

[MS-OXGLOS]: Exchange Server Protocols Master Glossary

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft's Open Specification Promise (available here: <http://www.microsoft.com/interop/osp>) or the Community Promise (available here: <http://www.microsoft.com/interop/cp/default.mspx>). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting 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.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
04/04/2008	0.1	Major	Initial Availability.
04/25/2008	0.2	Editorial	Revised and updated property names and other technical content.
06/27/2008	1.0	Major	Initial Release.
08/06/2008	1.01	Editorial	Updated references to reflect date of initial release.
09/03/2008	1.02	Editorial	Changed title of document.
12/03/2008	1.03	Editorial	Revised and edited technical content.
03/04/2009	1.04	Editorial	Revised and edited technical content.
04/10/2009	2.0	Major	Updated technical content for new product releases.
07/15/2009	3.0	Major	Revised and edited the technical content.
11/04/2009	4.0.0	Major	Updated and revised the technical content.
02/10/2010	4.1.0	Minor	Updated the technical content.
05/05/2010	5.0.0	Major	Updated and revised the technical content.
08/04/2010	6.0	Major	Significantly changed the technical content.
11/03/2010	6.1	Minor	Clarified the meaning of the technical content.

Contents

1	Non-Alphanumeric	5
2	0-9	6
3	A	7
4	B	9
5	C	10
6	D	13
7	E	15
8	F	17
9	G	18
10	H	19
11	I	20
12	J	21
13	K	22
14	L	23
15	M	24
16	N	26
17	O	28
18	P	29
19	Q	31
20	R	32
21	S	35
22	T	38
23	U	39
24	V	40
25	W	41
26	X	42
27	Y	43
28	Z	44

1 Non-Alphanumeric

.jpg: A file format for images compressed using an algorithm developed by the Joint Photographic Experts Group, as described in [\[ISO/IEC10918-1\]](#).

2 0-9

8.3 name: A file name string restricted in length to 12 characters that includes a base name of up to eight characters, one character for a period, and up to three characters for a file name extension. For more information about 8.3 file names, see [\[CIFS\]](#) section 3.2.

3 A

access control entry (ACE): An entry in an **access control list (ACL)** that contains a set of user rights and a security identifier (SID) that identifies a principal for whom the rights are allowed, denied, or audited.

access control list (ACL): A list of **access control entries (ACEs)** that collectively describe the security protections that apply to an object.

action: A discrete operation that is executed on an incoming **message** when all **conditions** in the same **rule** are TRUE. A **rule** contains one or more **actions**.

Active Directory: A general-purpose network directory service. **Active Directory** also refers to the Windows implementation of a directory service. **Active Directory** stores information about a variety of objects in the network. Importantly, user accounts, computer accounts, groups, and all related credential information used by the Windows implementation of Kerberos are stored in **Active Directory**. **Active Directory** first became available as part of Windows 2000 and is available as part of Windows 2000 Server products and Windows Server 2003 products, and planned for Windows Server 2008. **Active Directory** is not present in Windows NT 4.0 or in Windows XP. For more information, see [\[MS-SECO\]](#) section 2.2.2 and [\[MS-ADTS\]](#).

active replica: The name given to a server that hosts content and is expected to serve that content to clients.

address book: A collection of **Address Book objects**, each of which are contained in any number of **address lists**.

address book container: An **Address Book object** that describes an **address list**.

address book distinguished name (ABDN): A **distinguished name (DN)** that identifies an entry in the **address book**, such as a mail user or server. See also **address book**.

address book hierarchy table: A collection of **address book containers** arranged in a hierarchy.

Address Book object: An entity in an **address book** that contains a set of attributes, each attribute with a set of associated values.

Address Creation table: A table containing information about the templates that an address book server supports for creating new e-mail addresses.

address creation template: A template that describes how to present a dialog to a messaging user along with a script describing how to construct a new e-mail address from the user's response.

address list: A collection of distinct **Address Book objects**.

address type: An identifier for the type of e-mail address, such as **SMTP** and EX.

AddressEntry: A set of properties representing one addressee.

AddressList: A counted set of **AddressEntry** structures. Each **AddressEntry** represents one addressee.

alias: An alternative name by which an **Address Book object** can be identified.

ambiguous name resolution (ANR): A search across multiple naming-related attributes on **Address Book objects** by using a target string.

ANSI character set: The standard for the core fonts supplied with U.S. versions of Windows up to, and including, Windows 95 and Windows NT 4.0, prior to the use of **Unicode**. ANSI is an eight-bit character encoding scheme based on the U.S. English alphabet, and is used to extend the **ASCII** character set. ANSI codes represent a total of 217 text characters in computers, communications equipment, and other devices that work with text. Also known as Windows-1252.

appointment: An event created by the **organizer** for their own use.

Appointment object: A **Calendar object** that has no **attendees**.

Archive Policy: A feature used to manage when items are moved into an alternate **mailbox** for archival purposes.

ASCII: An 8-bit character encoding scheme based on the U.S. English alphabet. **ASCII** codes represent text in computers, communications equipment, and other devices that work with text. All references to **ASCII** refer to a single eight-bit **ASCII** character or an array of eight-bit **ASCII** characters with the high bit of each character set to zero. When arrays of **ASCII** characters are defined, details are included that indicate if the array of **ASCII** characters are null-terminated.

associated message: See **folder associated information (FAI)**.

Asynchronous Context Handle (ACXH): An **RPC** context **handle** used by a client when issuing **RPC** calls against a server on AsyncEMSMDB interface methods. Represents a **handle** to a unique session context on the server.

attachment: See **Attachment object**.

Attachment object: A set of properties representing a file, **Message object**, or structured storage attached to a **Message object** and visible through the **Message object's attachment table**. See also **attachment table**, **Message object**.

attachments table: The part of a **Message object** that links attachment objects to the main **message**. Each row of the table is a set of properties representing one attachment. One property represents the attachment's content. See also **Attachment object**.

attendee: A person or **resource** who is invited to a **meeting**.

Augmented Backus-Naur Form (ABNF): A meta-language used for defining formal syntax of protocols, as described in [\[RFC5234\]](#).

Autodiscover client: A client that, given an [\[RFC2821\]](#) routing address, queries for a set of server locations where setup and configuration information for that e-mail address might be stored.

Autodiscover server: A server in a managed environment that makes setup and configuration information available to **Autodiscover clients**. The location of **Autodiscover servers** is made available via the Autodiscover HTTP Service Protocol, as described in [\[MS-OXDISCO\]](#).

4 B

base64 encoding: A binary-to-text encoding scheme whereby an arbitrary sequence of bytes is converted to a sequence of printable **ASCII** characters.

basic flag: A flag on a **Message object** that indicates that the object has an associated work item or shares a defining characteristic with other **Message objects** with such flags.

Bcc recipient: See **blind carbon-copy (Bcc) recipient**.

best body: The text format that provides the richest representation of a **message body**. The algorithm for determining the best-body format is described in [\[MS-OXBBODY\]](#).

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

binary large object (BLOB): A discrete packet of binary data that can have an exceptionally large size. The designation "binary large object" typically refers to a packet of data that is treated as a sequence of uninterpreted bytes.

binding handle: Data structure that represents the logical connection between a client and a server.

blind carbon copy (Bcc) recipient: A person for whom the message is being sent but the **primary** and **Cc recipients** are the only addressees who know that they have been included.

body part: A part of an Internet message, as described in [\[RFC2045\]](#) section 2.5.

busy: One of the possible values of the **free/busy** status on an **appointment**. A **busy** status indicates that the user is not available for other **appointments** during this time.

button control: A dialog control that consists of a static string and when the control is clicked, an action is triggered.

