33
3.1.5.1.1.1	Request Body.....	34
3.1.5.1.1.2	Response Body.....	34
3.1.5.1.1.3	Processing Details.....	34
3.1.5.2	Share Discovery.....	35
3.1.5.2.1	GET.....	35
3.1.5.2.1.1	Request Body.....	36
3.1.5.2.1.2	Response Body.....	36
3.1.5.2.1.3	Processing Details.....	36
3.1.5.3	Server Capabilities.....	36
3.1.5.3.1	GET.....	36
3.1.5.3.1.1	Request Body.....	37
3.1.5.3.1.2	Response Body.....	37
3.1.5.3.1.3	Processing Details.....	38
3.1.6	Timer Events.....	38
3.1.7	Other Local Events.....	38
3.2	DetectServerChanges Server Details.....	38
3.2.1	Abstract Data Model.....	38
3.2.2	Timers.....	38
3.2.3	Initialization.....	38
3.2.4	Higher-Layer Triggered Events.....	38
3.2.5	Message Processing Events and Sequencing Rules.....	38
3.2.5.1	Detect Server Changes.....	38
3.2.5.1.1	HEAD.....	39
3.2.5.1.1.1	Request Body.....	39
3.2.5.1.1.2	Response Body.....	39
3.2.5.1.1.3	Processing Details.....	39
3.2.6	Timer Events.....	40
3.2.7	Other Local Events.....	40
3.3	UserConfiguration Server Details.....	40
3.3.1	Abstract Data Model.....	40
3.3.2	Timers.....	40
3.3.3	Initialization.....	40
3.3.4	Higher-Layer Triggered Events.....	40
3.3.5	Message Processing Events and Sequencing Rules.....	40
3.3.5.1	User Configuration.....	40
3.3.5.1.1	GET.....	40
3.3.5.1.1.1	Request Body.....	41

3.3.5.1.1.2	Response Body.....	41
3.3.5.1.1.3	Processing Details.....	41
3.3.6	Timer Events .....	42
3.3.7	Other Local Events .....	42
3.4	PeerSynchronizationSession Server Details .....	42
3.4.1	Abstract Data Model .....	42
3.4.1.1	Global .....	42
3.4.1.2	Per Session.....	42
3.4.1.2.1	Per ChangeBatch.....	42
3.4.2	Timers .....	42
3.4.3	Initialization .....	43
3.4.4	Higher-Layer Triggered Events.....	43
3.4.5	Message Processing Events and Sequencing Rules.....	43
3.4.5.1	Create Session .....	44
3.4.5.1.1	PUT .....	44
3.4.5.1.1.1	Request Body.....	45
3.4.5.1.1.2	Response Body.....	45
3.4.5.1.1.3	Processing Details.....	46
3.4.5.1.2	Sync Batch Parameters .....	47
3.4.5.2.1	GET .....	47
3.4.5.2.1.1	Request Body.....	48
3.4.5.2.1.2	Response Body.....	48
3.4.5.2.1.3	Processing Details.....	48
3.4.5.2.2	PUT .....	48
3.4.5.2.2.1	Request Body.....	49
3.4.5.2.2.2	Response Body.....	49
3.4.5.2.2.3	Processing Details.....	50
3.4.5.3	Prepare Batch .....	50
3.4.5.3.1	PUT .....	50
3.4.5.3.1.1	Request Body.....	51
3.4.5.3.1.2	Response Body.....	51
3.4.5.3.1.3	Processing Details.....	51
3.4.5.4	Upload Batch .....	51
3.4.5.4.1	PUT .....	51
3.4.5.4.1.1	Request Body.....	52
3.4.5.4.1.2	Response Body.....	52
3.4.5.4.1.3	Processing Details.....	53
3.4.5.5	Delete Session .....	53
3.4.5.5.1	DELETE.....	53
3.4.5.5.1.1	Request Body.....	54
3.4.5.5.1.2	Response Body.....	54
3.4.5.5.1.3	Processing Details.....	54
3.4.5.6	Download Batch .....	54
3.4.5.6.1	GET .....	54
3.4.5.6.1.1	Request Body.....	55
3.4.5.6.1.2	Response Body.....	55
3.4.5.6.1.3	Processing Details.....	55
3.4.6	Timer Events .....	56
3.4.7	Other Local Events .....	56
3.5	ServerAPI Server Details .....	56
3.5.1	Abstract Data Model .....	56
3.5.2	Timers .....	56
3.5.3	Initialization .....	56

3.5.4	Higher-Layer Triggered Events .....	56
3.5.5	Message Processing Events and Sequencing Rules .....	57
3.5.5.1	Upload Data .....	57
3.5.5.1.1	PUT .....	57
3.5.5.1.1.1	Request Body .....	58
3.5.5.1.1.2	Response Body .....	58
3.5.5.1.1.3	Processing Details .....	58
3.5.5.2	Download Data .....	58
3.5.5.2.1	PUT .....	58
3.5.5.2.1.1	Request Body .....	59
3.5.5.2.1.2	Response Body .....	59
3.5.5.2.1.3	Processing Details .....	59
3.5.6	Timer Events .....	60
3.5.7	Other Local Events .....	60
3.6	ECS Client Details .....	60
3.6.1	Abstract Data Model .....	60
3.6.1.1	Global .....	60
3.6.1.2	Per UploadFile .....	61
3.6.1.3	Per Share .....	61
3.6.1.4	Per DownloadFile .....	61
3.6.2	Timers .....	61
3.6.3	Initialization .....	61
3.6.4	Higher-Layer Triggering Events .....	63
3.6.4.1	Application Requests Uploading Data To Sync Target .....	63
3.6.5	Message Processing Events and Sequencing Rules .....	63
3.6.5.1	Upload Scenario .....	63
3.6.5.2	Download Scenario .....	65
3.6.6	Timer Events .....	66
3.6.7	Other Local Events .....	66
<b>4</b>	<b>Protocol Examples .....</b>	<b>67</b>
4.1	Query User Configuration Information and Detect Server Changes .....	67
4.2	Upload Scenario .....	71
4.3	Download Scenario .....	76
<b>5</b>	<b>Security .....</b>	<b>79</b>
5.1	Security Considerations for Implementers .....	79
5.2	Index of Security Parameters .....	79
<b>6</b>	<b>Appendix A: Product Behavior .....</b>	<b>80</b>
<b>7</b>	<b>Change Tracking .....</b>	<b>81</b>
<b>8</b>	<b>Index .....</b>	<b>95</b>

# 1 Introduction

This document specifies the Enterprise Client Synchronization (ECS) Protocol.

The Enterprise Client Synchronization Protocol is designed for synchronizing files across multiple devices in an enterprise network.

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in RFC 2119. Sections 1.5 and 1.9 are also normative but cannot contain those terms. All other sections and examples in this specification are informative.

## 1.1 Glossary

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

**Active Directory**  
**fully qualified domain name (FQDN)**  
**MD5 hash**  
**URI**  
**URL**

The following terms are specific to this document:

**Sync Framework:** A data synchronization platform that can be used to synchronize data across multiple data stores.

**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

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the documents, which are updated frequently. References to other documents include a publishing year when one is available.

### 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](mailto:dochelp@microsoft.com). We will assist you in finding the relevant information.

[MS-DTYP] Microsoft Corporation, "[Windows Data Types](#)".

[MS-FSVCA] Microsoft Corporation, "[File Set Version Comparison Algorithms](#)".

[RFC1321] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992, <http://www.ietf.org/rfc/rfc1321.txt>

[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>

[RFC2616] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, <http://www.ietf.org/rfc/rfc2616.txt>

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000, <http://www.ietf.org/rfc/rfc2818.txt>

### 1.2.2 Informative References

[MSDN-FSA] Microsoft Corporation, "File Attribute Constants", [http://msdn.microsoft.com/en-us/library/windows/desktop/gg258117\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/gg258117(v=vs.85).aspx)

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

### 1.3 Overview

The Enterprise Client Synchronization Protocol is used to access REST-based file sync services over web-based transport.

The protocol is used for uploading and downloading file data between a client and server participating in the synchronization of a namespace. Both the upload and download scenarios are driven by the client role of this protocol.

Before creating any sessions for the data transfer, the client initially queries the server for information on the sync target share and the server's capabilities.

In the upload scenario, the client is notified of local changes to the file data from an underlying **Sync Framework** and takes the following steps:

1. Create the server session.
2. Get the server's sync knowledge.
3. Prepare the server for upload.
4. Perform data transfer (upload).
5. Commit change.
6. Delete session.

In the download scenario, the protocol provides a mechanism for the client to receive notifications about changes on the server that will need to be synchronized. When the client receives a notification for server-side changes, the client takes the following steps:

1. Create the server session.
2. Update the server on the client's sync knowledge.
3. Get the change batch from the server.
4. Perform data transfer (download).
5. Delete session.

### 1.4 Relationship to Other Protocols

None.



## 1.5 Prerequisites/Preconditions

The Enterprise Client Synchronization Protocol does not provide a mechanism for a client to discover the existence and location of a global sync **URI** for syncing a namespace. It is a prerequisite that the client's local configuration include a global sync URI that can be used to enumerate all servers exposing that namespace for synchronization.

The Enterprise Client Synchronization Protocol does not define an authentication or authorization scheme. Implementers of this protocol should review the recommended security prerequisites in Security Considerations for Implementers (section [5.1](#)) of this document.

## 1.6 Applicability Statement

This protocol is applicable where file data is required to be synchronized across multiple devices in an enterprise network.

## 1.7 Versioning and Capability Negotiation

This protocol currently supports one version (1.0). The server returns the supported versions in the response for Server Capabilities as specified in section [3.1.5.3](#). The protocol does not provide any mechanism for capability negotiation based on versions. [<1>](#)

Version	Value
ECS Protocol version 1	"1.0"

## 1.8 Vendor-Extensible Fields

None.

## 1.9 Standards Assignments

None.

## 2 Messages

### 2.1 Transport

The Enterprise Client Synchronization Protocol uses HTTP or secure HTTP 1.1 as transport, as specified in [\[RFC2616\]](#) and [\[RFC2818\]](#).

### 2.2 Common Data Types

The following table summarizes the server-side resources used by the Enterprise Client Synchronization Protocol.

Resource type	Description
Server Discovery	The resource used to enumerate all servers exposing the namespace for sync.
Share Discovery	The resource used to discover the sync share on the server.
Server Capabilities	The resource used to query the capabilities of the sync service.
Detect Server Changes	The resource used to do polling for the server-side changes.
User Configuration	The resource used to query the user configuration information for a sync target share.
Create Session	The resource used to create a new session on the server.
Sync Batch Parameters	The resource used to retrieve or update synchronization batch parameters.
Prepare Batch	The resource used to send information to the server to prepare for the change batch.
Upload Batch	The resource used to commit a change batch on the server.
Delete Session	The resource used to delete a sync session on the server.
Download Batch	The resource used to receive a change batch from the server to the client.
Upload Data	The resource used to send file data to the server.
Download Data	The resource used to receive file data from the server.

#### 2.2.1 HTTP Headers

The following table summarizes the set of HTTP headers defined by this specification.

Header	Description
x-ecs-admin-contact	A string representing the email contact of the server administrator.
x-ecs-changes-URL	The <b>URL</b> to be used by the client to poll for changes on the server.
x-ecs-continue	The opaque continuation token received from the server in a download batch operation.

Header	Description
x-ecs-devicename	A string representing the name and operating system of the device issuing the request.
x-ecs-domain	The <b>fully qualified domain name (FQDN)</b> of the domain to which the client is joined.
x-ecs-metadata-version	A string containing the version of the server metadata for the current sync replica.
x-ecs-partnershipID	The PartnershipID string returned by the server in Share Discovery, as specified in section <a href="#">3.1.5.2.1.2</a>
x-ecs-request-error	A string describing request failure information.
x-ecs-session-id	The identifier of sync session.
x-ecs-session-location-url	The URL to access a session.
x-ecs-share-type	A string representing the type of the share to discover. Allowed values for this string are "User Data".

### 2.2.1.1 x-ecs-request-error

This header describes request failure information. The value for this header MUST be a string representation of the HRESULT in hexadecimal format.

Example:

x-ecs-request-error: "0x8007005"

```
hexdigit = "0" / "1" / "2" / "3" / "4" / "5" / "6" / "7" / "8" / "9" / "a" / "b" / "c" / "d" / "e" / "f" / "A" / "B" / "C" / "D" / "E" / "F" / "x"
x-ecs-request-error = 8 hexdigit
```

### 2.2.1.2 x-ecs-devicename

A string representing the name and operating system of the device issuing the request in the following format:

{Device Name, OS Family, OS Major Version, OS Minor Version, UserAgent Name}

Example:

x-ecs-devicename: {HOMEPC,Windows,6,3,MS\_WorkFoldersClient}

```
String = *(%x20-7E)
DeviceName = String
OSFamily = String
MajorVersion = String
MinorVersion = String
UserAgentName = String
x-ecs-devicename = "{" DeviceName "," OSFamily "," MajorVersion "," MinorVersion ","
UserAgentName "}"
```

### 2.2.1.3 x-ecs-share-type

A string representing the type of the share to discover.

```
String = *(%x20-7E)
x-ecs-share-type = String
```

### 2.2.1.4 x-ecs-changes-URL

This header provides the URL to be used by the client to poll for changes on the server.

```
String = *(%x20-7E)
x-ecs-changes-URL = String
```

### 2.2.1.5 x-ecs-partnershipID

A string representing the PartnershipID for a sync target. This value MUST be Base64-encoded.

```
String = *(%x20-7E)
x-ecs-partnershipID = String
```

### 2.2.1.6 x-ecs-admin-contact

A string representing the email contact of the server admin.

```
String = *(%x20-7E)
x-ecs-admin-contact = String
```

### 2.2.1.7 x-ecs-session-location-url

The URL prefix to access a session.

```
String = *(%x20-7E)
x-ecs-session-location-url = String
```

### 2.2.1.8 x-ecs-session-id

An implementation-specific string that identifies the session.

```
String = *(%x20-7E)
x-ecs-session-id = String
```

### 2.2.1.9 x-ecs-continue

The opaque continuation token received from the server in a download batch operation.

```
String = *(%x20-7E)
x-ecs-continue = String
```

### 2.2.1.10 x-ecs-metadata-version

A string containing the version of the server metadata for the current sync replica.

```
String = *(%x20-7E)
x-ecs-metadata-version = String
```

### 2.2.1.11 x-ecs-domain

When present, this header indicates that the client is part of an **Active Directory** domain and the header value contains the FQDN of the domain. If omitted, or if the value is an empty string, the client is considered to not be in a domain.

```
String = *(%x20-7E)
x-ecs-domain = String
```

## 2.2.2 Common Data Structures

The following table summarizes the set of common data structures defined by this specification. Common data types are specified in [\[MS-DTYP\]](#).

Unless otherwise specified, multiple-byte fields (16-bit, 32-bit, and 64-bit fields) MUST be transmitted in little-endian order (with the least-significant byte first). Unless otherwise indicated, numeric fields are integers of the specified byte length.

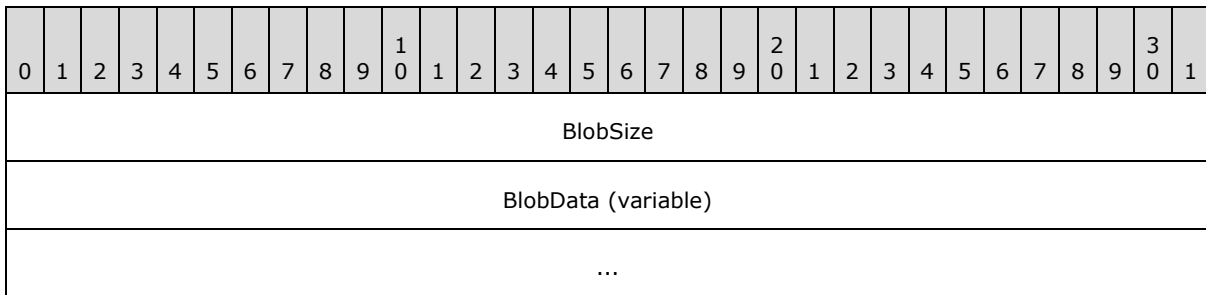
Structure Name	Section	Description
SYNC_GID	<a href="#">[MS-FSVCA]</a> section 2.1	The structure used to represent an identifier for a file that is part of a sync batch.
CLOCK_VECTOR_ELEMENT	<a href="#">[MS-FSVCA]</a> section 2.9	The structure used to represent a version for the file in the sync batch.
SYNC_BLOB	<a href="#">2.2.2.1</a>	The structure used to specify a binary stream of data.
QUOTA_USAGE_ENTRY	<a href="#">2.2.2.2</a>	The structure used to specify the current data usage of a user on the sync target share.

Structure Name	Section	Description
POLICY_ENTRY	<a href="#">2.2.2.3</a>	The structure used to specify the policies on how the client is expected to set up its target directory.
BATCH_LIMITS_ENTRY	<a href="#">2.2.2.4</a>	The structure used to specify the parameters that describe sync batch characteristics.
FILE_METADATA_ENTRY	<a href="#">2.2.2.5</a>	The structure used to specify the metadata information for a file in a sync batch.
FILE_INFO_INPUT_ENTRY	<a href="#">2.2.2.6</a>	The structure used to specify the sync preparation information of a file that the client sends to a server before an upload.
FILE_INFO_ENTRY	<a href="#">2.2.2.7</a>	The structure used to specify the sync preparation information of a file that the server sends to the client before an upload.
FILE_STATUS_ENTRY	<a href="#">2.2.2.8</a>	The structure used to specify the status of a file commit in a sync process.
UPLOAD_ENTRY	<a href="#">2.2.2.9</a>	The structure used to specify information of the data for a file that is being uploaded to the server.
UPLOAD_RESPONSE_ENTRY	<a href="#">2.2.2.10</a>	The structure used to specify the server response data for a file being uploaded.
DOWNLOAD_ENTRY	<a href="#">2.2.2.11</a>	The structure used to specify the information of the data for a file that is being downloaded from the server.
DOWNLOAD_RESPONSE_ENTRY	<a href="#">2.2.2.12</a>	The structure used to specify the server response data for a file being downloaded.
FILE_DOWNLOAD_INFO_ENTRY	<a href="#">2.2.2.13</a>	The structure that specifies the information on how the file content needs to be downloaded by the client.
SYNC_CHANGE_BATCH	<a href="#">2.2.2.14</a>	The structure that defines the metadata describing the changes to be synchronized.
SYNC_MD5HASH	<a href="#">2.2.2.15</a>	The structure that defines the serialization format used by this protocol for <b>MD5 hash</b> data.
VECTOR_POLICY_ENTRY	<a href="#">2.2.2.16</a>	The structure representing a collection of POLICY_ENTRY structures.
VECTOR_FILE_METADATA_ENTRY	<a href="#">2.2.2.17</a>	The structure representing a collection of FILE_METADATA_ENTRY structures.
VECTOR_FILE_INFO_INPUT_ENTRY	<a href="#">2.2.2.18</a>	The structure representing a collection of FILE_INFO_INPUT_ENTRY structures.
VECTOR_FILE_INFO_ENTRY	<a href="#">2.2.2.19</a>	The structure representing a collection of FILE_INFO_ENTRY structures.

