[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 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.
- **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.
- **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
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.
03/18/2011	6.2	Minor	Clarified the meaning of the technical content.
08/05/2011	6.3	Minor	Clarified the meaning of the technical content.
10/07/2011	6.4	Minor	Clarified the meaning of the technical content.
01/20/2012	6.5	Minor	Clarified the meaning of the technical content.
04/27/2012	6.6	Minor	Clarified the meaning of the technical content.
07/16/2012	6.7	Minor	Clarified the meaning of the technical content.
10/08/2012	6.8	Minor	Clarified the meaning of the technical content.
02/11/2013	6.9	Minor	Clarified the meaning of the technical content.
07/26/2013	6.10	Minor	Clarified the meaning of the technical content.
11/18/2013	6.11	Minor	Clarified the meaning of the technical content.
02/10/2014	6.12	Minor	Clarified the meaning of the technical content.
04/30/2014	6.12	No change	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
07/31/2014	6.13	Minor	Clarified the meaning of the technical content.
10/30/2014	7.0	Major	Significantly changed the technical content.

Table of Contents

1 Non-Alphanumeric

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 on 8.3 **file** names, see [MS-CIFS] section 2.2.1.1.1.

88 object class: An **object class** as specified in the X.500 directory specification ([X501] section 8.4.3). An **88 object class** can be instantiated as a new **object**, like a **structural object class**, and on an existing **object**, like an **auxiliary object class**.

3 A

Abstract Syntax Notation One (ASN.1): A notation to define complex data types to carry a message, without concern for their binary representation, across a network. **ASN.1** defines an encoding to specify the data types with a notation that does not necessarily determine the representation of each value. **ASN.1** encoding rules are sets of rules used to transform data that is specified in the **ASN.1** language into a standard format that can be decoded on any system that has a decoder based on the same set of rules. **ASN.1** and its encoding rules were once part of the same standard. They have since been separated, but it is still common for the terms **ASN.1** and **Basic Encoding Rules (BER)** to be used to mean the same thing, though this is not the case. Different encoding rules can be applied to a given **ASN.1** definition. The choice of encoding rules used is an option of the protocol designer. ASN.1 is described in the following specifications: [ITUX660] for general procedures; [ITUX680] for syntax specification; [ITUX690] for the **Basic Encoding Rules (BER)**, Canonical Encoding Rules (CER), and Distinguished Encoding Rules (DER) encoding rules; and [ITUX691] for the **Packed Encoding Rules (PER)**. Further background information on ASN.1 is also available in [DUBUISSON].

abstract type: A type used in this specification whose representation need not be standardized for interoperability because the type's use is internal to the specification. See **concrete type**.

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 rules for authorizing access to some resource; for example, an object or set of objects.

access mask: A 32-bit value present in an **access control entry (ACE)** that specifies the allowed or denied rights to manipulate an object.

account: (1) A collection of data and settings for a SharePoint Workspace or Groove identity that represents a user. This includes shared spaces, messages, and preferences that are associated with a user's identity. An account can reside on one or more devices.

(2) A **user** (including machine account), **group**, or **alias** object. Also a synonym for security principal or principal.

accounting: Information gathered and maintained by the **management service** about the runtime behavior of processes. The **management service** provides an accounting state switch with two settings: enabled and disabled. When enabled, accounting information is gathered and persisted across invocations of the **management service**. Accounting information gathered by the **management service** on one computer can be persisted by the **management service** on a different computer. When the accounting state is disabled, no accounting data is gathered or persisted.

action: (1) The smallest unit of work in a workflow system. An action can contain one or more tasks that define work that **actors** need to do. Actions are deployed and registered in the workflow system to be activated by protocol client users.

(2) A unit of work that can be performed by a workflow and is typically defined in a workflow markup file.

(3) A discrete operation that is executed on an incoming **Message object** when all **conditions** in the same **rule** (4) are TRUE. A rule contains one or more actions.

(4) A string that is returned as part of a GetAction response in the Desired State Configuration Pull Model Protocol [MS-DSCPM].

(5) A command exposed by a **service** which takes one or more input or output arguments and which may have a return value. For more information, see [UPNPARCH1.1] sections 2 and 3.

(6) A remote procedure call from the control point to a particular service on the device.

(7) A command that is exposed by a **service**, as defined in [UPNPARCH1.1] section i.7.

(8) An interactivity event in a report, such as a hyperlink, bookmark link, or drillthrough link, that is associated with an item in a report.

(9) A business rule argument that determines what occurs when the business rule is run at validation time.

(10) An OLAP object, such as a cube, dimension, and cell, that has an action associated with it, so that a user can perform that action when browsing OLAP data. For example, a user can jump to a URL, execute a command, or drill through to data.

Action: A type of **MetadataObject** that represents a URL that triggers the display or manipulation of data related to an Entity or EntityInstance. Actions are contained by an Entity. Actions contain **ActionParameters**.

action instance: The runtime instance of a specific **action** (1). Action instances are building blocks for an **activity flow**. Several action instances can be chained together to form an activity flow, and multiple action instances of the same action can exist in a single activity flow.

ActionParameter: A type of **MetadataObject** that defines how to parameterize the URL of an **Action** with specific data about an EntityInstance. ActionParameters are contained by Actions.

activation: (1) An operation that creates a new **action instance**.

(2) In **COM**, a local mechanism by which a client provides the **CLSID** of an **object class** (3) and obtains an **object** (3), either an **object** from that **object class** or a **class factory** that is able to create such **objects**.

(3) In the **DCOM** protocol, a mechanism by which a client provides the **CLSID** of an **object class** (4) and obtains an **object** (4), either from that **object class** or a **class factory** that is able to create such **objects**. For more information, see [MS-DCOM].

(4) The process of creating a server object.

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**. See also **Lightweight Directory Access Protocol (LDAP)** versions 2 and 3, **Kerberos**, and **DNS**. For more information, see [MS-AUTHSOD] section 1.1.1.5.2 and [MS-ADTS]. **Active Directory** is either deployed as **Active Directory Domain Services (AD DS)** or **Active Directory Lightweight Directory Services (AD LDS)**. [MS-ADTS] describes both forms.

Active Directory Domain Services (AD DS): An operating system **directory service (DS)** implemented by a **domain controller (DC)**. The **DS** provides a data store for **objects** that is

distributed across multiple **DCs**. The **DCs** interoperate as peers to ensure that a local change to an **object** replicates correctly across **DCs**. For more information, see [MS-AUTHSOD] section 1.1.1.5.2. For information about product versions, see [MS-ADTS].

Active Directory Lightweight Directory Services (AD LDS): A **directory service (DS)** implemented by a **domain controller (DC)**. The most significant difference between **AD LDS** and **Active Directory Domain Services (AD DS)** is that **AD LDS** does not host **domain naming contexts (domain NCs)**. A server can host multiple **AD LDS DCs**. Each **DC** is an independent **AD LDS** instance, with its own independent state. **AD LDS** can be run as an operating system **DS** or as a directory service provided by a standalone application (ADAM). For more information, see [MS-ADTS].

Active Directory object: A set of **directory objects** that are used within **Active Directory** as defined in [MS-ADTS] section 3.1.1. An **Active Directory object** can be identified by a **dsname**. See also **directory object**.

Active Directory partition: A synonym for naming context (NC) replica.

active reminder: A **reminder** that is enabled on an object and is either pending or overdue, depending on whether the **signal time** has passed.

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

active search folder: A **search folder** (2) that has a **search folder container** and is up-to-date with the correct **search criteria**.

activity flow: A running instance of a workflow that consists of a sequence of **action instances** and **activity model** instances. Action instances and activity model instances can be sequenced in any order to create a single activity flow.

activity model: A predefined sequence of **actions** (1).

actor: A person or process that starts or participates in an **activity flow**. An actor can be an initiator or a target.

AD LDS: See **Active Directory Lightweight Directory Services (AD LDS)**.

adapter: The hardware that connects to a particular network segment. A bound LAN card is one example of an **adapter**. Similarly, a computer with two modems, each capable of connecting to a remote network, has two adapters, one to represent each modem.

add-in: Supplemental functionality that is provided by an external application or macro to extend the capabilities of an application.

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 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** (1), 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 email 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 email address from the user's response.

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

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

AD-type server: An LDAP server that returns an object identifier (OID) value of "1.2.840.113556.1.4.800" when it is queried for the supportedCapabilities LDAP attribute.

Advanced Systems Format (ASF): An extensible file format that is designed to facilitate streaming digital media data over a network. This file format is used by Windows Media.

alias: (1) An alternate name that can be used to reference an object or element.

(2) A simple identifier that is typically used as a short name for a **namespace**.

(3) A **group** (1) that is local to a particular machine (as opposed to a **group** that has security permissions and settings for the entire **domain**).

ambiguous name resolution (ANR): (1) A search algorithm that permits a client to search multiple naming-related **attributes** (2) on objects by way of a single clause of the form "(anr=value)" in a Lightweight Directory Access Protocol (LDAP) search filter. This permits a client to query for an object when the client possesses some identifying material related to the object but does not know which attribute

(2) of the object contains that identifying material.

(3) A search algorithm that permits a **client** to search multiple naming-related **attributes** on **objects** by way of a single clause of the form "(anr=value)" in a **Lightweight Directory Access Protocol (LDAP)** search filter. This permits a **client** to query for an **object** when the **client** possesses some identifying material related to the **object** but does not know which **attribute** of the **object** contains that identifying material.

American National Standards Institute (ANSI) character set: A **character set** (1) defined by a **code page** approved by the American National Standards Institute (ANSI). The term "ANSI" as used to signify Windows code pages is a historical reference and a misnomer that persists in the Windows community. The source of this misnomer stems from the fact that the Windows code page 1252 was originally based on an ANSI draft, which became International Organization for Standardization (ISO) Standard 8859-1 [ISO/IEC-8859-1]. In Windows, the ANSI character set can be any of the following code pages: 1252, 1250, 1251, 1253, 1254, 1255, 1256, 1257, 1258, 874, 932, 936, 949, or 950. For example, "ANSI application" is usually a reference to a non-**Unicode** or code-page-based application. Therefore, "ANSI character set" is often misused to refer to one of the character sets defined by a Windows code page that can be used as an active system code page; for example, character sets defined by code page 1252 or character sets defined by code page 950. Windows is now based on **Unicode**, so the use of ANSI character sets is strongly discouraged unless they are used to interoperate with legacy applications or legacy data.

anchor text: The text that is included with a hyperlink to describe the target content of a hyperlink.

anonymous user: A user who presents no credentials when identifying himself or herself. The process for determining an anonymous user can differ based on the authentication protocol, and the documentation for the relevant authentication protocol should be consulted.

app for Office: A cloud-enabled app that integrates rich, scenario-focused content and services into an Office application or equivalent protocol client.

application: A participant that is responsible for beginning, propagating, and completing an atomic transaction. An application communicates with a transaction manager in order to begin and complete transactions. An application communicates with a transaction manager in order to marshal transactions to and from other applications. An application also communicates in application-specific ways with a resource manager in order to submit requests for work on resources.

application NC: A specific type of **naming context (NC)**, or an instance of that type, that supports only full replicas (no partial replicas). An **application NC** cannot contain **security principal objects**. An **application NC** can contain **dynamic objects**. A **forest** can have zero or more **application NCs**. Application NCs do not appear in the **global catalog (GC)**. The root of a **domain NC** is an object of **class** domainDns.

application server: A computer that provides infrastructure and services for applications that are hosted on a server farm.

Appointment object: A **Calendar object** that has an organizer but no attendees.

archive: The Fax Archive Folder, as described in section 3.1.1.

archive policy: A feature that determines when items are moved into an alternate **mailbox** for archival purposes.

archive tag: An element that contains information about the **archive policy** of a **Message object** or folder.

array: A **Remoting Type** that is an ordered collection of values. The values are identified by their position and position is determined by a set of integer indices. The number of indices required to represent the position is called the Rank of the **Array**. An **Array** is part of the **Remoting Data Model** and also specifies the **Remoting Type** of its items. For more information, [MS-NRTP] section 3.1.1.

ASCII: The American Standard Code for Information Interchange (ASCII) is an 8-bit character-encoding scheme based on the English alphabet. ASCII codes represent text in computers, communications equipment, and other devices that work with text. ASCII refers to a single 8-bit ASCII character or an array of 8-bit ASCII characters with the high bit of each character set to zero.

ASN.1: Abstract Syntax Notation One. ASN.1 is used to describe Kerberos datagrams as a sequence of components, sent in messages. ASN.1 is described in the following specifications: [ITUX660] for general procedures; [ITUX680] for syntax specification, and [ITUX690] for the Basic Encoding Rules (BER), Canonical Encoding Rules (CER), and **Distinguished Encoding Rules (DER)** encoding rules.

ASP.NET: A web server technology for dynamically rendering HTML pages using a combination of HTML, Javascript, CSS, and server-side logic. For more information, see [ASPNET].

association: A named independent relationship between two EntityType definitions. Associations in the **Entity Data Model (EDM)** are first-class concepts and are always bidirectional. Indeed, the first-class nature of associations helps distinguish the **EDM** from the relational model. Every association includes exactly two association ends.

Association: A **MethodInstance** that enables the traversal and manipulation of a data model relationship between a set of source **Entities** and a single destination Entity. An Association

can retrieve, associate, and disassociate EntityInstances of a destination Entity if given EntityInstances of other source Entities.

asynchronous context handle: A **remote procedure call (RPC)** context handle that is used by a client when issuing RPCs against a server on AsyncEMSMDB interface methods. It represents a handle to a unique session context on the server.

Asynchronous JavaScript + XML (AJAX): A web programming model that incorporates a set of web technologies including Extensible HyperText Markup Language (XHTML), **cascading style sheets (CSS)**, Document Object Model (DOM), **XML**, Extensible Stylesheet Language Transformation (XSLT), **XMLHTTPRequest (XHR)**, and JavaScript. AJAX is designed to make user interaction with the web more responsive.

atom feed: An **XML** structure that contains metadata about content, such as the language version and the date when the content was last modified, and is sent to subscribers by using the Atom Publishing Protocol (AtomPub), as described in [RFC4287].

Atom Publishing Protocol (AtomPub): An application-level protocol for publishing and editing web resources, as described in [RFC5023].

atomic transaction: A shared activity that provides mechanisms for achieving the atomicity, consistency, isolation, and durability (ACID) properties when state changes occur inside participating **resource managers**.

attachment: An external file that is included with an **Internet message** or associated with an item in a SharePoint list.

Attachment object: A set of properties that represents a file, **Message object**, or structured storage that is attached to a Message object and is visible through the **attachments table** for a Message object.

attachments table: A **Table object** whose rows represent the **Attachment objects** that are attached to a **Message object**.

attribute: (1) A characteristic of some object or entity, typically encoded as a name-value pair.

(2) (A specialization of the previous definition.) An identifier for a single or multivalued data element that is associated with a directory object. An object consists of its attributes and their values. For example, cn (common name), street (street address), and mail (email addresses) can all be attributes of a user object. An attribute's schema, including the syntax of its values, is defined in an attributeSchema object.

(3) A characteristic of some **object** or entity, typically encoded as a name-value pair.

(4) (A specialization of the previous definition.) An identifier for a single or multivalued data element that is associated with a directory **object**. An **object** consists of its **attributes** and their values. For example, cn (common name), street (street address), and mail (email addresses) can all be **attributes** of a **user object**. An **attribute's** schema, including the syntax of its values, is defined in an attributeSchema **object**.

attribute syntax: Specifies the format and range of permissible values of an attribute. The syntax of an attribute is defined by several attributes on the attributeSchema object. Attribute syntaxes supported by **Active Directory** include Boolean, Enumeration, Integer, LargeInteger, String(UTC-Time), Object(DS-DN), and String(Unicode).

Augmented Backus-Naur Form (ABNF): A modified version of Backus-Naur Form (BNF), commonly used by Internet specifications. ABNF notation balances compactness and simplicity

with reasonable representational power. ABNF differs from standard BNF in its definitions and uses of naming rules, repetition, alternatives, order-independence, and value ranges. For more information, see [RFC5234].

authenticated context: The runtime state that is associated with the successful authentication of a security principal between the client and the server, such as the security principal itself, the cryptographic key that was generated during authentication, and the rights and privileges of this security principal.

authentication: (1) The ability of one entity to determine the identity of another entity.

(2) The act of proving an identity to a server while providing key material that binds the identity to subsequent communications.

(3) The ability of one entity to determine the identity of another entity by proving an identity to a **server** while providing key material that binds the identity to subsequent communications.

authentication server: The entity that verifies that a person or thing is who or what it claims to be (typically using a cryptographic protocol) and issues a ticket or token attesting to the validity of the claim. The total set of authentication protocol **security support providers (SSPs)** that are typically available on a Windows server release.

Authentication Service (AS): A service that issues ticket granting tickets (TGTs), which are used for authenticating **principals** within the **realm** or **domain** served by the **Authentication Service**.

authority: (1) The first portion of a peer name. For secure peer names, this is a hash of a public key represented as 40 hexadecimal characters in printable form. For unsecured peer names, this is "0".

(2) A hierarchical element in a **URI scheme** used for delegating governance of the name space defined by the remainder of the **URI**, as defined in [RFC3986] section 3.2.

authorization: The secure computation of roles and accesses granted to an identity.

Autodiscover client: A client that queries for a set of server locations where setup and configuration information for an [RFC2821]-compliant email address is 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].

auxiliary class: See **auxiliary object class**.

auxiliary object class: An **object class** that cannot be instantiated in the directory but can be either added to, or removed from, an existing **object** to make its **attributes** available for use on that **object**; or associated with an **abstract** or **structural object class** to add its **attributes** to that **abstract** or **structural object class**.

availability: A numerical value that indicates whether a user can be interrupted for communication. The higher the number, the less available the user.

4 B

back link attribute: A **constructed attribute** whose values include **object references** (for example, an **attribute** of **syntax** Object(DS-DN)). The **back link values** are derived from the values of a related **attribute**, a **forward link attribute**, on other **objects**. If f is the **forward link attribute**, one **back link value** exists on **object** o for each **object** r that contains a value of o for **attribute** f. The relationship between the **forward link attributes** and **back link attributes** is expressed using the linkId **attribute** on the attributeSchema **objects** representing the two **attributes**. The forward link's linkId is an even number, and the back link's linkId is the forward link's linkId plus one. For more information, see [MS-ADTS] section 3.1.1.1.6.

back link value: The value of a **back link attribute**.

back-end database server: A server that hosts data, configuration settings, and stored procedures that are associated with one or more applications.

bare email address: A specific Internet identifier that contains a locally interpreted string followed by an at sign (@) and an Internet domain, as described in [RFC2822].

base license: A reference Windows Media DRM **policy** from which a Windows Media DRM **license** (1) is derived.

base property type: The type of the **property**, if the **property** is single-valued, or the type of an element of the **property**, if the **property** is multi-valued.

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

base64 encoding: A binary-to-text encoding scheme whereby an arbitrary sequence of bytes is converted to a sequence of printable **ASCII** characters, as described in [RFC4648].

basic disk: A disk on which each **volume** can be composed of exclusively one **partition**.

Basic Encoding Rules (BER): A set of encoding rules for **ASN.1** notation. These encoding schemes allow the identification, extraction, and decoding of data structures. These encoding rules are defined in [ITUX690].

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.

best body: The text format that provides the richest representation of a **message body** (2). 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): (1) A discrete packet of data that is stored in a database and is treated as a sequence of uninterpreted bytes.

(2) A collection of binary data stored as a single entity in a database.

bind: (1) The process of connecting controls to fields or groups in the **data source** (2) of an InfoPath form. When controls are bound to fields and groups, changes to the data in a control automatically update the data that is in the corresponding fields or groups in the data source. Similarly, changes to the data in the data source automatically update the data that is displayed in the controls that are bound to the corresponding fields and groups.

(2) To associate two EntityType [MC-CSDL] instances. An EntityType instance in a data service (described by using **Entity Data Model (EDM)** constructs) may be related to one or more other **conceptual schema definition language (CSDL)** instances. This relationship is represented by using an **association** in an **EDM**. The **cardinality** of a relationship can be determined by inspecting the **EDM** that describes the data service. The act of associating two EntityType instances is known as "binding" and of disassociating two instances is known as "unbinding". If two EntityType instances are already associated, they are considered to be "bound".

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

blind carbon copy (Bcc) recipient: An addressee on a **Message object** that is not visible to recipients of the Message object.

block: (1) A set of **deltas** that are used to define an order for those deltas. Each block consists of one or more groups of deltas.

(2) A subdivision of a segment. Each segment is divided into blocks of equal size (64 kilobytes (KB)) except for the last block in the last segment, which can be smaller if the content size is not a multiple of the standard segment sizes. In version 2.0 Content Information, segments are not divided into blocks.

(3) A chunk of **content** that composes a **segment**. Each **segment** is divided into one or more **blocks**. Every **block** belongs to a specific **segment**, and within a **segment**, **blocks** are identified by their progressive index. (Block 0 is the first **block** in the **segment**, block 1 is the second, and so on.) See [MS-PCCRC] for more details.

blog: (1) A website that contains a series of posts about a subject and is arranged in reverse chronological order. Also referred to as web log.

(2) The process of writing or publishing entries to a blog.

body: (1) The contents of a **body part** or an entire message that contains several body parts, as described in [RFC2045].

(2) The fax pages other than the cover page.

(3) The main component of the **report** within which the details and contents are included.

body part: A part of an Internet message, as described in [RFC2045].

bookmark: (1) An entity that is used in a document to denote the beginning and ending character positions of specific text in the document, and optionally, metadata about that text or its relationship to other referenced parts of the document.

(2) A data structure that the server uses to point to a position in the **Table object**. There are three pre-defined bookmarks (beginning, end, and current). A custom bookmark is a server-specific data structure that can be stored by the client for easily navigating a **Table object**.

(3) A marker that uniquely identifies a **row** within a set of rows.

(4) An anchor that is used in a report to assist navigation, typically through the use of hyperlinks. A bookmark link in a report sends the user to another location in the report.

bot: A structured HTML comment that is processed by a front-end web server when the containing document is opened by or saved to the server. Also referred to as web bot.

bucket web: A site that is used to store content for a specific **category** (1).

business logic: (1) A set of rules, formulas, validation, and code that define the limits and methods for processing data that is entered into an InfoPath form.

(2) The part of an application that processes data according to the requirements defined in a **line-of-business (LOB) system**. It refers to the routines that perform the data entry, update, query, and report processing, and more specifically to the processing that takes place behind the scenes rather than the presentation logic that is required to display the data.

5 C

cabinet (.cab) file: A single file that stores multiple compressed files to facilitate storage or transmission.

calendar: (1) A date range that shows availability, **meetings**, and appointments for one or more users or **resources**. See also **Calendar object**.

(2) A method of controlling which **resource allocation policy (RAP)** is selected as the **current resource policy**. The calendar maintains start and end dates and times for **RAP** and is either enabled or disabled. When enabled, the **management service** continuously monitors start and end dates and times of the scheduled **RAP** to activate the correct **current resource policy**. When disabled, the **RAP** scheduled on the calendar has no effect on which **RAP** is the **current resource policy**.

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

Calendar object: A **Message object** that represents an event, which can be a one-time event or a recurring event. The Calendar object includes properties that specify event details such as description, organizer, date and time, and status.

calendar options dictionary: A dictionary that contains calendar configuration data. It is stored in a **folder associated information (FAI)** message that is in a **Calendar special folder**.

Calendar special folder: A **Calendar folder** that is in a user's **mailbox** and in which meetings are created by default.

call: A communication between **peers** that is configured for a multimedia conversation.

callback: (1) A concept in which the originator of a **call** is called back by the responder. In dial-up communication (like ISDN/PSTN), the originator of the dial-up hangs up after indicating the interest to be called back. The responder then calls up the originator to establish the communication.

(2) The mechanism through which a remote access client gets called back by the server in order to establish connectivity.

callback address: An object that encapsulates an Internet address that is registered by a client and that a server can use for push **notifications**.

callee: An **endpoint** (5) to which a **call** is initiated by a **caller**.

caller: (1) An **endpoint** (5) that initiates a **call** to establish a media session.

(2) The originator of a **call**. The **network access client (NAC)** is typically the **caller**. The **NAC** and **NAS** might choose to negotiate and use **callback**, in which case the **caller** role is reversed for the **callback** itself, with the **NAS** being the **caller**.

candidate: A set of **transport addresses** that form an atomic unit for use with a media session. For example, in the case of Real-Time Transport Protocol (RTP) there are two transport addresses for each candidate, one for RTP and another for the Real-Time Transport Control Protocol (RTCP). A candidate has properties such as type, priority, foundation, and base.

carbon copy (Cc) recipient: An address on a **Message object** that is visible to recipients of the Message object but is not necessarily expected to take any action.

cardinality: The measure of the number of elements in a set.

Cartridge: A unit of **physical media** on which information may be stored. Cartridges come in various types, including 8-mm tape, magnetic disks, optical disks, and CD-ROMs. Some **cartridges** have multiple **sides**.

cascading style sheet (CSS): An extension to **HTML** that enables authors and users of HTML documents to attach style sheets to those documents, as described in [CSS-LEVEL1] and [CSS-LEVEL2]. A style sheet includes typographical information about the appearance of a page, including the font for text on the page.

catalog: (1) A table that defines the structure and relationships of a set of tables in a database.

(2) A data store that holds the configuration properties for **components** and **conglomerations**.

(3) The highest-level unit of organization in the **indexing service**. It represents a set of **indexed** documents against which queries can be executed by using the [MS-MCIS].

(4) The highest-level unit of organization in the Windows Search service. It represents a set of indexed documents against which queries can be executed by using the [MS-WSP].

category: (1) A custom string that is used to group one or more documents.

(2) A string that is used as a suggestion for a document category on a site.

(3) A subdivision of items into useful groups such as geographical regions. For example, categories that represent geographical regions could be North, South, East, and West.

(4) An enhanced presence concept that is used by a **Session Initiation Protocol (SIP)** client to publish or subscribe to **presence** (2) information. A category enables basic identification of the data that is being published; it implies an agreed-upon schema for interpreting the data. A category name identifies a contract between a publisher and a subscriber.

(5) A grouping of rows in a **Table object** that all have the same value for a specified property.

(6) A logical grouping of **updates** identified by a **GUID** and described by **metadata**. A category can be treated as an **update** with no associated **content**.

(7) A hierarchical grouping of rows. For example, a query result that contains author and title columns can be categorized based on author. Each group of rows containing the same value for author would constitute a category.

(8) A group of updates. Each update belongs to zero or more update categories. An update category can be a product category that contains updates for a particular product, or a classification category that contains updates of a particular classification (for example, all security updates). A category can have a parent category as well as child categories.

certificate: (1) A certificate is a collection of **attributes** (1) and extensions that can be stored persistently. The set of attributes in a certificate can vary depending on the intended usage of the certificate. A certificate securely binds a public key to the entity that holds the corresponding private key. A certificate is commonly used for **authentication** (2) and secure exchange of information on open networks, such as the Internet, extranets, and intranets. Certificates are digitally signed by the issuing **certification authority (CA)** (1) and can be issued for a user, a computer, or a service. The most widely accepted format for certificates is

defined by the ITU-T X.509 version 3 international standards. For more information about attributes and extensions, see [RFC3280] and [X509] sections 7 and 8.

(2) When referring to X.509v3 certificates, that information consists of a public key, a **distinguished name (DN)** (3) of some entity assumed to have control over the private key corresponding to the **public key** in the certificate, and some number of other attributes and extensions assumed to relate to the entity thus referenced. Other forms of certificates can bind other pieces of information.