5 C

cabinet file: A compressed package that contains a number of related files, as defined in [\[MS-CAB\]](#).

cabinet folder: A unit of solid compression. A **cabinet folder** is not the same as a file system directory or **folder**. Also called a CAB file.

calendar: A date range which shows availability, **meetings**, and **appointments** for one or more users or **resources**. See also **Calendar object**.

Calendar folder: A **Folder object** that contains **Calendar objects**.

Calendar object: A **Message object** that describes an event.

calendar options dictionary: A **dictionary** of **calendar** configuration data stored in an **FAI message** in the **Calendar special folder**.

Calendar special folder: A **Calendar folder** in a user's **mailbox** that **meetings** will be created in by default. For details about **special folders**, see [\[MS-OXOSFLD\]](#).

carbon copy (Cc) recipient: A secondary person to whom a **message** is being sent.

Cc recipient: See **carbon copy (Cc) recipient**.

change number (CN): A number that identifies a version of a **messaging object**. **Change numbers** are identical in format to **message IDs (MID)** and **folder IDs (FID)**.

change number set (CNSET): A data structure that is similar to an **IDSET**, in which the **global counters (GLOBCNTs)** represent changes rather than **messaging objects**.

character set: The range of characters used to represent textual data within a **message body part**, as described in [\[RFC2046\]](#) section 4.1.2.

charset: See **character set**.

class: A collection of **XML** elements representing an abstract data type, such as an e-mail **message**, task, document, **contact** or **calendar** item.

client-side rule: A **rule** that has at least one **action** that is executed by the client because it cannot be executed by the server.

code page: (1) An ordered set of characters of a specific script in which a numerical index (code-point value) is associated with each character.

(2) A map that specifies the one-to-one relationships between the **XML** tags of a namespace and the tags' numeric representations called tokens. Each code page corresponds to an **XML namespace**.

collection: A **folder** that contains e-mail **messages**, **contacts**, or **calendar** items.

common name (CN): A string attribute of a certificate that is one component of a **distinguished name (DN)**. The Web site or e-mail address of the certificate owner is often used as a **common name**. Client applications often refer to a certificate authority (CA) by the **common name** of its signing certificate.

Common Views folder: The **special folder** that contains the data for default views that are standard for the **message store** and that can be used by any user of a client accessing the **message store**.

complete flag: A flag on a **messaging object** that indicates that the associated work item has been completed.

component: In the context of [\[RFC2445\]](#), a hierarchical object described in an iCalendar file. An iCalendar component can recursively contain any number of child **components** and any number of **properties**. More detailed information regarding iCalendar **components** can be found in section 4.6 of [\[RFC2445\]](#).

condition: A logical expression comparing one or more properties in all incoming **messages** against a set of clauses. This logical expression can evaluate to TRUE or FALSE.

conflict: A condition that is created when an **appointment** is scheduled at the same time as another **appointment**.

Conflicts folder: The **special folder** that contains **Message objects** that indicate synchronization conflicts between the client and the server.

Connection-Oriented NTLM: One of the two variants of the NT LAN Manager (NTLM) Authentication Protocol, as described in [\[MS-NLMP\]](#).

contact: A collection of properties such as telephone numbers, e-mail addresses, and pager numbers pertaining to a person or business external to the server.

Contact object: A **Message objects** that contains properties pertaining to a contact.

contents table: A table of **messages** in a **folder**.

conversation: A single representation of a send-response series of e-mail **messages**. A **conversation** appears in the Inbox as one unit and allows the user to view and read the series of related e-mail **messages** in a single effort.

conversation action: A limited set of actions that a user applies to all **Message objects**, currently in the store or delivered in the future, that share the same [PidTagConversationId](#).

conversation ID: A unique value that is associated with a **conversation**. Each e-mail and **SMS message** is assigned a **conversation** ID to identify the **conversation** to which the e-mail or **SMS message** belongs.

conversation index: A value that specifies the location of an e-mail **message** within a **conversation**. A client can generate a tree view of a **conversation** by using a **message's conversation index** to determine the **message's** parent **message** and its child **messages**.

Coordinated Universal Time (UTC): A standard time relative to which local time zones are defined, as described in [\[ISO-8601\]](#).

COUNT: A data type that is either a 2-byte **WORD** or a 4-byte **DWORD**, depending on the context where this data type is referenced: within a given buffer, is always 2 bytes or always 4 bytes, never a mix of the two.

counter proposal: A request from an **attendee** to the **organizer** to change the date and/or time of a **meeting**.

CRLF: The **ABNF** grammar defined in [\[RFC5234\]](#).

cyclic redundancy check (CRC): (1) An algorithm used to produce a checksum (a small, fixed number of bits) against a block of data, such as a packet of network traffic or a block of a computer file. The **CRC** is used to detect errors after transmission or storage. A **CRC** is designed to catch random errors, as opposed to intentional errors. If errors might be introduced by a motivated and intelligent adversary, use a cryptographic hash function instead.

(2) A computable value which can be used to validate content when sent over the wire or decompressed.

6 D

Deferred Action Folder (DAF): A **special folder** where the server places all **Deferred Action Messages** and **Deferred Error Messages** to be acted on by the client. This **folder** is not visible to the user.

Deferred Action Message (DAM): A hidden **message** indicating to the client that it needs to execute one or more **rules** on another (user-visible) **message** in the **store**.

Deferred Error Message (DEM): A hidden **message** indicating to the client that it needs to present the user with an error indicating that a **server-side rule** failed to execute.

delegate: An individual who is acting on behalf of someone else (a **delegator**).

Delegate Information object: A special **Message object** that holds properties specifying delegator settings.

delegator: Someone granting permissions to a **delegate** to act on their behalf.

Deleted Items folder: The **special folder** that is the default location for objects that have been deleted.

delivery receipt: A report **message** generated and sent by the messaging system to the e-mail's sender or designated **recipient** when an e-mail has reached its intended **recipient**.

departmental group: A distribution list that describes a department within an organization.

dictionary: A type of configuration data that consists of a table of name-value property pairs. Each setting has a unique name property within the table.

display template: A **template** that describes how to display or allow the user to modify information about another **Address Book object**.

distinguished name (DN): A multipart name that uniquely identifies a node in a tree-structured directory. Each part of the name corresponds to a level in the tree. Mail users, servers, and other configuration elements are all identified by a **DN**.

distribution list: An **Address Book object** representing a group of one or more other **Address Book objects**. A **distribution list** can contain individual mail users and other **distribution lists**.

domain: A network of computers that share a user account database. For more information, see [\[MS-SECO\]](#) section 2.2.

Domain Name System (DNS): A hierarchical, distributed database that contains mappings of domain names to various types of data, such as IP addresses. **DNS** enables the location of computers and services by user-friendly names, and it also enables the discovery of other information stored in the database.

double-byte character set (DBCS): A **charset**, such as SHIFT_JIS, in which characters are encoded in either one or two bytes. See also **multiple-byte character set (MBCS)**, **character set**.

download: Transmission of data (payload) from a server to a client.

Draft Message object: An e-mail **Message object** that has not yet been sent.

Drafts folder: The **special folder** that is the default location for composed e-mail **Message objects** that have been saved but not sent.

dynamic endpoint: A network-specific server address that is requested and assigned at run time. For more information, see [\[C706\]](#).

