[MS-OXCSTOR]:

Store Object Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **Copyrights**. This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
- **Patents**. Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks**. The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names**. The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
4/4/2008	0.1	New	Initial Availability.
4/25/2008	0.2	Minor	Revised and updated property names and other technical content.
6/27/2008	1.0	Major	Initial Release.
8/6/2008	1.01	Minor	Revised and edited technical content.
9/3/2008	1.02	Minor	Revised and edited technical content.
12/3/2008	1.03	Minor	Revised and edited technical content.
2/4/2009	1.04	Minor	Revised and edited technical content.
4/10/2009	2.0	Major	Updated technical content for new product releases.
7/15/2009	3.0	Major	Revised and edited for technical content.
11/4/2009	4.0.0	Major	Updated and revised the technical content.
2/10/2010	5.0.0	Major	Updated and revised the technical content.
5/5/2010	6.0.0	Major	Updated and revised the technical content.
8/4/2010	7.0	Major	Significantly changed the technical content.
11/3/2010	8.0	Major	Significantly changed the technical content.
3/18/2011	8.0	None	No changes to the meaning, language, or formatting of the technical content.
8/5/2011	8.1	Minor	Clarified the meaning of the technical content.
10/7/2011	8.2	Minor	Clarified the meaning of the technical content.
1/20/2012	9.0	Major	Significantly changed the technical content.
4/27/2012	10.0	Major	Significantly changed the technical content.
7/16/2012	10.0	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	11.0	Major	Significantly changed the technical content.
2/11/2013	12.0	Major	Significantly changed the technical content.
7/26/2013	12.1	Minor	Clarified the meaning of the technical content.
11/18/2013	13.0	Major	Significantly changed the technical content.
2/10/2014	13.0	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	13.0	None	No changes to the meaning, language, or formatting of the technical content.
7/31/2014	13.0	None	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
10/30/2014	14.0	Major	Significantly changed the technical content.
3/16/2015	15.0	Major	Significantly changed the technical content.
5/26/2015	16.0	Major	Significantly changed the technical content.
9/14/2015	17.0	Major	Significantly changed the technical content.
6/13/2016	18.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction

The Store Object Protocol is used by clients to log on to a private user **mailbox** or **public folder**; read and write mailbox-level properties for that user mailbox; perform various housekeeping tasks for that mailbox, such as interacting with the server during and after a move mailbox operation; and determine the availability of content for public folders.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

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

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

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

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.

binary large object (BLOB): A discrete packet of data that is stored in a database and is treated as a sequence of uninterpreted bytes.

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

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.

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).

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

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

enterprise/site/server distinguished name (ESSDN): An X500 DN that identifies an entry in an abstract naming scheme that is separate from an address book. The naming scheme defines enterprises, which contain sites, and sites contain servers and users. There is no concrete data structure that embodies an ESSDN. Instead, an address book entry can contain an ESSDN as a property of the entry.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Out of Office (OOF): One of the possible values for the free/busy status on an appointment. It indicates that the user will not be in the office during the appointment.

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

Receive folder: A Folder object that is configured to be the destination for email messages that are delivered.

remote operation (ROP): An operation that is invoked against a server. Each ROP represents an action, such as delete, send, or query. A ROP is contained in a ROP buffer for transmission over the wire.

remote procedure call (RPC): A context-dependent term commonly overloaded with three meanings. Note that much of the industry literature concerning RPC technologies uses this term interchangeably for any of the three meanings. Following are the three definitions: (*) The runtime environment providing remote procedure call facilities. The preferred usage for this meaning is "RPC runtime". (*) The pattern of request and response message exchange between two parties (typically, a client and a server). The preferred usage for this meaning is "RPC exchange". (*) A single message from an exchange as defined in the previous definition. The preferred usage for this term is "RPC message". For more information about RPC, see [C706].

replica: A server that hosts an instance of a message item in a folder.

replica GUID (REPLGUID): A value that represents a namespace for identifiers. If a REPLGUID is combined with a GLOBSET, the result is a set of global identifiers. A REPLGUID value has an associated **replica ID (REPLID)** that is used in its place on disk and on the wire.

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

Root folder: The **special folder** that is the top-level folder in a message store hierarchy. It contains all other Folder objects in that message store.

ROP request: See **ROP request buffer**.

ROP request buffer: A ROP buffer that a client sends to a server to be processed.

ROP response: See **ROP response buffer**.

ROP response buffer: A ROP buffer that a server sends to a client to be processed.

Server object: An object on a server that is used as input or created as output for **remote operations (ROPs)**.

Session Context: A server-side partitioning for client isolation. All client actions against a server are scoped to a specific Session Context. All messaging objects and data that is opened by a client are isolated to a Session Context.

special folder: One of a default set of Folder objects that can be used by an implementation to store and retrieve user data objects.

Store object: An object that is used to store **mailboxes** and **public folder** content.

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).

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

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

[MS-LCID] Microsoft Corporation, "Windows Language Code Identifier (LCID) Reference".

[MS-OXCDATA] Microsoft Corporation, "Data Structures".

[MS-OXCFOLD] Microsoft Corporation, "Folder Object Protocol".

[MS-OXCFXICS] Microsoft Corporation, "Bulk Data Transfer Protocol".

[MS-OXCMAPIHTTP] Microsoft Corporation, "Messaging Application Programming Interface (MAPI) Extensions for HTTP".

[MS-OXCMSG] Microsoft Corporation, "Message and Attachment Object Protocol".

[MS-OXCNOTIF] Microsoft Corporation, "Core Notifications Protocol".

[MS-OXCPRPT] Microsoft Corporation, "Property and Stream Object Protocol".

[MS-OXCROPS] Microsoft Corporation, "Remote Operations (ROP) List and Encoding Protocol".

[MS-OXCRPC] Microsoft Corporation, "Wire Format Protocol".

[MS-OXDISCO] Microsoft Corporation, "Autodiscover HTTP Service Protocol".

[MS-OXDSCLI] Microsoft Corporation, "Autodiscover Publishing and Lookup Protocol".

[MS-OXODLGT] Microsoft Corporation, "Delegate Access Configuration Protocol".

[MS-OXORULE] Microsoft Corporation, "Email Rules Protocol".

[MS-OXOSFLD] Microsoft Corporation, "Special Folders Protocol".

[MS-OXPROPS] Microsoft Corporation, "Exchange Server Protocols Master Property List".

[MS-OXWOOF] Microsoft Corporation, "Out of Office (OOF) Web Service Protocol".

[MS-UCODEREF] Microsoft Corporation, "Windows Protocols Unicode Reference".

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

1.2.2 Informative References

None.

1.3 Overview

1.3.1 Private and Public Stores

The client can log on to a private user mailbox for access to that user's mailbox data (folders, messages, and attachments). Once logged on, the client can use this protocol to perform operations on that mailbox. The client can also simultaneously log on to another user'' mailboxes and, granted sufficient permissions by that other user, access that user's mailbox data as well as perform

operations on the mailbox. Additionally, the client can simultaneously log on to a public folder **message store**.

The content within an entire private mailbox is confined to a single server. The client determines which server to log on to from **global directory** data about the user. If the mailbox has been moved to another server, then an attempt to log on to the wrong server results in an error response from the server, along with a return value providing guidance about which server to try next.

Move mailbox is an administrative operation during which a mailbox is moved from one server mailbox database to another server mailbox database. The client interacts with the server during and after the move mailbox operation to determine the location of the source and destination servers that hold the mailbox databases.