(3) As used in this document, **certificates** are expressed in [XRML] section 1.2.

certificate authority (CA): See **certification authority (CA)**.

certificate revocation list (CRL): A list of **certificates** (1) that have been revoked by the **certification authority (CA)** that issued them (that have not yet expired of their own accord). The list must be cryptographically signed by the **CA** that issues it. Typically, the certificates are identified by serial number. In addition to the serial number for the revoked certificates, the CRL contains the revocation reason for each certificate and the time the certificate was revoked. As described in [RFC3280], two types of CRLs commonly exist in the industry. Base CRLs keep a complete list of revoked certificates, while delta CRLs maintain only those certificates that have been revoked since the last issuance of a base CRL. For more information, see [X509] section 7.3, [MSFT-CRL], and [RFC3280] section 5.

certification authority (CA): (1) A third party that issues public key **certificates** (1). Certificates serve to bind public keys to a user identity. Each user and certification authority (CA) can decide whether to trust another user or CA for a specific purpose, and whether this trust should be transitive.

(2) A software component that issues digital (X.509) **certificates** (2) to identities based on a public/private key pair. For more information, see [RFC2865].

(3) A third party that issues **public key certificates**. **Certificates** serve to bind **public keys** to a user identity. Each user and **certification authority (CA)** may decide whether to trust another user or **CA** for a specific purpose, and whether this trust should be transitive. For more information, see [RFC3280].

Challenge-Handshake Authentication Protocol (CHAP): A protocol for user authentication to a remote resource. For more information, see [RFC1994] and [RFC2759].

change number: A number that identifies a version of a messaging object. A change number is identical in format to a message ID (MID) or folder ID (FID).

character set: (1) A mapping between the characters of a written language and the values that are used to represent those characters to a computer.

(2) The range of characters used to represent textual data within a **MIME body part**, as described in [RFC2046].

(3) A mapping of characters to their identifying code values. For more information, see [MSDN-CS].

checkpoint ICS state: An **Incremental Change Synchronization (ICS)** state that is provided by a server in the middle of an ICS operation, which reflects the state of the **local replica**, indicated by the **initial ICS state**, after applying all differences transmitted in the ICS operation.

checksum: A value that is the summation of a byte stream. By comparing the checksums computed from a data item at two different times, one can quickly assess whether the data items are identical.

child: (1) An object that is immediately below the current object in a hierarchy.

(2) A data item within the Master Data Services (MDS) system that has a superior data item. A child in MDS can be a leaf member or a consolidated member.

chunk: A sequence of words that are treated as a single unit by a module that checks spelling.

CIM class: A **CIM object** that represents a **CIM class** definition as a **CIM object**. It is the template representing a **manageable entity** with a set of **properties** and methods.

CIM instance: An instantiation of a **CIM class** representing a **manageable entity**.

CIM object: Refers to a **CIM class** or a **CIM instance**.

claim: (1) A set of operations that are performed on a workflow task to specify the user who owns it.

(2) A statement that one subject makes about itself or another subject. For example, the statement can be about a name, identity, key, group, privilege, or capability. Claims have a provider that issues them, and they are given one or more values. They are also defined by a claim value type and, possibly, associated metadata.

(3) An assertion about a security principal expressed as an n-tuple containing an {Identifier, ValueType and m-Values of type ValueType} where m > = 1. A claim with only 1 value in the n-tuple is called a **single-valued claim** and a claim with more than 1 value is called a **multi-valued claim**.

(4) A declaration made by an entity (for example, name, identity, key, group, privilege, and capability). For more information, see [WSFedPRP] sections 1.4 and 2.

class: (1) User-defined binary data that is associated with a key.

(2) A **Remoting Type** that encapsulates a set of named values and a set of methods that operate on those values. The named values are called Members of the Class. A Class is part of the **Remoting Data Model**. For more information, see [MS-NRTP] section 3.1.1.

(3) See **object class**.

(4) A reference to a class module whose methods and properties can be used within a report.

class factory: An object (3 or 4) whose purpose is to create objects (3 or 4) from a specific object class (3 or 4).

class identifier (CLSID): A **GUID** that identifies a software component; for instance, a DCOM **object class** (4) or a **COM class**.

classifier: A Unicode string used in conjunction with an authority to form a Peer Name.

clear-signed message: An Internet email message that is in the format described by [RFC1847] and is identified with the media type "multipart/signed", or the **Message object** representing such a message. An important class of clear-signed message, based on a "multipart/signed" format, is the S/MIME clear-signed message, as described in [RFC5751] and [RFC3852].

client: (1) A computer on which the remote procedure call (RPC) client is executing.

(2) An execution environment that holds object references and issues object RPC (ORPC) calls.

(3) In DFS-R, a replicating machine acts as a client when it receives replicated files from its upstream partner. Use of the terminology **client** stipulates that the machine contact its upstream server, and is responsible for initiating communication related to receiving replicated files. It does not imply anything about the operating system version or the function of the machine.

(4) The sending endpoint of a web services request message, and receiver of any resulting web services response message.

(5) For the Peer Content Caching and Retrieval Framework, a client is a client-role peer; that is, a peer that is searching for content, either from the server or from other peers or hosted cashes. In the context of the Retrieval Protocol, a client is a peer that requests a block-range from a server_role_peer. It acts as a Web Services Dynamic Discovery (WS-Discovery) [WS-Discovery] client.

(6) Synonym for **client computer** (4).

(7) In [MS-GPOL], the capitalized use of this term refers to a **domain** member, including the **domain controller (DC)**, that is involved in a **policy application** sequence.

(8) The entity that initiates the **HTTP** connection.

(9) A client device that is capable of issuing OMA-DM commands to a server and responding to OMA-DM commands issued by a server.

(10) Identifies the system that consumes WMI services and initiates DCOM ([MS-DCOM]) calls to WMI servers.

(11) The entity that has created the logging message, or an entity that receives a logging message from a client. In the latter case, the client is a proxy.

(12) The software that is used by a **user** to access the service. It represents the **user** in [MS-PASS]. A synonym is **client** application.

(13) Used as described in [RFC2616] section 1.3.

(14) The term "Client" that is defined in [WS-Discovery1.1].

(15) The client application using the **WS-Management** Protocol to access the management **service**, on the local or a remote machine.

(16) A client, also called a client computer, is a computer that receives and applies settings of a **Group Policy Object (GPO)**, as specified in [MS-GPOL].

(17) A user participating in or intending to participate in collaboration.

(18) The **target location** machine.

(19) The entity that initiates communication with the **hosted cache**, to offer it **segments** of data.

(20) An application or a system that accesses a Web service endpoint as defined in [WSAddressing].

(21) A **client** application that uses the WS-Management Protocol (see [DMTF-DSP0226]) to access the management **service** on a local or remote computer.

(22) A **domain** member that is involved in a **policy application** mode sequence.

(23) Any process that initiates **commands** for execution on a server by using the PowerShell Remoting Protocol.

Client Access License (CAL): A license that gives a user the right to access the services of a server. To legally access the server software, a CAL can be required. A CAL is not a software product.

client computer: (1) A computer that instigates a connection to a well-known port on a server.

(2) A computer that receives and applies settings from a **Group Policy Object (GPO)**, as specified in [MS-GPOL].

(3) A computer that gets its **updates** from an **update server**. A client can be a desktop computer, a server, or the **update server**. For more information, see [MS-WUSP] and [MS-WSUSSS].

(4) The client machine in the **domain** or network topology of clients, servers, and **domain controllers**. Alternatively, a computer that is not a **domain controller server**; the computer may or may not be joined to a domain.

client/server mode: A mode that consists of one server with many client connections (one-to-many). From the perspective of each client, there is only one connection: the connection to the server.

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

cluster: (1) A group of computers that are able to dynamically assign resource tasks among nodes in a group.

(2) A group of computers that are able to dynamically assign resource tasks among nodes in a group. The group of computers that can be accessed as though they are a single host. A **cluster** is generally accessed by using a virtual IP address. For more information, see [MSFT-WLBS].

(3) The smallest allocation unit on a **volume**.

cluster node: Cluster node defined in [MS-CMRP] section 1.3.

cluster resource group: Resource group defined in [MS-CMRP] section 1.1.

code page: An ordered set of characters of a specific script in which a numerical index (code-point value) is associated with each character. Code pages are a means of providing support for **character sets** (1) and keyboard layouts used in different countries. Devices such as the display and keyboard can be configured to use a specific code page and to switch from one code page (such as the United States) to another (such as Portugal) at the user's request.

codec: An algorithm that is used to convert media between digital formats, especially between raw media data and a format that is more suitable for a specific purpose. Encoding converts the raw data to a digital format. Decoding reverses the process.

collection: (1) A grouping of one or more EDM types that are type compatible. A collection can be used as the return type for a FunctionImport.

(2) A **resource** that contains a set of **URIs** that identify member **resources**. Use of this term is consistent with what is specified in [RFC4918] section 5.2.

(3) A user-defined group of data items from the same entity.

(4) An element that is used when a Function element is declared whose parameter or return type is not a single value but many. For example, a Function element may return a collection of varchar, that is, collection(varchar).

color flag: A flag that extends the concept of a **basic flag** by associating one of a chosen set of color values with a flagged **Message object**.

column: (1) See **field** (3).

(2) A single set of data that is displayed vertically in a worksheet or a table.

(3) See **column chart**.

(4) The container for a single type of information in a **row**. Columns map to property names and specify what properties are used for the search query's **command tree** elements.

column chart: A chart that displays data in vertical bars to facilitate data comparison.

COM class: An **object class** (3).

command: Any entity that can be executed on the server.

command tree: A combination of **restrictions** (1) and sort orders that are specified for a search query.

common byte stack: A list of arrays of bytes. Byte values of contained arrays, when together in their natural order, represent common high-order bytes of GLOBCNT values. Common byte stacks are used in a last-in first-out (LIFO) fashion during serialization or deserialization of GLOBSETs.

Common Information Model (CIM): The **Distributed Management Task Force (DMTF)** model that describes how to represent real-world computer and network objects. CIM uses an object-oriented paradigm, where managed objects are modeled using the concepts of classes and instances. See [DMTF-DSP0004].

Common Information Model (CIM) object: An object that represents a **Common Information Model (CIM)** object. This may be either a **CIM class** or a **CIM instance** of a **CIM class**.

common name (CN): A string attribute of a **certificate** (1) that is one component of a **distinguished name (DN)** (1). In Microsoft Enterprise uses, a CN must be unique within the forest where it is defined and any forests that share trust with the defining forest. The website or email address of the certificate owner is often used as a common name. Client applications often refer to a **certification authority (CA)** by the CN of its signing certificate.

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

component: A representation of a constituent **transport address** if a **candidate** consists of a set of transport addresses. For example, media streams that are based on the Real-Time Transfer Protocol (RTP) have two components, one for RTP and another for the Real-Time Transfer Control Protocol (RTCP).

component configuration entry: An entry in the catalog that represents a particular configuration of a component.

Component Object Model (COM): An object-oriented programming model that defines how objects interact within a single process or between processes. In **COM**, clients have access to an object through interfaces implemented on the object. For more information, see [MS-DCOM].

compound file: (1) A structure for storing a file system, similar to a simplified FAT file system inside a single file, by dividing the single file into sectors.

(2) A file that is created as defined in [MS-CFB] and that is capable of storing data that is structured as storage and streams.

computer object: An **object** of class computer. A **computer object** is a **security principal object**; the principal is the operating system running on the computer. The shared secret allows the operating system running on the computer to authenticate itself independently of any user running on the system. See **security principal**.

conceptual schema definition language (CSDL): A language that is based on XML and that can be used to define conceptual models that are based on the **Entity Data Model (EDM)**. For more information, see [MC-CSDL].

conceptual schema definition language (CSDL) document: A document that contains a conceptual model that is described by using the **CSDL** code. For more information, see [MC-CSDL].

concrete type: A type used in this specification whose representation must be standardized for interoperability. Specific cases include types in the **IDL** definition of an **RPC** interface, types sent over **RPC** but whose representation is unknown to **RPC**, and types stored as byte strings in **directory attributes**.

condition: (1) A logical expression comparing one or more properties in all incoming **Message objects** against a set of clauses. This logical expression can evaluate to TRUE or FALSE.

(2) A **condition** of a **policy** that specifies one of the fields in a **DHCP Client** request and the value that the field should contain to match the **condition**. The **condition** also contains an index that identifies the expression with which the **condition** is associated.

(3) A predicate (for example, the machine is idle) that must be satisfied for a **task** to run. A **task** runs when any of its **triggers** and all of its conditions are met.

(4) A method of controlling which **RAP** is selected as the **current resource policy**. Conditions are rules that are automatically triggered in response to notifications of any of the **conditional events**. A condition is composed of a **condition state** and **RAP**. When a **conditional event** is triggered, conditions with the associated Name attribute value are evaluated in the order of their ID attribute value; that is, a condition with the ID value 0 will be evaluated first and so on. In condition evaluation, the **condition state** is evaluated and if it is found to be TRUE, the **RAP** associated with that condition is selected as the **current resource policy**. If no condition has its **condition state** as TRUE, the condition with the name ANY is evaluated.

(5) A business rule argument that determines when to apply the actions of the business rule. Conditions can be parsed together by using the logical operators AND and OR.

condition state: A part of a **condition** consisting of a predicate that evaluates some current state of the computer being managed. The predicate is a series of expressions separated by

AND and OR operators, evaluated in order. Expressions are selected from the following fixed set: an equality or inequality test of the amount of hardware memory, an equality or inequality test of the number of processors, or a predicate test of the online or offline status of a **cluster node** or **cluster resource group**.

conditional events: Unscheduled events that can trigger the following WSRM policy changes: Processor hot add, Memory hot add, Cluster node goes up or down, or Cluster resource group goes online or offline.

conference: (1) A **Real-Time Transport Protocol (RTP)** session that includes more than one **participant** (2).

(2) An **RTP session** involving multiple **participants**.

(3) A set of two or more communicating users along with the software they are using to communicate.

configuration naming context (config NC): A **naming context (NC)** that contains configuration information. In **Active Directory**, a single **config NC** is shared among all **domain controllers (DCs)** in the forest. A **config NC** cannot contain **security principal objects**.

conglomeration: (1) A collection of **component configuration entries**, together with a component-independent configuration that is conceptually shared by the **component configuration entries**. A conglomeration is identified by a **conglomeration identifier**.

(2) A collection of **event classes** and **subscriptions** together with independent configuration data that is conceptually shared by the both the **event classes** and **subscriptions**. A conglomeration is identified by a **conglomeration identifier**.

conglomeration identifier: A **GUID** that identifies a **conglomeration**.

connection: (1) A link between two devices that uses the **Simple Symmetric Transport Protocol (SSTP)**. Each connection can support one or more SSTP sessions.

(2) A link that two physical machines or applications share to pass data back and forth.

(3) Each user that has a session with a server can create multiple share connections, or resource connections, using that user ID. This resource connection is created using a tree connect **Server Message Block (SMB)** and is identified by an **SMB** TreeID or TID.

(4) Firewall rules are specified to apply to connections. Every packet is associated with a connection based on TCP, UDP, or IP endpoint parameters; see [IANAPORT].

(5) In DFS-R, a pair of client and server replication partners.

(6) In OleTx, an ordered set of logically related messages. The relationship between the messages is defined by the higher-layer protocol, but they are guaranteed to be delivered exactly one time and in order relative to other messages in the connection.

(7) Either a **TCP** or **NetBIOS** over TCP connection between an SMB 2 Protocol client and an SMB 2 Protocol server.

(8) A time-bounded association between two **endpoints** that allows the two **endpoints** to exchange messages.

(9) A logical communication path identified by a pair of sockets, as defined in [RFC793].

(10) An instantiation of the protocol that can be used as a scoping entity for channel. The server may instantiate multiple simultaneous connections to the same client.

(11) The successful completion of necessary protocol arrangements (authentication, network parameters negotiation, and so on) between a remote client computer and the RRAS server to set up a dial-up or **virtual private networking (VPN)** association. **Connection** enables the remote client computer to function on the RRAS server network as if it were connected to the server network directly.

connection-oriented NTLM: A particular variant of **NTLM** designed to be used with connection-oriented **remote procedure call (RPC)**, as described in [MS-NLMP].

consolidated to-do list: A list of all tasks and flagged **Message objects** that are in a user's **mailbox**.

constructed attribute: (1) An attribute whose values are computed from normal attributes (for read) and/or have effects on the values of normal attributes (for write).

(2) See [MS-ADTS] section 3.1.1.1.4.

contact: (1) A presence entity (presentity) whose presence information can be tracked.

(2) An object of the contact class that represents a company or person whom a user can contact.

(3) A person, company, or other entity that is stored in a directory and is associated with one or more unique identifiers and **attributes** (2), such as an Internet message address or login name.

(4) A **node** that publishes a **contact record**. **Contacts** are used by **graph maintenance** to detect partitions.

contact attachment: An attached **message** item that has a message type of "IPM.Contact" and adheres to the definition of a **Contact object**.

contact identifier: A **universally unique identifier (UUID)** that identifies a partner in the MSDTC Connection Manager: OleTx Transports Protocol. These **UUIDs** are frequently converted to and from string representations. This string representation must follow the format specified in [C706] Appendix A. In addition, the **UUIDs** must be compared, as specified in C706-AppendixAUUID.

Contact object: A **Message object** that contains properties pertaining to a **contact** (3).

contact record: A **record** published by a **contact** that includes the **contact's** address and the **graph signature** at the time of publication.

Contacts folder: A **Folder object** that contains **Contact objects**.

Container class: The value of the PidTagContainerClass property on a folder, which indicates the default **Message object** type for the folder.

content: (1) Multimedia data. **content** is always in **ASF**, for example, a single **ASF** music file or a single **ASF** video file. Data in general. A file that an application accesses. Examples of content include web pages and documents stored on either web servers or SMB file servers.

(2) Items that correspond to a file that an application attempts to access. Examples of **content** include web pages and documents stored on either HTTP servers or SMB file servers. Each **content** item consists of an ordered collection of one or more **segments**.

(3) A package that contains all the associated files for an **update** that is to be installed on a **client computer**.

(4) Identified by a unique name under a given multicast namespace. The **content metadata** cannot change during the lifetime of a multicast session, and is required to allow random access to the data.

content database: A database that is stored on a **back-end database server** and contains stored procedures, site collections, and the contents of those site collections.

Content Metadata: Specifies an opaque binary data that is associated with the content.

content synchronization: The process of keeping synchronized versions of **Message objects** and their properties on a client and server.

contents table: A **Table object** whose rows represent the **Message objects** that are contained in a **Folder object**.

control level: The permissions that are granted to a **participant** in a **shared** desktop. The **control levels** include "view" (the **participant** is able to see, but not interact with, **shared** content), "full" (the **participant** is able to both see and interact with **shared** content), and "none" (the **participant** can neither see nor interact with **shared** content).

conversation: (1) A single representation of a send/response series of email messages. A conversation appears in the Inbox as one unit and allows the user to view and read the series of related email messages in a single effort.

(2) In **LU 6.2**, **conversations** connect transaction programs, and are used by the transaction programs to transfer messages. For a more complete definition, see [LU62Peer].

conversation action: A limited set of actions that a user applies to all **Message objects** that have the same PidTagConversationId value. The action is applied to all Message objects that are currently in the store or are delivered in the future.

conversation ID: A unique value that is associated with a conversation. It is assigned to each **Message object** that is part of a conversation and it is used to identify the conversation to which the message belongs.

conversation index: A value that specifies the location of a message within a conversation. A client can use this value to identify the parent and child messages of a message, and then generate a tree view of the conversation that contains those messages.

cookie: (1) A small data file that is stored on a user's computer and carries state information between participating protocol servers and protocol clients.

(2) A randomly generated, 16-byte sequence that is used to authenticate the client to the server during the creation of a multitransport connection.

(3) An HTTP header that carries state information between participating origin servers and user agents. For more information, see [RFC2109].

coordinate space: A space based on Cartesian coordinates, which provides a means of specifying the location of each point in the space. A two-dimensional coordinate space requires two axes that are perpendicular and equal in length. Three two-dimensional coordinate spaces are generally used to describe an output surface: world, **page**, and **device**. To scale device-independent output for a particular physical device, a rectangular area in the world or **page** coordinate space is mapped into the **device** coordinate space using a **transform**

Coordinated Universal Time (UTC): A high-precision atomic time standard that approximately tracks Universal Time (UT). It is the basis for legal, civil time all over the Earth. Time zones around the world are expressed as positive and negative offsets from UTC. In this role, it is also referred to as Zulu time (Z) and Greenwich Mean Time (GMT). In these specifications, all references to UTC refer to the time at UTC–0 (or GMT).

counter proposal: A request that an attendee sends to an **organizer** when requesting a change to the date or time of a meeting.

cryptographic hash function: A function that maps an input of any length to a short output bit string of fixed length, such that finding an input that maps to a particular bit string of the correct output length, or even finding two inputs that map to the same output bit string, is computationally infeasible. For more information, see [SCHNEIER] chapters 2 and 18.

current resource policy: While in the running **management state**, the **management service** always selects exactly one **RAP** to be the **current resource policy**.

cyclic redundancy check (CRC): 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, a **cryptographic hash function** should be used instead.

6 D

data region: A region of a table that encompasses the range of cells that contains the table records. A data region does not include the **header row** (1), insert row, or total row of a table.

data source: (1) A database, web service, disk, file, or other collection of information from which data is queried or submitted. Supported data sources vary based on application and data provider.

(2) A collection of fields and groups that define and store the data for an InfoPath form. Controls in a form are bound to the fields and groups in the data sources of the form. See also **main data source** and **secondary data source**.

(3) A specified data source type, connection string, and credentials, which can be saved separately to a report server and shared among report projects or embedded in a report definition (.rdl) file.

(4) A physical data source.

data value: An instance of a **Remoting Type**, which may be a **Class**, **Array**, **Enum**, or Primitive. A **Data Value** is part of the **Remoting Data Model**. For more information, see [MS-NRTP] section 3.1.1.

database: (1) For the purposes of the Netlogon RPC, a database is a collection of user accounts, machine accounts, aliases, groups, and policies, managed by a component. The database, or the component managing the database, must expose a mechanism to enable Netlogon to gather changes from and apply changes to the database. Additionally, it must export a database serial number in order to track changes for efficient replication.

(2) In **Distributed File System Replication (DFS-R)**, the database maintained by the Microsoft implementation of **DFS-R** maintains the local version chain vector and one record for each resource that is tracked, including **tombstones** for deleted resources, such that deletion of files can be propagated in a timely fashion.

(3) The set of all non-expired **records** published in a **graph**.

database object: (1) An object such as a table, query, form, report, macro, or module that can be referenced by name in a database, database application, or database project.

(2) A representation of a named set of attribute value pairs that a protocol exposes.

DataClass: A type of **MetadataObject** that represents a type of a business data object obtained from a **line-of-business (LOB) system**. Instances of a DataClass have transient identity. DataClasses are contained by **LobSystems** and Methods.

datagram: A style of communication offered by a network transport protocol where each message is contained within a single network packet. In this style, there is no requirement for establishing a session prior to communication, as opposed to a connection-oriented style.

dataset: (1) A set of multidimensional data that is returned when a multidimensional expression (MDX) SELECT statement is executed. A dataset represents a slice of a cube as defined by the members and axes that are specified in the query.

(2) A named specification that includes a data source definition, a query definition, and optional parameter values, calculated fields, and filtering and collation information as part of a report definition (.rdl) file. An .rdl file can have multiple datasets.

DCOM: See **Distributed Component Object Model (DCOM)**. Can also refer to the Distributed Component Object Model (DCOM) Remote Protocol Specification [MS-DCOM].

declared property: A property that is statically declared by a Property element as part of the definition of a structural type. For example, in the context of an EntityType, a declared property includes all properties of an EntityType that are represented by the Property child elements of the EntityType element that defines the EntityType.

decryption: In cryptography, the process of transforming encrypted information to its original clear text form.

de-encapsulating RTF reader: A **Rich Text Format (RTF)** reader, as described in [MSFT-RTF], that recognizes if an input RTF document contains encapsulated HTML or plain text, and extracts and renders the original **HTML** or plain text instead of the encapsulating RTF content.

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

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

delegate: A user or resource that has permissions to act on behalf of another user or resource.

delegate access: The access that is granted by a delegator to a delegate and is used by the delegate to access the delegator's account.

delegate data folder: A **special folder** that contains the Delegate Information object.

Delegate Information object: A **Message object** that contains properties specifying delegate access settings for resources in a delegator's mailbox.

delegate rule: A **server-side rule** that is used to send mail to delegates on behalf of a delegator.

delegator: A user or resource for which another user or resource has permission to act on its behalf.

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

deleted-object: An **object** that has been deleted, but remains in storage until a configured amount of time (the **deleted-object lifetime**) has passed, after which the **object** is transformed to a **recycled-object**. Unlike a **recycled-object** or a **tombstone**, a **deleted-object** maintains virtually all the state of the **object** before deletion, and may be undeleted without loss of information. **Deleted-objects** exist only when the **Recycle Bin optional feature** is enabled.

deleted-object lifetime: The time period that a **deleted-object** is kept in storage before it is transformed into a **recycled-object**.

delivery receipt: A report message that is generated and sent by a client or server to the sender of a message or another designated recipient when an email message is received by an intended recipient.

delivery status notification (DSN): (1) A message that reports the result of an attempt to deliver a message to one or more recipients, as described in [RFC3464].

(2) A DSN is an SMTP message that describes the progress of delivery of another SMTP message. The SMTP MTA sends a DSN message to the sender when delivery is delayed or obstructed.

delta: (1) A unit of transactional consistency in a **shared space**. A delta can contain one or more commands.

(2) One of a set of possible changes that can be made to a **database**.

Department object: An **Address Book object** that describes a department within an organization.

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

device: (1) A client or server computer that uses a **device URL** to identify itself as an **endpoint** (5) for synchronizing account data.

(2) Any peripheral or part of a computer system that can send or receive data.

(3) The Devices Profile for Web Services (DPWS) term for a special instance of a service that is discoverable and contains other services with metadata describing those services.

(4) A logical device and/or a container that may embed other logical devices and that embeds one or more services and advertises its presence on network(s). For more information, see [UPNPARCH1.1] sections 1 and 2.

(5) A device can be any UPnP-enabled device.

device space: The output space for graphics **transforms**. It usually refers to the client area of an application window; however, it can also include the entire desktop, a complete window, or a page of printer or plotter paper. Physical device space dimensions vary according to the dimensions set by the display, printer, or plotter technology.

device URL: A unique identifier for a client device, as described in [RFC3986].

DFS-R: A service that keeps **DFS** and **SYSVOL** folders in sync automatically. DFS-R is a state-based, multimaster replication system that supports replication scheduling and bandwidth throttling. This is a rewrite and new version of **FRS**. For more information, see [MS-FRS2].

DHCP client: The **remote procedure call (RPC) clients** that use the Dynamic Host Configuration Protocol Server Management Protocol (DHCPM) to configure, manage, and monitor the Dynamic Host Configuration Protocol (DHCP) **server**.

dictionary: A collection of key/value pairs. Each pair consists of a unique key and an associated value. Values in the dictionary are retrieved by providing a key for which the dictionary returns the associated value.

digital certificate: See the "digital certificate definition standard," as described in [X509].

directory: (1) The database that stores information about objects such as users, groups, computers, printers, and the **directory service** that makes this information available to users and applications.

(2) A **forest**.

directory object: (1) A **Lightweight Directory Access Protocol (LDAP)** object, as described in [RFC2251], that is a specialization of an object.

(2) A **Lightweight Directory Access Protocol (LDAP) object**, as specified in [RFC2251], that is a specialization of an **object**.

(3) An **Active Directory object**, which is a specialization of the "object" concept that is described in [MS-ADTS] section 1 or [MS-DRSR] section 1, Introduction, under Pervasive Concepts. An **Active Directory object** can be identified by the objectGUID **attribute** of a **dsname** according to the matching rules defined in [MS-DRSR] section 5.50, DSNAME. The **parent**-identifying **attribute** (not exposed as an **LDAP attribute**) is parent. **Active Directory objects** are similar to **LDAP** entries, as defined in [RFC2251]; the differences are specified in [MS-ADTS] section 3.1.1.3.1.

directory partition: A synonym for **Active Directory partition** and naming context (NC) replica.

directory server: A persistent storage for DNS **zones** and records. A DNS server can access DNS data stored in a **directory server** using the **LDAP** protocol or a similar directory access mechanism.

directory service (DS): (1) A service that stores and organizes information about a computer network's users and network shares, and that allows network administrators to manage users' access to the shares. See also **Active Directory**.