7 E

E-mail object: A **Message object** that represents an e-mail **message** in a messaging **store** and that adheres to the **property** descriptions in [\[MS-OXOMSG\]](#).

E-Mail Text Body: The textual portion of a **message** that is displayed by convention by industry standard e-mail clients. The Internet mail format ([\[RFC822\]](#) section 2.3) only allowed text **messages** to be transmitted. The concept of transmitting content other than a text **message** was not codified until **MIME** was standardized. Because **MIME** derives from [\[RFC822\]](#) **messages**, the concept of a default textual portion that is displayed by e-mail clients remains. This **body part** or entity is referred to as the **E-Mail Text Body**. Handling of entities other than the **E-Mail Text Body** (**attachments**, for example) is left up to each mail client.

Embedded Message object: A **Message object** stored as an **Attachment object** within another **Message object**.

encrypted S/MIME message: An Internet e-mail **message** in the format described by [\[RFC5751\]](#) that uses the EnvelopedData CMS content type [\[RFC3852\]](#), or the **Message object** that represents such a **message**.

endpoint: (1) A client on the network that is requesting access to a network access server (NAS).

(2) A network-specific address of an RPC server process for remote procedure calls. The actual name and type of the endpoint depends on the RPC protocol sequence being used. For example, for RPC over TCP (RPC Protocol Sequence `ncacn_ip_tcp`), an endpoint might be TCP port 1025. For RPC over Server Message Block (SMB) (RPC Protocol Sequence `ncacn_np`), an endpoint might be the name of a named pipe. For more information, see [\[C7061\]](#).

(3) Any computer on a network that can transmit or receive data.

enterprise/site/server distinguished name (ESSDN): An **X500 DN** that identifies an entry in an abstract naming scheme separate from the **address book**. The naming scheme defines enterprises, which contain sites, which contain servers and users. There is no concrete data structure that embodies **ESSDNs**; instead, many address book entries contain an **ESSDN** as one of their properties.

entry ID: See **EntryID**.

EntryID: A sequence of bytes used to identify and access an object. An **EntryID** that identifies a **folder** on a messaging server includes a **folder ID (FID)**. An **EntryID** that identifies a **message** includes the **FID** in which the **folder** is contained and a **message ID (MID)**. An **EntryID** that identifies a recipient or sender of an e-mail **message** includes information specific to the address book in which the **recipient** object exists. The **recipient** type native to the Exchange Server protocols includes the **recipient's distinguished name (DN)**.

EntryList: A list used in **search folder** criteria to serialize a list of **EntryIDs**.

exception: See **Exception object**.

Exception Attachment object: An **Attachment object** on a **Recurring Calendar object** that contains the data for an **exception**, including an **Exception Embedded Message object**.

Exception Embedded Message object: An **Embedded Message object** that contains the changes for an **exception**.

Exception object: An instance of a **recurring series** that differs from the rest of the **recurring series**, for example by start time.

extended rule: A **rule** that is added to, modified, and deleted from the server using a different mechanism than regular **rules** (standard **rules**), but is otherwise functionally identical to a standard **rule**.

external users: Users who are outside an organization.

8 F

FAI contents table: A **table** of **FAI messages** in a **folder**.

feed: A data source that provides information about frequently updated content.

Finder folder: The **special folder** that contains the default **search folders**.

flags: A set of values used to configure or report options or settings.

folder: See **Folder object**.

folder associated information (FAI): Messages in a **folder** that are typically hidden from view by mail clients. **FAI messages** are used to store a variety of settings and auxiliary data, including forms, views, **calendar** options, favorites, and category lists.

folder ID (FID): An ID that refers to a **folder** in a **store**, as described in [\[MS-OXCADATA\]](#) section 2.2.1.1.

Folder object: A messaging construct. Clients and servers organize data by providing a hierarchy of objects known as **folders**, which contain **messages** and **FAI** messages.

free: One of the possible values of the **free/busy** status on an **appointment**. A **free** status indicates that the user is available during this appointment.

free/busy: Information that shows an **attendee's** availability status based on **Calendar objects**.

from properties: A group of properties that identify the original sender of a **message**.

fully qualified domain name (FQDN): In a domain naming system, an unambiguous domain name that specifies the node's position in the domain naming service tree hierarchy absolutely.

9 G

Gateway Address Routing Table (GWART): A list of values specifying the address types supported by transport gateways.

Generic Security Service Application Program Interface (GSSAPI): A programming interface that provides security services to a caller (typically, a communications protocol) in a generic fashion and that allows source-level portability of applications to different environments. For details about **GSSAPI**, see [\[RFC2743\]](#).

ghosted: A **property** that is not deleted by the server if the element is not included in a **Sync** <Change> request **message**. By default, elements that are not included in a **Sync** <Change> request are deleted from the **store**.

ghosted folder: A **folder** whose contents are located on another server.

Global Address List (GAL): The **address list** that conceptually represents the default address list for an **address book**.

global counter (GLOBCNT): An auto-incrementing 6-byte value. When a **GLOBCNT** is paired with a **REPLID**, it forms a **message ID (MID)**, **folder ID (FID)**, or **change number (CN)**. When a **GLOBCNT** is paired with a **REPLGUID**, it forms a **global identifier (GID)**.

global identifier (GID): A form of encoding of an internal identifier that makes it globally unique (across all stores). **GIDs** are a subset of external identifiers (XIDs). **GIDs** consist of a **REPLGUID** followed by a 6-byte **GLOBCNT**.

GLOBSET: A series of **global counter (GLOBCNT)** ranges.

group header: A **navigation shortcut** that groups other **navigation shortcuts**.

GUID: A 128-bit value with a low statistical likelihood of being duplicated, used in cross-process communication to identify entities such as client and server interfaces and **remote procedure call (RPC)** objects. For more information, see [\[C706\]](#).

10 H

handle: Any token that can be used to identify and access an object such as a device, a file, or a window.

handle array: An array of object **handles** that are sent to and received from the server as part of a **remote procedure call (RPC)** accompanying **ROP request buffers** and **ROP response buffers** respectively. Also known as a **server object handle table** or an HSOT table.

hard delete: To permanently remove an item from the system. When a **message** or **folder** is deleted, a backup copy of that item can be kept by the server for a defined period of time. It is not possible for the messaging client to access or restore **hard deleted** items for any period of time.

header: A name-value pair that supplies structured data in an Internet e-mail **message** [[RFC2822](#)] or **MIME** entity. Many header fields correspond to properties of the Exchange Server protocols.

hierarchy table: A **table** of **folders** in a **folder**.

Hypertext Markup Language (HTML): Text with markup as described in [[RFC2854](#)].

Hypertext Transfer Protocol (HTTP): An application-level protocol for distributed, collaborative, hypermedia information systems (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.

Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS): An extension of **HTTP** that securely encrypts and decrypts Web page requests.

11 I

ICS state: A set of properties that determine the state of a **local replica** narrowed down to a particular **synchronization scope**.

identifier: A 64-bit quantity that is used to identify items such as **folders**, **messages**, changes to **messages**, and rules. An **identifier** is composed of a **replica ID (REPLID)** and a **global counter (GLOBCNT)**. See also **folder ID (FID)**, **message ID (MID)**, **global identifier (GID)**.

IDSET: A set of IDs, or **replica ID (REPLID)** and **global counter (GLOBCNT)** pairs. Has to be represented as a formatted IDSET to be serialized on the wire.