The content within the public folders message store is typically spread across many different containers and is replicated among those containers. A container can be a mailbox or a database. The client determines which public folder container to log on to by using the global directory information about the user. All of the containers that host public folders contain a complete copy of the folder hierarchy of the public folders message store. However, a specific container does not have to contain the contents of any particular public folder. The set of containers with content for a specific folder are said to be content **replicas** for that folder. If the client attempts to read folder content from a container that is not a content replica for that folder, it will result in an error response from the server. The client is then able to use the operations described in this protocol to discover which containers have content replicas for the folder. After making that determination, the client then logs on to one of those containers to read or update the content for that public folder.

1.4 Relationship to Other Protocols

This protocol relies on the Remote Operations (ROP) List and Encoding Protocol, as described in [MS-OXCROPS].

All protocols that issue ROPs rely on this protocol.

When a mailbox has been moved to a different mailbox server, this protocol uses the Autodiscover HTTP Service Protocol, as described in [MS-OXDISCO], and the Autodiscover Publishing and Lookup Protocol, as described in [MS-OXDSCLI], to discover the new mailbox server.

For conceptual background information and overviews of the relationships and interactions between this and other protocols, see [MS-OXPROTO].

1.5 Prerequisites/Preconditions

This protocol assumes that the client has previously established a **Session Context** with the server, as specified in [MS-OXCRPC] and [MS-OXCMAPIHTTP]. Once that connection is made, the client is then able to follow the protocol specified in this document to establish a logon session with a private mailbox, or the public folders. After the logon session is established, the client follows the protocol specified in this document to perform various operations on the user mailbox and make discoveries about where public folder content is located.

All ROPs described in [MS-OXCROPS], except the **RopLogon** ROP (section 2.2.1.1), are performed with the assumption that the client has successfully logged on to the server using **RopLogon** ROP.

1.6 Applicability Statement

The **Store object** represents the connection to a specific mailbox or the public folder message store and is identified by a **Logon object handle**. This Logon object handle is used by all other protocols which issue ROPs, including the ROPs described in this protocol.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

The **ROP request buffers** and **ROP response buffers** specified by this protocol are sent to and received from the server by using the underlying Remote Operations (ROP) List and Encoding Protocol, as specified in [MS-OXCROPS].

2.2 Message Syntax

Unless otherwise specified, unit sizes in the following sections are expressed in bytes.

2.2.1 Remote Operations

The following sections specify the ROP request buffers and ROP response buffers that are specific to the Store Object protocol.

2.2.1.1 RopLogon ROP

The **RopLogon** ROP ([MS-OXCROPS] section 2.2.3.1) establishes a logon session between the client and the server. It is the basis of all further ROPs, and successfully completing a **RopLogon** ROP is a prerequisite for performing all other ROPs listed in this specification.

The complete syntax of the **ROP** request and response buffers for this ROP is specified in [MS-OXCROPS]. The following sections specify the syntax and semantics of various fields that are not fully specified in [MS-OXCROPS].

2.2.1.1.1 RopLogon ROP Request Buffer

The following descriptions define valid fields for the **RopLogon** ROP request buffer ([MS-OXCROPS] section 2.2.3.1.1).

LogonFlags: This field contains flags that control the behavior of the logon. Individual flag values and their meanings are specified in the following table. The client MUST NOT set any unspecified flags.

Flag name	Value	Description
Private	0x01	This flag is set for logon to a private mailbox and is not set for logon to public folders.
Undercover	0x02	This flag is ignored by the server.
Ghosted	0x04	This flag is ignored by the server.<1>
SpoolerProcess	0x08	This flag is ignored by the server.

OpenFlags: This field contains additional flags that control the behavior of the logon. Individual flag values and their meanings are specified in the following table. The client MUST NOT set any unspecified flags.

Name	Value	Description
USE_ADMIN_PRIVILEGE	0x00000001	A request for administrative access to the mailbox.
PUBLIC	0x00000002	A request to open a public folders message store. This

Name	Value	Description
		flag MUST be set for public logons.
HOME_LOGON	0x00000004	This flag is ignored.<2>
TAKE_OWNERSHIP	0x00000008	This flag is ignored.<3>
ALTERNATE_SERVER	0x00000100	Requests a private server to provide an alternate public server.
IGNORE_HOME_MDB	0x00000200	This flag is used only for public logons. When set, this flag allows the client to log on to a public message store that is not the user's default public message store; otherwise, attempts to log on to a public message store that is not the user's default results in the client being redirected back to the user's default public message store.
NO_MAIL	0x00000400	A request for a nonmessaging logon session. Nonmessaging sessions allow clients to access the message store, but do not allow messages to be sent or received.
USE_PER_MDB_REPLID_MAPPING	0x01000000	For a private-mailbox logon. This flag SHOULD<4> be set. For logons to a public folder message store, this flag is ignored.
SUPPORT_PROGRESS	0x20000000	Indicates that the client supports asynchronous processing of **RopSetReadFlags**, as specified in [MS-OXCMSG] section 2.2.3.10.1.<5>

EssdnSize: This field contains the size, in bytes, of the **Essdn** field.

Essdn: In the case of a private mailbox logon, this field contains an **ASCII** string that uniquely identifies a mailbox to log on to. In the case of a public folder logon, this field SHOULD<6> contain an ASCII string that uniquely identifies the public folder mailbox to log on to. The string value contained in the **Essdn** field includes the terminating NULL character. The string length (including the terminating NULL character) MUST be equal to the value specified by the **EssdnSize** field. If the value of the **EssdnSize** field is 0x00, the **Essdn** field is empty.

In a private mailbox logon, the string to be used in this field is the value of the legacy **distinguished name (DN)** attribute of the user object that is obtained by using the Autodiscover Publishing and Lookup Protocol, as specified in [MS-OXDSCLI]. In a public folder logon, the string to be used in this field is the value of the legacy DN attribute for the public folder mailbox that is obtained by either using the Autodiscover Publishing and Lookup Protocol or from the response of the **RopGetOwningServers** ROP (section 2.2.1.6), the **RopOpenFolder** ROP ([MS-OXCFOLD] section 2.2.1.1), or the **RopCreateFolder** ROP ([MS-OXCFOLD] section 2.2.1.2), which all return the legacy DN of the mailbox containing the public folders contents.

2.2.1.1.2 RopLogon ROP Redirect Response Buffer

The following descriptions define valid fields for the redirect response buffer of the RopLogon ROP ([MS-OXCROPS] section 2.2.3.1.4), when the value of the **ReturnValue** field is 0x00000478 (ecWrongServer).

LogonFlags: This field contains the Private, Undercover, and Ghosted flags. The server returns these flags unchanged from the **LogonFlags** field of the **RopLogon** request (section 2.2.1.1.1). The client MUST ignore all other flags.

ServerName: This field contains the **enterprise/site/server distinguished name (ESSDN)** of server for the client to connect to, as the server included in the request either no longer hosts the requested mailbox (it was moved), or was the wrong server to connect to for access to public folders. The string includes the terminating NULL character. The string length (including the terminating NULL character) MUST be equal to the value specified by the **ServerNameSize** field.

2.2.1.1.3 RopLogon ROP Success Response Buffer for Private Mailbox

The following descriptions define valid fields for the success response buffer for private folders of the **ROPLogon** ROP ([MS-OXCROPS] section 2.2.3.1.2). The following field values are included in the **RopLogon** response only when the Private bit is set in the **LogonFlags** field of the **RopLogon** request (section 2.2.1.1.1).

LogonFlags: This field is composed of the Private, Undercover, and Ghosted flags. The server returns these flags unchanged from the **LogonFlags** field of the **RopLogon** request (section 2.2.1.1.1). The client MUST ignore all other flags.