(2) An entity that maintains a collection of objects. These objects can be remotely manipulated either by the Message Queuing (MSMQ): Directory Service Protocol, as specified in [MS-MQDS], or by the Lightweight Directory Access Protocol (v3), as specified in [RFC2251].

(3) A distributed data storage system that allows computers connected to a network to store, edit, and retrieve information.

DirectPlay: A network communication library included with the Microsoft **DirectX** application programming interfaces. **DirectPlay** is a high-level software interface between applications and communication services that makes it easy to connect games over the Internet, a modem link, or a network.

DirectPlay 4: A programming library that implements the IDirectPlay4 programming interface. **DirectPlay 4** provides peer-to-peer session-layer services to applications, including session lifetime management, data management, and media abstraction. **DirectPlay 4** first shipped with the DirectX 6 multimedia toolkit. Later versions continued to ship up to, and including, DirectX 9. **DirectPlay 4** was subsequently deprecated. The **DirectPlay 4** DLL continues to ship in current versions of Windows operating systems, but the development library is no longer shipping in Microsoft development tools and software development kits (SDKs).

DirectPlay 8: A programming library that implements the IDirectPlay8 programming interface. **DirectPlay 8** provides peer-to-peer session-layer services to applications, including session lifetime management, data management, and media abstraction. **DirectPlay 8** first shipped with the DirectX 8 software development toolkit. Later versions continued to ship up to, and including, DirectX 9. **DirectPlay 8** was subsequently deprecated. The **DirectPlay 8** DLL continues to ship in current versions of Windows operating systems, but the development library is no longer shipping in Microsoft development tools and Software Development Kits (SDKs).

DirectX: Microsoft **DirectX** is a collection of application programming interfaces for handling tasks related to multimedia, especially game programming and video, on Microsoft platforms.

DirectX Diagnostic (DXDiag): DXDiag.exe is an application that uses the DirectPlay DXDiag Usage Protocol [MS-DPDX] traffic.

discretionary access control list (DACL): An **access control list (ACL)** that is controlled by the owner of an object and that specifies the access particular users or groups can have to the object.

disk encapsulation: The process of converting a **basic disk** to a **dynamic disk**. Encapsulating a disk lays down disk metadata that is used for managing the disk dynamically.

dismiss: A process that disables an **overdue reminder**. After a reminder is dismissed, it is not considered overdue anymore and is not signaled or displayed to a user or any agents who are acting on behalf of that user.

display name: A text string that is used to identify a principal or other object in the user interface. Also referred to as title.

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

Distinguished Encoding Rules (DER): A method for encoding a data object based on Basic Encoding Rules (BER) encoding but with additional constraints. DER is used to encode X.509 **certificates** (2) that need to be digitally signed or to have their signatures verified.

distinguished name (DN): (1) A name that uniquely identifies an object by using the **relative distinguished name (RDN)** for the object, and the names of container objects and domains that contain the object. The distinguished name (DN) identifies the object and its location in a tree.

(2) In the **Active Directory** directory service, the unique identifier of an object in **Active Directory**, as described in [MS-ADTS] and [RFC2251].

(3) In X.500, the globally unique name string that identifies an entity in an X.500 directory, as described in [X500]. The DN consists of several components and is used in X.509 **certificates** (2) to identify the subject and issuer principals, as described in [X509].

(4) In **Lightweight Directory Access Protocol (LDAP)**, an **LDAP Distinguished Name**, as described in [RFC2251] section 4.1.3. The DN of an object is the DN of its parent, preceded by the RDN of the object. For example: CN=David Thompson, OU=Users, DC=Microsoft, DC=COM. For definitions of CN and OU, see [RFC2256] sections 5.4 and 5.12, respectively.

(5) A name that uniquely identifies an object by using the **relative distinguished name (RDN)** for the object, plus the names of container objects and domains that contain the object. The **DN** identifies the object as well as its location in a tree.

(6) In the **Active Directory** directory service, the unique identifier of an object in **Active Directory**, as specified in [MS-ADTS] and [RFC2251].

(7) In X.500, the globally unique name string that identifies an entity in an X.500 directory, as specified in [X500]. The **DN** consists of several components and is used in X.509 certificates to identify the subject and issuer principals, as specified in [X509].

(8) In **Lightweight Directory Access Protocol (LDAP)**, an **LDAP DN**, as specified in [RFC2251] section 4.1.3. The **DN** of an object is the **DN** of its parent, preceded by the **RDN** of the object. For example: CN=David Thompson,OU=Users,DC=Microsoft,DC=COM. For definitions of CN and OU, see [RFC2256] sections 5.4 and 5.12, respectively.

Distributed Component Object Model (DCOM): The Microsoft Component Object Model (COM) specification that defines how components communicate over networks, as specified in [MS-DCOM].

Distributed File System (DFS): A file system that logically groups physical shared folders located on different servers by transparently connecting them to one or more hierarchical namespaces. **DFS** also provides fault-tolerance and load-sharing capabilities. **DFS** refers to the Microsoft DFS available in Windows Server platforms.

Distributed File System Replication (DFS-R): A service that keeps **DFS** folders in sync automatically. **DFS-R** is a state-based, multi-master replication system that supports replication scheduling and bandwidth throttling. This is a rewrite and new version of the **File Replication Service (FRS)**. For more information, see [MS-FRS2].

Distributed Management Task Force (DMTF): An industry organization that develops management standards and integration technology for enterprise and Internet environments.

distribution list: (1) A collection of users, computers, contacts, or other groups that is used only for email distribution, and addressed as a single recipient.

(2) An **Active Directory** object that can contain explicit references only to destinations published in **Active Directory**; that is, to **public queues**, queue aliases, and other distribution lists, but not to private and URL-named queues.

Distribution List object: A **Message object** that contains properties that describe a **distribution list**.

Document object: A **Message object** that represents a single file, such as a document generated by a word-processing application. The Message object contains the file as an **Attachment object** and includes additional properties to describe the file.

domain: (1) A set of users and computers sharing a common namespace and management infrastructure. At least one computer member of the set must act as a **domain controller (DC)** and host a member list that identifies all members of the domain, as well as optionally hosting the **Active Directory** service. The domain controller provides **authentication** (2) of members, creating a unit of trust for its members. Each domain has an identifier that is shared among its members.

(2) A set of users and computers sharing a common namespace and management infrastructure. At least one computer member of the set must act as a **domain controller** and host a member list that identifies all members of the domain, as well as optionally hosting the **Active Directory** service. The domain controller provides authentication of members, creating a unit of trust for its members. Each domain has an identifier that is shared among its members. For more information, see [MS-AUTHSOD] section 1.1.1.5 and [MS-ADTS].

(3) A capture of the data semantics. Example domains include email address, gender, and state.

domain client in a workstation role: A domain member that offers other services to other domain clients.

domain controller (DC): The service, running on a server, that implements **Active Directory**, or the server hosting this service. The service hosts the data store for **objects** and interoperates with other **DCs** to ensure that a local change to an **object** replicates correctly across all **DCs**. When **Active Directory** is operating as **Active Directory Domain Services (AD DS)**, the **DC** contains full NC replicas of the **configuration naming context (config NC)**, **schema naming context (schema NC)**, and one of the **domain NCs** in its **forest**. If

the **AD DS** **DC** is a **global catalog server (GC server)**, it contains partial NC replicas of the remaining **domain NCs** in its **forest**. For more information, see [MS-AUTHSOD] section 1.1.1.5.2. When **Active Directory** is operating as **Active Directory Lightweight Directory Services (AD LDS)**, several **AD LDS** **DCs** can run on one server. When **Active Directory** is operating as **AD DS**, only one **AD DS** **DC** can run on one server. However, several **AD LDS** **DCs** can coexist with one **AD DS** **DC** on one server. The **AD LDS** **DC** contains full NC replicas of the **config NC** and the **schema NC** in its **forest**.

domain controller server: A domain member, which can be a client or a server that offers other services to its clients. When the domain client acts as a supplicant to another domain client, the supplicant is referred to as a **domain client in a workstation role** and the latter as a domain client in a server role.

domain name: (1) The name given by an administrator to a collection of networked computers that share a common directory. Part of the domain naming service naming structure, domain names consist of a sequence of name labels separated by periods.

(2) A name with a structure indicated by dots.

(3) A **domain name** (2) used by the **Domain Name System (DNS)**.

(4) A **domain name** (3) or a **NetBIOS name** that identifies a **domain**.

Domain Name System (DNS): A hierarchical, distributed database that contains mappings of **domain names** (1) 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.

domain naming context (domain NC): (1) A partition of the directory that contains information about the domain and is replicated with other **domain controllers (DCs)** in the same domain.

(2) A **naming context (NC)** whose replicas are able to contain **security principal** objects. No other **NC replica** can contain **security principal** objects. The **distinguished name (DN)** of a **domain NC** takes the form "dc=n1,dc=n2, ... dc=nk" where each ni satisfies the syntactic requirements of a DNS name component. For more information, see [RFC1034]. Such a **DN** corresponds to the **domain naming service name**: "n1.n2. ... .nk". This is the **domain naming service name** of the **domain NC**. **Domain NCs** appear in the **global catalog (GC)**. A **forest** has one or more **domain NCs**. The root of a **domain NC** is an object of class domainDns.

(3) A specific type of **naming context (NC)** that represents a **domain**. A **domain NC** can contain **security principal objects**; no other type of **NC** can contain **security principal objects**. **Domain NCs** appear in the **global catalog (GC)**. A **domain NC** is hosted by one or more **domain controllers (DCs)** operating as **AD DS**. In **AD DS**, a **forest** has one or more **domain NCs**. The root of a **domain NC** is an **object of class** domainDNS; for directory replication [MS-DRSR], see **domainDNS**. A **domain NC** cannot exist in **AD LDS**.

domain naming service name: The **fully qualified domain name (FQDN)** as known by the domain name system (DNS), as specified in [RFC1035] and [RFC1123].

domainDNS: A specific **object class**. The root of a **domain NC** or an **application NC** is an **object of class** domainDNS. The **DN** of such an **object** takes the form dc=n1,dc=n2, ... dc=nk, where each ni satisfies the syntactic requirements of a **fully qualified domain name (FQDN)** component (for more information, see [RFC1034]). Such a **DN** corresponds to the **FQDN** n1.n2. ... .nk. This is the **FQDN** of the **NC**, and it allows **replicas** of the **NC** to be located by using DNS.

double-byte character set (DBCS): A **character set** (1) that can use more than one byte to represent a single character. A DBCS includes some characters that consist of 1 byte and some characters that consist of 2 bytes. Languages such as Chinese, Japanese, and Korean use DBCS.

Draft Message object: A **Message object** that has not been sent.

Drafts folder: A **special folder** that is the default location for **Message objects** that have been saved but not sent.

drive: (1) See **volume**.

(2) A device that can read or write to a **cartridge**. A **library** has at least one **drive**.

dsname: (1) A tuple that contains between one and three identifiers for an object. The term **dsname** does not stand for anything. The possible identifiers are the object's **GUID** (attribute objectGuid), **security identifier (SID)** (attribute objectSid), and **distinguished name (DN)** (attribute distinguishedName). A **dsname** can appear in a protocol message and as an attribute value (for example, a value of an attribute with syntax Object(DS-DN)). Given a **DSName**, an **object** can be identified within a set of **NC replicas** according to the matching rules defined in [MS-DRSR] section 5.49.

(2) A **dsname** is a field 3-tuple<guid: **GUID**, sid: **security identifier (SID)**, dn: **distinguished name (DN)**>. A **dsname** can appear in a protocol message and as a value of an attribute. In either context, it identifies an object. If all three fields are null, the **dsname** is null. As a value of an attribute, a **dsname** always contains a non-null **GUID** and **DN**, and sometimes contains a non-null **SID**. Such a **dsname** n refers to the unique object o such that o.objectGuid = n.guid. The **SID** and **DN** are not used for identification in this case. As a value within a protocol message, a non-null **dsname** n refers to: if n.guid ≠ null, the unique object o such that o.objectGuid = n.guid (failing if no such object); otherwise if n.dn ≠ null, the unique object o such that o.distinguishedName = n.dn (failing if no such object); otherwise the unique object o such that o.objectSid = n.sid. Note that the **SID** is used only if no other part of the dsname is specified. If o is an object, the function dsname(o) equals [o.objectGuid, o.objectSid, o.distinguishedName].

dynamic disk: A disk on which volumes may be composed of more than one partition on disks of the same pack, as opposed to basic disks where a partition and a volume are equivalent.

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

Dynamic Host Configuration Protocol (DHCP) client: An Internet host using DHCP to obtain configuration parameters such as network addresses.

dynamic object: An object with a time-to-die (attribute msDS-Entry-Time-To-Die). The directory service garbage-collects a **dynamic object** immediately after its time-to-die has passed. The constructed attribute entryTTL gives a **dynamic object's** current time-to-live, that is, the difference between the current time and msDS-Entry-Time-To-Die. For more information, see [RFC2589].

dynamic property: A designation for an instance of an OpenEntityType that includes additional nullable properties (of a scalar type or ComplexType) beyond its **declared properties**. The set of additional properties, and the type of each, may vary between instances of the same OpenEntityType. Such additional properties are referred to as dynamic properties and do not have a representation in a **CSDL document**.

dynamic web template: An HTML-based master copy of a page that contains settings, formatting, and elements such as text, graphics, page layout, styles, and regions of a page that can be modified. Dynamic web templates have a .dwt file name extension.

7 E

email address: A string that identifies a user and enables the user to receive Internet messages.

email alias: A string which is the local-part of a mailbox as specified in [RFC2821].

email enabled list: A SharePoint list that is configured to accept incoming email messages.

Email object: A **Message object** that represents an email message in a message store and adheres to the property descriptions that are described in in [MS-OXOMSG].

Email Text Body: The textual portion of a message that is displayed, by convention, by industry standard email clients. The Internet mail format, as described in [RFC822], allowed only text messages to be transmitted. The concept of transmitting content other than text was not codified until **MIME** was standardized. Handling of entities other than the textual portion of a message, such as attachments, varies by implementation in email clients.

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

embedded object: (1) An object that is created by using one application and is hosted in a document that was created by using another application. Embedding an object, rather than inserting or pasting it, ensures that the object retains its original format. Users can double-click an embedded object and edit it with the toolbars and menus from the application that was used to create it. See also **Object Linking and Embedding (OLE)**.

(2) Application data that is stored in documents from other applications.

encapsulating RTF writer: A **Rich Text Format (RTF)** writer, as described in [MSFT-RTF], that produces an RTF document as a result of format conversion from other formats, such as plain text or HTML, and also stores the original document in a form that allows for subsequent retrieval.

encapsulation: (1) A process of encoding one document in another document in a way that allows the first document to be re-created in a form that is nearly identical to its original form.

(2) See **disk encapsulation**.

enclosure: An **XML element** that is in a feed and contains information such as a **URL** for a file, typically a media file, that is associated with an **RSS** item or **Atom** entry, for example, a podcast.

encoding: (1) A process that specifies a Content-Transfer-Encoding for transforming character data from one form to another.

(2) The binary layout that is used to represent a **Common Information Model (CIM) object**, whether a **CIM class** or **CIM instance** definition. The **encoding** is what is actually transferred by the protocol.

(3) The annotation of an object with metadata so that it can be sent to a client or server.

encrypted message: An Internet email message that is in the format described by [RFC5751] and uses the EnvelopedData CMS content type described in [RFC3852], or the **Message object** that represents such a message.

encryption: In cryptography, the process of obscuring information to make it unreadable without special knowledge.

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

(2) A network-specific address of a remote procedure call (RPC) server process for remote procedure calls. The actual name and type of the endpoint depends on the RPC protocol sequence that is 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 (RPC Protocol Sequence ncacn_np), an endpoint might be the name of a named pipe. For more information, see [C706].

(3) A participant that uses the Microsoft Groove Dynamics Protocol, as described in [MS-GRVDYNM], to synchronize with a shared space. An endpoint is identified by the combination of an identity URL and a client device URL. Each endpoint maintains a copy of the data in a shared space.

(4) A communication port that is exposed by an **application server** for a specific shared service and to which messages can be addressed.

(5) A device that is connected to a computer network.

(6) A **client** on the network that is requesting access to a **network access server (NAS)**.

(7) A network-specific address of a remote procedure call (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 [C706].

(8) In the context of a web service, a network target to which a **SOAP** message can be addressed. See [WSADDR].

(9) An entity, processor, or resource that can be referenced where Web service messages are originated or targeted.

(10) A node that sends or receives a **protocol stream**.

(11) A tuple (composed of an IP address, port, and protocol number) that uniquely identifies a communication **endpoint**.

(12) The IP address of a network interface on which the Dynamic Host Configuration Protocol (DHCP) **server** is listening for **DHCP client** requests.

(13) A **resource** that can be addressed by an **endpoint reference**.

endpoint reference (EPR): (1) A resource that conveys the information that is needed to address an **endpoint**.

(2) As specified in section 2 of [WSA].

(3) A combination of WS-Addressing ([WSAddressing]) and **WS-Management**–addressing elements that together describe an address for a **resource** in the **SOAP** message header.

enterprise/site/server distinguished name (ESSDN): An **X500 DN** that identifies an entry in an abstract naming scheme that is separate from an **address book**. The naming scheme

defines enterprises, which contain sites, and sites contain servers and users. There is no concrete data structure that embodies an ESSDN. Instead, an address book entry can contain an ESSDN as a property of the entry.

entity: (1) An instance of an EntityType element that has a unique identity and an independent existence. An entity is an operational unit of consistency.

(2) The payload of a transfer (by analogy to the definition in [RFC2616]).

(3) Any document on a server that is accessible by using a **Hypertext Transfer Protocol (HTTP)** URL.

(4) A unit that is part of the system such as a component or an element.

(5) A single business object about which data can be stored. It is the subject of a table in a relational database.

(6) Tabular data that is stored within the Master Data Services (MSD) system.

Entity: A type of **DataClass** that represents a type of business data object that is stored in a line-of-business (LOB) system and whose instances have a persistent EntityInstanceId.

Entity Data Model (EDM): A set of concepts that describes the structure of data, regardless of its stored form, as described in [MC-CSDL].

entry ID: See **EntryID**.

EntryID: A sequence of bytes that is used to identify and access an object.

Enum: A Primitive type whose members are constrained to a set of values. The Primitive type is considered to be an underlying **Remoting Type** of the **Enum**. Each value has a name associated with it. An **Enum** is part of the **Remoting Data Model**, and an abbreviation for "enumeration." For more information, see [MS-NRTP] section 3.1.1.

enumerator: A **station** that seeks all LLTD–capable **stations** on the link by using **quick discovery**.

event: (1) Any significant occurrence in a system or an application that requires users to be notified or an entry to be added to a log.

(2) An action or occurrence to which an application might respond. Examples include state changes, data transfers, key presses, and mouse movements.

(3) A discrete unit of historical data that an application exposes that may be relevant to other applications. An example of an event would be a particular user logging on to the computer.

(4) As defined in [UPNPARCH1.1] section i.7, a notification of one or more changes in state variables exposed by a **service**.

event class: (1) A collection of events that are grouped together based on criteria that the publishing application specifies.

(2) A collection of historical data grouped together using criteria specified by the publishing application.

event log: A collection of records, each of which corresponds to an event.

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 object.

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

exclusion list: (1) A list of items to exclude from query results and to remove from a search index the next time that a crawl occurs.

(2) A list of processes that cannot be managed because of the negative system impact such management could create. Processes in an exclusion list are unmanaged and can consume resources freely. Both system-defined and user-defined exclusion lists are defined.

export: The process of creating an **installer package file** for a **conglomeration** or **partition** on a COMA server, so that it can be **imported** onto another server.

expression: (1) A combination of operators, symbols, constants, literal values, functions, names of fields or **columns** (2), controls, and properties that evaluates to a single value.

(2) A construct that serves two purposes: specifies the logical operator (AND/OR) to be used between 2 conditions of a **policy**; and specifies the index of the **expressions** that are parent to it. Taken together, **conditions** and **expressions** specify **policy** classification criteria.

(3) A combination of symbols (identifiers, literals, functions, and operators) that yields a single data value.

extended payload: An arbitrary BLOB of data associated with a **Peer Name** and published by an application.

extended rule: A **rule** that is added to, modified, and deleted from a server by using a mechanism other than standard rules, but is otherwise functionally identical to a standard rule.

external identifier: A globally unique identifier for an entity that represents either a **foreign identifier** or an **internal identifier** (2). It consists of a GUID that represents a namespace followed by one or more bytes that contain an identifier for an entity within that namespace. If an external identifier represents an internal identifier, it can be also called a **global identifier**.

external OOF message: An **OOF message** that is sent to external users.

8 F

FAI contents table: A table of **folder associated information (FAI)** Message objects that are stored in a Folder object.

FastTransfer context: Either a **FastTransfer download context** or a **FastTransfer upload context**.

FastTransfer download context: A **Server object** that represents a context for a FastTransfer download.

FastTransfer stream: A binary format for encoding full or partial folder and message data. It can also encode information about differences between mailbox replicas.

FastTransfer upload context: A **Server object** that represents a context for a FastTransfer upload.

FAT32 file system: A derivative of the **file allocation table (FAT)** file system. **FAT32** supports smaller cluster sizes and larger **volumes** than **FAT**, which results in more efficient space allocation on **FAT32 volumes**. **FAT32** uses 32-bit addressing.

fax message: (1) A **Message object** that contains a digital representation of content received from a fax machine.

(2) See **message**.

field: (1) An element or **attribute** (1) in a data source that can contain data.

(2) A container for metadata within a SharePoint list and associated list items.

(3) A discrete unit of a record that has a name, a data type, and a value.

(4) The data elements that constitute an **Entity** in a line-of-business (LOB) system.

(5) An attribute or role of an **entity**.

file: (1) A single, discrete unit of content.

(2) An entity of data in the **file system** that a user can access and manage. A **file** must have a unique name in its directory. It consists of one or more streams of bytes that hold a set of related data, plus a set of **attributes** (also called properties) that describe the **file** or the data within the **file**. The creation time of a **file** is an example of a file **attribute**.

(3) A unit of data in the **file system**. An encrypted file consists of encrypted data along with the metadata required for a user to **decrypt** the file. The file and its metadata are protected using **public key** cryptography such that an authorized user's **private key** is required to **decrypt** the file.

(4) A **file** is a typed data stream. A **file** does not imply storage of the data stream in any particular medium or with any particular organization, or, for example, in a file system (italic is used when referring to traditional files).

file allocation table (FAT): A data structure that the operating system creates when a volume is formatted by using **FAT** or **FAT32 file systems**. The operating system stores information about each **file** in the **FAT** so that it can retrieve the **file** later.

File Allocation Table (FAT): A file system that is used by MS-DOS and Windows operating systems to organize and manage files.

File Replication Service (FRS): One of the services offered by a **domain controller (DC)**, which is advertised through the Domain Controller Location protocol. The service being offered to **clients** is a replicated data storage **volume** that is associated with the default **naming context (NC)**. The running or paused state of the **FRS** on a **DC** is available through protocols documented in [MS-ADTS] section 6.3.

file system: (1) A system that enables applications to store and retrieve files on storage devices. Files are placed in a hierarchical structure. The file system specifies naming conventions for files and the format for specifying the path to a file in the tree structure. Each file system consists of one or more drivers and DLLs that define the data formats and features of the file system. File systems can exist on the following storage devices: diskettes, hard disks, jukeboxes, removable optical disks, and tape backup units.

(2) A system that enables applications to store and retrieve **files** on storage devices. Files are placed in a hierarchical structure. The **file system** specifies naming conventions for **files** and the format for specifying the path to a **file** in the tree structure. Each **file system** consists of one or more drivers and DLLs that define the data formats and features of the **file system**. File systems can exist on the following storage devices: diskettes, hard disks, jukeboxes, removable optical disks, and tape backup units.

(3) A set of data structures for naming, organizing, and storing files in a **volume**. **NTFS**, **FAT**, and **FAT32** are examples of **file system** types.

File Transfer Protocol (FTP): A member of the TCP/IP suite of protocols that is used to copy files between two computers on the Internet if both computers support their respective FTP roles. One computer is an FTP client and the other is an FTP server.

final ICS state: An **Incremental Change Synchronization (ICS)** state that is provided by a server upon completion of an ICS operation. A final ICS state is a **checkpoint ICS state** that is provided at the end of the ICS operation.

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

floating: A time that is interpreted in an observer's location and does not necessarily translate into the same **Coordinated Universal Time (UTC)** time in different locations, as described in [RFC2445]. For example, a reminder for an appointment that starts at 1/1/2008 at 2:00 P.M. floating time would signal two hours earlier in Athens than in London.

folder: (1) A file system construct. File systems organize data by providing a hierarchy of objects, which are referred to as folders or directories, that contain files and can also contain other folders.

(2) A **file system** construct. File systems organize a **volume's** data by providing a hierarchy of objects known as **folders** or directories, which contain **files**.

(3) A container for **files** and other folders. A folder may be encrypted. The semantics of encrypting a folder are implementation-dependent. In the Windows implementation, encrypting a folder does not directly cause any data to be encrypted. Encrypting a folder in Windows has the following consequences of EFSRPC Metadata is created and stored with the folder and an **NTFS** attribute is set on the folder to signify that it is encrypted. **NTFS** checks this attribute when any new **files** or folders are created in the folder. **NTFS** will automatically encrypt any **files** or folders created within a folder that has this attribute set.

folder associated information (FAI): A collection of **Message objects** that are stored in a Folder object and are typically hidden from view by email applications. An FAI Message object is used to store a variety of settings and auxiliary data, including forms, views, calendar options, favorites, and category lists.

Folder object: A messaging construct that is typically used to organize data into a hierarchy of objects containing Message objects and **folder associated information (FAI)** Message objects.

foreign identifier: An identifier that is assigned to an entity by a foreign system, typically a client. It always has a form of an **external identifier**, but not all external identifiers are foreign identifiers.

forest: (1) One or more domains that share a common schema and trust each other transitively. An organization can have multiple forests. A forest establishes the security and administrative boundary for all objects that reside within the domains that belong to the forest. In contrast, a domain establishes the administrative boundary for managing objects, such as users, groups, and computers. In addition, each domain has individual security policies and trust relationships with other domains.

(2) In the **Active Directory** directory service, a forest is a set of **naming contexts (NCs)** consisting of one schema NC, one config NC, and one or more domain NCs. Because a set of NCs can be arranged into a tree structure, a forest is also a set of one or several trees of NCs.

(3) One or more **domains** that share a common **schema** and trust each other transitively. An organization can have multiple **forests**. A **forest** establishes the security and administrative boundary for all the **objects** that reside within the **domains** that belong to the **forest**. In contrast, a **domain** establishes the administrative boundary for managing **objects**, such as users, groups, and computers. In addition, each **domain** has individual security policies and trust relationships with other **domains**.

(4) In the **Active Directory** directory service, a **forest** is a set of **naming contexts (NCs)** consisting of one **schema NC**, one **config NC**, and one or more **domain NCs**. Because a set of **NCs** can be arranged into a tree structure, a **forest** is also a set of one or several trees of **NCs**.

(5) For **Active Directory Domain Services (AD DS)**, a set of **naming contexts (NCs)** consisting of one **schema naming context (schema NC)**, one **configuration naming context (config NC)**, one or more **domain naming contexts (domain NCs)**, and zero or more application naming contexts (application NCs). Because a set of **NCs** can be arranged into a tree structure, a **forest** is also a set containing one or several trees of **NCs**. For **AD LDS**, a set of **NCs** consisting of one **schema NC**, one **config NC**, and zero or more **application NCs**. (In Microsoft documentation, an **AD LDS forest** is called a "configuration set".)

form: (1) A structured document with controls and spaces that are reserved for entering and displaying information. Forms can contain special coding for actions such as submitting and querying data.