Structure Name	Section	Description
VECTOR_FILE_STATUS_ENTRY	<a href="#">2.2.2.20</a>	The structure representing a collection of FILE_STATUS_ENTRY structures.
VECTOR_UPLOAD_ENTRY	<a href="#">2.2.2.21</a>	The structure representing a collection of UPLOAD_ENTRY structures.
VECTOR_UPLOAD_RESPONSE_ENTRY	<a href="#">2.2.2.22</a>	The structure representing a collection of UPLOAD_RESPONSE_ENTRY structures.
VECTOR_DOWNLOAD_ENTRY	<a href="#">2.2.2.23</a>	The structure representing a collection of DOWNLOAD_ENTRY structures.
VECTOR_DOWNLOAD_RESPONSE_ENTRY	<a href="#">2.2.2.24</a>	The structure representing a collection of DOWNLOAD_RESPONSE_ENTRY structures.
VECTOR_FILE_DOWNLOAD_INFO_ENTRY	<a href="#">2.2.2.25</a>	The structure representing a collection of FILE_DOWNLOAD_INFO_ENTRY structures.
VECTOR_STRING	<a href="#">2.2.2.26</a>	The structure representing a collection of ECS_STRING structures.
ECS_STRING	<a href="#">2.2.2.27</a>	The structure representing a UTF-8 string of a prescribed length.

### 2.2.2.1 SYNC\_BLOB

The SYNC\_BLOB structure represents a binary stream of data.

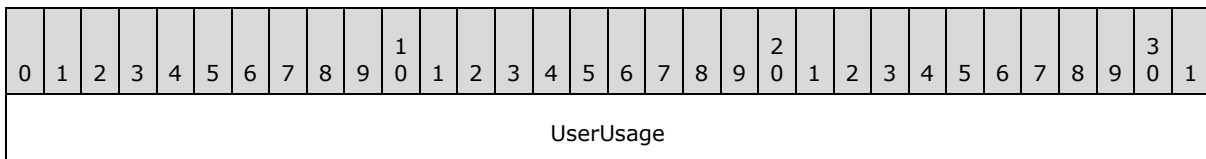


**BlobSize (4 bytes):** A 32-bit unsigned integer that contains the size of the blob data in bytes.

**BlobData (variable):** A variable stream of bytes containing the blob data.

### 2.2.2.2 QUOTA\_USAGE\_ENTRY

The QUOTA\_USAGE\_ENTRY structure specifies the current data usage of user on the sync target share.



...
UserDataFreeSpace
...

**UserUsage (8 bytes):** A 64-bit unsigned integer that contains the amount of data, in bytes, in the user's share.

**UserDataFreeSpace (8 bytes):** A 64-bit unsigned integer that contains the amount of available free space, in bytes, in the user's share.

### 2.2.2.3 POLICY\_ENTRY

The POLICY\_ENTRY structure specifies the policies on how the client is expected to set up its target directory.

<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
PolicyName										PolicyType					

**PolicyName (1 byte):** An unsigned 8-bit value that identifies the policy. This MUST be one of the following values.

Value	Description
0x01	Encryption
0x02	Password
0x03	AutoLock

**PolicyType (1 byte):** A nonzero value indicates that this policy is enforced.

### 2.2.2.4 BATCH\_LIMITS\_ENTRY

The BATCH\_LIMITS\_ENTRY structure specifies the parameters that describe sync batch characteristics.

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
MaxFileDataSize																															
MaxFileCount																															

**MaxFileDataSize (4 bytes):** A 32-bit unsigned integer that contains the maximum size, as a multiple of 1048576 bytes, of file data in one batch.

**MaxFileCount (4 bytes):** A 32-bit unsigned integer that contains the maximum count of files in one batch.



### 2.2.2.5 FILE\_METADATA\_ENTRY

The FILE\_METADATA\_ENTRY structure specifies the metadata information for a file in a sync batch.

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
FileId																															
...																															
...																															
...																															
...																															
...																															
SyncVersion																															
...																															
...																															
FileStreamVersion																															
...																															
...																															
...																															
ParentID																															
...																															
...																															
...																															
...																															
...																															
FileAttributes																															

NamespaceChangeTime
...
AttributeChangeTime
...
CreatedTime
...
ModifiedTime
...
ContentSize
...
Name (variable)
...
OriginatingDeviceNameIndex

**FileId (24 bytes):** An identifier that uniquely identifies the file or directory. This MUST be in the format specified in [\[MS-FSVCA\]](#) section 2.1.

**SyncVersion (12 bytes):** **SyncVersion** is the version of the file entry. This MUST be in the format specified in [\[MS-FSVCA\]](#) section 2.9.

**FileStreamVersion (16 bytes):** A GUID representing the version of the file content. This MUST not be used for directories.

**ParentId (24 bytes):** ParentId is an identifier for the metadata of the parent item. The parent item MUST be a directory. Files without a parent MUST set **ParentId.GidPrefix** to 0x0000000000000000 and **ParentId.UniqueId** to 0x00700012000000000000000000000000. This MUST be in the format specified in [\[MS-FSVCA\]](#) section 2.1.

**FileAttributes (4 bytes):** A 32-bit unsigned integer representing the attributes of the file allowed by the underlying object store. <2>

**NamespaceChangeTime (8 bytes):** The time at which the name of the file or its **ParentId** was changed. This field is used for resolving conflicts during synchronization. This MUST be in the format specified in [\[MS-DTYP\]](#) section 2.3.3.

**AttributeChangeTime (8 bytes):** The time at which the attributes of the file were changed. This field is used for resolving conflicts during synchronization. This MUST be in the format specified in [\[MS-DTYP\]](#) section 2.3.3.

**CreatedTime (8 bytes):** The time at which the file was created. This MUST be in the format specified in [\[MS-DTYP\]](#) section 2.3.1.

**ModifiedTime (8 bytes):** The time at which the file was last modified. This MUST be in the format specified in [\[MS-DTYP\]](#) section 2.3.1.

**ContentSize (8 bytes):** A 64-bit unsigned integer representing the size of the file in bytes. This MUST be zero for a directory entry.

**Name (variable):** An ECS\_STRING structure representing the name of the file or directory. The string in this structure MUST not be empty.

**OriginatingDeviceNameIndex (2 bytes):** A 16-bit unsigned integer representing the index of the string representing the device in the **DeviceNames** field of the SYNC\_CHANGE\_BATCH structure, as specified in section [2.2.2.14](#).

### 2.2.2.6 FILE\_INFO\_INPUT\_ENTRY

The FILE\_INFO\_INPUT\_ENTRY structure specifies the sync preparation information of a file that the client sends to the server before an upload.

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
FileExtension (variable)																															
...																															
SyncItemId																															
...																															
...																															
...																															
...																															
...																															
StreamId																															
...																															
...																															
...																															
...																															
FileSize																															

...

**FileExtension (variable):** An ECS\_STRING structure containing the extension of the file.

**SyncItemId (24 bytes):** An identifier that is used to correlate the file metadata with the Sync Framework change item. This MUST be in the format specified in [\[MS-FSVCA\]](#) section 2.1.

**StreamId (16 bytes):** A GUID containing the stream identifier of the file.

**FileSize (8 bytes):** A 64-bit unsigned integer representing the size of the file in bytes.

### 2.2.2.7 FILE\_INFO\_ENTRY

The FILE\_INFO\_ENTRY structure specifies the sync preparation information of a file that the server sends to the client before an upload.

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
SyncItemId																															
...																															
...																															
...																															
...																															
...																															
Uri (Variable)																															
...																															
ProtocolType																PrepareResult															
...																															

**SyncItemId (24 bytes):** A SYNC\_GID structure as described in [\[MS-FSVCA\]](#) section 2.1. An identifier that is used to correlate the file metadata with the Sync Framework change item.

**Uri (variable):** An ECS\_STRING structure containing the URI used to perform data transfer for this file. The string in this structure MUST be empty.

**ProtocolType (1 byte):** Indicates the type of data transfer that the server supports for this file. This value MUST be one of the following.

Value	Meaning
0x00	No upload is required for this file.
0x01	File batching data transfer is required to transfer this file.

**PrepareResult (4 bytes):** This value contains the result of the preparation process.

### 2.2.2.8 FILE\_STATUS\_ENTRY

The FILE\_STATUS\_ENTRY structure specifies the information of a file commit in a sync process.

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
SyncItemId																															
...																															
...																															
...																															
...																															
...																															
Status																															

**SyncItemId (24 bytes):** A SYNC\_GID structure as specified in [\[MS-FSVCA\]](#) section 2.1.

**SyncItemId** is an identifier that is used to correlate the file metadata with the Sync Framework change item.

**Status (4 bytes):** Indicates the result of applying changes to the file. This MUST be in the format specified in [\[MS-DTYP\]](#) section 2.2.18.

### 2.2.2.9 UPLOAD\_ENTRY

The UPLOAD\_ENTRY structure specifies the information of the data for a file that is being uploaded to the server.

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
SyncItemId																															
...																															

...
...
...
...
FileHash
...
...
...
Offset
...
Length
Reserved
...
UploadData (Variable)
...

**SyncItemId (24 bytes):** A SYNC\_GID structure as specified in [\[MS-FSVCA\]](#) section [2.1](#). This value uniquely identifies the file replica within the change batch.

**FileHash (16 bytes):** The MD5 hash, as specified in [\[RFC1321\]](#) of the file contents. The hash is serialized in SYNC\_MD5HASH format as specified in section [2.2.2.15](#).

**Offset (8 bytes):** A 64-bit unsigned integer containing the offset, in bytes, in the file for the data to be uploaded.

**Length (4 bytes):** A 64-bit unsigned integer containing the length, in bytes, of the file data to be uploaded.

**Reserved (8 bytes):** A 64-bit unsigned integer. This field MUST NOT be used and MUST be reserved. The client SHOULD [<3>](#) set this field to 0, and the server MUST ignore it on receipt.

**UploadData (variable):** A SYNC\_BLOB structure as specified in section [2.2.2.1](#). The payload data to be uploaded.

### 2.2.2.10 UPLOAD\_RESPONSE\_ENTRY

The UPLOAD\_RESPONSE\_ENTRY structure specifies the server response data for a file being uploaded.

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
SyncItemId																															
...																															
...																															
...																															
...																															
...																															
Result																															

**SyncItemId (24 bytes):** A SYNC\_GID structure as specified in [\[MS-FSVCA\]](#) section 2.1. A value that uniquely identifies the file replica within the change batch.

**Result (4 bytes):** The result of the upload process for the file. If the upload is successful, this value MUST be set to S\_OK. This MUST be in the format as specified in [\[MS-DTYP\]](#) section 2.2.18.

### 2.2.2.11 DOWNLOAD\_ENTRY

The DOWNLOAD\_ENTRY structure specifies the information of the data for a file that is being downloaded from the server.

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
SyncItemId																															
...																															
...																															
...																															
...																															
...																															

FileVersion (variable)
...

**SyncItemId (24 bytes):** A SYNC\_GID structure as specified in [\[MS-FSVCA\]](#) section [2.1](#). An identifier that is used to correlate the file metadata with the Sync Framework change item.

**FileVersion (variable):** A SYNC\_BLOB structure as specified in section [2.2.2.1](#). An opaque stream of bytes that identifies the file version.

### 2.2.2.12 DOWNLOAD\_RESPONSE\_ENTRY

The DOWNLOAD\_RESPONSE\_ENTRY structure specifies the server response data for a file being downloaded.

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
SyncItemId																															
...																															
...																															
...																															
...																															
...																															
DataLength																															
...																															
Data (variable)																															
...																															
Result																															
FileHash																															
...																															
...																															
...																															



**SyncItemId (24 bytes):** A SYNC\_GID structure as specified in [\[MS-FSVCA\]](#) section 2.1. An identifier that is used to correlate the file metadata with the Sync Framework change item.

**DataLength (8 bytes):** A 64-bit unsigned integer containing the length, in bytes, of the **Data** field.

**Data (variable):** The contents of the file being downloaded from the server.

**Result (4 bytes):** The result of the file download operation. When the result indicates an error, the **DataLength** field MUST be zero and the **Data** field MUST be empty. This MUST be in the format specified in [\[MS-DTYP\]](#) section 2.2.18.

**FileHash (16 bytes):** The MD5 hash, as specified in [\[RFC1321\]](#), of the file contents. The hash is serialized in SYNC\_MD5HASH format as specified in section [2.2.2.15](#).

### 2.2.2.13 FILE\_DOWNLOAD\_INFO\_ENTRY

The FILE\_DOWNLOAD\_INFO\_ENTRY structure specifies information that is required for the client to download the file content.

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
SyncItemId																															
...																															
...																															
...																															
...																															
...																															
Uri (Variable)																															
...																															
ProtocolType																															

**SyncId (24 bytes):** A SYNC\_GID structure as specified in [\[MS-FSVCA\]](#) section 2.1. A value that uniquely identifies the file item within the sync batch.

**Uri (variable):** An ECS\_STRING structure containing the URI used to perform data transfer of the file stream. The string in this structure MUST be empty.

**ProtocolType (1 byte):** Indicates the type of data transfer that the server supports for this file. This value MUST be one of the following.

Value	Meaning
0x01	File batching data transfer is required to transfer this file.

### 2.2.2.14 SYNC\_CHANGE\_BATCH

The SYNC\_CHANGE\_BATCH structure defines the metadata that describes the changes to be synchronized.

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
Files (variable)																															
...																															
DeviceNames (variable)																															
...																															
SyncMetadata (variable)																															

**Files (variable):** A VECTOR\_FILE\_METADATA\_ENTRY structure containing metadata information for all files in the sync batch.

**DeviceNames (variable):** An array of VECTOR\_STRING structures that contains a list of the names of the devices that generated the changes to the files in this batch.

**SyncMetadata (variable):** A stream of bytes in SYNC\_BLOB format that represents the serialized SYNC\_CHANGE\_INFORMATION structure (specified in [\[MS-FSVCA\]](#) section 2.14).

### 2.2.2.15 SYNC\_MD5HASH

The SYNC\_MD5HASH structure defines the serialization format used by this protocol for MD5 hash data. The calculation of the MD5 hash is specified in [\[RFC1321\]](#).

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
High																															
...																															
Low																															
...																															

**High (8 bytes):** A 64-bit unsigned integer containing the first 64 bits of the MD5 hash data.

**Low (8 bytes):** A 64-bit unsigned integer containing the last 64 bits of the MD5 hash data.

### 2.2.2.16 VECTOR\_POLICY\_ENTRY

The VECTOR\_POLICY\_ENTRY structure represents a collection of POLICY\_ENTRY structures, as specified in section [2.2.2.3](#), in the following format:

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
NumEntries																															
EntryStream (variable)																															
...																															

**NumEntries (4 bytes):** A 32-bit unsigned integer containing the number of entries in the vector.

**EntryStream (variable):** An array of zero or more POLICY\_ENTRY structures representing client policy entry elements.

### 2.2.2.17 VECTOR\_FILE\_METADATA\_ENTRY

The VECTOR\_FILE\_METADATA structure represents a collection of FILE\_METADATA\_ENTRY structures, as specified in section [2.2.2.5](#), in the following format:

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
NumEntries																															
EntryStreamBytes (variable)																															
...																															

**NumEntries (4 bytes):** A 32-bit unsigned integer containing the number of entries in the vector.

**EntryStreamBytes (variable):** An array of zero or more FILE\_METADATA\_ENTRY structures representing metadata information for a file.

### 2.2.2.18 VECTOR\_FILE\_INFO\_INPUT\_ENTRY

The VECTOR\_FILE\_INFO\_INPUT structure represents a collection of FILE\_INFO\_INPUT\_ENTRY structures, as specified in section [2.2.2.6](#), in the following format:

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
NumEntries																															
EntryStreamBytes (variable)																															

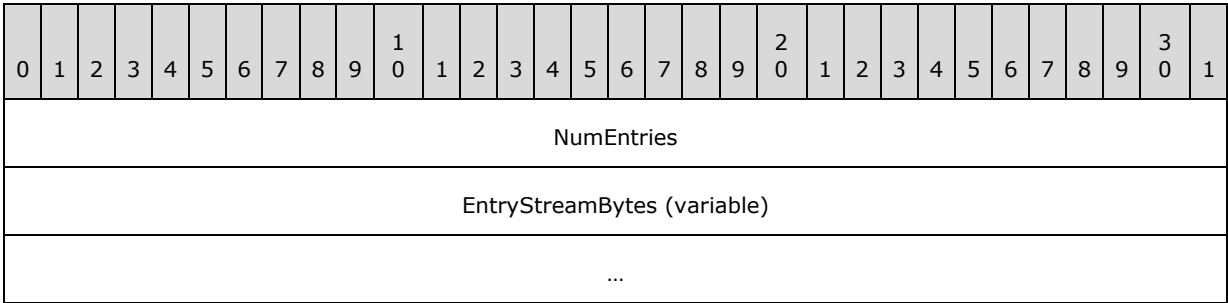
...

**NumEntries (4 bytes):** A 32-bit unsigned integer containing the number of entries in the vector.

**EntryStreamBytes (variable):** An array of zero or more FILE\_INFO\_INPUT\_ENTRY structures representing the sync preparation information of a file.

### 2.2.2.19 VECTOR\_FILE\_INFO\_ENTRY

The VECTOR\_FILE\_INFO\_ENTRY structure represents a collection of FILE\_INFO\_ENTRY structures, as specified in section [2.2.2.7](#), in the following format:

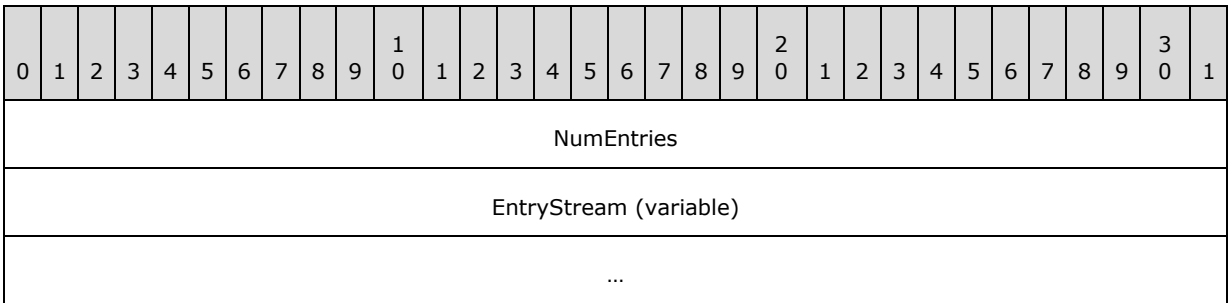


**NumEntries (4 bytes):** A 32-bit unsigned integer containing the number of entries in the vector.

**EntryStreamBytes (Vector):** An array of zero or more FILE\_INFO\_ENTRY structures representing the server sync preparation information of a file.

### 2.2.2.20 VECTOR\_FILE\_STATUS\_ENTRY

The VECTOR\_FILE\_STATUS\_ENTRY structure represents a collection of FILE\_STATUS\_ENTRY structures, as specified in section [2.2.2.8](#), in the following format:



**NumEntries (4 bytes):** A 32-bit unsigned integer containing the number of entries in the vector.

**EntryStream (variable):** An array of zero or more FILE\_STATUS\_ENTRY structures representing information on a file commit in a sync process.

### 2.2.2.21 VECTOR\_UPLOAD\_ENTRY

The VECTOR\_UPLOAD\_ENTRY structure represents a collection of UPLOAD\_ENTRY structures, as specified in section [2.2.2.9](#), in the following format:

0	1	2	3	4	5	6	7	8	9	1	0	1	2	3	4	5	6	7	8	9	2	0	1	2	3	4	5	6	7	8	9	3	0	1
NumEntries																																		
EntryStreamBytes (variable)																																		
...																																		

