[MS-GLOS]:
Windows Protocols Master Glossary

Revision Summary

Date	Revision History	Revision Class	Comments
03/14/2007	1.0	Major	Updated and revised the technical content.
04/10/2007	1.1	Minor	Updated the technical content.
05/18/2007	1.2	Minor	Added additional terms
06/08/2007	1.2.1	Editorial	Revised and edited the technical content.
07/10/2007	1.2.2	Editorial	Revised and edited the technical content.
08/17/2007	1.2.3	Editorial	Revised and edited the technical content.
09/21/2007	1.2.4	Editorial	Monthly Release
10/26/2007	1.2.5	Editorial	Revised and edited the technical content.
01/25/2008	1.3	Minor	Updated the technical content.
03/14/2008	1.3.1	Editorial	Revised and edited the technical content.
06/20/2008	2.0	Major	Updated and revised the technical content.
07/25/2008	3.0	Major	Updated and revised the technical content.
08/29/2008	4.0	Major	Updated and revised the technical content.
10/24/2008	5.0	Major	Updated and revised the technical content.
12/05/2008	6.0	Major	Updated and revised the technical content.
01/16/2009	6.0.1	Editorial	Revised and edited the technical content.
02/27/2009	7.0	Major	Updated and revised the technical content.
04/10/2009	8.0	Major	Updated and revised the technical content.
05/22/2009	9.0	Major	Updated and revised the technical content.
07/02/2009	10.0	Major	Updated and revised the technical content.
08/14/2009	11.0	Major	Updated and revised the technical content.
09/25/2009	12.0	Major	Updated and revised the technical content.
11/06/2009	13.0	Major	Updated and revised the technical content.
12/18/2009	13.1	Minor	Updated the technical content.
01/29/2010	13.1.1	Editorial	Revised and edited the technical content.
03/12/2010	13.2	Minor	Updated the technical content.

Date	Revision History	Revision Class	Comments
04/23/2010	14.0	Major	Updated and revised the technical content.
06/04/2010	14.0.1	Editorial	Revised and edited the technical content.
07/16/2010	15.0	Major	Significantly changed the technical content.
08/27/2010	16.0	Major	Significantly changed the technical content.
10/08/2010	16.1	Minor	Clarified the meaning of the technical content.
11/19/2010	16.2	Minor	Clarified the meaning of the technical content.
01/07/2011	16.3	Minor	Clarified the meaning of the technical content.
02/11/2011	16.4	Minor	Clarified the meaning of the technical content.
03/25/2011	17.0	Major	Significantly changed the technical content.
05/06/2011	17.0	No change	No changes to the meaning, language, or formatting of the technical content.
06/17/2011	17.0	No change	No changes to the meaning, language, or formatting of the technical content.

Contents

1 Non-Alphanumeric

@GMT token: A special token that can be present as part of a **file** path to indicate a request to see a previous version of the **file** or directory. The format is "@GMT-YYYY.MM.DD-HH.MM.SS". This 16-bit **Unicode string** represents a time and date in **Coordinated Universal Time (UTC)**, with YYYY representing the year, MM the month, DD the day, HH the hour, MM the minute, and SS the seconds.

.nsc file: A Windows Media Station **file** that serves as an announcement of a Media Stream Broadcast Distribution (MSBD) Protocol.

2 0-9

64-bit Network Data Representation (NDR64): A specific instance of a **remote procedure call (RPC) transfer syntax**. For more information about **RPC transfer syntax**, see [C706-Ch14TransSyntaxNDR].

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

ABNF: See **Augmented Backus-Naur Form (ABNF)**.

abort request: An action that a participant performs to force a transaction to reach an abort outcome.

abstract class: See **abstract object class**.

abstract object class: An **object class** whose only function is to be the basis of inheritance by other **object classes**, thereby simplifying their definition.

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.

acceptor: A participant that receives a session or connection request. This role is also known as the "subordinate".

access check: A verification to determine whether a specific **access type** is allowed by checking a **security context** against a **security descriptor**.

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

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

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

access point: A **network access server (NAS)** that is implementing 802.11.

access profile: A set of configuration data for a **network access server (NAS)** to determine the level of service to provide to an **endpoint**. This configuration data is sent from the **RADIUS server** to the **NAS** as a set of **RADIUS attributes**.

access type: An action defined for access such as "read", "write", "full control", control access right "x", and so on. Used in **security descriptors**.

account: A **user**, **group**, or **alias** object.

account domain: A **domain**, identified by a **security identifier (SID)**, that is the **SID** namespace for which a given machine is authoritative. The **account domain** is the same as the **primary domain** for a **domain controller (DC)** and is its default **domain**. For a Windows machine that is joined to a **domain**, the **account domain** is the **SID** namespace defined by the local Security Accounts Manager [MS-SAMR].

account domain object (account domain): A **domain object** that represents an issuing authority in which **user objects** can be created. For more information about the concept of an issuing authority, see [MS-WSO] section 3.1.2.1.5.

account domain security identifier: The **security identifier (SID)** of the **account domain object**.

account group: A **group object** whose members always include the **security identifier (SID)** of the group in the **authorization context**.

account object: An element of a **Local Security Authority (LSA)** policy database that describes the rights and privileges granted by the **server** to a **security principal**. The **security identifier (SID)** of the **security principal** matches that of the **account object**.

ACID: A term that refers to the four properties that any database system must achieve in order to be considered transactional: Atomicity, Consistency, Isolation, and Durability [GRAY].

ACE: See **access control entry (ACE)**.

acknowledgment (ACK): A signal passed between communicating processes or computers to signify successful receipt of a transmission as part of a communications protocol.

ACL: See **access control list (ACL)**.

activation: (1) 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**.

(2) 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].

Active Directory: A general-purpose network **directory service**.

Active Directory also refers to the Windows implementation of a **directory service**. **Active Directory** stores information about a variety of **objects** in the network. Importantly, user accounts, computer accounts, groups, and all related credential information used by the Windows implementation of **Kerberos** are stored in **Active Directory**. **Active Directory** first became available as part of Windows 2000 and is available as part of Windows 2000 Server products, Windows Server 2003 products, Windows Server 2008 products, and Windows Server 2008 R2 products. **Active Directory** is not present in Windows NT 4.0 or in Windows XP. For more information, see [MS-WSO] section 3.1.2.1.5.2 and [MS-ADTS].

Active Directory domain: A **domain** hosted on **Active Directory**. For more information, see [MS-ADTS].

Active Directory Domain Services (AD DS): See **Active Directory**. **Active Directory Domain Services (AD DS)** is a new term that is replacing the phrase "Active Directory" in the Windows Server 2008 project.

Active Directory Lightweight Directory Services (AD LDS): A **directory service** that is implemented by a **domain controller (DC)**. For more information on **AD LDS**, 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 Directory replication: The process by which the changes that are made to **Active Directory objects** on one **domain controller (DC)** are automatically synchronized with other **DCs**.

Active Directory schema: The Microsoft **Active Directory schema** contains formal definitions of every **object class** that can be created in an **Active Directory forest**. The schema also contains formal definitions of every **attribute** that can exist in an **Active Directory object**.

Active Directory table (ADT): A database of **domain** information, as specified in [MS-ADTS].

active node: A **node** that is currently successfully executing the implementation-specific server-to-server protocols that constitute participation in a **cluster**.

active partition: A **partition** on a **master boot record (MBR)** disk that becomes the **system partition** at system startup if the BIOS is configured to select that disk for boot. An **MBR** disk can have exactly one **active partition**. This attribute is stored within the **partition table** on the disk.

active volume: See **active partition**.

AD: See **Active Directory**.

AD DS: See **Active Directory Domain Services (AD DS)**.

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

AddRef: The process of calling the second IUnknown method (IUnknown::ADDref()) on an **object**. For more information, see [MS-DCOM].

administrative plug-in GUID: See **tool extension GUID**.

administrative template: A file associated with a **Group Policy object (GPO)** that combines information on the syntax of registry-based policy settings with human-readable descriptions of the settings, as well as other information.

administrative tool: A tool that allows administrators to read and write **policy settings** to and from a **Group Policy object (GPO)**.

administrator: A user who has complete and unrestricted access to the computer or **domain**.

administrator in Admin Approval Mode or Consent Admin: A user mode in which **administrators** are prompted for permission before allowing an administrative task to be performed. Also referred to as a "Consent Admin".

administrators: An **alias object** with the **security identifier (SID)** S-1-5-32-544.

Advanced Encryption Standard (AES): A block **cipher** that supersedes the **Data Encryption Standard (DES)**. **AES** is used in symmetric-key cryptography and is also known as the Rijndael **symmetric encryption** algorithm.

Advanced Systems Format (ASF): The file format used by Windows Media.

advertise: To publish descriptive identifying information in a name service.

advertised: An installation state of an application on a **client computer**. An **advertised** application is one that does not have all of the binaries and files necessary for executing the

application present on the computer, but does have metadata on the **client** that allows it to present the application to the user as if all the files were present and also allows the **client** to install all of the missing files at a later time.

alias object: See **resource group**.

allocation unit size: The size (expressed in bytes) of the units used by the **file system** to allocate space on a disk for the **file system** used by the **volume**. The size, in bytes, must be a power of two and must be a multiple of the size of the sectors on the disk. Typical **allocation unit sizes** of most **file systems** range from 512 bytes to 64 KB.

alternate stream: See **named stream**.

ambiguous name resolution (ANR): 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 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.

ancestor object: An **object A** is an ancestor of **object O** if there is a directed path from **A** to **O** (in other words, **A** is on the path from **O** to the root of the tree containing **O**).

anonymous authentication: An **authentication mode** in which neither party verifies the identity of the other party.

anonymous session: A session created for an **anonymous user**.

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.

anywhere access gateway: A **network access server (NAS)** that provides remote connectivity to a network.

AP exchange: See **Authentication Protocol (AP) exchange**.

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 advertise script: A file that contains a sequence of installation operations and configuration data for installing an application on a **client** machine. The installer follows the installation operations in the file and configures the metadata of the application to match the state information specified in the script.

application configuration file (ACF): A supplemental file that accompanies an **Interface Definition Language (IDL)** specification and is used to specify stub processing rules. For more information, see "The Attribute Configuration Source" in Part 2 of [C706] and [MS-RPCE].

Application Desktop Toolbar: A window (anchored to an edge of the screen) that is similar to the taskbar and that typically contains buttons that give the user quick access to other applications and windows.

application directory partition: An **application NC**.

application domain: A virtual process space within which managed code applications are hosted and executed. It is possible to have multiple managed code applications running inside a single process. Each managed code application runs within its own **application domain** and is isolated from other applications that are running in separate **application domains**.

application domain identifier (ID): A number used to uniquely identify an application domain.

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 protocol: A network protocol that visibly accomplishes the task that the user or other agent wants to perform. This is distinguished from all manner of support protocols: from Ethernet or IP at the bottom to security and routing protocols. While necessary, these are not always visible to the user. Application protocols include, for instance, HTTP and Server Message Block (SMB).

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.

AS exchange: See **Authentication Service (AS) exchange**.

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.

Note There is a charge to download these documents.

assigned application: An application that is to be installed at computer startup or user logon.

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**.

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 (e-mail 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).

AttributeId: An OID-valued **attribute** of each attributeSchema **object** in the **schema naming context (schema NC)**. In many **Lightweight Directory Access Protocol (LDAP)** directory implementations, this OID value (although not necessarily referred to as the attributeId) is the standard internal representation of an **attribute**. In the directory model used in [MS-ADTS], however, an attribute is represented by the more familiar **LDAP** display name (stored as the ldapDisplayName attribute on the corresponding attributeSchema object).

AttributeStamp: The type of a **stamp** attached to an **attribute**.

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 [RFC4234].

Authenticated IP (AuthIP): An **Internet Key Exchange (IKE)** protocol extension, as specified in [MS-AIPS].

authenticated users: A built-in security group specified in [MS-WSO] whose members include all users that can be authenticated by a computer.

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.

authentication header (AH): An **Internet Protocol Security (IPsec)** encapsulation mode that provides **authentication** and message integrity. For more information, see [RFC4302] section 1.

authentication level: A numeric value indicating the level of **authentication** or message protection that **remote procedure call (RPC)** will apply to a specific message exchange. For more information, see [C706] section 13.1.2.1 and [MS-RPCE].

authentication mode: One of several modes in which an **authentication** exchange may be performed.

Authentication Protocol (AP) exchange: The **Kerberos** subprotocol called the "authentication protocol", sometimes referred to as the "Client/Server Authentication Exchange", in which the **client** presents a **service ticket** and an **authenticator** to a service to establish an

authenticated communication session with the service. The protocol is specified in [RFC4120] section 3.2.

authentication server: An entity that provides **authentication services** to **authenticators** so that these services do not have to be implemented by the **authenticators**.

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**.

Authentication Service (AS) exchange: The **Kerberos** subprotocol in which the **Authentication Service** component of the **key distribution center (KDC)** accepts an initial logon or **authentication** request from a **client** and provides the **client** with a **ticket granting ticket (TGT)** and necessary cryptographic keys to make use of the **ticket**. This is specified in [RFC4120] section 3.1. The **AS exchange** is always initiated by the **client**, usually in response to the initial logon of a **principal** such as a user.

authentication type: A numeric identifier that uniquely identifies a **security provider**.

authenticator: (1) The entity requesting the **authentication** of a peer.

(2) A protocol message or data structure within a message that carries **authentication** information.

(3) When used in reference to the Netlogon Protocol, the data stored in the NETLOGON_AUTHENTICATOR structure.

(4) When used in reference to **Kerberos**, see **Kerberos authenticator**.

AuthIP: See **Authenticated IP (AuthIP)**.

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

authorization context: The set of identities for groups and the identity of the user made available to a **server** for the purpose of determining **authorization** to a **resource**.

authorization data: An extensible field within a **Kerberos ticket**, used to pass authorization data about the **principal** on whose behalf the **ticket** was issued to the application service.

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**.

AV pair: An attribute/value pair. The name of some **attribute**, along with its value. **AV pairs** in **NTLM** have a structure specifying the encoding of the information stored in them.

4 B

back link: An **attribute** whose value refers to a directory object, and whose Attribute-Schema **object** has an odd value for **attribute** **linkId**. A **back link** exists only in response to the existence of a forward link. Forward links can exist with no **back links**.

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**.

backup browser server: A **browser server** that was selected by the **local master browser server** on that subnet to be available to share the processing load that is required to serve **browser clients**. **Backup browser servers** keep copies of the information that is maintained by the **local master browser server** by periodically querying that server.

backup domain controller (BDC): A **domain controller (DC)** that receives a copy of the **domain** directory database from the **primary domain controller (PDC)**. There is only one **PDC** or **PDC** emulator in a **domain**, and the rest are **backup domain controllers**.

backup stream: The components of a Windows NT backup file. It is important not to confuse a **backup stream** with a **named stream**. **Backup streams** are bytes within the main stream of a Windows NT backup file, while a **named stream** is part of a file that is not a Windows NT backup file that requires a separate open call to access.

Backus-Naur Form (BNF): A syntax used to describe context-free grammars, which is a prescribed way to describe languages.

balloon tooltip: A tooltip displayed inside a balloon-shaped window. It usually has an icon, a title, and the tooltip text.

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

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.

basic provider: A **virtual disk service (VDS)** provider that manages basic disks.

basic volume: A **partition** on a **basic disk**.

BDC: See **backup domain controller (BDC)**.

Bezier curve: A type of curve, defined by a mathematical formula and a number of points greater than or equal to two, which is used in computer graphics and in the mathematical field of numeric analysis. A cubic **Bezier curve** is defined by four points: two endpoints and two control points. The curve does not pass through the control points, but the control points act

like magnets, pulling the curve in certain directions and influencing the way the curve bends. With multiple **Bezier curves**, the endpoint of one is the starting point of the next.

bidirectional domain trust relationship: **Domains** X and Y are said to have a **bidirectional trust relationship** if there is a **domain trust relationship** from X to Y and also one from Y to X.

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

binary large object (BLOB): A collection of binary data stored as a single entity in a database.

binding: The string representation of the protocol sequence, NetworkAddress, and optionally the **endpoint**. Also referred to as "string binding". For more information, see [C706] section "String Bindings".

BitLocker: BitLocker Drive Encryption. A Microsoft-developed feature appearing in Windows Vista that provides **encryption** for an entire **volume**.

BLOB: See **binary large object (BLOB)**.

blocking mode: Determines if input/output (I/O) operations will wait for their entire data to be transferred before returning to the caller. For a write operation, if blocking is enabled, the write request will not complete until the **named pipe** reader has consumed all of the data inserted into the **named pipe** as part of a write request. If blocking is not enabled, the write will complete as soon as the data has been inserted into the **named pipe**, regardless of when the data in the **named pipe** is consumed. For a read operation, if blocking is enabled, the read request will be suspended until the data is available to be read. If blocking is not enabled, the read will complete immediately, even if there is no data available to be read.

BNF: See **Backus-Naur Form (BNF)**.

boot configuration file: A file that contains a list of paths to boot partitions. On architectures featuring the **Extensible Firmware Interface (EFI)**, the **boot configuration file** may be stored on other non-volatile media, such as NVRAM. On all other architectures, it resides in the system partition.

boot file: A file that contains a list of paths to **boot partitions**. On some systems, the **boot file** may be stored on other non-volatile media, such as NVRAM.

boot loader: An architecture-specific file that loads the operating system on the **boot partition** as specified by the **boot configuration file**.

boot loader file: See **boot loader**.

boot partition: A **partition** containing the operating system.

boot volume: See **boot partition**.

boot.ini: The name of the **boot loader file** on Windows–based computers.

boxcar: A set of messages transmitted together by way of an underlying MSDTC Connection Manager: OleTx Transports Protocol session.

bridgehead domain controller (bridgehead DC): A **domain controller (DC)** that may replicate updates to or from **DCs** in sites other than its own.

broadcast: A style of resource location in which a **client** makes a request to all parties on the network simultaneously (a one-to-many communication). Also, a mode of resource location that does not use a name service.

browser: See **browser server**.

browser client: A computer on the network that queries or sends information to a **browser server**. There are three types of **browser clients**: workstations, nonbrowser servers, and **browser servers**. In the context of browsing, nonbrowser servers supply information about themselves to **browser servers**, and workstations query **browser servers** for information. **Browser servers** can behave as nonbrowser servers and as workstations.

browser server: An entity that maintains or could be elected to maintain information about other servers and **domains**.

bucket rate: A value in a **TSpec** that is used to specify an aspect of network traffic behavior, as specified in [RFC2212].

built-in administrator: A built-in account for administering the computer/**domain**.

built-in domain: A **domain object** with the issuing authority **security identifier (SID)** of S-1-5-32.

built-in domain security identifier: The **security identifier (SID)** of the built-in **domain object**.

built-in principal: A default **security principal** whose **security identifier (SID)** is identical in every **domain**.

bus: Computer hardware to which peripheral devices may be connected. Messages are sent between the CPU and the peripheral devices using the **bus**. Examples of **bus** types include **SCSI**, **USB**, and 1394.

bus type: A type of **bus**. Examples of **bus types** include **SCSI**, **USB**, and 1394.

5 C

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

canonical name: A syntactic transformation of an **Active Directory distinguished name (DN)** into something resembling a path that still identifies an **object** within a **forest**. **DN** "cn=Peter Houston, ou=NTDEV, dc=microsoft, dc=com" translates to the canonical name "microsoft.com/NTDEV/Peter Houston", while the **DN** "dc=microsoft, dc=com" translates to the canonical name "microsoft.com/".

CAPI: See **Cryptographic Application Programming Interface (CAPI) or CryptoAPI**.

causality identifier (CID): A **GUID** that is passed as part of an **ORPC** call to identify a chain of calls that are causally related.

CEIP: See **Customer Experience Improvement Program (CEIP)**.

CEIP data: Anonymous information contained in a set of files that describes usability, performance, reliability, and quality metrics. This data is used by a **Customer Experience Improvement Program (CEIP)**.

certificate: (1) A **certificate** is a collection of attributes and extensions that can be stored persistently. The set of attributes in a **certificate** may 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 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)** 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 on 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)** 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.

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

certificate chain: A sequence of **certificates**, where each **certificate** in the sequence is signed by the subsequent **certificate**. The last **certificate** in the chain is normally a self-signed **certificate**.

certificate issuance: See **certification**.

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

certificate revocation: The process of invalidating a **certificate**. For more information, see [RFC3280] section 3.3.

certificate revocation list (CRL): A list of **certificates** 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** also contains the revocation reason for each **certificate** and the time the **certificate** was revoked. As specified 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 section 7.3 of [X509], [MSFT-CRL], and section 5 of [RFC3280].

certificate services: The Microsoft implementation of a **certification authority (CA)** that is part of the server operating system. **Certificate services** include tools to manage issued **certificates**, publish **CA certificates** and **CRLs**, configure **CAs**, import and export **certificates** and keys, and recover archived **private keys**.

certificate store: A database of **certificates**, or **certificates** and the accompanying **private key**. Used to store a variety of **certificates** with different attributes or constraints.

certificate template: A list of attributes that define a blueprint for creating an **X.509 certificate**. It is often referred to in non-Microsoft documentation as a "certificate profile". A **certificate template** is used to define the content and purpose of a **digital certificate**, including issuance requirements (certificate policies), implemented **X.509** extensions such as application policies, key usage, or extended key usage as specified in [X509], and enrollment permissions. Enrollment permissions define the rules by which a **certification authority (CA)** will issue or deny certificate requests. In Windows environments, **certificate templates** are stored as **objects** in the **Active Directory** and used by Microsoft enterprise **CAs**.

certification: The **certificate** request and issuance process whereby an **end entity (EE)** first makes itself known to a **certification authority (CA)** (directly, or through a registration authority) through the submission of a certificate enrollment request, prior to that **CA** issuing a certificate or certificates for that **EE**.

certification authority (CA): (1) 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.

(2) A software component that issues digital (**X.509**) **certificates** to identities based on a **public**/**private key** pair. For more information, see [RFC3280].

challenge: A piece of data used to authenticate a user. Typically a challenge takes the form of a **nonce**.

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

challenge/response authentication: A common authentication technique in which a principal is prompted (the challenge) to provide some private information (the response) to facilitate authentication.

Challenge-Response Protocol: A type of authentication protocol in which the authentication is carried by sending a challenge from one party to another, with the other party providing a response that proves its identity.

change journal: The database to which records of file or directory changes are written by the **NTFS** file system. Each volume on a system has its own change journal.

change order: A message that contains information about a file or folder that has changed on a replica. The change order is sent to the member's outbound partners. If the outbound partners accept the change, the partners request the associated staging file. After installing the changed file in their individual replica trees, the partners propagate the change order to their outbound partners.

chart data region: A report item on a report layout that displays data in a graphical format.

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 object, children: An **object** that is not the root of its tree. The children of an **object o** are the set of all **objects** whose parent is **o**.

chunks: The pieces of a file defined by the cut points.

CIM: See **Common Information Model (CIM)**.

cipher: A cryptographic algorithm used to encrypt and decrypt files and messages.

ciphersuite: A set of cryptographic algorithms used to encrypt and decrypt files and messages.

ciphertext: The encrypted form of a message. **Ciphertext** is achieved by encrypting the **plaintext** form of a message, and can be transformed back to **plaintext** by decrypting it with the proper key. Without that transformation, a **ciphertext** contains no distinguishable information.

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

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**.

class store container distinguished name (class store container DN): A **distinguished name (DN)** of the form "CN=Class Store,<scoped gpo dn>" where <scoped gpo dn> is a Scoped **Group Policy object (GPO) DN**. The class store container **DN** refers to an **object** of objectClass "classStore" in the **Active Directory** schema.

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.

client area: (1) The area of the desktop that is available for a window or notification icon to paint on.

(2) In an application, the display area that is used to create data, such as drawing or typing functions. The **client area** does not include toolbars, menus, or status bars.

client challenge: A 64-bit **nonce** generated on the client side.

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].

client context: A context describing an execution environment from which an activation request has originated.

client locator: A service that enables lookup of entries exported to the **remote procedure call (RPC)** name service.