(2) A document with a set of controls into which users can enter information. Controls on a form can be bound to elements in the data source of the form, such as fields and groups. See also **bind**.

form template: A file or set of files that defines the data structure, appearance, and behavior of a **form** (2).

format: (1) To submit a command for a **volume** to write metadata to the disk, which is used by the **file system** to organize the data on the disk. A volume is **formatted** with a specific **file system**.

(2) A data structure that is used to define the encoding of audio and video data. The actual structures are opaque to [MS-RDPEV].

(3) A set of flags that encapsulates text layout information such as alignment, text direction, and trimming.

forward link attribute: An **attribute** whose values include **object** references (for example, an **attribute** of syntax Object(DS-DN)). The **forward link values** can be used to compute the values of a related **attribute**, a **back link attribute**, on other **objects**. If an **object** o refers to **object** r in **forward link attribute** f, and there exists a **back link attribute** b corresponding to f, then a **back link value** referring to o exists in **attribute** b on object r. The relationship between the forward and **back link attributes** is expressed using the linkId **attribute** on the attributeSchema **objects** representing the two **attributes**. The forward link's linkId is an even number, and the back link's linkId is the forward link's linkId plus one. A **forward link attribute** can exist with no corresponding **back link attribute**, but not vice-versa. For more information, see [MS-ADTS].

forward link value: The value of a **forward link attribute**.

free/busy message: A message that is stored in a public folder and contains free/busy data.

free/busy status: A property of an appointment that indicates how an appointment on the **calendar** of an attendee or resource affects their availability.

front-end web server: A server that hosts webpages, performs processing tasks, and accepts requests from protocol clients and sends them to the appropriate back-end server for further processing.

FRS: See **File Replication Service (FRS)**.

full reminder domain: The maximum scope that a client is allowed to use when searching for objects that have **reminders** enabled. The full reminder domain includes all folders except the following: Deleted Items, Junk Email, Drafts, Outbox, Conflicts, Local Failures, Server Failures, and Sync Issues.

full update: A **Meeting Update object** that includes a change to the recurrence pattern or the date or time, and requires a response from attendees.

fully qualified domain name (FQDN): (1) An unambiguous **domain name** (2) that gives an absolute location in the Domain Name System's (DNS) hierarchy tree, as defined in [RFC1035] section 3.1 and [RFC2181] section 11.

(2) In **Active Directory**, a **fully qualified domain name (FQDN)** (1) that identifies a **domain**.

(3) A fully qualified domain name (FQDN) (1) that does not include the "ldap/" prefix.

9 G

game: An application that uses a **DirectPlay** protocol to communicate between computers.

game session: The metadata associated with the collection of computers participating in a single **instance** of a computer **game**.

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

GC: See **global catalog (GC)**.

Generic Security Services (GSS): An Internet standard, as described in [RFC2743], for providing security services to applications. It consists of an application programming interface (GSS-API) set, as well as standards that describe the structure of the security data.

ghosted: (1) 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.

(2) See **uncustomized**.

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

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

global catalog (GC): A unified partial view of multiple **naming contexts (NCs)** in a distributed partitioned directory. The **Active Directory** directory service **GC** is implemented by **GC servers**. The definition of **global catalog** is specified in [MS-ADTS] section 3.1.1.1.8.

global catalog server (GC server): A **domain controller (DC)** that contains a naming context (NC) replica (one full, the rest partial) for each **domain naming context** in the **forest**.

global counter: A 6-byte value that is incremented automatically. If a global counter is paired with a **REPLID** it forms a message ID (MID), folder ID (FID), or change number. If a global counter is paired with a **REPLGUID** it forms a **global identifier**.

global directory: A globally accessible database containing entries that correlate servers, databases, and user **mailboxes**. The server uses the correlated data to determine, for a specific user, which server and database to access for a private mailbox logon or a public folder logon. The global directory also contains other pertinent configuration information that is crucial to the overall operation of the client/server deployment. **Active Directory** can be used for the global directory, but the implementer determines what to use for the global directory.

global identifier: A form of encoding for an internal identifier that makes it unique across all stores. Global identifiers are a subset of **external identifiers**, and they consist of a **REPLGUID** followed by a 6-byte **global counter**.

global partition: The default, required partition on a COMA server.

global version sequence number (GVSN): A pair of numbers that includes the **machine identifier** and the **version sequence number (VSN)**. While two machines might assign the same **VSN**, because they have different machine identifiers, the associated **GVSNs** differ. A **GVSN** is used to identify a unique version of a unique resource. In other words, no two

different resources are ever assigned the same **GVSN**, and no two different updates to the same resource are ever assigned the same **GVSN**.

Global Version Sequence Numbers (GVSN): A GVSN is a pair: Machine identifier and **version sequence number (VSN)**. Although two machines might assign the same **VSN**, because they have different machine identifiers, the associated GVSNs differ. A GVSN is used to identify a unique version of a unique resource. In other words, no two different resources ever get assigned the same GVSN, and no two different updates to the same resource ever get assigned the same GVSN.

Globally Routable User Agent URI (GRUU): A **URI** that identifies a **user agent** and is globally routable. A URI possesses a GRUU property if it is useable by any **user agent client (UAC)** that is connected to the Internet, routable to a specific user agent instance, and long-lived.

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

graph: A set of connected nodes.

graph maintenance: The process by which each **node** attempts to improve its connectivity within the **graph**.

group: (1) An element that can contain fields and other groups in the data source for an InfoPath form. Controls that contain other controls, such as repeating tables and sections, are bound to groups.

(2) A named collection of users who share similar access permissions or roles.

(3) A named collection of quick links, colleagues, or memberships for the purpose of organization.

(4) A process of combining similar elements into a set in accordance with logical criteria. It is frequently used to combine sets of data from Online Analytical Processing (OLAP) databases or PivotTable reports.

(5) A collection of **objects** that can be treated as a whole.

(6) A collection of **players** within a **game session**. Typically, **players** are placed in a **group** when they serve a common purpose.

(7) A graph in which each node implements the group security model.

(8) A **cluster** group is a container for zero or more **cluster resources**, when referring to cluster groups. **Groups** enable **resources** to be combined into larger logical units and are owned by only one **node** in the **cluster** at a time.

(9) A **group object**.

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

group object: (1) A database object that represents a collection of user and group objects and has a **security identifier (SID)** value.

(2) In **Active Directory**, a group object has an object class group. A group has a forward link attribute member; the values of this **attribute** (2) either represent elements of the group (for example, objects of class user or computer) or subsets of the group (objects of class group). The back link attribute memberOf enables navigation from group members to the groups containing them. Some groups represent groups of security principals and some do not and are, for instance, used to represent email distribution lists.

(3) A database **object** that represents a collection of user and group **objects** and has a **security identifier (SID)** value.

(4) In **Active Directory**, a **group object** has an **object class** group. A **group** has a **forward link attribute** member; the values of this **attribute** either represent elements of the **group** (for example, **objects** of class user or computer) or subsets of the **group** (**objects** of class group). The representation of group subsets is called "nested group membership" .The **back link attribute** memberOf enables navigation from **group** members to the **groups** containing them. Some **groups** represent **groups** of **security principals** and some do not and are, for instance, used to represent email distribution lists.

Group Policy Object (GPO): A collection of administrator-defined specifications of the policy settings that can be applied to groups of computers in a domain. Each GPO includes two elements: an object that resides in the **Active Directory** for the domain, and a corresponding file system subdirectory that resides on the sysvol DFS share of the Group Policy server for the domain.

GUID: (1) A term used interchangeably with universally unique identifier (UUID) in Microsoft protocol technical documents (TDs). Interchanging the use of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] need to be used to generate the GUID. See also **universally unique identifier (UUID)**.

(2) See **globally unique identifier (GUID)**.

GUIDString: A **GUID** in the form of an **ASCII** or **Unicode** string, consisting of one group of 8 hexadecimal digits, followed by three groups of 4 hexadecimal digits each, followed by one group of 12 hexadecimal digits. It is the standard representation of a GUID, as described in [RFC4122] section 3. For example, "6B29FC40-CA47-1067-B31D-00DD010662DA". Unlike a curly braced GUID string, a GUIDString is not enclosed in braces.

10 H

handle: (1) Any token that can be used to identify and access an object such as a device, file, or a window.

(2) A 32-bit numerical ID that uniquely identifies a resource or a channel. Handles are allocated by the server and communicated to the client via resource or channel creation messages.

(3) A recipient of a message.

(4) A token that can be used to identify and access cursors, chapters, and bookmarks.

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

hard delete: A process that removes an item permanently from the system. If an item is hard deleted, a server does not retain a back-up copy of the item and a client cannot access or restore the item. See also **soft delete**.

hash: (1) A fixed-size result that is obtained by applying a one-way mathematical function, which is sometimes referred to as a hash algorithm, to an arbitrary amount of data. If the input data changes, the hash also changes. The hash can be used in many operations, including **authentication** (2) and digital signing.

(2) A hash, such as SHA-1, on the content or content block.

(3) A term that refers to either a hash function, the value computed by such a function, or the act of computing such a value.

hash list: A list of **hashes** that include the **block hashes** and the **content hash**.

header: (1) A line, or lines, of content in the top margin area of a page in a document or a slide in a presentation. A header typically contains elements such as the title of the chapter, the title of the document, a page number, or the name of the author.

(2) A name-value pair that supplies structured data in an Internet email message or **MIME entity**.

(3) The structure at the beginning of a **compound file**.

header field: (1) A component of a Session Initiation Protocol (SIP) message header, as described in [RFC3261].

(2) As specified in section 4.2 of [RFC2616].

header message object: A **Message object** that contains partial information about a message on a server, such as an identifier for the message, the display names of the recipients and the sender, the subject of the message, and the delivery time of the message. It allows a client to display enough information about a message to let a user choose whether to download the message.

header row: (1) A row in a table, typically the first row, that contains labels for **columns** (2) in the table.

(2) A row at the beginning of a **category** (5) that does not represent data in the **Table object**, but provides information about a grouping.

Help file: A file that contains the documentation for a specific product or technology.

hierarchy synchronization: The process of keeping synchronized versions of folder hierarchies and their properties on a client and server.

hierarchy table: A **Table object** whose rows represent the **Folder objects** that are contained in another Folder object.

host: (1) A general-purpose computer that is networking capable.

(2) In **DirectPlay**, the computer responsible for responding to **DirectPlay** game session enumeration requests and maintaining the master copy of all the **player** and group lists for the game. One computer is designated as the **host** of the **DirectPlay** game session. All other participants in the **DirectPlay** game session are called **peers**. However, in peer-to-peer mode the name table entry representing the **host** of the session is also marked as a **peer**.

(3) A subcomponent of the naming **authority** in a **URI scheme**, as defined in [RFC3986] section 3.2.2.

(4) An interface between an application **runspace** and a user capable of responding to the host method calls specified in [MS-PSRP] section 2.2.3.17.

(5) The machine with the desktop or applications that are being **shared** with the other **participants**.

hosted cache: A centralized cache comprised of **blocks** added by **peers**.

Hypertext Markup Language (HTML): An application of the Standard Generalized Markup Language (SGML) that uses tags to mark elements in a document, as described in [HTML].

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 webpage requests.

11 I

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

identifier: A string value that is used to uniquely identify a component of the **CSDL** and that is of type SimpleIdentifier.

identity provider (IP): A **security token service (STS)** that performs identity verification as part of its processing. For more information, see [WSFedPRP].

identity provider/security token service (IP/STS): An **STS** that may or may not be an **identity provider (IP)**. This term is used as shorthand to see both identity that verifies token services and general token services that do not verify identity. Note that the "/" symbol implies an "or" relationship.

IDL: See **Interface Definition Language (IDL)**.

import: The process of creating a **conglomeration** or **partition** on a COMA server based on **modules** and configurations extracted from an **installer package file**.

inactive search folder: A **search folder** (2) that does not have a **search folder container**.

Inbox folder: A **special folder** that is the default location for **Message objects** received by a user or resource.

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

indexing: The process of extracting text and properties from files and storing the extracted values into the indexes (for text) and the property cache (for properties).

indexing service: (1) A service that traverses URL spaces and file systems to acquire items, including properties, to record in catalogs for those spaces and systems. The catalogs can then be used for tasks such as searching and auditing content.

(2) A service that creates **indexed** **catalogs** for the contents and properties of file systems. Applications can search the **catalogs** for information from the files on the **indexed** file system.

Information Rights Management (IRM): A technology that provides persistent protection to digital data by using encryption, **certificates** (1), and **authentication** (2). Authorized recipients or users acquire a license to gain access to the protected files according to the rights or business rules that are set by the content owner.

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

initial ICS state: An **Incremental Change Synchronization (ICS)** state that is provided by a client when it configures an ICS operation.

installer package file: A file that packages together **modules** and configuration states sufficient to create a **conglomeration** or **partition** on a server.

instance: (1) A unique publication of data for a **category** (4). It enables a **publisher** to publish data for the same category multiple times. An example is a publisher who uses two different **endpoints** (5) to publish data. These endpoints can publish the same category. However,

each endpoint requires a different instance number to be considered a distinct publication by the **server** (2). An instance number is provided by the publishing client.

(2) A specific occurrence of a **game** running in a **game session**. A **game** application process or module may be created more than one time on a single computer system, or on separate computer systems. Each time a **game** application process or module is created, the occurrence is considered to be a separate **instance**.

instant messaging: A method of real-time communication over the Internet in which a sender types a message to one or more recipients and the recipient immediately receives the message in a pop-up window.

Integrated Services Digital Network (ISDN): A high-speed digital technology that uses existing telephone lines to provide Internet access and digital network services.

interface: (1) A specification in a **Component Object Model (COM)** server that describes how to access the methods of a class. For more information, see [MS-DCOM].

(2) A group of related function prototypes in a specific order, analogous to a C++ virtual interface. Multiple objects, of different object class, may implement the same interface. A derived interface may be created by adding methods after the end of an existing interface. In the Distributed Component Object Model (DCOM), all interfaces initially derive from IUnknown.

(3) This term is used exactly as specified in [C706] section "Introduction to the RPC API" in Part 2.

(4) A collection of messages used together. **Interfaces** support inheritance and extensibility through the Interface Query message as defined in [MS-RDPEXPS] section 1.3.2.1.1.

(5) Represents a network that can be reached over an **adapter**. Each **interface** has a unique **interface** identifier also known as an **interface** index. **interfaces** that are active have an **adapter** that is providing connectivity to the network they represent. **interfaces** that are inactive do not have an **adapter** providing connectivity unless an administrator disabled the **interface** after it already had an **adapter**. Routing a packet to a network represented by an **interface** will cause the router to allocate an **adapter** for that **interface**, and will establish a wide area network (WAN) connection to the remote network. Allocating an **adapter** to an **interface** is referred to as binding. In the case of a local area network (LAN) **interface**, the **interface** corresponds to an actual physical device in the computer, a LAN **adapter**. In the case of a WAN interface, the **interface** is mapped to a port at the time that a connection is established. The port could be a COM port, a parallel port, or a virtual port (for tunnels such as **PPTP** [RFC2637] and **L2TP** [RFC2661]). WAN interfaces have the additional quality that they typically receive a network address only at the time that a connection is established. For example, a WAN interface using PPP [RFC1661] receives its network layer address from the remote peer during the connection process. Receiving a network address as part of the connection process is sometimes referred to as late-binding.

Interface Definition Language (IDL): The International Standards Organization (ISO) standard language for specifying the **interface** for remote procedure calls. For more information, see [C706] section 4.

internal identifier: (1) An integer that uniquely identifies any item in a **term store**.

(2) A Folder ID or Message ID, as described in [MS-OXCDATA].

Internationalized Resource Identifier (IRI): A resource identifier that conforms to the rules for Internationalized Resource Identifiers, as defined in [RFC3987].

Internet Key Exchange (IKE): The protocol that is used to negotiate and provide authenticated keying material for **security associations (SAs)** in a protected manner. For more information, see [RFC2409].

Internet Mail Connector Encapsulated Address (IMCEA): A means of encapsulating an email address that is not compliant with [RFC2821] within an email address that is compliant with [RFC2821].

Internet message: A message, such as an email message, that conforms to the syntax that is described in [RFC2822].

Internet Message Access Protocol - Version 4 (IMAP4): A protocol that is used for accessing email and news items from mail servers, as described in [RFC3501].

Internet Protocol security (IPsec): A framework of open standards for ensuring private, secure communications over Internet Protocol (IP) networks through the use of cryptographic security services. IPsec supports network-level peer authentication, data origin authentication, data integrity, data confidentiality (encryption), and replay protection. The Microsoft implementation of IPsec is based on standards developed by the Internet Engineering Task Force (IETF) IPsec working group.

Internet Protocol version 6 (IPv6): A revised version of the Internet Protocol (IP) designed to address growth on the Internet. Improvements include a 128-bit IP address size, expanded routing capabilities, and support for **authentication** (2) and privacy.

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

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

INVITE: A **Session Initiation Protocol (SIP)** method that is used to invite a user or a service to participate in a session.

item: A unit of content that can be indexed and searched by a search application.

12 J

JavaScript Object Notation (JSON): A text-based, data interchange format that is used to transmit structured data, typically in **Asynchronous JavaScript + XML (AJAX)** web applications, as described in [RFC4627]. The JSON format is based on the structure of ECMAScript (Jscript, JavaScript) objects.

job: (1) An inbound or outbound fax transmission that is awaiting transmission in the Fax Queue; the Fax Jobs are qualified as inbound or outbound based on this. The Fax Jobs are further qualified as follows: queued qualifies a Fax Job as awaiting transmission, and active qualifies a Fax Job as in process of being sent or received by the fax server.

(2) An object identifying an administrative action (for example, running a program) to be performed on specified **triggers** and **conditions** (for example, every day at a specific time). Synonym for Task.

Joint Photographic Experts Group (JPEG): A raster graphics file format for displaying high-resolution color graphics. JPEG graphics apply a user-specified compression scheme that can significantly reduce the file sizes of photo-realistic color graphics. A higher level of compression results in lower quality, whereas a lower level of compression results in higher quality. JPEG-format files have a .jpg or .jpeg file name extension.

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

Journal object: A **Message object** that represents an entry in a journal or log and adheres to the property descriptions that are described in in [MS-OXOJRNL].

Junk Email folder: A **special folder** that is the default location for **Message objects** that are determined to be junk email by a Junk Email rule.

Junk Email rule: An **extended rule** that describes a **spam filter**.

13 K

Kerberos: (1) An **authentication** (2) system that enables two parties to exchange private information across an otherwise open network by assigning a unique key (called a **ticket**) to each user that logs on to the network and then embedding these tickets into messages sent by the users. For more information, see [MS-KILE].

(2) An **authentication** system that enables two parties to exchange private information across an otherwise open network by assigning a unique **key** (called a **ticket**) to each user that logs on to the network and then embedding these **tickets** into messages sent by the users. For more information, see [MS-KILE].

(3) An authentication access type as defined by [RFC1964].

Kerberos principal: A unique individual account known to the **Key Distribution Center (KDC)**. Often a user, but it can be a service offering a resource on the network.

key: (1) In the **registry**, a node in the logical tree of the data store.

(2) In cryptography, a generic term used to refer to cryptographic data that is used to initialize a cryptographic algorithm. **Keys** are also sometimes referred to as **keying material**.

(3) A 256-bit unsigned integer used internally by MC-DRT to identify a resource.

Key Distribution Center (KDC): The **Kerberos** service that implements the **authentication** and **ticket** granting services specified in the **Kerberos** protocol. The service runs on computers selected by the administrator of the **realm** or domain; it is not present on every machine on the network. It must have access to an **account** database for the **realm** that it serves. Windows **KDCs** are integrated into the **domain controller** role of a Windows Server acting as a Domain Controller. It is a network service that supplies **tickets** to **clients** for use in authenticating to services.

keying material: The data from which the **main mode (MM)** and **quick mode (QM) security association (SA) authentication** and **encryption** **keys** are generated.

14 L

L2TP: Layer Two Tunneling Protocol, as defined in [RFC2661].

language code identifier (LCID): A 32-bit number that identifies the user interface human language dialect or variation that is supported by an application or a client computer.

LDAP: (1) See **Lightweight Directory Access Protocol (LDAP)**.

(2) Lightweight Directory Access Protocol, which can be either version 2 [RFC1777], or version 3 [RFC3377].

LDAP Data Interchange Format (LDIF): A standard that defines how to import and export directory data between directory servers that use the **Lightweight Directory Access Protocol (LDAP)**, as described in [RFC2849].

LDAP Distinguished Name: A string representation of a **distinguished name (DN)** (4) used to access an object on a directory server via **Lightweight Directory Access Protocol (LDAP)**.

leaf license: A **license** that specifies rules that augment or restrict the rules in a **root license**. A **leaf license** can be more or less restrictive than a **root license**.

Lempel-Ziv Extended (LZX): An LZ77-based compression engine, as described in [UASDC], that is a universal lossless data compression algorithm. It performs no analysis on the data.

Lempel-Ziv Extended Delta (LZXD): A derivative of the Lempel-Ziv Extended (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.

Library: (1) Part of the **Remoting Data Model**. A **Library** is a named unit that contains a collection of **Remoting Types**. For more information, see Library in [MS-NRTP] section 3.1.1.

(2) A storage device that contains one or more tape drives, a number of slots to hold tape cartridges, and an automated method for loading tapes.

license: (1) A data structure that contains, but is not limited to, policies and an encrypted content key. WMDRM: Network Devices Protocol has four types of licenses: standard licenses, **root licenses**, **base licenses**, and **leaf licenses**.

(2) An XrML1.2 document that describes usage policy for **protected content**.

Lightweight Directory Access Protocol (LDAP): The primary access protocol for **Active Directory**. Lightweight Directory Access Protocol (LDAP) is an industry-standard protocol, established by the Internet Engineering Task Force (IETF), which allows users to query and update information in a **directory service (DS)**, as described in [MS-ADTS].

line-of-business (LOB) system: A software system that is used to store business data and can also contain business rules and **business logic** (2) that support business processes.

linked object: (1) An object that is inserted into a document and continues to exist in a separate source file. If the object in the source file changes, the object in the document is updated automatically to reflect those changes.

(2) Application data that is referenced by documents from other applications.

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

LobSystem: A type of **MetadataObject** that represents a specific version of a line-of business (LOB) system. An LOB system can be a database or a web service.

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

locale: (1) A collection of rules and data that are specific to a language and a geographical area. A locale can include information about sorting rules, date and time formatting, numeric and monetary conventions, and character classification.

(2) An identifier, as specified in [MS-LCID], that specifies preferences related to language. These preferences indicate how dates and times are to be formatted, how items are to be sorted alphabetically, how strings are to be compared, and so on.

logical unit (LU): An addressable network element in the Systems Network Architecture that serves as an access point to the network for programs and users, allowing them to access resources and communicate with other programs and users. For more information on logical units, see [SNA].

Logon object: A **Server object** that provides access to a private **mailbox** or a **public folder**. A client obtains a Logon object by issuing a RopLogon **remote operation (ROP)** to a server.

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

lowest-cost server: A server whose communication cost to access is the lowest in a list of servers.

LU Type 6.2 (LU 6.2): A type of **logical unit** designed to provide support for two or more distributed **application** programs cooperating to carry out some **work**. All communication provided by **LU 6.2** is program-to-program. For more information, see [LU62Peer].

15 M

machine identifier: A **GUID** that is unique for each machine.

Magazines: See Slots.

mail app: An **app for Office** that enhances an email or appointment item.

mail tip: A note that is presented to the author of a message when the author is composing the message. A mail tip provides information about the recipients of a message and issues that might impact 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 to compose and read email messages.

mailbox: A **message store** that contains email, calendar items, and other **Message objects** for a single recipient.

main data source: An **XML document** or **XML schema** that defines the collection of **fields** (1) and **groups** (1) that store data for an InfoPath form.

main mode (MM): The first phase of an **Internet Key Exchange (IKE)** negotiation that performs authentication and negotiates a **main mode security association (MM SA)** between the peers. For more information, see [RFC2409] section 5.

main mode security association (MM SA): A security association that is used to protect **Internet Key Exchange (IKE)** traffic between two peers. For more information, see [RFC2408] section 2.

manageable entity: A **Common Information Model (CIM)** instance that represents a manageable component of an operating system.

Managed Object Format (MOF): A textual encoding for **Common Information Model (CIM)** objects, this representation is not used within protocol operations defined in [MS-WMI]. MOF is defined in [DMTF-DSP0004] section 3. The MOF text encoding is only used for illustrative purposes. The binary encoding can be translated to and from the MOF format.

management client: An application that uses the WSRM Protocol interfaces for the purpose of presenting a user interface that allows a user to perform the functions exposed by the WSRM Protocol.

management service: An agent that implements the WSRM Protocol on a given computer by applying specified resource policies, returning requested **accounting** information, and storing the **accounting** data dumped by the other management services running on remote servers.

management state: A state switch with two values, running and stopped, that tells whether the **management service** can be active or inactive. "Running" means the service will perform all **resource management** and **accounting** functions according to its current policies. "Stopped" means that it will remain in an inactive state, doing nothing except making configuration changes that will take effect when the **management service** becomes active again; for example, import, export, creation, deletion, or modification of resource allocation policy.

mapper: A **station** that initiates a **topology discovery test**.

mapping mode: The way in which logical (device-independent) coordinates are mapped to **device space** (device-specific) coordinates. It also specifies the orientation of the axes and size of the units used for drawing operations.

marker: An unsigned 32-bit integer value that adheres to **property tag** syntax and is used to denote the start and end of related data in a **FastTransfer stream**. The property tags that are used by markers do not represent valid properties.

marshal: To encode one or more data structures into an octet stream using a specific **remote procedure call (RPC)** transfer syntax (for example, marshaling a 32-bit integer).

media: Compressed audio, video, and text data that is used by the client to play a **presentation**.

media data: The data for an audio or video **stream** in a **presentation**, encoded in a specific **format**.

media type: A value that is specified in a Content-Type Header field, as described in [RFC2045].

meeting: An event with attendees.

Meeting Cancellation object: A **Message object** that represents a meeting organizer's cancellation of a previously scheduled meeting.

Meeting Forward Notification object: A **Message object** that represents a notification that is sent to the meeting organizer when an attendee forwards a meeting request.

Meeting object: A **Calendar object** that has both an organizer and attendees.

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

Meeting Request object: A **Message object** that represents an invitation from the meeting organizer to an attendee.

Meeting Response object: A **Message object** that represents an attendee’s response to a meeting organizer's invitation. The response indicates whether the attendee accepted, tentatively accepted, or declined the meeting request. The response can include a proposed new date or time for the meeting.

meeting suggestions: A possible meeting time based on the **availability** of the meeting attendees.

meeting update: An instance of a **Meeting Update object**.

Meeting Update object: A **Message object** that represents a meeting organizer's changes to a previously scheduled meeting. The update is categorized as either a full update or an informational update.

Meeting Workspace: A website that is created by using the Meetings Web Services protocol, as described in [MS-MEETS]. It can host documents, discussions, and other information about a meeting.

meeting-related object: A **Message object** that represents a relay of information between a meeting organizer and an attendee. It can be any of the following: **Meeting Request object**, **Meeting Update object**, **Meeting Cancellation object**, or **Meeting Response object**.

mesh: (1) A network of nodes that are all identified with the same **mesh name**.

(2) The covering of the surface of an object by triangular shapes without gaps or overlaps.

mesh name: A set of nodes that establish connections to each other to form a **mesh**.

message: (1) A data structure representing a unit of data transfer between distributed applications. A message has **message properties**, which may include message header properties, a **message body** property, and message trailer properties.

(2) See **message tag (MTAG)**.

(3) A fax that a fax server has completely received or transmitted, and **archived** to the Fax Archive Folder described in [MS-FAX] section 3.1.1.

(4) An atomic unit in the OMA-DM protocol.

message body: (1) The content within an HTTP message, as described in [RFC2616].

(2) The main message text of an email 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.

(3) A distinguished **message property** that represents the application payload.

(4) As specified in [RFC2616] section 4.3.