FolderIds: This field identifies the folder ID (FID) ([MS-OXCDATA] section 2.2.1.1) of all of the following folders:

- Mailbox **Root folder**. All other folders listed here are direct or indirect children of this folder.
- Deferred Action
- Spooler Queue
- Interpersonal messages subtree (Root folder of the user-visible portion of the folder hierarchy)
- Inbox
- Outbox
- Sent Items
- Deleted Items
- Common Views
- Schedule
- Search
- Views
- Shortcuts

Response Flags: This field contains flags that provide details about the state of the mailbox. Individual flag values and their meanings are specified in the following table.

Name	Value	Description
Reserved	0x01	This bit MUST be set and MUST be ignored by the client.
OwnerRight	0x02	The user has owner permission on the mailbox.
SendAsRight	0x04	The user has the right to send mail from the mailbox.
OOF	0x10	The **Out of Office (OOF)** state is set on the mailbox. For details about the OOF state, see [MS-OXWOOF].

ReplId: This field contains the short form of the value specified in the REPLGUID field, which is the **replica ID** for the logon.

ReplGuid: This field contains the GUID used to identify the source of the REPLID to REPLGUID mapping and named property mappings. If the client did not set the USE_PER_MDB_REPLID_MAPPING bit in the **OpenFlags** field, this value MUST be identical for all private mailbox logons on the same **remote procedure call (RPC)** session.

LogonTime: This field contains the **Coordinated Universal Time (UTC)** time on the server when the logon was performed. For more details about the format of this field see [MS-OXCROPS] section 2.2.3.1.2.1.

GwartTime: This field contains a numeric value that tracks the currency of the **Gateway Address Routing Table (GWART)**. This value is unique for each update of the GWART.

The client can use the value of the **GwartTime** field to determine whether the client's **address type** configuration data is current. If the most recent value of **GwartTime** matches the one that was returned on the previous logon to the mailbox, the client's address-type configuration data is up-to-date.

The client only uses the value of **GwartTime** in a comparison to detect a change; it does not interpret the value of **GwartTime** in any way.

StoreState: This field is unused and MUST be set to 0x00000000 by the server and MUST be ignored by the client.<7>

2.2.1.1.4 RopLogon ROP Success Response Buffer for Public Folders

The following descriptions define valid fields for the success response buffer for public folders of the **ROPLogon** ROP ([MS-OXCROPS] section 2.2.3.1.3). The success response buffer for public folders is sent only when the Private bit is not set in the **LogonFlags** field of the **RopLogon** request (section 2.2.1.1.1).

LogonFlags: This field is composed of the Private, Undercover, and Ghosted flags. The server returns these flags unchanged from the **LogonFlags** field of the **RopLogon** request (section 2.2.1.1.1). The client MUST ignore all other flags.

FolderIds: This field contains 13 64-bit Folder ID structures, as specified in [MS-OXCDATA] section 2.2.1.1, of all of the following folders:<8>

- Public Folders Root Folder. All other folders listed here are direct or indirect children of this folder.
- Interpersonal messages subtree
- Non-interpersonal messages subtree
- EForms Registry
- Free/Busy Data
- Offline address book Data
- EForms Registry for the user's locale
- Local Site's Free/Busy Data
- Local Site's Offline Address Book Data
- NNTP Article Index
- Empty

- Empty
- Empty

ReplId: This field contains the short form of the value specified in the **ReplGuid** field.

ReplGuid: This field contains the GUID used to identify the origin of ID and named property mappings. This value is randomly assigned to a database when it is created and is an integral part of all IDs assigned in the database. It is used in forming **LongTermID** structures, as defined in [MS-OXCDATA] section 2.2.1.3.1.

PerUserGuid: This field is not used and is ignored by the client. The server SHOULD set this field to an empty GUID (all zeroes).<9>

2.2.1.2 RopGetReceiveFolder ROP

The **RopGetReceiveFolder** ROP ([MS-OXCROPS] section 2.2.3.2) is used to determine the **Receive folder** for messages of a specific **message class**. This ROP examines the message class string and returns the Folder ID, as specified in [MS-OXCDATA] section 2.2.1.1, of the Receive folder to which messages of that class and all subclasses are delivered. This ROP also returns the specific parent message class configured to deliver to that folder.

The complete syntax of the ROP request and response buffers for this ROP is specified in [MS-OXCROPS]. This section specifies the syntax and semantics of various fields that are not fully specified in [MS-OXCROPS].

2.2.1.2.1 RopGetReceiveFolder ROP Request Buffer

The following descriptions define valid fields for the **RopGetReceiveFolder** ROP request buffer ([MS-OXCROPS] section 2.2.3.2.1). This operation is only valid when the Logon object refers to a private mailbox logon.

MessageClass: This field contains a string that specifies the message class. The string includes the terminating NULL character. Examination of the string is case-insensitive. The string MUST meet the following requirements:

- The string uses ASCII encoding.
- The length (including the terminating NULL character) is greater than zero and less than or equal to 255.
- Each character value in the string is in the numeric range of 32 to 126, inclusive.
- The string does not begin with a period (".").
- The string does not end with a period.
- The string does not contain adjacent periods.

2.2.1.2.2 RopGetReceiveFolder ROP Success Response Buffer

The following descriptions define valid fields for the **RopGetReceiveFolder** ROP success response buffer ([MS-OXCROPS] section 2.2.3.2.2).

FolderId: This field contains the **Folder Id**, as specified in [MS-OXCDATA] section 2.2.1.1, of the folder to which messages are being delivered. The folder MUST be a folder within the user's mailbox.

ExplicitMessageClass: This field contains a string specifying the message class that is actually configured for the Receive folder. The string includes the terminating NULL character, and the case of the characters in the string is insignificant. The string MUST meet the following requirements:

- The string uses ASCII encoding.
- The length (including the terminating NULL character) is greater than zero and less than or equal to 255.
- Each character value in the string is in the numeric range of 32 to 126, inclusive.
- The string does not begin with a period (".").
- The string does not end with a period.
- The string does not contain adjacent periods.

2.2.1.2.3 RopGetReceiveFolder ROP Failure Response Buffer

The syntax of the **RopGetReceiveFolder** ROP failure response buffer is specified in [MS-OXCROPS] section 2.2.3.4.1.

This protocol adds no additional field information to the **RopGetReceiveFolder** ROP failure response buffer..

2.2.1.3 RopSetReceiveFolder ROP

The **RopSetReceiveFolder** ROP ([MS-OXCROPS] section 2.2.3.3) is used to establish the Receive folder for messages of a given message class. As a result, all messages of the specified message class will be delivered to the folder identified by the specified Folder Id, as defined in [MS-OXCDATA] section 2.2.1.1.

Multiple message classes are permitted to be registered to the same Receive folder. A client can change an existing Receive folder configuration for a message class by simply issuing this ROP with a different value in the **FolderId** field.

The complete syntax of the ROP request and response buffers for this ROP is specified in [MS-OXCROPS]. This section specifies the syntax and semantics of various fields that are not fully specified in [MS-OXCROPS].

2.2.1.3.1 RopSetReceiveFolder ROP Request Buffer

This operation MUST be issued against a private mailbox logon.

The following descriptions define valid fields for the request buffer of the **RopSetReceiveFolder** ROP ([MS-OXCROPS] section 2.2.3.3.1).

FolderId: Contains the FID of the desired Receive folder for the message class and all nonspecifically configured subclasses of that class.

MessageClass: Contains the string identifying the message class whose delivery folder is being set. The string includes the terminating NULL character. The string MUST comply with all of the following restrictions:

- The string uses ASCII encoding.
- The length (including the terminating NULL character) is greater than zero and less than or equal to 255.
- Each character value in the string is in the numeric range of 32 to 126, inclusive.
- The string does not begin with a period (".").
- The string does not end with a period.