Inbox folder: The **special folder** that is the default location for incoming (received) e-mail **Message objects**.

Incremental Change Synchronization (ICS): A data format and algorithm that is used to synchronize **folders** and **messages** between two sources.

informational update: A **Meeting Update object** that includes a change such as adding agenda details, which does not require **attendees** to re-respond.

instance: A single occurrence of an **Appointment object** or **Meeting object** that has a **recurring series** specified.

Interface Definition Language (IDL): The ISO standard language for specifying the interface for **remote procedure calls (RPC)**. For more information, see section "Interface Definition Language" in Part 3 of [\[C706\]](#).

Internet Message Access Protocol (IMAP): A protocol that is used for accessing e-mail and news items from mail servers.

Internet Message Access Protocol – Version 4 (IMAP4): The protocol described in [\[RFC3501\]](#).

Internet Message Access Protocol – Version 4 Revision 1 (IMAP4rev1): The current version of **IMAP**, as described in [\[RFC3501\]](#).

Inter-Personal Mail (IPM): Typical user messaging items, such as e-mail and **calendar** items.

interpersonal messaging subtree: The root of the hierarchy of **folders** commonly visible in a messaging client. This would include **mailbox folders** (such as **Inbox** and **Outbox**) and user-created **folders**, including user-created **public folders**.

IPM subtree: See **interpersonal messaging subtree**.

12 J

journal: To generate a **Journal-Report** for an **original-message**.

Journal object: A **Message object** that represents an entry in a **journal** or log and that adheres to the property descriptions in [\[MS-OXOJRN\]](#).

Journal-Report: A special **message** generated by the server that captures information about a single **original-message**. When an **original-message** is sent to or from users of the e-mail system, a **Journal-Report message** is generated if the **original-message** meets certain criteria configured by an administrator. **Journal-Report messages** consist of two logical parts: the body text of the **Journal-Report messages**, and the **original-message**.

Junk E-mail folder: The **special folder** that is the default location for e-mail **Message objects** that are determined to be Junk e-mail by a **Junk E-Mail rule**.

Junk E-Mail rule: A server-side **extended rule** that follows the E-Mail Rules protocol, as described in [\[MS-OXORULE\]](#), and the properties of which describe preferences for a **spam filter**.

13 K

None.

14 L

LDAP server: A server implementing the **LDAP** protocol, as described in [\[RFC4511\]](#).

legacy DN: See **enterprise/site/server distinguished name (ESSDN)**.

Lempel-Ziv Extended (LZX): An LZ77-based compression engine described in the Microsoft Cabinet SDK. **LZX** compression is a universal lossless data compression algorithm that is fast to implement and performs no analysis on the data.

Lempel-Ziv Extended Delta (LZXD): A derivative of the Microsoft Cabinet **LZX** format with some modifications to facilitate efficient delta compression. Delta compression is a technique in which one set of data can be compressed within the context of a reference set of data that is supplied both to the compressor and decompressor. Delta compression is commonly used to encode updates to similar existing data sets so that the size of compressed data can be significantly reduced relative to ordinary non-delta compression techniques. Expanding a delta-compressed set of data requires that the exact same reference data be provided during decompression.

Lightweight Directory Access Protocol (LDAP): The protocol defined by [\[RFC4511\]](#), [\[RFC4512\]](#), and [\[RFC4519\]](#).

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

local replica: A copy of the data in a **mailbox** that exists on the client.

locale: The user interface human language variation supported by an application or client computer.

Logon object: A **Server object** that provides access to either a private **mailbox** or a **public folder**. The **Logon object** is obtained by the client by issuing a **RopLogon remote operation** to the server.

LogonID: An 8-bit value used to identify a logon session within a single **RPC** session.

Long ID (LID): A 32-bit quantity that, together with a **GUID**, defines a **named property**.

long-term ID: See **LongTermID**.

LongTermID: A 192-bit value that uniquely identifies a messaging object across all messaging systems. It is composed of a 128-bit namespace **GUID**, a 48-bit counter, and 16 bits of padding. The counter's value is taken from that namespace at the time the ID was assigned.

15 M

mail tip: A note that is presented to the author of a **message** when the **message** is composed. **Mail tips** provide information about the recipients of the **message** and issues that might impact the delivery of the **message** (such as moderation or delivery restrictions).

mail user: An **Address Book object** that represents a person or entity that can receive deliverable **messages**.

Mail User Agent (MUA): A client application that is used for composing and reading e-mail, as described in [\[RFC2183\]](#).

mailbox: The combination of an **Active Directory** user object that has **mailbox** attributes and the associated **mailbox** data consisting of **messaging objects**.

meeting: An event with **attendees**.

Meeting Cancellation object: A **Message object** sent to **attendees** when the **organizer** of a **meeting** cancels a previously scheduled event.

Meeting object: A **Calendar object** with **attendees**.

meeting request: An instance of a **Meeting Request object**.

Meeting Request object: A **Message object** sent from the **organizer** to **attendees** that includes the details of a **meeting**.

Meeting Response object: A **Message object** sent from an attendee to the organizer with an **attendee's** response of accepted, tentatively accepted, or declined and optionally a new proposed **meeting** date and/or time.

Meeting Update object: A **Message object** sent from the **organizer** to **attendee** to notify them of a change to a previously scheduled **Meeting object**.

Meeting Workspace: A Web site created using the Meetings Web Services protocol, as described in [\[MS-MEETS\]](#), where documents, discussions, and other related information can be shared for an upcoming **meeting** or can be kept for a past **meeting**.

meeting-related object: Any **Message object** exchanged between the **organizer** and **attendee**. It could be any of the following **Meeting Request object**, **Meeting Update object**, **Meeting Cancellation object**, and **Meeting Response object**.

message: See **Message object**.

message body: The main **message** text of an e-mail **message**. A few properties of a **Message object** represent its **message body**, with one **property** containing the text itself and others defining its **code page** and its relationship to alternative body formats. See also **Message object**.

message class: A **property** that loosely defines the type of a **message**, **contact**, or other **Personal Information Manager (PIM)** objects in the **mailbox**.

message database (MDB): A logical and physical grouping of **mailbox** or **public folder** data.

message ID (MID): An ID that refers to a **message** in a **store**, as described in [\[MS-OXCDATA\]](#) section 2.2.1.2.

Message object: A set of properties representing an e-mail **message**, **appointment**, **contact**, or other **Personal Information Manager (PIM)** object. In addition to its own properties, a **Message object** contains recipients representing its addressees, and an **attachment table** representing files and other **Message objects** attached to the first **message**.

Message object schema: The model that describes which constituent elements the **Message object** can have that is, the sub-structure of a **Message object**.

message part: A **message body** with a string **property** containing only the portion of an e-mail **message** that is original with the **message**, not including any previous quoted **messages**. The **message part** is identical to the **message body** when the **message** does not quote a previous **message**.

Messaging Application Programming Interface (MAPI): A programming interface that enables applications to send e-mail.

messaging object: An object that exists in a **mailbox**. A **messaging object** can be either a **Folder** or **Message object** only.

metafile: A collection of structures that can store an image in an application-independent format. The stored image can be recreated by processing the **metafile** structures. Also called a vector image, a **metafile** contains a sequence of drawing commands, object definitions, and configuration settings. The commands, objects, and settings recorded in a **metafile** can be used to render its contents on a display, output by a printer or plotter, stored in memory, or saved to a file or stream.