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

Message object: A set of properties that represents an email message, appointment, contact, or other type of personal-information-management object. In addition to its own properties, a Message object contains recipient properties that represent the addressees to which it is addressed, and an **attachments table** that represents any files and other Message objects that are attached to it.

message part: A **message body** (2) with a string property that contains only the portion of an email message that is original to the message. It does not include any previous, quoted messages. If a message does not quote a previous message, the message part is identical to the message body.

message property: A data structure that contains a **property identifier** and a value, and that is associated with a **message**.

message queue: A data structure containing an ordered list of zero or more **messages**. A **queue** has a head and a tail and supports a first in, first out (FIFO) access pattern. **Messages** are appended to the tail through a write operation (Send) that appends the **message** and increments the tail pointer. **Messages** are consumed from the head through a destructive read operation (Receive) that deletes the **message** and increments the head pointer. A **message** at the head may also be read through a nondestructive read operation (Peek).

message store: A unit of containment for a single hierarchy of Folder objects, such as a mailbox or public folders.

message tag (MTAG): (1) A message that is sent between participants in the context of connections.

(2) A 4-byte integer value that describes the message type and its interpretation.

message transfer agent (MTA): An **SMTP** server that accepts mail from a client or another MTA and delivers the mail or relays it to another MTA.

Messaging Application Programming Interface (MAPI): (1) A messaging architecture that enables multiple applications to interact with multiple messaging systems across a variety of hardware platforms.

(2) A Windows programming interface that enables email to be sent from within a Windows application.

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

metadata: (1) XML-formatted data that defines the characteristics of an **update**, including its title, description, rules for determining whether the **update** is applicable to a **client computer**, and instructions for installing the **update** content.

(2) A generic term for a **hash** or **hash list**.

MetadataObject: An abstract data structure that consists of a set of **attributes** (1) that represent a LobSystem, LobSystemInstance, DataClass, Entity, Method, MethodInstance, Parameter, TypeDescriptor, Identifier, FilterDescriptor, **Action**, **ActionParameter**, or **Association**.

metafile: (1) A file that stores an image as graphical objects, such as lines, circles, and polygons, instead of pixels. A metafile preserves an image more accurately than pixels when an image is resized.

(2) A sequence of record structures that store an image in an application-independent format. Metafile records contain drawing commands, object definitions, and configuration settings. When a metafile is processed, the stored image can be rendered on a display, output to a printer or plotter, stored in memory, or saved to a file or stream.

meta-property: An entity that is identified with a **property tag** containing information (a value) that describes how to process other data in a **FastTransfer stream**.

MethodInstance: A type of **MetadataObject** that associates a normalized or stereotypical semantic with a Method that represents a native API in a line-of-business (LOB) system. MethodInstances identify which part of the data that is returned by a Method is relevant for the semantic by defining a **ReturnTypeDescriptor**. MethodInstances are contained by Methods.

Microsoft Message Queuing (MSMQ): A communications service that provides asynchronous and reliable **message** passing between distributed **applications**. In **Message Queuing**, **applications** send **messages** to **queues** and consume **messages** from **queues**. The **queues** provide persistence of the **messages**, enabling the sending and receiving **applications** to operate asynchronously from one another.

MIME attachment: A body part that is in a **MIME** message, for example, an email message or a file that is attached to an email message.

MIME body: The content of a **MIME** entity, which follows the header of the MIME entity to which they both belong.

MIME content-type: A content type that is as described in [RFC2045], [RFC2046], and [RFC2047].

MIME Encapsulation of Aggregate HTML Documents (MHTML): A MIME-encapsulated HTML document, as described in [RFC2557].

MIME entity: An entity that is as described in [RFC2045], [RFC2046], and [RFC2047].

MIME entity header: A type of header that is as described by [RFC2045].

MIME message: A message that is as described in [RFC2045], [RFC2046], and [RFC2047].

MIME part: A message part that is as described in [RFC2045], [RFC2046], and [RFC2047].

MIME writer: An agent that performs **MIME** generation. It can be a client or a server.

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

minimal reminder domain: The smallest scope that a client is allowed to use when searching for objects that have an **active reminder**. The minimal reminder domain includes the following folders: Inbox, primary Contacts, primary Calendar, and primary Tasks. It does not include sub-folders.

module: (1) A collection of routines and data structures that performs a specific task or implements a specific abstract data type. Modules usually consist of two parts, a module header and a module body. A module header is a set of name/value attribute pairs that specify the linguistic characteristics of the module. A module body is the VBA source code, a set of declarations followed by procedures. VBA supports two types of modules, **procedural modules** and class modules.

(2) A file used by a server to register and instantiate one or more components. It contains either implementations of the components or metadata that a server can use to find implementations.

(3) A BLOB in the Desired State Configuration Pull Model Protocol [MS-DSCPM]. The protocol does not process the content of the BLOB, and it is passed as it is to the higher layer.

Mount: To move **physical media** from a **library slot** to a **drive**.

MSMQ: See **Microsoft Message Queuing (MSMQ)**.

MSMQ Directory Service: A network directory service that provides directory information, including key distribution, to **MSMQ**. It initially shipped in the Windows NT 4.0 Option Pack as part of **MSMQ**. This directory service predates and is superseded by **Active Directory (AD)**.

MSMQ object: Any one of the objects stored by **MSMQ** in its directory service. An object has a class name and a set of properties.

multibyte character set (MBCS): An alternative to **Unicode** for supporting character sets, like Japanese and Chinese, that cannot be represented in a single byte. Under **MBCS**, characters are encoded in either one or two bytes. In two-byte characters, the first byte, or "lead" byte, signals that both it and the following byte are to be interpreted as one character. The first byte comes from a range of codes reserved for use as lead bytes. Which ranges of bytes can be lead bytes depends on the **code page** in use. For example, Japanese **code page** 932 uses the range 0x81 through 0x9F as lead bytes, but Korean **code page** 949 uses a different range.

Multimedia Messaging Service (MMS): A communications protocol that is designed for messages containing text, images, and other multimedia content that is sent between mobile phones.

Multipurpose Internet Mail Extensions (MIME): A set of extensions that redefines and expands support for various types of content in email messages, as described in [RFC2045], [RFC2046], and [RFC2047].

multivalue instance: A row that is in a table and corresponds to a single value in a multivalue property. There are multiple rows for each **Message object** in a table and each row corresponds to one value of the multivalue property. Each row has a single value for the property and the properties for the other columns are repeated.

multivalue property: A property that can contain multiple values of the same type.

multi-valued claim: See the definition of **claim**.

16 N

name identifier: The identifier that is 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, depending on the value of the Kind member variable.

name record: The **NetBIOS name**-to-IPv4 address mapping.

name service provider interface (NSPI): A method of performing address-book-related operations on **Active Directory**.

name table: The list of systems participating in a **DXDiag**, **DirectPlay 4**, or **DirectPlay 8** session, as well as any application-created groups.

named pipe: A named, one-way, or duplex pipe for communication between a pipe server and one or more pipe clients.

named property: A property that is identified by both a GUID and either a string name or a 32-bit identifier.

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

namespace: (1) A name that is defined on the **schema** (2) and that is subsequently used to prefix **identifiers** to form the **namespace qualified name** of a structural type.

(2) An abstract container that provides context for the items (names, technical terms, or words) that it holds and allows disambiguation of items that have the same name (residing in different **namespaces**).

(3) The entire **collection** (as specified in [RFC4918] section 5.2) of items under a request **URI**.

namespace qualified name: A qualified name that refers to a structural type by using the name of the **namespace** (1), followed by a period, followed by the name of the structural type.

naming context (NC): An **NC** is a set of objects organized as a tree. It is referenced by a DSName. The **DN** of the DSName is the distinguishedName **attribute** of the tree root. The **GUID** of the DSName is the objectGUID **attribute** of the tree root. The **security identifier (SID)** of the DSName, if present, is the objectSid **attribute** of the tree root; for **Active Directory Domain Services (AD DS)**, the **SID** is present if and only if the **NC** is a **domain naming context (domain NC)**. **Active Directory** supports organizing several **NCs** into a tree structure.

NAP: See **Network Access Protection (NAP)**.

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

NC: See **naming context (NC)**.

NC replica: A variable containing a tree of **objects** whose root **object** is identified by some **naming context (NC)**.

NetBIOS: A particular network transport that is part of the LAN Manager protocol suite. **NetBIOS** uses a broadcast communication style that was applicable to early segmented local area networks. The LAN Manager protocols were the default in Windows NT environments

prior to Windows 2000. A protocol family including name resolution, datagram, and connection services. For more information, see [RFC1001] and [RFC1002].

NetBIOS name: A 16-byte address that is used to identify a **NetBIOS** resource on the network. For more information, see [RFC1001] and [RFC1002].

NetBIOS Name Server (NBNS): A server that stores NetBIOS name-to-IPv4 address mappings and that resolves NetBIOS names for NBT-enabled hosts. A server running the Windows Internet Name Service (WINS) is the Microsoft implementation of an NBNS.

network access client (NAC): An endpoint that establishes a call session to a **NAS** in order to perform network access.

Network Access Protection (NAP): A feature of an operating system that provides a platform for system health-validated access to private networks. **NAP** provides a way of detecting the health state of a network client that is attempting to connect to or communicate on a network, and limiting the access of the network client until the health policy requirements have been met. **NAP** is implemented through quarantines and health checks, as specified in [TNC-IF-TNCCSPBSoH].

network access server (NAS): A computer server that provides an access service for a user who is trying to access a network. A **NAS** operates as a client of **RADIUS**. The **RADIUS client** is responsible for passing user information to designated **RADIUS servers** and then acting on the response returned by the **RADIUS server**. Examples of a NAS include: a VPN server, Wireless Access Point, 802.1x-enabled switch, or **Network Access Protection (NAP)** server.

network byte order: The order in which the bytes of a multiple-byte number are transmitted on a network, most significant byte first (in **big-endian** storage). This may or may not match the order in which numbers are normally stored in memory for a particular processor.

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] section 14.

node: (1) A location in a diagram that can have links to other locations.

(2) A computer system that is configured as a member of a **cluster**. That is, the computer has the necessary software installed and configured to participate in the **cluster**, and the **cluster** configuration includes this computer as a member.

(3) An instance of the Peer-to-Peer Graphing Protocol.

(4) An entry identified by name in a DNS **zone**. A **node** contains all of the DNS records sets associated with the name.

(5) An **endpoint** in the computer network that can receive, send, or process a SOAP message, as specified in [SOAP1.2-2/2007].

(6) An instance of PNRP running on a machine.

(7) An instance of DRT running on a machine.

(8) An instance of a channel **endpoint** participating in the **mesh** that implements the Peer Channel Protocol.

node ID: A statistically unique 64-bit identifier for a **node** in a **graph**. A **node ID** must be unique within a **graph**.

non-delivery report: A report message that is generated and sent by a server to the sender of a message if an email message could not be received by an intended recipient.

non-read receipt: A message that is generated when an email message is deleted at the expiration of a time limit or due to other client-specific criteria.

non-Unicode: A **character set** (1) that has a restricted set of glyphs, such as Shift_JIS or ISO-2022-JP.

normal message: A message that is not a **folder associated information (FAI)** message.

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

Notes folder: A **Folder object** that contains **Note objects**.

notification: (1) A process in which a subscribing **Session Initiation Protocol (SIP)** client is notified of the state of a subscribed resource by sending a NOTIFY message to the subscriber.

(2) A typed buffer of data sent by a **print server** to a **print client** as a result of an event on the server.

(3) The act of a notifier sending a **NOTIFY** message to a subscriber to inform the subscriber of the state of a resource.

NOTIFY: A method that is used to notify a **Session Initiation Protocol (SIP)** client that an event requested by an earlier SUBSCRIBE method has occurred. The notification optionally provides details about the event.

novice: The side of a **Remote Assistance connection** that shares its screen with the other computer in order to receive help.

NT File System (NTFS): The native file system for Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2. For more information, see [MSFT-NTFS].

NT LAN Manager (NTLM) Authentication Protocol: A protocol using a challenge-response mechanism for **authentication** (2) in which clients are able to verify their identities without sending a password to the server. It consists of three messages, commonly referred to as Type 1 (negotiation), Type 2 (challenge) and Type 3 (authentication). For more information, see [MS-NLMP].

NTFS: See **NT File System (NTFS)**.

NTLM message: A message that carries **authentication** (2) 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, and NTLM AUTHENTICATE_MESSAGE.

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

numerical named property: A **named property** that has a numerical **name identifier**, which is stored in the LID field of a PropertyName structure.

17 O

OAB manifest: A file that contains information about data files in a 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 can receive all the information that is necessary to download any published data file in a specific WDP, as necessary.

OAB web distribution: A distribution mechanism that is specific to **offline address book (OAB)** version 4 and is used to publish OAB data files and an OAB manifest as a collection of files that a client application can download by using the HTTP/1.1 protocol, as described in [RFC2616].

OAL data sequence number: An integer that is associated with **offline address list (OAL)** data that represents the generation number of this data. The value of the initial sequence number is "1". Each subsequent data generation process that produces a data set that is not identical to the previous data set is incremented by one.

object: (1) A set of **attributes** (1), each with its associated values. Two attributes of an object have special significance: an identifying attribute and a parent-identifying attribute. An identifying attribute is a designated single-valued attribute that appears on every object; the value of this attribute identifies the object. For the set of objects in a replica, the values of the identifying attribute are distinct. A parent-identifying attribute is a designated single-valued attribute that appears on every object; the value of this attribute identifies the object's parent. That is, this attribute contains the value of the parent's identifying attribute, or a reserved value identifying no object. For the set of objects in a replica, the values of this parent-identifying attribute define a tree with objects as vertices and child-parent references as directed edges with the **child** as an edge's tail and the parent as an edge's head. Note that an object is a value, not a variable; a replica is a variable. The process of adding, modifying, or deleting an object in a replica replaces the entire value of the replica with a new value. As the word replica suggests, it is often the case that two replicas contain "the same objects". In this usage, objects in two replicas are considered the same if they have the same value of the identifying attribute and if there is a process in place (replication) to converge the values of the remaining attributes. When the members of a set of replicas are considered to be the same, it is common to say "an object" as shorthand referring to the set of corresponding objects in the replicas.

(2) In **Active Directory**, an **entity** consisting of a set of attributes, each attribute with a set of associated values. For more information, see [MS-ADTS].

(3) In **COM**, a software entity that implements the IUnknown interface and zero or more additional interfaces that may be obtained from each other using the IUnknown interface. A COM object can be exposed to remote clients via the DCOM protocol, in which case it is also a DCOM object.

(4) In the DCOM protocol, a software entity that implements one or more object remote protocol (ORPC) interfaces and which is uniquely identified, within the scope of an object exporter, by an **object identifier (OID)** (1). For more information, see [MS-DCOM].

(5) A set of attributes, each with its associated values. For more information on objects, see [MS-ADTS] section 1 or [MS-DRSR] section 1.

(6) In Active Directory, an entity consisting of a set of attributes, each attribute with a set of associated values. For more information, see [MS-ADTS]. See also **directory object**.

(7) In **COM**, a software entity that implements the IUnknown interface and zero or more additional interfaces that may be obtained from each other using the IUnknown interface. A

COM object can be exposed to remote clients via the DCOM protocol, in which case it is also a **DCOM object** (4).

(8) In the **DCOM** protocol, a software entity that implements one or more object remote protocol (ORPC) interfaces and which is uniquely identified, within the scope of an **object exporter**, by an **object identifier (OID)** (1). For more information, see [MS-DCOM].

(9) In COM, an instance of an object class. Each object implements one or more interfaces that may be obtained from each other by using the IUnknown interface.

(10) The root of the type hierarchy. For more information, see [ECMA-335].

(11) A file, email, email attachment, contact, calendar appointment or any other self-contained item that can be **indexed** and searched for by the **GSS**.

object class: (1) A predicate defined on **objects** (1) that constrains their **attributes** (1). Also an identifier for such a predicate.

(2) A set of restrictions on the construction and update of objects. An object class can specify a set of must-have attributes (every object of the class must have at least one value of each) and may-have attributes (every object of the class may have a value of each). An object class can also specify the allowable classes for the parent object of an object in the class. An object class can be defined by single-inheritance; an object whose class is defined in this way is a member of all object classes used to derive its most specific class. An object class is defined in a classSchema object.

(3) In **COM**, a category of **objects** (3) identified by a CLSID, members of which can be obtained through activation of the CLSID.

(4) In the DCOM protocol, a category of **objects** (4) identified by a CLSID, members of which can be obtained through activation of the CLSID. An object class is typically associated with a common set of interfaces that are implemented by all objects in the object class.

(5) A predicate defined on objects that constrains their attributes. Also an identifier for such a predicate.

(6) A set of restrictions on the construction and update of objects. An **object class** can specify a set of must-have attributes (every object of the class must have at least one value of each) and may-have attributes (every object of the class may have a value of each). An **object class** can also specify the allowable classes for the parent object of an object in the class. An **object class** can be defined by single inheritance; an object whose class is defined in this way is a member of all **object classes** used to derive its most specific class. An **object class** is defined in a classSchema object. See section 1 of [MS-ADTS] and section 1 of [MS-DRSR].

(7) In **COM**, a category of **objects** (3) identified by a **CLSID**, members of which can be obtained through **activation** of the **CLSID**.

(8) In the **DCOM** protocol, a category of **objects** (4) identified by a **CLSID**, members of which can be obtained through **activation** of the **CLSID**. An **object class** is typically associated with a common set of interfaces that are implemented by all **objects** in the **object class**.

object exporter: An object container (for example, process, machine, thread) in an object server. **Object exporters** are callable using RPC interfaces, and they are responsible for dispatching calls to the objects they contain.

object identifier (OID): (1) In the context of an object server, a 64-bit number that uniquely identifies an object.

(2) In the context of a directory service, a number identifying an object class or **attribute** (2). Object identifiers are issued by the ITU and form a hierarchy. An OID is represented as a dotted decimal string (for example, "1.2.3.4"). For more information on OIDs, see [X660] and [RFC3280] Appendix A. OIDs are used to uniquely identify certificate templates available to the **certification authority (CA)** (1). Within a **certificate** (1), OIDs are used to identify standard extensions, as described in [RFC3280] section 4.2.1.x, as well as non-standard extensions.

(3) In the Lightweight Directory Access Protocol (LDAP), a sequence of numbers in a format described by [RFC1778]. In many LDAP directory implementations, an OID is the standard internal representation of an attribute. In the directory model used in this specification, the more familiar ldapDisplayName represents an attribute.

(4) In the context of **ASN.1**, an object identifier, as described in [ITUX680].

(5) A variable-length identifier from a namespace administered by the ITU. Objects, protocols, and so on that make use of **ASN.1** or Basic Encoding Rules (BER), **Distinguished Encoding Rules (DER)**, or Canonical Encoding Rules (CER) encoding format leverage identities from the ITU. For more information, see [ITUX680].

(6) In the context of a directory service, a number identifying an object class or attribute. Object identifiers are issued by the ITU and form a hierarchy. An OID is represented as a dotted decimal string (for example, "1.2.3.4"). For more information on OIDs, see [X660] and Appendix A of [RFC3280]. **OIDs** are used to uniquely identify certificate templates available to the **certificate authority (CA)**. Within a certificate, **OIDs** are used to identify standard extensions as covered in [RFC3280] section 4.2.1.x, as well as non-standard extensions.

(7) In the Lightweight Directory Access Protocol (LDAP), a sequence of numbers in a format specified by [RFC1778]. In many LDAP directory implementations, an **OID** is the standard internal representation of an attribute. In the directory model used in [MS-ADTS], the more familiar ldapDisplayName represents an attribute.

(8) In the context of **Abstract Syntax Notation One (ASN.1)**, an object identifier, as specified in [ITUX680].

(9) A variable-length identifier from a namespace administered by the ITU. Objects, protocols, and so on that make use of ASN.1 or Basic Encoding Rules (BER), Distinguished Encoding Rules (DER), or Canonical Encoding Rules (CER) encoding format leverage identities from the ITU. For more information, see [ITUX680].

Object Linking and Embedding (OLE): A technology for transferring and sharing information between applications by inserting a file or part of a file into a compound document. The inserted file can be either embedded or linked. See also **embedded object** and **linked object**.

object of class x (or x object): An object o such that one of the values of its objectClass attributes is x. For instance, if objectClass contains the value user, o is an object of class user. This is often contracted to "user object".

object reference: (1) An attribute value that references an **object**. Reading a reference gives the **distinguished name (DN)** of the **object**.

(2) In the **DCOM** protocol, a reference to an **object** (4), represented on the wire as an **OBJREF**. An **object reference** enables the **object** to be reached by entities outside the **object's object exporter**.

(3) An **attribute** value that references an **object**; reading a reference gives the **distinguished name (DN)** or full **dsname** of the **object**.

OBJREF: The **marshaled** form of an object reference.

offline: (1) The condition of not being connected to or not being on a network or the Internet. Offline can also refer to a device, such as a printer that is not connected to a computer, and files that are stored on a computer that is not connected to or not on a network or the Internet.

(2) An operational state applicable to **volumes** and disks. In the offline state, the **volume** or disk is unavailable for data input/output (I/O) or configuration.

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

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

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

OleTx: A comprehensive distributed transaction manager processing protocol that uses the protocols specified in the following document(s): [MS-CMPO], [MS-CMP], [MS-DTCLU], [MS-DTCM], [MS-DTCO], [MC-DTCXA], [MS-TIPP], and [MS-CMOM].

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

One-Way Method: A **Remote Method** that has no application response sent from the implementation of the **Remote Method** back to the caller. This pattern is sometimes referred to as "fire and forget".

OOF message: A message that is sent in response to incoming messages and indicates that the user is currently **Out of Office (OOF)**.

opaque-signed message: An Internet email message that is in the format described by [RFC5751] and uses the SignedData CMS content type described in [RFC3852], or the **Message object** that represents such a message.

opnum: An operation number or numeric identifier that is used to identify a specific **remote procedure call (RPC)** method or a 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.

optional feature: A non-default behavior that modifies the **Active Directory** state model. An **optional feature** is enabled or disabled in a specific scope, such as a **forest** or a **domain**. For more information, refer to [MS-ADTS] section 3.1.1.9.

Organization object: An **Address Book object** that describes an entire organization.

organizer: The owner or creator of a conference or event.

originating update: An update that is performed to an **NC replica** via any protocol except replication. An **originating update** to an attribute or link value generates a new **stamp** for the attribute or link value.

orphan instance: An instance of an event that is in a **recurring series** and 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 the **free/busy status** on an appointment. It indicates that the user will not be in the office during the appointment.

Out of Office rule: A **rule** that is only evaluated when the mailbox is in an Out of Office state.

Outbox folder: A **special folder** that contains **Message objects** that are submitted to be sent.

outcome: One of the three possible results (Commit, Abort, In Doubt) reachable at the end of a life cycle for an **atomic transaction**.

outstanding RPC call: An asynchronous **remote procedure call (RPC)** that has not been completed by a server yet.

overdue reminder: An **active reminder** whose **signal time** has passed.

18 P

Packed Encoding Rules (PER): A set of encoding rules for **ASN.1** notation, specified in [ITUX691]. These rules enable the identification, extraction, and decoding of data structures.

padding: Bytes that are inserted in a data stream to maintain alignment of the protocol requests on natural boundaries.

page: (1) A file that consists of HTML and can include references to graphics, scripts, or dynamic content such as Web Parts.

(2) Represents the layout settings for page-oriented report rendering formats.

parent: A data item within the MDS system that can contain childmembers.

parent distinguished name (PDN): A **distinguished name (DN)** (1) of an object that is the next immediate object closer to the root of a tree of relative distinguished names (RDNs) (1).

participant: (1) An **actor** in an **activity flow**. A participant can be either an initiator or a target.

(2) A user who is participating in a **conference** or peer-to-peer **call**, or the object that is used to represent that user.

(3) Any of the parties that are involved in an **atomic transaction** and that have a stake in the operations that are performed under the **transaction** or in the **outcome** of the **transaction** ([WSAT10], [WSAT11]).

(4) A user who is participating in a **conference** or peer-to-peer call. May also be used in reference to the object that is used to represent this participant on the implementation.

(5) A machine that is accessing the desktop content **shared** by the **host**.

partition: (1) An area within a shared services database, such as an area that isolates different tenants within a service, or the process of creating such an area in a shared services database.

(2) A storage block that contains the content in binary files or metadata about file content.

(3) In the context of hard disks, a logical region of a hard disk. A hard disk may be subdivided into one or more **partitions**.

(4) In the context of **directory services**, a synonym for **directory partition** and naming context (NC) replica.

(5) A container for a specific configuration of a COM+ **object class**.

(6) A container for **conglomerations**. Every COMA server has at least one partition--the **Global Partition**--and may have additional partitions. A partition is identified by a **partition identifier**.

(7) One of the storage containers for data and aggregations of a cube. Every cube contains one or more partitions. For a cube with multiple partitions, each partition can be stored separately in a different physical location. Each partition can be based on a different data source. Partitions are not visible to users; the cube appears to be a single object.

partition identifier: A GUID that identifies a **partition** (1).

partner: (1) A computer connected to a local computer through either inbound or outbound connections.

(2) A participant in the MSDTC Connection Manager: OleTx Transports Protocol. Each **partner** has its own **contact identifier (CID)**, and uses the IXnRemote interface to invoke and receive **remote procedure calls (RPCs)**. The IXnRemote interface is described within the full **Interface Definition Language (IDL)** for [MS-CMPO] in section 6.

(3) A computer that is participating in **DFS-R** file replication.

(4) In the context of [MS-PASS], an organization in a business relationship with the **Authentication Service (AS)**. A **partner** needs to be able to access the **token** issued by the **AS**. Typically, a **partner** site is the actual service or site a consumer visits and, in the process, is **authenticated** by the **AS**. Examples of **partners** are the MSN Money and MSN Messenger sites.

peer: (1) An additional **endpoint** (5) that is associated with an endpoint in a session. An example of a peer is the **callee** endpoint for a **caller** endpoint.

(2) The entity being authenticated by the authenticator.

(3) In **DirectPlay**, a player within a DirectPlay game session that has an established connection with every other peer in the game session, and which is not performing game session management duties. The participant that is managing the game session is called the host.

(4) An instance of the Retrieval Protocol for the Peer Content Caching and Retrieval Framework running on a host. A peer can be both a client and a server in the Retrieval Protocol operations.

(5) A node participating in the content caching and retrieval system. A peer is a node that both accesses the content and serves the content it caches for other peers.

(6) The entity on either end of an established SMP session.

(7) A single device or node in a **peer-to-peer** networking system.

(8) When used in context with the IETF standard Layering 2 Tunnel Protocol (L2TP), as specified in [MS-L2TPIE], peer refers to either the LAC or LNS. LNS is a peer to LAC and vice versa.

(9) When used in context with [MS-PTPT], **peer** refers to either the **PAC** or **PNS**. A PAC's **peer** is a **PNS** and vice versa.

peer name: A string composed of an **authority** and a **classifier**. This is the string used by applications to resolve to a list of **endpoints** and/or an **extended payload**. A **peer name** is not required to be unique. For example, several **nodes** that provide the same service may register the same **Peer Name**.

peer-to-peer: A server-less networking technology that allows several participating network devices to share resources and communicate directly with each other.

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

permission: A rule that is associated with an object and that regulates which users can gain access to the object and in what manner. See also **rights**.

permissions list: A list of users and the **permissions** for each of those users.

permissions table: A **Table object** whose rows represent entries in a permissions list for a **Folder object**.

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

Personal Information Manager (PIM): A category of software packages for managing commonly used types of personal information, including contacts, email messages, calendar appointments, and meetings.

phishing: The luring of sensitive information, such as passwords or other personal information, from a recipient by masquerading as someone who is trustworthy and has a real need for such information.

phishing message: An email message that is designed to trick a recipient into divulging sensitive information, such as passwords or other personal information, to a non-trustworthy source.