**NumEntries (4 bytes):** A 32-bit unsigned integer containing the number of entries in the vector.

**EntryStreamBytes (variable):** An array of zero or more UPLOAD\_ENTRY structures representing information of the data for a file.

### 2.2.2.22 VECTOR\_UPLOAD\_RESPONSE\_ENTRY

The VECTOR\_UPLOAD\_RESPONSE\_ENTRY structure represents a collection of UPLOAD\_RESPONSE\_ENTRY structures, as specified in section [2.2.2.10](#), in the following format:

0	1	2	3	4	5	6	7	8	9	1	0	1	2	3	4	5	6	7	8	9	2	0	1	2	3	4	5	6	7	8	9	3	0	1
NumEntries																																		
EntryStream (variable)																																		
...																																		

**NumEntries (4 bytes):** A 32-bit unsigned integer containing the number of entries in the vector.

**EntryStream (variable):** An array of zero or more UPLOAD\_RESPONSE\_ENTRY structures representing server response data for a file being uploaded.

### 2.2.2.23 VECTOR\_DOWNLOAD\_ENTRY

The VECTOR\_DOWNLOAD\_ENTRY structure represents a collection of DOWNLOAD\_ENTRY structures, as specified in section [2.2.2.11](#), in the following format:

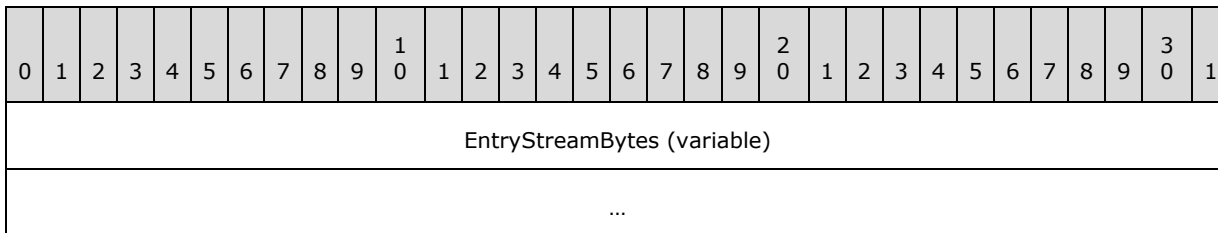
0	1	2	3	4	5	6	7	8	9	1	0	1	2	3	4	5	6	7	8	9	2	0	1	2	3	4	5	6	7	8	9	3	0	1
NumEntries																																		
EntryStreamBytes (variable)																																		
...																																		

**NumEntries (4 bytes):** A 32-bit unsigned integer containing the number of entries in the vector.

**EntryStreamBytes (variable):** An array of zero or more DOWNLOAD\_ENTRY structures representing information on the data for a file being downloaded.

### 2.2.2.24 VECTOR\_DOWNLOAD\_RESPONSE\_ENTRY

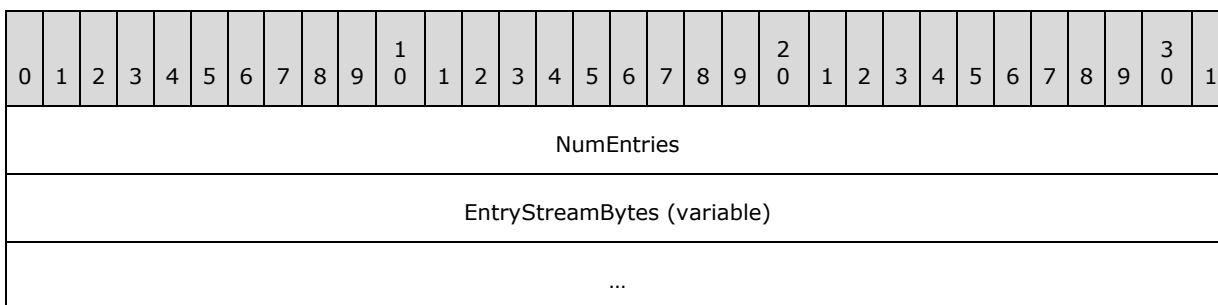
The VECTOR\_DOWNLOAD\_RESPONSE\_ENTRY structure represents a collection of DOWNLOAD\_RESPONSE\_ENTRY structures, as specified in section [2.2.2.12](#), in the following format:



**EntryStreamBytes (variable):** An array of zero or more DOWNLOAD\_RESPONSE\_ENTRY structures representing server response data for a file being downloaded.

### 2.2.2.25 VECTOR\_FILE\_DOWNLOAD\_INFO\_ENTRY

The VECTOR\_FILE\_DOWNLOAD\_INFO\_ENTRY structure represents a collection of FILE\_DOWNLOAD\_INFO\_ENTRY structures, as specified in section [2.2.2.13](#), in the following format:

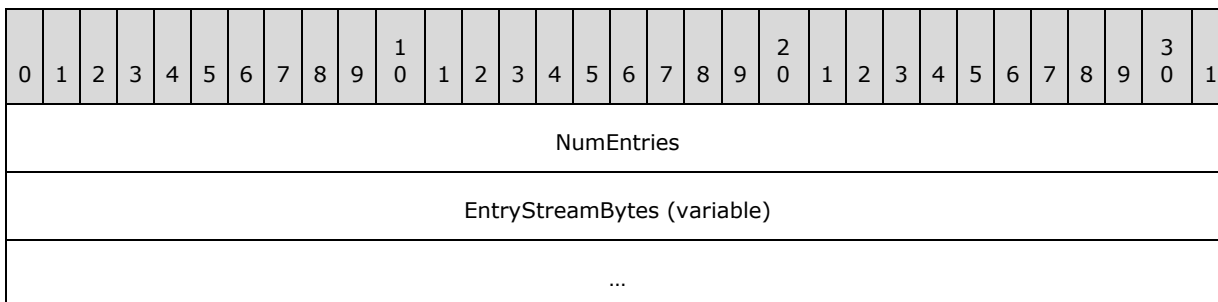


**NumEntries (4 bytes):** A 32-bit unsigned integer containing the number of entries in the vector.

**EntryStreamBytes (Vector):** An array of zero or more FILE\_DOWNLOAD\_INFO\_ENTRY structures representing information that is required to download file content.

### 2.2.2.26 VECTOR\_STRING

The VECTOR\_STRING structure represents a collection of strings in the following format:

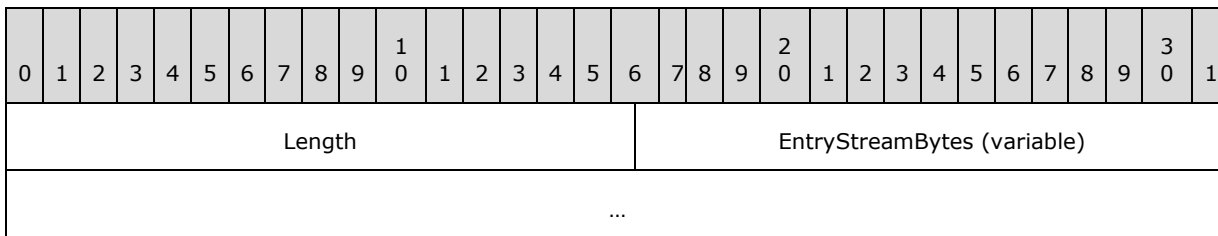


**NumEntries (4 bytes):** A 32-bit unsigned integer containing the number of ECS\_STRING structures in **EntryStreamBytes**.

**EntryStreamBytes (variable):** An array of ECS\_STRING structures, as specified in section [2.2.2.27](#).

### 2.2.2.27 ECS\_STRING

The ECS\_STRING structure represents a string in the following format:



**Length (2 bytes):** A 16-bit unsigned integer containing the size, in bytes, of the **EntryStreamBytes** field.

**EntryStreamBytes (variable):** A UTF-8 string.

## 3 Protocol Details

The resources defined by this protocol are categorized as follows:

- Service Discovery Resources
- Server Changes Resources
- User Configuration Resources
- Session Resources
- Data Transfer Server Resources

The "Request body" and "Response body" entries are transmitted in the same order they are listed in this document.

The following sections describe the processing of each resource in the above mentioned categories.

### 3.1 ServiceDiscovery Server Details

#### 3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in the Enterprise Client Synchronization Protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

The server implements the following:

**HostURLPrefixList:** A list of URLs for the servers that expose the synchronization namespace.

**PartnershipIdList:** A list of identifiers in the string format that relate a client with a specific sync share.

**ClientRequestsCountTable:** A table of number of pending requests indexed by PartnershipId, as specified in **Per ClientRequestsCount**.

#### **Per ClientRequestsCount**

**UserId:** An identifier that uniquely identifies a user.

**Count:** A numeric value specifying the number of pending requests on the server.

#### 3.1.2 Timers

None.

#### 3.1.3 Initialization

The server MUST initialize the following:

**ClientRequestsCountTable** MUST be set to empty.

**HostURLPrefixList** MUST be set to a list of server URLs in an implementation-specific manner.



**PartnershipIdList** MUST be set to empty.

### 3.1.4 Higher-Layer Triggered Events

None.

### 3.1.5 Message Processing Events and Sequencing Rules

Resource	Description
Server Discovery	The resource used to enumerate all servers exposing the namespace for sync.
Share Discovery	The resource uses to discover the sync share on the server.
Server Capabilities	The resource used to query the capabilities of the sync service.

The responses to all the operations can result in the following status codes.

Status code	Reason phrase	Description
200	OK	Server is discovered and URL is sent back in response body.
404	Not Found	Cannot find the sync server URL for the given user. Either the user doesn't have a value for the attribute or the AD schema does not support the attribute.
500	Internal Server Error	Unexpected server error. Specific error code in the request error header.

#### 3.1.5.1 Server Discovery

The Server Discovery resource represents a list of servers that expose the synchronization namespace.

The following operations are allowed to be performed on this resource.

HTTP method	Description
GET	Enumerate a list of servers that expose the synchronization namespace.

##### 3.1.5.1.1 GET

This operation is transported by an HTTP GET request.

The operation can be invoked through the following URI suffix:

```
Sync/{version}/Discover/ServerUrl
```

The following is an example of a complete URI for this operation:

```
https://contoso.com/Sync/1.0/Discover/ServerUrl
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
x-ecs-devicename	Optional	A string representing the name and operating system of the device issuing the request in the following format: {Device Name, OS Family, OS Major Version, OS Minor Version, Sync App Version} Example: x-ecs-devicename: { homelp, Windows, 6.2, 9200, 1.0}

The response message for this operation contains the following HTTP headers.

Response header	Usage	Value
x-ecs-request-error	Optional	A string describing request failure information. This MUST be a string representation of the HRESULT in hexadecimal format. Example: x-ecs-request-error: "0x8007005"

The response message for this operation can result in the following status codes.

Status code	Description
200	Server is discovered and URL is sent back in response body.
404	Cannot find sync server URL for the given user. Either the user doesn't have a value for the attribute or the AD schema does not support the attribute.
500	Unexpected server error. Specific error code in the request error header.

### 3.1.5.1.1.1 Request Body

None.

### 3.1.5.1.1.2 Response Body

The response for this method contains the following:

Entry	Type
ServerUrls	VECTOR_STRING

**ServerUrls:** A list of host server URL prefixes that the client can connect to for all subsequent sync operations.

### 3.1.5.1.1.3 Processing Details

The server MUST fill the response body with the URLs from **HostURLPrefixList**.

### 3.1.5.2 Share Discovery

This resource represents the sync share on the server for a user.

The following operations are allowed to be performed on this resource.

HTTP method	Description
GET	Query the properties of the sync share

#### 3.1.5.2.1 GET

This operation is transported by an HTTP GET request.

The operation can be invoked through the following URI suffix:

```
Sync/{version}/Discover/Share
```

The following is an example of a complete URI for this operation:

```
https://contoso.com/Sync/1.0/Discover/Share
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
x-ecs-share-type	Optional	A string representing the type of the share to discover. If this header is present in the request, its value MUST be set to "User Data".

The response message for this operation contains the following HTTP headers.

Response header	Usage	Value
x-ecs-request-error	Optional	A string describing request failure information. This MUST be a string representation of the HRESULT in hexadecimal format. Example: x-ecs-request-error: "0x8007005"

The response message for this operation can result in the following status codes.

Status code	Description
200	Success
404	There is no sync share for the user (possible reasons include the type not existing or the user not having access).
500	Unexpected server error.

### 3.1.5.2.1.1 Request Body

None.

### 3.1.5.2.1.2 Response Body

The response for this method contains the following:

Entry	Type
PartnershipId	ECS_STRING
EnterpriseId	ECS_STRING
DataSize	ULONG64

**PartnershipId:** A VECTOR\_STRING structure containing the unique ID on the server that represents a combination of the client and the sync share. This ID will be used by the client as an x-ecs-partnershipID header in subsequent requests.

**EnterpriseId:** An identifier that signifies the owning entity for all content synchronized with the share.

**DataSize:** Total size of the user data on the sync share for this user in bytes.

### 3.1.5.2.1.3 Processing Details

If the request contains x-ecs-share-type header and its value is not "User Data", the server MUST fail the request with a status code of 404.

The server MUST generate a unique ID in the form of a string and set it to **PartnershipId** in the response body. The server MUST add the generated ID to **PartnershipIdList**.

If the server cannot evaluate the size of the user data, it MUST set DataSize in the response body to the decimal value of 18446744073709551615, that is, the maximum value allowed by ULONG64.

The server MUST set **EnterpriseId** to an administratively configured value.

The server MUST allocate a **ClientRequestsCount** object and MUST be inserted into **ClientRequestsCountTable**. **ClientRequestsCount.UserId** MUST be set in an implementation-specific manner and **ClientRequestsCount.Count** MUST be set to zero.

### 3.1.5.3 Server Capabilities

The Server Capabilities resource represents information about the protocol support on the server.

The following operations are allowed to be performed on this resource.

HTTP method	Description
GET	Query protocol support information from the server.

#### 3.1.5.3.1 GET

This operation is transported by an HTTP GET request.

The operation can be invoked through the following URI suffix:

`Sync/{version}/Capabilities`

The following is an example of a complete URI for this operation:

`https://contoso.com/Sync/1.0/Capabilities`

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
x-ecs-devicename	Optional	A string representing the name and operating system of the device issuing the request in the following format: {Device Name, OS Family, OS Major Version, OS Minor Version, Sync App Version} Example: x-ecs-devicename: {home!p, Windows, 6.2, 9200, 1.0}

The response message for this operation contains the following HTTP header.

Response header	Usage	Value
x-ecs-request-error	Optional	A string describing request failure information. This MUST be a string representation of the HRESULT in hexadecimal format. Example: x-ecs-request-error: "0x8007005"

The response message for this operation can result in the following status codes.

Status code	Description
200	Success
500	Unexpected server error.

### 3.1.5.3.1.1 Request Body

None.

### 3.1.5.3.1.2 Response Body

The response for this method contains the following:

Entry	Type
ProtocolType	UCHAR

**ProtocolType:** The data transfer type that the server supports. This MUST be set to the following value:

Value	Description
0x01	File batching data transfer protocol.

### 3.1.5.3.1.3 Processing Details

If the server supports file batching for data transfer, the server MUST set the bit to 0x01 in ProtocolType in the response body.

### 3.1.6 Timer Events

None.

### 3.1.7 Other Local Events

None.

## 3.2 DetectServerChanges Server Details

### 3.2.1 Abstract Data Model

None.

### 3.2.2 Timers

None.

### 3.2.3 Initialization

None.

### 3.2.4 Higher-Layer Triggered Events

None.

### 3.2.5 Message Processing Events and Sequencing Rules

Resource	Description
Detect Server Changes	The resource used to poll for the server-side changes.

The responses to all the operations can result in the following status codes.

Status code	Reason phrase	Description
200	OK	Server knowledge has changed.
304	Not Modified	Server knowledge did not change.

#### 3.2.5.1 Detect Server Changes

The Detect Server Changes resource allows the client to poll for the changes on the server.

HTTP method	Description
HEAD	Query the changes on the server.

### 3.2.5.1.1 HEAD

This operation is transported by an HTTP HEAD request.

The operation can be invoked through the following URI suffix:

`Sync/{version}/Changes`

The following is an example of a complete URI for this operation:

`https://contoso.com/Sync/1.0/Changes`

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
If-None-Match	Optional	The ETag from a previous response
x-ecs-partnershipID	Required	The PartnershipID for the sync target share.
x-ecs-devicename	Optional	A string, as specified in section <a href="#">2.2.1.2</a> .

The response message for this operation contains the following HTTP headers.

Response header	Usage	Value
ETag	Optional	A value identifying the version of the data for this PartnershipID.
x-ecs-changes-URL	Optional	A URL to be used by the client to poll for changes on the server.

The response message for this operation can result in the following status codes.

Status code	Description
200	Server knowledge has changed; at least one of the two ETags is different.
304	Server knowledge did not change.

#### 3.2.5.1.1.1 Request Body

None.

#### 3.2.5.1.1.2 Response Body

None.

#### 3.2.5.1.1.3 Processing Details

If there is any change in the URL where the client should poll, the server SHOULD [<4>](#) set the new URI in the x-ecs-changes-URL response header and send a response with status code 200.

If the *If-None-Match* header is not provided in the request, the server MUST set a version of the data for the provided **PartnershipID** in the *ETag* response header and MUST set the status code to 200.

If the *If-None-Match* header is provided in the request and its value is different from the version of the data for the provided **PartnershipID**, the server MUST set a version of the data for the provided **PartnershipID** in the *ETag* response header and MUST set the status code to 200.

If the *If-None-Match* header is provided in the request and its value is same as the version of the data for the provided PartnershipID, the server MUST set the status code to 304.

### 3.2.6 Timer Events

None

### 3.2.7 Other Local Events

None.

## 3.3 UserConfiguration Server Details

### 3.3.1 Abstract Data Model

None.

### 3.3.2 Timers

None.

### 3.3.3 Initialization

None.

### 3.3.4 Higher-Layer Triggered Events

None.

### 3.3.5 Message Processing Events and Sequencing Rules

Resource	Description
User Configuration	The resource that represents user configuration information for a sync target share.

#### 3.3.5.1 User Configuration

HTTP method	Description
GET	Query user configuration information for a sync target share.

##### 3.3.5.1.1 GET

This operation is transported by an HTTP GET request.

The operation can be invoked through the following URI suffix:



Sync/{version}/Configuration

The following is an example of a complete URI for this operation:

https://contoso.com/Sync/1.0/Configuration

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
x-ecs-partnershipID	Required	The PartnershipID for the sync target that was received in the GET response for the Share Discovery resource.
x-ecs-domain		The FQDN of the domain to which the client is part of. Omit or use empty string if the client is not part of a domain.
x-ecs-devicename	Optional	A string, as specified in section <a href="#">2.2.1.2</a> .

### 3.3.5.1.1.1 Request Body

None.

### 3.3.5.1.1.2 Response Body

The response for this method contains the following:

Entry	Type
Quota Usage	QUOTA_USAGE_ENTRY structure
Policy List	VECTOR_POLICY_ENTRY
Length	Word
AdminInfo	UTF-8 String

**QuotaUsage:** Describes the current data usage for the user.

**PolicyList:** A list of policy entries that describe how the client needs to set up its target directory. The policies processed first take precedence.

**Length:** This indicates the size of the **AdminEmail** field.

**AdminInfo:** A server-configured **String** value containing the server admin contact information. If the **Length** field is 0, this field is empty.