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 extension GUID (CSE GUID): A well-known **GUID** that associates a specific client-side Group Policy plug-in with a set of policy settings that can be stored in a **Group Policy object (GPO)**.

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

cluster name: The computer name that is associated with a cluster, rather than with a single computer system.

cluster size: See **allocation unit size**.

cluster state: A state that consists of all the non-volatile configuration data and volatile current status data that is maintained by the **cluster** and accessible to active nodes.

CN: See **common name (CN)**.

CNG: See **Cryptography API: Next Generation (CNG)**.

coalesced payload: A special form of payload that consists of multiple traditional payloads combined into a single packet.

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 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.

collision-resistant hash function: A hash function having the property that, in practice, differing inputs do not produce the same hash (that is, they do not collide).

color profile: A file that contains information about how to convert colors in the color space and the color gamut of a specific device into a device-independent color space. A device-specific color profile is called a "device profile". For more information on using color and device profiles, see [MSDN-UDP].

COM: See **Component Object Model (COM)**.

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

commit request: The action that is performed by a root application to initiate the Two-Phase Commit Protocol for an atomic transaction.

Common Information Model (CIM): An object-oriented information model that provides a conceptual framework for describing management data, as specified in [DMTF-DSP004].

Common Information Model (CIM) class: A collection of **Common Information Model (CIM)** instances that support the same type, that is, the same CIM properties and **CIM methods**, as specified in [DMTF-DSP004].

Common Information Model (CIM) instance: Provides values for the **CIM** properties associated with the **CIM instance's** defining **CIM class**. A **CIM instance** does not carry values for any other **CIM** properties or **CIM methods** that are not defined in (or inherited by) its defining **CIM class**. For more information, see [DMTF-DSP004].

Common Information Model (CIM) method: An operation describing the behavior of a **CIM class** or a **CIM instance**. It is generally an action that can be performed against the manageable entity made of a **CIM class**.

Common Information Model (CIM) namespace: A logical grouping of a set of **Common Information Model (CIM)** classes designed for the same purpose or sharing a common management objective within the database used to store all CIM class definitions. This is a term mostly referenced in the Windows Management Instrumentation (WMI) implementation.

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 Information Model (CIM) Object Manager (CIMOM): A component that implements a set of operations used to access and manipulate **Common Information Model (CIM) objects**.

Common Information Model (CIM) path: A string expression locating a class or an instance of a class in the operating system. The **CIM path** includes the computer name, the namespace, the name of **CIM class**, and the unique identifier locating the **CIM class** or **CIM instance**.

Common Information Model (CIM) property: Assigns values used to characterize instances of a **CIM class**. A **CIM property** can be thought of as a pair of **Get** and **Set** functions that, when applied to an object, return state and set state, respectively. For more information, see [DMTF-DSP004].

Common Information Model (CIM) qualifier: Used to characterize named elements, as specified in [DMTF-DSP004]. For example, there are **CIM qualifiers** that define the characteristics of a **CIM property** or the key of a **CIM class**.

Common Information Model (CIM) relative path: A string expression where elements like the computer and/or the namespace of the **CIM class** and/or **CIM instance** are not used.

common name (CN): A string attribute of a **certificate** that is one component of a **distinguished name (DN)**. 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 Web site or e-mail 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.

Compact Disc File System (CDFS): A file system used for storing files on CD-ROMs.

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].

compression chunk: When compression is used for replication data, the data is divided into smaller units that are suitable for the particular algorithm. The chunk size is specific to the compression algorithm being employed.

computer account: See **machine account**.

computer account object: An **object** **o** of class **user** such that **o.userAccountControl** and ADS_UF_WORKSTATION_TRUST_ACCOUNT ≠ 0.

computer name: The **DNS** or **NetBIOS** name.

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.

computer policy mode: A mode of policy application intended to retrieve settings for the computer account of the client.

computer-scoped Group Policy object distinguished name: A scoped **Group Policy object (GPO) distinguished name (DN)** that begins with "CN=Machine".

computer-scoped Group Policy object path: A scoped **Group Policy object (GPO) path** that ends in "\Machine".

configuration naming context (config NC): A **naming context (NC)** containing configuration information. In **Active Directory**, a single **config NC** is shared among all **domain controllers (DCs)** in the forest.

connection: (1) 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.

(2) 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].

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

(4) 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.

connection-oriented NTLM: A particular variant of **NTLM** designed to be used with connection-oriented **remote procedure call (RPC)**.

connection-oriented RPC: A **remote procedure call (RPC)** protocol dialect built on top of an **RPC** transport that supports connections. For more information, see [C706-Ch12RPC_PDU_Encode].

connection security rule: A group of settings that specify how and when connections into and out of a client computer should be protected using **Internet Protocol security (IPsec)**.

connection type: A specific set of interactions between participants in an OleTx protocol that accomplishes a specific set of state changes. A connection type consists of a bidirectional sequence of messages that are conveyed by using the MSDTC Connection Manager: OleTx Transports Protocol and the MSDTC Connection Manager: OleTx Multiplexing Protocol

transport protocol. A specified transaction typically involves many different connection types during its lifetime.

ConnectionId: A **GUID** that uniquely identifies a connection.

connectionless NTLM: A particular variant of **NTLM** designed to be used with connectionless **RPC**.

connectionless RPC: An **RPC** protocol dialect built on top of an **RPC** transport that does not support connections. For more information, see [C706-Ch12RPC_PDU_Encode].

constrained delegation: A Windows feature used in conjunction with **S4U2proxy**. This feature limits the proxy services for which the application service is allowed to get tickets on behalf of a user.

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

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-AppendixAUUID]. In addition, the **UUIDs** must be compared, as specified in [C706-AppendixAUUID].

container: An **object** in the directory that can serve as the parent for other **objects**. In the absence of schema constraints, all **objects** would be **containers**. The schema allows only **objects** of specific classes to be **containers**.

content set: See **replicated folder**.

ContentSetId: The **GUID** assigned to a specific **replicated folder** within a **replica set**.

context: A collection of context properties that describe an execution environment.

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

context property: An attribute of an execution environment.

context property identifier: A **GUID** that identifies a **context property**.

control access right: (1) An extended access right that can be granted or denied on an **access control list (ACL)**.

(2) A variable access type with a specialized access **GUID** identifying the specific access type.

Control menu: See **Window menu**.

conversation callback: A **remote procedure call (RPC)** request/response message exchange initiated by an **RPC Server** and received by an **RPC Client**. The message exchange is internal to the connectionless **RPC** engine.

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).

Copychunk Resume Key: A 24-byte value generated by a **Server Message Block (SMB)** server in response to a request by an **SMB client** that uniquely identifies an open file on the **SMB** server. A **Copychunk Resume Key** is used by **SMB** server-side data movement operations between files without requiring the data to be read by the **client** and then written back to the server.

Note that this is different from the resume key specified in [MS-CIFS] section 2.2.6.2 that is returned by the server in response to a TRANS2_FIND_FIRST2 subcommand of an SMB_COM_TRANSACTION2 **client** request.

core transaction manager facet: The facet that acts as the internal coordinator of each transaction that is inside the transaction manager. The core transaction manager facet communicates with other facets in its transaction manager to ensure that each transaction is processed correctly. To accomplish this, the core transaction manager facet maintains critical transaction state, in both volatile memory and in a durable store, such as in a log file.

correlation: In an **Interface Definition Language (IDL)** file, the runtime properties of one argument dictate the allowed runtime properties of another argument.

crash dump file: A file that may be created by an operating system when an unrecoverable fault occurs. This file contains the contents of memory at the time of the crash and may be used to debug the problem.

credential: Previously established authentication data, such as a password, that is used by a **security principal** to establish its own identity. When used in reference to the Netlogon Protocol, it is the data stored in the NETLOGON_CREDENTIAL structure.

critical object: A subset of the **objects** in the default **naming context (NC)**, identified by the attribute **isCriticalSystemObject** having the value TRUE. The **objects** that are marked in this way are essential for the operation of a **domain controller (DC)** hosting the **NC**.

cross-certification: The **certificate** issuance process by which two **certificate authorities (CAs)**, **CA1** and **CA2**, issue specialized certificates so that any **relying party (RP)** that has **CA1** in its trust root but not **CA2** can link from **CA1** to **CA2** and thereby validate **certificates** in the hierarchy under **CA2** and make use of those. For more information on cross-certification, see section 3.5 of [RFC3280]. For an introduction to cross-certificates and cross-certification, see [MSFT-CROSSCERT].

cross-certificate: An **X.509 digital certificate** issued between two existing independent **certificate authorities (CAs)** for the purpose of extending or constraining **PKI** trust hierarchies. A cross-certificate is specified in section 3.3.21 of [X509]. For an introduction to cross-certificates and cross-certification, see [MSFT-CROSSCERT].

crossRef object: An **object** residing in the partitions container of the **config NC** that describes the properties of a **naming context (NC)**, such as its **domain naming service name**, operational settings, and so on.

Cryptographic Application Programming Interface (CAPI) or CryptoAPI: The Microsoft cryptographic application programming interface (API). An API that enables application developers to add authentication, encoding, and encryption to Windows-based applications.

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.

cryptographic service provider (CSP): A software module that implements cryptographic functions for calling applications.

cryptographically generated address (CGA): An IPv6 address for which the interface identifiers (the low-order 64 bits) are generated by computing a **cryptographic hash function** on a **public key**. The corresponding **private key** can be used to sign messages sent from this IPv6 address. **CGA** is specified in [RFC3972].

Cryptography API: Next Generation (CNG): The second generation of the **CryptoAPI** and its long-term replacement. **CNG** allows the implementer to replace existing algorithm providers with the implementer's own providers and to add new algorithms as they become available. **CNG** also allows the same APIs to be used from user and kernel mode applications.

curly braced GUID string: The string representation of a 128-bit **globally unique identifier (GUID)** using the form {XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX}, where X denotes a hexadecimal digit. The string representation between the enclosing braces is the standard representation of a GUID as defined in [RFC4122] section 3. Unlike a **GUIDString**, a **curly braced GUID string** includes enclosing braces.

CurrentRefreshTime: The current time, in units of days, measuring the time since the value was initialized.

Customer Experience Improvement Program (CEIP): A program in which participating systems send information to a software publisher about how they use certain products. Received **CEIP data** is combined to help the software publisher solve problems and to improve the products and features that customers use most often.

cut points: The locations in a file where **remote differential compression (RDC)** has determined boundary points between blocks, or chunks. The cut points for a particular file depend on the contents of the file and the parameters with which **RDC** is running.

cycle: A series of one or more replication responses associated with the same invocation ID, concluding with the return of a new **update sequence number (USN)** that defines the high water mark in which to begin the next replication cycle.

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.

cylinder: The set of disk tracks that appear in the same location on each platter of a disk.

6 D

DACL: See **discretionary access control list (DACL)**.

Data Encryption Standard (DES): A specification for encryption of computer data that uses a 56-bit key developed by IBM and adopted by the U.S. government as a standard in 1976.

data manipulation language (DML): The subset of SQL statements that is used to retrieve and manipulate data.

data recovery agent (DRA): A logical entity corresponding to an asymmetric key pair, which is configured as part of **Encrypting File System (EFS)** administrative policy by an administrator. Whenever an **EFS** file is created or modified, it is also automatically configured to give authorized access to all **DRAs** in effect at that time.

data source: 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.

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.

database object: A representation of a named set of attribute value pairs that a protocol exposes.

database serial number: A numeric value that is incremented each time a database transaction is applied to the database.

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.

DAV: See **Distributed Authoring and Versioning**.

DC: See **domain controller**.

DCOM: See **Distributed Component Object Model (DCOM)**.

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

default naming context replica (default NC replica): The full domain **naming context (NC)** replica hosted by a **domain controller (DC)**. The default **NC** always contains the **DC's** computer object.

delta time: A negative FILETIME. It represents a period of time, expressed in a negative number of 100-nanosecond time slices. For example, a period of 20 minutes is represented as -12000000000.

deserialize: See **unmarshal**.

desktop switch: The act of switching from one user desktop to another, or to the Windows Secure Desktop.

device: Any peripheral or part of a computer system that can send or receive data.

device interface: Device functionality exposed by a driver. Each **device interface** is a member of a **device interface class**. A driver can expose instances of zero, one, or more than one **device interface classes** for a device. For example, a device can have a joystick and a keypad, and the device's driver stack can expose instances of three interface classes for the device: a joystick, a keypad, and a combined joystick/keypad

device interface class: A way of exporting device and driver functionality to other components, including other drivers and user-mode applications. A driver can register a **device interface class**, and then enable an instance of the class for each device object to which user-mode I/O requests might be sent. On the highest level, a **device interface class** is a grouping of devices by functionality. Each **device interface class** is associated with a **GUID**. Vendors can create and define their own **GUIDs** for **device interface classes**.

device driver: The software that the system uses to communicate with a device such as a display, printer, mouse, or communications adapter. It is often referred to simply as a "driver".

DFS: See **Distributed File System (DFS)**.

DHCP: See **Dynamic Host Configuration Protocol (DHCP)**.

dictionary attack: A technique for defeating an authentication mechanism by systematically searching through a large number of possibilities to deduce shared secrets.

differentiated services code point (DSCP): A value in an IPv4 or IPv6 header that is used to select a particular set of quality-of-service behaviors, as specified in [RFC2474] section 3.

digest: The fixed-length output string from a one-way hash function that takes a variable-length input string and is probabilistically unique for every different input string.

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

digital fingerprint: See **hash function**.

digital signature: (1) A message authenticator that is typically derived from a cryptographic operation using an asymmetric algorithm and **private key**. When a symmetric algorithm is used for this purpose, the authenticator is typically called a **Message Authentication Code (MAC)**.

(2) A value generated using a digital signature algorithm, taking as input a **private key** and an arbitrary-length string, such that a particular verification algorithm is satisfied by the value, the input string, and the **public key** corresponding to the input **private key**. For more information, see [SCHNEIER] chapters 2 and 20.

directed change order: A **change order** that is directed to a single outbound partner and produced when the partner is doing a **vvjoin**, such as during initial synchronization.

directory: 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.

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

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

directory service (DS): 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**.

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 4 protocol: The **DirectPlay 4 protocol** is used by multiplayer games to perform low-latency communication between two or more computers.

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).

DirectPlay 8 application: A software process that communicates with one or more software processes over a communications network by using the **DirectPlay 8** family of protocols.

DirectPlay 8 client application: A **DirectPlay 8 application** seeking to connect to another **DirectPlay 8 application** that is hosting a **DirectPlay 8** session. When connected, the actual communication between nodes in a **DirectPlay 8** session may be client/server or peer to peer. The term "client" in this definition is meant to indicate the role that the **DirectPlay 8 client application** is taking in the **host** enumeration process, which is the **DirectPlay 8 application** that is seeking to find and connect to a **host** of a **DirectPlay 8** session.

DirectPlay 8 protocol: The **DirectPlay 8 protocol** is used by multiplayer games to perform low-latency communication between two or more computers.

DirectPlay 8 server application: A **DirectPlay 8 application** that is hosting a **DirectPlay 8** session. When connected, the actual communication between nodes in a **DirectPlay 8** session may be client/server or peer to peer. The term "server" in this definition is meant to indicate the role that the **DirectPlay 8 server application** is taking in the **host** enumeration process, which is the **DirectPlay 8 application** that is currently hosting a **DirectPlay 8** session.

DirectPlay 8 service provider: A service provider that may be implemented on top of the DirectPlay 8 protocol, as described in the DirectPlay 8 Protocol: Core and Service Providers Specification. When a message is passed through for processing, the DirectPlay 8 Protocol:

Host and Port Enumeration Specification interacts directly with the **DirectPlay 8 service provider**.

DirectPlay client: A **player** in a **DirectPlay** client/server game session that has a single established connection with a **DirectPlay server** and is not performing game session management duties. It also refers to a potential **player** that is enumerating available **DirectPlay servers** to join.

DirectPlay host: The **player** in a **DirectPlay** peer-to-peer game session that is responsible for performing game session management duties, such as responding to game session enumeration requests and maintaining the master copy of all the **player** and group lists for the game. It has connections to all **DirectPlay peers** in the game session.

DirectPlay Name Server (DPNSVR): A forwarding service for enumeration requests that eliminates problems caused by conflicts between port usages for multiple **DirectPlay** applications.

DirectPlay peer: A **player** in a **DirectPlay** peer-to-peer game session that has a connection to the **DirectPlay host** and all other **peers** in the game session that are not performing game session management duties. It also refers to a potential **player** that is enumerating available **DirectPlay hosts** to join.

DirectPlay server: The **player** in a **DirectPlay** client/server game session that is responsible for performing game session management duties, such as responding to session enumeration requests and maintaining the master copy of all the **player** and group lists for the game. It has connections to all **DirectPlay clients** in the game session.

DirectPlay protocol: Refers to either the **DirectPlay 4** or the **DirectPlay 8** protocol.

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 a diagnostic utility included with Windows that is used to test Microsoft DirectX functionality, including DirectPlay traffic.

DirectX runtime: A set of libraries created for the family of Windows operating systems that provide interfaces to ease the development of video games.

DirectX Software Development Kit (DirectX SDK): A set of libraries, called the **DirectX runtime**, and supporting infrastructure for building applications for those libraries.

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

disk: A persistent storage device that can include physical hard disks, removable disk units, optical drive units, and **logical unit numbers (LUNs)** unmasked to the system.

disk adapter: Computer hardware that controls a disk.

disk adapter name: A string of characters returned by a disk adapter. This string of characters is provided by the vendor of the disk adapter and identifies the adapter make or model.

disk block: A logical unit consisting of a fixed number of contiguous sectors. Block sizes range from 512 bytes to several kilobytes.

disk controller: Computer hardware that controls a disk.

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.

disk extent: A contiguous set of one or more disk sectors. A disk extent can be used as a partition or part of a volume, or it can be free, which indicates that it is not in use or that it may be unusable for creating partitions or volumes.

disk geometry: The disk's three-dimensional address space: a sector address consists of a cylinder number, the track number within the cylinder, and the sector number on that track.

disk group: In the context of **dynamic disks**, this term describes a logical grouping of disks.

disk group import: The act of merging a set of disks belonging to one disk group into another set of disks belonging to a second disk group. The result is a single disk group that includes all disks involved in the import.

disk group name: A unique string of characters that is used to identify a disk group.

Disk Management Remote Protocol: The protocol used to configure disks and volumes in Windows 2000 and Windows XP. For more information, see [MS-DMRP].

disk modification sequence number: See **modification sequence number**.

disk pack: See **disk group**.

disk platter: A circular disk on which magnetic data is stored. A hard disk drive consists of several platters mounted onto a spindle that spins the disks for a magnetic head to read and write.

disk regions: See **disk extent**.

disk signature: A unique identifier for a disk. For a **master boot record (MBR)**-formatted disk, this identifier is a 4-byte value stored at the end of the **MBR**, which is located in sector 0 on the disk. For a **GUID partitioning table (GPT)**-formatted disk, this value is a **GUID** stored in the **GPT** disk header at the beginning of the disk.

disk type: A disk that is hardware-specific. A disk can only communicate with the CPU using a bus of matching type. Examples of bus types include SCSI, USB, and 1394.

disk vendor name: A string of characters returned by a disk that identifies the disk maker.

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 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, 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.

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

(3) 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].

(4) 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 Authoring and Versioning (DAV): A series of extensions to HTTP that define how basic file functions such as copy, move, delete, and create folder are performed across HTTP.

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 (DFS) client: A computer that is used to access a **DFS** namespace. It also can refer to the **DFS** software on a **client** that accesses the **DFS** namespace.

Distributed File System (DFS) client target failback: An optional feature that, when enabled, permits a **DFS client** to revert to a more optimal DFS target at an appropriate time after a **DFS client target failover**. The term "failback" refers to **DFS client target failback**. The DFS Referral Protocol, as specified in [MS-DFSC], describes the mechanisms by which a DFS server provides a list of DFS targets in decreasing order of optimality to the client.

Distributed File System (DFS) client target failover: When a **DFS referral** response has multiple targets, a **DFS client** attempts to find the first target that is both available and accessible. If the first DFS target in the list is not available or accessible, the **DFS client** determines whether the next target in the list is available and accessible. The client repeats this process until an available and accessible target is found or no more targets are left in the list of targets. **DFS clients** support **DFS client target failover** only for operations involving pathnames. The term "failover" refers to **DFS client target failover**. The DFS Referral Protocol, as specified in [MS-DFSC], describes the mechanisms by which a DFS server provides a list of DFS targets in decreasing order of optimality to the client.

Distributed File System (DFS) in-site referral mode: A mode in which **DFS root** or **DFS link** referral requests to a DFS server result in **DFS referral** responses with only those DFS targets in the same **Active Directory Domain Services (AD DS) site** as the **DFS client** requesting the **DFS referral**. When this mode is disabled, there is no restriction on the **AD DS site** of the targets returned in the referral response. This can be enabled per **DFS namespace**. If there are no DFS targets in the same **AD DS site** as the client, the **DFS referral** response may be empty.

Distributed File System (DFS) interlink: A special form of **DFS link** whose link target is a DFS domain-based namespace.

Distributed File System (DFS) link: A component in a **DFS path** that lies below the **DFS root** and maps to one or more **DFS link targets**. Also interchangeably used to refer to a **DFS path** that contains the **DFS link**.

Distributed File System (DFS) link target: The mapping destination of a **link**. A **link** target can be any **Universal Naming Convention (UNC)** path. For example, a **link** target could be a share or another **Distributed File System (DFS)** path.

Distributed File System (DFS) metadata: Information about a **Distributed File System (DFS)** namespace such as namespace name, **DFS links**, **DFS link** targets, and so on, that is maintained by a **DFS** server. For domain-based **DFS**, the metadata is stored in an **Active Directory Domain Services (AD DS)** object corresponding to the **DFS** namespace. For a

stand-alone **DFS namespace**, the **DFS root target** stores the **DFS metadata** in an implementation-defined manner; for example, in the registry.

Distributed File System (DFS) namespace: A virtual view of shares on different servers as provided by **DFS**. Each file in the namespace has a logical name and a corresponding address (path). A **DFS namespace** consists of a root and many **links** and targets. The namespace starts with a root that maps to one or more root targets. Below the root are **links** that map to their own targets.

Distributed File System (DFS) namespace, clustered: A standalone **DFS** namespace, which is hosted on a file server cluster.

Distributed File System (DFS) namespace, domain-based: A **DFS namespace** that has configuration information stored in the **Active Directory** directory service. The **DFS namespace** may span over a distributed system that is organized hierarchically into logical domains, each with a **domain controller (DC)**. The path to access the root or a **link** starts with the host domain name. A domain-based **DFS root** can have multiple root targets, which offers fault tolerance and load sharing at the root level.

Distributed File System (DFS) namespace, standalone: A **DFS namespace** that has metadata stored locally on the host server. The path to access the root or a **link** starts with the host server name. A stand-alone **DFS root** has only one root target. Stand-alone roots are not fault-tolerant; when the root target is unavailable, the entire **DFS namespace** is inaccessible. Stand-alone **DFS roots** can be made fault tolerant by creating them on clustered file servers.

Distributed File System (DFS) path: Any **Universal Naming Convention (UNC)** path that starts with a **DFS root** and is used for accessing a file or directory in a **DFS namespace**.

Distributed File System (DFS) referral: A **DFS client** issues a **DFS referral** request to a **DFS root target** or a **DC**, depending on the **DFS path** accessed, to resolve a **DFS root** to a set of **DFS root targets**, or a **DFS link** to a set of **DFS link targets**. The **DFS client** uses the referral request process as needed to finally identify the actual **share** on a server that has accessed the leaf component of the **DFS path**. The request for a **DFS referral** is referred to as **DFS referral request**, and the response for such a request is referred to as **DFS referral response**.

Distributed File System (DFS) referral request: The request for a **DFS referral**.

Distributed File System (DFS) referral response: The response to a **Distributed File System (DFS) referral request**.

Distributed File System (DFS) referral site costing: When appropriately enabled for a **DFS namespace**, an optional feature that results in a **DFS referral** response. In the referral response, targets are grouped into sets based on increasing **Active Directory Domain Services (AD DS)** site cost from the **DFS client** that is requesting the referral to the **DFS** target server. When this feature is disabled, the referral response consists of at most two target sets: one set consisting of all **DFS** targets in the same **AD DS** site as the **DFS client**, and the other set consisting of **DFS** targets that are not in the same **AD DS** site as the **DFS client**.