Physical Media: The tangible media that are inserted into and removed from **libraries** and **mounted** in **drives**.

pipeline: An ordered collection of commands, with the output of one command passed as input to the next.

plain text: Text that does not have markup. See also **plain text message body**.

plain text message body: A **message body** (2) for which the Content-Type value of the Email Text Body header field is "text/plain". A plain text message body can be identified explicitly in the content, or implicitly if it is in a message that is as described in [RFC822] or a message that does not contain a Content-Type header field.

player: A person who is playing a computer game. There may be multiple players on a computer participating in any given game session. See also **name table**.

playlist: One or more **content** items that are **streamed** sequentially.

point: A unit of measurement for fonts and spacing. A point is equal to 1/72 of an inch.

policy: (1) A set of rules that governs all interactions with an object such as a document or item.

(2) The set of rules that govern the interaction between a subject and an object or resource.

(3) A collection of settings that contains global settings, profile settings, firewall rules, and connection security rules. Together these settings specify how the host firewall and **Internet Protocol security (IPsec)** behave on the client computer.

(4) The description of actions permitted for a specified set of **content**, and restrictions placed on those actions. Restrictions are described in the license associated with the **content**.

(5) A set of **conditions** and actions. The **conditions** provide a mechanism for classifying DHCP Clients. Classification is based on the **conditions** and **expressions** configured by the user as part of the **policy**. **DHCP Client** requests received by the server are evaluated as per the classification specified in the **policy**. The actions can have an associated IP address range and/or option values. If a **DHCP Client** request matches **policy** conditions, the client is given an IP address from the IP address range of the **policy**. The client will also be given options

configured for the matched **policy**. A **policy** can be configured at the **scope** or server level. Multiple policies can be configured at both the **scope** and server levels.

policy application: The protocol exchange by which a client obtains all of the **Group Policy Object (GPO)** and thus all applicable Group Policy settings for a particular policy target from the server, as specified in [MS-GPOL]. Policy application can operate in two modes, user policy and computer policy.

POP3 response: A message sent by a POP3 server in response to a message from a POP3 client. The structure of this message, as specified in [RFC1939], is as follows: <+OK> <response text><CR><LF> or <-ERR> <response text><CR><LF>.

port: (1) A TCP/IP numbered connection point that is used to transfer data.

(2) A logical name that represents a connection to a **device**. A **port** can represent a network address (for example, a TCP/IP address) or a local connection (for example, a **USB port**).

(3) A subcomponent of the naming **authority** in a **URI scheme** ([RFC3986] section 3.2.3).

(4) The abstraction that transport protocols use to distinguish among multiple destinations within a given host computer. TCP/IP protocols identify ports by using small positive integers. The transport selectors (TSEL) used by the OSI transport layer are equivalent to ports. **RTP** depends upon the lower-layer protocol to provide some mechanism such as ports to multiplex the **RTP** and **RTCP packets** of a session. For more information, see [RFC3550] section 3.

(5) A place to add or remove **physical media** from a **library**.

(6) The logical endpoint of a remote access connection on the client or server.

Portable Network Graphics (PNG): A bitmap graphics file format that uses lossless data compression and supports variable transparency of images (alpha channels) and control of image brightness on different computers (gamma correction). PNG-format files have a .png file name extension.

Post object: A **Message object** that represents an entry in a discussion thread stored in a messaging store.

Post Office Protocol - Version 3 (POP3): A protocol that is used for accessing email from mail servers, as described in [RFC1939].

postmark: A computational proof that is applied to outgoing messages to help recipient messaging systems distinguish legitimate email messages from junk email messages, which reduces the chance of false positives.

PPTP: Point-to-Point Tunneling Protocol (PPTP) Profile, as defined in [MS-PTPT].

PPTP Access Concentrator (PAC): A node that acts as one side of a PPTP tunnel endpoint and is a **peer** to the **PPTP Network Server (PNS)**. **PAC** refers to the **server** that terminates the PPTP tunnel and provides VPN connectivity to a remote **client**.

PPTP Network Server (PNS): A node that acts as one side of a PPTP tunnel endpoint and is a **peer** to the **PPTP Access Concentrator (PAC)**. **PNS** refers to the remote **client** that requests to establish a VPN connectivity using PPTP tunnel.

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

presence: (1) A status indicator on a client device that is transmitted by using the Wide Area Network Device Presence Protocol (WAN DPP).

(2) A setting for the User field that determines whether instant-messaging status information appears with user names in that field.

presentation: (1) A collection of slides that are intended to be viewed by an audience.

(2) A set of audio and video data **streams** and related metadata that are synchronized for playback on a client.

presolution header: A string that contains the prepended solutions for the puzzle.

Pre-Solver: A component that, given specific inputs, generates a message **postmark**.

primary calendar: The **calendar** that contains free/busy information for a specific user or resource. It enables a user or resource to schedule their appointments and other types of events, and the **Calendar objects** within it are used to process and respond to meeting requests.

primary flag storage location: The typical location that is used to store flagging properties, as opposed to the **secondary flag storage location**.

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

primary SMTP proxy address: The **Simple Mail Transfer Protocol (SMTP)** email address that is used to designate a message server user in all SMTP traffic. Proxy addresses are stored in a user's **address book** entry, in the PidTagAddressBookProxyAddresses multivalued string property. The primary SMTP proxy address can be identified by its **address type** field, which is set to "SMTP" (uppercase). Non-primary SMTP proxy addresses have the **address type** field set to "smtp" (lowercase).

principal: (1) An authenticated **entity** that initiates a message or channel in a distributed system.

(2) An **identifier** of such an entity.

(3) In **Kerberos**, a **Kerberos principal**.

(4) An authenticated entity that initiates a message or channel in a distributed system.

(5) An ID of such an entity.

(6) In Kerberos, a Kerberos principal.

(7) A unique entity identifiable by a **security identifier (SID)** that is typically the requester of access to securable **objects** or resources. It often corresponds to a human user but can also be a computer or service. It is sometimes referred to as a **security principal**.

(8) A unique, individual account known to the **KDC**. Often a user, but it can be a **service** offering a resource on the network.

print client: The application or user that is trying to apply an operation on the print system either by printing a job or by managing the data structures or devices maintained by the print system.

print server: A machine that hosts the print system and all its different components.

private key: One of a pair of keys used in public-key cryptography. The private key is kept secret and is used to decrypt data that has been encrypted with the corresponding public key. For an introduction to this concept, see [CRYPTO] section 1.8 and [IEEE1363] section 3.1.

privilege attribute certificate (PAC): A Microsoft-specific authorization data present in the authorization data field of a ticket. The **PAC** contains several logical components, including group membership data for authorization, alternate credentials for non-Kerberos authentication protocols, and policy control information for supporting interactive logon.

procedural module: A collection of subroutines and functions.

process matching criteria (PMC): A **resource policy object** that selects a subset of currently executing processes. Since processes are dynamically created and terminated by the operating system in the course of running workloads, the WSRM Protocol uses PMCs as a means of identifying processes for **resource management** purposes. PMCs specify partial or full values to be matched against process property fields. Each PMC includes a name and a nonempty set of matching values and can also include a nonempty set of exclusion values. All running processes under management whose path and the associated user name match the values provided in a PMC are selected by that PMC, provided that they are not already selected by another PMC and do not match the exclusion values. Processes selected by a PMC specification at any given time are said to match, or be in, the PMC. A process can be selected by only one PMC at a time. The term **resource group**" and PMC are used interchangeably.

processor affinity: An element of **process matching criteria (PMC)**, the association between a task or process and a specific processor needed to execute that task. Processor affinity takes advantage of the fact that some remnants of a process might remain in one processor's state (in particular, in its cache) from the last time the process ran, and so scheduling it to run on the same processor the next time could make the process run more efficiently than if it were to run on another processor.

property: (1) A data field within a **Common Information Model (CIM)** class definition. This consists of a simple name, a type, and a value.

(2) A typed value associated with a property identifier and optionally a property name.

(3) An EntityType or ComplexType can have one or more properties of the specified EDMSimpleType or ComplexType. A property of an EntityType can be a **declared property** or a **dynamic property**, as specified in [MC-CSDL]. A property of ComplexType can only be a declared property.

(4) A name/value pair that associates metadata with a resource. This term is used as specified in [RFC4918] section 4.

(5) A name/value pair that describes a unit of data for a class. **Property** values must have a valid **Managed Object Format (MOF)** data type.

Property: A typed name/value pair that is associated with a **MetadataObject**. Properties enable consumers of a protocol client to annotate or decorate the MetadataObject with consumer-specific extensions. A MetadataObject can contain multiple Properties.

property ID: A 16-bit numeric identifier of a specific **attribute** (1). A property ID does not include any **property type** information.

property identifier: (1) A unique integer or a 16-bit, numeric identifier that is used to identify a specific **attribute** (1) or property.

(2) A numerical value that uniquely identifies a property in a property set.

(3) A DWORD value associated with an **MSMQ object** property that defines the property type and its semantic meaning.

property name: A string that, in combination with a **property set**, identifies a **named property**.

property set: (1) A set of **attributes** (1), identified by a **GUID**. Granting access to a property set grants access to all the attributes in the set.

(2) A set of attributes, identified by a **GUID**. Granting access to a property set grants access to all the attributes in the set.

(3) A set of properties, along with an FMTID, identifying the property set format and an associated class identifier (CLSID). The CLSID is used to identify the application or component that created the property set.

property tag: A 32-bit value that contains a property type and a property ID. The low-order 16 bits represent the property type. The high-order 16 bits represent the property ID.

property type: A 16-bit quantity that specifies the data type of a property value.

protected content: (1) Any content or information, such as a file, Internet message, or other object type, to which a rights-management usage policy is assigned and is encrypted according to that policy. See also **Information Rights Management (IRM)**.

(2) **Content** for which usage is governed by policies specified in a license.

(3) Any media content that has a DRM usage policy assigned to it, and is encrypted according to that policy.

(4) Any content or information (file, email) that has an RMS usage policy assigned to it, and is encrypted according to that policy. Also known as "Protected Information".

protocol stream: A continuous stream of records flowing in one direction.

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

public key: One of a pair of keys used in public-key cryptography. The public key is distributed freely and published as part of a **digital certificate**. For an introduction to this concept, see [CRYPTO] section 1.8 and [IEEE1363] section 3.1.

public queue: An application-defined **message queue** that is registered in the **MSMQ Directory Service**. A public queue may be deployed at any **queue manager**.

publisher: (1) A **SIP protocol client** that is making a publish request.

(2) An application that needs to publish historical data that may be of interest to other applications.

(3) The side of a **Remote Assistance connection** that registers a **Peer Name**. It is the same as the **novice** role.

(4) A set of resources that are contained in the same **workspace**.

(5) In the context of events: The source of event generation. An application or component that writes to one or more **event logs**. An application that publishes events.

publishing license: An XrML 1.2 license that defines the 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 that they are authorized to take with the content, in addition to any usage conditions. The publishing license tells a server which usage policies apply to a specific piece of content and grants a server the right to issue Use Licenses (Uls) based on that policy. The publishing license is created when content is protected. Also referred to as "Issuance License (IL)."

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).

publishing range: The number of months of free/busy calendar data to be published, beginning at the start date of the publishing range, which is defined by the PidTagFreeBusyPublishStart property, and continuing for the number of months defined by the PidTagFreeBusyCountMonths property.

pure MIME message: A **MIME** representation of an email message that does not contain a **Transport Neutral Encapsulation Format (TNEF)** body part.

19 Q

query server: A server that has been assigned the task of fulfilling search queries.

queue: An object that holds **messages** passed between applications or **messages** passed between **Message Queuing** and applications. In general, applications can send **messages** to queues and read **messages** from queues.

queue manager: A message queuing service that manages **queues** deployed on a computer. A queue manager may also provide asynchronous transfer of **messages** to **queues** deployed on other queue managers.

quick discovery: The process of discovering **responders** on a network.

quick mode (QM): The second phase of an **Internet Key Exchange (IKE)** negotiation, during which the peers negotiate quick mode security associations (quick mode security association (QM SA)). For more information, see [RFC2409] section 5.5.

qWave-WD: The Quality Windows Audio/Video Experience (qWave): Wireless Diagnostics Protocol.

20 R

RADIUS client: A client that is responsible for passing user information to designated RADIUS servers, and then acting on the response that is returned.

RADIUS server: A server that is responsible for receiving user connection requests, authenticating the user, and then returning all configuration information necessary for the client to deliver service to the user. A RADIUS server can act as a proxy client to other RADIUS servers or other kinds of authentication servers.

read receipt: An email message that is sent to the sender of a message to indicate that a message recipient received the message.

Really Simple Syndication (RSS): An XML-based syndication format for content, as described in [RSS2.0].

realm: (1) An administrative boundary that uses one set of authentication servers to manage and deploy a single set of unique identifiers. A realm is a unique logon space.

(2) A collection of key distribution centers (KDCs) with a common set of principals, as described in [RFC4120] section 1.2.

(3) A collection of **users**, **partners**, and **authentication servers** bound by a common **authentication** policy.

realm autodiscovery: A process used by client applications to obtain the name of a server resource's source **realm** (1) and then use that information to locate a **security token service (STS)** that can issue access tokens to the resource.

Real-Time Transport Protocol (RTP): A network transport protocol that provides end-to-end transport functions that are suitable for applications that transmit real-time data, such as audio and video, as described in [RFC3550].

Receive folder: A **Folder object** that is configured to be the destination for email messages that are delivered.

recipient: (1) An entity that can receive email messages.

(2) An entity that is in an **address list**, can receive email messages, and contains a set of **attributes** (1). Each attribute has a set of associated values.

(3) The **recipient** of a **fax message**.

recipient flag: A collection of property values indicating that a draft **Message object** is marked such that it will appear as flagged with a **basic flag** to recipients.

recipient information cache: An information store that contains a list of the **contacts** (3) with whom a user has interacted most often and most recently, and with whom the user is likely to interact again.

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

recipient reminder: A collection of property values indicating that a **Draft Message object** is marked such that it will have an **active reminder** for the recipients of the Message Object.

recipient table: The part of a **Message object** that represents users to whom a message is addressed. Each row of the table is a set of properties that represents one **recipient** (2).

record: (1) A group of related **fields** (3), which are sometimes referred to as columns, of information that are treated as a unit. Also referred to as row.

(2) The fundamental unit of information in the .NET Binary Format: XML Data Structure encoded as a variable length series of bytes. [MC-NBFX] section 2 specifies the format for each type of **record**.

(3) A variable-length sequence of bytes with a predefined structure.

(4) A sequence of octets.

(5) The data structure that contains an **event** that is currently represented in an **event log**.

(6) A piece of data that is published by a **node** to the **graph**. **Records** are the primary mechanism of communication in a **graph**.

recurrence BLOB: The **binary large object (BLOB)** encoding of a recurrence pattern, a recurrence range, and recurrence exceptions.

recurrence part: A name/value pair in a property of type Recurrence Rule, as described in [RFC2445].

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

Recurring Calendar object: A **Calendar object** that describes an event that repeats according to a recurrence pattern.

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

recurring task: A series of **Task objects** that are described by a recurrence pattern.

Recurring Task object: A **Task object** that represents a recurring task.

Recycle Bin: (1) The location where deleted files are stored until they are either restored, if they were deleted erroneously, or destroyed permanently.

(2) An **optional feature** that modifies the state model of object deletions and undeletions, making undeletion of **deleted-objects** possible without loss of the object's attribute values. For more information, see [MS-ADTS] section 3.1.1.9.1.

recycled-object: An **object** that has been deleted, but remains in storage until a configured amount of time (the **tombstone lifetime**) has passed, after which the **object** is permanently removed from storage. Unlike a **deleted-object**, most of the state of the **object** has been removed, and the **object** may no longer be undeleted without loss of information. By keeping the **recycled-object** in existence for the **tombstone lifetime**, the deleted state of the **object** is able to replicate. **Recycled-objects** exist only when the **Recycle Bin optional feature** is enabled.

registry: A local system-defined database in which applications and system components store and retrieve configuration data. It is a hierarchical data store with lightly typed elements that are logically stored in tree format. Applications use the registry API to retrieve, modify, or delete registry data. The data stored in the registry varies according to the version of Windows.

relative distinguished name (RDN): (1) An attribute-value pair used in the distinguished name of an object. For more information, see [RFC2251].

(2) In the **Active Directory** directory service, the unique name of a child element relative to its parent in Active Directory. The RDN of a child element combined with the **fully qualified domain name (FQDN)** (2) of the parent forms the FQDN of the child.

(3) The name of an **object** relative to its parent. This is the leftmost attribute-value pair in the **distinguished name (DN)** of an **object**. For example, in the **DN** "cn=Peter Houston, ou=NTDEV, dc=microsoft, dc=com", the **RDN** is "cn=Peter Houston". For more information, see [RFC2251].

(4) In the **Active Directory** directory service, the unique name of a child element relative to its parent in Active Directory. The RDN of a child element combined with the fully qualified distinguished name (FQDN) of the parent forms the FQDN of the child.

(5) As specified in [X500], the portion of a distinguished name that is unique to an organization unit but might not be unique inside a domain.

relative identifier (RID): The last item in the series of **SubAuthority** values in a SID (as specified in [SIDD]). It distinguishes one account or group from all other accounts and groups in the domain. No two accounts or groups in any domain share the same relative identifier.

reliable messaging destination (RMD): An endpoint that receives a message. For more information, see [WSRM1-0], [WSRM1-1], and [WSRM1-2].

reliable messaging source (RMS): An endpoint that sends a message. For more information, see [WSRM1-0], [WSRM1-1], and [WSRM1-2].

relying party (RP): (1) The entity (person or computer) using information from a certificate in order to make a security decision. Typically, the RP is responsible for guarding some resource and applying access control policies based on information learned from a certificate.

(2) A web application or service that consumes **security tokens** issued by an **security token service (STS)**.

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 it can be applied to any object type.

reminder domain: A set of folders that are searched for objects that have an **active reminder**.

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 a reminder is signaled or displayed.

reminder queue: A sorted list of objects that are in a reminder domain and have been stamped with properties implying that they could have an **active reminder**.

Remote Administration Protocol (RAP): A synchronous request/response protocol, used prior to the development of the remote procedure call (RPC) protocol, for marshaling and unmarshaling procedure call input and output arguments into messages and for reliably transporting messages to and from clients and servers.

remote application: An application running on a remote server.

Remote Assistance (RA): A feature of the operating system that allows screen, keyboard, and mouse sharing so that a computer user can be assisted by a remote helper.

Remote Assistance connection: A communication framework that is established between two computers that facilitates **Remote Assistance**.

Remote Authentication Dial-In User Service (RADIUS): A protocol for carrying authentication, authorization, and configuration information between a **network access server (NAS)** that prefers to authenticate connection requests from endpoints and a shared server that performs authentication, authorization, and accounting.

remote differential compression (RDC): Any of a class of compression algorithms that are designed to compare two files residing on different machines without requiring one of the files to be transmitted in its entirety to the other machine. For more information, see [MS-RDC].

Remote Method: Part of the **Remoting Data Model**. A **Remote Method** is a remotely callable operation. A **Remote Method** can either be One-Way or **Two-Way**. In the case of a **One-Way Method**, there is no reply from the implementation. For more information, see [MS-NRTP] section 3.1.1

remote operation (ROP): An operation that is invoked against a server. Each ROP represents an action, such as delete, send, or query. A ROP is contained in a **ROP buffer** for transmission over the wire.

remote procedure call (RPC): A context-dependent term commonly overloaded with three meanings. Note that much of the industry literature concerning RPC technologies uses this term interchangeably for any of the three meanings. Following are the three definitions: The runtime environment providing remote procedure call facilities. The preferred usage for this meaning is "RPC runtime". The pattern of request and response message exchange between two parties (typically, a client and a server). The preferred usage for this meaning is "RPC exchange". A single message from an exchange as defined in the previous definition. The preferred usage for this term is "RPC message". For more information, see [C706].

remote user: (1) A user who has a persistent identity within an enterprise and is connected from outside the enterprise network boundary.

(2) An **Address Book object** known to be from a foreign or remote messaging system.

Remoting Data Model: A model that is used to represent higher-layer–defined data structures and values, and to represent a **Remote Method** invocation and the **Return Value** or error information from that invocation. A protocol, such as [MS-NRLS], that is built on top of this protocol can be defined by using the **Remoting Data Model**, and can be agnostic to the **serialization format**. For more information, see Abstract Data Model (section 3.1.1).

Remoting Type: Part of the **Remoting Data Model**. **Class**, **Array**, **Enum**, and Primitive are different kinds of **Remoting Types**. All **Remoting Types** are identified by a name that is case sensitive. For more information, see [MS-NRTP] section 3.1.1

rendering position: A location in a **Rich Text Format (RTF)** document where an **attachment** is placed visually.

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

(2) A copy of the data that is in a user's **mailbox** at a specific point in time.

(3) A variable containing a set of objects.

(4) A particular repository of file and directory information to be synchronized, and the metadata store that represents that repository.

(5) **NBNS** database **name records** (name-to-IPv4 address mapping) replicated from other **NBNS servers**.

(6) A set of data together with associated synchronization metadata.

replica GUID (REPLGUID): A value that represents a namespace for identifiers. If a REPLGUID is combined with a GLOBSET, the result is a set of **global identifiers**. A REPLGUID value has an associated **replica ID (REPLID)** that is used in its place on disk and on the wire.

replica ID (REPLID): A value that is mapped to a **replica GUID (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.

replicated update: An update performed to a naming context (NC) replica by the **replication** system, to propagate the effect of an **originating update** at another **NC replica**. The **stamp** assigned during the **originating update** to attribute values or a link value is preserved by **replication**.

replication: (1) The process of propagating the effects of all originating writes to any replica of a **naming context (NC)**, to all replicas of the **NC**. If originating writes cease and replication continues, all replicas converge to a common application-visible state.

(2) An administration scenario in which a **replication client application** automatically copies multiple **conglomerations** from a **replication source** to one or more **replication targets**.

replication client application: A client application that provides automatic copying of **conglomerations** between COMA servers, typically using COMA **export** and **import** functionality.

replication source: A COMA server whose **catalog** contains **conglomerations** to be copied.

replication target: A COMA server whose **catalog** is to contain the copied **conglomerations** after **replication** is performed.

report: (1) A database object that provides a static representation of a set of data and can be used to group, sort, summarize, and aggregate data. The data in a report cannot be edited.

(2) An object that is a combination of three kinds of information: data or other kinds of information about how to obtain the data (queries) as well as the structure of the data; layout or formatting information that describes how the data is presented; and properties of the report, such as author of the report, report parameters, and images included in the report.

request: (1) A **SOAP** message with additional constraints as specified in [MS-WSRVCRR] section 2.2.1.

(2) A message from a **client** to an OCSP **responder**. The message requests the **revocation** status of an X.509 **certificate** (see [RFC2560]).

(3) An HTTP message sent from the client to the server, as defined in [RFC2616].

required attendee: An attendee of an event whom the organizer lists as a mandatory participant.

resend message: A message that is submitted for message delivery after it failed to be sent to all or some of its **recipients** (1).

resource: (1) Any component that a computer can access where data can be read, written, or processed. This resource could be an internal component such as a disk drive, or another computer on a network that is used to access a file.

(2) A logical entity or unit of data whose state changes in accordance with the **outcome** of an **atomic transaction**. **Resources** are either durable or volatile.

(3) Any physical or logical component that can be managed by a **cluster**. A resource is owned by a single **node** at any one time.

(4) An endpoint that represents a distinct type of management operation or value. A client exposes one or more resources and some resources can have more than one instance. For example, the Win32_LogicalDisk class represents a resource and Win32_LogicalDisk="C:\" is a specific instance of the resource.

(5) A network-accessible data object or service that is identified by an **IRI**, as defined in [RFC2616].

(6) An object that a **client** is requesting access to, typically referenced by a Uniform Resource Locator (**URL**) or Uniform Resource Identifier (**URI**), as specified in [RFC3986].

(7) An object created and retained by the composition engine running on the client, on behalf of the server. Resources are referenced in the protocol via handles. Resource handles are scoped to the channel on which they are created. The server may create multiple resources per channel.

(8) An **entity** that can be identified by a URI. This term is used as specified in [RFC2616] section 1.3.

(9) In WS-Transfer, resources are entities that have an XML representation and can be addressed by an **endpoint** reference [WXFR].

resource allocation: The part of a **RAP** that specifies the part of a computer's hardware resources that can be allocated to the processes in a **PMC**. A resource allocation can specify any of the following: a percentage of processor bandwidth, a **processor affinity** mask, an amount of physical memory, or an amount of virtual memory. A resource allocation includes a specification of at least one resource.

resource allocation policy (RAP): A named specification for allocating computer resources to all of the managed processes on a computer. A **RAP** specifies how the managed resources of a computer can be divided among managed processes running on the computer by providing an ordered list of **PMC** names, each with an associated **resource allocation**. All processes not matched by any **PMC** named in the list and not included in the **exclusion list** are selected into an implicit "residual" **PMC**.

resource group: (1) A security or distribution group that can contain universal groups, global groups, other domain local groups from its own domain, and accounts from any domain in the forest. Resource groups can be granted rights and permissions on resources that reside only in the same domain where the domain local group is located.

(2) A group object whose membership is added to the authorization context only if the server receiving the context is a member of the same domain as the resource group.

(3) Used interchangeably with **process matching criteria (PMC)** in Windows System Resource Manager (WSRM) Protocol [MS-WSRM].

resource management: A method of allocating the hardware resources of a computer to tasks being performed on that computer. It includes a system of **accounting** for hardware resources by task. The purpose of the WSRM Protocol is to control the resource management of a computer.

resource manager (RM): The participant that is responsible for coordinating the state of a resource with the outcome of atomic transactions. For a specified transaction, a resource manager enlists with exactly one transaction manager to vote on that transaction outcome and to obtain the final outcome. A resource manager is either durable or volatile, depending on its resource.

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

resource policy object: A persistent object that is maintained by the **management service**, created by a **management client**, or built-in to the **management service**. Resource policy objects specify the desired **resource management** behavior of the computer whose resources are under management.

responder: (1) The computer that responds to request messages.

(2) The party that responds to the first message of an AuthIP exchange.

(3) The party that responds to the first message of an IKE exchange.

(4) An OCSP Extensions server that provides OCSP **responses** (see [RFC2560]).

(5) An LLTD–capable **station** to which **mappers** and **enumerators** send LLTD commands.

response: (1) A **SOAP** message with additional constraints as specified in [MS-WSRVCRR] section 2.2.2.

(2) A message from an OCSP **responder**. The message specifies the status of an X.509 **certificate** (see [RFC2560]).

(3) An HTTP message sent from the server to the client, as defined in [RFC2616].

(4) A typed buffer of data sent by the client to the server in response to a **notification**.

response message: (1) A Traversal Using Relay NAT (TURN) message that is sent from a protocol server to a protocol client in response to a request message. It is sent when the request message is handled successfully by the protocol server.

(2) A **SOAP** message with additional constraints as specified in Response Message (section 2.2.2).

restriction: (1) A set of conditions that an item meets to be included in the search results that are returned by a query server in response to a search query.

(2) 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 **Table objects** or to define new ones, such as **search folder** (2) or rule criteria.

(3) A set of conditions that a file must meet to be included in the search results returned by the Generic Search Service (GSS) in response to a search query. A restriction narrows the focus of a search query, limiting the files that the Generic Search Service (GSS) will include in the search results only to those files matching the conditions.

(4) A set of conditions that a file must meet to be included in the search results returned by the indexing service in response to a search query. A restriction narrows the focus of a search query, limiting the files that the indexing service includes in the search results only to those files matching the conditions.

retention policy: A policy that specifies the length of time during which data, documents, and other records must be available for recovery.

retention tag: An element that contains information about the **retention policy** of a **Message object** or folder.

Return Value: A **Data Value** that is returned as part of the results of a **Remote Method** invocation. For more information, see **Remote Method** in Abstract Data Model (section 3.1.1).

ReturnTypeDescriptor: An **attribute** (1) of a **MethodInstance**. It is the TypeDescriptor that identifies the portion of a Method's return or output Parameters to extract and return as the result of executing the MethodInstance. It defines the View of the EntityInstances returned, with its child TypeDescriptors denoting the Fields of the View.

revocation: The process of invalidating a certificate. For more details, see [RFC3280] section 3.3.

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

rights: Tasks that a user is permitted to perform on a computer, site, domain, or other system resource. See also **permission**.

rights policy template: An XrML 1.2 document that contains a predefined usage policy that is used to create the **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**.