- The string does not contain adjacent periods.

2.2.1.3.2 RopSetReceiveFolder ROP Response Buffer

The syntax of the **RopSetReceiveFolder** ROP response buffer is specified in [MS-OXCROPS] section 2.2.3.3.2.

This protocol adds no additional field information to the **RopSetReceiveFolder** ROP response buffer.

2.2.1.4 RopGetReceiveFolderTable ROP

The **RopGetReceiveFolderTable** ROP ([MS-OXCROPS] section 2.2.3.4) is used to obtain a comprehensive list of all configured message classes and their associated Receive folders.

The complete syntax of the ROP request and response buffers for this ROP is specified in [MS-OXCROPS]. This section specifies the syntax and semantics of various fields that are not fully specified in [MS-OXCROPS].

2.2.1.4.1 RopGetReceiveFolderTable ROP Request Buffer

The syntax of the **RopGetReceiveFolderTable** ROP request buffer is specified in [MS-OXCROPS] section 2.2.3.4.1.

There are no explicit fields for this operation. The data to retrieve is limited to the mailbox linked to the Logon object passed as part of the normal **ROP request** process. This operation MUST be issued against a private mailbox logon.

2.2.1.4.2 RopGetReceiveFolderTable ROP Success Response Buffer

The following descriptions define valid fields for the **RopGetReceiveFolderTable** ROP success response buffer ([MS-OXCROPS] section 2.2.3.4.2).

RowCount: The number of rows in the table. The rows themselves can be returned in any order.

Rows: An array that contains the rows of the Receive folder table. Each row is returned in either a **StandardPropertyRow** structure or a **FlaggedPropertyRow** structure, both of which are specified in [MS-OXCDATA] section 2.8.1.1 and [MS-OXCDATA] section 2.8.1.2, respectively. The value of each structure's **Flag** field indicates which structure is being used: 0x00 for the **StandardPropertyRow** structure; 0x01 for the **FlaggedPropertyRow** structure.

The **ValueArray** field of either **StandardPropertyRow** or **FlaggedPropertyRow** MUST include only the following properties, in the order given, and no other properties.

1. **PidTagFolderId** property ([MS-OXPROPS] section 2.691) — A **PtypInteger64** value that specifies the folder ID (FID) ([MS-OXCDATA] section 2.2.1.1) of the Receive folder, which is the folder to which messages of the specified message class will be delivered. The Receive folder MUST be a folder that is within the user's mailbox.

2. **PidTagMessageClass** property ([MS-OXPROPS] section 2.778)— A **PtypString8** value that specifies the message class that is configured for the Receive folder. The string can be all upper case, all lower case, or as originally stored by the client. The string includes the terminating NULL character and MUST meet the following requirements:

   - The string uses ASCII encoding.
   - The length (including the terminating NULL character) is greater than zero and less than or equal to 255.
   - Each character value in the string is in the numeric range of 32 to 126, inclusive.

- The string does not begin with a period (".").
- The string does not end with a period.
- The string does not contain adjacent periods.

3. **PidTagLastModificationTime** property ([MS-OXPROPS] section 2.755) — A **PtypTime** value that specifies the time, in Coordinated Universal Time (UTC), when the server created or last modified the row in the Receive folder table.

2.2.1.4.3 RopGetReceiveFolderTable ROP Failure Response Buffer

The syntax of the **RopGetReceiveFolderTable** ROP failure response buffer is specified in [MS-OXCROPS] section 2.2.3.4.1.

This protocol adds no additional field information to the **RopGetReceiveFolderTable** ROP failure response buffer.

2.2.1.5 RopGetStoreState ROP

The **RopGetStoreState** ROP ([MS-OXCROPS] section 2.2.3.5) is used to obtain state information about the current mailbox.<10>

The complete syntax of the ROP request and response buffers for this ROP is specified in [MS-OXCROPS]. This section specifies the syntax and semantics of various fields that are not fully specified in [MS-OXCROPS].

2.2.1.5.1 RopGetStoreState ROP Request Buffer

There are no explicit fields for this operation. The data to be retrieved is limited to the mailbox that is linked to the **LogonID** that is passed as part of the ROP request, as specified in [MS-OXCROPS] section 2.2.3.5.1. This operation MUST be issued against a private mailbox logon.

2.2.1.5.2 RopGetStoreState ROP Success Response Buffer

The following descriptions define valid fields for the **RopGetStoreState** ROP success response buffer ([MS-OXCROPS] section 2.2.3.5.2).

StoreState: If the mailbox currently has any active search folders, this field MUST have the STORE_HAS_SEARCHES flag (0x01000000) set. All other bits MUST NOT be set. If the STORE_HAS_SEARCHES flag is set, the user has created one or more active searches in the message store.

2.2.1.5.3 RopGetStoreState ROP Failure Response Buffer

The syntax of the **RopGetStoreState** ROP failure response buffer is specified in [MS-OXCROPS] section 2.2.3.5.3.

This protocol adds no additional field information to the **RopGetStoreState** ROP failure response buffer.

2.2.1.6 RopGetOwningServers ROP

The **RopGetOwningServers** ROP ([MS-OXCROPS] section 2.2.3.6) is used to obtain the set of servers that host content for a replicated public folder.

This ROP is useful for a situation in which the client issues a ROP that reads content from a public folder on a specific server (**RopGetContentsTable** ([MS-OXCROPS] section 2.2.4.14),

RopOpenMessage ([MS-OXCROPS] section 2.2.6.1), and **RopCreateMessage** ([MS-OXCROPS] section 2.2.6.2) are such ROPs), and the server that receives the request does not contain a replica, resulting in a failure of the ROP with ecNoReplicaHere (0x00000468). The client can issue a **RopGetOwningServers** ROP request to obtain the set of servers that do contain a replica.

The complete syntax of the ROP request and response buffers for this ROP is specified in [MS-OXCROPS]. This section specifies the syntax and semantics of various fields that are not fully specified in [MS-OXCROPS].

2.2.1.6.1 RopGetOwningServers ROP Request Buffer

This operation SHOULD be issued against a public folders logon.<11>

The following descriptions define valid fields for the request buffer of the **RopGetOwningServers** ROP ([MS-OXCROPS] section 2.2.3.6.2).

FolderId: Contains the FID of the public folder for which to obtain the replica set server names.

2.2.1.6.2 RopGetOwningServers ROP Success Response Buffer

The following descriptions define valid fields for the success response buffer of the **RopGetOwningServers** ROP ([MS-OXCROPS] section 2.2.3.6.2).

OwningServersCount: Identifies the number of strings contained in the **OwningServers** field.

CheapServersCount: Identifies the number of entries at the front of the list that have the same lowest network cost. This value MUST be less than or equal to **OwningServersCount** and MUST be greater than zero if **OwningServersCount** is greater than zero.

OwningServers: Contains an array of null-terminated ASCII strings. Each string is the ESSDN of a public folder database that hosts an **active replica** of the content of the folder. The number of strings MUST be equal to the value specified in the **OwningServersCount** field. The entries are sorted by the server's interpretation of the network cost for each entry in the list.

2.2.1.6.3 RopGetOwningServers ROP Failure Response Buffer

The syntax of the **RopGetOwningServers** ROP failure response buffer is specified in [MS-OXCROPS] section 2.2.3.6.3.

This protocol adds no additional field information to the **RopGetOwningServers** ROP failure response buffer.

2.2.1.7 RopPublicFolderIsGhosted ROP

The **RopPublicFolderIsGhosted** ROP ([MS-OXCROPS] section 2.2.3.7) is used to obtain the replication state for a folder on the current server. Folders can exist in one of several replica states, but all states except the Active state are implementation-specific.<12>

The complete syntax of the ROP request and response buffers for this ROP is specified in [MS-OXCROPS]. This section specifies the syntax and semantics of various fields that are not fully specified in [MS-OXCROPS].