Distributed File System (DFS) root: The starting point of the **DFS namespace**. The root is often used to refer to the namespace as a whole. A **DFS root** maps to one or more root targets, each of which corresponds to a share on a separate server. A **DFS root** has one of the following formats:

\\<ServerName>\<RootName>

\\<DomainName>\<RootName>

where <ServerName> is the name of the root target server hosting the **DFS namespace**; <DomainName> is the name of the domain that hosts the **DFS root**; and is <RootName> is the name of the root of a domain-based **DFS**.

The **DFS root** must reside on an NTFS volume.

Distributed File System (DFS) root scalability mode: Domain-based **DFS root** targets normally poll the **primary domain controller (PDC)** to check for any change in the **DFS metadata** of a **DFS namespace**. When the **DFS** server on a **DFS root** target supports this mode, and it is enabled for a **DFS namespace**, the **DFS** server instead polls a **domain controller (DC)** closer to it in terms of **Active Directory Domain Services (AD DS)** site cost.

Distributed File System (DFS) root target: A server that hosts a **DFS root** of a **DFS namespace**. A domain-based DFS namespace can have multiple **DFS root targets**; a standalone DFS namespace can have only one **DFS root target**.

Distributed File System Remote Procedure Call (DFS-RPC): The **remote procedure call (RPC)** interfaces and methods that make up the Microsoft Distributed File System, Server-To-Server Protocol.

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 Link Tracking (DLT): A protocol that enables **client** applications to track sources that have been sent to remote locations using **remote procedure call (RPC)** interfaces, and to maintain **links** to files. It exposes methods that belong to two interfaces, one of which exists on the server (trksvr) and the other on a workstation (trkwks).

Distributed Transaction Coordinator (DTC): A Windows service that coordinates transactions across multiple databases.

DLT: See **Distributed Link Tracking (DLT)**.

DML: See **data manipulation language (DML)**.

DN: See **distinguished name (DN)**.

DNS: See **Domain Name System (DNS)**.

document type definition (DTD): A language that can be used to define the rules of an **XML** document, as specified in [XML] section 2.8.

domain: 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.

domain account: A stored set of attributes representing a principal that is used to authenticate a user or machine to an **Active Directory** domain.

domain admins: A group with a **security identifier (SID)** with the **relative ID** value of 512 in the account domain.

domain controller (DC): The service, running on a **server**, that implements **Active Directory**, or the **server** hosting this service. For more information on **domain controllers** and **Active Directory**, see [MS-ADTS].

domain controller account object: An object in the directory that represents the computer in the role of a **domain controller (DC)**. A DC account is an object **O** in the default **naming context (NC)** replica of a server such that **O** is of class **computer** and (O.userAccountControl and ADS_UF_SERVER_TRUST_ACCOUNT ¹0).

domain controller locator: A function within a **domain** that provides for location of a **domain controller (DC)** and the ability to determine certain properties of **DCs**. For more information, see [MS-ADTS].

domain controllers (DCs): A well-known set of machines that host domain-wide information.

domain database: A database where security principal information is stored. This database is the directory service **Active Directory** in the case of a **domain controller (DC)** running on a Windows machine. On a Windows machine that is not a **DC**, this database is a local database, manageable through the Security Accounts Manager Remote Protocol, as specified in [MS-SAMR].

domain local group: A security group that is only valid for inclusion within **access control lists (ACLs)** from its own domain. Its membership may include users, global groups, and universal groups from any domain. It may additionally include, and be a member of, other domain local groups from within its domains.

domain master browser: A server responsible for combining information for an entire domain, across all subnets.

domain master browser server: A **master browser server** that is responsible for combining information for an entire domain, across all subnets. A **domain master browser server** is responsible for keeping multiple subnets in synchronization by periodically querying **local master browser servers** for information concerning user accounts, security, and available resources such as printers.

domain member (member machine): A machine that is joined to a domain by sharing a secret between the machine and the domain.

domain name: (1) A name with a structure indicated by dots.

(2) A **domain name (1)** used by the **Domain Name System (DNS)**.

(3) A **domain name (2)** or a **NetBIOS name** that identifies a **domain**.

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

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**.

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

domain object: (1) A unit of data storage in a **domain** that is maintained and made available to **domain** members by a **domain controller (DC)**.

(2) A database object that represents an issuing authority as specified in [MS-WSO] section 3.1.2.1.3. An account is said to be "in" a particular **domain** if the **domain** prefix of its **security identifier (SID)** is the **SID** of the particular **domain**.

domain of interpretation (DOI): A domain that defines the manner in which a group of protocols uses the **ISAKMP** (as specified in[RFC2408]) framework to negotiate **security associations (SAs)** (for example, identifiers for cryptographic algorithms, interpretation of payload contents, and so on). For example, the Internet Protocol security (IPsec) **DOI** (as specified in [RFC2407]) defines the use of the **ISAKMP** framework for protocols that negotiate **main mode (MM)** and **quick mode (QM) security associations (SAs)**. Both **Internet Key Exchange (IKE)** and **AuthIP** fall under the IPsec **DOI**.

domain prefix: A **security identifier (SID)** of a **domain** without the relative identifier (rid) portion. The domain prefix refers to the issuing authority **SID**. For example, the domain prefix of S-1-5-21-397955417-626881126-188441444-1010 is S-1-5-21-397955417-626881126-188441444.

domain tree: A set of **domains** that are arranged hierarchically, typically following an accompanying DNS hierarchy, with trusts between parents and children. An example **domain tree** might be **a.example.com**, **b.example.com**, and **example.com**; domain A and domain B each trust **example.com** but do not trust each other directly. They will have a transitive trust relationship through **example.com**.

domain trust relationship: A relationship in which one domain trusts the directory information stored in another domain. The domain that does the trusting is called the "trusting domain", while the domain that contains the information being trusted is called the "trusted domain".

domain user: A user with an account in the **domain's** user account database.

downlevel trust: A trust in which one of the peers is running Windows NT 4.0.

downstream partner: The partner that receives change orders, files, and folders.

DPNID: A 32-bit identification value assigned to a DirectPlay player as part of its participation in a DirectPlay game session.

drive: See **volume**.

drive letter: One of the 26 alphabetical characters A-Z, in uppercase or lowercase, that is assigned to a volume. Drive letters serve as a namespace through which data on the volume can be accessed. A volume with a drive letter can be referred to with the drive letter followed by a colon (for example, C:).

drive path: See **mounted folder**.

driver package: A collection of the files needed to successfully load a driver. This includes the device information (.inf) file, the catalog file, and all of the binaries that are copied by the .inf file.

driver store: A secure location on the local hard disk where the entire driver package is copied.

DS: See **directory service (DS)**.

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)).

(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:

1. If n.guid ≠ null, the unique object o such that o.objectGuid = n.guid (failing if no such object); otherwise

2. If n.dn ≠ null, the unique object o such that o.distinguishedName = n.dn (failing if no such object); otherwise

3. 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].

DXDiag: See **DirectX Diagnostic (DXDiag)**.

dynamic connection string: An expression that the implementer builds into the report, allowing the user to select which data source to use at run time. The implementer must build the expression and data source selection list into the report when the report is created.

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): A protocol that provides a framework for passing configuration information to hosts on a TCP/IP network, as defined in [RFC2131].

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

Dynamic Host Configuration Protocol (DHCP) scope: The full consecutive range of possible IP addresses for a network. Scopes typically define a single physical subnet on a network to which DHCP services are offered. Scopes also provide the primary way for the server to manage distribution and assignment of IP addresses and any related configuration parameters to **clients** on the network.

Dynamic Host Configuration Protocol (DHCP) server: A computer running a DHCP service that offers dynamic configuration of IP addresses and related information to DHCP-enabled **clients**.

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 provider: A **Virtual Disk Service (VDS)** provider that manages dynamic disks.

dynamic volume: A volume on a dynamic disk.

dynamic volume subdisk: See **disk extent**.

7 E

EAP: See **Extensible Authentication Protocol (EAP)**.

EAP identity: The identity of the **Extensible Authentication Protocol (EAP)** peer as specified in [RFC3748].

EAP method: An **authentication** mechanism that integrates with the **Extensible Authentication Protocol (EAP)**; for example, EAP-TLS, Protected EAP v0 (PEAPv0), EAP-MSCHAPv2, and so on.

EAP server: The backend **authentication server**; typically a RADIUS (as specified in [RFC2865]) server.

elliptic curve cryptography (ECC): A **public-key** cryptosystem that is based on high-order elliptic curves over finite fields.

empty CIM object: A data structure conforming to the Windows Management Instrumentation (WMI) serialization model having no properties, method, or derivation.

encapsulation: See **disk encapsulation**.

Encapsulating Security Payload (ESP): An **Internet Protocol security (IPsec) encapsulation** mode that provides **authentication**, data confidentiality, and message integrity. For more information, see [RFC4303] section 1.

encoding: 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.

Encrypting File System (EFS): The name for the **encryption** capability of the **NTFS** file system. When a file is encrypted using **EFS**, a **symmetric key** known as the **file encryption key (FEK)** is generated and the contents of the file are encrypted with the **FEK**. For each user or **data recovery agent (DRA)** that is authorized to access the file, a copy of the **FEK** is encrypted with that user's or **DRA's public key** and is stored in the file's metadata. For more information about **EFS**, see [MSFT-EFS].

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

encryption key: One of the input parameters to an **encryption** algorithm. Generally speaking, an **encryption** algorithm takes as input a clear-text message and a **key**, and results in a cipher-text message. The corresponding **decryption** algorithm takes a cipher-text message and the **key**, and results in the original clear-text message.

end entity (EE): The keyholder (person or computer) to whose **key** or name a particular **certificate** refers.

endpoint: (1) A **client** on the network that 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 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].

endpoint mapper: A service on a **remote procedure call (RPC) server** that maintains a database of **dynamic endpoints** and allows **clients** to map an interface/object UUID pair to a local **dynamic endpoint**. For more information, see [C706].

enforcement client: An **enforcement client** uses the health state of a computer to request a certain level of access to a network. For more information about **enforcement clients**, see [MSDN-NAP].

enhanced key usage (EKU): An extension that is a collection of **object identifiers (OIDs)** that indicate the applications that use the **key**.

enhanced metafile format (EMF): A file format that supports the device-independent definitions of images.

enhanced metafile format plus extensions (EMF+): A file format that supports the device-independent definitions of images.

enhanced metafile spool format (EMFSPOOL): A format that specifies a structure of **enhanced metafile format (EMF)** records used for defining application and device-independent printer **spool files**.

enlistment: The relationship between a participant and a **transaction manager** in an **atomic transaction**. The term typically refers to the relationship between a **resource manager** and its **transaction manager**, or between a **subordinate transaction manager** facet and its **superior transaction manager** facet.

enroll/enrollment: See **certification**.

enrollment permissions: A list of administrator-defined rights or **access control lists (ACLs)** that define the capability of a given **client** (user, machine, or device). **Enrollment permissions** can define a **client** capability to read a **certificate template**, write a **certificate template**, enroll for a **certificate** based on a specified **certificate template**, auto-enroll for a **certificate** based on a specified **certificate template**, or change permissions on a **certificate template**. **Enrollment permissions** are stored on a **certificate template** and are enforced by the **certificate authority (CA)**. For more information, see [MSFT-TEMPLATES].

enterprise certificate authority: A **certificate authority (CA)** that is a member of a **domain** and that uses the **domain's Active Directory** service to store policy, **authentication**, and other information related to the operation of the **certificate authority (CA)**.

environment variables: A set of string name/value pairs that are used to abstract host-specific parameters, such as the location of the operating system or installed binaries.

envoy context: A context that is marshaled and returned to a **client** as a result of obtaining an object reference.

error code: An integer that indicates success or failure. In Microsoft implementations, this is defined as a Windows error code. A zero value indicates success; a nonzero value indicates failure.

error correction object: A block of data that contains the error correction data specified by the forward error correction (FEC) algorithm. When included, it is located between the ASF Data Packet Error Correction Data and the MSBPACKETHEADER.

error record: A structured description of an occurrence of an error.

error sequence: An ordered sequence of **error records**, such that **error record** N+1 is the immediate error cause for **error record** N.

exchange: A pair of messages, consisting of a request and a response.

exchange certificate: A **certificate** that can be used for **encryption** purposes. This **certificate** can be used by **clients** to encrypt their private keys as part of their **certificate** request. In Windows environments, an enterprise **certificate authority (CA)** creates an **exchange certificate** periodically (by default, weekly), and returns the **exchange certificate** upon request of a **client**. For more information, see [MSFT-ARCHIVE].

exchange type: A specification of the format and number of messages in each phase of the **Internet Key Exchange (IKE)** protocol.

execution context: A context that is established when a process or thread is started. **Execution context** establishes the identity against which permissions to execute statements or perform actions are checked and is represented by a pair of **security tokens**: a primary token and an impersonation token.

expunge: To permanently remove an object from a **naming context (NC) replica**, without converting it to a **tombstone**.

extended key usage (EKU): An **X.509 certificate** extension that indicates one or more purposes for which the **certificate** may be used.

extended mode (EM): An optional phase of **AuthIP** negotiation during which the peers perform a second round of **authentication**. This phase does not exist in the **Internet Key Exchange (IKE)** protocol.

extended partition: A construct that is used to partition a disk into logical units. A disk may have up to four primary **partitions** or up to three primary **partitions** and one extended **partition**. The extended **partition** may be further subdivided into multiple logical drives.

Extensible Authentication Protocol (EAP): A framework for **authentication** that is used to provide a pluggable model for adding **authentication** protocols for use in network access **authentication**, as specified in [RFC3748].

Extensible Firmware Interface (EFI): A system developed by Intel designed to replace the BIOS. It is responsible for bootstrapping the operating system on **GUID** partitioning table disks.

extent: A contiguous area of storage in a computer file system, reserved for a file.

external trust: A type of **trust** that refers to a node **trusting** a domain that is outside the **forest** in which the node participates.

extrinsic event: An event that is generated by a component outside the implementation.

8 F

facet: In **OleTx**, a subsystem in a **transaction manager** that maintains its own per-**transaction** state and responds to intra-**transaction manager** events from other **facets**. A **facet** can also be responsible for communicating with other participants of a **transaction**.

failback: A failback operation. The process of returning production to its original location after a primary system failure or scheduled maintenance period.

failover: A backup operation that automatically switches to a standby database, server, or network if the primary system fails or is temporarily shut down for servicing. **Failover** is an important fault tolerance function of mission-critical systems that rely on constant accessibility.

To the user, **failover** automatically and transparently redirects requests from the failed or down system to the backup system that mimics the operations of the primary system.

A **failover** operation is always followed by a **failback** operation, which is the process of returning production to its original location.

failover cluster: A set of independent computers that work together to increase the availability of services and applications.

fast reconnect: A shortcut negotiation that capitalizes on information exchanged in the initial **authentication** to expedite the reestablishment of a **session**.

FAT: See **file allocation table (FAT)**.

FAT file system: A **file system** used by MS-DOS and other Windows operating systems to organize and manage **files**. The **file allocation table (FAT)** is 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.

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.

fault-tolerant: The ability of computer hardware or software to ensure data integrity when hardware failures occur. **Fault-tolerant** features appear in many server operating systems and include mirrored **volumes** and RAID-5 **volumes**. A **fault-tolerant volume** maintains more than one copy of the **volume's** data. In the event of disk failure, a copy of the data is still available.

fault-tolerant mirror set: A **volume** configuration such that more than one copy of the **volume data** is maintained. Each copy of the data is placed on separate sets of disks. If a disk in one disk set fails, the **volume's** data is still available on the second set of disks.

fiber channel bus: A **bus** technology that uses optical fiber for communication.

Fid: A 16-bit value that the **Server Message Block (SMB)** server uses to represent an opened **file**, **named pipe**, printer, or device. A **Fid** is returned by an **SMB** server in response to a **client** request to open or create a **file**, **named pipe**, printer, or device. The **SMB** server guarantees that the **Fid** value returned is unique for a given **SMB** connection until the **SMB** connection is closed, at which time the **Fid** value may be reused. The **Fid** is used by the **SMB**

client in subsequent **SMB** commands to identify the opened **file**, **named pipe**, printer, or device.

file: 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**.

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 units: Units of a specific size that are used by the **file system** to allocate space on a disk for the **file system** used by the **volume**.

file attribute: A 32-bit bitmask containing information on a **file's** properties. For instance, 0x00000001 is used for the read-only **attribute**.

file/directory attributes: The **attributes** of a **file** or a directory, as specified in [MS-FSCC].

file encryption key (FEK): The symmetric key that is used to encrypt the data in an **Encrypting File System (EFS)**-protected **file**. The **FEK** is further encrypted and stored in the **file** metadata such that only authorized users can access it.

file event time: The time at which a change to a **file** or folder is made. This may not be the same as the **file** create or last-write time. Restoring a **file** from a backup tape preserves the **file** create and last-write times, but the **file event time** is the time when the actual **file** restoration was performed.

file extension: The sequence of characters in a **file's** name between the end of the **file's** name and the last "." character. Vendors of applications choose such sequences for the applications to uniquely identify **files** that were created by those applications. This allows file management software to determine which application should be used to open a **file**.

file GUID: An identifying property of a **file** or folder in a **replica tree**. **File Replication Service (FRS)** creates and manages **file GUIDs** that, along with the replication version number and event time, are stored in the **IDTable**. Each **file** and folder stores its **file GUID** as part of its **attributes**; therefore, corresponding **files** and folders across all replica set members have the same **file GUID**.

FileId: (1) A 64-bit value that is used to represent a **file**. The value of a **FileId** is unique on a single **volume** of a local **file system** or a remote file server. A **FileId** is not guaranteed to be unique across **volumes**, but the **file system** on the server must guarantee that it is unique within a given **volume** if **FileIds** are supported. **FileIds** are not supported by all local **file systems**. On Windows, **NTFS** supports **FileIds**, but the **file allocation table (FAT) file system** does not support them.

(2) The FileLocation of a **file** at the time it was originally created. A **file's FileId** never changes.

FileInformation: Information that is maintained about a **file**, such as its **FileId**.

FileLinkInformation: Information about a **file** that is necessary to identify and locate that **file**, including the **file's** last known **Universal Naming Convention (UNC)** name, the **MachineID** of the computer on which the **file** was last known to be located, the last known **FileLocation** of the **file**, and the **file's** permanent **FileID**.

FileLocation: A **VolumeID** with an appended **ObjectID**, which together represent the location of a **file** at some point in time, though the **file** may no longer be there.

file property: See **file attribute**.

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)**.

file stream: See **main stream** and **named stream**.

file system: 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.

file system control (FSCTL): A command issued to a **file system** to alter or query the behavior of the **file system** and/or set or query metadata that is associated with a particular **file** or with the **file system** itself.

file system extension: The act of extending the **file system** on a **volume** to include more disk sectors. If the size of the **volume** is larger than the size of the **file system** for that **volume**, the **file system** may be extended to manage more of the **volume's** disk extensions.

file system flags: A set of values used by a **file system** to configure and report **file system** features and operations.

file system label: A non-unique string of characters that the **file system** assigns to the **volume**, as specified by the user when formatting the **volume**.

FileTable: Maps a **FileLocation** or **FileID** to a current **FileLocation**.

FILETIME: A 64-bit value that represents a time, as specified in [MS-RPCE] Appendix A.

file version number: A property of a **file** and folder in a **replica tree** that is incremented each time the **file** or folder is updated. The **file version number** is used to resolve concurrent updates originating from more than one member of the **replica set**. The version number is incremented only by the member that originated the file update. Other members that propagate the update do not change the version number.

filter: (1) A setting that excludes subfolders (and their contents) or **files** from replication.

(2) In the context of the **Lightweight Directory Access Protocol (LDAP)**, the **filter** is one of the parameters in a search request. The **filter** specifies matching constraints for the candidate objects.

(3) A configuration on a **network access server (NAS)** that specifies the types of traffic that are acceptable for IP local host traffic. **Filters** can block or allow traffic by IP address, IP protocol, TCP port, or **User Datagram Protocol (UDP)** port.

Filter Max: The chunking algorithm that is used in **remote differential compression (RDC)** to split **files** into **chunks**. With **Filter Max**, chunk boundaries are determined by local maxima for a fixed horizon size h. The local maxima may be determined by comparing hash values summarizing a limited window of bytes around each **file** position.

finite state machine: A computer, or operating system, in which a set of inputs determines not only the set of outputs but also the internal state of a computer, so that processing is optimized.

firewall rule: A group of settings that specify which connections are allowed into and out of a **client** computer.

fixup: A location in a program image that depends on an absolute address. Because Win32 programs must be able to run at any address, the linker writes a **fixup** table to the Preinstallation Environment (PE) file containing a list of all such locations in the program. Windows will process the **fixup** table when the program is loaded. RTTarget-32 performs this function in the locate process.

fix-up servers: See **remediation server**.

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

flexible single master operation (FSMO): See **FSMO role**, **FSMO role owner**, and **FSMO object**.

flow: A TCP session or **User Datagram Protocol (UDP)** pseudosession, identified by a 5-tuple (source and destination IP and ports, and protocol). By extension, a request/response Internet Control Message protocol (ICMP) exchange (for example, ICMP echo) is also a **flow**.

folder: 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**.

folder redirection: The ability to change the location of certain predetermined **folders** in a **file system** from their default location to another location on the same machine or to a network storage location.

foreign: A dynamic disk group that is not part of a machine's primary disk group. The term **foreign** denotes "foreign to this machine". **Foreign** disk and **foreign** disk groups are not online. This means that these disks may not be configured and no data input/output (I/O) to the disks or the **volumes** on the disks is permitted.

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 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**.

(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**.

forest functional level: A specification of functionality available in a **forest**. It must be consistent with the Windows versions of the **domain controllers (DCs)** in the **forest**. The following table gives the constants and the specific operating systems on which they are supported.

Identifier	Domain controller operating systems that are allowed in the forest
DS_BEHAVIOR_WIN2000	Windows 2000 Server Windows Server 2003 Windows Server 2008 Windows Server 2008 R2
DS_BEHAVIOR_WIN2003_WITH_MIXED_DOMAINS	Windows Server 2003 Windows Server 2008 Windows Server 2008 R2
DS_BEHAVIOR_WIN2003	Windows Server 2003 Windows Server 2008 Windows Server 2008 R2
DS_BEHAVIOR_WIN2008	Windows Server 2008 Windows Server 2008 R2
DS_BEHAVIOR_WIN2008R2	Windows Server 2008 R2

These values are specified in [MS-ADTS] section 7.1.4.4.

forest trust information: Information about namespaces, **domain names**, and **security identifiers (SIDs)** owned by a trusted **forest**.

format: When a **volume** is formatted, metadata is written 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**.

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**.

forwardable: A flag, as specified in [RFC4120] section 2.6, used in an **S4U2self** KRB_TGS_REQ message to request that the resulting **service ticket** be marked as forwardable, allowing it to be used in a subsequent **S4U2proxy** KRB_TGS_REQ message.

free space: Space on a disk not in use by any **volumes**, primary partitions, or logical drives.

FrontPage: The FrontPage Server Extensions. These extensions are used by some **clients** to manage resources or documents on Microsoft's Web servers. These extensions are a series of CGI and POSTs for server-side processing.

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

FSCTL: See **file system control (FSCTL)**.

FSMO object: The **object** in the directory that represents a specific **FSMO role**. This **object** is a member of the **FSMO role** and contains the **fSMORoleOwner attribute**.

FSMO role: A set of **objects** that may be updated in only one **naming context (NC) replica** at any given time. For more information, see [MS-ADTS].

FSMO role abandon: A request to a **domain controller (DC)** "d". The effect is for d to request the current holder of a specified **FSMO role** to transfer the role to d. Abandon is typically initiated by the current role holder in anticipation of being unable to host the role, for example, because the **DC** is being decommissioned. Saying "DC x abandoned the y role" means that x, the current holder of role y, picked another **DC** "d" and made a **FSMO role** role y abandon request to d.