### 3.3.5.1.1.3 Processing Details

The server MUST query the quota usage on the sync share in an implementation-specific manner, construct the QUOTA\_USAGE\_ENTRY structure as specified in section [2.2.2.2](#), and set the Quota Usage field in the response body to this structure. If the server cannot determine the current quota usage for the user, it MUST set the **UserUsage** field of the QUOTA\_USAGE\_ENTRY structure to the decimal value of 18446744073709551615, that is, the maximum value allowed by ULONG64.

The server MUST query all the policies that apply to the sync share and MUST construct a VECTOR\_POLICY\_ENTRY structure using each policy being applied. The server MUST set the Policy List in the response body to the VECTOR\_POLICY\_ENTRY structure. The server MUST set **AdminInfo** in an implementation-specific manner.

### 3.3.6 Timer Events

None.

### 3.3.7 Other Local Events

None.

## 3.4 PeerSynchronizationSession Server Details

### 3.4.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in the Enterprise Client Synchronization Protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

#### 3.4.1.1 Global

The server implements the following:

**SessionList:** A list of **Sessions** defined in section [3.4.1.2](#).

#### 3.4.1.2 Per Session

**SessionLocationURL:** The URL prefix used to access this session.

**SessionId:** A string that identifies the session.

**ChangeBatchList:** This contains the change batches as specified in section [3.4.1.2.1](#).

**LastSentToken:** This contains the last x-ecs-continue token sent to the client.

**LastReceivedToken:** This contains the last x-ecs-continue token sent by the client.

**ClientID:** An identifier for the client.

**SessionType:** The type of session. This contains one of the values specified in section [3.4.5.1.1.1](#).

#### 3.4.1.2.1 Per ChangeBatch

**MetaData:** Contains the metadata describing the changes to be synchronized, as specified in section [2.2.2.16](#).

**FileInfoList:** Contains the collection of FILE\_DOWNLOAD\_INFO\_ENTRY structures as specified in section [2.2.2.15](#).

### 3.4.2 Timers

None.

### 3.4.3 Initialization

The server MUST set **SessionList** to empty.

The server MUST set **LastSentToken** and **LastReceivedToken** to empty.

### 3.4.4 Higher-Layer Triggered Events

None.

### 3.4.5 Message Processing Events and Sequencing Rules

If x-ecs-partnershipID is not present in the request header, the server MUST set x-ecs-request-error to ECS\_E\_SYNC\_REQUIRED\_HTTP\_HEADER\_MISSING and HTTP Status Code to 400. The server MUST send the response to the client.

The server MUST obtain share name, user folder, and user identifier from x-ecs-partnershipID in the request in an implementation-specific manner. If none of them are available on the server, the server MUST set x-ecs-request-error to ECS\_E\_SYNC\_INVALID\_PROTOCOL\_FORMAT and HTTP Status Code to 400. The server MUST send the response to the client.

The server MUST look up **ClientRequestsCount** in **ClientRequestsCountTable** where **ClientRequestsCount.UserId** matches with the user identifier of the requesting user, obtained from x-ecs-partnershipID in the request.

If **ClientRequestsCount.Count** is greater than or equal to an implementation-specific value [<5>](#), the server MUST set x-ecs-request-error to ECS\_E\_SYNC\_SERVER\_BUSY and HTTP status code to 503. The server MUST send the response to the client.

If the client is required to re-establish sync partnership with the server, the server MUST set x-ecs-request-error to ECS\_E\_DISCOVERY\_NEEDED and HTTP Status Code to 403. The server MUST send the response to the client.

If access to the server is paused, the server MUST set x-ecs-request-error to ECS\_E\_ERROR\_SYNC\_SHARE\_BLOCKED and HTTP Status Code to 503. The server MUST send the response to the client.

The server MUST increment **ClientRequestsCount.Count** by 1.

The server MUST continue to process the request. After sending the response to the client, the server MUST decrement **ClientRequestsCount.Count** by 1.

Resource	Description
Create Session	The resource used to create a new session on the server.
Sync Batch Parameters	The resource used to retrieve or update synchronization batch parameters.
Prepare Batch	The resource used to send information to the server to prepare for the change batch.
Upload Batch	The resource used to commit a change batch on the server.
Delete Session	The resource used to delete a sync session on the server.
Download Batch	The resource used to receive a change batch from the server to the client.

The responses to all the operations can result in the following status codes.

Status code	Reason phrase	Description
200	OK	Success. See individual methods on each resource in the subsequent sections for more details.
201	Created	Session created.
400	Bad Request	Batch is out of sequence, or the last batch has already been received.
404	Not Found	No session exists with the specified ID.
202	Accepted	An asynchronous operation for retrieving the sync batch parameters has started.
409	Conflict	Batch already received.

### 3.4.5.1 Create Session

The Create Session resource facilitates a client to start a sync session on the server.

HTTP method	Description
PUT	Create a new sync session.

#### 3.4.5.1.1 PUT

This operation is transported by an HTTP PUT request.

The operation can be invoked through the following URI suffix:

```
Sync/{version}/Session
```

The following is an example of a complete URI for this operation:

```
https://contoso.com/Sync/1.0/Session
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
x-ecs-partnershipID	Required	The PartnershipID for the sync target.
x-ecs-devicename	Optional	A string as specified in section <a href="#">2.2.1.2</a> .

The response message for this operation contains the following HTTP headers.

Response header	Usage	Value
x-ecs-request-error	Optional	A string describing request failure information. This MUST be a string representation of the HRESULT in hexadecimal format.

Response header	Usage	Value
		Example: x-ecs-request-error: "0x8007005"
x-ecs-admin-contact	Optional	A string representing the email contact of the server administrator.
x-ecs-metadata-version		A string containing the version of the server metadata for the current sync replica.
x-ecs-session-id	Optional	A GUID in string format or an implementation-specific string that identifies the session. The client MUST use this value to build the complete URL in the subsequent requests on a session.

The response message for this operation can result in the following status codes.

Status code	Description
200	The response contains the URI prefix for a session that already exists.
201	The response contains the URI prefix for a newly created session.
400	Invalid session type.

#### 3.4.5.1.1.1 Request Body

The request body for this method MUST contain the following.

Entry	Type
Type	UCHAR
ClientId	GUID

**Type:** The type of data transfer from the client's perspective. This MUST be one of the following values.

Value	Meaning
0x01	Upload
0x02	Download
0x03	Upload with full enumeration
0x04	Download with full enumeration

**ClientId:** An identifier for the client.

#### 3.4.5.1.1.2 Response Body

None.

### 3.4.5.1.1.3 Processing Details

If the **Type** field of the request body does not contain one of the listed values, the server MUST fail the request with status code 400 and set x-ecs-request-error to a string representation of 0x00000012.

The server MUST look up all **Sessions** in **SessionList** where **Session.ClientID** matches ClientId in the request. If the number of **Sessions** found is greater than or equal to an implementation-specific value [<6>](#), the server MUST set x-ecs-request-error to ECS\_E\_SYNC\_TOO\_MANY\_SESSIONS.

The Server MUST search for the **Session** object in **SessionList** where **Session.ClientID** matches with the ClientId in the request.

If **Session** is not found, the server MUST create a new **Session** and initialize the following values:

- **SessionLocationURL** MUST be set to an implementation-specific URL that provides access to this session.
- **SessionId** MUST be set to an implementation-specific string that identifies the session.
- **ClientID** MUST be set to ClientId in the request.
- **SessionType** MUST be set to Type in the request.

The server MUST insert the **Session** in **SessionList** and the server MUST set the HTTP status code to 201.

If **Session** is found and **Session.SessionType** matches with Type in the request, the server MUST set the HTTP status code to 200.

If **Session** is found and **SessionType** does not match with Type in the request, the server MUST create a new session and set the values as specified above.

If a session is found and Session.SessionType is 0x03 (Upload with full enumeration) or 0x04 (Download with full enumeration), the server MUST set x-ecs-request-error to ECS\_E\_SYNC\_SESSION\_BUSY and HTTP status code to 503. The server MUST send the response to the client.

If a session is found and in the following conditions, the server MUST create a new Session object:

- Session.SessionType is 0x01 (Upload) and Type in the request is 0x03 (Upload with full enumeration).
- Session.SessionType is 0x02 (Download) and Type in the request is 0x04 (Download with full enumeration).
- Session.SessionType is 0x01 (Upload) or 0x03 (Upload with full enumeration) and Type in the request is 0x02 (Download) or 0x04 (Download with full enumeration).
- Session.SessionType is 0x02 (Download) or 0x04 (Download with full enumeration) and Type in the request is 0x01 (Upload) or 0x03 (Upload with full enumeration).

The server MUST initialize the Session object with the following values:

- SessionLocationURL MUST be set to an implementation-specific URL that provides access to this session.
- SessionId MUST be set to Session.SessionId.

- ClientID MUST be set to ClientId in the request.
- SessionType MUST be set to Type in the request.

The server MUST set HTTP status code to 200 and MUST send the response to the client.

### 3.4.5.2 Sync Batch Parameters

The Sync Batch Parameters resource represents synchronization batch parameters that are used to calculate the changes that need to be transmitted in an upload or download scenario.

HTTP method	Description
GET	Retrieve the synchronization batch parameters (that is, server knowledge) from the server.
PUT	Update the client's synchronization batch parameters (that is, client knowledge) to the server.

#### 3.4.5.2.1 GET

The GET method on the Sync Batch Parameters resource is issued by the client to obtain the server's sync knowledge in an upload scenario.

This operation is transported by an HTTP GET request.

The operation can be invoked through the following URI suffix on the *x-ecs-session-location-url* returned in the response header of the PUT method on the Creation Session resource:

```
/SyncBatchParameters
```

The following is an example of a complete URI for this operation:

```
https://contoso.com/Sync/1.0/Session/{60b8ca1d-59f3-4495-b299-f6cad89ada3b}/SyncBatchParameters
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
x-ecs-partnershipID	Required	The PartnershipID for the sync target share returned by the server in the GET response on Share Discovery resource.

The response message for this operation contains the following HTTP headers.

Response header	Usage	Value
x-ecs-request-error	Optional	A string describing request failure information. This MUST be a string representation of the HRESULT in hexadecimal format. Example: x-ecs-request-error: "0x8007005"

The response message for this operation can result in the following status codes.

Status code	Description
200	Batch parameters are returned in the response body.
404	No session exists with the specified ID.

### 3.4.5.2.1.1 Request Body

None.

### 3.4.5.2.1.2 Response Body

The response body for this method contains the following:

Entry	Type
SyncKnowledge	SYNC_BLOB structure
BatchLimits	BATCH_LIMITS_ENTRY structure

**SyncKnowledge:** The SYNC\_BLOB structure's BlobData element contains a SYNC\_KNOWLEDGE structure, as defined in [\[MS-FSVCA\]](#) section 2.3, that represents the serialized server sync knowledge.

**BatchLimits:** Parameters describing batch characteristics.

### 3.4.5.2.1.3 Processing Details

The server MUST query the sync knowledge, as specified in [\[MS-FSVCA\]](#) section 3.1.4.1. The server MUST set the received sync knowledge as SyncKnowledge, set **BatchLimits.MaxFileSize** and **BatchLimits.MaxFileCount** with implementation-specific [<7>](#) values in the response, and set the status code to 200.

### 3.4.5.2.2 PUT

The PUT method on the Sync Batch Parameters resource is issued by the client to update the client's sync knowledge to the server in an upload scenario.

This operation is transported by an HTTP PUT request.

The operation can be invoked through the following URI suffix:

```
/SyncBatchParameters
```

The following is an example of a complete URI for this operation:

```
https://contoso.com/Sync/1.0/Session/{60b8ca1d-59f3-4495-b299-f6cad89ada3b}/SyncBatchParameters
```

The request message for this operation contains the following HTTP headers.



Request header	Usage	Value
x-ecs-partnershipID	Required	The PartnershipID for the sync target share returned by the server in the GET response on Share Discovery resource.

The response message for this operation can result in the following status codes.

Status code	Description
200	Success
202	Background processing initiated.
404	No session exists with the specified ID.
409	Client knowledge already received.

### 3.4.5.2.2.1 Request Body

The request body for this method contains the following.

Entry	Type
SyncKnowledge	SYNC_BLOB structure
BatchLimits	BATCH_LIMITS_ENTRY structure
FullEnumerationLowerBound	SYNC_GID structure

**SyncKnowledge:** The SYNC\_BLOB structure's BlobData element contains a SYNC\_KNOWLEDGE structure as defined in [\[MS-FSVCA\]](#) section 2.3, that represents the serialized server sync knowledge.

**BatchLimits:** Parameters describing batch characteristics.

**FullEnumerationLowerBound:** The sync item ID that represents the lower bound of the changes to be included in the change batch generated by the server. This value is ignored if the session is not a full enumeration one.

### 3.4.5.2.2.2 Response Body

The response body for this method contains the following.

Entry	Type
TotalFileCount	ULONG
TotalFileSize	ULONGLONG

**TotalFileCount:** Total number of files that have changed on the server.

**TotalFileSize:** Sum of the sizes of all the files that have changed on the server.

### 3.4.5.2.2.3 Processing Details

Upon receiving this request, the server MUST generate a change batch, as specified in [\[MS-FSVCA\]](#) section 3.1.4.2, by passing 'TRUE' to **IsRecoverySynchronization** as input if **Session.SessionType** is 0x04, or 'FALSE' if **Session.SessionType** is 0x02, corresponding to the files to be downloaded to the client, and set **TotalFileCount** and **TotalFileSize** with implementation-specific<8> values.

The server MUST create an entry in **Session.ChangeBatchList** and set **ChangeBatch.MetaData** and **ChangeBatch.FileInfoList** to the generated change batch.

### 3.4.5.3 Prepare Batch

The Prepare Batch resource is used by the client to request server-side preparation for a file upload in a sync process.

HTTP method	Description
PUT	Update server with the file information necessary to prepare for the upload.

#### 3.4.5.3.1 PUT

This operation is transported by an HTTP PUT request.

The operation can be invoked through the following URI suffix:

```
/PrepareBatch/{BatchIndex}
```

The BatchIndex in the URI suffix is an unsigned integer representing a specific batch that the client wants to upload.

The following is an example of a complete URI for this operation:

```
https://contoso.com/Sync/1.0/Session/{60b8ca1d-59f3-4495-b299-f6cad89ada3b}/PrepareBatch/3
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
x-ecs-partnershipID	Required	The PartnershipID for the sync target share returned by the server in the GET response on Share Discovery resource.

The response message for this operation contains the following HTTP headers.

Response header	Usage	Value
x-ecs-request-error	Optional	A string describing request failure information. This MUST be a string representation of the HRESULT in hexadecimal format. Example: x-ecs-request-error: "0x8007005"

The response message for this operation can result in the following status codes.

Status code	Description
200	Batch parameters are returned in the response body.
404	No session exists with the specified ID.

### 3.4.5.3.1.1 Request Body

The request body for this method contains the following.

Entry	Type
FileVector	VECTOR_FILE_INFO_INPUT_ENTRY

**FileVector:** A list of file info input entries that need to be prepared for the synchronization process.

### 3.4.5.3.1.2 Response Body

The response body for this method contains the following:

Entry	Type
FileList	VECTOR_FILE_INFO_ENTRY

**FileList:** A list of file status entries indicating the status of the change batch preparation.

### 3.4.5.3.1.3 Processing Details

None.

### 3.4.5.4 Upload Batch

The Upload Batch resource is used by the client to commit a change batch on the server.

HTTP method	Description
PUT	Commit change batch.

#### 3.4.5.4.1 PUT

This operation is transported by an HTTP PUT request.

The operation can be invoked through the following URI suffix:

```
/UploadBatch/{BatchIndex}
```

The BatchIndex in the URI suffix is an unsigned integer representing a specific batch that the client wants to upload.

The following is an example of a complete URI for this operation:

```
https://contoso.com/Sync/1.0/Session/{60b8ca1d-59f3-4495-b299-f6cad89ada3b}/UploadBatch/3
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
x-ecs-partnershipID	Required	The PartnershipID for the sync target share returned by the server in the GET response on Share Discovery resource.

The response message for this operation contains the following HTTP headers.

Response header	Usage	Value
x-ecs-request-error	Optional	A string describing request failure information. This MUST be a string representation of the HRESULT in hexadecimal format. Example: x-ecs-request-error: "0x8007005"

The response message for this operation can result in the following status codes.

Status code	Description
200	Batch has been processed synchronously.
202	Batch has been accepted and will be processed in the background.
400	Batch is out of sequence, or the last batch has already been received.
404	Session not found.
409	Batch already received.

#### 3.4.5.4.1.1 Request Body

The request body for this method contains the following.

Entry	Type
SerializedSyncBatch	SYNC_CHANGE_BATCH

**SerializedSyncBatch:** The metadata describing the changes to be synchronized.

#### 3.4.5.4.1.2 Response Body

The response body for this method contains the following.

Entry	Type
FileStatusList	VECTOR_FILE_STATUS_ENTRY

**FileStatusList:** A list of file status entries indicating the status of change batch commit operation.

### 3.4.5.4.1.3 Processing Details

If the client is suspended from committing a change batch on the server, the server MUST set `x-ecs-request-error` to `ECS_E_USER_SUSPENDED` and set HTTP Status Code to 503. The server MUST send the response to the client.

For each file metadata entry specified in `SerializedSyncBatch`, the server MUST perform the following:

- Commit the file metadata entry in an implementation-specific manner
- Construct `FILE_STATUS_ENTRY` structure
- Set `SyncItemId` to `FileId` in the request
- Set `Status` to zero if the entry is committed successfully. Otherwise `Status` is set to an implementation-specific error code.

The server MUST insert all structures constructed above into `FileStatusList` as specified in section [2.2.2.20](#).

The server MUST set HTTP Status Code to 200.

The server MUST send `FileStatusList` as the response to the client.

### 3.4.5.5 Delete Session

The Delete Session resource facilitates a client to remove a server session after a synchronization process is completed or aborted.

HTTP method	Description
DELETE	Delete server session.

#### 3.4.5.5.1 DELETE

This operation is transported by an HTTP DELETE request.

The client MUST use the `x-ecs-session-location-url` returned by the server in the Creation Session response (see section [3.4.5.1.1](#)) as the URI for this operation.

The following is an example of a complete URI for this operation:

`https://contoso.com/Sync/1.0/Session/{60b8ca1d-59f3-4495-b299-f6cad89ada3b}`

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
<code>x-ecs-partnershipID</code>	Required	The PartnershipID for the sync target share returned by the server in the GET response on Share Discovery resource.

The response message for this operation can result in the following status codes.

Status code	Description
200	The server will delete the session.
404	Session not found.

### 3.4.5.5.1.1 Request Body

None.

### 3.4.5.5.1.2 Response Body

None.

### 3.4.5.5.1.3 Processing Details

The server MUST extract the session ID from the URI in an implementation-specific manner and verify whether there is a **Session** that exists in the global **SessionList** with a matching session ID. If no session is found, the server MUST return a status code of 404. Otherwise, the server MUST delete the **Session** entry from **SessionList** and return a status code of 200.

### 3.4.5.6 Download Batch

The Download Batch resource is used by the client to obtain change batch information from the server in a download scenario.

HTTP method	Description
GET	Query server knowledge on a change batch.

#### 3.4.5.6.1 GET

The GET method on the Download Batch resource is issued by the client to obtain metadata information for all files in a change batch.

This operation is transported by an HTTP GET request.

The operation can be invoked through the following URI on the *x-ecs-session-location-url* that is returned in the response header of the PUT method on the Creation Session resource:

`/DownloadBatch`

The following is an example of a complete URI for this operation:

`https://contoso.com/Sync/1.0/Session/{60b8ca1d-59f3-4495-b299-f6cad89ada3b}/DownloadBatch`

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
x-ecs-	Required	The PartnershipID for the sync target share returned by the server in the

Request header	Usage	Value
partnershipID		GET response on Share Discovery resource.
x-ecs-continue	Optional	The continuation token received from the server in the prior download batch operation response header. This MUST be present in the request headers of all batches subsequent to the first batch of a sync process.