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

RMD: See **reliable messaging destination (RMD)**.

RMS: See **reliable messaging source (RMS)**.

Root folder: The **special folder** that is the top-level folder in a message store hierarchy. It contains all other **Folder objects** in that message store.

root license: A license whose rules can be further restricted or augmented by terms in a **leaf license**. A **root license** can be more or less restrictive than a **leaf license**.

ROP buffer: A structure containing an array of bytes that encode a **remote operation (ROP)**. The first byte in the buffer identifies the ROP. This byte is followed by ROP-specific fields. Multiple ROP buffers can be packed into a single **remote procedure call (RPC)** request or response.

ROP request: See **ROP request buffer**.

ROP request buffer: A **ROP buffer** that a client sends to a server to be processed.

ROP response: See **ROP response buffer**.

ROP response buffer: A **ROP buffer** that a server sends to a client to be processed.

row: (1) A collection of **columns** (1) that contains property values that describe a single item in a set of items that match the **restriction** (1) specified in a query.

(2) A single set of data that is displayed horizontally in a worksheet or a table.

(3) The collection of columns containing the property values that describe a single result from the set of objects that matched the restrictions specified in the search query submitted to the Generic Search Service (GSS).

(4) The collection of columns that contains the property values that describe a single file from the set of files that matched the restriction specified in the search query submitted to the indexing service

RPC dynamic endpoint: A network-specific server address that is requested and assigned at run time, as described in [C706].

RPC protocol sequence: A character string that represents a valid combination of a **remote procedure call (RPC)** protocol, a network layer protocol, and a transport layer protocol, as described in [C706] and [MS-RPCE].

RPC transport: The underlying network services used by the remote procedure call (RPC) runtime for communications between network nodes. For more information, see [C706] section 2.

RSS item: An item element in an RSS feed, as described in [RSS2.0].

RSS object: A Message object that represents an entry from an **RSS item** or **atom feed**.

RTCP packet: A control packet consisting of a fixed header part similar to that of **RTP packets**, followed by structured elements that vary depending upon the RTCP packet type. Typically, multiple RTCP packets are sent together as a compound RTCP packet in a single packet of the underlying protocol; this is enabled by the length field in the fixed header of each RTCP packet. See [RFC3550] section 3.

RTP packet: A data packet consisting of the fixed RTP header, a possibly empty list of contributing sources, and the payload data. Some underlying protocols may require an encapsulation of the RTP packet to be defined. Typically one packet of the underlying protocol contains a single RTP packet, but several RTP packets can be contained if permitted by the encapsulation method. See [RFC3550] section 3.

RTP session: An association among a set of **participants** (2) who are communicating by using the **Real-Time Transport Protocol (RTP)**, as described in [RFC3550]. Each RTP session maintains a full, separate space of **Synchronization Source (SSRC)** identifiers.

rule: (1) A condition or action, or a set of conditions or actions, that performs tasks automatically based on events and values.

(2) A set of qualifiers, such as enumeration values, and quantifiers, such as numeric arguments, that are specified as usage guidelines for a set of objects or data.

(3) A mapping of a file type to a location in a document repository.

(4) An item that defines a **condition** and an action. The condition is evaluated for each **Message object** as it is delivered, and the action is executed if the new Message object matches the condition.

Rule FAI message: A **folder associated information (FAI)** message stored in the Inbox special folder where the client can store extra rule-related information that is opaque to the server.

rules table: A **Table object** whose rows represent the rules that are contained in a **Folder object**.

runspace: An entity capable of running one (and only one) **pipeline** of **commands**.

21 S

S/MIME (Secure/Multipurpose Internet Mail Extensions): A set of cryptographic security services, as described in [RFC5751].

SASL: The Simple Authentication and Security Layer, as described in [RFC2222]. This is an **authentication** (2)mechanism used by the **Lightweight Directory Access Protocol (LDAP)**.

Scale Secure Real-Time Transport Protocol (SSRTP): A Microsoft proprietary extension to the **Secure Real-Time Transport Protocol (SRTP)**, as described in [RFC3711].

schema: (1) The set of **attributes** and **object classes** that govern the creation and update of **objects**.

(2) A container that defines a **namespace** that describes the scope of EDM types. All EDM types are contained within some **namespace**.

schema naming context (schema NC): A specific type of **naming context (NC)** or an instance of that type. A **forest** has a single **schema NC**, which is replicated to each **domain controller (DC)** in the **forest**. No other **NC replicas** can contain these **objects**. Each **attribute** and class in the **forest's** schema is represented as a corresponding **object** in the **forest's schema NC**.

scheme: The name of a specification to refer to when assigning identifiers within a particular **URI scheme**, as defined in [RFC3986] section 3.1.

scope: (1) A range of IP addresses and associated configuration options that are allocated to **DHCP clients** in a specific subnet.

(2) The term "Scope" that is defined in [WS-Discovery1.1].

(3) An item that represents a hierarchy in a **report**. There are explicit scopes (such as **data region**, **dataset**, group) and implicit scopes (such as report scope). At any level in the hierarchy, there can be only one ancestor scope (except for the top-level report scope and the **page** scope) but an unlimited number of descendants as well as peer scopes.

search criteria: A criteria used to determine which messages are included in a folder with specific characteristics. It is composed of a restriction, which is the filter to be applied, and a search scope, which are the folders that contain the content to search.

search folder: (1) A collection of related items to be crawled by a search service.

(2) A **Folder object** that 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: A **Folder object** that is created according to the specifications in the definition message. It is in the Finder folder of the message database.

search folder definition message: A **folder associated information (FAI)** message that persists all the information that defines a search folder. It is in the associated contents table of the **Common Views folder** in the message database.

search key: A binary-comparable key that identifies related objects for a search.

search template: A template that defines a dialog box which enables users to specify search criteria for **Address Book objects**.

secondary data source: An **XML** data file, a database, or a **web service** that is used to populate controls or provide values in an InfoPath form.

secondary flag storage location: A binary property that is used to encode a second set of flagging properties, which do not affect the flagged state of a **Message object**.

secret key: A symmetric encryption key shared by two entities, such as between a user and the **domain controller (DC)**, with a long lifetime. A password is a common example of a secret key. When used in a context that implies **Kerberos** only, a principal's secret key.

Secure Real-Time Transport Protocol (SRTP): A profile of **Real-Time Transport Protocol (RTP)** that provides encryption, message **authentication** (2), and replay protection to the RTP data, as described in [RFC3711].

Secure Sockets Layer (SSL): A security protocol that supports confidentiality and integrity of messages in client and server applications that communicate 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** (2) using **X.509 certificates** (2). 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 association (SA): A simplex "connection" that provides security services to the traffic carried by it. See [RFC4301] for more information.

security descriptor: A data structure containing the security information associated with a securable **object**. A **security descriptor** identifies an **object's** owner by its **security identifier (SID)**. If access control is configured for the **object**, its **security descriptor** contains a **discretionary access control list (DACL)** with **SIDs** for the **security principals** who are allowed or denied access. Applications use this structure to set and query an **object's** security status. The **security descriptor** is used to guard access to an **object** as well as to control which type of auditing takes place when the **object** is accessed. The **security descriptor** format is specified in [MS-DTYP] section 2.4.6; a string representation of **security descriptors**, called SDDL, is specified in [MS-DTYP] section 2.5.1.

security identifier (SID): An identifier for **security principals** in Windows that is used to identify an account or a group. Conceptually, the **SID** is composed of an account authority portion (typically a **domain**) and a smaller integer representing an identity relative to the account authority, termed the **relative identifier (RID)**. The **SID** format is specified in [MS-DTYP] section 2.4.2; a string representation of **SIDs** is specified in [MS-DTYP] section 2.4.2 and [MS-AZOD] section 1.1.1.2.

security principal: (1) A unique entity that is identifiable through cryptographic means by at least one key. It frequently corresponds to a human user, but also can be a service that offers a resource to other security principals. Also referred to as principal.

(2) An identity that can be used to regulate access to resources. A security principal can be a user, a computer, or a group that represents a set of users.

(3) A unique entity identifiable through cryptographic means by at least one **key**. A **security principal** often corresponds to a human user but can also be a service offering a resource to other **security principals**. Sometimes referred to simply as a "principal".

(4) An identity that can be used to regulate access to resources, as specified in [MS-AUTHSOD] section 1.1.1.1. A **security principal** can be a user, a computer, or a group that represents a set of users.

(5) A unique entity, also referred to as a principal, that can be authenticated by **Active Directory**. It frequently corresponds to a human user, but also can be a service that offers a resource to other security principals. Other security principals might be a group, which is a set of principals. Groups are supported by **Active Directory**.

(6) An entity that is associated with a human user or a program that can be authenticated. At a minimum, it has two basic attributes, a name and an identifier, that uniquely identifies it and makes it meaningful to the system, administrators, and users. A security principal is also known as a principal or an account.

security principal identifier: A value that is used to uniquely identify a **security principal** (2). In Windows-based systems, it is a **security identifier (SID)**. In other types of systems, it can be a user identifier or other type of information that is associated with a **security principal** (2).

security principal object: An **object** that corresponds to a **security principal**. A **security principal object** contains an identifier, used by the system and applications to name the principal, and a secret that is shared only by the principal. In **Active Directory**, a **security principal object** has the objectSid **attribute**. In **Active Directory**, the user, computer, and group **object classes** are examples of **security principal object classes** (though not every **group object** is a **security principal object**). In **AD LDS**, any **object** containing the msDS-BindableObject **auxiliary class** is a **security principal**. See also **computer object**, **group object**, and **user object**.

security protocol: A protocol that performs **authentication** and possibly additional security services on a network.

security provider: (1) A Component Object Model (COM) object that provides methods that return custom information about the security of a site.

(2) A pluggable security module that is specified by the protocol layer above the **remote procedure call (RPC)** layer, and will cause the RPC layer 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.

(3) A pluggable security module that is specified by the protocol layer above **remote procedure call (RPC)**, and will cause **RPC** to use this module to secure messages in a communication session with the server. Sometimes referred to as an **authentication** service. For more information, see [C706] and [MS-RPCE].

security support provider (SSP): A dynamic-link library (DLL) that implements the **Security Support Provider Interface (SSPI)** by making one or more security packages available to applications. Each security package provides mappings between an application's **SSPI** function calls and an actual security model's functions. Security packages support **security protocols** such as **Kerberos** authentication and NTLM.

Security Support Provider Interface (SSPI): A Windows-specific API implementation that provides the means for connected applications to call one of several security providers to establish authenticated connections and to exchange data securely over those connections. This is the Windows equivalent of Generic Security Services (GSS)-API, and the two families of APIs are on-the-wire compatible.

security token: (1) An opaque message or data packet produced by a **Generic Security Services (GSS)**-style **authentication** package and carried by the application protocol. The application has no visibility into the contents of the **token**.

(2) A collection of one or more **claims**. Specifically in the case of mobile devices, a **security token** represents a previously authenticated user as defined in the Mobile Device Enrollment Protocol [MS-MDE].

security token service (STS): (1) A web service that issues **claims** (2) and packages them in encrypted security tokens.

(2) A web service that issues **security tokens**. That is, it makes assertions based on evidence that it **trusts**; these assertions are for consumption by whoever trusts it. For more information, see [WSFedPRP] sections 1.4 and 2 and [WSTrust] section 2.4. For [MS-ADFSPP], [MS-ADFSWAP], and [MS-MWBF], **STS** refers to services that support (either directly or via a front end) the protocol defined in each of those specifications.

(3) To communicate trust, a service requires proof, such as a **signature** to prove knowledge of a **security token** or set of **security tokens**. A service itself can generate tokens or it can rely on a separate **STS** to issue a **security token** with its own trust statement. (Note that for some **security token** formats, this can be just a re-issuance or co-**signature**.) This forms the basis of trust brokering.

(4) A special type of server defined in WS-Trust [WSTrust1.3].

segment: (1) A subdivision of content. In version 1.0 Content Information, each segment has a size of 32 megabytes, except the last segment which can be smaller if the content size is not a multiple of the standard segment sizes. In version 2.0 Content Information, segments can vary in size.

(2) A set of stations that see each other's link-layer frames without being changed by any device in the middle, such as a switch.

(3) A unit of content for discovery purposes. A segment is identified on the network by its public identifier, also known as segment ID or HoHoDk. A segment does not belong to any particular content; it can be shared by many content items if all those content items have an identical segment-sized portion at some offset.

send on behalf: A special permission that is granted to a **delegate**. It allows the delegate to send **Message objects** representing the **delegator**.

sendable attendee: An attendee to whom a **meeting request** or **meeting update** will be sent. A sendable attendee can be a **required attendee** or an **optional attendee**, or a **resource**.

sender flag: A collection of property values that indicate that a **Draft Message object** has been marked such that the copy of the **Message object** that is saved in the sender's mailbox after the message is sent will appear flagged to the sender.

sender reminder: A collection of property values that indicate that a **Draft Message object** has been marked such that the copy of the Message object that is saved in the sender's **mailbox** after the message is sent will have an **active reminder**.

Sent Items folder: A **special folder** that is the default location for storing copies of **Message objects** after they are submitted or sent.

sequence: (1) A unique identifier for a delta that includes the user identifier for the **endpoint** (3) that created the delta.

(2) The set of message packets sent over a session that represent a message **sequence**. A message is associated with a **sequence** number that corresponds to its position within the **sequence**. **Sequence** numbers begin with 1 and increment by 1 with each subsequent message.

(3) A one-way, uniquely identifiable batch of messages between an **RMS** and an **RMD**.

sequence number: (1) A numeric value that is used to define the order in which a series of events occurs in an execution sequence or transaction.

(2) The revision number of a **Meeting object**. The sequence number is used to determine the most recent **meeting update** that was sent by the organizer.

(3) An 8-bit identifier that specifies the location order of a **speech frame** within a **voice burst**, and which is used to reorder speech frames upon their receipt. The value of the number starts at 0, increases by one for each **speech frame** within the **voice burst**, and may wrap through reuse of older low values that are no longer in the computing system. Sequence number wrapping occurs when the transmitting client reuses a **sequence number** that was previously used. For example, after using **sequence numbers** 0x00 through 0xFF, the client transmits a **speech frame** that reuses the **sequence number** 0x00. It is the responsibility of the receiver to be aware that the **sequence numbers** of received speech frames may wrap. In order to allow for guaranteed reordering, the receiver must appropriately handle the situation where more than one received **speech frame** uses the same **sequence number**.

(4) In the NTLM protocol, a sequence number can be explicitly provided by the application protocol, or generated by NTLM. If generated by NTLM, the sequence number is the count of each message sent, starting with 0.

(5) A number that uniquely identifies a request and response that is sent on an SMB 2 Protocol **connection**. For a description of how **sequence numbers** are allocated, see [MS-SMB2] sections 3.2.4.1.6 and 3.3.1.1.

Serialization Format: The structure of the serialized message content, which can be either binary or **SOAP**. Binary serialization format is specified in [MS-NRBF]. SOAP serialization format is specified in [MS-NRTP].

server: (1) A computer on which the **remote procedure call (RPC)** server is executing.

(2) A replicating machine that sends replicated files to a partner (client). The term "server" refers to the machine acting in response to requests from partners that want to receive replicated files.

(3) A **DirectPlay** system application that is hosting a **DirectPlay** game session. In the context of **DirectPlay 8**, the term is reserved for **hosts** using **client/server mode**.

(4) For the Peer Content Caching and Retrieval Framework, a server is a server-role peer; that is, a peer that listens for incoming block-range requests from client-role peers and responds to the requests.

(5) Used as a synonym for domain controller. See [MS-DISO].

(6) Refers to the Group Policy server that is involved in a policy application sequence. See [MS-GPOL].

(7) The entity that responds to the HTTP connection. See [MS-TSWP].

(8) A server capable of issuing OMA-DM commands to a client and responding to OMA-DM commands issued by a client. See [MS-MDM]

(9) Used to identify the system that implements WMI services, provides management services, and accepts DCOM ([MS-DCOM]) calls from WMI clients.

(10) A domain controller. Used as a synonym for domain controller. See [MS-ADOD]

(11) An entity that transfers content to a client through streaming. A server might be able to do streaming on behalf of another server; thus, a server can also be a proxy. See [MS-WMLOG]

(12) Used as described in [RFC2616] section 1.3. See [MS-NTHT]

(13) For the purposes of [MS-RDC], the server is the source location.

(14) Any process that accepts commands for execution from a client by using the PowerShell Remoting Protocol.

Server Message Block (SMB): A protocol that is used to request file and print services from server systems over a network. The SMB protocol extends the CIFS protocol with additional security, file, and disk management support. For more information, see [CIFS] and [MS-SMB].

Server object: An object on a server that is used as input or created as output for **remote operations (ROPs)**.

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 **ROP responses**.

server replica: A copy of a user's **mailbox** that exists on a server.

server-side rule: A **rule** for which all actions are executed by a server.

service: (1) A process or agent available on the network, offering resources or services for clients. Examples of services include file servers, web servers, and so on.

(2) A process or agent that is available on the network, offering resources or services for clients. Examples of services include file servers, web servers, and so on.

(3) A program that is managed by the **Service Control Manager (SCM)**. The execution of this program is governed by the rules defined by the **SCM**.

(4) The receiving endpoint of a **web services** request message, and sender of any resulting **web services** response message.

(5) A logical functional unit that represents the smallest units of control and that exposes actions and models the state of a physical device with state variables. For more information, see [UPNPARCH1.1] section 3.

(6) An application that provides management services to clients through the WS-Management Protocol and other web services.

(7) A **SIP method** defined by Session Initiation Protocol Extensions used by the **client** to request a service from the **server**.

service binding information: The **URIs** that are needed to bind to a service.

service connection point: An object that is made available by a directory service and that clients can use to discover **Autodiscover servers**.

Service Control Manager (SCM): An **RPC** server that enables configuration and control of **service** programs.

session: (1) A unidirectional communication channel for a stream of messages that are addressed to one or more destinations. A destination is specified by a resource URL, an identity URL, and a device URL. More than one session can be multiplexed over a single connection.

(2) A representation of application data in system memory. It is used to maintain state for application data that is being manipulated or monitored on a protocol server by a user.

(3) A collection of multimedia senders and receivers and the data streams that flow between them. A multimedia conference is an example of a multimedia session.

(4) In **Kerberos**, an active communication channel established through **Kerberos** that also has an associated cryptographic **key**, message counters, and other state.

(5) In **Server Message Block (SMB)**, a persistent-state association between an **SMB** client and **SMB** server. A **session** is tied to the lifetime of the underlying **NetBIOS** or TCP connection.

(6) In the **Challenge-Handshake Authentication Protocol (CHAP)**, a **session** is a lasting connection between a peer and an authenticator.

(7) In the Workstation service, an authenticated connection between two computers.

(8) An active communication channel established through NTLM, that also has an associated cryptographic **key**, message counters, and other state.

(9) In **OleTx**, a transport-level connection between a **Transaction Manager** and another Distributed Transaction participant over which multiplexed logical connections and messages flow. A **session** remains active so long as there are logical connections using it.

(10) The state maintained by the server when it is **streaming** **content** to a client. If a server-side **playlist** is used, the same **session** is used for all **content** in the **playlist**.

(11) An **authenticated context** that is established between an SMB 2 Protocol client and an SMB 2 Protocol server over an SMB 2 Protocol **connection** for a specific security principal. There could be multiple active **sessions** over a single SMB 2 Protocol **connection**. The SessionId field in the SMB2 packet header distinguishes the various **sessions**.

(12) An authenticated communication channel between the client and server correlating a group of messages into a conversation.

(13) A collection of state information on a directory server. An implementation of the **SOAP session extensions (SSE)** is free to choose the state information to store in a session.

(14) In **LU 6.2**, a **session** is a **connection** between **LUs** that can be used by a succession of **conversations**. A given pair of LU 6.2s may be connected by multiple **sessions**. For a more complete definition, see [LU62Peer].

(15) A context for managing communication over LLTD among **stations**.

(16) The operational environment in which an application and its **commands** execute.

(17) A context for managing communication over **qWave-WD** among devices. This is equivalent to a **TCP** connection.

(18) A multimedia **session** is a set of multimedia senders and receivers and the data streams flowing from senders to receivers. A multimedia conference is an example of a multimedia **session**.

(19) A set of multimedia senders and receivers and the data streams flowing from senders to receivers. A multimedia **conference** is an example of a multimedia session.

Session Context: A server-side partitioning for client isolation. All client actions against a server are scoped to a specific Session Context. All **messaging objects** and data that is opened by a client are isolated to a Session Context.

session context handle: A **remote procedure call (RPC)** context handle that is used by a client when issuing RPCs against a server on EMSMDB interface methods. It represents a handle to a unique session context on the server.

Session Initiation Protocol (SIP): An application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants. **SIP** is defined in [RFC3261].

session key: (1) A symmetric key that is derived from a master key and is used to encrypt or authenticate a specific media stream by using the **Secure Real-Time Transport Protocol (SRTP)** and **Scale Secure Real-Time Transport Protocol (SSRTP)**.

(2) A relatively short-lived **symmetric key** (a cryptographic key negotiated by the client and the server based on a shared secret). A session key's lifespan is bounded by the **session** to which it is associated. A **session key** should be strong enough to withstand cryptanalysis for the lifespan of the **session**.

SHA-1 hash: A hashing algorithm as specified in [FIPS180-2] that was developed by the National Institute of Standards and Technology (NIST) and the National Security Agency (NSA).

share: (1) A resource offered by a Common Internet File System (CIFS) server for access by CIFS clients over the network. A **share** typically represents a directory tree and its included files (referred to commonly as a "disk share" or "file share") or a printer (a "print share"). If the information about the **share** is saved in persistent store (for example, Windows registry) and reloaded when a file server is restarted, then the **share** is referred to as a "sticky share". Some **share** names are reserved for specific functions and are referred to as special **shares**: IPC$, reserved for interprocess communication, ADMIN$, reserved for remote administration, and A$, B$, C$ (and other local disk names followed by a dollar sign), assigned to local disk devices.

(2) To make content on a **host** desktop available to **participants**. **Participants** with a sufficient **control level** may interact remotely with the **host** desktop by sending input commands.

(3) A local resource that is offered by an SMB 2 Protocol server for access by SMB 2 Protocol clients over the network. The SMB 2 Protocol defines three types of **shares**: file (or disk) **shares**, which represent a directory tree and its included files; pipe **shares**, which expose access to named pipes; and print **shares**, which provide access to print resources on the server. A pipe **share** as defined by the SMB 2 Protocol must always have the name "IPC$". A pipe **share** must only allow named pipe operations and DFS referral requests to itself.

shared folder: A folder for which a sharing relationship has been created to share items in the folder between two servers.

shared space: A set of tools that is synchronized between different **endpoints** (3), as described in [MS-GRVDYNM].

sharing invitation: A type of **Sharing Message object** that informs a user that the user was granted access to another user's folder and provides the information necessary to locate that folder.

Sharing Message object: A **Message object** that is used to inform a recipient that they were granted access to another user's folder, request access to a recipient's folder, or respond to a request for access to a folder.

sharing provider: A software agent that is responsible for properly generating and processing a predefined **Sharing Message object** format.

sharing request: A type of **Sharing Message object** that is used to request access to a user's folder.

sharing response: A type of **Sharing Message object** that is used to respond to a sharing request.

Short Message Service (SMS): A communications protocol that is designed for sending text messages between mobile phones.

Side: An area on a **physical medium** that can store data. Although most **physical media** have only a single **side**, some may have two **sides**. For instance, a magneto-optic (MO) disk has two **sides**: an "A" **side** and a "B" **side**. When an MO disk is placed in a **drive** with the "A" **side** up, the "A" **side** is accessible and the "B" **side** is not. To access the "B" **side**, the disk must be inserted with the "B" **side** up. The data stored on different **sides** of the same **physical medium** are independent of one another.

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. can be 10:45 A.M., thus allowing the user 15 minutes to prepare for or travel to the meeting upon receiving the notification.

signature: (1) A synonym for hash.

(2) A value computed with a cryptographic algorithm and bound to data in such a way that intended recipients of the data can use the **signature** to verify that the data has not been altered and/or has originated from the signer of the message, providing message integrity and **authentication**. The **signature** can be computed and verified either with **symmetric key** algorithms, where the same **key** is used for signing and verifying, or with asymmetric **key** algorithms, where different **keys** are used for signing and verifying (a private and public **key** pair are used). For more information, see [WSFedPRP].

(3) The lowest **node ID** in the **graph**.

(4) A structure containing a hash and block **chunk** size. The hash field is 16 bytes, and the **chunk** size field is a 2-byte unsigned integer.

significant change: A change that is made by an organizer to a **Meeting object** and requires a **Meeting Update object** to be sent.

Simple Mail Transfer Protocol (SMTP): A member of the TCP/IP suite of protocols that is used to transport Internet messages, as described in [RFC5321].

Simple Symmetric Transport Protocol (SSTP): A protocol that enables two applications to engage in bi-directional, asynchronous communication. SSTP supports multiple application **endpoints** (5) over a single network connection between client nodes.

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

single-valued claim: See **claim**.

SIP element: An entity that understands the **Session Initiation Protocol (SIP)**.

SIP method: The primary function that an **SIP request** is meant to call on a **server**. This method is carried in the **request** message itself. Example methods are **INVITE** and BYE.

SIP protocol client: A network client that sends **Session Initiation Protocol (SIP)** requests and receives SIP responses. An SIP client does not necessarily interact directly with a human user. **User agent clients (UACs)** and proxies are SIP clients.

SIP request: A **Session Initiation Protocol (SIP)** message that is sent from a **user agent client (UAC)** to a **user agent server (UAS)** to call a specific operation.

site collection: A set of **websites** (1) that are in the same **content database**, have the same owner, and share administration settings. A site collection can be identified by a **GUID** or the **URL** of the **top-level site** for the site collection. Each site collection contains a top-level site, can contain one or more subsites, and can have a shared navigational structure.

site mailbox: A repository comprised of a mailbox and a web-based collaboration environment that is presented to users as a mailbox in an email client. A site mailbox uses team membership to determine which users have access to the repository.

site template: An XML-based definition of site settings, including formatting, lists, views, and elements such as text, graphics, page layout, and styles. Site templates are stored in .stp files in the content database.

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.

Slot: A storage location within a **library**. For example, a tape **library** has one **slot** for each tape that the **library** can hold. A stand-alone **drive library** has no **slots**. Most **libraries** have at least four **slots**. Sometimes **slots** are organized into collections of **slots** called **magazines**. **Magazines** are usually removable.

SMS object: A **Message object** that represents a **Short Message Service (SMS)** message in a message store.

snooze: A process that delays an **overdue reminder** by a specified time interval. At the end of the time interval, the reminder becomes overdue again.

SOAP: A lightweight protocol for exchanging structured information in a decentralized, distributed environment. **SOAP** uses **XML** technologies to define an extensible messaging framework, which provides a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular

programming model and other implementation-specific semantics. **SOAP 1.2** supersedes **SOAP 1.1**. See [SOAP1.2-1/2003].

SOAP 1.1: (1) Version 1.1 of the **SOAP** (Simple Object Access Protocol) standard. For the complete definition of **SOAP 1.1**, see [SOAP1.1].

(2) Simple Object Access Protocol (SOAP) 1.1 [SOAP1.1].

SOAP 1.2: Version 1.2 of the **SOAP** standard. Some examples of changes introduced in **SOAP 1.2** include an updated envelope structure, as well as updates to the structure and semantics for **SOAP faults**. The binding framework was also updated to allow binding to non-HTTP transports. Starting with version 1.2, **SOAP** is no longer an acronym. See also **SOAP**. For the complete specification of **SOAP 1.2**, see [SOAP1.2-1/2007] and [SOAP1.2-2/2007].

SOAP action: The HTTP request header field used to indicate the intent of the **SOAP** request, using a **URI** value. See [SOAP1.1] section 6.1.1 for more information.

SOAP body: A container for the payload data being delivered by a **SOAP message** to its recipient. See [SOAP1.2-1/2007] section 5.3 for more information.