FSMO role owner: The **domain controller (DC)** holding the **naming context (NC) replica** in which the **objects** of a **FSMO role** can be updated.

FSMO role transfer: A request to a **domain controller (DC)** "d". If d is the current holder of the specified **FSMO role**, the effect is to transfer that role to the **client**; if d is not the current holder of the role, the effect is to update the **client's** role objects from d's replica, so that the **client** can try the request again on another **DC**.

full database synchronization: A mechanism for synchronizing an entire database record set on a particular replication partner.

full format: A format in which all data sectors for the **volume** are initialized at the time the **file system** metadata is created.

full NC replica: A **naming context (NC) replica** that contains all the **attributes** of the **objects** it contains. A **full replica** accepts originating updates.

full token: A **token** that consists of all administrator rights and privileges.

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**.

9 G

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

garbage collection: The process of identifying logically deleted **objects** (also known as **tombstones**) and **link values** that have passed their **tombstone** lifetime, and then permanently removing these **objects** from a **naming context (NC) replica**. **Garbage collection** does not generate replication traffic.

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

GC server: See **global catalog server**.

Generic Security Services (GSS): An Internet standard (as specified 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.

ghosting: Custom **client** behavior in which **file** contents are downloaded lazily in response to applications accessing **files**.

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 group: Also called domain global group. An **Active Directory** group that can appear in **access control lists (ACLs)** anywhere in the **forest**, and can contain other **global groups** and users from its own **domain**. Universal groups can contain domain global groups.

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**.

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)**.

Graphics Device Interface (GDI): A Windows API, supported on 16-bit and 32-bit versions of the operating system, that supports graphics operations and image manipulation on logical graphics objects.

Graphics Device Interface, Extended (GDI+): A Windows API, supported on 32-bit and 64-bit versions of the operating system, that extends **GDI** to include support for Bezier curves, gradient brushes, image effects, and **EMF+** metafiles.

group: A collection of **objects** that can be treated as a whole.

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** 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 e-mail distribution lists.

Group Policy: A mechanism that allows the implementer to specify managed configurations for users and computers in an **Active Directory** service environment.

Group Policy extension: A protocol mechanism that extends the basic capability of the **Group Policy** protocol as specified in [MS-GPOL].

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**.

Group Policy object (GPO) container version: A **Group Policy object (GPO)** version stored in the **Active Directory** portion of the **GPO**.

Group Policy object (GPO) distinguished name (DN): A **Lightweight Directory Access Protocol (LDAP) distinguished name (DN)** for an **Active Directory object** of **object class** **groupPolicyContainer**. All such **object** paths will be paths of the form "LDAP://<gpo guid>,CN=policies,CN=system,<rootdse>", where <rootdse> is the root **DN** path of the **Active Directory domain** and <gpo guid> is a **Group Policy object (GPO) GUID**.

Group Policy object (GPO) file system version: A **Group Policy object (GPO)** version stored in the file system portion of the **GPO**.

Group Policy object (GPO) GUID: A **curly braced GUID string** that uniquely identifies a **Group Policy object (GPO)**.

Group Policy object (GPO) path: A **domain**-based **Distributed File System (DFS)** path for a directory on the server that is accessible through the **DFS**/**SMB** protocols. This path will always be a **Universal Naming Convention (UNC)** path of the form: "\\<dns domain name>\sysvol\<dns domain name>\policies\<gpo guid>", where <dns domain name> is the **DNS domain name** of the **domain** and <gpo guid> is a **Group Policy object (GPO) GUID**.

Group Policy object (GPO) version: A version number that combines the user and machine **Group Policy object (GPO)** versions as one 32-bit quantity. The upper 16 bits of the integer are the user **GPO** version and the bottom 16 bits of the integer are the machine **GPO** version.

Group Policy server: A server holding a database of **Group Policy objects (GPOs)** that can be retrieved by other machines.

guest account: A security account available to users who do not have an account on the computer.

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

GUID partitioning table (GPT): A disk-partitioning scheme that is used by the Extensible Firmware Interface (EFI). **GPT** offers more advantages than **master boot record (MBR)** partitioning because it allows up to 128 **partitions** per disk, provides support for **volumes** up to 18 exabytes in size, allows primary and backup **partition tables** for redundancy, and supports unique disk and partition IDs through the use of **globally unique identifiers (GUIDs)**. Disks with **GPT** schemes are referred to as **GPT** disks.

GUID partitioning table (GPT) disk: A disk with **GUID partitioning table (GPT)** schemes.

GUID-based DNS name: The **domain naming service name** of a **domain controller (DC)**, constructed by concatenating the dashed string representation of the **objectGuid** of the **DC's** nTDSDSA **object**, the string "._msdcs.", and the syntactic transformation of the **root domain's distinguished name (DN)** to a **domain naming service name**.

GUIDString: A **globally unique identifier (GUID)** in the form of an **ASCII** or **Unicode** string, consisting of one group of eight hexadecimal digits, followed by three groups of four hexadecimal digits each, followed by one group of 12 hexadecimal digits. It is the standard representation of a **GUID** as defined 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: Any token that can be used to identify and access an object such as a device, file, or a window.

handshake: An initial negotiation between a **peer** and an **authenticator** that establishes the parameters of their transactions.

hard disk: A peripheral device that provides persistent data storage and does not have removable media.

hard disk drive: A **disk drive** that controls the positioning, reading, and writing of the **hard disk**.

hard disk physical name: An implementation-specific **path** that can be used to refer to a specific hard disk on a machine.

Hash-based Message Authentication Code (HMAC): A mechanism for message authentication using **cryptographic hash functions**. **HMAC** can be used with any iterative **cryptographic hash function** (for example, MD5 and SHA-1) in combination with a **secret shared key**. The cryptographic strength of **HMAC** depends on the properties of the underlying **hash function**.

hash function: A function that takes an arbitrary amount of data and produces a fixed-length result (a "hash") that depends only on the input data. A **hash function** is sometimes called a **message digest** or a **digital fingerprint**.

hash window: The length, in bytes, of the domain of the **rolling hash function**. That is, the parameter *n* in the definition of **rolling hash function**.

hashes and checksums: The collision-resistant substrate of a sequence of bytes. Well-known hash algorithms for computing hashes include MD4, **MD5**, and **SHA-1**.

health certificate: (1) A **digital certificate** that is used to authenticate the health status of a **Network Access Protection (NAP)** client.

(2) An **X.509** certificate that is used to certify the **health state** of a machine as determined by the **policies** administered for a network.

health certificate enrollment agent (HCEA): The client-side component in the Health Certificate Enrollment Protocol. The **HCEA** is responsible for receiving **health certificates** from a **health registration authority (HRA)**. This term can also be used to refer to the **client** machine in the Health Certificate Enrollment Protocol.

health ID: An identifier of a component or service that supplies host status information in a **statement of health (SoH)** message or that performs evaluation of host status information in a **Statement of Health Response (SoHR)** message.

health messages: The set of messages exchanged between peers or **clients** and servers. These messages are of the types **statement of health (SoH)** and **statement of health response (SoHR)**.

health policy server: An entity in a network that has network **policies** administered on it and that is capable of validating a **statement of health (SoH)** against the specified **policies**.

health registration authority (HRA): The server-side component in the Health Certificate Enrollment Protocol. The **HRA** is a **registration authority (RA)** that requests a **health certificate** from a **certification authority (CA)** upon validation of health.

health state: An abstract notion of the state of a machine that is used to indicate its compliance with network **policies**. Some examples of such state would include the state of the firewall on the machine, the version of the virus signature files for an antivirus application, and so on.

HexConvertedUnicodeString: A **Unicode string** created from a binary, byte-granular value. The string is created by converting each byte, starting with the most significant byte and ending with the least significant byte, into two **Unicode characters**. The characters are the hexadecimal representation of each nibble of the byte, starting with the high-order nibble.

hibernation image: An image that contains metadata required to support a Windows operating system feature known as hibernation. Hibernation allows a system's state to be preserved in persistent storage while the system is shut down.

HMAC: See **Hash-based Message Authentication Code (HMAC)**.

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**.

host bus adapter (HBA): (1) A hardware device that adapts the signals of one electronic interface to another.

(2) An **HBA** that can be discovered by way of the SNIA Common HBA API on the system. For more information, see [HBAAPI].

host migration: The protocol-specific procedure that occurs when the **DirectPlay peer** that is designated as the **host** or voice server leaves the **DirectPlay** game or voice session and another **peer** assumes that role.

HRESULT: An integer value that indicates the result or status of an operation. A particular **HRESULT** can have different meanings depending on the protocol using it. See [MS-ERREF] section 2.1 and specific protocol documents for further details.

HTTP client: A program that establishes connections for the purpose of sending requests, as specified in [RFC2616].

HTTP Internal Server Error: An **HTTP** response with status code 500, as specified in [RFC2616] section 6.1.1.

HTTP OK: An **HTTP** response with status code 200, as specified in [RFC2616] section 6.1.1.

HTTP proxy: An intermediary program that acts as both a server and a **client** for the purpose of making requests on behalf of other **clients**. For more information, see [RFC2616].

HTTP server: An application that accepts connections in order to service requests by sending back responses. For more information, see [RFC2616].

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

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

11 I

IDL: See **Interface Definition Language**.

IdTable: A table of **File Replication Service (FRS)** state information that contains an entry with version and identity information for each file and folder in the **replica tree**. It is used to keep track of all files in the **replica set** and their history.

Image Color Management (ICM): Technology that ensures that a color image, graphic, or text object is rendered as closely as possible to its original intent on any device despite differences in imaging technologies and color capabilities between devices.

immediate error cause: An error in a protocol layer within a software agent that prevents the successful completion of a task in the same or different protocol layer/software agent. Any error resulting from such failure is also said to be caused by the **immediate error cause**.

impersonation: The ability of an operating system process or thread to run temporarily in the **security context** of a specific caller and to gain authorized access to **resources** using that identity.

inbound: The network traffic flowing from the **client** to the **server**.

inbound connection: For a given **replica member**, a component of the NTFRS member object in **Active Directory** that identifies **inbound partners**. An **inbound connection** exists for each **inbound partner**.

inbound log: A queue storing pending change orders to be processed. As entries in the queue are processed, acknowledgments are sent to the **inbound partners**.

inbound partner: The partner that sends out change orders, files, and folders.

inbound trust: A state in which the **trusted domain** trusts the **primary domain** to perform operations such as name lookups and **authentication**.

incoming authentication: A mode in which each party (the reference party) verifies the identity of the other party, as specified in [RFC3748] section 7.2.1, but not vice-versa.

INF file: A file providing Windows Setup with the information required to set up a device, such as a list of valid logical configurations for the device and the names of driver files associated with the device.

information level: A number used to identify the **volume**, file, or device information being requested by a **client**. Corresponding to each **information level**, the **server** returns a specific structure to the **client** that contains different information in the response.

inheritance: See **object class inheritance**.

initial sync: The process that a new member to the **replica set** has to go through before it is allowed to synchronize with its **outbound partners**. Also called **vvjoin**.

initiator: The party that sends the first message of an **Internet Key Exchange (IKE)**.

inner EAP method: An **Extensible Authentication Protocol (EAP) method** that is tunneled within another **EAP method**.

in-site: In-site targets. Two or more targets sharing the same namespace as a **client**.

installation files: Files referenced in the metadata of a software installation package. **Installation files** are used to install the software described by the software installation package on **client** computers.

Integrated Drive Electronics (IDE) bus: A standard electronic interface used between a computer motherboard's **bus** and the computer's disk storage devices.

interactive logon: A software method in which the account information and credentials input by the user interactively are authenticated by a **server** or **domain controller (DC)**.

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**.

Interface Definition Language (IDL): The International Standards Organization (ISO) standard language for specifying the **interface** for remote procedure calls. For more information, see [C706-Ch4InterfaceDef].

interface identifier (IID): A **GUID** that identifies an **interface**.

interface pointer: A pointer to an **interface** that is implemented by an [MS-DCOM] **object**.

interface pointer identifier (IPID): A 128-bit number that uniquely identifies an **interface** on an **object** within an object exporter.

Internet host name: The name of a host as defined in [RFC1123] section 2.1, with the extensions described in [MS-HNDS].

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 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 4 (IPv4): An Internet protocol that has 32-bit source and destination addresses. **IPv4** is the predecessor of **IPv6**.

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** and privacy.

Internet SCSI (iSCSI): For terms related to **iSCSI**, see [RFC3720].

Internet Security Association and Key Management Protocol (ISAKMP): A cryptographic protocol specified in [RFC2408] that defines procedures and packet formats to establish, negotiate, modify and delete **security associations (SAs)**. It forms the basis of the **Internet Key Exchange (IKE)** protocol, as specified in [RFC2409].

Internetwork Packet Exchange (IPX): A protocol maintained by Novell's NetWare product that provides connectionless **datagram** delivery of messages. IPX is based on Xerox Corporation's Internetwork Packet protocol, XNS.

intersite topology generator (ISTG): A **domain controller (DC)** within a given **site** that computes a **naming context (NC) replica graph** for each **NC replica** on any **DC** in its **site**. This **DC** creates, updates, and deletes corresponding **nTDSConnection objects** for edges directed from **NC replicas** in other **sites** to **NC replicas** in its **site**.

intrinsic event: An event that defines an event generated by the implementation itself.

invocation ID: A unique number that identifies the version of the directory database that is running on the **domain controller (DC)**. Replication partners use the **invocation ID** and an **update sequence number (USN)** to determine the most current changes for replication.

IPsec administrative plug-in: The **Internet Protocol security (IPsec)** extension plug-in that operates as part of the group policy configuration tool that reads and writes **IPsec** policy using the Group Policy: IP Security (IPsec) Protocol Extension [MS-GPIPSEC].

IPsec client-side plug-in: The **Internet Protocol security (IPsec)** extension plug-in that operates on the **client** machine to retrieve the policy using the Group Policy: IP Security (IPsec) Protocol Extension [MS-GPIPSEC].

IPsec component: The implementation of the **Internet Protocol security (IPsec)**/**Internet Key Exchange (IKE)** functionality on a **client** machine. This is the component that the Group Policy: IP Security (IPsec) Protocol Extension [MS-GPIPSEC] configures with the **IPsec**/**IKE** policy that is transferred as part of the protocol.

IPsec Group Policy Extension: The group policy extension protocol that transfers **Internet Protocol security (IPsec)**/**Internet Key Exchange (IKE)** configuration information (**IKE** settings, **IPsec** framing configuration, and so on).

IPv4: See **Internet Protocol version 4 (IPv4)**.

IPv4 address in string format: A string representation of an **IPv4** address in dotted-decimal notation, as specified in [RFC1123] section 2.1.

IPv6: See **Internet Protocol version 6 (IPv6)**.

IPv6 address in string format: A string representation of an **IPv6** address as specified in [RFC4291] section 2.2.

ISAKMP payload: A modular building block for constructing **ISAKMP** messages. A payload is used to transfer information such as **security association (SA)** data, or **key** generation and authentication data. The presence and order of payloads in a packet is defined by and dependent upon the type of exchange specified in the **ISAKMP** header of the **ISAKMP** message. For more information, see [RFC2408] section 4.1.

iSCSI initiator: For terms related to **iSCSI initiator**, see [RFC3720].

iSCSI initiator adapter: For terms related to **iSCSI initiator adapter**, see [RFC3720].

iSCSI initiator portal: For terms related to **iSCSI initiator portal**, see [RFC3720].

iSCSI session: For terms related to **iSCSI session**, see [RFC3720].

ISO/OSI reference model: The **International Organization for Standardization Open Systems Interconnection (ISO/OSI) reference model** is a layered architecture (plan)

that standardizes levels of service and types of interaction for computers that are exchanging information through a communications network. Also called the OSI reference model.

issuer name: The name of the **certificate authority (CA)** that signed and issued a **certificate**. The name is an **X.509** format name, as specified in [X509].

ISTG: See **intersite topology generator (ISTG)**.

12 K

Kerberos: 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].

Kerberos authenticator: A record sent with a **ticket** to a server to certify the **client's** knowledge of the **session key** in the **ticket**; to help the server detect replay attacks by proving that the authenticator is recently constructed; and to help the two parties select additional **session keys** for a particular connection authenticated by **Kerberos**. The use of authenticators, including how authenticators are validated, is specified in [RFC4120] section 5.5.1. For more information, see [KAUFMAN].

Kerberos constrained delegation: A form of authentication delegation in which Kerberos can be used to impersonate users that send requests for certain services, as opposed to all services.

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**.

key agreement: A **key** establishment procedure in which the resulting secret keying material is a function of information contributed by two participants, so that no party can predetermine the value of the secret keying material independently from the contributions of the other parties. See also **key transport**. For more information, see [SP800-56A] section 3.1 and [IEEE1363] section 3.

key archival: Also referred to as **key escrow**. The process by which the entity requesting the **certificate** also submits the **private key** during the process. The **private key** is encrypted such that only a **key recovery agent** can obtain it, preventing accidental disclosure, but preserving a copy in case the entity is unable or unwilling to decrypt data.

key archival certificate: See **key recovery certificate**.

key derivation: The act of deriving a cryptographic **key** from another value (for example, the derivation of a cryptographic **key** from a password).

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 Windows 2000 Server, Windows Server 2003, Windows Server 2003 R2, Windows Server 2008, and Windows Server 2008 R2. It is a network service that supplies **tickets** to **clients** for use in authenticating to services.

key escrow: See **key archival**.

key establishment: See **key exchange**.

key exchange: A synonym for **key establishment**. The procedure that results in shared **secret keying** material among different parties. **Key agreement** and key transport are two forms of **key exchange**. For more information, see [CRYPTO] section 1.11, [SP800-56A] section 3.1, and [IEEE1363] section 3.

key exchange key: The **key** that is used to protect the **session key** generated by the **client**. The **key exchange key** is derived from the **response key** during **authentication**.

key handle: The **remote procedure call (RPC)** context handle to a **key**.

key recovery agent (KRA): A user, machine, or registration authority that has enrolled and obtained a **key recovery certificate**. A **KRA** is any entity that possesses a **KRA private key** and **certificate**. For more information on **KRAs** and the archival process, see [MSFT-ARCHIVE].

key recovery certificate: A **certificate** with the unique **object identifier (OID)** in the extended key usage extension for **key archival**.

key transport: A **key establishment** procedure whereby one party (the sender) selects a value for the **secret keying** material and then securely distributes that value to another party (the receiver).

keyed hash: A cryptographic hash computed over both a **symmetric key** and data, as specified in [RFC2617]. For more information, see [RFC2104].

keyed-hash Message Authentication Code: A **symmetric keyed** hashing algorithm used to verify the integrity of data to help ensure that it has not been modified while in storage or transit.

keyholder: The entity that holds a **private key** and is therefore capable of signing and decrypting. The **keyholder** of a **public key** is defined as the **keyholder** of the corresponding **private key**.

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

Knowledge Consistency Checker (KCC): An internal Windows component of the **Active Directory** replication that is used to create spanning trees for **domain controller** to **domain controller** replication and to translate those trees into a set of abstract variables.

KRB_AP_REQ/KRB_AP_REP: The request and response messages used in the **Authentication Protocol (AP) exchange**.

KRB_AS_REQ/KRB_AS_REP: The request and response messages used in the **Authentication Service (AS) Exchange**.

KRB_CRED exchange: The **Kerberos** subprotocol used by clients that require the ability to send credentials from one host to another. This exchange is initiated when a client sends a KRB_CRED message, as specified in [RFC4120] 3.6.

KRB_PRIV exchange: The **Kerberos** subprotocol used by **clients** that require confidentiality and the ability to detect modifications of the messages they exchange with a server in a **session** that is already established through the **Authentication Protocol (AP) exchange**. This exchange is initiated when a **client** sends a KRB_PRIV message, as specified in [RFC4120] section 3.5.

KRB_SAFE exchange: The **Kerberos** subprotocol used by **clients** to detect modifications of messages they exchange with a server in a **session** that is already established through the **AP exchange**. This exchange is initiated when a **client** sends a KRB_SAFE message, as specified in [RFC4120] section 3.4.

KRB_TGS_REQ/KRB_TGS_REP: The request and response messages used in the **ticket-granting service (TGS) exchange**.

13 L

LangID: See **primary language identifier**.

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 client computer.

language identifier: See **language code identifier (LCID)**.

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

LDAP connection: A TCP connection from a client to a server over which the client sends Lightweight Directory Access Protocol (LDAP) requests and the server sends responses to the client's requests.

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

Lightweight Directory Access Protocol (LDAP): The primary access protocol for **Active Directory** [MS-ADTS]. **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 (as specified in [MS-ADTS]).

lingering object: A **domain controller (DC)** that was offline for longer than the value of the **tombstone lifetime** can contain objects that have been deleted on other **domain controllers (DCs)** and for which **tombstones** no longer exist.

link: When the value of an attribute refers to a directory object, and the attribute's Attribute-Schema object has an even value for attribute **linkId**, then that attribute value is a link. Sometimes referred to as a forward link.

link attribute: A **forward link attribute** or a **back link attribute**.

link order: An integer that describes the precedence of a **Group Policy object (GPO)** associated with a **scope of management (SOM)**, compared to other **GPOs** associated with that **SOM**.

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

LinkValueStamp: The type of a **stamp** attached to a **link value**.

list data region: A report item on a report layout that displays data in a list format.

listening state: A server or proxy state in which the server or proxy is able to accept and respond to events coming from the network.

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

LM hash: A DES-based cryptographic hash of a clear text password.

local area network (LAN): A group of computers and other devices dispersed over a relatively limited area and connected by a communications link that enables any device to interact with any other device on the network.

local area network adapter (LANA): (1) A hardware device that facilitates network communication via support of the first two levels (physical and datalink) of the ISO OSI

standards, providing physical network access and low-level addressing capability through MAC addresses.

(2) In NetBios-specific implementations, a number that is used to identify the network adapter to which NetBIOS is bound.

local change order: A change order that is created because of a change to a file or folder on the local server. The local server becomes the originator of the change order and constructs a staging file.

local domain controller (DC): A **domain controller (DC)** on which the current method is executing.

local master browser: The browser on a given subnet that was elected to maintain the master copy of information related to a given domain. That is, different domains have different local master browsers on the same subnet.

local master browser server: A server that is elected **master browser server** on a particular subnet across a domain.

local maximum: A pair consisting of an offset i in a file and the hash value $h(b_{i\text{-Hash Window}} .. b_i)$ that has the property that for all $j \geq 0$, such that $i - \text{horizon} \leq j \leq i + \text{horizon}$, $j = i$ OR $h(b_{j\text{-Hash Window}} .. b_j) < h(b_{i\text{-Hash Window}} .. b_i)$, where for all $k < 0$, b_k is defined to be 0. **Local maxima** are used to find cut points by the remote differential compression (RDC) FilterMax algorithm.

Local Security Authority (LSA): A protected subsystem that authenticates and logs users onto the local system. **LSA** also maintains information about all aspects of local security on a system, collectively known as the **local security policy** of the system.

Local Security Authority (LSA) database: A Microsoft-specific terminology for the part of the user **account** database containing **account** privilege information (such as specific **account** rights) and domain security policy information.

local security policy: A collection of security settings present on a machine that affects how that machine regulates access to the resources it provides.

locale ID: See **language code identifier (LCID)**.

localizable: Anything that is specific to a language or country.

localizable information: The portion of information in a **Common Information Model (CIM)** class definition that could be specific to a language or country.

locally unique identifier (LUID): A 64-bit value guaranteed to be unique within the scope of a single machine.

locator: (1) In **remote procedure call (RPC)**, a component of the **remote procedure call name service** that runs on a given machine and facilitates the name service operations of exports and lookups.

(2) In the context of **domain controllers (DCs)**, the functionality encompassed by the DC Locator Protocol. In a broader context, the cooperative function between clients and **DCs**, which allow clients to locate the nearest **DC** that offers particular feature services.

locked partition: A partition that is inaccessible.

locked volume: A **volume** that is not accessible because it is locked by an application that needs exclusive access to the **volume**.

logical cluster number (LCN): The cluster number relative to the beginning of the volume. The first cluster on a volume is zero (0).

logical connection: The state maintained on client and server in association with a connectionId.