The response message for this operation contains the following HTTP headers.

Response header	Usage	Value
x-ecs-continue	Required	The continuation token that the client needs to use in the subsequent download batch request on this batch.

The response message for this operation can result in the following status codes.

Status code	Description
200	Success
400	Invalid continuation token.
404	No session exists with the specified ID.

#### 3.4.5.6.1.1 Request Body

None.

#### 3.4.5.6.1.2 Response Body

The response body for this method contains the following.

Entry	Type
BatchMetadata	SYNC_CHANGE_BATCH
DownloadInfo	VECTOR_FILE_DOWNLOAD_INFO_ENTRY

**BatchMetadata:** The metadata describing the changes to be synchronized.

**DownloadInfo:** A collection of FILE\_DOWNLOAD\_INFO\_ENTRY structures as specified in section [2.2.2.13](#).

#### 3.4.5.6.1.3 Processing Details

The server MUST fail the request with a status code of 400 for the following conditions.

- If there are no entries in **ChangeBatchList**.
- If the x-ecs-continue token is not present and **LastSentToken** is not empty.
- If the x-ecs-continue token is present and **LastSentToken** is empty.

- If the x-ecs-continue token is present and not equal to **LastSentToken** or **LastReceivedToken**.

If the x-ecs-continue token is not present, the server MUST return the first batch from **Session.ChangeBatchList**. The server MUST generate a token that points to the second batch. The server MUST additionally set x-ecs-continue in the response and **LastSentToken** to the newly generated token.

If **Session.SessionType** is 0x04, **ChangeBatch.Metadata** in the response body contains the entire list of SyncIds available on the server.

If the x-ecs-continue token is present and x-ecs-continue is equal to **LastSentToken** or **LastReceivedToken**, the server MUST search for the corresponding batch in **Session.ChangeBatchList**.

If no entry is found, the server MUST fail the request with the status code of 400. Otherwise, the server MUST return the corresponding batch, indicated by x-ecs-continue, from the **Session.ChangeBatchList**.

If the corresponding batch that is indicated by x-ecs-continue is not the last batch in **Session.ChangeBatchList**, the server MUST generate a new token that points to the next batch in **Session.ChangeBatchList**. Otherwise, the server MUST generate a token that does not point to any batch in **Session.ChangeBatchList**.

The server MUST set **LastReceivedToken** to x-ecs-continue. The server MUST also set x-ecs-continue in the response and **LastSentToken** to the newly generated token.

### 3.4.6 Timer Events

None.

### 3.4.7 Other Local Events

None.

## 3.5 ServerAPI Server Details

### 3.5.1 Abstract Data Model

None.

### 3.5.2 Timers

None.

### 3.5.3 Initialization

None.

### 3.5.4 Higher-Layer Triggered Events

None.



### 3.5.5 Message Processing Events and Sequencing Rules

Resource	Description
Upload Data	The resource used to send file data to the server.
Download Data	The resource used to receive file data from the server.

The responses to all the operations can result in the following status codes.

Status code	Reason phrase	Description
200	OK	Request completed.
409	Conflict	This is returned when a request is received to upload data that is out of order or is the wrong version of the batch.
416	Requested Range Not Satisfiable	This is returned when a request is received to upload data that is out of order or is the wrong version of the batch.

#### 3.5.5.1 Upload Data

The Upload Data resource facilitates the uploading of file data from client to server.

HTTP method	Description
PUT	Upload file data to server.

##### 3.5.5.1.1 PUT

This operation is transported by an HTTP PUT request.

The operation can be invoked through the following URI suffix on the session location URL constructed by the client as specified in section [3.6.5.1](#).

[/UploadData/](#)

The following is an example of a complete URI for this operation:

<https://contoso.com/Sync/1.0/Session/{60b8ca1d-59f3-4495-b299-f6cad89ada3b}/UploadData>

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
x-ecs-partnershipID	Required	The PartnershipID for the sync target share returned by the server in the GET response on the Share Discovery resource.

The response message for this operation can result in the following status codes.

Status code	Description
200	Request completed.

Status code	Description
409	This is returned when a request is received to upload data that is out of order or is the wrong version of the batch.

### 3.5.5.1.1.1 Request Body

The request body for this method contains the following:

Entry	Type
UploadEntryVector	VECTOR_UPLOAD_ENTRY

UploadEntryVector: A list of UPLOAD\_ENTRY structures, as specified in section [2.2.2.9](#), for each file being uploaded.

### 3.5.5.1.1.2 Response Body

The response body for this method contains the following:

Entry	Type
UploadResponseEntryVector	VECTOR_UPLOAD_RESPONSE_ENTRY

**UploadResponseEntryVector:** A list of UPLOAD\_RESPONSE\_ENTRY structures, as specified in section [2.2.2.10](#), indicating the server's response for each file being uploaded by the client.

### 3.5.5.1.1.3 Processing Details

The server MUST extract the session ID from the URI in an implementation-specific manner and verify whether a **Session** exists in a global **SessionList** with a matching session ID. If no session is found, the server MUST return a status code of 404.

## 3.5.5.2 Download Data

The Download Data resource facilitates downloading file data from server to client.

HTTP method	Description
PUT	Download file data from server.

### 3.5.5.2.1 PUT

This operation is transported by an HTTP PUT request.

The operation MUST be invoked through the following URI suffix on the session location URL constructed by the client as specified in section [3.6.5.2](#):

```
/DownloadData/
```

The following is an example of a complete URI for this operation:

https://contoso.com/Sync/1.0/Session/{60b8ca1d-59f3-4495-b299-f6cad89ada3b}/DownloadData

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
x-ecs-partnershipID	Required	The PartnershipID for the sync target share returned by the server in the GET response on Share Discovery resource.

The response message for this operation can result in the following status codes.

Status code	Description
200	Request completed.
416	This is returned when a request is received to download data that is out of order or is the wrong version of the batch.

### 3.5.5.2.1.1 Request Body

The request body for this method contains the following.

Entry	Type
DownloadEntryVector	VECTOR_DOWNLOAD_ENTRY

#### **DownloadEntryVector:**

A list of `DOWNLOAD_ENTRY` structures, as specified in section [2.2.2.11](#), for each file to be downloaded from the server.

### 3.5.5.2.1.2 Response Body

The response body for this method contains the following.

Entry	Type
DownloadResponseEntryVector	VECTOR_DOWNLOAD_RESPONSE_ENTRY

**DownloadResponseEntryVector:** A list of `DOWNLOAD_RESPONSE_ENTRY` structures, as specified in section [2.2.2.12](#), indicating the server's response for each file being downloaded by the client.

### 3.5.5.2.1.3 Processing Details

For each entry in `DownloadEntryVector`, the server **MUST** perform the following:

- Attempt to retrieve the file data in an implementation-specific manner.
- Construct the `DOWNLOAD_RESPONSE_ENTRY` structure.
- Set `SyncItemId` to the value received in the request.
- Set **Data** to empty if the server is unable to retrieve the file data. Otherwise, set **Data** to the retrieved data.

- Set **DataLength** to the size of **Data**.
- Set Result to an implementation-specific error code if the server is unable to retrieve the file data. Otherwise, set Result to zero.
- Set Hash to zero if the server is unable to retrieve the file data. Otherwise, the server MUST generate an identifier by computing MD5 hash of the retrieved data as specified in [\[RFC1321\]](#). Server MUST serialize the identifier in SYNC\_MD5HASH format as specified in section [2.2.2.15](#) and set Hash to this identifier.

The server MUST insert all the entries into DownloadResponseEntryVector in the response body. The server MUST set HTTP status code to 200. The server MUST send the response to the client.

### 3.5.6 Timer Events

None.

### 3.5.7 Other Local Events

None.

## 3.6 ECS Client Details

### 3.6.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in the Enterprise Client Synchronization Protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

#### 3.6.1.1 Global

**ClientID:** An identifier for the client.

**SyncClientVersion:** The highest version of the Enterprise Client Synchronization Protocol that the client supports. This MUST be one of the version values specified in section [1.7](#).

**GlobalSyncURI:** The URI to be used to enumerate all servers exposing a namespace for file synchronization.

**SyncTargetHostList:** A list of server URIs that expose the sync namespace.

**HostURLPrefix:** The URL prefix that the client currently uses to connect to for all sync operations.

**SyncTargetPartnershipId:** An identifier for the sync target share for a specific user.

**SyncServerDataTransferType:** The type of data transfer that the server supports for file uploading or downloading.

**SyncChangesURL:** The URL to be used by the client to poll for changes on the server.

**SessionLocationURL:** The URL to access the session on the server.

**UserDataUsed:** The amount of data, in bytes, used by the user in the sync target share.

**UserDataFree:** The amount of data, in bytes, available for the user in the sync target share.

**SyncBatchServerKnowledge:** An opaque serialized binary stream representing the version of the file data on the server.

**SyncBatchClientKnowledge:** An opaque serialized binary stream representing the version of the file data on the client.

**UploadFileList:** A list of UploadFile objects, as specified in section [3.6.1.2](#).

**DownloadFileList:** A list of DownloadFile objects, as specified in section [3.6.1.4](#).

**UploadBatchLimits:** The parameters that describe batch characteristics in an upload scenario. This MUST be in the format of the BATCH\_LIMITS structure as specified in section [2.2.2.4](#).

**DownloadBatchLimits:** The parameters that describe batch characteristics in a download scenario. This MUST be in the format of the BATCH\_LIMITS structure as specified in section [2.2.2.4](#).

**ContinueToken:** This contains the last x-ecs-continue token that was sent by the server.

### 3.6.1.2 Per UploadFile

**SyncId:** An identifier of the file that is used to correlate the change batch information from the underlying Sync Framework.

**DataUploadURL:** The URI to be used for performing data transfer for this file.

**CommitStatus:** The status, as an HRESULT value, of the commit operation after file data has been uploaded.

### 3.6.1.3 Per Share

**EnterpriseId:** An identifier that signifies the owning entity for all content synchronized with the share.

### 3.6.1.4 Per DownloadFile

**SyncId:** An identifier of the file, in the format specified in [\[MS-FSVCA\]](#) section [2.1](#), that is used to correlate the with the change batch information from the underlying Sync Framework.

**ConcurrencyInfo:** An opaque stream of bytes that identifies the file version, as specified in section [2.2.2.1](#).

## 3.6.2 Timers

None.

## 3.6.3 Initialization

When the ECS client is started, it MUST do the following:

The client MUST initialize **ClientID** with an administratively configured GUID.

The client MUST initialize **SyncClientVersion** in an implementation-specific<9> manner.

The client MUST initialize **GlobalSyncURI** based on local configuration.

The client MUST initialize **SyncTargetHostList** to empty.

The client MUST initialize **HostURLPrefix** to **GlobalSyncURI**.

The client MUST use secure HTTP as the transport.

The client MUST set **ContinueToken** to empty.

The client MUST then synchronize with the data on the server by performing the following steps:

#### **Server Discovery:**

The client MUST send a GET request on Discover Server resource, as specified in section [3.1.5.1](#), using **GlobalSyncURI** as the URI.

If the server responds with a status of 200, the client MUST set **SyncTargetHostList** to the ServerUrls entry that the server returned in the response body.

#### **Share Discovery:**

For each host URL Prefix in the **SyncTargetHostList**:

The client MUST attempt to send a GET request on Share Discovery resource, as specified in section [3.1.5.2.1](#), using the format "<host-url-prefix>/ Sync/**ClientSyncVersion**/Discover/Share".

If the server responds with a status of 200, the client MUST store the *PartnershipId* entry from the response body as **SyncTargetPartnershipId**. The client MUST set **HostURLPrefix** to the URL prefix of the server that returned a status code of 200 and skip the remaining entries in the **SyncTargetHostList** for share discovery. The **EnterpriseId** identifier from the response signifies the owning entity for all content synchronized with the share. The client SHOULD choose a local storage method that satisfies the requirements of the owning entity as dictated by its policies. These can include encryption, integrity, storage location, and the storage retention policy. The client MUST set **Share.EnterpriseId** to the **EnterpriseId** value that is received in the response.

If the server responds with a status code other than 200, the client MUST move to the next entry in the **SyncTargetHostList** and attempt the share discovery. The server MUST repeat this process until it gets a success response (200) or all the entries in the **SyncTargetHostList** are exhausted. If all the entries in **SyncTargetHostList** get exhausted before the client gets a success response, the client MUST wait for an implementation-specific period of time and retry the Server Discovery and Share Discovery steps mentioned earlier.

The client MUST set **SyncChangesURL** to "**HostURLPrefix/Sync/SyncClientVersion/Changes**".

#### **Server Capabilities:**

The client MUST send a GET request on the Server Capabilities resource, as specified in section [3.1.5.3.1](#), using the URI format "**HostURLPrefix/Sync/ClientSyncVersion/Capabilities**".

If the server responds with a status of 200, the client MUST store the entries in the response body as follows:

- SyncServerDataTransferType is set to ProtocolType in the response body

#### **Obtain User Configuration:**

The client MUST send a GET request on the User Configuration resource, as specified in section [3.3.5.1.1](#), by using the URI format "**HostURLPrefix/Sync/SyncClientVersion/UserConfiguration**".

If the server responds with a status code of 200, the client MUST store the entries in the response body as follows:

- *UserUsage* member of *QuotaUsage* entry is set to **UserDataUsed**.
- *UserDataFreeSpace* member of *QuotaUsage* entry is set to **UserDataFree**.

#### **Poll for Server Changes:**

The client MUST poll for the server changes by sending a HEAD request on the Detect Server Changes resource, as specified in section [3.2.5.1.1](#), using the URI format **SyncChangesURL**.

If the server responds with a status code of 200 and the x-ecs-changes-URL header is present, the client MUST set **SyncChangesURL** to the value in x-ecs-changes-URL header. The client MUST poll for the server changes by sending a HEAD request using the new **SyncChangesURL**.

If the server responds with a status code of 200 and the x-ecs-changes-URL header is not present, the client MUST initiate the file download process as outlined in section [3.6.5.2](#). Once the download process is completed, the client MUST continue to poll for the server changes by issuing a new request using the ETag returned by the server in the previous response as the *If-None-Match* header in the new request.

If the server responds with a status code of 304, the client SHOULD [<10>](#) wait for an implementation-specific period of time and reissue the same request.

If the server does not respond within an implementation-specific timeout, the client MUST reissue the same request.

### **3.6.4 Higher-Layer Triggering Events**

The following sections describe the operations performed by the ECS client in response to events triggered by higher-layer applications.

#### **3.6.4.1 Application Requests Uploading Data To Sync Target**

The application provides the following:

- An opaque binary stream representing the version of the files changed
- Size of the file data, in bytes, to be uploaded to the server

If the size of the file data to be uploaded is greater than the value of **UserDataFree**, the client MUST return an implementation-specific error to the calling application.

The client MUST perform the steps as specified in section [3.6.5.1](#) to upload the file data to the sync server.

### **3.6.5 Message Processing Events and Sequencing Rules**

The following sections describe the sequence of operations performed by the client in upload and download scenarios.

#### **3.6.5.1 Upload Scenario**

The client MUST do the following:

##### **Create Session:**

The client MUST send a PUT request on Create Session resource, as specified in section [3.4.5.1.1](#), by using the URI format "**HostURLPrefix/Sync/SyncClientVersion/Session**". The `x-ecs-partnershipId` header MUST be set to **SyncTargetPartnershipId**. The `x-ecs-devicename` header MUST be set to an implementation-specific string. The client MUST set *ClientID* in the request body to **ClientID**.

The calling application intends to:

- Commit SyncIds on the server since last sync, if Type is set to 0x01.
- Commit entire list of SyncId's available on the client, if Type is set to 0x03.

If the server responds with a status code of 200, the client MUST construct the SessionLocationURL using the `x-ecs-session-id` header as follows:

"**HostUrlPreifx/Sync/SyncClientVersion/Session/<x-ecs-session-id>/**"

#### **Get Sync Batch Parameters from Server:**

The client MUST send a GET request on the Sync Batch Parameters resource, as specified in section [3.4.5.2.1](#), by using the URI format "**SessionLocationURL/SyncBatchParameters**". The `x-ecs-partnershipId` header MUST be set to **SyncTargetPartnershipId**.

If the server responds with a status code of 200, the client MUST store the entries in the response body as follows:

- SyncBatchServerKnowledge is set to **SyncKnowledge**.
- UploadBatchLimits is set to **BatchLimits**.

#### **Inform Server to Prepare Batch:**

The client MUST send a PUT request on the Prepare Batch resource, as specified in section [3.4.5.3.1](#), by using the URI format "**SessionLocationURL/PrepareBatch/<BatchIndex>**". The BatchIndex in the URI MUST be set to a value that uniquely identifies the batch within the session. Using the **SyncBatchServerKnowledge** and the application-supplied version information, the client MUST construct a VECTOR\_FILE\_INFO\_INPUT\_ENTRY structure in an implementation-specific manner and insert the same as the FileVector entry in the request body.

If the server responds with a status code of 200, the client MUST store the information in FILE\_INFO\_ENTRY structures in FileList in the response body as follows:

- If the ProtocolType is 1, create a new **UploadFile** object and initialize it as follows:
  - **UploadFile.SyncId** MUST be set to *SyncItemId*.
- Insert the UploadFile object into the global **UploadFileList**.

#### **Data Transfer:**

From each entry in UploadFileList, the client MUST construct a VECTOR\_UPLOAD\_ENTRY structure and MUST send a PUT request on Upload Data resource, as specified in section [3.5.5.1.1](#), by using the URI format "SessionLocationURL/UploadData".

#### **Commit Change Batch:**

The client MUST send a PUT request on the Upload Batch resource, as specified in section [3.4.5.4.1](#), by using the URI format "**SessionLocationURL/UploadBatch/<BatchIndex>**". The BatchIndex in



the URI MUST be set to an implementation-specific value that identifies the batch to be uploaded. The SerializedSyncBatch in the request body MUST be set to an opaque binary stream representing the change batch. If **Session.SessionType** in the request is 0x03, SyncMetadata of SerializedSyncBatch in the request body contains the entire list of SyncIds on the client.

For each **UploadFile** in **UploadFileList**, the client MUST update **UploadFile.CommitStatus** to the Status entry returned in the FILE\_STATUS structure.

#### **Delete Session:**

The client MUST send a DELETE request on Delete Session resource, as specified in section [3.4.5.5.1](#), by using the URI format "**SessionLocationURL**".

The client MUST clear all entries in the **UploadFileList**.

### **3.6.5.2 Download Scenario**

The client MUST do the following:

#### **Create Session:**

The client MUST send a PUT request on the Create Session resource, as specified in section [3.4.5.1.1](#), by using the URI format "**HostURLPrefix/Sync/SyncClientVersion/Session**". The x-ecs-partnershipId header must be set to **SyncTargetPartnershipId**. The x-ecs-devicename header MUST be set to an implementation-specific string. The client MUST set *ClientID* in the request body to **ClientID**.

The calling application intends to

- Get SyncId's from the server since last sync, if Type is set to 0x02.
- Get entire list of SyncId's available on the server, if Type is set to 0x04.

If the server responds with a status code of 200, the client MUST construct the SessionLocationURL using the x-ecs-session-id header as follows:

"**HostURLPrefix/Sync/SyncClientVersion/Session/<x-ecs-session-id>/**"

#### **Update Server with Sync Batch Parameters:**

The client MUST obtain the information on the version of files in an implementation-specific manner and set it to **SyncBatchClientKnowledge**. The client MUST set **DownloadBatchLimits** to implementation-specific values.