2.2.1.7.1 RopPublicFolderIsGhosted ROP Request Buffer

This operation SHOULD only be issued against a public folders logon.

The following descriptions define valid fields for the request buffer of the **RopPublicFolderIsGhosted** ROP ([MS-OXCROPS] section 2.2.3.7).

FolderId: Contains the FID of the public folder for which to obtain the ghosted state.

2.2.1.7.2 RopPublicFolderIsGhosted ROP Success Response Buffer

The following descriptions define valid fields for the success response buffer of the **RopPublicFolderIsGhosted** ROP ([MS-OXCROPS] section 2.2.3.7).

IsGhosted: Contains a Boolean value that is TRUE when the server is not an active replica of the public folder; otherwise, FALSE. Other fields are included in the response only when the **IsGhosted** field is set to TRUE.

ServersCount: Identifies the number of strings contained in the **Servers** field. This field is present if the **IsGhosted** field is set to TRUE and is not present otherwise.

CheapServersCount: Identifies the number of entries at the front of the list that have the same lowest network cost. This value MUST be less than or equal to **ServersCount** and MUST be greater than zero if **ServersCount** is greater than zero. This field is present if the **IsGhosted** field is set to TRUE and is not present otherwise.

Servers: Contains an array of null-terminated ASCII strings. Each string is the ESSDN of a public folder database that itself hosts an active replica of the content of the folder. The number of strings MUST be equal to the value specified in the **ServersCount** field. The entries are sorted by the server's interpretation of the network cost for each entry in the list.

This field is present if the **IsGhosted** field is set to TRUE and is not present otherwise.

2.2.1.7.3 RopPublicFolderIsGhosted ROP Failure Response Buffer

The syntax of the **RopPublicFolderIsGhosted** ROP failure response buffer is specified in [MS-OXCROPS] section 2.2.3.7.3.

This protocol adds no additional field information to the **RopPublicFolderIsGhosted** ROP failure response buffer.

2.2.1.8 RopLongTermIdFromId ROP

The **RopLongTermIdFromId** ROP ([MS-OXCROPS] section 2.2.3.8) is used to obtain a **LongTermID** structure, as specified in [MS-OXCDATA] section 2.2.1.3.1, given a **Folder ID** structure or **Message ID** structure, as specified in [MS-OXCDATA] section 2.2.1.1 or section 2.2.1.2.

The complete syntax of the ROP request and response buffers for this ROP is specified in [MS-OXCROPS]. This section specifies the syntax and semantics of various fields that are not fully specified in [MS-OXCROPS].

2.2.1.8.1 RopLongTermIdFromId ROP Request Buffer

The following descriptions define valid fields for the request buffer of the **RopLongTermIdFromId** ROP ([MS-OXCROPS] section 2.2.3.8.1).

ObjectId: Contains the **Folder ID** structure or **Message ID**, as specified in [MS-OXCDATA] sections 2.2.1.1 or 2.2.1.2, that is mapped to a **LongTermID**, as specified in [MS-OXCDATA] section 2.2.1.3.1. The **Folder ID** or **Message ID** is a 64-bit value composed of a 16-bit replica ID (REPLID) followed by a 48-bit global counter. The 16-bit REPLID portion of the Folder ID or Message ID MUST be a valid entry in the REPLID and REPLGUID to-and-from mapping table.

2.2.1.8.2 RopLongTermIdFromId ROP Success Response Buffer

The following descriptions define valid fields for the success response buffer of the **RopLongTermIdFromId** ROP ([MS-OXCROPS] section 2.2.3.8).

LongTermId: Contains the **LongTermID** structure, as specified in [MS-OXCDATA] section 2.2.1.3.1, that is mapped from the given **Folder ID** or **Message ID**, as specified in[MS-OXCDATA] sections 2.2.4.1 or 2.2.4.2, that is contained in the **ObjectID** field (section 2.2.1.8.1).

2.2.1.8.3 RopLongTermIdFromId ROP Failure Response Buffer

The syntax of the **RopLongTermIdFromId** ROP failure response buffer is specified in [MS-OXCROPS] section 2.2.3.8.3.

This protocol adds no additional field information to the **RopLongTermIdFromId** ROP failure response buffer.

2.2.1.9 RopIdFromLongTermId ROP

The **RopIdFromLongTermId** ROP ([MS-OXCROPS] section 2.2.3.9) is used to obtain the **Folder ID** or **Message ID**, as specified in [MS-OXCDATA] sections 2.2.4.1 or 2.2.4.2, given the **LongTermID**, as specified in [MS-OXCDATA] section 2.2.1.3.1.

The complete syntax of the ROP request and response buffers for this ROP is specified in [MS-OXCROPS]. This section specifies the syntax and semantics of various fields that are not fully specified in [MS-OXCROPS].

2.2.1.9.1 RopIdFromLongTermId ROP Request Buffer

The following descriptions define valid fields for the request buffer of the **RopIdFromLongTermId** ROP ([MS-OXCROPS] section 2.2.3.9).

LongTermId: Contains the **LongTermID**, as specified in [MS-OXCDATA] section 2.2.1.3.1, to be mapped to the **Folder ID** or **Message ID**, as specified in [MS-OXCDATA] sections 2.2.1.1 or 2.2.1.2.

2.2.1.9.2 RopIdFromLongTermId ROP Success Response Buffer

The following descriptions define valid fields for the success response buffer of the **RopIdFromLongTermId** ROP ([MS-OXCROPS] section 2.2.3.9).

ObjectId: Contains the **Folder ID** or **Message ID**, as specified in [MS-OXCDATA] sections 2.2.1.1 or 2.2.1.2, that is mapped from the given **LongTermID**, as specified in [MS-OXCDATA] section 2.2.1.3.1. The **Folder ID** or **Message ID** is a 64-bit value composed of the 16-bit replica ID (REPLID) followed by the 48-bit global counter portion of the given **LongTermID**.

2.2.1.9.3 RopIdFromLongTermId ROP Failure Response Buffer

The syntax of the **RopIdFromLongTermId** ROP failure response buffer is specified in [MS-OXCROPS] section 2.2.3.9.3.

This protocol adds no additional field information to the **RopIdFromLongTermId** ROP failure response buffer.

2.2.1.10 RopGetPerUserLongTermIds ROP

The **RopGetPerUserLongTermIds** ROP ([MS-OXCROPS] section 2.2.3.10) is used to obtain the **LongTermIDs** (as specified in [MS-OXCDATA] section 2.2.1.3.1) of folders in a public folders message store that contain per-user read/unread data identified by a REPLGUID.

The complete syntax of the ROP request and response buffers for this ROP is specified in [MS-OXCROPS]. This section specifies the syntax and semantics of various fields that are not fully specified in [MS-OXCROPS].

2.2.1.10.1 RopGetPerUserLongTermIds ROP Request Buffer

This ROP MUST be issued against a logon that was made to a private mailbox.

The following descriptions define valid fields for the request buffer of the **RopGetPerUserLongTermIds** ROP ([MS-OXCROPS] section 2.2.3.10.1).

DatabaseGuid: Identifies the replica database for which the client is querying against. This GUID is obtained from the result of a **RopLogon** (section 2.2.1.1) issued against a public message store. For more details, see section 2.2.1.1.3.

2.2.1.10.2 RopGetPerUserLongTermIds ROP Success Response Buffer

The following descriptions define valid fields for the success response buffer of the **RopGetPerUserLongTermIds** ROP ([MS-OXCROPS] section 2.2.3.10.2).