Logical Disk Manager (LDM): A subsystem of Windows that manages dynamic disks. Dynamic disks contain a master boot record (MBR) at the beginning of the disk, one **LDM** partition, and an **LDM** database at the end. The **LDM** database contains partitioning information used by the **LDM**.

logical drive: A set of disk **extents** that compose a **volume**.

logical partition: See **logical drive**.

logical unit number (LUN): A number that is used to identify a disk on a given disk controller.

Lost and Found container: A container holding objects in a given **naming context (NC)** that do not have parent objects due to add and remove operations that originated on different **domain controllers (DCs)**. The container is a child of the **NC** root and has RDN CN=LostAndFound in **domain NCs** and CN=LostAndFoundConfig in **config NCs**.

14 M

machine account: An **account** that is associated with individual client or server machines in an **Active Directory** domain.

machine connection: A connection to a printer (shared from a print server) on a client machine. A connection is displayed in the user interface as a printer. **Machine connections** are displayed for all users (in all user environments) of a particular client machine.

machine Group Policy object (GPO) version: A version number of the changes for the computer policy portion of a **Group Policy object (GPO)**. This is a 16-bit integer encoded in the lower 16 bits of a **GPO** version.

MachineID: A unique identifier that represents the identity of a computer.

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

mailslot: (1) A mechanism for one-way interprocess communications (IPC). For more information, see [MSLOT] and [MS-MAIL].

(2) In the NetBIOS protocol, refers to the datagram style of communication.

mailslot class: An indication of the expected service of the **mailslot**. Class 1 is guaranteed delivery, and class 2 is not guaranteed delivery.

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.

main stream: The place within a file where data is stored or the data stored therein. A **main stream** has no name. The **main stream** is what is ordinarily thought of as the contents of a file.

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-DSP004] section 3. The MOF text encoding is only used for illustrative purposes. The binary encoding can be translated to and from the MOF format.

mandatory type-length-value: An attribute that is required in an **statement of health (SoH)** or **statement of health response (SoHR)** message in order for that message to be valid and complete.

man in the middle (MITM): An attack that deceives a server or client into accepting an unauthorized upstream host as the actual legitimate host. Instead, the upstream host is an attacker's host that is manipulating the network so that the attacker's host appears to be the desired destination. This enables the attacker to decrypt and access all network traffic that would go to the legitimate host.

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).

marshaled server object (MSO): A **server object** that is created by a higher layer, and not in response to an incoming request. (See **server-activated object (SAO)** for more information on the latter.)

marshaling: The act of formatting COM parameters for transmission over a **remote procedure call (RPC)**. For more information, see [MS-DCOM].

masked disk: A disk that is invisible to the local machine, even though a physical connection exists between the disk and the machine.

mass storage device: Any hardware device that provides persistent storage of data.

master boot record (MBR): Metadata such as the partition table, the disk signature, and the executable code for initiating the operating system boot process that is located on the first sector of a disk. Disks that have **MBRs** are referred to as **MBR** disks. **GUID partitioning table (GPT)** disks, instead, have unused dummy data in the first sector where the **MBR** would normally be.

master browser server: A server that is responsible for maintaining a master list of available resources on a subnet and for making the list available to **backup browser servers**. Each subnet requires a **master browser server**. The **master browser server** for a particular domain is called the **domain master browser server**.

master locator: A server that enables querying for server entries exported on a different machine.

master session key: A temporary cryptographic key that is used to derive other cryptographic keys to be used to encrypt and decrypt parts of a session-based protocol.

maximum transmission unit (MTU): The size, in bytes, of the largest packet that a given layer of a communications protocol can pass onward.

matrix data region: A report item on a report layout that displays data in a variable columnar format.

MD5 hash: A hashing algorithm, as specified in [RFC1321], that was developed by RSA Data Security, Inc. An **MD5 hash** is used by the **File Replication Service (FRS)** to verify that a file on each replica member is identical.

member (DFS-R): In the Distributed File System Replication Protocol, a computer participating in replication.

member server: A server that is joined to a domain and is not acting as an **Active Directory domain controller (DC)**.

merge disks or disk groups: The act of combining disks in two separate and distinct disk groups to form a single disk group.

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

Message Authentication Code (MAC): A message authenticator computed through the use of a symmetric key.

Message Authentication Code protocol data unit (MPDU): The unit of data exchanged between two peer **Message Authentication Code (MAC)** entities by using the services of the physical layer.

Message Authentication Code sublayer management entity (MLME): An entity that provides the layer management service interfaces through which layer management functions may be invoked.

message digest: See **hash function**.

message digest 4 (MD4): As specified in [RFC1320], a collision-resistant, non-rolling hash function that produces a 16-byte hash. While **MD4** is no longer considered to be cryptographically secure, **remote differential compression (RDC)** does not rely on cryptographic security in its hash function.

message identifier: An index into a message table. A message table is a collection of localizable strings. For Windows implementations, the message table is stored in the resource section of a dynamic link library.

message mode: A named pipe can be of two types: byte mode or **message mode**. In byte mode, the data sent or received on the named pipe does not have message boundaries but is treated as a continuous Stream. In message mode, message boundaries are enforced.

message server: A **remote procedure call (RPC)** server that implements this protocol.

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

Messaging Application Programming Interface (MAPI): A Windows programming interface that enables e-mail to be sent from within a Windows application.

metafile: 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.

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

Microsoft Management Console (MMC): The Microsoft Management Console (MMC) provides a framework that consists of a graphical user interface (GUI) and a programming platform in which snap-ins (collections of administrative tools) can be created, opened, and saved. MMC is a multiple-document interface (MDI) application.

mirrored volume: A fault-tolerant volume that maintains two or more copies of the volume's data. In the event that a disk is lost, at least one copy of the volume's data remains and can be accessed.

mixed mode: A state of an **Active Directory domain** that supports **domain controllers (DCs)** running Windows NT Server 4.0. **Mixed mode** does not allow organizations to take advantage of new **Active Directory** features such as universal groups, nested group membership, and interdomain group membership.

modification sequence number: An implementation-defined value for objects such as disks, volumes, drive letters, partitions, and regions that increases monotonically each time a configuration operation takes place on the object.

mount path: See **mounted folder**.

mount point: See **mounted folder**.

mount point access path: See **mounted folder**.

mounted folder: A file system directory that contains a linked path to a second volume. A user may link a path on one volume to another. For example, given two volumes C: and D:, a user can create a directory or folder C:\mountD and link that directory with volume D:. The path C:\MountD can then be used to access the root folder of volume D:.

MSZIP compression algorithm: The compression algorithm implementing RFC 1591 that is used between Windows 2000 **domain controllers (DCs)**. For more information, see [RFC1591].

multicast: A content delivery method in which a single stream is transmitted from a media server to multiple clients. The clients have no connection with the server. Instead, the server sends a single copy of the stream across the network to multicast-enabled routers, which replicate the data. Clients can then receive the stream by monitoring a specific multicast IP address and port.

multipartition volume: A **volume** containing data that exists on more than one partition.

multiplexed request: A request in which client **Server Message Block (SMB)** requests from various applications and users are all sent over the same **SMB** transport connection.

MULTI_SZ: A character buffer for holding null-terminated strings, as specified in [MS-DTYP] section 2.3.6.

mutual authentication: A mode in which each party verifies the identity of the other party, as specified in [RFC3748] section 7.2.1.

15 N

name service entry: A unit of advertisement that is exported to the RPC Name Service. These entries are of three types: a Server Entry, which contains bindings for a single server and optionally a set of Object UUIDs (for more information, see [C706-Ch2Intro], "Name Service Attributes"); a Group Entry, which contains names of one or more server entries, other groups, or both (for more information, see [C706-Ch2Intro], "Name Service Attributes"); and a Profile Entry, which contains a prioritized set of profile elements (for more information, see [C706-Ch2Intro], "Name Service Attributes").

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 stream: A place within a file in addition to the main stream where data is stored, or the data stored therein. File systems support a mode in which it is possible to open either the main stream of a file and/or to open a **named stream**. **Named streams** have different data than the main stream (and than each other) and may be read and written independently. Not all file systems support **named streams**. See also, **main stream**.

naming context (NC): A set of **objects** organized as a tree that is referenced by a DSName. The **distinguished name (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.

naming context (NC) replica: A tree of objects whose root object is identified by the **naming context**, which is a **dsname**.

naming context (NC) replica graph: A directed graph containing **naming context (NC)** replicas as nodes and repsFrom tuples as inbound edges by which originating updates replicate from each full replica of a given **NC** to all other **NC** replicas of the **NC**, directly or transitively.

naming context root (NC Root): The specific directory object referenced by the **naming context** dsname.

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

native mode: A state of an **Active Directory** domain in which all current and future **domain controllers (DCs)** run Windows 2000 Server, Windows Server 2003, Windows Server 2008, and Windows Server 2008 R2; no **DCs** run Windows NT Server 4.0. **Native mode** allows organizations to take advantage of the new **Active Directory** features such as universal groups, nested group membership, and interdomain group membership.

NBNS: See **NetBIOS Name Server (NBNS)**.

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

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

NDR64: See **64-bit Network Data Representation (NDR64)**.

negotiation: A series of exchanges. The successful outcome of a **negotiation** is the establishment of one or more **security associations (SAs)**. For more information, see [RFC2408] section 2.

negotiation discovery: An **Internet Key Exchange (IKE)** extension that improves interoperation between **Internet Protocol security (IPsec)** and non-IPsec-aware hosts. Detecting that the peer host is not capable of **IPsec** usually involves waiting for the **IKE** negotiation to time out, then sending traffic in the clear. With **negotiation discovery**, the host starts the **IKE** negotiation and sends clear text traffic in parallel. If the **IKE** negotiation succeeds and **security associations (SAs)** are established, further traffic is secured.

.NET Framework: An integral Windows component that supports building and running applications and **XML** Web services. The .NET Framework has two main components: the common language runtime and the .NET Framework class library. For more information about the .NET Framework, see [MSDN-.NET-FRAMEWORK].

The following versions of the .NET Framework are available in the following released Windows products or as supplemental software.

.NET Framework version	Windows version
.NET Framework 1.0	Windows 98, Windows Millennium Edition, Windows NT 4.0, Windows 2000, Windows XP, and Windows Server 2003
.NET Framework 1.1	Windows 98, Windows Millennium Edition, Windows 2000, Windows XP, Windows Vista, Windows Server 2003, Windows Server 2003 R2, and Windows Server 2008
.NET Framework 2.0	Windows 98, Windows Millennium Edition, Windows 2000, Windows XP, Windows Vista, Windows 7, Windows Server 2003, Windows Server 2003 R2, Windows Server 2008, and Windows Server 2008 R2
.NET Framework 3.0	Windows XP, Windows Vista, Windows 7, Windows Server 2003, Windows Server 2003 R2, Windows Server 2008, and Windows Server 2008 R2
.NET Framework 3.5	Windows XP, Windows Vista, Windows 7, Windows Server 2003, Windows Server 2003 R2, Windows Server 2008, and Windows Server 2008 R2
.NET Framework 4.0	Windows XP, Windows Vista, Windows 7, Windows Server 2003, Windows Server 2003 R2, Windows Server 2008, and Windows Server 2008 R2

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.

NetBIOS datagram service: An implementation of **NetBIOS** services in a datagram environment as specified in [RFC1001] section 17.

NetBIOS host name: The **NetBIOS name** of a host (as specified in [RFC1001] section 14 and [RFC1002] section 4), with the extensions described in [MS-NBTE].

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 NetBT-enabled hosts. A server running the Windows Internet Name Service (WINS) is the Microsoft implementation of an NBNS.

NetBIOS over TCP/IP (NetBT): A feature that allows **NetBIOS** to be used over the TCP/IP protocol, as defined in [RFC1001] and [RFC1002].

NetBIOS suffix: The 16th byte of a 16-byte **NetBIOS name** that is constructed using the optional naming convention defined in [MS-NBTE] section 1.8.

NetBT: See **NetBIOS over TCP/IP (NetBT)**.

Netlogon: (1) In a Windows NT-compatible network security environment, the component responsible for synchronization and maintenance functions between a **primary domain controller (PDC)** and **backup domain controllers (BDC)**. **Netlogon** is a precursor to the directory replication server (DRS) protocol.

(2) Used to refer to the Windows **Netlogon** **security support provider (SSP)**. This is not provided for use by other applications. It has neither the full functionality of public **SSPs** nor access from non-**Local Security Authority (LSA)** applications.

(3) The Netlogon Remote Protocol, as specified in [MS-NRPC].

Network Access Policy: A set of rules that determines the behavior of a **network access server (NAS)**. The policy consists of a set of conditions that matches an access request to the policy and an access profile.

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.

Network Access Protection (NAP) client: A computer that supports the **NAP** feature by complying with the corresponding policy settings.

Network Access Protection (NAP) Group Policy (GP) Extension GUID: A **GUID** defined separately for each computer policy setting that associates a specific administrative tool extension with a set of policy settings that can be stored in a **Group Policy object (GPO)**.

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 address translation (NAT): The process of converting between IP addresses used within an intranet, or other private network, and Internet IP addresses.

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-Ch14TransSyntaxNDR].

network logon: A software method in which the **account** information and credentials previously supplied by the user as part of an interactive logon are used again to log the user onto another network resource.

Network Policy Server (NPS): For Windows Server 2008, **NPS** replaces the Internet Authentication Service (IAS) in Windows Server 2003. **NPS** acts as a health policy server for the following technologies:

- Internet Protocol security (IPsec) for host-based authentication
- IEEE 802.1X authenticated network connections
- Virtual private networks (VPNs) for remote access
- Dynamic Host Configuration Protocol (DHCP)

network redirector: A software component on a connected computer that handles requests for remote files and printer operations.

node: 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.

nonce: A number that is used only once. This is typically implemented as a random number large enough that the probability of number reuse is extremely small. A **nonce** is used in authentication protocols to prevent replay attacks. For more information, see [RFC2617].

nonreplicated attribute: An attribute whose values are not replicated between **naming context (NC)** replicas. The nonreplicated attributes of an object are, in effect, local variables of the **domain controller (DC)** hosting the **NC** replica containing that object, since changes to these attributes have no effect outside that **DC**.

nonvolatile random access memory (NVRAM): Read/write memory that persists in its state when the power is removed, or normally volatile memory that has been fitted with a battery backup to retain data.

normal sync: The synchronization among replicas after initial sync is done.

notification area: An area of the desktop's **taskbar** containing program icons that provide status and notifications on events and system state, such as incoming e-mail messages, updates, and network connectivity.

notification icon: An icon placed in the notification area.

NT backup file: A file that contains the representation of another file. It is made up of zero or more backup streams.

NT Directory Service (NTDS): A previous name for **Active Directory**.

NTDS: See **NT Directory Service (NTDS)**.

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].

NTFS: See **NT file system (NTFS)**.

NT hash: An MD5-based cryptographic hash of a clear text password. For more information, see [MS-NLMP].

NT LAN Manager (NTLM) Authentication Protocol: A protocol using a challenge-response mechanism for authentication 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].

NTOWF: A general-purpose function used in the context of an NTLM authentication protocol, as specified in [MS-NLMP], which computes a one-way function of the user's password. For more information, see [MS-NLMP] section 6.

NULL GUID: A **GUID** of all zeros.

nullable column: A database table column that is allowed to contain no value for a given row.

16 O

object: (1) A set of attributes, each with its associated values. Two attributes of an object have special significance:

- Identifying attribute: A designated single-valued attribute 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.
- Parent-identifying attribute: A designated single-valued attribute 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)**.

(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].

object class: (1) A predicate defined on objects that constrains their attributes. 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**.

object class inheritance: The process of defining one **object class** in terms of its variations from an existing **object class**. The may-have, must-have, and possible superiors restrictions of an **object class** are all inherited.

object class name: The lDAPDisplayName of the **classSchema** object of an **object class**. The correspondence between Lightweight Directory Access Protocol (LDAP) display names and numeric **object identifiers (OIDs)** is specified in [MS-ADTS].

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 exporter identifier (OXID): A 64-bit number that uniquely identifies an **object exporter** within an object server.

objectGUID: (1) The attribute on an object whose value is a **GUID** that uniquely identifies the object. The value of **objectGUID** is assigned when an object is created and is immutable thereafter. The integrity of both object references between **naming contexts (NCs)** and of replication depends on the integrity of the **objectGUID** attribute.

(2) The **GUID** of an **Active Directory** object. For more information, see [MS-ADTS].

Object ID: See **ObjectID**.

ObjectID: A unique identifier that represents the identity of a file within a file system volume. For more information, see [MS-DLTM].

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. 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.

(3) 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.

(4) In the context of **Abstract Syntax Notation One (ASN.1)**, an object identifier, as specified 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].

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**.

object remote procedure call (ORPC): A remote procedure call whose target is an interface on an object. The target interface (and therefore the object) is identified by an **interface pointer identifier (IPID)**.

object resolver: A service in an object server that supports instantiating objects, obtaining remote procedure call (RPC) binding information for object exporters, and managing object lifetimes. **Object resolvers** may be reachable via well-known or dynamic RPC endpoints.

object server: An execution environment that contains a particular object resolver service and its associated object exporters.

object UUID: A **UUID** that is used to represent a resource available on the remote procedure call (RPC) servers. For more information, see [C706].

OBJREF: The **marshaled** form of an object reference.

OEM character: See **original equipment manufacturer (OEM) character**.

OEM character set: See **original equipment manufacturer (OEM) character set**.

OEM code page: See **original equipment manufacturer (OEM) code page**.

offline: 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.

OID: See **object identifier**.

OleTx: A comprehensive distributed transaction manager processing protocol.

one-way authentication: An authentication mode in which only one party verifies the identity of the other party.

one-way function (OWF): The calculation of a hash of the password using the Rivest-Shamir-Adleman (RSA) MD4 function. **OWF** is used to refer to the resulting value of the hash operation.

online: An operational state applicable to **volumes** and disks. In the online state, the volume or disk is available for data input/output (I/O) or configuration.

operating system upgrade: The action of replacing the existing operating system on a computer with a later version of the operating system while maintaining the original configuration and data of that computer.

operational attribute: An attribute that is returned only when requested by name in an **Lightweight Directory Access Protocol (LDAP)** search request. An **LDAP** search request requesting "all attributes" does not return operational attributes and their values.

oplock break: An unsolicited request sent by a **Server Message Block (SMB)** server to an **SMB** client to inform the client to change the **oplock** level for a file.

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].

opportunistic lock (oplock): A mechanism designed to allow clients to dynamically alter their buffering strategy in a consistent manner to increase performance and reduce network use. The network performance for remote file operations may be increased if a client can locally buffer file data, which reduces or eliminates the need to send and receive network packets. For example, a client may not have to write information into a file on a remote server if the client knows that no other process is accessing the data. Likewise, the client may buffer read-ahead data from the remote file if the client knows that no other process is writing data to the remote file.

There are three types of **oplocks**:

- Exclusive oplock allows a client to open a file for exclusive access and allows the client to perform arbitrary buffering.
- Batch oplock allows a client to keep a file open on the server even though the local accessor on the client machine has closed the file.
- Level II oplock indicates that there are multiple readers of a file and no writers. Level II Oplocks are supported if the negotiated SMB Dialect is NT LM 0.12 or later.

When a client opens a file, it requests the server to grant it a particular type of **oplock** on the file. The response from the server indicates the type of **oplock** granted to the client. The client uses the granted **oplock** type to adjust its buffering policy.

optical media drive: A drive that controls the positioning, reading, and writing of removable media on optical disks such as CD-ROMs and DVDs.

oriented tree: A directed acyclic graph such that for every vertex v, except one (the root), there is a unique edge whose tail is v. There is no edge whose tail is the root. For more information, see [KNUTH1] section 2.3.4.2.

original equipment manufacturer (OEM) character: An 8-bit encoding used in MS-DOS and Windows operating systems to associate a sequence of bits with specific characters. The **ASCII** character set maps the letters, numerals, and specified punctuation and control characters to the numbers from 0 to 127. The term "code page" is used to refer to extensions of the **ASCII** character set that map specified characters and symbols to the numbers from 128 to 255. These code pages are referred to as OEM character sets. For more information, see [MSCHARSET].

original equipment manufacturer (OEM) character set: A character encoding used where the mappings between characters is dependent upon the **code page** configured on the machine, typically by the manufacturer.

original equipment manufacturer (OEM) code page: A code page used to translate between non-Unicode encoded strings and UTF-16 encoded strings.

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.

originating write: An update operation that should be replicated to other replicas. The **originating write** changes the server state. The inputs of the operation are the DSNAME of

the object, the old value of replication metadata, and the list of modified attributes and values. The result of the operation is the new replication metadata stamped on the object.

originator GUID: A **GUID** that is associated with each replica member. All change orders produced by a given replica member carry the replica member's originator **GUID**, which is saved in the IDTable. The **originator GUID** is not the same as the member GUID, which is the objectGUID of the NT File Replication Service (NTFRS) member object in **Active Directory**. For more information, see [MS-ADTS].

ORPC extension: An out-of-band (not part of the explicit method signature), **GUID**-tagged binary large object (BLOB) of data that is sent or received in an object remote procedure call (ORPC) call.

OSF-DCE: The Distributed Computing Environment from the Open Software Foundation. It consists of multiple components, including **remote procedure call (RPC)**, that have been integrated to work closely together.

outbound: Network traffic flowing from the server to the client.

outbound connection: For a given replica member, a component of the NT File Replication Service (NTFRS) member object in **Active Directory** that identifies outbound partners. An **outbound connection** exists for each outbound partner.

outbound log (OutLog): A table in the **File Replication Service(FRS)** database that stores pending change orders to be sent to outbound partners. The changes can originate locally or come from an inbound partner. These change orders are eventually sent to all outbound replica partners.

outbound partner: The partner that receives change orders, files, and folders.

outbound trust: A relationship in which the primary domain trusts another domain to perform operations such as name lookups and authentication.

out-of-band policy application: A protocol exchange between a client and a server in which policy enforcement occurs for some subset of Group Policy settings from **Group Policy objects (GPOs)** encountered during some previous policy application exchange. This is referred to as "out-of-band" because, unlike policy application, an out-of-band policy application retrieves settings separately from **GPO** retrieval.

OXID resolution: The process of obtaining the remote procedure call (RPC) binding information that is required to communicate with the object exporter.

17 P

pack: See **disk group**.

PackageRegistration object: An **Active Directory** directory service container that represents a software installation extension setting. The container is an object of class **groupPolicyContainer**, as specified in [MS-ADSC] section 2.56).

packet marking: The act of filling out a special value, such as a **differentiated services code point (DSCP)** value, on individual packets, as specified in [RFC2474].

padding: Bytes that are inserted in a data stream to maintain alignment of the protocol requests on natural boundaries.

page description language (PDL): The language for describing the layout and contents of a printed page. Common examples are PostScript and **Printer Control Language (PCL)**.

page file or paging file: A file that is used by operating systems for managing virtual memory.

parent GUID: The **GUID** of the parent folder that contains a particular file or folder in the replica tree.

parent object: An object is either the root of a tree of objects or has a parent. If two objects have the same parent, they must have different values in their **relative distinguished names (RDNs)**. See also, **object**.

partial attribute set (PAS): The subset of attributes that replicate to partial **naming context (NC)** replicas. Also, the particular partial attribute set that is part of the state of a forest and that is used to control the attributes that replicate to **global catalog (GC)** servers.

partial database synchronization: A mechanism for synchronizing a set of database records on a particular replication partner.

partial replica: A **naming context (NC)** replica that contains a schema-specified subset of attributes for the objects it contains. A partial replica is not writable as it does not accept originating updates.

partition: (1) In the context of hard disks, a logical region of a hard disk. A hard disk may be subdivided into one or more **partitions**.