The client MUST send a PUT request on the Sync Batch Parameters resource, as specified in section [3.4.5.2.2](#), by using the URI format "**SessionLocationURL/SyncBatchParameters**". The *SyncKnowledge* and *BatchLimits* in the request body MUST be set to **SyncBatchClientKnowledge** and **DownloadBatchLimits**, respectively.

#### **Obtain Change Batch details:**

The client performs the following steps to obtain server knowledge on the change batch:

1. Send a GET request on the Download Batch resource, as specified in section [3.4.5.6.1](#), by using the URI format "**SessionLocationURL/DownloadBatch**". If **ContinueToken** is not empty, the client MUST set x-ecs-continue in the request header to **ContinueToken**. Otherwise, if this is the first **DownloadBatch** request, the client MUST NOT include x-ecs-continue in the request header.

2. If the server does not respond with a status code of 200, or if the server does not provide a nonempty 'x-ecs-continue' token with the response, the change batch download fails and an implementation-defined local error is returned.
3. Otherwise, the client inserts the **DownloadFile** objects from the **BatchMetadata** entry in the **Download Batch** response body into the **DownloadFileList**. The client MUST set **ContinueToken** to the x-ecs-continue token that is received in the response header.
4. If the **IsLastChangeBatch** field in the SYNC\_CHANGE\_INFORMATION structure (specified in [\[MS-FSVCA\]](#) section 2.14) received in the **SyncMetadata** field of **BatchMetadata** in the Download Batch response is not set to 1, send a subsequent Get request by setting 'x-ecs-continue' entry in the request header, which is received in the response header of Download Batch response.
5. To obtain complete change batch details, the client can repeat steps 1 through 4 until the **IsLastChangeBatch** field is set to 1.
6. Change batch details are obtained.

If **Session.SessionType** in the request is 0x04, **SyncMetadata** of **BatchMetadata** in the response body contains the entire list of SyncIds on the server.

#### **Data Transfer:**

The client MUST send a PUT request on the Download Data resource, as specified in section [3.5.5.2.1](#). The DownloadEntryVector in the request body MUST be set to a VECTOR\_DOWNLOAD\_ENTRY constructed in an implementation-specific manner using each **DownloadFile** object in **DownloadFileList**.

If the server responds with a status code of 200, the client MUST verify the **Result** field of each DOWNLOAD\_RESPONSE\_ENTRY structure in the response body. If the **Result** is zero, the client MUST update the local file data corresponding to SyncItemId member of each DOWNLOAD\_RESPONSE\_ENTRY structure in the response body, and MUST remove each corresponding DownloadFile entry from DownloadFileList. The client MUST retry the data transfer request as specified above if the DownloadFileList is not empty.

The client MUST associate the **Share.EnterpriseId** value with the local file data in an implementation-defined manner.

#### **Delete Session:**

The client MUST send a DELETE request on the Delete Session resource, as specified in section [3.4.5.5.1](#), by using the URI format "**SessionLocationURL**".

### **3.6.6 Timer Events**

None.

### **3.6.7 Other Local Events**

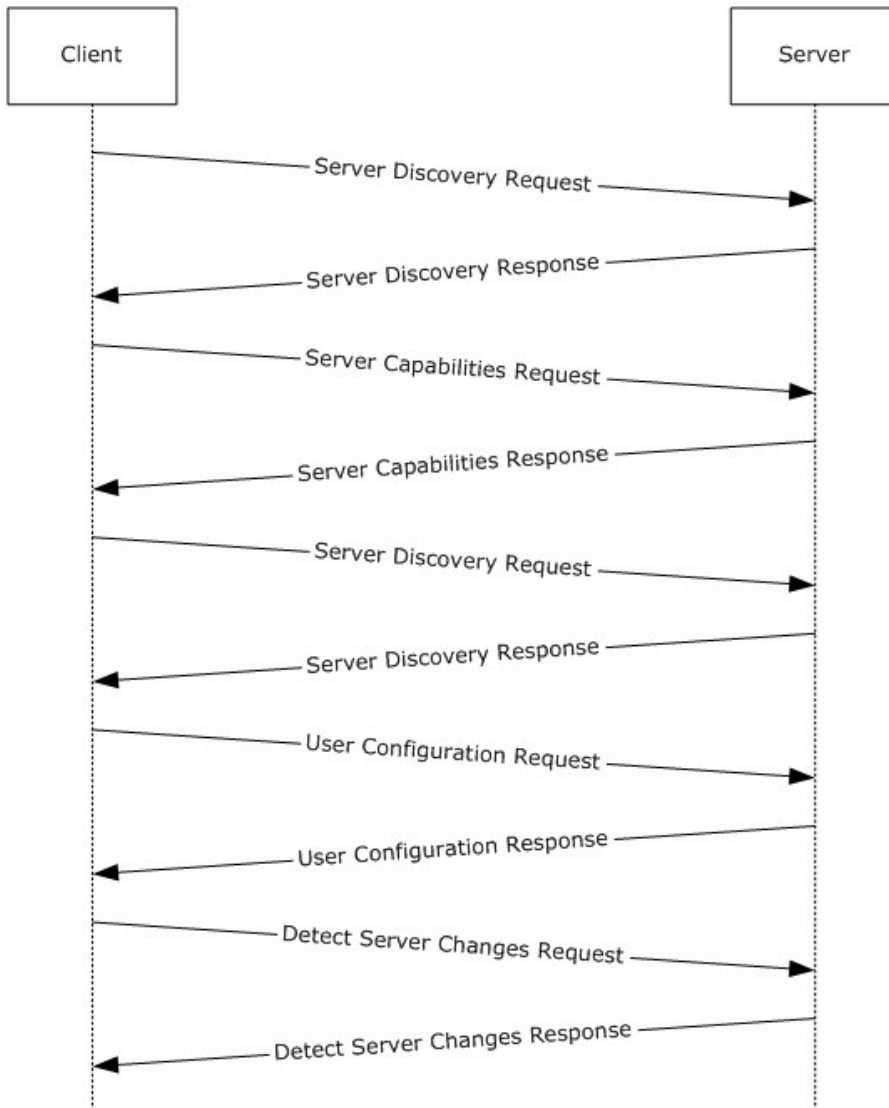
None.

## 4 Protocol Examples

The following section describes common scenarios that indicate normal traffic flow in order to illustrate the function of the Enterprise Client Synchronization (ECS) Protocol.

### 4.1 Query User Configuration Information and Detect Server Changes

The following diagram demonstrates the steps taken by the client to query user configuration information for a sync target share and to poll for URL changes on the server.



**Figure 1: Query user configuration information and detect server changes**

1. The client sends an HTTP GET request on the Server Discovery resource to enumerate a list of server URLs that expose the synchronization namespace.

```
Http: Request, GET /sync/1.0/discover/serverurl , Using GSS-API Authorization
Command: GET
+ URI: /sync/1.0/discover/serverurl
ProtocolVersion: HTTP/1.1
Cache-Control: no-cache
Connection: Keep-Alive
Pragma: no-cache
UserAgent: MS_WorkFoldersClient
x-ecs-devicename: HOMEPC,Windows,6,3,MS_WorkFoldersClient
ContentLength: 0
Host: syncsvr.contoso.com
+ Authorization: Negotiate
HeaderEnd: CRLF
```

2. The server sends a list of host server URL prefixes that the client can connect to for all subsequent sync operations.

```
- Http: Response, HTTP/1.1, Status: Not found, URL: /sync/1.0/discover/serverurl ,
Using GSS-API Authentication
ProtocolVersion: HTTP/1.1
StatusCode: 404, Not found
Reason: Not Found
Server: Microsoft-IIS/8.5
x-ecs-request-error: 0x80c80036
+ N2HTPersistentAuth:
+ WWWAuthenticate: Negotiate
oYG1MIgYoAMKAQChCwYJKoZIGvcSAQICooGdBIGaYIGXBgkqhkiG9xIBAgICAG+BhzCBhKADAgEFoQMCAQ+ieD
B2oAMCARKibwRt15fx9aLWbR8jTfGevFeDCNY13lpAgquJkxc7wRUXqkFo+DP0qPTP97qWyT1bbo7OpLTa879/
lFa9JrH3CMUp30WsD7aCh4UgxGjNH09MA9Cn94mruVkgRiWHC44ALH7/L3
Date: Tue, 24 Dec 2013 08:35:57 GMT
ContentLength: 0
HeaderEnd: CRLF
```

3. The client sends an HTTP GET request on the Server Capabilities resource to query protocol support information from the server.

```
- Http: Request, GET /sync/1.0/capabilities , Using GSS-API Authorization
Command: GET
+ URI: /sync/1.0/capabilities
ProtocolVersion: HTTP/1.1
Cache-Control: no-cache
Connection: Keep-Alive
Pragma: no-cache
UserAgent: MS_WorkFoldersClient
x-ecs-devicename: HOMEPC,Windows,6,3,MS_WorkFoldersClient
ContentLength: 0
Host: syncsvr.contoso.com
+ Authorization: Negotiate
HeaderEnd: CRLF
```

4. The server responds with status 200 to indicate that the version requested by the client is supported and to indicate the current server capabilities that it supports.

```
- Http: Response, HTTP/1.1, Status: Ok, URL: /sync/1.0/capabilities , Using GSS-API
Authentication
```

```
ProtocolVersion: HTTP/1.1
StatusCode: 200, Ok
Reason: OK
Server: Microsoft-IIS/8.5
+ N2HTPersistentAuth:
+ WWWAuthenticate: Negotiate ...
Date: Tue, 24 Dec 2013 08:36:08 GMT
ContentLength: 1
HeaderEnd: CRLF
ProtocolType FileBatching(1)
```

5. The client sends a HTTP GET request on Share Discovery resource to establish a sync partnership with the server and receive an ID that allows it to sync with the sync share selected by the server.

```
- Http: Request, GET /sync/1.0/discover/share , Using GSS-API Authorization
  Command: GET
+ URI: /sync/1.0/discover/share
  ProtocolVersion: HTTP/1.1
  Cache-Control: no-cache
  Connection: Keep-Alive
  Pragma: no-cache
  UserAgent: MS_WorkFoldersClient
  x-ecs-devicename: HOMEPC,Windows,6,3,MS_WorkFoldersClient
  x-ecs-share-type: User Data
  ContentLength: 0
  Host: syncsvr.contoso.com
+ Authorization: Negotiate
  HeaderEnd: CRLF
```

6. The server responds with PartnershipId, EnterpriseId, and DataSize.

```
- Http: Response, HTTP/1.1, Status: Ok, URL: /sync/1.0/discover/share , Using GSS-API
Authentication
  ProtocolVersion: HTTP/1.1
  StatusCode: 200, Ok
  Reason: OK
  Server: Microsoft-IIS/8.5
+ N2HTPersistentAuth:
+ WWWAuthenticate: Negotiate ...
  Date: Tue, 24 Dec 2013 08:35:57 GMT
  ContentLength: 103
  HeaderEnd: CRLF
  PartnershipId SalesShare\Administrator\Documents|S-1-5-21-2507039251-2369267751-
3911940733- 500
  Length 80
  Content SalesShare\Administrator\Documents|S-1-5-21-2507039251-2369267751-
3911940733-500
  EnterpriseId contoso.com
  Length 11
  Content contoso.com
  DataSize 0
```

7. The client sends an HTTP GET request on the User Configuration resource to query configuration information for a sync target share.

```
- Http: Request, GET /sync/1.0/configuration
  Command: GET
+ URI: /sync/1.0/configuration
  ProtocolVersion: HTTP/1.1
  Cache-Control: no-cache
  Connection: Keep-Alive
  Pragma: no-cache
  UserAgent: MS_WorkFoldersClient
  x-ecs-devicename: HOMEPC,Windows,6,3,MS_WorkFoldersClient
  x-ecs-domain: Y29udG9zby5jb20=
  x-ecs-partnershipId:
U2FsZXNTaGFyZXXxBZG1pbmlzdHJhdG9yXERvY3VtZW50c3xTLTEtNS0yMS0yNTA3MMDM5MjUxLTIzNjkyNjc3NT
EtMzkxMTk0MDczMy01MDA=
  ContentLength: 0
  Host: syncsvr.contoso.com
  HeaderEnd: CRLF
```

## 8. The server responds with QuotaUsage, PolicyList, Length, and AdminInfo.

```
- Http: Response, HTTP/1.1, Status: Ok, URL: /sync/1.0/configuration
  ProtocolVersion: HTTP/1.1
  StatusCode: 200, Ok
  Reason: OK
  Server: Microsoft-IIS/8.5
+ N2HTPersistentAuth:
  Date: Tue, 24 Dec 2013 08:35:57 GMT
  ContentLength: 28
  HeaderEnd: CRLF
- payload: HttpContentType = NetmonNull
  HTTPPayloadLine:
  QuotaUsage
  UserUsage 18053250867527680
  UserDataFreeSpace 18446744073709551615
  PolicyList
  NumEntries 3
  EntryStream
  [0]
  PolicyName Password(2)
  PolicyType 0
  [1]
  PolicyName AutoLock(3)
  PolicyType 0
  [2]
  PolicyName Encryption(1)
  PolicyType 0
  AdminEmail ...
```

## 9. The client sends a HTTP GET request on the Detect Server Changes resource to poll for the changes on the server.

```
- Http: Request, HEAD /sync/1.0/changes
  Command: HEAD
+ URI: /sync/1.0/changes
  ProtocolVersion: HTTP/1.1
  Cache-Control: no-cache
  Connection: Keep-Alive
```

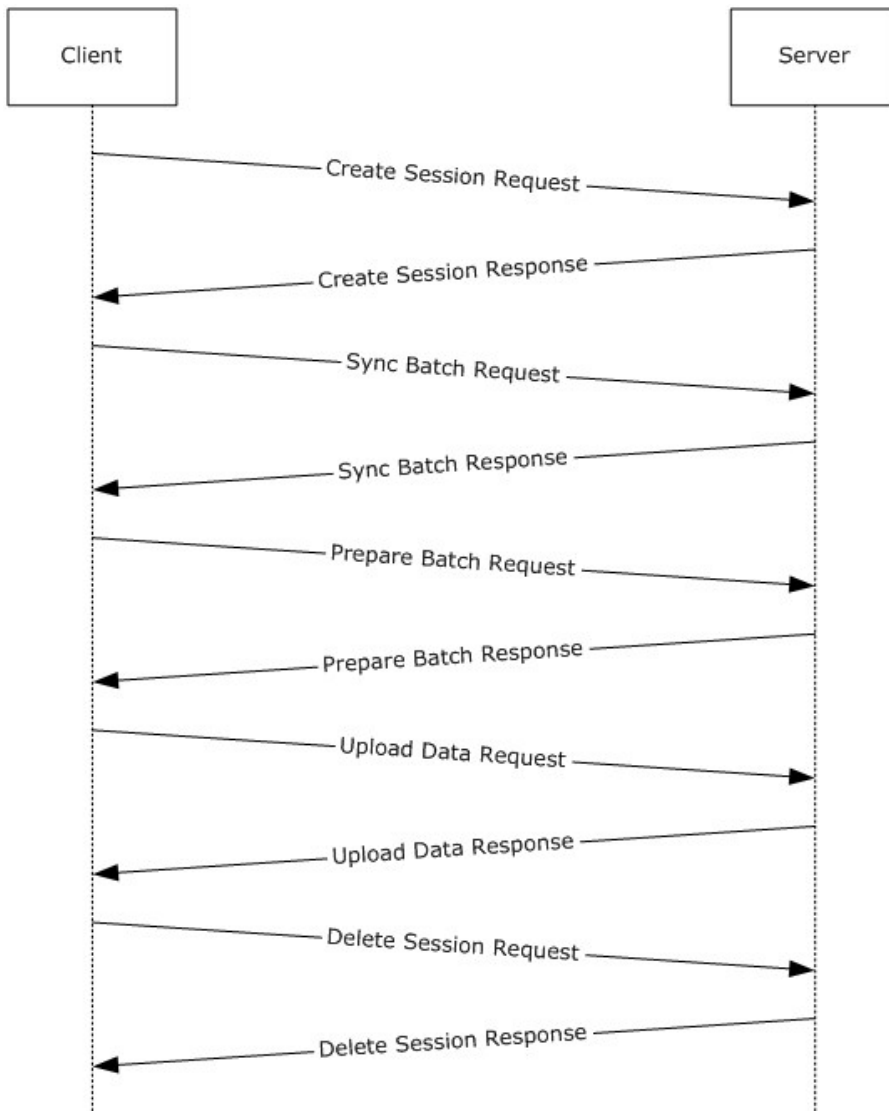
```
Pragma: no-cache
UserAgent: MS_WorkFoldersClient
x-ecs-devicename: HOMEPC,Windows,6,3,MS_WorkFoldersClient
x-ecs-partnershipId:
U2FsZXXNTaGFyZXxBZG1pbmlzdHJhdG9yXERvY3VtZW50c3xTLTEtNS0yMS0yNTA3MMDM5MjUxLTIzNjkyNjc3NT
EtMzkxMTk0MDczMy01MDA=
ContentLength: 0
Host: syncsvr.contoso.com
HeaderEnd: CRLF
```

10.The server responds with a new URI in the x-ecs-changes-URL response header.

```
- Http: Response, HTTP/1.1, Status: Ok, URL: /sync/1.0/changes
  ProtocolVersion: HTTP/1.1
  StatusCode: 200, Ok
  Reason: OK
  ContentLength: 0
  ETag: 0
  Server: Microsoft-IIS/8.5
+ N2HTPersistentAuth:
  Date: Tue, 24 Dec 2013 08:36:08 GMT
  HeaderEnd: CRLF
```

## 4.2 Upload Scenario

The following diagram demonstrates the steps taken to upload the data between client and server.