Microsoft Interface Definition Language (MIDL): The Microsoft implementation and extension of OSF-DCE **Interface Definition Language (IDL)**. **MIDL** can also mean the **IDL** compiler provided by Microsoft. For more information, see [\[MS-RPCE\]](#).

MIME: The extension described in [\[RFC2045\]](#), [\[RFC2046\]](#), and [\[RFC2047\]](#). See also **Multipurpose Internet Mail Extensions (MIME)**.

MIME content-type: The content type described in [\[RFC2045\]](#), [\[RFC2046\]](#), and [\[RFC2047\]](#).

MIME entity: The entity described in [\[RFC2045\]](#), [\[RFC2046\]](#), and [\[RFC2047\]](#).

MIME message: The **message** described in [\[RFC2045\]](#), [\[RFC2046\]](#), and [\[RFC2047\]](#).

MIME part: The **message** part described in [\[RFC2045\]](#), [\[RFC2046\]](#), and [\[RFC2047\]](#).

multiple-byte character set (MBCS): A **charset**, such as ISO-2022-JP, in which more than one byte is required to encode at least some characters. See also **double-byte character set (DBCS)**, **charset**.

Multipurpose Internet Mail Extensions (MIME): A set of related Internet standards track recommendations, [\[RFC2045\]](#) through [\[RFC2049\]](#), that describe how to format content of various kinds for transport.

16 N

name identifier: The identifier used to refer to a **named property**. It can be either a **LONG** numerical value or a **Unicode** string. It is represented by the **Kind** member variable of the **PropertyName** structure ([\[MS-OXCADATA\]](#) section 2.6). The **Kind** member variable is of type **Union**. The value in the **Kind** member variable determines whether the name identifier is numerical or a **Unicode** string. If the value of **Kind** is 0, then the name identifier is numerical, else if the value is 1, then the name identifier is a **Unicode** string.

Name Service Provider Interface (NSPI): A method of performing **address book**-related operations on **Active Directory**.

named property: A **property** defined by a **GUID** (the property set) and either a string name or a 32-bit identifier (**LID**, dispid). A **property ID** for getting and setting **property** values is obtained using [RopGetPropertyIdsFromNames](#), and combined with a **property type** to form a **property tag**. See also **property set**, **Long ID**.

NameID: **Identifier** used to refer to a **named property**.

navigation shortcut: An object that contains identifying information to locate a **folder** in a **message database (MDB)** or an object that groups other **navigation shortcuts**.

Network Data Representation (NDR): A specification that defines a mapping from **Interface Definition Language (IDL)** data types onto octet streams. **NDR** also refers to the runtime environment that implements the mapping facilities (for example, data provided to **NDR**). For more information, see [\[MS-RPCE\]](#) and [\[C706\]](#) chapter 14.

non-delivery report (NDR): A report **message** that is generated and sent to the e-mail's sender by the server when an e-mail could not reach an intended **recipient**.

non-interpersonal messaging subtree: The root of the hierarchy of **folders** not commonly visible in a client. This would include server and client-created **folders** used principally for containing operational metadata.

non-IPM subtree: See **non-interpersonal messaging subtree**.

non-read receipt: A **message** generated during e-mail **message** deletion at the expiration of a time limit or other client-specific criteria.

non-Unicode: A **character set** that has a restricted set of glyphs, such as Shift_JIS or ISO-2022-JP.

normal message: Any **message** that is not an **FAI message**.

Note object: A **Message object** that represents a simple text note in a messaging **store** and that adheres to the property descriptions in [\[MS-OXONOTE\]](#). A **Note object** functions as an electronic equivalent of a paper sticky note.

notification: A **message** the client receives when a specific event occurs on the server.

NTLM: See **NTLM software**.

NTLM AUTHENTICATE_MESSAGE: A packet that defines an NTLM authenticate **message** that is sent from the client to the server after the **NTLM CHALLENGE_MESSAGE** is processed by the client. **Message** structure and other details of this packet are described in [\[MS-NLMP\]](#) section 2.2.1.3.

NTLM CHALLENGE_MESSAGE: A packet that defines an NTLM challenge **message** that is sent from the server to the client. The **NTLM CHALLENGE_MESSAGE** is generated by the local **NTLM software** and passed to the application that supports embedded NTLM authentication. This **message** is used by the server to challenge the client to prove its identity. **Message** structure and other details of this packet are described in [\[MS-NLMP\]](#) section 2.2.1.2.

NTLM message: A **message** that carries authentication information. Its payload data is passed to the application that supports embedded NTLM authentication by the **NTLM software** installed on the local computer. **NTLM messages** are transmitted between the client and server embedded within the application protocol that is using NTLM authentication. There are three types of **NTLM messages**: **NTLM NEGOTIATE_MESSAGE**, **NTLM CHALLENGE_MESSAGE**, **NTLM AUTHENTICATE_MESSAGE**.

NTLM NEGOTIATE_MESSAGE: A packet that defines an NTLM negotiate **message** that is sent from the client to the server. The **NTLM NEGOTIATE_MESSAGE** is generated by the local **NTLM software** and passed to the application that supports embedded NTLM authentication. This **message** allows the client to specify its supported NTLM options to the server. **Message** structure and other details are described in [\[MS-NLMP\]](#) section 2.2.1.1.

NTLM software: Software that implements the NT LAN Manager (NTLM) Authentication Protocol.

17 O

OAB data file: A file that contains **offline address book (OAB)** version 4–specific data, as described in [\[MS-OXOAB\]](#).

OAB manifest: A file that contains information about data files in the version 4 **OAB** and has a fixed, well-known name "oab.xml". By discovering the Web Distribution Point (WDP) URI and downloading the manifest, a client application gets all information necessary to download any published data file in given WDP as needed.

OAB Web distribution: A distribution mechanism, specific to **OAB** version 4, that is used to publish **OAB** data files and **OAB** manifest as a collection of files that can be downloaded by client applications using the HTTP/1.1 protocol, as described in [\[RFC2616\]](#).

OAL data sequence number: The integer number associated with Offline Address List (OAL) data that represents the generation number of this data. The initial sequence number is 1. Every subsequent data generation that produces a data set not identical to the previous one increments the sequence number by one.

offline: A client state in which the client has no connection to the server.

offline address book (OAB): A collection of **address list** in a format that the client can save and use locally.

offline address list (OAL): A portion of data in an **OAB** that is related to a single **address list**.

one-off address: An e-mail address encoded as a mail-type and address pair. Valid mail-types include values such as **SMTP**, X400, X500, and MSMAIL.

one-off EntryID: A special address object **EntryID** that encapsulates electronic address information as described in [\[MS-OXCDATA\]](#).

OOF message: A **message** sent in reply to incoming **messages** indicating that the user is currently **out of office (OOF)**.

opnum: An operation number or numeric identifier used to identify a specific **RPC** method or method in an interface. For more information, see [\[C706\]](#) section 12.5.2.12 or [\[MS-RPCE\]](#).

optional attendee: An **attendee** of an event whom the **organizer** lists as an optional participant.

organizer: The owner of an event.

original-message: A **message** for which a **Journal-Report** has been generated.

orphan instance: An instance of a **recurring series** that is in a **Calendar folder** without the **recurring series**. For all practical purposes, this is a single instance.

Out of Office (OOF): One of the possible values for **free/busy** status on an **free/busy**. It indicates that the user has indicated that he or she will be out of the office during the time of this **appointment**.

Outbox folder: The **special folder** that contains outgoing e-mail **Message object** at submit time (when the **Message object** is sent).