LongTermIdCount: Specifies the number of entries in the following array. This field can be set to zero.

LongTermIds: Contains an array of **LongTermID** structures, as specified in [MS-OXCDATA] section 2.2.1.3.1, of folders in the public message store for which this user has cached read/unread information. The number of items in this array MUST be the same as the value of the **LongTermIdCount** field (section 2.2.1.10.2).

2.2.1.10.3 RopGetPerUserLongTermIds ROP Failure Response Buffer

The syntax of the **RopGetPerUserLongTermIds** ROP failure response buffer is specified in [MS-OXCROPS] section 2.2.3.10.3.

This protocol adds no additional field information to the **RopGetPerUserLongTermIds** ROP failure response buffer.

2.2.1.11 RopGetPerUserGuid ROP

The **RopGetPerUserGuid** ROP ([MS-OXCROPS] section 2.2.3.11) obtains the REPLGUID of the public message store that previously provided the now cached per-user read/unread data for a specific public folder. For more details about how the client uses the **RopGetPerUserGuid** ROP, see section 3.1.4.3.

The complete syntax of the ROP request and response buffers for this ROP is specified in [MS-OXCROPS]. This section specifies the syntax and semantics of various fields that are not fully specified in [MS-OXCROPS].

2.2.1.11.1 RopGetPerUserGuid ROP Request Buffer

This ROP MUST be issued against a logon that was made to a private mailbox.

The following descriptions define valid fields for the request buffer of the **RopGetPerUserGuid** ROP ([MS-OXCROPS] section 2.2.3.11.1).

LongTermId: Contains a **LongTermID** structure ([MS-OXCDATA] section 2.2.1.3.1) that specifies the folder to query.

2.2.1.11.2 RopGetPerUserGuid ROP Success Response Buffer

The following descriptions define valid fields for the success response buffer of the **RopGetPerUserGuid** ROP ([MS-OXCROPS] section 2.2.3.11.2).

DatabaseGuid: Contains the REPLGUID of the last public folder database for which relevant read/unread information was cached.

2.2.1.11.3 RopGetPerUserGuid ROP Failure Response Buffer

The syntax of the **RopGetPerUserGuid** ROP failure response buffer is specified in [MS-OXCROPS] section 2.2.3.11.2.

This protocol adds no additional field information to the **RopGetPerUserGuid** ROP failure response buffer.

2.2.1.12 RopReadPerUserInformation ROP

The **RopReadPerUserInformation** ROP ([MS-OXCROPS] section 2.2.3.12) is used to obtain a set of change numbers, each of which is associated with a message that the user has read in a specific public folder.

When this ROP is issued against a private mailbox logon, cached data for the public folder is retrieved. When this ROP is issued against a public folders logon, the current per-user read/unread data for the public folder is retrieved. The client can use this ROP in conjunction with the **RopWritePerUserInformation** ROP, as specified in section 2.2.1.13, to synchronize per-user read/unread data for a public folder.

The complete syntax of the ROP request and response buffers for this ROP is specified in [MS-OXCROPS]. This section specifies the syntax and semantics of various fields that are not fully specified in [MS-OXCROPS].

2.2.1.12.1 RopReadPerUserInformation ROP Request Buffer

The following descriptions define valid fields for the request buffer of the **RopReadPerUserInformation** ROP ([MS-OXCROPS] section 2.2.3.12.1).

FolderId: Contains a **LongTermID** structure ([MS-OXCDATA] section 2.2.1.3.1) that specifies the folder to query.

Reserved: This field MUST be set to 0x00 and is ignored by the server.

DataOffset: Identifies the offset into the stream of data. This value is the position of the first byte of data to be returned. The value MUST be greater than or equal to zero.

The client MUST NOT set **DataOffset** to an arbitrary value. The value MUST be zero in the first **RopReadPerUserInformation** request ([MS-OXCROPS] section 2.2.3.12.1). If subsequent requests are necessary to retrieve all the data, then the client MUST update **DataOffset** by adding to it the value of the **DataSize** field of the previous **RopReadPerUserInformation** response ([MS-OXCROPS] section 2.2.3.12.2). In other words, if **HasFinished** equals FALSE, then **DataOffset** MUST be updated, as follows, after each **RopReadPerUserInformation** response.

```
DataOffset = DataOffset + DataSize
```

MaxDataSize: Specifies the maximum amount of data to be returned to the client in a single **RopReadPerUserInformation** response ([MS-OXCROPS] section 2.2.3.12.2). The client can set **MaxDataSize** to zero, which indicates to the server that a default value MUST be used as the maximum size. When multiple **RopReadPerUserInformation** requests ([MS-OXCROPS] section 2.2.3.12.1) are necessary to retrieve all of the data, the client can set **MaxDataSize** to a different value in each invocation of the ROP.

2.2.1.12.2 RopReadPerUserInformation ROP Success Response Buffer

The following descriptions define valid fields for the success response buffer of the **RopReadPerUserInformation** ROP ([MS-OXCROPS] section 2.2.3.12.2).

HasFinished: Indicates whether the last block of data is being returned. The value of this field is TRUE if the last block of data is being returned and FALSE otherwise. This field MUST be set to TRUE if the underlying data has not changed since the last successful retrieval of per-user read/unread data, as specified in section 3.1.5.2.

DataSize: Contains the size, in bytes, of the data being returned. The value of this field MUST be less than or equal to the value of the **MaxDataSize** field of the request. This value MUST be zero if the underlying data has not changed since the last successful retrieval of per-user read/unread data, as specified in section 3.1.5.2.

Data: Contains the **change number set (CNSET)**, which is serialized into a **binary large object (BLOB)**. The format of the BLOB is the same as that of a serialized identifier set, which is specified in [MS-OXCFXICS] section 2.2.2.4.2. The size of the BLOB MUST be equal to the value specified in the **DataSize** field. The client is not expected to interpret this data in any way, but simply provide it unaltered in a future sequence of invocations of **RopWritePerUserInformation** ([MS-OXCROPS] section 2.2.3.13).

2.2.1.12.3 RopReadPerUserInformation ROP Failure Response Buffer

The syntax of the **RopReadPerUserInformation** ROP failure response buffer is specified in [MS-OXCROPS] section 2.2.3.12.3.

This protocol adds no additional field information to the **RopReadPerUserInformation** ROP failure response buffer.

2.2.1.13 RopWritePerUserInformation ROP

The **RopWritePerUserInformation** ROP ([MS-OXCROPS] section 2.2.3.13) is used to establish the set of change numbers of messages the user has read in a specific public folder.

When this ROP is issued against a private mailbox logon, data for the public folder is saved. When this ROP is issued against a public folders logon, the current per-user read/unread data for the public folder is established. The client can use this ROP in conjunction with **RopReadPerUserInformation**, which is specified in section 2.2.1.12, to synchronize per-user read/unread data for a public folder.

The complete syntax of the ROP request and response buffers for this ROP is specified in [MS-OXCROPS]. This section specifies the syntax and semantics of various fields that are not fully specified in [MS-OXCROPS].

2.2.1.13.1 RopWritePerUserInformation ROP Request Buffer

The following descriptions define valid fields for the request buffer of the **RopWritePerUserInformation** ROP ([MS-OXCROPS] section 2.2.3.13.1).

FolderId: Contains a **LongTermID** structure ([MS-OXCDATA] section 2.2.1.3.1) that specifies the folder for which data is being saved.

HasFinished: Indicates whether the **Data** field contains the last block of data to be written. If the **Data** field contains the last block of data to be written, this field is set to TRUE; otherwise, this field is set to FALSE.