**Figure 2: Upload scenario**

1. The client sends a HTTP PUT request on Create Session resource to start a sync session on the server.

```

- Http: Request, PUT /sync/1.0/session
  Command: PUT
+ URI: /sync/1.0/session
  ProtocolVersion: HTTP/1.1
  Cache-Control: no-cache
  Connection: Keep-Alive
  Pragma: no-cache
  UserAgent: MS_WorkFoldersClient
  x-ecs-devicename: HOMEPC,Windows,6,3,MS_WorkFoldersClient
  
```



```
x-ecs-partnershipId:
U2FsZXRhZGFyZSxkZG1pbmlzdHJhdG9yXERvY3VtZW50c3xTLTEtNS0yMS0yNTA3MMDM5MjUxLTIzNjkyNjc3NT
EtMzkxMTk0MDczMy01MDA=
ContentLength: 17
Host: syncsvr.contoso.com
Type Upload(1)
ReplicaId 60f4c9fd-9c9d-421c-97a0-bdcb740424c3
```

## 2. The server creates a new session and responds with session information.

```
- Http: Response, HTTP/1.1, Status: Created, URL: /sync/1.0/session
ProtocolVersion: HTTP/1.1
StatusCode: 201, Created
Reason: Created
Server: Microsoft-IIS/8.5
x-ecs-session-id: {01FF313D-1BE3-49C9-B7BA-1D668531E1EF}
x-ecs-metadata-version: {95B11081-8EBA-411D-AA5B-467AB44AEFF0}
+ N2HTPersistentAuth:
Date: Tue, 24 Dec 2013 08:36:08 GMT
ContentLength: 0
HeaderEnd: CRLF
```

## 3. The client sends a HTTP GET request on the Sync Batch Parameters resource, to obtain the server's sync knowledge in an upload scenario.

```
- Http: Request, GET /sync/1.0/session/%7B01ff313d-1be3-49c9-b7ba-
1d668531e1ef%7D/syncbatchparameters
Command: GET
+ URI: /sync/1.0/session/%7B01ff313d-1be3-49c9-b7ba-
1d668531e1ef%7D/syncbatchparameters
ProtocolVersion: HTTP/1.1
Cache-Control: no-cache
Connection: Keep-Alive
Pragma: no-cache
UserAgent: MS_WorkFoldersClient
x-ecs-devicename: HOMEPC,Windows,6,3,MS_WorkFoldersClient
x-ecs-partnershipId:
U2FsZXRhZGFyZSxkZG1pbmlzdHJhdG9yXERvY3VtZW50c3xTLTEtNS0yMS0yNTA3MMDM5MjUxLTIzNjkyNjc3NT
EtMzkxMTk0MDczMy01MDA=
ContentLength: 0
Host: syncsvr.contoso.com
HeaderEnd: CRLF
```

## 4. The server responds with SyncKnowledge and BatchLimits.

```
Frame: Number = 122, Captured Frame Length = 398, MediaType = ETHERNET
+ Ethernet: Etype = Internet IP (IPv4),DestinationAddress:[00-15-5D-8E-71-
8B],SourceAddress:[00-15-5D-8E-71-89]
+ Ipv4: Src = 10.10.1.12, Dest = 10.10.1.16, Next Protocol = TCP, Packet ID = 32727,
Total IP Length = 384
+ Tcp: Flags=...AP..., SrcPort=HTTP(80), DstPort=63384, PayloadLen=344, Seq=1544306057
- 1544306401, Ack=2864974975, Win=513 (scale factor 0x8) = 131328
- Http: Response, HTTP/1.1, Status: Ok, URL: /sync/1.0/session/%7B01ff313d-1be3-49c9-
b7ba-1d668531e1ef%7D/syncbatchparameters
ProtocolVersion: HTTP/1.1
StatusCode: 200, Ok
```

```

Reason: OK
Server: Microsoft-IIS/8.5
+ N2HTPersistentAuth:
Date: Tue, 24 Dec 2013 08:36:08 GMT
ContentLength: 217
HeaderEnd: CRLF
SyncKnowledge ...
BlobSize 205
BlobData ...
BatchLimits ...
MaxFileDataSize 200
MaxFileCount 1000

```

5. The client sends an HTTP PUT request on the Prepare Batch resource, to request server-side preparation for a file upload in a sync process.

```

- Http: Request, PUT /sync/1.0/session/%7B01ff313d-1be3-49c9-b7ba-1d668531e1ef%7D/preparebatch/0
Command: PUT
+ URI: /sync/1.0/session/%7B01ff313d-1be3-49c9-b7ba-1d668531e1ef%7D/preparebatch/0
ProtocolVersion: HTTP/1.1
Cache-Control: no-cache
Connection: Keep-Alive
Pragma: no-cache
UserAgent: MS_WorkFoldersClient
x-ecs-devicename: HOMEPC,Windows,6,3,MS_WorkFoldersClient
x-ecs-partnershipId:
U2FsZXNTaGFyZlZlZG9yXERvY3VtZW50c3xTLTEtNS0yMS0yNTA3MMDM5MjUxLTIzNjkyNjc3NT
EtMzkxMTk0MDczMy0lMDA=
ContentLength: 4
Host: syncsvr.contoso.com
PrepareBatchRequestBody
FileVector ...
NumEntries 0
EntryStreamBytes []

```

6. The server responds with list of file status entries indicating the status of the change batch preparation.

```

- Http: Response, HTTP/1.1, Status: Ok, URL: /sync/1.0/session/%7B01ff313d-1be3-49c9-b7ba-1d668531e1ef%7D/preparebatch/0
ProtocolVersion: HTTP/1.1
StatusCode: 200, Ok
Reason: OK
Server: Microsoft-IIS/8.5
+ N2HTPersistentAuth:
Date: Tue, 24 Dec 2013 08:36:08 GMT
ContentLength: 4
HeaderEnd: CRLF
FileList VECTOR_FILE_INFO_ENTRY{NumEntries=0,EntryStream=[]}

```

7. For each entry client constructs a VECTOR\_UPLOAD\_ENTRY structure and MUST send a PUT request on Upload Data resource for uploading the file data from client to server.
8. The server responds with a list of UPLOAD\_RESPONSE\_ENTRY structures, indicating the server's response for each file being uploaded by the client.

9. The client sends a HTTP PUT request on the Upload Batch resource, to commit a change batch on the server.

```
- Http: Request, PUT /sync/1.0/session/%7B01ff313d-1be3-49c9-b7ba-1d668531e1ef%7D/uploadbatch/0
  Command: PUT
+ URI: /sync/1.0/session/%7B01ff313d-1be3-49c9-b7ba-1d668531e1ef%7D/uploadbatch/0
  ProtocolVersion: HTTP/1.1
  Cache-Control: no-cache
  Connection: Keep-Alive
  Pragma: no-cache
  UserAgent: MS_WorkFoldersClient
  x-ecs-devicename: HOMEPC,Windows,6,3,MS_WorkFoldersClient
  x-ecs-partnershipId:
  U2FsZXNTaGFyZXXxBZG1pbmlzdHJhdG9yXERvY3VtZW50c3xTLTEtNS0yMS0yNTA3MDM5MjUxLTIzNjkyNjc3NT
  EtMzkxMTk0MDczMy01MDA=
  ContentLength: 735
  Host: syncsvr.contoso.com
  SerializedSyncBatch ...
  Files VECTOR_FILE_METADATA_ENTRY{NumEntries=0,EntryStreamBytes=[]}
  DeviceNames VECTOR_ECSSTRING{NumEntries=0,EntryStreamBytes=[]}
  SyncMetadata ...
```

10. The server responds with a list of file status entries indicating the status of the change batch commit operation.

```
- Http: Response, HTTP/1.1, Status: Ok, URL: /sync/1.0/session/%7B01ff313d-1be3-49c9-b7ba-1d668531e1ef%7D/uploadbatch/0
  ProtocolVersion: HTTP/1.1
  StatusCode: 200, Ok
  Reason: OK
  Server: Microsoft-IIS/8.5
+ N2HTPersistentAuth:
  Date: Tue, 24 Dec 2013 08:36:08 GMT
  ContentLength: 4
  HeaderEnd: CRLF
  FileStatusList VECTOR_FILE_STATUS_ENTRY{NumEntries=0,EntryStream=[]}
```

11. The client sends a HTTP DELETE request on the Delete Session resource, to remove a server session after a synchronization process is completed or aborted.

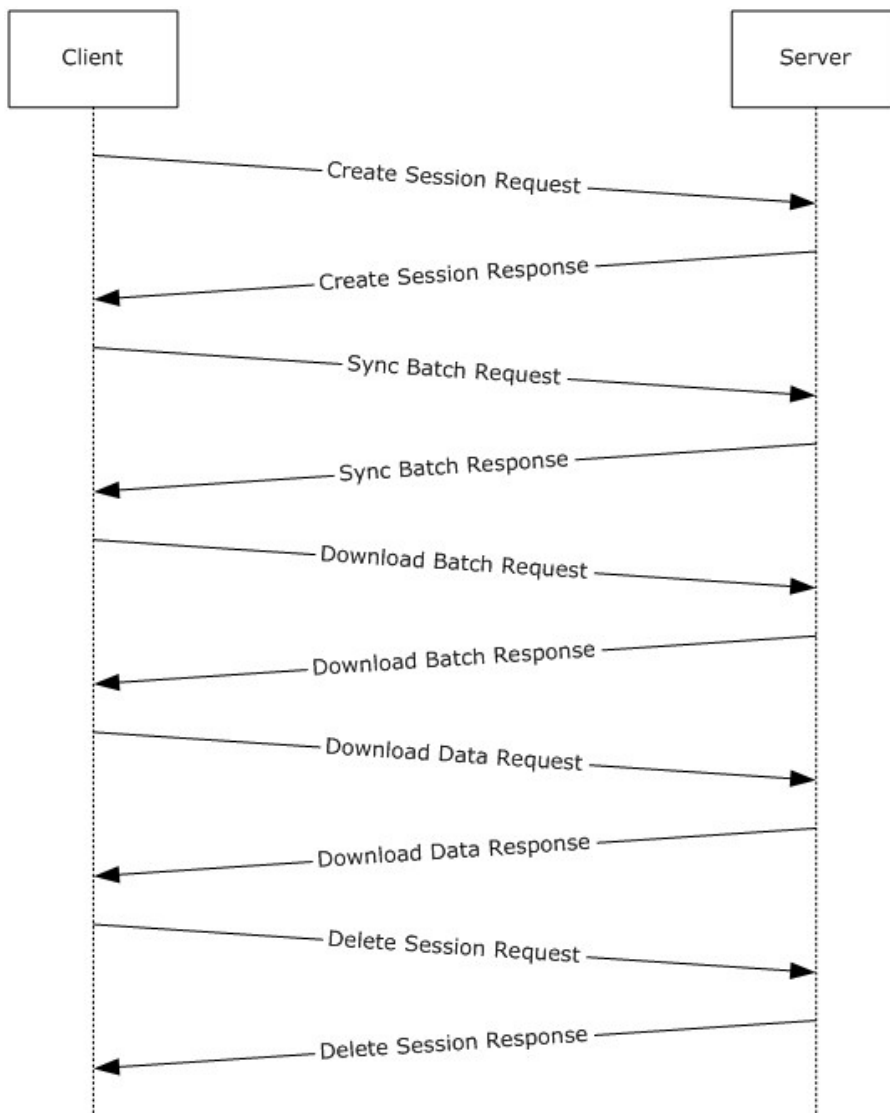
```
- Http: Request, DELETE /sync/1.0/session/%7B01ff313d-1be3-49c9-b7ba-1d668531e1ef%7D
  Command: DELETE
+ URI: /sync/1.0/session/%7B01ff313d-1be3-49c9-b7ba-1d668531e1ef%7D
  ProtocolVersion: HTTP/1.1
  Cache-Control: no-cache
  Connection: Keep-Alive
  Pragma: no-cache
  UserAgent: MS_WorkFoldersClient
  x-ecs-devicename: HOMEPC,Windows,6,3,MS_WorkFoldersClient
  x-ecs-partnershipId:
  U2FsZXNTaGFyZXXxBZG1pbmlzdHJhdG9yXERvY3VtZW50c3xTLTEtNS0yMS0yNTA3MDM5MjUxLTIzNjkyNjc3NT
  EtMzkxMTk0MDczMy01MDA=
  ContentLength: 0
  Host: syncsvr.contoso.com
  HeaderEnd: CRLF
```

## 12.The server deletes the session.

```
- Http: Response, HTTP/1.1, Status: Accepted, URL: /sync/1.0/session/%7B01ff313d-1be3-49c9-b7ba-1d668531e1ef%7D
  ProtocolVersion: HTTP/1.1
  StatusCode: 200, Accepted
  Reason: Accepted
  Server: Microsoft-IIS/8.5
+ N2HTPersistentAuth:
  Date: Tue, 24 Dec 2013 08:36:08 GMT
  ContentLength: 0
  HeaderEnd: CRLF
```

### 4.3 Download Scenario

The following diagram demonstrates the steps taken to download the data between client and server.



**Figure 3: Download scenario**

1. The client sends a HTTP PUT request on Create Session resource to start a sync session on the server.
2. The server creates a new session and responds with session information.
3. The client sends a HTTP PUT request on the Sync Batch Parameters resource, to upload its sync knowledge which will be used by the server to generate the list of changes to be downloaded".
4. The server responds with TotalFileCount and TotalFileSize.
5. The client sends a HTTP GET request on the Download Batch resource, to obtain metadata information for all files in a change batch.
6. The server responds with BatchMetadata and DownloadInfo.

7. The client sends a HTTP PUT request on the Download Data resource the DownloadEntryVector in the request body MUST be set to a VECTOR\_DOWNLOAD\_ENTRY.
8. For each file represented by a DOWNLOAD\_ENTRY structure in DownloadEntryVector, the server constructs the DOWNLOAD\_RESPONSE\_ENTRY and adds it to VECTOR\_DOWNLOAD\_RESPONSE\_ENTRY.
9. The client sends a HTTP DELETE request on Delete Session resource, to remove a server session after a synchronization process is completed or aborted.
10. The server deletes the session.

## 5 Security

### 5.1 Security Considerations for Implementers

The Enterprise Client Synchronization Protocol requires that all the requests from the client be authenticated. The client is expected to use an implementation-dependent authentication mechanism to obtain a security token and include that token in the standard HTTP Authorization header. The server will validate the token and use it to authorize the request.

### 5.2 Index of Security Parameters

None.

## 6 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:

- Windows 8.1 operating system
- Windows Server 2012 R2 operating system

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.

<1> [Section 1.7](#): The following table illustrates the Windows operating system versions that support the ECS client and server.

ECS client	ECS Server
Windows 8.1	Windows Server 2012 R2

<2> [Section 2.2.2.5](#): Windows uses the File Attribute Constants as defined in [\[MSDN-FSA\]](#).

<3> [Section 2.2.2.9](#): Windows clients will set the **Reserved** field to an arbitrary value

<4> [Section 3.2.5.1.1.3](#): Windows servers do not send this header.

<5> [Section 3.4.5](#): Windows Server 2012 R2 limits outstanding requests by a user to 25.

<6> [Section 3.4.5.1.1.3](#): Windows Server 2012 R2 limits number of download sessions established by a user to 20 and number of upload sessions to 1 per user.

<7> [Section 3.4.5.2.1.3](#): Windows-based servers will set **MaxFileDataSize** to a maximum of 200 MiB and **MaxFileCount** to a maximum of 1000.

<8> [Section 3.4.5.2.2.3](#): Windows-based server will set **TotalFileCount** to a maximum of 1000 and **TotalFileSize** to a maximum of 200 MiB in one batch.

<9> [Section 3.6.3](#): Windows clients set this to "1.0".

<10> [Section 3.6.3](#): Windows 8.1 client uses a default wait time of 10 minutes.



## 7 Change Tracking

This section identifies changes that were made to the [MS-ECS] protocol document between the February 2014 and May 2014 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.
- The removal of a document from the documentation set.

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 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 changes were introduced. Minor editorial and formatting changes may have been made, but the technical content of the document is identical to the last released version.

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.

- 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 [dochelp@microsoft.com](mailto:dochelp@microsoft.com).

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
<a href="#">1.3 Overview</a>	Removed the product behavior note for Sync Framework dependency.	N	Product behavior note removed.
<a href="#">2.2.1 HTTP Headers</a>	71037 Removed x-ecs-change-detection and its description from the list of HTTP headers.	Y	Content updated.
<a href="#">2.2.1.2 x-ecs-devicename</a>	71028 Updated the string format and example; changed SyncAppVersion to UserAgent Name.	Y	Content updated.
<a href="#">2.2.2 Common Data Structures</a>	71076 Updated the table of common data structures.	Y	Content updated.
<a href="#">2.2.2 Common Data Structures</a>	70966 Clarified the byte order to be used to transmit multiple-byte fields and the type and byte length of numeric fields.	Y	Content updated.
<a href="#">2.2.2 Common Data Structures</a>	71029 Changed the description of the VECTOR_STRING structure and added the ECS_STRING structure.	Y	Content updated.
<a href="#">2.2.2.1 SYNC_BLOB</a>	70966 Changed the representation of the structure to a bit table and modified the field descriptions.	Y	Content updated.
<a href="#">2.2.2.2 QUOTA_USAGE_ENTRY</a>	70966 Changed the representation of the structure to a bit table and modified the field descriptions.	Y	Content updated.

<b>Section</b>	<b>Tracking number (if applicable) and description</b>	<b>Major change (Y or N)</b>	<b>Change type</b>
<a href="#">2.2.2.3 POLICY_ENTRY</a>	71015 Clarified the description of the PolicyType member of the POLICY_ENTRY structure.	Y	Content updated.
<a href="#">2.2.2.3 POLICY_ENTRY</a>	70966 Changed the representation of the structure to a bit table and modified the field descriptions.	Y	Content updated.
<a href="#">2.2.2.4 BATCH_LIMITS_ENTRY</a>	70966 Changed the representation of the structure to a bit table and modified the field descriptions.	Y	Content updated.
<a href="#">2.2.2.5 FILE_METADATA_ENTRY</a>	71076 Updated the description of the FileID and SyncVersion fields	Y	Content updated.
<a href="#">2.2.2.5 FILE_METADATA_ENTRY</a>	71079 Updated the description of the ParentId field.	Y	Content updated.
<a href="#">2.2.2.5 FILE_METADATA_ENTRY</a>	70966 Changed the representation of the structure to a bit table and modified the field descriptions.	Y	Content updated.
<a href="#">2.2.2.5 FILE_METADATA_ENTRY</a>	71029 Changed the description of the Name field to indicate that it is a variably sized field containing an ECS_STRING structure.	Y	Content updated.
<a href="#">2.2.2.6 FILE_INFO_INPUT_ENTRY</a>	71016 Added Length and SyncItemId fields.	Y	Content updated.
<a href="#">2.2.2.6 FILE_INFO_INPUT_ENTRY</a>	70966 Changed the representation of the structure to a bit table and modified the field descriptions.	Y	Content updated.
<a href="#">2.2.2.6 FILE_INFO_INPUT_ENTRY</a>	71029 Removed the Length field; changed FileExtension to a variably sized field containing an ECS_STRING structure.	Y	Content updated.
<a href="#">2.2.2.7 FILE_INFO_ENTRY</a>	71029 Added the Length member to the structure and amended the type of the Uri[] array.	Y	Content updated.