18 P

parameter: In the context of [\[RFC2445\]](#), a **parameter** refers to a name-value pair associated with a **property**. More detailed information regarding iCalendar parameters can be found in section 4.2 of [\[RFC2445\]](#).

Permanent Entry ID: A property of an **Address Book object** that can be used to uniquely identify the object.

permissions: Rights to access a **folder** or to perform certain operations on the **folder** based on the credentials of the user making the request.

Personal Distribution List object: A **Message object** that contains **properties** that pertain specifically to user-created **distribution lists**.

Personal Information Manager (PIM): A category of software packages for managing commonly used types of personal information. The types of personal information include **contact**, e-mail **message**, **calendar appointments**, and **meetings**.

phishing: The luring of sensitive information (such as passwords and/or other personal information) from a **recipient** by masquerading as someone trustworthy with a real need for such information.

phishing message: An e-mail **message** designed to trick the recipient into divulging sensitive information (such as passwords and/or other personal information) to a non-trustworthy source.

plain text: Text without markup. See also **plain text message body**.

plain text message body: A **message body** for which the content-type of the **E-Mail Text Body** is "text/plain". A **plain text message body** can be identified explicitly in the content received, or implicitly by the fact that the content is an [\[RFC822\]](#)-compliant **message** or the Content-Type header is missing.

port: A TCP/IP numbered connection point used to transfer data. For more information, see [\[RFC814\]](#) section 6.

portable network graphics (PNG): An extensible file format for compressed storage of bitmap images, as described in [\[ISO/IEC15948\]](#).

Post Office Protocol - Version 3 (POP3): The protocol described in [\[RFC1939\]](#).

Predecessor Change List (PCL): A set of **change numbers (CNs)** that specify the latest versions of a **messaging object** in all **replicas** that were integrated into the current version. Used for conflict detection.

primary recipient: A person for whom the **message** is directly intended.

property: (1) A strongly typed piece of data belonging to an object stored in the **mailbox**. The subject of an e-mail **message** and the e-mail address of a **recipient** are both examples of **properties**. **Properties** are identified by **property tag** or by **property name**; clients use those identifiers to get and set **property** values on specific objects. See also **tagged property**, **named property**, **property type**.

(2) In the context of [\[RFC2445\]](#), a name-value pair associated with its parent component. Properties can contain any number of parameters. For more information about iCalendar properties, see [\[RFC2445\]](#) section 4.5.

(3) A named value associated with an entity, as described in [\[RFC2518\]](#) section 13.

property ID: A 16-bit numeric identifier of a specific attribute. The **property ID** does not include any **property type** information.

property name: A string that, along with the **property set**, identifies a **named property**.

property set: A **GUID** that groups related named properties into a set.

property tag: A 32-bit value comprising a **property type** and a **property ID**. The low order 16 bits are the **property type** and the high order 16 bits are the **property ID**.

property type: A 16-bit quantity that specifies the data type of the **property's** value.

public folder: A **folder** that is stored in a location that is publicly available.

publishing: Writing **free/busy** data to a shared location.

publishing license (PL): An XrML 1.2 license that defines usage policy for protected content and contains the content key with which that content is encrypted. The usage policy identifies all authorized users and the actions they are authorized to take with the content, along with any conditions on that usage. The publishing license tells the server what usage policies apply to a given piece of content and grants the server the right to issue **use licenses (ULs)** based on that policy. The PL is created when content is protected. Also known as an "Issuance License (IL)".

19 Q

None.

20 R

read receipt: An e-mail **message** that indicates one of the **recipients** received your e-mail.

Receive folder: A **folder** that is set up to be the destination for delivered e-mails.

recipient: (1) An entity that can receive e-mail.

(2) An entity in an **address list** that can receive e-mail and contains a set of attributes, each attribute with a set of associated values.

recipient information cache: An information store that contains a list of **contacts** that the user has interacted with most often in the near term, and with whom the user is likely to interact again.

Recipient object: A set of properties representing the **recipient** of a **Message object**.

recipient properties: A group of **properties** that identify an intended **recipient** of a **message**.

recipient table: The part of a **Message object** that represents users to whom the **message** is addressed. Each row of the table is a set of properties representing one **recipient**. See also **Message object**.

recurrence BLOB: The BLOB encoding of a **recurrence pattern**, a recurrence range, and **exceptions**.

recurrence pattern: Information for a repeating event, such as the start and end time, the number of occurrences, and how occurrences are spaced (daily, weekly, monthly, and so on).

Recurring Calendar object: A **Calendar objects** that describes a **recurring series**.

recurring series: An event that repeats, at specific intervals of time, according to a **recurrence pattern**.

recurring task: A series of tasks described by a **recurrence pattern**.

relative distinguished name (RDN): A partial **DN** that is unique relative to a container at some level below the root. See also **distinguished name (DN)**.

reminder: A generally user-visible notification that a specified time has been reached. A **reminder** is most commonly related to the beginning of a **meeting** or the due time of a task but can be applied to any object type.

reminder properties: A set of **properties** that specify the attributes of a **reminder**. These attributes include the time at which and the method by which the **reminder** is to be signaled or displayed.

remote operation (ROP): An operation that can be invoked against a server. Each **ROP** represents an action, such as Delete, Send, or Query. It is encoded on the wire by a **ROP** number, followed by parameters that are specific to the operation. More than one **ROP** can be packed into a single **RPC** request or response.

remote procedure call (RPC): A framework that is used to define client-server network protocols. **RPC** protocols enable network clients to request the services of a server by using semantics that are similar to a procedure call.

replica: (1) A server that hosts an instance of the **message** items in a **folder**.

(2) A copy of the data in a user's **mailbox** at a particular version.

replica GUID (REPLGUID): A value that represents a namespace for IDs. When a **REPLGUID** is combined with a **GLOBSET**, they produce a set of **GIDs**. **REPLGUID** values can be converted into **REPLIDs** to produce **MIDs** and **FIDs**.

replica ID (REPLID): A shortened version of a **REPLGUID** that identifies a namespace for IDs within a given logon. **REPLIDs** are used on disk and on the wire for compactness, and are replaced with the corresponding **REPLGUID** for external consumption.

resource: See **Resource object**.

Resource object: An **Address Book object** that represents an asset that can be reserved, such as a room or equipment.

restriction: A filter used to map some domain into a subset of itself, by passing only those items from the domain that match the filter. **Restrictions** can be used to filter existing tables or to define new ones, such as **search folder** or **rule** criteria.

Retention Policy: A feature used to manage when items expire in accordance with policies set by the server administrator.

Rich Text Format (RTF): Text with formatting as described in [\[MSFT-RTF\]](#).

rights-managed e-mail message: An e-mail **message** that specifies permissions that are designed to protect its content from inappropriate access, use, and distribution.

rights policy template: An XrML 1.2 document that contains a predefined usage policy that is used to create the **Publishing License (PL)** when content is protected. Conceptually, a **rights policy template** (or template) is a blueprint for a **PL**, identifying authorized users and the actions they are authorized to take with the content (along with any conditions on that usage). Unlike a **PL**, a template does not contain a content key or information about the content owner. The content key and information about the content owner are required to be added when the **PL** for a given piece is created from the template. End users can use a template when protecting a document instead of defining the specifics of the usage policy themselves. When a document is published using a **template**, the **template** is used to generate the **PL**.

Root folder: The **special folder** that is the store hierarchy's top-level **folder** which contains all other **Folder objects** in that store.