DataOffset: Identifies the offset into the stream where this block of data is to be written. This value MUST be equal to the total size of the data previously written.

DataSize: Specifies the size, in bytes, of the data to be written.

Data: Contains the CNSET to be written. The CNSET is the one received in the **Data** field of the **RopReadPerUserInformation** response. The size of the data MUST be equal to the value specified in the **DataSize** field.

ReplGuid: Identifies which public database was the source of the data. The value is the REPLGUID of the last database for which relevant read/unread information was obtained. This GUID is obtained from the result of a **RopLogon** (section 2.2.1.1) issued against a public message store. For more details, see section 2.2.1.1.4. This field MUST NOT be present for operations against public folders logons. This field MUST be present when the value of the **DataOffset** field is zero. This field MUST NOT be present when **DataOffset** is not zero.

2.2.1.13.2 RopWritePerUserInformation ROP Response Buffer

There are no fields other than the **ReturnValue** field for the request buffer of the **RopWritePerUserInformation** ROP ([MS-OXCROPS] section 2.2.3.13.2).

2.2.2 Logon-Specific Properties

The following properties are available on Logon objects. A Logon object is obtained by issuing a **RopLogon** request (section 2.2.1.1.1), and receiving a successful response. Some logon properties are read-only. Some logon properties are write-only. Some properties can be deleted by the client. Some properties are available only on public folder logons. Some properties are available only on private mailbox logons.

To read any of the readable properties, the client issues a **RopGetPropertiesSpecific** ROP with the Logon object obtained from a successful invocation of **RopLogon** (section 2.2.1.1). To write any of the writable properties, the client issues a **RopSetProperties** ROP with the Logon object obtained from a successful invocation of **RopLogon**. To delete any of the deletable properties, the client issues **RopDeleteProperties** with the Logon object obtained from a successful invocation of **RopLogon**. For more details about **RopSetProperties**, **RopGetPropertiesSpecific**, or **RopDeleteProperties** see [MS-OXCROPS] and [MS-OXCPRPT].

2.2.2.1 Private Mailbox Logon Properties

The following sections specify the properties that are available on a private mailbox logon.

2.2.2.1.1 Read-Only Properties

The read-only properties that are available on a private mailbox logon are specified in section 2.2.2.1.1.1 through section 2.2.2.1.1.14.

2.2.2.1.1.1 PidTagExtendedRuleSizeLimit Property

Type: **PtypInteger32** ([MS-OXCDATA] section 2.11.1)

The **PidTagExtendedRuleSizeLimit** property ([MS-OXPROPS] section 2.685) contains the Maximum size, in bytes, the user is allowed to accumulate for a single "extended" rule. For details of extended rules, see [MS-OXORULE] section 2.2.4.

2.2.2.1.1.2 PidTagMaximumSubmitMessageSize Property

Type: **PtypInteger32** ([MS-OXCDATA] section 2.11.1)

The **PidTagMaximumSubmitMessageSize** property ([MS-OXPROPS] section 2.772) contains the maximum size, in kilobytes, of a message a user is allowed to submit for transmission to another user. An unset value or a value of -1 indicates that there is no limit.

2.2.2.1.1.3 PidTagProhibitReceiveQuota Property

Type: **PtypInteger32** ([MS-OXCDATA] section 2.11.1)

The **PidTagProhibitReceiveQuota** property ([MS-OXPROPS] section 2.865) contains the maximum size, in kilobytes, a user is allowed to accumulate in their mailbox, before no further mail will be delivered. An unset value or a value of -1 indicates that there is no limit.

2.2.2.1.1.4 PidTagProhibitSendQuota Property

Type: **PtypInteger32** ([MS-OXCDATA] section 2.11.1)

The **PidTagProhibitSendQuota** property ([MS-OXPROPS] section 2.866) contains the maximum size, in kilobytes, a user is allowed to accumulate in their mailbox, before the user can no longer submit any more mail. An unset value or a value of -1 indicates that there is no limit.

2.2.2.1.1.5 PidTagStoreState Property

Type: **PtypInteger32** ([MS-OXCDATA] section 2.11.1)

The **PidTagStoreState** property ([MS-OXPROPS] section 2.1019) indicates whether the mailbox has any active search folders. The value FALSE indicates that the mailbox does not have active search folders.

2.2.2.1.1.6 PidTagContentCount Property

Type: **PtypInteger32** ([MS-OXCDATA] section 2.11.1)

The **PidTagContentCount** property ([MS-OXPROPS] section 2.637) contains the cumulative count of non-**folder associated information (FAI)** messages in the mailbox.

2.2.2.1.1.7 PidTagMailboxOwnerEntryId Property

Type: **PtypBinary** ([MS-OXCDATA] section 2.11.1)

The **PidTagMailboxOwnerEntryId** property ([MS-OXPROPS] section 2.768) contains the **EntryID** in the **Global Address List (GAL)** of the owner of the mailbox.

2.2.2.1.1.8 PidTagMailboxOwnerName Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagMailboxOwnerName** property ([MS-OXPROPS] section 2.769) contains the display name of the owner of the mailbox.

2.2.2.1.1.9 PidTagMessageSize Property

Type: **PtypInteger32** ([MS-OXCDATA] section 2.11.1)

The **PidTagMessageSize** property ([MS-OXPROPS] section 2.787) contains the cumulative size, in bytes, of all content in the mailbox. Value is limited to 32 bits and becomes undefined if the content size exceeds 4 gigabytes.

2.2.2.1.1.10 PidTagMessageSizeExtended Property

Type: **PtypInteger64** ([MS-OXCDATA] section 2.11.1)

The **PidTagMessageSizeExtended** property ([MS-OXPROPS] section 2.788) contains the cumulative size, in bytes, of all content in the mailbox.

2.2.2.1.1.11 PidTagUserEntryId Property

Type: **PtypBinary** ([MS-OXCDATA] section 2.11.1)

The **PidTagUserEntryId** property ([MS-OXPROPS] section 2.1043) contains the Address book EntryID of the user logged on to the mailbox.

2.2.2.1.1.12 PidTagLocaleId Property

Type: **PtypInteger32** ([MS-OXCDATA] section 2.11.1)

The **PidTagLocaleId** property ([MS-OXPROPS] section 2.765) establishes the language locale for translating system-generated messages, such as delivery reports. For more details, see [MS-LCID].

2.2.2.1.1.13 PidTagSortLocaleId Property

Type: **PtypInteger32** ([MS-OXCDATA] section 2.11.1)

The **PidTagSortLocaleId** property ([MS-OXPROPS] section 2.1011) establishes the language locale for sorting the contents of tables. For more details, see [MS-LCID]. For additional information about this property, see section 3.2.5.1.1.

2.2.2.1.1.14 PidTagCodePageId Property

Type: **PtypInteger32** ([MS-OXCDATA] section 2.11.1)

The **PidTagCodePageId** property ([MS-OXPROPS] section 2.627) establishes the client code page for **Unicode** to **double-byte character set (DBCS)** string conversion. For details, see [MS-UCODEREF].

2.2.2.1.2 Read/Write Properties

The read/write properties that are available on a private mailbox logon are specified in section 2.2.2.1.2.1 through section 2.2.2.1.1.13.