<b>Section</b>	<b>Tracking number (if applicable) and description</b>	<b>Major change (Y or N)</b>	<b>Change type</b>
<a href="#">2.2.2.7 FILE_INFO_ENTRY</a>	70966 Changed the representation of the structure to a bit table and modified the field descriptions.	Y	Content updated.
<a href="#">2.2.2.7 FILE_INFO_ENTRY</a>	71029 Removed the Length field; changed the description of the Uri field to indicate that it contains an ECS_STRING structure.	Y	Content updated.
<a href="#">2.2.2.7 FILE_INFO_ENTRY</a>	71044 Updated the description of the ProtocolType and Prepare Results fields.	Y	Content updated.
<a href="#">2.2.2.8 FILE_STATUS_ENTRY</a>	70966 Changed the representation of the structure to a bit table and modified the field descriptions.	Y	Content updated.
<a href="#">2.2.2.9 UPLOAD_ENTRY</a>	71030 Removed FileHash field and added FileSize field and related description. Changed UploadData data type to BYTE[]. Updated Length and Reserved descriptions.	Y	Content updated.
<a href="#">2.2.2.9 UPLOAD_ENTRY</a>	71046 Changed Reserved field description to: "The client MUST set this field to 0, and the server MUST ignore it on receipt."	Y	Content updated.
<a href="#">2.2.2.9 UPLOAD_ENTRY</a>	70966 Changed the representation of the structure to a bit table and modified the field descriptions.	Y	Content updated.
<a href="#">2.2.2.10 UPLOAD_RESPONSE_ENTRY</a>	71030 Added HttpStatus and Hash fields and associated descriptions.	Y	New content added.
<a href="#">2.2.2.10 UPLOAD_RESPONSE_ENTRY</a>	70966 Changed the representation of the structure to a bit table and modified the field descriptions.	Y	Content updated.
<a href="#">2.2.2.11 DOWNLOAD_ENTRY</a>	70966 Changed the representation of the structure to a bit table and modified the field descriptions.	Y	Content updated.
<a href="#">2.2.2.11</a>	71047 Updated the description of the	Y	Content

<b>Section</b>	<b>Tracking number (if applicable) and description</b>	<b>Major change (Y or N)</b>	<b>Change type</b>
<a href="#">DOWNLOAD ENTRY</a>	FileVersion field.		updated.
<a href="#">2.2.2.12 DOWNLOAD RESPONSE ENTRY</a>	71031 Changed Data field to Byte[] type and updated DataLength description.	Y	Content updated.
<a href="#">2.2.2.12 DOWNLOAD RESPONSE ENTRY</a>	70966 Changed the representation of the structure to a bit table and modified the field descriptions.	Y	Content updated.
<a href="#">2.2.2.13 FILE DOWNLOAD INFO ENTRY</a>	71029 Removed the Length field; changed the description of the Uri field to indicate that it contains an ECS_STRING structure.	Y	Content updated.
<a href="#">2.2.2.13 FILE DOWNLOAD INFO ENTRY</a>	70966 Changed the representation of the structure to a bit table and modified the field descriptions.	Y	Content updated.
<a href="#">2.2.2.14 SYNC CHANGE BATCH</a>	70969 Removed product behavior note from the description of the SyncMetadata member of the SYNC_CHANGE_BATCH structure.	Y	Product behavior note removed.
<a href="#">2.2.2.14 SYNC CHANGE BATCH</a>	71029 Updated the underlying type of the DeviceNames member and description to indicate that it is an array.	Y	Content updated.
<a href="#">2.2.2.14 SYNC CHANGE BATCH</a>	70966 Changed the representation of the structure to a bit table and modified the field descriptions.	Y	Content updated.
<a href="#">2.2.2.15 SYNC MD5HASH</a>	70966 Changed the representation of the structure to a bit table and modified the field descriptions.	Y	Content updated.
<a href="#">2.2.2.16 VECTOR POLICY ENTRY</a>	70966 Changed the representation of the structure to a bit table and modified the field descriptions.	Y	Content updated.
<a href="#">2.2.2.17 VECTOR FILE METADATA ENTRY</a>	70966 Changed the representation of the structure to a bit table and modified the field descriptions.	Y	Content updated.

<b>Section</b>	<b>Tracking number (if applicable) and description</b>	<b>Major change (Y or N)</b>	<b>Change type</b>
<a href="#">2.2.2.18</a> <a href="#">VECTOR FILE INFO INPUT ENTRY</a>	70966 Changed the representation of the structure to a bit table and modified the field descriptions.	Y	Content updated.
<a href="#">2.2.2.19</a> <a href="#">VECTOR FILE INFO ENTRY</a>	70966 Changed the representation of the structure to a bit table and modified the field descriptions.	Y	Content updated.
<a href="#">2.2.2.20</a> <a href="#">VECTOR FILE STATUS ENTRY</a>	70966 Changed the representation of the structure to a bit table and modified the field descriptions.	Y	Content updated.
<a href="#">2.2.2.21</a> <a href="#">VECTOR UPLOAD ENTRY</a>	70966 Changed the representation of the structure to a bit table and modified the field descriptions.	Y	Content updated.
<a href="#">2.2.2.22</a> <a href="#">VECTOR UPLOAD RESPONSE ENTRY</a>	70966 Changed the representation of the structure to a bit table and modified the field descriptions.	Y	Content updated.
<a href="#">2.2.2.23</a> <a href="#">VECTOR DOWNLOAD ENTRY</a>	70966 Changed the representation of the structure to a bit table and modified the field descriptions.	Y	Content updated.
<a href="#">2.2.2.24</a> <a href="#">VECTOR DOWNLOAD RESPONSE ENTRY</a>	71031 Removed ULONG NumEntries typedef struct member and related description.	Y	Content updated.
<a href="#">2.2.2.24</a> <a href="#">VECTOR DOWNLOAD RESPONSE ENTRY</a>	70966 Changed the representation of the structure to a bit table and modified the field descriptions.	Y	Content updated.
<a href="#">2.2.2.25</a> <a href="#">VECTOR FILE DOWNLOAD INFO ENTRY</a>	70966 Changed the representation of the structure to a bit table and modified the field descriptions.	Y	Content updated.
<a href="#">2.2.2.26</a> <a href="#">VECTOR STRING</a>	71029 Added a bit table representation of the structure; changed the descriptions of the included fields: NumEntries and EntryStreamBytes.	Y	Content updated.
<a href="#">2.2.2.26</a> <a href="#">VECTOR STRING</a>	70966 Changed the representation of the structure to a bit table and	Y	Content updated.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
	modified the field descriptions.		
<a href="#">2.2.2.27</a> <a href="#">ECS_STRING</a>	71029 Added section.	Y	New content added.
<a href="#">3</a> <a href="#">Protocol Details</a>	70966 Added information on the order of transmission of the "Request body" and "Response body" entries.	Y	Content updated.
<a href="#">3.1.1</a> <a href="#">Abstract Data Model</a>	Removed the ADM element SupportedVersions.	Y	Content updated.
<a href="#">3.1.1</a> <a href="#">Abstract Data Model</a>	71045 Added ClientRequestsCountTable and Per ClientRequestsCount elements.	Y	Content updated.
<a href="#">3.1.3</a> <a href="#">Initialization</a>	Removed the initialization rules for SupportedVersions.	Y	Content updated.
<a href="#">3.1.3</a> <a href="#">Initialization</a>	71045 Added initialization rules for ClientRequestsCountTable.	Y	Content updated.
<a href="#">3.1.5.1.1.2</a> <a href="#">Response Body</a>	71154 Clarified that the method response contains the listed entries.	Y	Content updated.
<a href="#">3.1.5.2.1</a> <a href="#">GET</a>	71054 Updated the valid value of the x-ecs-share-type header.	Y	Content updated.
<a href="#">3.1.5.2.1.2</a> <a href="#">Response Body</a>	70971 Changed the description of the EnterpriseId entry to clarify that it represents the owning entity of the content.	Y	Content updated.
<a href="#">3.1.5.2.1.2</a> <a href="#">Response Body</a>	70988 Changed the type of the PartnershipId and EnterpriseId entries to VECTOR_STRING and amended the descriptions of the entries accordingly.	Y	Content updated.
<a href="#">3.1.5.2.1.2</a> <a href="#">Response Body</a>	71154 Clarified that the method response contains the listed entries.	Y	Content updated.
<a href="#">3.1.5.2.1.2</a> <a href="#">Response Body</a>	71029 Changed the type of the PartnershipId and EnterpriseId entries to ECS_STRING; changed	Y	Content updated.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
	the description of the EnterpriseId field.		
<a href="#">3.1.5.2.1.3 Processing Details</a>	70971 Added server instructions on setting the EnterpriseId value.	Y	Content updated.
<a href="#">3.1.5.2.1.3 Processing Details</a>	71045 Updated processing details related to the ClientRequestsCountTable.	Y	Content updated.
<a href="#">3.1.5.3.1 GET</a>	Removed x-ecs-supported-versions from the response header.	Y	Content updated.
<a href="#">3.1.5.3.1.2 Response Body</a>	71154 Clarified that the method response contains the listed entries.	Y	Content updated.
<a href="#">3.1.5.3.1.3 Processing Details</a>	Removed the processing rules for x-ecs-supported-versions.	Y	Content updated.
<a href="#">3.2.5.1 Detect Server Changes</a>	71096 Changed the GET HTTP method to the HEAD method.	Y	Content updated.
<a href="#">3.2.5.1.1 HEAD</a>	71037 Removed x-ecs-change-detection and its description from the list of HTTP headers.	Y	Content updated.
<a href="#">3.2.5.1.1 HEAD</a>	71028 Updated the description of the x-ecs-devicename request header.	Y	Content updated.
<a href="#">3.2.5.1.1 HEAD</a>	71096 Changed the GET HTTP method to the HEAD method.	Y	Content updated.
<a href="#">3.2.5.1.1.3 Processing Details</a>	71037 Added section.	Y	New content added.
<a href="#">3.3.5.1.1 GET</a>	71028 Updated the usage and description of the x-ecs-devicename request header.	Y	Content updated.
<a href="#">3.3.5.1.1.2 Response Body</a>	70987 Added the Length and AdminInfo fields to the listing of fields to be included in the response body.	Y	Content updated.
<a href="#">3.3.5.1.1.2 Response Body</a>	71154 Clarified that the method response	Y	Content updated.



Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
	contains the listed entries.		
<a href="#">3.3.5.1.1.3 Processing Details</a>	70987 Updated processing instructions for the setting of the AdminInfo field.	Y	Content updated.
<a href="#">3.4.1.2 Per Session</a>	71042 Added new variables ChangeBatchList, LastSentToken, and LastReceivedToken and their descriptions.	Y	Content updated.
<a href="#">3.4.1.2 Per Session</a>	71049 Added ClientID and SessionType variables and their descriptions.	Y	Content updated.
<a href="#">3.4.1.2.1 Per ChangeBatch</a>	71042 Added section.	Y	New content added.
<a href="#">3.4.3 Initialization</a>	71042 Added initialization instructions for LastSentToken and LastReceivedToken.	Y	Content updated.
<a href="#">3.4.5 Message Processing Events and Sequencing Rules</a>	71045 Updated processing rules related to ClientRequestsCount and ClientRequestsCountTable.	Y	Content updated.
<a href="#">3.4.5.1.1 PUT</a>	70967 Added status code 400 to the table of codes that may be returned with the response message.	Y	Content updated.
<a href="#">3.4.5.1.1.1 Request Body</a>	71154 Clarified that the method request contains the listed entries.	Y	Content updated.
<a href="#">3.4.5.1.1.1 Request Body</a>	71049 Changed ReplicaId to ClientID and changed ReplicaId description to ClientID description.	Y	Content updated.
<a href="#">3.4.5.1.1.3 Processing Details</a>	70967 Added processing instructions based on the value of the Type field.	Y	Content updated.
<a href="#">3.4.5.1.1.3 Processing Details</a>	71049 Updated Session processing instructions related to ClientID and SessionType.	Y	Content updated.
<a href="#">3.4.5.1.1.3</a>	71045	Y	Content

<b>Section</b>	<b>Tracking number (if applicable) and description</b>	<b>Major change (Y or N)</b>	<b>Change type</b>
<a href="#">Processing Details</a>	Updated processing rules for Sessions in SessionList and Session.SessionType.		updated.
<a href="#">3.4.5.2.1.2 Response Body</a>	71032 Changed SyncKnowledge Entry Type from "Array of BYTEs" to "SYNC_BLOB structure" and updated SyncKnowledge description to match this change.	Y	Content updated.
<a href="#">3.4.5.2.1.2 Response Body</a>	71154 Clarified that the method response contains the listed entries.	Y	Content updated.
<a href="#">3.4.5.2.1.3 Processing Details</a>	71050 Updated the processing rules for generating a change batch.	Y	Content updated.
<a href="#">3.4.5.2.2.1 Request Body</a>	71032 Changed SyncKnowledge Entry Type from "Array of BYTEs" to "SYNC_BLOB structure" and updated SyncKnowledge description to match this change.	Y	Content updated.
<a href="#">3.4.5.2.2.1 Request Body</a>	71154 Clarified that the method request contains the listed entries.	Y	Content updated.
<a href="#">3.4.5.2.2.2 Response Body</a>	71154 Clarified that the method response contains the listed entries.	Y	Content updated.
<a href="#">3.4.5.2.2.3 Processing Details</a>	71043 Updated the processing rules for generating a change batch.	Y	Content updated.
<a href="#">3.4.5.2.2.3 Processing Details</a>	71042 Added server instruction for the generation and use of a change batch.	Y	Content updated.
<a href="#">3.4.5.2.2.3 Processing Details</a>	71050 Updated the processing rules for generating a change batch.	Y	Content updated.
<a href="#">3.4.5.3.1.1 Request Body</a>	71154 Clarified that the method request contains the listed entries.	Y	Content updated.
<a href="#">3.4.5.3.1.2 Response Body</a>	71154 Clarified that the method response contains the listed entries.	Y	Content updated.

<b>Section</b>	<b>Tracking number (if applicable) and description</b>	<b>Major change (Y or N)</b>	<b>Change type</b>
<a href="#">3.4.5.4.1.1 Request Body</a>	71154 Clarified that the method request contains the listed entries.	Y	Content updated.
<a href="#">3.4.5.4.1.2 Response Body</a>	71154 Clarified that the method response contains the listed entries.	Y	Content updated.
<a href="#">3.4.5.4.1.3 Processing Details</a>	71030 Added section.	Y	New content added.
<a href="#">3.4.5.4.1.3 Processing Details</a>	71045 Updated processing details.	Y	Content updated.
<a href="#">3.4.5.6 Download Batch</a>	71033 Changed "upload scenario" to "download scenario".	N	Content updated.
<a href="#">3.4.5.6.1 GET</a>	70969 Modified the request and response descriptions for the x-ecs-continue HTTP header for usage and value.	Y	Content updated.
<a href="#">3.4.5.6.1.2 Response Body</a>	71154 Clarified that the method response contains the listed entries.	Y	Content updated.
<a href="#">3.4.5.6.1.3 Processing Details</a>	70969 Added processing details about x-ecs-continue.	Y	Content updated.
<a href="#">3.4.5.6.1.3 Processing Details</a>	71042 Updated server processing instructions.	Y	Content updated.
<a href="#">3.4.5.6.1.3 Processing Details</a>	71050 Updated the processing rules for downloading a batch.	Y	Content updated.
<a href="#">3.5.5.1.1.1 Request Body</a>	71154 Clarified that the method request contains the listed entries.	Y	Content updated.
<a href="#">3.5.5.1.1.2 Response Body</a>	71154 Clarified that the method response contains the listed entries.	Y	Content updated.
<a href="#">3.5.5.2.1.1 Request Body</a>	71154 Clarified that the method request contains the listed entries.	Y	Content updated.
<a href="#">3.5.5.2.1.2 Response Body</a>	71052 Clarified the entry for the response body.	Y	Content updated.

<b>Section</b>	<b>Tracking number (if applicable) and description</b>	<b>Major change (Y or N)</b>	<b>Change type</b>
<a href="#">3.5.5.2.1.2 Response Body</a>	71154 Clarified that the method response contains the listed entries.	Y	Content updated.
<a href="#">3.5.5.2.1.3 Processing Details</a>	71048 Added item list for server to perform for each entry in DownloadEntryVector.	Y	Content updated.
<a href="#">3.6.1.1 Global</a>	71042 Added ContinueToken to the list of global variables.	Y	Content updated.
<a href="#">3.6.1.1 Global</a>	71155 Added ClientID to the list of global variables.	Y	Content updated.
<a href="#">3.6.1.1 Global</a>	Removed the ADM element SyncServerVersions.	Y	Content updated.
<a href="#">3.6.1.3 Per Share</a>	70971 Added the EnterpriseId value and its description.	Y	Content updated.
<a href="#">3.6.1.4 Per DownloadFile</a>	71047 Updated the descriptions of the SyncId and ConcurrencyInfo fields.	Y	Content updated.
<a href="#">3.6.3 Initialization</a>	71037 Updated client and server instructions on share discovery and polling.	Y	Content updated.
<a href="#">3.6.3 Initialization</a>	71042 Added ContinueToken to the list of variables to be initialized.	Y	Content updated.
<a href="#">3.6.3 Initialization</a>	70971 Updated the share discovery process to include description of the EnterpriseId identifier for server responses of status 200.	Y	Content updated.
<a href="#">3.6.3 Initialization</a>	71096 Changed the GET HTTP method to the HEAD method.	Y	Content updated.
<a href="#">3.6.3 Initialization</a>	71155 Added ClientID to the list of variables to be initialized.	Y	Content updated.
<a href="#">3.6.3 Initialization</a>	Removed the initialization rules for SyncServerVersions.	Y	Content updated.
<a href="#">3.6.5.1</a>	71050	Y	Content

<b>Section</b>	<b>Tracking number (if applicable) and description</b>	<b>Major change (Y or N)</b>	<b>Change type</b>
<a href="#">Upload Scenario</a>	Updated the processing rules for creating a session.		updated.
<a href="#">3.6.5.1 Upload Scenario</a>	71155 Updated processing rules for creating a session.	Y	Content updated.
<a href="#">3.6.5.2 Download Scenario</a>	71053 Clarified that in the case of the status code of 200, the client uses the BatchMetadata entry to create the list of DownloadFile objects.	Y	Content updated.
<a href="#">3.6.5.2 Download Scenario</a>	70969 Updated the instructions that the client uses to obtain server knowledge of the change batch.	Y	Content updated.
<a href="#">3.6.5.2 Download Scenario</a>	71042 Added conditional instructions to steps 1 and 3 of the process the client uses to retrieve change batch details.	Y	Content updated.
<a href="#">3.6.5.2 Download Scenario</a>	70971 Amended the data transfer instructions to specify the association of the Share.EnterpriseId identifier with local file data.	Y	Content updated.
<a href="#">3.6.5.2 Download Scenario</a>	71048 Updated processing instructions for Data Transfer when the server responds with a status code of 200.	Y	Content updated.
<a href="#">3.6.5.2 Download Scenario</a>	71050 Updated the processing rules for creating a session.	Y	Content updated.
<a href="#">3.6.5.2 Download Scenario</a>	71155 Updated processing rules for creating a session.	Y	Content updated.
<a href="#">4 Protocol Examples</a>	70970 Added content to section.	Y	Content updated.
<a href="#">4.1 Query User Configuration Information and Detect Server Changes</a>	70970 Added section.	Y	New content added.
<a href="#">4.2 Upload Scenario</a>	70970 Added section.	Y	New content added.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
<a href="#">4.3 Download Scenario</a>	70970 Added section.	Y	New content added.
	71076 Removed sections 2.2.2.1 SYNC_GID and 2.2.2.2 SYNC_VERSION.	Y	Content removed.
	Removed section 2.2.1..11 x-ecs-supported-versions.	Y	Content removed.

## 8 Index

### A

[Applicability](#) 9

### C

[Capability negotiation](#) 9

[Change tracking](#) 81

Client

[ECS client details abstract data model](#) 60

[overview](#) 32

### D

[Download scenario example](#) 76

### E

Examples

[download scenario](#) 76

[overview](#) 67

[query user configuration information](#) 67

[upload scenario](#) 71

### F

[Fields - vendor-extensible](#) 9

### G

[Glossary](#) 7

### I

[Implementer - security considerations](#) 79

[Index of security parameters](#) 79

[Informative references](#) 8

[Introduction](#) 7

### M

Messages

[common data structures](#) 13

[common data types](#) 10

[HTML headers](#) 10

[transport](#) 10

### N

[Normative references](#) 7

### O

[Overview \(synopsis\)](#) 8

### P

[Parameters - security index](#) 79

[Preconditions](#) 9

[Prerequisites](#) 9

[Product behavior](#) 80

### Q

[Query user configuration information example](#) 67

### R

References

[informative](#) 8

[normative](#) 7

[Relationship to other protocols](#) 8

### S

Security

[implementer considerations](#) 79

[parameter index](#) 79

Server

[DetectServerChanges](#) 38

[overview](#) 32

[PeerSynchronizationSession](#) 42

[ServerAPI](#) 56

[ServiceDiscovery](#) 32

[UserConfiguration](#) 40

[Standards assignments](#) 9

### T

[Tracking changes](#) 81

### U

[Upload scenario example](#) 71

### V

[Vendor-extensible fields](#) 9

[Versioning](#) 9