ROP buffer: A buffer containing a **ROP request buffer** or **ROP response buffer** and a **Server object handle table**.

ROP request: An array of bytes specifying an operation request of a client.

ROP request buffer: A buffer containing information the messaging client sends to the server. The first BYTE in the request buffer is a **ROP**, followed by **ROP**-specific parameters.

ROP response: An array of bytes specifying an operation response of a server.

ROP response buffer: A buffer containing information the server returns to the client in response to a **ROP request buffer**. The first BYTE in the response buffer is a **ROP** (matching the **ROP** in the request buffer), followed by **ROP**-specific results.

RPC protocol sequence: A character string that represents a valid combination of an RPC protocol, a network layer protocol, and a transport layer protocol. For more information, see [\[C706\]](#) and [\[MS-RPCE\]](#).

rule: An item that defines a **condition** and an **action**. The most common use of **rules** is for new e-mail. The **condition** is evaluated for each **message** as it is delivered, and the **action** is executed if the new **message** matches the **condition**.

rules table: A **table** of **rules**.

21 S

- search criteria:** A criteria used to determine which **messages** are included in the **folder** with particular characteristics. It is composed of a restriction (the filter to be applied) and a search scope (actual **folders** where the content will be searched).
- search folder:** Provides a means of querying for items that match certain criteria. The **search folder** includes the **search folder definition message** and the **search folder container**.
- search folder container:** The **Folder object** that is created according to the specifications in the definition **message**. This **folder** is in the **Finder folder** of the **message database**.
- search folder definition message:** The **FAI message** that persists all the information defining the **search folder**. This **message** is in the associated contents table of the **Common Views folder** in the **message database**.
- Secure Sockets Layer (SSL):** A security protocol that supports confidentiality and integrity of **message** in client and server applications communicating over open networks. **SSL** uses two keys to encrypt data—a public key known to everyone and a private or secret key known only to the recipient of the **message**. **SSL** supports server and, optionally, client authentication using X.509 certificates (for more information, see [\[X509\]](#)). The **SSL** protocol is precursor to **Transport Layer Security (TLS)**. The **TLS** version 1.0 specification is based on **SSL** version 3.0.
- security descriptor:** The SECURITY_DESCRIPTOR structure that defines an entity's security. It specifies the security principal that owns the entity, the security principal(s) that can access the entity and what they can do with it, what level of audit logging has to be applied to the entity, and what type of restrictions apply to the use of the security descriptor. For more information, see [\[MS-DTYP\]](#).
- security provider:** A pluggable security module that is specified by the protocol layer above **RPC**, and will cause **RPC** to use this module to secure **messages** in a communication session with the server. The security provider is sometimes referred to as an authentication service. For more information, see [\[C706\]](#) and [\[MS-RPCE\]](#).
- security token service (STS):** A web service that issues claims and packages them in encrypted security tokens, as described in [\[WSSP1.2\]](#) and [\[WSTRUST\]](#).
- sender properties:** A group of properties that identify the sender of a **message**.
- Sent Items folder:** The **special folder** that is the default location in which copies of e-mail **Message object** are placed after they have been submitted (sent).
- Server object:** An object on a server that is used as input or created as output for **ROP**.
- Server object handle:** A 32-bit value that identifies a **Server object**.
- Server object handle table:** An array of 32-bit **handles** that are used to identify input and output **Server objects** for **ROP** requests and responses.
- server replica:** The copy of the user's **mailbox** that exists on the server.
- server-side rule:** A **rule** for which all actions are executed by the server.
- Service Connection Point (SCP):** An object made available by a directory service that clients can use to discover **Autodiscover servers**. For more information, see [\[MS-ADTS\]](#).

Session Context Handle (CXH): An **RPC** context **handle** used by a client when issuing **RPC** calls against a server on EMSMDB interface methods. Represents a **handle** to a unique session context on the server.

Short Message Service (SMS): A communications protocol designed for text **messages** to be sent between mobile phones.

ShortTermID: A 64-bit value that uniquely identifies a **messaging object** in a single namespace. It is composed of a 16-bit namespace **REPLID** and a 48-bit counter. The **REPLID** is a local namespace substitute for the actual unique namespace **GUID**. The mapping of namespace **GUIDs** to namespace **REPLID** is unique to that namespace. A **REPLID** value in one namespace can map to a different namespace **GUID** in another mapping namespace. The counter's value is taken from that namespace at the time the ID was assigned.

short-term ID: See **ShortTermID**.

signal time: The time at which a **reminder** has been specified to notify the user (or an agent acting on behalf of the user). For example, the signal time for a **meeting** that starts at 11:00 A.M. could be 10:45 A.M., thus allowing the user 15 minutes to prepare and/or travel to the **meeting** upon receiving the notification.

Simple Authentication and Security Layer (SASL): The protocol described in [\[RFC2222\]](#). This is an authentication mechanism used by the **Lightweight Directory Access Protocol (LDAP)**.

Simple Mail Transfer Protocol (SMTP): A protocol used for message delivery between Message Transfer Agents (MTAs) and between Message Submission Agents (MSAs) and MTAs. For more information, see [\[RFC2821\]](#).

single instance: An **Appointment**, **Meeting**, or **Task object** that occurs only once.

single-byte character set (SBCS): A **charset**, such as US-**ASCII**, in which all characters are encoded as a single byte. See also **DBCS**, **MBCS**, **charset**.

skip block: The block in a **binary large object (BLOB)** that acts as padding, reserving space that can be used by future versions to insert data. The block consists of a ULONG that describes how many additional ULONGs to skip ahead.

S/MIME: The standard described by [\[RFC5751\]](#).

SOAP body: A collection of zero or more element information items targeted at an ultimate SOAP receiver in the **SOAP message** path. For more information, see [\[SOAP1.1\]](#).

SOAP fault: A container for error and status information within a **SOAP message**.

SOAP header: A collection of zero or more **SOAP header** blocks, each of which might be targeted at any SOAP receiver within the **SOAP message** path.

SOAP message: An **XML** document that consists of a mandatory SOAP envelope, an optional **SOAP header**, and a mandatory **SOAP body**. **SOAP messages** are used to transmit information between a client and a server.

soft delete: To delete an item, such as a **message** or a **folder**, such that a backup copy of that item is kept by the server for a defined period of time. The backup item is referred to as a **soft deleted** item. It is possible for the messaging client to access, restore, or permanently delete **soft deleted** items.

sort order: The order in which the rows in a **table** are requested to appear. This can involve sorting on multiple properties and sorting of categories.

spam: An unsolicited e-mail **message**.

spam confidence level: A number in the range of -1 to 10 that indicates the likelihood, based on a software algorithm, that a particular **message** is **spam** (unsolicited commercial e-mail).

spam filter: A filter that checks certain conditions in a **message** to determine a **spam confidence level**.

special folder: One of a default set of **folders** used by an implementation for the storage and retrieval of user data objects.

STAT: A data structure used in the methods of the Name Service Provider Interface (NSPI) Protocol Specification that describes a variety of state information, as described in [\[MS-NSPI\]](#).

storage: A construct that can act as a container for **streams** and other **storages**. It can be thought of as analogous to a directory in a file system.

store: The unit of containment for a single hierarchy of **folders**, such as a **mailbox** or **public folders**.

Store object: An object used to store **mailboxes** and **public folder** content. For more information about **Store objects**, see [\[MS-OXCSTOR\]](#).

stream: (1) See **Stream object**.