(2) In the context of **directory services**, a synonym for **directory partition** and **naming context (NC) replica**.

partition table: An area of a disk that is used to store metadata information about the **partitions** on the disk. See also, **GUID partitioning table (GPT)**.

partition type: A value indicating the **partition's** intended use, or indicating the type of file system on the **partition**. For example, **partition** type 0x07 indicates that the **partition** is formatted with the NTFS file system. Original equipment manufacturers may designate a **partition** type of 0x12 to indicate that manufacturer-specific data is stored on the **partition**.

partner: A computer connected to a local computer through either inbound or outbound connections.

password policy: A set of rules that is designed to enhance computer security by encouraging users to employ strong passwords and use them properly.

path: When referring to a file path on a file system, a hierarchical sequence of folders. When referring to a connection to a storage device, a connection through which a machine can communicate with the storage device.

paused: A service that is not available because it has been placed in a suspended state, usually as a result of explicit administrative action.

PDC: See **primary domain controller**.

PDU: See **protocol data unit**.

PDU stream: An ordered sequence of **RPC** and RPC over HTTP **protocol data units**.

peak rate: A value in a **TSpec** that is used to specify an aspect of network traffic behavior, as specified in [RFC2212].

peer: (1) The entity being authenticated by the authenticator.

(2) In **DirectPlay**, a peer refers to 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.

peer-to-peer mode: A game-playing mode that consists of multiple peers. Each peer has a connection to all other peers in the DirectPlay game session. If there are N peers in the game session, each peer has N–1 connections.

perfect forward secrecy (PFS): A property of key exchange protocols, which holds when session keys from previous communications are not compromised by the disclosure of longer-term keying material. In the context of **Internet Protocol security (IPsec)**, **PFS** requires a Diffie-Hellman exchange to generate the keys for each **quick mode (QM) security association (SA)**.

permission X on object Y: An access check where the access type is X and the security descriptor is read from object Y's Lightweight Directory Access Protocol (LDAP) attribute securityDescriptor.

phase: A series of exchanges that provide a particular set of security services (for example, authentication or creation of **security associations (SAs)**).

phase I authentication set: A collection of settings that specifies how **Internet Protocol security (IPsec)** performs phase I (or main mode) authentication.

phase I cryptographic set: A collection of settings that specifies how **Internet Protocol security (IPsec)** performs phase I (or main mode) key exchange.

phase I cryptographic suite: One or more phase I cryptographic suites are associated with a phase I cryptographic set. Each phase I cryptographic suite contains a Diffie-Hellman algorithm, an encryption algorithm, and an integrity algorithm.

phase II authentication method: One or more phase I authentication methods are associated with each phase I authentication set. Each phase I authentication method specifies an authentication credential and in some cases additional information about how the authentication credential is used.

phase II authentication set: A collection of settings that specifies how **Internet Protocol security (IPsec)** performs **AuthIP** extended mode authentication.

phase II cryptographic set: A collection of settings that specifies how **Internet Protocol security (IPsec)** performs phase II (or quick mode) data protection.

phase II cryptographic suite: One or more phase II cryptographic suites are associated with each phase II cryptographic set. Each phase II cryptographic suite contains a protocol specifying how the packet is modified by **Internet Protocol security (IPsec)**, an encryption algorithm, an integrity algorithm, and information about how frequently to regenerate the keys used to protect the data.

Phase One: The initial phase of a two-phase commit sequence. During this phase, the participants in the transaction are requested to prepare to be committed. This phase is also known as the "Prepare" phase. At the end of Phase One, the outcome of the transaction is known.

Phase Two: The second phase of a two-phase commit sequence. This phase occurs after the decision to commit or abort is determined. During this phase, the participants in the transaction are ordered to either commit or rollback.

Phase Zero: A phase in distributed transaction processing that is composed of one or more **Phase Zero waves**. At the beginning of a **Phase Zero wave**, all Phase Zero participants are notified that the transaction has entered Phase Zero. While the participants process the Phase Zero notification, they can continue to marshal the transaction to new participants. Consequently, participating transaction managers can still accept new enlistments during Phase Zero.

Phase Zero enlistment: An enlistment that indicates that the subordinate participant participates in Phase Zero.

Phase Zero participant: A participant with a Phase Zero enlistment.

Phase Zero wave: A discrete stage inside Phase Zero processing in which Phase Zero notifications are sent to all known Phase Zero enlistments. New Phase Zero enlistments that appear during a Phase Zero wave are processed during the next Phase Zero wave. The process is repeated until a Phase Zero wave is processed without the creation of new Phase Zero enlistments.

ping: In the Domain Controller (DC) Locator Protocol, a client sends a ping request to a **DC** to determine its responsiveness. When a client is actively soliciting the attention of a **DC**, it is said to be pinging the **DC**.

ping set: A set of DCOM objects on a particular object server in use by a particular client. The set is grouped in order to maintain the lifetimes of object references collectively for the set rather than individually for each object.

ping set identifier (SETID): A 64-bit number that uniquely identifies a ping set within an object server.

pinging: The process by which a client periodically contacts an object server to maintain the lifetime of its references to objects on that object server.

pipe instance: A request to open a **named pipe** by a client application. Multiple Server Message Block (SMB) clients can open the same **named pipe**. Each request to open the same **named pipe** is a **pipe instance**.

pipe state: A series of attributes that describe how the pipe interacts with processes for various input/output (I/O) operations and that indicate how much data is currently available to be read from the **named pipe**.

plaintext: In cryptography, ordinary readable text before it is encrypted into **ciphertext**, or after it has been decrypted.

player: Represents a person that is playing a computer game. There may be multiple players on a computer participating in any given game session. See also, **name table**.

plex: See **volume plex**.

policies path: A domain-based **Distributed File System (DFS)** path for a directory on the server that is accessible through the **Server Message Block (SMB)** protocol. This path must be of the form \\<dns domain name>\sysvol\<dns domain name>\policies.

policy: (1) The set of rules that govern the interaction between a subject and an object or resource.

(2) 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.

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.

policy setting: A statement of the possible behaviors of an element of a domain member computer's behavior that can be configured by an administrator.

policy target: A user or computer account for which policy settings can be obtained from a server in the same domain, as specified in [MS-GPOL]. For user policy mode, the policy target is a user account. For computer policy mode, the policy target is a computer account.

PostScript: A **page description language** developed by Adobe Systems that is primarily used for printing documents on laser printers. It is the standard for desktop publishing.

preauthentication: In Kerberos, preauthentication allows a **key distribution center (KDC)** to demand that the requestor in the Authentication Service (AS) Exchange demonstrate knowledge of the key associated with the account before the **KDC** will issue a ticket-granting ticket (TGT) as specified in [RFC4120] sections 5.2.7 and 7.5.2.

PREDEFINED_KEY: Root keys that can be referenced by using well-known names and conforms to the tree structure.

prefix table: A data structure that is used to translate between an **object identifier (OID)** and a compressed representation for **OIDs**.

primary disk group: In the context of **dynamic disk**, it is the disk group whose disks are online, which means they are accessible for input/output (I/O) and configuration. Each machine may have only one primary disk group. Disks on the machine belonging to other disk groups are referred to as "foreign disks" and their disk group is referred to as a "foreign disk group".

primary domain: A **domain** (identified by a **security identifier (SID)**) that the server is joined to. For a **domain controller (DC)**, the **primary domain** is that of the **domain** itself.

primary domain controller (PDC): A master **domain controller (DC)** that performs authentication on access requests from workstations and other servers, and that manages information concerning network security and resources.

primary domain controller (PDC) role owner: The **domain controller (DC)** that hosts the **primary domain controller** emulator FSMO role for a given domain **naming context (NC)**.

primary language identifier: The lower 10 bits of a language identifier. It identifies the user interface human language supported by an application or client computer without regard to variations such as dialect.

primary partition: A type of partition on a **master boot record (MBR)**-formatted disk.

principal: (1) An authenticated entity that initiates a message or channel in a distributed system.

(2) An ID of such an entity.

(3) In Kerberos, a Kerberos principal.

principal name: The computer or user name that is maintained and authenticated by the **Active Directory** directory service.

principal self: A **well-known security identifier (SID)** used to represent the identity of a security principal when that security principal is also the object that is being protected with a security descriptor. Applicable only to directory objects that are representing security principals, the principal self identifier allows the security descriptor on the directory object to grant specific user rights to the principal itself. As an example, a user object for fred@domain.com might have a security descriptor that allowed principal-self:update-shoe-size. The intent is to allow fred to update his own shoe size. The use of the fixed value SID for principal self prevents every user object from needing a unique security descriptor, thus conserving space in the directory database.

principal's secret key: In Kerberos, a symmetric encryption key shared between an entity and the **key distribution center (KDC)**, with a long lifetime and for the purpose of authentication. A password is a common example of a **principal's secret key**.

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.

printer driver: The interface component between the operating system and the printer device. It is responsible for processing the application data into a **page description language (PDL)** that can be interpreted by the printer device.

print job: The rendered **page description language (PDL)** output data sent to a print device for a particular application or user request.

print queue: The logical entity to which jobs may be submitted for a particular print device. Associated with a print queue is a print driver, a user's print configuration in the form of a DEVMODE structure, and a system print configuration stored in the system registry.

print server: A machine that hosts the print system and all its different components.

print system: A system component that is responsible for coordinating and controlling the operation of print queues, print drivers, and print jobs.

Printer Control Language (PCL): A **page description language (PDL)** developed by Hewlett Packard for its laser and ink-jet printers.

printer form: A preprinted blank paper form, or a print job's virtual representation of this form, that enables a printer to position form elements in their physical location on the page.

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: (1) The right of a user to perform system-related operations, such as debugging the system. A user's authorization context specifies what privileges are held by that user.

(2) The capability of a security principal to perform a type of operation on a computer system regardless of restrictions placed by discretionary access control.

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.

process identifier (PID): A number used by some operating systems (for example, Windows and UNIX) to uniquely identify a process. For more information, see [PROCESS].

product identifier GUID: A unique identifier in the form of a **GUID** for the application described by a software installation package. Two such packages with the same product identifier **GUID** describe the same application.

profile: A grouping of settings that is applied based on the network location of connected interfaces on the client computer. Three profiles are supported by Windows Firewall with Advanced Security: domain (used when connected to a corporate environment, private (used when connected to a home or small business behind a gateway device), and public (used when connected to a public hotspot such as a coffee shop or airport).

profile element: A record that corresponds to a single **remote procedure call (RPC)** interface and that refers to a server entry, group, or profile. For more information, see [C706-Ch2Intro], "Name Service Attributes".

property: A data field within a **Common Information Model (CIM)** class definition. This consists of a simple name, a type, and a value.

property set: A set of attributes, identified by a **GUID**. Granting access to a property set grants access to all the attributes in the set.

protected attribute: A sensitive protected attribute that is not readable outside the **Local Security Authority (LSA)** running on a **domain controller (DC)**.

protected subsystem: The part of a system that is isolated from the rest of the system such that it cannot be affected by the non-protected parts of the system.

protocol: A set of rules governing the exchange or transmission of data between devices to accomplish a specific task or group of tasks.

protocol data unit (PDU): Information that is delivered as a unit among peer entities of a network and that may contain control information, address information, or data. For more information on remote procedure call (RPC)-specific PDUs, see [C706-Ch12RPC_PDU_Encode].

protocol dialect: A protocol version that is distinct and non-interoperable from other protocol versions from the same group of related protocols.

protocol extension: An addition of new integrated behavior to an existing **protocol**.

protocol identifier: A numeric value that uniquely identifies an RPC transport protocol when describing a protocol in the context of a protocol tower. For more information, see [C706-AppendixIProtocolID].

protocol role: A class of protocol functionality that is identified as such for the purposes of a specification.

protocol sequence identifier: A numeric value that uniquely identifies an RPC transport protocol when describing a protocol in the context of a protocol tower. For more details, see [C706-AppendixIProtocolID].

protocol state: Information stored by a protocol that affects its behavior.

protocol tower: A protocol sequence along with its related address and protocol-specific information. For more information, see [C706-Ch6RPCCallModel].

protocol type: A special set of standardized rules that the endpoints in a communications connection use when transferring data.

prototype context: A context that is sent as part of an activation request.

proxy: A network node that accepts network traffic originating from one network agent and transmits it to another network agent.

pseudo-random number generator (PRNG): An algorithm that generates values (numbers, bits, and so on) that give the appearance of being random from the point of view of any known test. If initialized with a true random value (called its "seed"), the output of a cryptographically strong PRNG will have the same resistance to guessing as a true random source.

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 key algorithm: An asymmetric cipher that uses two cryptographic keys: one for encryption, the public key, and the other for decryption, the private key. In signature and verification, the roles are reversed: public key is used for verification, and private key is used for signature generation. Examples of public key algorithms are described in various standards, including Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA) in FIPS 186-2 (as specified in [FIPS186]), RSA in PKCS#1 (as specified in [PKCS1]), the National Institute of Standards and Technology (NIST) also published an introduction to public key technology in SP800-32 (as specified in [SP800-32]).

Public Key Cryptography Standards (PKCS): A group of Public Key Cryptography Standards published by RSA Laboratories.

public key infrastructure (PKI): The laws, policies, standards, and software that regulate or manipulate certificates and public and private keys. In practice, it is a system of digital certificates, certification authorities (CAs), and other registration authorities that verify and authenticate the validity of each party involved in an electronic transaction. For more information, see [X509] section 6.

public-private key pair: The association of a public key and its corresponding private key when used in cryptography. For an introduction to public-private key pairs, see [IEEE1363] section 3.

published application: An application that should not automatically be installed at computer startup or user logon unless it is a required upgrade of an application that is installed on the computer. However, software maintenance applications on the computer can display information about this software and install or uninstall it, often at the direction of a user.

18 Q

qualifier: A metadata item as specified in [DMTF-DSP0004] section 4.5.4. This consists of a simple name, a type, a value, and a flavor (a propagation rule for the qualifier).

Quality of Service (QoS): A set of technologies that do network traffic manipulation, such as packet marking and reshaping.

quick format: A formatting that does not zero the data sectors on the volume at the time the file system metadata is created.

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.

quick mode security association (QM SA): A **security association (SA)** that is used to protect IP packets between peers (the **Internet Key Exchange (IKE)** traffic is protected by the **main mode security association (MM SA)**). For more information, see [RFC2409] section 5.5.

19 R

RADIUS attribute: An abstract identifier for a value or set of values that describe elements of a RADIUS protocol exchange. RADIUS attributes describe the details of an endpoint's connection request and provides configuration data for a **network access server (NAS)** to provide service to the endpoint.

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.

RAID-0: A **RAID** volume that stripes its data across multiple **RAID columns**. Also called a striped volume.

RAID-1: See **mirrored volume**.

RAID-5: A fault-tolerant volume that maintains the volume's data across multiple **RAID columns**. Fault tolerance is provided by writing parity data for each stripe. In the event that one disk encounters a fault, that disk's data may be reconstructed using the parity data located on the other disks.

RAID column: A **RAID** construct for organizing disks and volumes.

raw read (on a named pipe): The act of reading data from a **named pipe** that ignores message boundaries even if the pipe was set up as a **message mode** pipe.

raw write (on a named pipe): The act of writing data into a **named pipe** where the data must contain the message boundaries if the pipe is a **message mode** pipe. The operation can allow a single write to insert multiple messages.

RC4: A variable key-length **symmetric encryption** algorithm. For more information, see [SCHNEIER] section 17.1.

RDN: See **relative distinguished name (RDN)**.

RDN attribute: The attribute used in a **relative distinguished name (RDN)**. In the **RDN** "cn=Peter Houston" the **RDN attribute** is **cn**. In the **Active Directory** directory service, the **RDN attribute** of an **object** is determined by the **88 object class** or the most specific **structural object class** of the **object**.

read-only: An attribute of storage media that denotes that the media is not available to be written.

read-only domain controller (RODC): A **domain controller (DC)** that does not accept originating updates.

read-only replicated folders: A folder where local changes are not replicated out and reverted by replicating back previous content.

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 specified in [RFC4120] 1.2.

receive window: The amount of memory that a recipient of network traffic has committed to queuing **protocol data units (PDUs)** that it cannot process immediately.

recovery: The process of reestablishing connectivity and synchronizing views on the outcome of transactions between two participants after a transient failure. Recovery occurs either between a resource manager and a transaction manager, or between a Superior Transaction Manager Facet and a Subordinate Transaction Manager Facet.

redeploy action: An action that an administrator may take for an application deployed through the software installation extension protocol that will cause all clients that receive the application through the protocol to perform an installation of the application on the client if the application is already installed. This is used by administrators as a mechanism to update the application.

redundant arrays of independent disks (RAID): A set of disk-organization techniques that is designed to achieve high-performance storage access and availability.

reference count: An integer value that is used to keep track of a Component Object Model (COM) object. When an object is created, its reference count is set to 1. Every time an interface is bound to the object, its reference count is incremented; when the interface connection is destroyed, the reference count is decremented. The object is destroyed when the reference count reaches zero. All interfaces to that object are then invalid.

RefreshTime: The last time that information for an entry in the VolumeTable or FileTable has been refreshed by its VolumeOwner.

region: See **disk extent**.

region flags: A set of values that describes the region's state or use.

region's status: The status of the region, such as whether the region is performing properly or encountering disk faults.

registration: See **certification**.

registration authority (RA): (1) A generic term for a software module, hardware component, or human operator thereof that enables a user or **public key infrastructure (PKI)** administrator to perform various administration and operational functions as part of the certification or revocation process.

(2) The authority in a **PKI** that verifies user requests for a digital certificate and indicates to the **certificate authority (CA)** that it is acceptable to issue a **certificate**.

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.

registry files: The physical representation of a logical tree in the registry.

registry policy file: A file associated with a **Group Policy object (GPO)** that contains a set of registry-based policy settings.

REGSAM: A bit field that specifies the user rights for a key object.

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) of the parent forms the FQDN of the child.

relative identifier (RID): The last item in the series of subauthority values in a SID (as specified in [SIDS]). 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.

release: The process of calling the third IUnknown method (IUnknown::Release()) on an object.

reliable time source: A time source that can provide accurate time. It is usually the primary reference with stratum 1 as specified in [RFC1305]; for example, a radio clock.

relying party (RP): 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.

remediation server: A server that is responsible for bringing a noncompliant computer back into a compliant state.

remote application: An application running on a remote server.

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 Access Service (RAS) server: A type of **network access server (NAS)** that provides modem dial-up or virtual private network (VPN) access to a network.

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 change order: A change order that is received from an inbound (or upstream) partner that originated elsewhere in the replica set.

Remote Desktop Protocol (RDP): The protocol used to implement remote connections (**Terminal Services**) on Windows operating systems. For more information, see [MSDN-RDP].

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 differential compression (RDC) FilterMax algorithm: The algorithm that **RDC** uses to determine the **cut points** in a file. The FilterMax algorithm has the property that it will often find **cut points** that result in identical chunks being found in differing files, even when the files differ by insertions and deletions of bytes, not simply by length-preserving byte modifications. For more information, see [MS-RDC] section 3.1.5.1.

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 procedure call (RPC) name service: A service that allows servers to export binding information, and clients to find it, in an efficient manner. For more information, see [C706-Ch2Intro], "Name Service Interface".

remote server name: A null-terminated Unicode string, supplied by an application, which in conjunction with an RPC protocol sequence is used to initiate communication with an object server.

remote unknown: An object exporter's remotely accessible implementation of the IUnknown interface. Each object exporter has exactly one such remotely accessible IUnknown implementation, which is responsible for handling all IUnknown invocations from clients.

removable media: Any type of storage that is not permanently attached to the computer. A persistent storage device stores its data on media. If the media can be removed from the device, the media is considered removable. For example, a floppy disk drive uses removable media.

reparse point: A collection of user-defined data associated with a file or directory. The format of this data is understood by the application or the file system that stores the data, and the file system filter that interprets the data and processes the file. Reparse points can contain data that instructs the file system or the operating system to take special actions. For more information, see [MS-FSCC].

replica: A variable containing a set of objects.

replica member (File Replication Service (FRS) replica): A member of a **replica set**. The replica contains machine-specific information.

replica set: (1) The representation of the replication group on a single computer. It is the slice of the replication group that affects the server that it exists on. For instance, it contains only the connections for which this computer is either the client or server.

(2) In File Replication Service (FRS), the replication of files and directories according to a predefined topology and schedule on a specific folder. The topology and schedule are collectively called a replica set. A replica set contains a set of replicas, one for each machine that participates in replication.

replica tree: The local replica root folder together with all files and directories underneath it, which usually is saved as a tree structure in the file system.

ReplicaSetId: The GUID that is assigned to a specific replication group.

replicated attribute: An attribute whose values are replicated to other naming context (NC) replicas. An attribute is replicated if its attributeSchema object o does not have a value for the systemFlags attribute or if bit 0 of the value is clear.

replicated folder: The root of a replicated tree. All files and subfolders (recursively) are replicated.

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: 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.

replication epoch: A state variable of a **domain controller (DC)** that changes when a **DC** is no longer compatible for replication with its former partners. A server receiving a replication request tests the client's replication epoch against its own and refuses the request if the two are not equal.

replication group: A container for a set of replicated folders sharing the same connections to replication partners.

replication latency: The time lag between a final originating update to a **naming context (NC)** replica and all **NC** replicas reaching a common application-visible state.

replication session: The state that is maintained when replicating files in the context of a replicated folder and connection.

replication traffic: Network traffic that is performed to accomplish replication.

replication transport: The transport (wire protocol) used by **Active Directory domain controllers** to perform **replication**. **Active Directory** supports **remote procedure call (RPC)** and **Simple Mail Transfer Protocol (SMTP)** transports.

report definition: The blueprint for a report before the report is processed or rendered. A report definition contains information about the query and layout for the report.

RequestMachine: The MachineID of the computer that is the client of the Distributed Link Tracking (DLT) Central Manager RPC protocol.

requestor: The computer that sends the request messages that are defined by this protocol.

reshaping: An act of buffering data until it can be sent in conformance to a TSpec, as specified in [RFC2212].

reshaping value: A value that is used for both the peak rate and the bucket rate in a TSpec to be used in reshaping.

resource: 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.

resource group: 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.

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.

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.

response key: A key essentially derived from a one-way hash of the password. It can be calculated slightly differently based on which NTLM version is being used. It is then used to derive the key exchange key.

retry change order: A change order that is in some state of completion but was blocked for some reason and must be retried later.

revocation: The process of invalidating a certificate. For more details, see [RFC3280] section 3.3.

Rivest-Shamir-Adleman (RSA): A system for **public key** cryptography. **RSA** is specified in [RFC2437].

role change: The act of changing the role of a computer. The act of configuring a server to be a domain controller (DC) is called "promotion". The act of configuring a DC to be a non-DC server is called "demotion".

role: The domain role quantifies the relationship between a computer and a domain. Domain roles include the following:

Joined: Linked to a domain for purposes of policy and security.

Standalone: Not associated with any domain.

Domain controller: Linked to a domain, and hosting that domain

role separation: The concept of using a **certificate authority (CA)** to enhance security by allowing a user to be assigned a single role such as auditor, backup manager, administrator, or certificate manager. Role separation ensures that a user may not posses multiple roles at one time. Role separation is a common criteria requirement for the Certificate Issuing and Management Components (CIMC) protection profile. For more information, see [CIMC-PP]. Not all CAs support role separation.

rolling hash function: A hash function that can be computed incrementally over a set of data. Given an arbitrary integer $n \geq 0$, some bytes $b_0 .. b_{n-1}$ and their hash $h(b_0 .. b_{n-1})$, a hash function h is a rolling hash function if one can compute $h(b_1 .. b_n)$ in time that does not depend on n.

root CA: (1) A type of **certificate authority (CA)** that is directly trusted by an end entity; that is, securely acquiring the value of a root CA public key requires some out-of-band steps. This term is not meant to imply that a root CA is necessarily at the top of any hierarchy, simply that the CA in question is trusted directly (as specified in [RFC2510]). A root CA is

implemented in software and in Windows, is the topmost CA in a CA hierarchy, and is the trust point for all certificates that are issued by the CAs in the CA hierarchy. If a user, computer, or service trusts a root CA, it implicitly trusts all certificates that are issued by all other CAs in the CA hierarchy. For more information, see [RFC3280].

(2) Any **certificate authority (CA)** that is directly trusted by a relying party.

root certificate: A **self-signed certificate** that identifies the public key of a root **certificate authority (CA)** and has been trusted to terminate a certificate chain.

root domain: (1) The **domain** that is created first in a **forest**.