SOAP envelope: A container for **SOAP message** information and the root element of a **SOAP** document. See [SOAP1.2-1/2007] section 5.1 for more information.

SOAP fault: A container for error and status information within a **SOAP message**. See [SOAP1.2-1/2007] section 5.4 for more information.

SOAP header: A mechanism for implementing extensions to a **SOAP message** in a decentralized manner without prior agreement between the communicating parties. See [SOAP1.2-1/2007] section 5.2 for more information.

SOAP message: An **XML** document consisting of a mandatory **SOAP envelope**, an optional **SOAP header**, and a mandatory **SOAP body**. See [SOAP1.2-1/2007] section 5 for more information.

SOAP session extensions (SSE): Extensions to DSML that make it possible to maintain state information across multiple request/response operations.

soft delete: A process that removes an item from the system, but not permanently. If an item is soft deleted, a server retains a back-up copy of the item and a client can access, restore, or permanently delete the item. See also **hard delete**.

sort order: (1) A set of rules in a search query that defines the order of relevant results. Each rule consists of a managed property, such as modified date or size, and a direction for order, such as ascending or descending. Multiple rules are applied sequentially.

(2) A specific arrangement of cells that is based on cell content. The order can be ascending or descending.

(3) The order in which the rows in a **Table object** are requested to appear. This can involve sorting on multiple properties and sorting of **categories** (5).

(4) The set of rules in a search query that define the ordering of rows in the search result. Each rule consists of a property (for example, name or size) and a direction for the ordering (ascending or descending). Multiple rules are applied sequentially.

spam: An unsolicited email message.

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 **Folder objects** that can be used by an implementation to store and retrieve user data objects.

speech frame: Encoded voice streams are broken into pieces; each piece is called a **speech frame**. For the DirectPlay Voice Protocol, the size of each **speech frame** depends on the **codec** selected. This list of **codecs** and their frame sizes is specified in section 1.3.7.

spooler queue: A series of outgoing messages that are ready for delivery to **recipients** (1).

stamp: Information that describes an **originating update** by a **domain controller (DC)**. The **stamp** is not the new data value; the **stamp** is information about the update that created the new data value. A **stamp** is often called metadata, because it is additional information that "talks about" the conventional data values. A **stamp** contains the following pieces of information: the unique identifier of the **DC** that made the **originating update**; a sequence number characterizing the order of this change relative to other changes made at the originating **DC**; a version number identifying the number of times the data value has been modified; and the time when the change occurred.

standard rule: A **rule** that is created, modified, or deleted by using the RopModifyRules remote operation.

station: Any device that implements LLTD.

storage: (1) An element of a compound file that is a unit of containment for one or more storages and streams, analogous to directories in a file system, as described in [MS-CFB].

(2) A set of elements with an associated CLSID used to identify the application or component that created the storage.

(3) A storage object, as defined in [MS-CFB].

Store object: An object that is used to store **mailboxes** and **public folder** content.

stream: (1) 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.

(2) A flow of data from one host to another host, or the data that flows between two hosts.

(3) A sequence of bytes written to a file on the **NTFS** file system. Every file stored on a **volume** that uses the **NTFS** file system contains at least one **stream**, which is normally used to store the primary contents of the file. Additional **streams** within the file may be used to store file **attributes**, application parameters, or other information specific to that file. Every file has a default data **stream**, which is unnamed by default. That data **stream**, and any other data **stream** associated with a file, may optionally be named.

(4) A sequence of bytes that typically encodes application data.

(5) A sequence of **ASF** media objects ([ASF] section 5.2) that can be selected individually. For example, if a movie has an English and a Spanish soundtrack, each may be encoded in the **ASF** file as a separate **stream**. The video data would also be a separate **stream**.

(6) A **sequence** of **messages** whose delivery is guaranteed exactly once and in order.

(7) A set of **tracks** interchangeable at the client when playing **media**.

(8) An individual audio or video data-flow in a **presentation**. The **media data** in an individual **stream** always uses the same **media data format**.

(9) A flow of data from one host to another host. May also be used to reference the flowing data.

(10) A stream object, as defined in [MS-CFB].

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

streaming: (1) The act of transferring **content** from a sender to a receiver.

(2) The act of processing a part of an XML Infoset without requiring that the entire XML Infoset be available.

string named property: A **named property** that has a Unicode string as a name identifier, which is stored in the Name field of a PropertyName structure. A string named property can have any property type; "string" refers only to its name identifier.

structural object class: An **object class** that is not an **88 object class** and can be instantiated to create a new **object**.

structured document: A document that is internally composed of multiple **streams** (1) that specify data for individual pieces of the document, such as style information, images, or embedded objects. The streams allow pieces of the document to be addressed and manipulated individually.

SubAuthority: A variable-length array of unsigned, 32-bit integer values that is part of a **security identifier (SID)**. Each of these values is called a **SubAuthority**. All **SubAuthority** values excluding the last one collectively identify a **domain**. The last value, termed as the **relative identifier (RID)**, identifies a particular **group** or account relative to the **domain**. For more information, see [SIDD].

subobject: For a folder, the messages and subfolders that are contained in that folder. For a message, the **recipients** (2) and attachments to that message. For an attachment, the **Embedded Message object** for that attachment.

SUBSCRIBE: A **Session Initiation Protocol (SIP)** method that is used to request asynchronous notification of an event or a set of events at a later time.

subscriber: (1) A **Session Initiation Protocol (SIP)** client that is making a SUBSCRIBE request.

(2) An application that needs to receive events that are published by another application.

(3) An application that needs to receive historical data published by another application.

subscription: (1) The result of a SUBSCRIBE request from a **Session Initiation Protocol (SIP)** element.

(2) The end result of an act of a **SIP element** sending a **SUBSCRIBE** request.

(3) A registration performed by a **subscriber** to specify a requirement to receive events, future messages, or historical data.

(4) A request for a copy of a publication to be delivered to a subscriber. For more information, see [MSDN-RepPub].

switch: (1) A data link-layer device that propagates frames between segments and allows communication among stations on different segments. Stations that are connected through a switch see only those frames destined for their segments. Compare this term with hub and router.

(2) A logical device type that provides options to run a terminal window or a custom script for a dial-up connection. This device type is not used for dialing a connection.

symmetric key: A **secret key** used with a cryptographic symmetric algorithm. The key needs to be known to all communicating parties. For an introduction to this concept, see [CRYPTO] section 1.5.

synchronization context: See **synchronization download context** or **synchronization upload context**.

synchronization download context: A **Server object** that represents a context for an **ICS** download.

synchronization scope: A set of complex criterion that defines a superset of all the **messaging objects** that are within a specific mailbox and are considered for a single synchronization operation.

Synchronization Source (SSRC): A 32-bit identifier that uniquely identifies a media **stream** (2) in a **Real-Time Transport Protocol (RTP)** session. An SSRC value is part of an **RTP packet** header, as described in [RFC3550].

synchronization upload context: A **Server object** that represents a context for an **ICS** upload.

syntax: See **attribute syntax**.

system volume (SYSVOL): A shared directory that stores the server copy of the **domain's** public files that must be shared for common access and replication throughout a **domain**.

22 T

Table object: An object that is used to view properties for a collection of objects of a specific type, such as a **Message object** or a **Folder object**. 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 that is defined by a 16-bit property ID and a 16-bit property type. The property ID for a tagged property is in the range 0x001 – 0x7FFF. Property IDs in the range 0x8000 – 0x8FFF are reserved for assignment to **named properties**.

target: An **actor** to which a **task** (2) is assigned.

target location: The **target location** is the destination location of a file that has been compressed by **RDC**.

task: (1) An act to be executed by all **query servers**, and any requisite information for those query servers to execute that act correctly.

(2) A component of an **action** (1) that defines the work that **actors** need to do within a workflow system. An action can have zero or more tasks that are each assigned to different **targets**. There is a one-to-one correlation between tasks and targets.

(3) An **object** (1) that represents an assignment to be completed.

(4) An object identifying an administrative action (for example, running a program) to be performed on specified **triggers** and **conditions** (for example, every day at a specific time). Synonym for **job**.

(5) The building block of a package. A task consists of code that executes a function, as specified by the options, settings, and parameters of the task that are specified when the task is called.

task acceptance: A **Message object** that is used to convey acceptance of a task assignment.

task assignee: A user to whom a task has been assigned.

task communication: Collectively, a **task request**, a **task acceptance** or **task rejection**, and a **task update**.

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

task owner: The user who is responsible for updating a task. For unassigned tasks, the local user is the owner. For assigned tasks, the **task assignee** is the owner.

task rejection: A **Message object** that is used to convey the rejection of a task assignment.

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

task update: A **Message object** that is used by a task assignee to send task changes to a task assigner.

Tasks folder: A **Folder object** that contains **Task objects**.

template: A file that contains pre-defined formatting including layout, text and graphics. It serves as the basis for new documents that have a similar look or purpose. See also **form template** (Microsoft InfoPath) and **site template** (SharePoint Products and Technologies).

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

term: A concept or an idea that is stored and can be used as metadata.

term set: A collection of terms that are arranged into and stored as a hierarchy or a flat list.

term store: A database in which managed metadata is stored in the form of **term sets** and **terms**.

ticket: A record generated by the **key distribution center (KDC)** that helps a client authenticate to a service. It contains the client's identity, a unique cryptographic key for use with this ticket (the **session key**), a time stamp, and other information, all sealed using the service's **secret key**. It only serves to authenticate a client when presented along with a valid authenticator.

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 a **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**.

token: (1) A word in an item or a search query that translates into a meaningful word or number in written text. A token is the smallest textual unit that can be matched in a search query. Examples include "cat", "AB14", or "42".

(2) A set of rights and privileges for a given user.

(3) The byte that specifies the start of a record.

(4) A block of data that is issued to a **user** on successful **authentication** by the **authentication server**. Such a **token** is presented to a service to prove one's identity and attributes to a service. The **token** is used in the process of determining the **user's** authorization and access privileges.

tombstone: (1) An individual record of scheduling data that represents a **Meeting object** where an attendee declined a meeting.

(2) An object that has been deleted, but remains in storage until a configured amount of time (the **tombstone lifetime**) has passed, after which the object is permanently removed from storage. By keeping the **tombstone** in existence for the **tombstone lifetime**, the deleted state of the object is able to replicate. **Tombstones** exist only when the **Recycle Bin optional feature** is not enabled.

(3) In Distributed File System Replication (DFS-R), an update pertaining to a file deletion.

(4) A marker that is used to represent an item that has been deleted. A tombstone is used to track deleted items and prevent their reintroduction into the synchronization community.

(5) An inactive DNS node which is not considered to be part of a DNS **zone** but has not yet been deleted from the **zone** database in the **directory server**. **Tombstones** may be permanently deleted from the **zone** once they reach a certain age. **Tombstones** are not used for DNS **zones** that are not stored in the **directory server**. A node is a tombstone if its dnsTombstoned attribute has been set to "TRUE".

tombstone lifetime: The amount of time a deleted directory object remains in storage before it is permanently deleted. To avoid inconsistencies in object deletion, the **tombstone lifetime** is configured to be many times longer than the worst-case replication latency.

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

top-level site: The first site in a site collection. All other sites within a site collection are child sites of the top-level site. The URL of the top-level site is also the URL of the site collection.

topology discovery test: A test that an application or higher-layer protocol can use to facilitate discovering the link-layer topology of a single link in a network. That is, to facilitate discovering the set of **segments** and **switches**, and determining which **responders** are on which segments. Compare this term with **quick discovery**.

track: (1) Any of the concentric circles on a disk platter over which a magnetic head (used for reading and writing data on the disk) passes while the head is stationary but the disk is spinning. A track is subdivided into sectors, upon which data is read and written.

(2) A time-ordered collection of samples of a particular type (such as audio or video).

transaction: (1) An object that stores the state and metadata for an item during a crawl.

(2) A single unit of work. If a transaction is successful, all data modifications that were made during the transaction are committed and become a permanent part of the database. If a transaction encounters an error and is canceled or rolled back, all data modifications are erased.

(3) The process of opening or creating an object on a server, and the subsequent committing of changes to the object by calling the required save function, at which time all changes to that instance of the object are either saved to the server, or discarded if a failure occurs before saving is finished successfully. Until successfully saved, changes are invisible to any other instances of the object.

(4) In OleTx, an **atomic transaction**.

transaction manager: The party that is responsible for managing and distributing the outcome of **atomic transactions**. A transaction manager is either a root transaction manager or a subordinate transaction manager for a specified transaction.

transform: (1) An operation that is performed on data to change it from one form to another. Two examples of transforms are compression and encryption.

(2) An algorithm that transforms the size, orientation, and shape of objects that are copied from one **coordinate space** into another. Although a transform affects an object as a whole, it is applied to each point, or to each line, in the object.

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 address: (1) A 3-tuple that consists of a port, an IPv4 or IPV6 address, and a transport protocol of User Datagram Protocol (UDP) or Transmission Control Protocol (TCP).

(2) The combination of a network address and **port** that identifies a transport-level endpoint, for example an IP address and a **UDP port**. Packets are transmitted from a source transport address to a destination transport address. See [RFC3550] section 3.

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 (as specified in [X509]). **TLS** is standardized in the IETF TLS working group. See [RFC4346].

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

Transport Neutral Encapsulation Format (TNEF) message: A **MIME** representation of an email message in which attachments and some message properties are carried in a **Transport Neutral Encapsulation Format (TNEF)** body part.

trigger: A change of state (for example, reaching a specific time of day) that signals when a **task** is to run. A **task** runs when any of its **triggers** and all of its **conditions** are satisfied.

trust: (1) The state of accepting another authority's statements for the purposes of authentication and authorization. If domain A trusts domain B, domain A will accept domain B's authentication and authorization statements for principals represented by security principal objects in domain B; for example, the list of groups to which a particular user belongs. As a noun, a trust is the relationship between two domains described in the previous sentence.

(2) To accept another authority's statements for the purposes of **authentication** and **authorization**, especially in the case of a relationship between two domains. If **domain** A trusts **domain** B, **domain** A accepts **domain** B's **authentication** and **authorization** statements for **principals** represented by **security principal objects** in **domain** B; for example, the list of groups to which a particular user belongs. As a noun, a **trust** is the relationship between two **domains** described in the previous sentence.

(3) The characteristic that one entity is willing to rely on a second entity to execute a set of actions and/or to make a set of assertions about a set of subjects and/or scopes. For more information, see [WSFedPRP] sections 1.4 and 2.

Two-Way Method: A **Remote Method** that has a response sent from the implementation of the **Remote Method** back to the caller.

23 U

uncustomized: A condition of a document whose content is stored in a location other than the content database. If a document is uncustomized, the front-end web server determines the location of the content by using the SetupPath value for the document. Also referred to as ghosted.

Unicode: (1) A character encoding standard developed by the Unicode Consortium that represents almost all of the written languages of the world. The Unicode standard provides three forms (UTF-8, UTF-16, and UTF-32) and seven schemes (**UTF-8**, **UTF-16**, UTF-16 BE, UTF-16 LE, UTF-32, UTF-32 LE, and UTF-32 BE).

(2) A character encoding standard developed by the Unicode Consortium that represents almost all of the written languages of the world. The **Unicode** standard [UNICODE5.0.0/2007] provides three forms (UTF-8, UTF-16, and UTF-32) and seven schemes (UTF-8, UTF-16, UTF-16 BE, UTF-16 LE, UTF-32, UTF-32 LE, and UTF-32 BE).

(3) The set of characters as defined by [UNICODE5.0.0/2007] that is encoded in UCS-2.

Unicode character: Unless otherwise specified, a 16-bit UTF-16 code unit.

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

Uniform Resource Identifier (URI): A string that identifies a resource. The URI is the Web service addressing mechanism defined in Internet Engineering Task Force (IETF) Uniform Resource Identifier (URI): Generic Syntax [RFC3986].

Uniform Resource Locator (URL): A string of characters in a standardized format that identifies a document or resource on the World Wide Web. The format is as specified in [RFC1738].

unique identifier (UID): A pair consisting of a **GUID** and a version sequence number to identify each resource uniquely. The UID is used to track the object for its entire lifetime through any number of times that the object is modified or renamed.

Universal Naming Convention (UNC): A string format that specifies the location of a resource. For more information, see [MS-DTYP] section 2.2.57.

universal serial bus (USB): An external bus that supports Plug and Play installation. It allows devices to be connected and disconnected without shutting down or restarting the computer.

universally unique identifier (UUID): A 128-bit value. UUIDs can be used for multiple purposes, from tagging objects with an extremely short lifetime, to reliably identifying very persistent objects in cross-process communication such as client and server interfaces, manager entry-point vectors, and **RPC** objects. UUIDs are highly likely to be unique. UUIDs are also known as a **globally unique identifiers (GUIDs)** and these terms are used interchangeably in the Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the UUID. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the UUID.

unsendable attendee: An attendee to whom a **meeting request** or **meeting update** is not sent.

update: (1) An add, modify, or delete of one or more objects or attribute values. See **originating update**, **replicated update**.

(2) The combination of **metadata** and associated **content** for a software update. An **update** is identified by a **GUID**.

(3) The set of metadata pertaining to a file or file deletion. The main fields in an update consist of the **unique identifier (UID)**, **global version sequence number (GVSN)**, file name, file attributes, and flags indicating whether the update is for an existing file, or for a file deletion.

update server: A computer that implements the Windows Update Services: Server-Server Protocol or the Windows Server Update Services: Client-Server Protocol to provide **updates** to **client computers** and other **update servers**.

Use License: An XrML 1.2 license that authorizes a user to gain access to a protected content file and describes the applicable usage policies. Also referred to as "End-User License (EUL)."

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: (1) A person who employs a **web browser requestor** to access a **WS resource**.

(2) The real person who has a member account. The user is **authenticated** by being asked to prove knowledge of the secret password associated with the user name.

user agent: An HTTP user agent, as specified in [RFC2616].

user agent client (UAC): A logical entity that creates a new **request**, and then uses the **client** transaction state machinery to send it. The role of **UAC** lasts only for the duration of that transaction. In other words, if a piece of software initiates a **request**, it acts as a **UAC** for the duration of that transaction. If it receives a **request** later, it assumes the role of a **user agent server (UAS)** for the processing of that transaction.

user agent server (UAS): A logical entity that generates a response to a **Session Initiation Protocol (SIP)** request. The response either accepts, rejects, or redirects the request. The role of the UAS lasts only for the duration of that transaction. If a process responds to a request, it acts as a UAS for that transaction. If it initiates a request later, it assumes the role of a **user agent client (UAC)** for that transaction.

User Datagram Protocol (UDP): The connectionless protocol within TCP/IP that corresponds to the transport layer in the ISO/OSI reference model.

user object: An object of class user. A user object is a security principal object; the principal is a person or service entity running on the computer. The shared secret allows the person or service entity to authenticate itself, as described in ([MS-AUTHSOD] section 1.1.1.1).

user principal name (UPN): A user account name (sometimes referred to as the user logon name) and a domain name that identifies the domain in which the user account is located. This is the standard usage for logging on to a Windows domain. The format is:someone@example.com (in the form of an email address). In **Active Directory**, the userPrincipalName **attribute** (2) of the account object, as described in [MS-ADTS].

UTF-16: A standard for encoding **Unicode characters**, defined in the Unicode standard, in which the most commonly used characters are defined as double-byte characters. Unless

specified otherwise, this term refers to the UTF-16 encoding form specified in [UNICODE5.0.0/2007] section 3.9.

UTF-16LE: The Unicode Transformation Format - 16-bit, Little Endian encoding scheme. It is used to encode **Unicode** characters as a sequence of 16-bit codes, each encoded as two 8-bit bytes with the least-significant byte first.

UTF-8: A byte-oriented standard for encoding **Unicode characters**, defined in the Unicode standard. Unless specified otherwise, this term refers to the UTF-8 encoding form specified in [UNICODE5.0.0/2007] section 3.9.

24 V

vCard: A format for storing and exchanging electronic business cards, as described in [RFC2426].

version sequence number (VSN): A 64-bit unsigned number. Version sequence numbers are assigned to global version sequence numbers as part of file metadata in monotonic increasing order.

virtual private networking (VPN): A private data network that makes use of the public telecommunication infrastructure.

voice burst: Individual speech frames are grouped together into **voice bursts**. A **voice burst** contains a set of speech frames during which the timing is preserved by receiving clients. Gaps in the **voice burst** will be filled with silence to preserve the timing of the received packets. Timing is not guaranteed to be preserved between **voice bursts**. For more information about **voice bursts**, see section 1.3.1.

voice message: A **Message object** that contains audio content recorded by a calling party.

Voice over IP (VoIP): The use of the Internet Protocol (IP) for transmitting voice communications. VoIP delivers digitized audio in packet form and can be used to transmit over intranets, extranets, and the Internet.

volume: A group of one or more partitions that forms a logical region of storage and the basis for a file system. A **volume** is an area on a storage device that is managed by the file system as a discrete logical storage unit. A partition contains at least one **volume**, and a volume can exist on one or more partitions.

volume sequence number (VSN) (for file replication service): A unique sequence number assigned to a change order to order the event sequence in a replica. It is a monotonically increasing sequence number assigned to each change that originates on a given replica member. If one change order has a smaller **volume sequence number (VSN)** than another change order, the change that the first change order represents occurs before the change that the second change order represents.

25 W

web analyzer: An entity that is part of a search service application and is used to assess the relevancy of **anchor text** in an item.

web analyzer view: A set of crawl collections that is subject to link analysis and **anchor text** aggregation. A crawl collection can be a member of more than one web analyzer view.

web application: (1) A container in a configuration database that stores administrative settings and entry-point **URLs** for **site collections**.

(2) A software application that uses **HTTP** as its core communication protocol and delivers information to the user by using web-based languages such as **HTML** and **XML**.

(3) A collection of URLs that share a server execution environment. This collection is defined relative to a root URL. A **web application** runs in response to HTTP requests for the URLs in the collection. The process or processes that run in response to such an HTTP request are termed the application host.

web application identifier: (1) A GUID that identifies a web application.

(2) Each **ASP.NET** application running on a web server is uniquely identified with a **web application identifier**. The **web application identifier** is the virtual path of the web application on the web server. A **web application identifier** is used as part of the identifying key on a state server when storing and retrieving session data for a specific browser session.

web bot: See **bot**.

web browser requestor: An HTTP 1.1 web browser client that transmits protocol messages between an **IP/STS** and a **relying party**.

web component: Any component, such as a bitmap, image, Java applet, or ActiveX control, that can be inserted into a webpage.

web crawler: A search component that traverses websites, downloads content from those sites, and submits that content for indexing.

web discussion: A component and **add-in** that enables users to enter comments about documents and pages without modifying the actual content of those documents or pages.

web discussion comment: An individual comment that is added to a web discussion.

Web Distributed Authoring and Versioning Protocol (WebDAV): The Web Distributed Authoring and Versioning Protocol, as described in [RFC2518] or [RFC4918].

Web Distribution Point (WDP): A location on a server where **offline address book (OAB)** files are published for web distribution. A client can discover the URI of a WDP by using the Autodiscover Publishing and Lookup Protocol, as described in [MS-OXDSCLI].

web log: See **blog** (1).

web log posting: A message that is submitted to a **blog** (1).

web query: An external data connection that retrieves a table from a website and inserts table data into a workbook.

web server: A server computer that hosts websites and responds to requests from applications.

web service: (1) A unit of application logic that provides data and services to other applications and can be called by using standard Internet transport protocols such as **HTTP**, **Simple Mail Transfer Protocol (SMTP)**, or **File Transfer Protocol (FTP)**. Web services can perform functions that range from simple requests to complicated business processes.

(2) A software entity that responds to SOAP messages ([SOAP1.1],.[WSDL]).

web service (WS) resource: A destination HTTP 1.1 web application or an HTTP 1.1 resource serviced by the application. In the context of this protocol, it refers to the application or manager of the resource that receives identity information and assertions issued by an **IP/STS** using this protocol. The **WS resource** is a **relying party** in the context of this protocol. For more information, see [WSFedPRP] sections 1.4 and 2.

web service method: A procedure that is exposed to web service clients as an operation that can be called on the web service. Also referred to as web method.

Web Services Description Language (WSDL): An XML format for describing network services as a set of endpoints that operate on messages that contain either document-oriented or procedure-oriented information. The operations and messages are described abstractly and are bound to a concrete network protocol and message format in order to define an endpoint. Related concrete endpoints are combined into abstract endpoints, which describe a network service. WSDL is extensible, which allows the description of endpoints and their messages regardless of the message formats or network protocols that are used.

web ticket: A security token that is sent by a protocol client to a web application during **authentication** (2). The security token can be included in either the body or the header of an HTTP message.

WebDAV client: A computer that uses **WebDAV**, as described in [RFC2518] or [RFC4918], to retrieve data from a **WebDAV server**.

WebDAV server: A computer that supports **WebDAV**, as described in [RFC2518] or [RFC4918], and responds to requests from **WebDAV clients**.

web-only view: A view of a workbook from within a web browser.

website: (1) A group of related webpages that is hosted by a server on the World Wide Web or an intranet. Each website has its own entry points, metadata, administration settings, and workflows. Also referred to as site.

(2) A group of related pages and data within a SharePoint site collection. The structure and content of a site is based on a site definition. Also referred to as SharePoint site and site.

week independent: A BYDAY **recurrence part** that does not specify any week numbers.

well-known endpoint: A preassigned, network-specific, stable address for a particular client/server instance. For more information, see [C706].

Windows Metafile Format (WMF): A vector graphics format for Windows-compatible computers. Windows Metafile Format is used primarily as a clip-art format in word-processing documents.

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

work: The set of state changes that are applied to **resources** inside an **atomic transaction**.

working hours: Times of the day that are valid for meetings to be considered for an attendee.

workspace: A set of remote resources, such as **remote applications** and desktops, which are published to end users.

WSDL message: An abstract, typed definition of the data that is communicated during a **WSDL operation**, as described in [WSDL].

WSDL operation: A single action or function of a web service. The execution of a WSDL operation typically requires the exchange of messages between the service requestor and the service provider.

WSDL port type: A named set of logically-related, abstract **Web Services Description Language (WSDL)** operations and messages.

WS-Management: A public standard **SOAP**-based protocol for sharing management data among all operating systems, computers, and devices.

26 X

X.509: An ITU-T standard for public key infrastructure subsequently adapted by the IETF, as specified in [RFC3280].

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

XML: The Extensible Markup Language, as described in [XML1.0].

XML document: A document object that is well formed, as described in [XML], and might be valid. An XML document has a logical structure that is composed of declarations, elements, comments, character references, and processing instructions. It also has a physical structure that is composed of entities, starting with the root, or document, entity.

XML element: An **XML** structure that typically consists of a start tag, an end tag, and the information between those tags. Elements can have **attributes** (1) and can contain other elements.

XML namespace: A collection of names that is used to identify elements, types, and attributes in XML documents identified in a URI reference [RFC3986]. A combination of XML namespace and local name allows XML documents to use elements, types, and attributes that have the same names but come from different sources. For more information, see [XMLNS-2ED].

XML namespace prefix: An abbreviated form of an **XML namespace**, as described in [XML].

XML schema: A description of a type of **XML document** that is typically expressed in terms of constraints on the structure and content of documents of that type, in addition to the basic syntax constraints that are imposed by **XML** itself. An XML schema provides a view of a document type at a relatively high level of abstraction.

XML schema definition (XSD): The World Wide Web Consortium (W3C) standard language that is used in defining XML schemas. Schemas are useful for enforcing structure and 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**.

XMLHTTPRequest (XHR): A software component that is used by browser-based scripts to transfer data between a web browser and a web server.

27 Y

28 Z

zone: A domain namespace is divided up into several sections called zones [RFC1034] and [RFC2181]. A **zone** represents authority over a portion of the DNS namespace, excluding any subzones that are below delegations.

29 Change Tracking

This section identifies changes that were made to the [MS-OXGLOS] protocol document between the July 2014 and October 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.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
	Multiple new terms were added to this document.	Y	New content added.
	Multiple terms have additional definitions added.	Y	New content added.