(2) An element of a compound file, as described in [\[MS-CFB\]](#). A **stream** contains a sequence of bytes that can be read from or written to by an application, and they can exist only in **storages**.

Stream object: A **Server object** used to read and write large string and binary properties.

subject: For **folders**, refers to contained **messages** and subfolders; for **messages**, refers to **recipient** and **attachments**; for **attachments**, refers to **Embedded Message objects**.

synchronization: The act of synchronizing data (upload, download, or both).

synchronization download context: A **server object** that represents a context for an **ICS** download. For more information, see [\[MS-OXCFXICS\]](#).

synchronization scope: A set of complex criterion that defines a superset of all **messaging object** within a particular **mailbox** that are considered for a single synchronization operation. For more information, see [\[MS-OXCFXICS\]](#) section 3.3.5.2.

synchronization type: The type of **synchronization** that is occurring. This can be either a hierarchy **synchronization** or a contents **synchronization**.

synchronization upload context: A **server object** that represents a context for an **ICS** upload. For more information, see [\[MS-OXCFXICS\]](#).

22 T

table: A set of data, arranged in rows and columns, that includes the current column set, sort order, restriction, expanded/collapsed state of header rows, and so on.

Table object: An object used to view a collection of properties to an object of a specific type, such as a Message object, or a folder. A **Table object** is structured in a row and column format with each row representing an object and each column representing a property of the object.

tagged property: A property defined by a 16-bit **property ID** and a 16-bit **property type**. The **property ID** for a tagged property falls in the range 0x001 – 0x7FFF. **Property IDs** in the range 0x8000 – 0x8FFF are reserved for assignment to **named properties**.

task: See **Task object**.

Task object: A **Message object** that represents the assignment to be completed.

task request: A **Message object** that is used to issue a task assignment.

Teletex: A string value expressed as a UTF-8 string restricted to characters with values between 0x20 and 0x7E, inclusive.

template: An **Address Book object** that describes the controls and layout of a dialog to display to the user.

tentative: One of the possible values of the **free/busy** status on an **appointment**. A **tentative** status indicates that the user is tentatively booked during this **appointment**.

time flag: A flag that extends the concept of a **basic flag** by associating time-related **properties** such as start and due dates with the flag information on the **Message object**. A time flagged **Message object** is also marked with a red color flag, but it is not considered to be color flagged by definition.

To recipient: See **primary recipient**.

top-level message: A **message** that is not included in another **message** as an **Embedded Message object**. **Top-level messages** are **messaging objects**.

Transmission Control Protocol (TCP): A protocol used with the Internet Protocol (IP) to send data in the form of **message** units between computers over the Internet. **TCP** handles keeping track of the individual units of data (called packets) that a **message** is divided into for efficient routing through the Internet.

Transport Layer Security (TLS): A security protocol that supports confidentiality and integrity of **messages** in client and server applications communicating over open networks. **TLS** supports server and, optionally, client authentication by using X.509 certificates [\[X509\]](#). **TLS** is standardized in the IETF **TLS** working group.

Transport Neutral Encapsulation Format (TNEF): A binary type-length-value encoding used to encode properties for transport, as described in [\[MS-OXTNEF\]](#).

23 U

Unicode: The universal character encoding scheme for written characters and text based on the original Unicode Standard, Version 2.0 beta, as described in [\[UNICODE\]](#).

Unified Messaging: A set of components and services that enable voice, fax, and e-mail **messages** to be stored in a user's **mailbox** so that they can be accessed from a variety of devices.

Uniform Resource Identifier (URI): The **identifier** of a resource, as described in [\[RFC3986\]](#).

Uniform Resource Locator (URL): The string defined in [\[RFC1738\]](#).

universal unique identifier (UUID): A 128-bit globally unique value used in cross-process communication to identify entities, such as client and server interfaces, manager entry-point vectors, and **RPC** objects. For more information, see [\[C706\]](#). See also **GUID**.

unsendable attendee: An **attendee** to whom **meeting-related objects** will not be sent.

upload: Transmission of data (payload) from a client to a server.

Use License (UL): An XrML 1.2 license that authorizes a user to access a given protected content file and describes the usage policies that apply. Also known as an "End-User License (EUL)".

User Datagram Protocol (UDP): An application-level protocol for sending **messages** with a minimum of protocol overhead, as described in [\[RFC768\]](#).

UTF-16LE (Unicode Transformation Format, 16-bits, Little-Endian): The encoding scheme described in [\[RFC2781\]](#) for encoding **Unicode** characters as a sequence of 16-bit codes, each encoded as two 8-bit bytes with least-significant-byte first.

24 V

vCard: A **MIME** content-type that is used to hold **contact** directory information, as described in [\[RFC2426\]](#).

25 W

Web server: A computer that is running Internet Information Server (IIS) or its equivalent that stores Web pages that can be retrieved by a client.

Web Services Description Language (WSDL): An **XML** framework for describing network services, as described in [\[WSDL\]](#).

WebDAV: Web Distributed Authoring and Versioning protocol, as described in [\[RFC2518\]](#).

WebDAV client: A computer that uses the **WebDAV** protocol, as described in [\[RFC2518\]](#), to retrieve data from the **WebDAV server**.

WebDAV server: A computer that supports the **WebDAV** protocol, as described in [\[RFC2518\]](#), to and from which **WebDAV clients** can connect and retrieve data.

well-known endpoint: A preassigned, stable **endpoint** that a server can use every time it runs. **Well-known endpoint** information is stored as part of the **binding handle**. For more information, see [\[C706\]](#).

Wireless Application Protocol (WAP): A protocol that is primarily used to access the Internet from a mobile device, as described in [\[WAP\]](#).

Wireless Application Protocol (WAP) Binary XML (WBXML): A compact binary representation of **XML** designed to reduce the transmission size of **XML** documents over narrowband communication channels.

WSDL message: An abstract, typed definition of data being communicated, as defined in [\[WSDL\]](#).

WSDL port type: A named set of abstract operations and the abstract **messages** involved.

26 X

X500 DN: A **distinguished name (DN)** of an object in an **address book**, in **Teletex** form. An **X500 DN** can be more limited in the size and number of **relative distinguished names (RDNs)** than the full **DN** of the object.

XML: The Extensible Markup Language, as described in [\[XML10\]](#).

XML namespace: A namespace that is identified by a URI reference [\[RFC3986\]](#). Element and attribute names can be placed in an **XML namespace** by using the mechanisms described in [\[MS-OXWUMS\]](#).

XML schema: A schema that consists of components such as type definitions and element declarations. These can be used to assess the validity of well-formed element and attribute information items.

XML schema definition (XSD): A language proposed by the W3C XML Schema Working Group for use in defining schemas. Schemas are useful for enforcing structure and/or constraining the types of data that can be used validly within other **XML** documents. **XML schema definition** refers to the fully specified and currently recommended standard for use in authoring **XML** schemas.

27 Y

None.

28 Z

None.

29 Change Tracking

This section identifies changes that were made to the [MS-OXGLOS] protocol document between the August 2010 and November 2010 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

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

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

- A document revision that incorporates changes to interoperability requirements or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- Changes made for template compliance.
- 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 language and formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

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

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

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

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

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

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

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

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change Type
15 M	58269 Added definition for the term "Messaging Application Programming Interface (MAPI)".	N	Content updated.
21 S	57125 Added "synchronization type" term and definition.	N	New content added.
21 S	58932 Added "security token service (STS)" term and definition.	N	New content added.