(2) In Active Directory, the unique **domain naming contexts (domain NCs)** of an Active Directory **forest** that is the parent of the **forest's config NC**. The **config NC's relative distinguished name (RDN)** is "cn=Configuration" relative to this parent.

root directory system agent-specific entry (rootDSE): The logical root of a **directory** server, whose **distinguished name (DN)** is the empty string. In the **Lightweight Directory Access Protocol (LDAP)**, the **rootDSE** is a nameless entry (a **DN** with an empty string) containing the configuration status of the server. Access to this entry is typically available to unauthenticated clients. The **rootDSE** contains **attributes** that represent the features, capabilities, and extensions provided by the particular server.

root error: The last error in an error sequence.

rootDSE: See **root directory system agent-specific entry (rootDSE)**.

round-trip time (RTT): The time that it takes a packet to be sent to a remote partner and for that partner's acknowledgment to arrive at the original sender. This is a measurement of latency between partners.

RPC client: A computer on the network that sends messages using remote procedure call (RPC) as its transport, waits for responses, and is the initiator in an RPC exchange.

RPC context handle: A representation of state maintained between a remote procedure call (RPC) client and server. The state is maintained on the server on behalf of the client. An RPC context handle is created by the server and given to the client. The client passes the RPC context handle back to the server in method calls to assist in identifying the state. For more information, see [C706].

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

RPC endpoint: A network-specific address of a server process for remote procedure calls (RPCs). The actual name of the RPC endpoint depends on the RPC protocol sequence being used. For example, for the NCACN_IP_TCP RPC protocol sequence an RPC endpoint might be TCP port 1025. For more information, see [C706].

RPC engine: The runtime environment that is providing remote procedure call (RPC) facilities.

RPC PDU: A protocol data unit (PDU) originating in the remote procedure call (RPC) runtime. For more information on RPC PDUs, see [C706-Ch12RPC_PDU_Encode] and [MS-RPCE] section 2.

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. For more information, see [C706] and [MS-RPCE].

RPC server: A computer on the network that waits for messages, processes them when they arrive, and sends responses using RPC as its transport acts as the responder during a remote procedure call (RPC) exchange.

RPC session key: See **session key**.

RPC transfer syntax: A method for encoding messages defined in an Interface Definition Language (IDL) file. Remote procedure call (RPC) can support different encoding methods or transfer syntaxes. For more information, see [C706].

RPC transport: The underlying network services used by the remote procedure call (RPC) runtime for communications between network nodes. For more information, see [C706-Ch2Intro].

RTT: See **round-trip time (RTT)**.

20 S

SA: See **security association (SA)**.

SAD: See **security association database (SAD)**.

salt: An additional random quantity, specified as input to an **encryption** function, that is used to increase the strength of the **encryption**.

sanitized name: The form of a **certification authority (CA)** name that is used in file names (such as for a **certificate revocation list (CRL)**; see [MSFT-CRL] for more information) and in other contexts where character sets are restricted. The process of sanitizing the **CA** name is necessary to remove characters that are illegal for file names, registry key names, or **distinguished name (DN)** values, or that are illegal for technology-specific reasons.

SASL: See **Simple Authentication and Security Layer (SASL)**.

schedule: The frequency at which data replicates.

schema: The set of **attributes** and **object classes** that govern the creation and update of **objects**.

schema container: The root **object** of the **schema naming context (schema NC)**.

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**.

schema object: An **object** that defines an **attribute** or an **object class**. **Schema objects** are contained in the **schema naming context (schema NC)**.

scope of management (SOM): An **Active Directory site**, **domain**, or organizational unit container. These containers contain user and computer accounts that can be managed through **Group Policy**. These **SOMs** are themselves associated with **Group Policy objects (GPOs)**, and the accounts within them are considered by the Group Policy Protocol [MS-GPOL] to inherit that association.

scoped Group Policy object (GPO) distinguished name (DN): A **Group Policy object (GPO) distinguished name (DN)** where the set of "CN=<cn>" elements is prepended with "CN=User" for the **user policy mode** of policy application and with "CN=Machine" for **computer policy mode**.

scoped Group Policy object (GPO) path: A **Group Policy object (GPO)** path appended with "\User" for the **user policy mode** of policy application, and "\Machine" for the **computer policy mode**.

screen coordinates: Coordinates relative to the top-left corner of the screen, which has the coordinates (0,0).

SCSI: See **small computer system interface (SCSI)**.

SCSI logical unit number (LUN): See **logical unit number (LUN)**.

SCSI port number: A number that uniquely identifies a port on a **small computer system interface (SCSI)** disk controller. Each **SCSI** disk controller may support multiple **SCSI** bus attachments or ports for connecting **SCSI** devices to a computer.

SCSI protocol: An architecture for **SCSI**, consisting of a group of standards created and maintained by the Technical Committee (T10) of the InterNational Committee on Information Technology Standards (INCITS).

SD: See **security descriptor**.

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**.

secret object: An element of the **Local Security Authority (LSA)** Policy Database, which contains a value that is secret in that access to it is strictly controlled through cryptographic protections and restrictive access control mechanisms.

sector: The smallest addressable unit of a disk.

secure channel: An authenticated **remote procedure call (RPC)** connection between two machines in a **domain** with an established **security context** used for signing and encrypting **RPC** packets.

secure desktop: Only trusted processes running as SYSTEM are allowed to run on the **secure desktop**.

Secure/Multipurpose Internet Mail Extensions (S/MIME): A standard for encrypted and digitally signed electronic mail that allows users to send encrypted messages and authenticate received messages.

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** using X.509 **certificates** (for more information, see [X509]). The **SSL** protocol is precursor to **Transport Layer Security (TLS)**. The **TLS** version 1.0 specification is based on **SSL** version 3.0.

security account manager (SAM) built-in database: Microsoft-specific terminology for the part of the user account database that contains account information (such as account names and passwords) for accounts and groups that are pre-created at the database installation.

security association (SA): A simplex "connection" that provides security services to the traffic carried by it. See [RFC4301] for more information.

security association database (SAD): A database that contains parameters that are associated with each established (keyed) **security association**.

security context: An abstract data structure that contains authorization information for a particular **security principal** in the form of a Token/Authorization Context (see [MS-DTYP] section 2.5.2). A server uses the authorization information in a **security context** to check access to requested resources. A **security context** also contains a **key** identifier that associates mutually established cryptographic **keys**, along with other information needed to perform secure communication with another security principal.

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.

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-WSO] section 3.1.2.1.3.

security policy: In the form of a collection of **security policy** settings, the policy itself is an expression of administrative intent regarding how computers and resources on a network should be secured.

security policy database (SPD): A database that specifies the policies that determine the disposition of all IP traffic inbound or outbound from a host or security gateway.

security policy settings: Contained in **security policies**, the policy settings are the actual expression of how various security-related parameters on the computer are to be configured.

security principal: (1) 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".

(2) An identity that can be used to regulate access to resources, as specified in [MS-WSO]. A **security principal** can be a user, a computer, or a group that represents a set of users.

security principal name (SPN): The name that identifies a **security principal** (for example, machinename$@domainname for a machine joined to a **domain** or username@domainname for a user). Domainname is resolved using the **Domain Name System (DNS)**.

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**).

security protocol: A protocol that performs **authentication** and possibly additional security services on a network.

security provider: 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: 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**.

seed file or seed data: A file or files at the target location that are used to supply data for reconstructing the **source file**. **Remote differential compression (RDC)** may use an arbitrary number of **seed files** in the process of copying a single **source file**. The process of selecting **seed files** can be guided by using similarity traits. For more information, see [MS-RDC] section 3.1.5.4.2.

selective single master: A **replication** mode in which changes from only a single machine propagate to other machines.

self-signed certificate: A **certificate** that is signed by its creator and verified using the **public key** contained in it. Such **certificates** are also termed **root certificates**.

semisynchronous operation: An operation that is executed on the server side while the client is regularly checking to see if there is no response available from the server.

sequence ID: A monotonically increasing 8-bit identifier for packets. This is typically represented as a field named **bSeq** in packet structures.

serial storage architecture (SSA) bus: Serial storage architecture (SSA) is a standard for high-speed access to high-capacity disk storage. An **SSA bus** is implemented to the SSA standard.

serialize: The process of taking an in-memory data structure, flat or otherwise, and turning it into a flat stream of bytes. See also **marshal**.

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**.

server-activated object (SAO): A **server object** that is created on demand in response to a client request. See also **marshaled server object**.

server authentication: A mode of **authentication** in which only the server in the transaction proves its identity.

server challenge: A 64-bit nonce generated on the server side.

server Group Policy object (GPO) distinguished name (DN): A **Group Policy object (GPO) distinguished name (DN)** that uses a specific server in the **Lightweight Directory**

Access Protocol (LDAP) path syntax, as specified in [RFC2251], where the server name is a **domain controller (DC)** that is located as specified in [MS-NRPC] section 3.5.5.3.2.

server Group Policy object (GPO) path: A **Group Policy object (GPO) path** in which the **Distributed File System (DFS)** path contains a server name in the **DFS** path syntax and where the server name is a **domain controller (DC)**.

server locator: Enables exporting of entries to the **remote procedure call (RPC)** name service.

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 [MS-CIFS] and [MS-SMB].

Note Whenever **SMB** is indicated, SMB2 can also be included (unless otherwise stated).

server object: A class of **object** in the **config NC**. A **server object** can have an nTDSDSA **object** as a child.

server role: The state of a **domain controller (DC)**, which can be one of two values--**primary DC** or **backup DC**.

server-scoped Group Policy object (GPO) distinguished name (DN): A scoped **Group Policy object (GPO) distinguished name (DN)** with a server name included in the path, as is the case for a **server GPO DN**.

server-scoped Group Policy object (GPO) path: A **Group Policy object (GPO)** path with a server name included in the path, as is the case for a **server GPO path**.

service: 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.

service account: A stored set of **attributes** that represent a **principal** that provides a **security context** for services.

Service for User (S4U): Microsoft-specific extensions to the **Kerberos** protocol that allow a service to obtain a **Kerberos service ticket** for a user that has not authenticated to the **Key Distribution Center (KDC)**. **S4U** includes S4U2proxy and S4U2self.

Service for User to Proxy (S4U2proxy): An extension that allows a service to obtain a **service ticket** on behalf of a user to a different service.

Service for User to Self (S4U2self): An extension that allows a service to obtain a **Kerberos service ticket** to itself. The **service ticket** contains the user's groups and can therefore be used in authorization decisions.

service principal: An entity that represents a service at the **Key Distribution Center (KDC)**. The **service principal** has a name and an associated **key**. A subclass of **principal**, a **service principal** generally does not correspond to a human user of the system, but rather to an automated service providing a resource, such as a file server.

service principal name (SPN): The name by which a client uniquely identifies an instance of a service for mutual **authentication**. See [SPNNAMES] for more information about **SPN** format and composing a unique **SPN**. Also see [RFC1964] section 2.1.1.

service provider: A module that abstracts details of underlying transports for generic **DirectPlay** message transmission. Each **DirectPlay** message is transmitted by a **DirectPlay service provider**. The **service providers** that shipped with **DirectPlay 4** are modem, serial, IPX, and TCP/IP.

service (SRV) resource record: A **Domain Name System (DNS)** resource record used to identify computers that host specific services, as specified in [RFC2782]. **SRV resource records** are used to locate **domain controllers (DCs)** for **Active Directory**.

service set identifier (SSID): A sequence of characters that names a wireless local area network (WLAN).

service ticket: A **ticket** for any service other than the **ticket-granting service (TGS)**. A **service ticket** serves only to classify a **ticket** as not a **ticket-granting ticket (TGT)** or cross-realm TGT, as specified in [RFC4120].

session: (1) In **Kerberos**, an active communication channel established through **Kerberos** that also has an associated cryptographic **key**, message counters, and other state.

(2) 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.

(3) In the **Challenge-Handshake Authentication Protocol (CHAP)**, a **session** is a lasting connection between a peer and an authenticator.

(4) In the Workstation service, an authenticated connection between two computers.

(5) An active communication channel established through NTLM, that also has an associated cryptographic **key**, message counters, and other state.

(6) 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.

session key: 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**.

session layer: The fifth layer in the Open Systems Interconnect (OSI) architectural model as defined by the International Organization for Standardization (ISO). The **session layer** is used for establishing a communication **session**, implementing security, and performing **authentication**. The **session layer** responds to service requests from the presentation layer and issues service requests to the **transport layer**.

Session Multiplex Protocol (SMUX): An entity on a network that implements the Secure Socket Tunneling Protocol (SSTP) and that listens for SSTP connections over TCP port 443.

session security: The provision of message integrity and/or confidentiality to a **session**.

SHA: See **system health agent (SHA)**.

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).

shadow copy: A duplicate of data held on a **volume** at a well-defined instant in time.

share: 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.
- A$, B$, C$ (and other local disk names followed by a dollar sign), assigned to local disk devices.

share connect: The act of establishing **authentication** and shared state between a Common Internet File System (CIFS) server and client that allows a CIFS client to access a **share** offered by the CIFS server.

shell: Part of the Windows user interface (UI) that organizes and controls user access to a wide variety of objects necessary for running applications and managing the operating system. The most numerous are the folders and files that reside on computer storage media. There are also a number of virtual objects such as network printers and other computers. The **shell** organizes these objects into a hierarchical namespace and provides an API to access them.

shell link: A data object that contains information used to access another object in the **shell's** namespace--that is, any object visible through Windows Explorer. The types of objects that can be accessed through shell links include files, folders, disk drives, and printers. A **shell link** allows an application to access an object from anywhere in the namespace. The application does not need to know the current name and location of the object.

shell shortcut: A **shell link** that has a shortcut icon; however, the terms **shell link** and **shell shortcut** are often used interchangeably.

SHV: See **system health validator (SHV)**.

SID: See **security identifier**.

signal: In **OleTx**, the act of communicating an event between **facets** inside a **transaction manager**.

signature: A structure that contains a hash and block chunk size. The hash field is 16 bytes, and the chunk size field is a 2-byte unsigned integer. For more information, see [MS-RDC] section 2.2.2.1.

signature file: A file containing the **signatures** of another (source) file. There is a simple header that identifies the type of the file as a **signature file**, the size of the header itself, and the **remote differential compression (RDC)** library version number. Following the header are the **signatures** from the source file in the order they are generated from the chunks. For more information, see [MS-RDC] section 2.2.2.

signing certificates: The **certificate** that represents the identity of an entity (for example, a **certification authority (CA)**, a Web server or an **S/MIME** mail author) and is used to verify **signatures** made by the **private key** of that entity. For more information, see [RFC3280].

similarity data: Information about a file that can be used to determine an appropriate **seed file** to select to reduce the amount of data transferred. **Similarity data** consists of one or more **similarity traits**.

similarity trait: A trait that summarizes an independent feature of a file. The features are computed by taking min-wise independent **hash functions** of a file's **signatures**. For information about how traits are computed, see [MS-RDC] section 3.1.5.4. **Similarity traits** are used in selecting **seed files**.

Simple and Protected GSS-API Negotiation Mechanism (SPNEGO): An **authentication** mechanism that allows **Generic Security Services (GSS)** peers to determine whether their credentials support a common set of GSS-API security mechanisms, to negotiate different options within a given security mechanism or different options from several security mechanisms, to select a service, and to establish a **security context** among themselves using that service. **SPNEGO** is specified in [RFC4178].

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

Simple Mail Transfer Protocol (SMTP): A TCP/IP protocol used in sending and receiving e-mail.

simple volume: A **volume** whose data exists on a single **partition**.

single-instance storage (SIS): An **NTFS** feature that implements links with the semantics of copies for files stored on an **NTFS volume**. **SIS** uses copy-on-close to implement the copy semantics of its links.

single-phase commit: An optimization of the Two-Phase Commit Protocol in which a **transaction manager** delegates the right to decide the outcome of a transaction to its only subordinate participant. This optimization can result in an In Doubt outcome.

site: An **Active Directory** term that defines a set of one or more TCP/IP subnets, where the subnets have high connectivity as measured in terms of latency (low) and bandwidth (high). By defining **sites** (represented by **site objects**) an administrator can easily configure **Active Directory** access and replication topology to take advantage of the physical network. When users log on, **Active Directory** clients find **domain controllers (DCs)** that are in the same **site** as the user or are near the same **site** if there is no **DC** in the **site**. For more information, see [MS-ADTS].

site coverage: The set of **sites** for which a **domain controller (DC)** is responsible, as configured by the administrator.

site distinguished name (DN): The **distinguished name (DN)** for an **object** in **Active Directory** that represents a **site**.

site object: An **object** of class site, representing a **site**.

site of domain controller (DC): The **site object** that is an ancestor of the **DC's** nTDSDSA **object**.

site settings object: For a given **site** with **site object** s, its **site settings object** o is the child of s such that o is of class nTDSSiteSettings and the **relative distinguished name (RDN)** of o is CN=NTDS site settings.

SKU: See **Stock Keeping Unit (SKU)**.

slow sync: The nominator for a synchronization subprotocol that is used to perform a consistency check between the databases of two partners.

small computer system interface (SCSI): A set of standards for physically connecting and transferring data between computers and peripheral devices.

small computer system interface (SCSI) bus: A standard for connecting peripheral devices to a computer. A **SCSI bus** is an implementation of this standard.

smart card: A portable device that is shaped like a business card and is embedded with a memory chip and either a microprocessor or some non-programmable logic. **Smart cards** are often used as **authentication tokens** and for secure **key** storage. **Smart cards** used for secure key storage have the ability to perform cryptographic operations with the stored **key** without allowing the **key** itself to be read or otherwise extracted from the card.

SMB connection: A transport connection between a **Server Message Block (SMB)** client and an **SMB** server. The **SMB connection** is assumed to provide reliable in-order message delivery semantics. An **SMB connection** can be established over any available **SMB** transport that is supported by both the **SMB** client and the **SMB** server, as specified in [MS-CIFS].

SMB dialect: There are several different versions and subversions of the **Server Message Block (SMB)** protocol. A particular version of the **SMB** protocol is referred to as an **SMB dialect**. Different **SMB dialects** can include both new **SMB** messages as well as changes to the fields and semantics of existing **SMB** messages used in other **SMB dialects**. When an **SMB** client connects to an **SMB** server, the client and server negotiate the **SMB dialect** to be used.

SMB session: An authenticated user connection established between an **SMB** client and an **SMB** server over an **SMB connection**. There can be multiple active **SMB sessions** over a single **SMB connection**. The Uid field in the **SMB** packet header distinguishes the various sessions.

SMTP: See **Simple Mail Transfer Protocol (SMTP)**.

snapshot: The point in time at which a **shadow copy** of a **volume** is made.

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**.

SOAP 1.1: Version 1.1 of the **SOAP** (Simple Object Access Protocol) standard. For the complete definition of **SOAP 1.1**, see [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 sematics 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 fault code: The algorithmic mechanism for identifying a **SOAP fault**. See [SOAP1.2-1/2007] section 5.6 for more information.

SOAP fault detail: A string containing a human-readable explanation of a **SOAP fault**, which is not intended for algorithmic processing. See [SOAP1.2-1/2007] section 5.4.5 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 header block: The **XML** block containing the **SOAP header** entries within a **SOAP header**. See [SOAP1.2-1/2007] section 5.2.1 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 mustUnderstand attribute: A global, Boolean **attribute** that is used to indicate whether a header entry is mandatory or optional for the recipient to process. See [SOAP1.2-1/2007] section 5.2.3 for more information.

software installation package: A file that describes other files and metadata necessary to describe an application's executable files and state and to install that application. Also referred to as a "package".

software installation package modification: A file that allows an administrator to specify configuration for an application that is installed on the client through a software installation package.

software maintenance utility: An application that allows users to perform software management activities such as installation, uninstallation, or inventory of applications available through the software installation extension.

software package container distinguished name (DN): A **distinguished name (DN)** of the form "CN=Packages,<ClassStore>" where <ClassStore> is a class store container **DN**.

software package distinguished name (DN): A **distinguished name (DN)** of the form "CN=<SoftwarePackageId>,CN=Packages,<ClassStore>", where <ClassStore> is a class store container **DN** and <SoftwarePackageId> is a **curly braced GUID string** string.

software scripts path: A file system path to a directory with a path of the form "<ScopedGPOPath>\Applications", where <ScopedGPOPath> is a **scoped GPO path**.

SoH: See **statement of health (SoH)**.

SoHR: See **statement of health response (SoHR)**.

source file or source data: A file on a source location that is to be copied by **remote differential compression (RDC)**. Sometimes referred to as "source".

sparse file: A file that has regions of data containing all zeros and in which some of the zero regions do not have disk space allocated for them.

SPD: See **security policy database**.

SPN: See **service principal name**.

spool file: A representation of application content data than can be processed by a print driver. Common examples are enhanced metafile format and XML paper specification. For more information, see [MSDN-META] and [MSDN-XMLP].

SSL: See **Secure Sockets Layer (SSL)**.

SSL/TLS handshake: The process of negotiating and establishing a connection protected by **Secure Sockets Layer (SSL)** or **Transport Layer Security (TLS)**. For more information, see [SSL3] and [RFC2246].

staging file: The backup of the changed file or folder. It encapsulates the data and **attributes** associated with a replicated file or folder. By creating the **staging file**, **File Replication Service (FRS)** ensures that file data can be supplied to partners regardless of any activity that might prevent access to the original file. The **staging files** can be compressed to save disk space and network bandwidth during replication.

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.

standalone CA: A **certification authority (CA)** that is not a member of a **domain**. For more information, see [MSFT-PKI].

standalone machine: A machine that is not a **domain member** or a **domain controller (DC)**.

standard user: A user that does not have administrative rights defined in its **token** and is a member of the users group. Users are prevented from making accidental or intentional system-wide changes but can perform normal daily computer tasks.

state machine: A model of computing behavior composed of a specified number of states, transitions between those states, and actions to be taken. A state stores information about past transactions as it reflects input changes from the startup of the system to the present moment. A transition (such as connecting a network share) indicates a state change and is described by a condition that would need to be fulfilled to enable the transition. An action is a description of an activity that is to be performed at a given moment.

There are several action types:

- Entry action: Performed when entering the state.
- Exit action: Performed when exiting the state.
- Input action: Performed based on the present state and input conditions.
- Transition action: Performed when executing a certain state transition.

statement of health (SoH): A collection of data generated by a **system health entity**, as specified in [MS-SOH], which defines the **health state** of a machine. The data is interpreted by a Health Policy Server, which determines whether the machine is healthy or unhealthy according to the policies defined by an administrator.

statement of health (SoH) client: A synonym for **system health entity**.

statement of health ReportEntry (SoH ReportEntry): A collection of data that represents a specific aspect of the **health state** of a client.

statement of health response (SoHR): A collection of data that represents the evaluation of the **statement of health (SoH)** according to network **policies**, as specified in [MS-SOH].

statement of health response ReportEntry (SoHR ReportEntry): A collection of data that represents the evaluation of a specific aspect of the **health state** of a client, according to network **policies**.

station (STA): Any device that contains an IEEE 802.11 conformant medium access control and physical layer (PHY) interface to the wireless medium (WM).

station management entity (SME): In general, a **station management entity (SME)** is regarded as responsible for functions such as the gathering of layer-dependent status from the various layer management entities and setting the value of layer-specific parameters. An **SME** would typically perform such functions on behalf of general system management entities and would implement standard management protocols.

Stock Keeping Unit (SKU): A unique code that refers to a particular manufactured object or source of revenue. A **SKU** can refer to a retail product (software in a box that is sold through a channel), a subscription program (such as MSDN), or an online service (such as MSN).

stored procedure: A function/method that predefines a set of T-SQL commands that resides in a database server and is available to be called by client applications.

StoreMaster: The single agent responsible for performing certain updates to file-link information stored in VolumeTable and FileTable within an **Active Directory Table (ADT)**. For more information on VolumeTable and FileTable, see [MSDLT].

stream: 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.

strict NDR/NDR64 data consistency check: A set of related rules for data validation during processing of an octet stream.

structural class: See **structural object class**.

structural object class: An **object class** that is not an **88 object class** and can be instantiated to create a new **object**.

sub-authentication: Optional and additional **authentication** functionality, usually provided by extending an **authentication** algorithm.

sub-authentication package: An optional component that provides additional **authentication** functionality. If a **sub-authentication package** is installed, the **authentication** package

calls the **sub-authentication package** before returning its **authentication** result. The request to verify by a **sub-authentication package** is indicated by the **ParameterControl** field of the *LogonInformation* parameter (see [MS-APDS] section 3.1.5.2.1, Verifying Responses with Sub-Authentication Packages).

subkey: A child node in the logical tree of the hierarchical data store.

subnet site: The association of a **site** with a particular client, based on the client's IP address.

subordinate transaction manager: A role taken by a **transaction manager** that is responsible for voting on the outcome of an **atomic transaction**. A **subordinate transaction manager** coordinates the voting and notification of its subordinate participants on behalf of its **superior transaction manager**. When communicating with those subordinate participants, the **subordinate transaction manager** acts in the role of **superior transaction manager**. The root **transaction manager** is never a **subordinate transaction manager**. A **subordinate transaction manager** has exactly one **superior transaction manager**.

SubRequest: A request within a SYNC_VOLUME or SEARCH request.

superclasses and subclasses: Types of **Common Information Model (CIM)** classes. A subclass is derived from a superclass. The subclasses inherit all features of its superclass but can add new features or redefine existing ones. A superclass is the **CIM** class from which a **CIM** class inherits.

superior transaction manager: A role taken by a **transaction manager** that is responsible for gathering outcome votes and providing the final transaction outcome. A root **transaction manager** can act as a **superior transaction manager** to a number of **subordinate transaction managers**. A **transaction manager** can act as both a **subordinate transaction manager** and a **superior transaction manager** on the same transaction.

symbolic link: A **symbolic link** is a **reparse point** that points to another **file system object**. The **object** being pointed to is called the target. **Symbolic links** are transparent to users; the links appear as normal files or directories, and can be acted upon by the user or application in exactly the same manner. **Symbolic links** can be created using the FSCTL_SET_REPARSE_POINT request as specified in [MS-FSCC] section 2.3.53. They can be deleted using the FSCTL_DELETE_REPARSE_POINT request as specified in [MS-FSCC] section 2.3.5. Implementing **symbolic links** is optional for a **file system**.

symmetric algorithm: A cryptographic algorithm that uses one **secret key** that may be shared between authorized parties. The key must be kept secret between communicating parties. The same key is used for both **encryption** and decryption. For an introduction to this concept and terminology, see [CRYPTO] section 1.5, [IEEE1363] section 3, and [SP800-56A] section 3.1.

symmetric encryption: An **encryption** method that uses the same cryptographic **key** to encrypt and decrypt a given message.

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.

synchronous operation: An operation that is executed on the server side while the client is waiting for the response message.

syntax: See **attribute syntax**.

system access control list (SACL): An **access control list (ACL)** that controls the generation of audit messages for attempts to access a securable object. The ability to get or set an object's **SACL** is controlled by a privilege typically held only by system administrators.

system command: A message that is sent to a window or notification icon via its system menu, or via a keyboard shortcut. Common **system commands** include minimize, maximize, move, and so on.

system directory: A **directory** that contains system files comprising the operating system.

system health agent (SHA): The client components that make declarations on a specific aspect of the client **health state** and generate a **statement of health ReportEntry (SoH ReportEntry)**.

system health entity: See **system health agent (SHA)**.

system health validator (SHV): The server counterpart to the **system health agent (SHA)**, which is responsible for verifying the declarations of client **health state** made by the respective **SHA**. The **SHV** generates a **statement of health response ReportEntry (SoHR ReportEntry)**.

System menu: See **window menu**.

system partition: A **partition** that contains the **boot loader** needed to invoke the operating system on the **boot partition**. A **system partition** must also be an **active partition**. It can be, but is not required to be, the same **partition** as the **boot partition**.

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**.

21 T

table response: A collection of data, all formatted in a specific manner, that is sent by the server to the client for the purpose of communicating the result of a client request. The server returns the result in a table response format for LOGIN7, SQL, and remote procedure call (RPC) requests.

target file: A file on the target location that is the destination of a **remote differential compression (RDC)** copy.

taskbar: A window, anchored to an edge of the screen, that contains the Start button and buttons for all open programs.

terminal server: A computer on which **Terminal Services** is running.

Terminal Services: A service on a server computer that allows delivery of applications, or the desktop itself, to various computing devices. When a user runs an application on a **terminal server**, the application execution takes place on the server computer and only keyboard, mouse, and display information is transmitted over the network. Each user sees only his or her individual session, which is managed transparently by the server operating system and is independent of any other client session.

tick count: In **DirectPlay**, the count from when the system was booted, in milliseconds.

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.

ticket-granting service (TGS): A service that issues **tickets** for admission to other services in its own domain or for admission to the ticket-granting service in another domain.

ticket-granting service (TGS) exchange: The Kerberos subprotocol in which the **key distribution center (KDC)** distributes a session key and a ticket for the service requested by the client, as specified in [RFC4120] section 3.3. This exchange is initiated when the client sends the **KDC** a KRB_TGS_REQ message.

ticket-granting ticket (TGT): A special type of **ticket** that can be used to obtain other **tickets**. The TGT is obtained after the initial authentication in the **Authentication Service (AS) exchange**; thereafter, users do not need to present their credentials, but can use the TGT to obtain subsequent tickets.

time peer: A **time source** with which a **time provider** is synchronized. A **time provider** can have more than one time peer.

time provider: A component that a **time service** relies on to either obtain accurate time stamps (from network or hardware time sources) or to provide those time stamps to other computers over the network.

time service: A system service that implements support for synchronizing a computer's local time with a **time source**.

time source: A component that possesses a clock and that makes the clock's time available to other components for synchronization. For more information, see "reference source" in [RFC1305].

TLN: See **top-level name**.

TLS: See **Transport Layer Security (TLS)**.

TLV: See **type-length-value**.

token: A set of rights and privileges for a given user.

tombstone: (1) A deleted object in the directory that remains in storage until a configured amount of time, called the **tombstone lifetime**, has passed. After the **tombstone lifetime** expires, the object is permanently deleted. By keeping the tombstone in existence for the **tombstone lifetime**, the deleted state of the object is able to replicate.

(2) In Distributed File System Replication (DFS-R), an update pertaining to a file deletion.

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.

tool extension GUID or administrative plug-in GUID: A GUID defined separately for each of the user policy settings and computer policy settings that associates a specific administrative tool plug-in with a set of policy settings that can be stored in a **Group Policy object (GPO)**.

tooltip: A window displaying text that is created when the mouse is moved over a window or notification icon.

top level name (TLN): The root namespace of a forest. For example, if the forest a.com contains the domains a.com, b.a.com, and c.a.com, the TLN would be a.com.

topology: The structure of the connections between members.

track: 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.

transaction: In OleTx, an **atomic transaction**.

transaction identifier: The **GUID** that uniquely identifies 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.

transaction propagation: The act of coordinating two transaction managers to work together on a single **atomic transaction**. When propagating a transaction to a transaction manager that is not already a participant in the transaction, that transaction manager plays the role of subordinate transaction manager to the originating transaction manager, which will play the role of superior transaction manager. When propagating a transaction to a transaction manager that is already a participant in the transaction, no new superior or subordinate relationship is established.

transitive trust: The state of two domains establishing trust through an intermediary domain. For example, if domain A trusts domain B, and domain B trusts domain C, then domain A may be configured to trust domain C through transitive trust.

Transmission Control Protocol (TCP): A protocol used with the Internet Protocol (IP) to send data in the form of message units between computers over the Internet. TCP handles keeping

track of the individual units of data (called packets) that a message is divided into for efficient routing through the Internet.

transport layer: The fourth layer in the Open Systems Interconnection (OSI) architectural model as defined by the International Organization for Standardization (ISO). The transport layer provides for transfer correctness, data recovery, and flow control. The transport layer responds to service requests from the session layer and issues service requests to the network layer.

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.

transport mode: An IP encapsulation mechanism, as specified in [RFC4301], that provides **Internet Protocol security (IPsec)** security for host-to-host communication.

Triple Data Encryption Standard: A block cipher that is formed from the **Data Encryption Standard (DES)** cipher by using it three times.

trust: To accept 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.

trust attributes: A collection of attributes that define different characteristics of a trust within a domain or a forest.

trust object: An object representing a **trust**.

trust path: In a graph of domain **trusts**, the path through the graph between two domains that are linked by transitive trust. For example, if domain A trusts domain B, and domain B trusts domain C, then the trust path is A->B->C.

trust root: A store within the computer of a relying party that is protected from tampering and in which the root keys of all root CAs are held. Those root keys are typically encoded within self-signed certificates, and the contents of a trust root are therefore sometimes called **root certificates**.

trust secret: A pair of keys used to encrypt or sign sensitive protocol data between two trust authorities, such as domain controllers.

trusted domain: A domain that is trusted to make authentication decisions for security principals in that domain.

trusted domain object (TDO): A collection of properties that define a trust relationship with another domain, such as direction (outbound, inbound, or both), trust attributes, name, and security identifier of the other domain. For more information, see [MS-ADTS].

trusted forest: A forest that is trusted to make authentication statements for security principals in that forest. Assuming forest A trusts forest B, all domains belonging to forest A will trust all domains in forest B, subject to policy configuration.

trusted platform module (TPM): A microcontroller that stores keys, passwords, and digital certificates, typically affixed to the motherboard of a PC.

trustee: The recipient, expressed as a **security identifier (SID)**, of an access control capability expressed in a security descriptor.

TSpec: A set of characteristics that is used to specify network traffic behavior, as specified in [RFC2212].

tunnel: The encapsulation of one network protocol within another.

tunnel mode: An IP encapsulation mechanism, as specified in [RFC4301], that provides Internet Protocol security (IPsec) security to tunneled IP packets. IPsec processing is performed by the tunnel endpoints, which can be (but are typically not) the end hosts.

two-phase commit: An agreement protocol that is used to resolve the outcome of an atomic transaction in response to a commit request from the root application. Phase One and Phase Two are the distinct phases of the Two-Phase Commit Protocol.

type-length-value (TLV): A method of organizing data that involves a Type code (16-bit), a specified length of a Value field (16-bit), and the data in the Value field (variable).

22 U

UCHAR: A single, unsigned byte.

ULONG: A single, unsigned long integer consisting of 32 bits. This data type can have a range from 0 to 4,294,967,395 (0xffffffff).

unallocated disk: A disk that is visible to the local machine but is not formatted with a recognized partitioning format such as **master boot record (MBR)** or **GUID partitioning table (GPT)**.

UncPath: The location of a file in a network of computers, as specified in **Universal Naming Convention (UNC)** syntax.

unicast: A delivery method used by media servers for providing content to connected clients in which each client receives a discrete stream that no other client has access to.

Unicode: 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).

Unicode character: Unless otherwise specified, a 16-bit UTF-16 code unit.

Unicode string: A **Unicode** 8-bit string is an ordered sequence of 8-bit units, a **Unicode** 16-bit string is an ordered sequence of 16-bit code units, and a **Unicode** 32-bit string is an ordered sequence of 32-bit code units. In some cases, it may be acceptable not to terminate with a terminating null character. Unless otherwise specified, all **Unicode strings** follow the **UTF-16LE** encoding scheme with no Byte Order Mark (BOM).

Uniform Resource Locator (URL): A string of characters in a standardized format that identifies a document or resource on the World Wide Web.

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 Disk Format (UDF): A type of file system for storing files on optical media.

universal group: A group that can appear in **access control lists (ACLs)** anywhere in the forest, and can contain other universal groups, global groups, and users from anywhere within the forest.

Universal Naming Convention (UNC): A string format that specifies the location of a resource. For more information, see [MS-DTYP] section 2.2.56.

Universal Plug and Play (UPnP): A set of computer network protocols, published by the UPnP Forum [UPnP], that allow devices to connect seamlessly and that simplify the implementation of networks in home (data sharing, communications, and entertainment) and corporate environments. UPnP achieves this by defining and publishing UPnP device control protocols built upon open, Internet-based communication standards.

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 **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 specified in [RFC4122] or [C706] must be used for generating the UUID.

unmarshal: (1) The process of deserializing one or more data structures from an octet stream using a specific transfer syntax (for example, unmarshaling a 32-bit integer).

(2) In **remote procedure call (RPC)**, the process of decoding one or more data structures from an octet stream using a specific **RPC** Transfer Syntax.

unmasked disk: A disk that is visible to the local machine.

unnamed stream: See **main stream**.

update: An add, modify, or delete of one or more objects or attribute values.

UPDATE: 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 sequence number (USN): (1) A monotonically increasing sequence number used in assigning a **stamp** to an originating update. For more information, see [MS-ADTS].

(2) The offset from the beginning of the change journal stream that uniquely identifies a change journal record.

(3) Contains a 64-bit value representing the number of 100-nanosecond intervals since January 1, 1601 Coordinated Universal Time (UTC). This is updated whenever the packageRegistration object is modified on the server to match the server's UTC time at the time of the update.

updates: A set of **UPDATE** entities.

upgrade: A relationship between software packages in which the upgrading application will replace a particular software installation package (the upgraded application) if it exists on a client and was installed through the software installation protocol extension. Logically, this means that the client application will be uninstalled and the upgrading application will be installed.

upgrading application: If two applications are related due to an upgrade, this is the application to remove from the client when the upgrading application is installed.

uplevel trust: A trust in which both peers are running Windows 2000 or later domain controllers.

upstream partner: The partner that sends out change orders, files, and folders.

URI: This term is used as specified in [RFC2616].

URL: See **Uniform Resource Locator (URL)**.

URL host name: The **URL** host name (as specified in [RFC3986] Appendix A), with the extensions described in [MS-HNDS].

User Account Control (UAC): A set of security options associated with a security principal. Particular options designate the role computers in the domain.

user account database: A database that maintains **user account information**.

user account database replication: A mechanism for synchronizing the user accounts database among multiple domain controllers.

user account information: The user name, password, and groups in which the user account has membership.

user agent: An HTTP user agent, as specified in [RFC2616].

user assistance resource: A UnicodeString that contains a URL pointing to information that might be helpful to users of the application when viewing information about the application through a software maintenance tool. This is defined by the administrator who deploys the application.

user connection: A connection to a printer (shared from a print server) on a client machine. A connection is displayed in the user interface as a printer. User connections are displayed for only one user of a particular client machine.

User Datagram Protocol (UDP): The connectionless protocol within TCP/IP that corresponds to the transport layer in the ISO/OSI reference model.

user-defined message tag (MTAG): A message that is sent within the context of an established connection. See also **message tag (MTAG)**.

user GPO version: A version number of the changes for the user policy portion of a **Group Policy object (GPO)**. This is a 16-bit integer encoded in the upper 16 bits of a **GPO** version.

user message: A message that is sent between instances of an application using the DirectPlay network library as a transport.

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.

user policy mode: A mode of policy application that is used to retrieve settings for an authenticated domain user account, interactively logged on to a client.

user principal name (UPN): A user account name (sometimes referred to as the user logon name) and a domain name identifying 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 e-mail address). In Active Directory, the UPN is the userPrincipalName attribute of the account object, as specified in [MS-ADTS].

user profile: A collection of attributes on a user object that are used to customize an end-user experience.

user profile folder: A storage location in an operating system that provides the operating system and applications with a per-user location with conventional semantics. For example, each user on a Windows operating system has his or her own documents, music, videos, and pictures user-profile folders in which he or she may store per-user data.

user-scoped Group Policy object distinguished name: A scoped **Group Policy object (GPO) distinguished name (DN)** that begins with "CN=User".

user-scoped Group Policy object path: A scoped **Group Policy object (GPO)** path that ends in "\User".

USHORT: A single, unsigned short integer consisting of 16 bits. This data type can have a range from 0 to 65,535 (0xffff).

USN: See **update sequence number (USN)**.

UTC (Coordinated Universal Time): 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).

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 (Unicode Transformation Format, 16-bits, little-endian): The encoding scheme specified in [UNICODE5.0.0/2007] section 2.6 for encoding **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.

UUID: See **universally unique identifier (UUID)**.

UUID or GUID: See **universally unique identifier (UUID)**. See **globally unique identifier (GUID)**.

23 V

value: A data element associated with a key.

vendor ID payload: A particular type of **ISAKMP payload** that contains a vendor-defined constant. The constant is used by vendors to identify and recognize remote instances of their implementations. This mechanism allows a vendor to experiment with new features while maintaining backward compatibility. For more information, see [RFC2408] section 3.16.

version chain vector: A data structure that maps machine **GUIDs** to sets of version sequence numbers.

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.

version vector: (1) A mapping from machine identifiers to **version sequence numbers**. The Distributed File System Replication (DFS-R) Protocol uses a generalization of version vectors called **version chain vectors**.

(2) A vector of **volume sequence numbers (VSNs)**, with one entry per replica set member, as identified by originator **GUID**. All change orders carry the originator **GUID** of the originating member and the associated **VSN**. As each replica member receives the update, it tracks the **VSN** in a vector slot that is assigned to the originating member. This vector specifies whether the replica tree is current with each member. The version vector is then used to filter updates from inbound partners that may have already been applied. The version vector is also delivered to the inbound partner when the two members join. When a new connection is created, the version vector is used to scan the file ID table for more recent updates that are not seen by the new outbound partner.

version vector join (vvjoin): The process in which a downstream partner joins with an upstream partner for the first time. It is also called initial sync.

virtual cluster number (VCN): The cluster number relative to the beginning of the file, directory, or stream within a file.

virtual connection: A **Server Message Block (SMB)** connection between an **SMB** client and **SMB** server.

Virtual Disk Service (VDS): The service component running on the server.

Virtual Disk Service (VDS) object: An instance of a class that exposes one or more DCOM interfaces to query or configure the **Virtual Disk Service**, the operating system device (such as a disk or volume), or the concept (such as a software provider) that the object represents.

Virtual Disk Service (VDS) provider: A concept that models the software responsible for storage management. A VDS software provider performs operations on disk and volume devices exposed to the operating system.

Virtual Disk Service (VDS) session: The point at which a client receives an instance of the VDS service object until the point at which it releases it. Unless otherwise indicated, the term **session** refers to a VDS session.

Virtual Interface Architecture (VIA): A high-speed interconnect that requires special hardware and drivers provided by third parties.

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 data: Data stored on a **volume**.

volume identifier (VolumeId): A 128-bit value used to represent a **volume**. The value of a **VolumeId** is unique on a single computer (the local file system or a remote file server).

volume label: See **file system label**.

volume manager: A system component that manages communication and data transfer between applications and disks.

volume members: See **RAID column**.

volume mount name: A path for a volume. The path consists of a **GUID** formatted as a string. Applications can use this path to open the volume.

volume parity – SCSI: See **RAID column**. See **RAID-5**.

volume plex: A member of a **volume** that represents a complete copy of data stored. For instance, mirrored volumes have more than one plex.

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.

VolumeID: A unique identifier that represents the identity of a file system volume.

VolumeInformation: Information about a volume, which is stored on the volume, such as its VolumeID and VolumeSequenceNumber.

VolumeOwner: A MachineID that is considered to be the owner of a VolumeID. A VolumeID can only have one VolumeOwner. For more information, see [MS-DLTM].

VolumeSecret: A value that is used to establish a VolumeOwner. For more information, see [MS-DLTM].

VolumeSequenceNumber: An integer value used to track the sequence of move notification messages received by the protocol server.

VolumeTable: Maps a VolumeID to a RefreshTime, VolumeSequenceNumber, VolumeSecret, and VolumeOwner. For more information, see [MS-DLTM].

vvjoin: See **version vector join (vvjoin)**.

24 W

Web server: A computer that delivers Web pages to browsers, and other files to applications by way of the HTTP protocol. A Web server includes the hardware, operating system, Web server software, TCP/IP protocols, and site content. A Web server typically accepts HTTP requests from clients and serves them HTTP responses that contain optional data, which are usually Web pages such as HTML documents, and linked objects such as images.

Web services: A software system designed to support interoperable machine-to-machine interaction over a network, using XML-based standards and open transport protocols.

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 Services Reliable Messaging (WSRM) Protocol: A protocol that defines mechanisms that enable Web services to ensure delivery of messages over unreliable communication networks. The WSRM Protocol allows different operating and middleware systems to reliably exchange these messages.

well-known endpoint: A preassigned, network-specific, stable address for a particular client/server instance. For more information, see [C706].

well-known security identifier: One of a number of **security identifiers (SIDs)** that has a constant value.

window menu: A context menu associated with a window or notification icon that contains a list of common operations to perform such as minimize, maximize, move, and so on.

Windows-1252 character set: An 8-bit character encoding of the Latin alphabet used by default in most legacy components of Windows. Sometimes referred to as an "ANSI" character set, this is a misnomer because a Windows-1252 character set is not an ANSI standard code page; it is properly described as a "single byte code page". The Windows-1252 character set is defined in [Windows-1252]. Windows is now based on Unicode, so the use of the Windows-1252 character set is strongly discouraged unless it is used to interoperate with legacy applications or legacy data.

window coordinates: Coordinates relative to the top-left corner of the window.

window visible region: The portion of the window that is not obscured by other user interface elements.

Windows error code: A 32-bit quantity where zero represents success and nonzero represents failure. The specific failure codes are specified in [MS-ERREF].

Windows Internet Name Service (WINS): A name service for the NetBIOS protocol, particularly designed to ease transition to a TCP/IP based network.

Windows Management Instrumentation (WMI): The Microsoft implementation of **Common Information Model (CIM)**.

Windows metafile format (WMF): A file format used by Windows that supports the definition of images.

Windows NT account name: A string identifying the name of a Windows NT account.

Windows NT domain: A domain hosted entirely on Windows NT 4.0 servers.

Windows NT domain name: A string identifying the name of a Windows NT domain to which an identity belongs.

Windows NT password: A string of 0 to 256 case-sensitive Unicode (as specified in [RFC2781]) characters used to prove entitlement to a Windows NT account.

Windows registry: The Windows implementation of the registry.

Windows security descriptor: A Windows NT **security descriptor**.

Windows Server Enterprise: A version (also referred to as a **Stock Keeping Unit (SKU)**) of the Windows Server 2003 R2 operating system.

Windows Time Service (W32Time): A time service implemented on Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2. The service supports time synchronization against network and hardware time sources. For more information, see [WTSREF] and [MS-SNTP].

wireless Local Area Network (WLAN): A local area network (LAN) to which mobile users (clients) can connect and communicate by means of high-frequency radio waves rather than wires. WLANs are specified in the IEEE 802.11 standard [IEEE802.11-2007].

Wi-Fi Protected Access (WPA): For more information, see [WPA].

Wi-Fi Protected Access 2 (WPA2): For more information, see [WPA2].

WMI Query Language (WQL): A subset of American National Standards Institute Structured Query Language (ANSI SQL). It differs from the standard SQL in that it retrieves from classes rather than tables and returns **CIM** classes or instances rather than rows.

writable naming context (NC) replica: A **naming context (NC)** replica that accepts originating updates. Partial replicas are not writable.

writability: The abstract feature capability representing the ability of a **domain controller (DC)** to accept modifications and issue originating updates, with respect to a given **naming context (NC)** replica.

write-protect: An attribute of storage media denoting that the media is not available to be written.

WSDL message: An element that describes the data being exchanged between Web service providers and clients.

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.

25 X

X.509: An ITU-T standard for public key infrastructure subsequently adapted by the IETF, as specified in [RFC3280].

XA Protocol: The protocol specified in [XOPEN-DTP].

XA resource manager bridge: A software component that allows an application to enlist an XA Resource Manager in an OleTx Transaction.

XA resource manager bridge facet: A software component that allows a Transaction Manager to communicate with an **XA resource manager bridge**.

XML: The Extensible Markup Language, as specified in [XML1.0].

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 schema (XSD): A language that defines the elements, attributes, namespaces, and data types for **XML** documents as defined by [XMLSCHEMA1/2] and [W3C-XSD] standards. An XML schema uses **XML** syntax for its language.

26 Z

z-order: The top-to-bottom order of the windows on a desktop. Windows lower in the z-order are obscured by overlapping windows higher in the z-order.

27 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.