2.2.2.1.2.1 PidTagComment Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagComment** property ([MS-OXPROPS] section 2.628) contains a mailbox comment.<13>

2.2.2.1.2.2 PidTagDeleteAfterSubmit Property

Type: **PtypBoolean** ([MS-OXCDATA] section 2.11.1)

The **PidTagDeleteAfterSubmit** property ([MS-OXPROPS] section 2.659) indicates whether a transport deletes all submitted mail after transmission. An unset value or a value of FALSE indicates that the mail is not deleted. The client can also delete this property.<14>

2.2.2.1.2.3 PidTagDisplayName Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagDisplayName** property ([MS-OXCFOLD] section 2.2.2.2.2.5) contains the mailbox display name.<15>

2.2.2.1.2.4 PidTagOutOfOfficeState Property

Type: **PtypBoolean** ([MS-OXCDATA] section 2.11.1)

The **PidTagOutOfOfficeState** property ([MS-OXPROPS] section 2.846) indicates whether the user is Out of Office (OOF). The value TRUE indicates that the user is OOF, in which case the out of office rules are evaluated and executed. When the value of this property is reset, regardless of the value,

the accumulated OOF history is cleared for all OOF rules. For more details about rules, see [MS-OXORULE].

2.2.2.1.2.5 PidTagSentMailSvrEID Property

Type: **PtypServerId** ([MS-OXCDATA] section 2.11.1)

The **PidTagSentMailSvrEID** property ([MS-OXPROPS] section 2.1002) contains the structure identifying the Sent Items folder. An unset value indicates that the server won't move sent items to a Sent Items folder after transmission. The client can also delete this property.

2.2.2.2 Public Folders Logon Properties

The properties that are available on a public folders logon are specified in section 2.2.2.2.1 and section 2.2.2.2.2. These properties are read only.

2.2.2.2.1 PidTagUserEntryId Property

Type: **PtypBinary** ([MS-OXCDATA] section 2.11.1)

The **PidTagUserEntryId** property ([MS-OXPROPS] section 2.1043) contains the Address book EntryID of the user logged on to the public folder. This property is also available on private mailbox logon (section 2.2.2.1.1.11).

2.2.2.2.2 PidTagAddressBookMessageId Property

Type: **PtypInteger64** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookMessageId** property ([MS-OXPROPS] section 2.542) contains the short-term MID of the first message in the local site's offline address book public folder, if it exists and has a **local replica**. The property MUST have an error value of ecNotFound (0x8004010F) if there is no local site offline address book public folder, the server can't open the folder, the server can't access the message, or there is no local replica of the folder.

3 Protocol Details

3.1 Client Details

3.1.1 Abstract Data Model

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

cache of REPLID / REPLGUID mapping: The client maintains a cache of the mapping between REPLIDs and REPLGUIDs used for **Folder IDs** or **Message IDs**, as specified in [MS-OXCDATA] sections 2.2.1.1 or 2.2.1.2 and **LongTermIDs**, as specified in [MS-OXCDATA] section 2.2.1.3.1.

cache of per-user data: The client maintains a cache of per-user data currently stored in the private message store. This enables the client to sync per-user data only when a change has been made.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

3.1.4.1 Logging on to a Store

When the user opens the client application, the client establishes a Session Context with the server, either by using the **EcDoConnectEx** method, as is specified in [MS-OXCRPC] section 3.1.4.1, or by using the **Connect** request type<16>, as specified in [MS-OXCMAPIHTTP] section 2.2.4.1. Once the client has successfully connected to the server, the client logs on to the message store by sending a **RopLogon** request (section 2.2.1.1.1). When the client sends **RopLogon**, the client MUST specify a **LogonID** to be used in the ROP request buffer. For more details about logging on and the **LogonID**, see [MS-OXCROPS] section 3.1.4.2

After successfully logging on, the client SHOULD cache the REPLGUID. In some cases, the client will have to re-attempt the logon. For more details, see section 3.1.5.1. The client cannot attempt any additional ROPs until it successfully logs on to the message store.

3.1.4.2 Converting Between LongTermIDs and Folder or Message IDs

When the client needs to persist an ID across logon sessions, the client MUST first convert the **Folder ID** or **Message ID**, as specified in [MS-OXCDATA] sections 2.2.1.1 or 2.2.1.2 to a **LongTermID**, as specified in [MS-OXCDATA] section 2.2.1.3.1, by using **RopLongTermIdFromId** (section 2.2.1.8). **Folder IDs** and **Message IDs** use a 16-bit REPLID in place of a REPLGUID. **Folder IDs** and **Message IDs** MUST NOT be persisted in any storage that could be accessed on a different logon session for the same mailbox. Any **Folder ID** or **Message ID** cached in non-persistent storage MUST be forgotten (deleted from non-persistent storage) if the client reconnects to the server, issues a **RopLogon** (section 2.2.1.1), and the return value of REPLGUID is different from the value obtained from a

previous **RopLogon**. To persist IDs in any long-term storage, the client MUST first convert the ID to a **LongTermID**.

When the client needs to specify a **Folder ID** or **Message ID** in a ROP request, the client uses **RopIdFromLongTermId** (section 2.2.1.9) to convert a **LongTermID** into a **Folder ID** or **Message ID**. Most ROPs that take IDs require the **Folder ID** or **Message ID**.

3.1.4.3 Syncing Per-User Read/Unread Data for Public Folders

When the user marks an item as read/unread, switches to a different public folder, or logs off from the message store, the client synchronizes the per-user read/unread data for the public folder. Other high-level events, as determined by the implementer, can trigger a synchronization.

Public folder data is replicated across multiple servers, with each server maintaining per-user read/unread data for each public folder. The read/unread information is valid for that server only. If a subsequent logon results in the client being redirected to a different replica server, it is the client's responsibility to synchronize the current read/unread data to the new server.

For each public folder, the client issues a **RopReadPerUserInformation** (section 2.2.1.12) against the public message store (this is not necessary if the public folder has not been modified) to retrieve the per-user read/unread data. This data is then stored in the private message store by using **RopWritePerUserInformation** (section 2.2.1.13).

When a public folder is subsequently reopened in a later logon session, the client MUST check to see if the replica server has changed. This is done by issuing a **RopGetPerUserGuid** (section 2.2.1.11) against the private message store and comparing the REPLGUID returned (in the **DatabaseGuid** field) to the public message store REPLGUID, which is returned by **RopLogon** (in the **ReplGuid** field of the public folders logon response) (section 2.2.1.1). If the REPLGUIDs match (or **RopGetPerUserGuid** doesn't find the REPLGUID), then the public folder is in sync. If the REPLGUIDs don't match, the client MUST synch the read/unread data from the private message store up to the public message store. This is done in the reverse manner as the previous sync: the data is retrieved from the private message store by using **RopReadPerUserInformation** and sent to the public message store using **RopWritePerUserInformation**.

When synching using **RopReadPerUserInformation** and **RopWritePerUserInformation**, it is important to note that the size of the return data potentially exceeds the maximum amount of data that can be communicated in a single ROP. For this reason, the operation is designed to stream the data to the client by having the client invoke these ROPs multiple times. For details, see section 3.1.5.2.

3.1.4.4 Registering for Notifications

When the user opens the client application, the client registers to receive notifications for a message store by using the Core Notifications protocol, as specified in [MS-OXCNOTIF]. The notifications for which the client registers are determined by the implementer. The various events for which the server sends a notification are listed in [MS-OXCNOTIF] section 2.2.1.1.

3.1.5 Message Processing Events and Sequencing Rules

3.1.5.1 Logon Failure or Connection Failure

If the server returns the value ecWrongServer (section 3.2.5.1.3) in the **ReturnValue** field of the **RopLogon** redirect response (section 2.2.1.1.2), then the client SHOULD create a new Session Context with the server that is specified by the **ServerName** field in the ROP response. Using that connection, the client then re-attempts the logon. For more details about creating a new Session Context, see [MS-OXCRPC] and [MS-OXCMAPIHTTP]. For more details about logging on to a message store, see section 3.1.4.1.

If the server returns either ecUnknownUser (section 3.2.5.1.3) or ecLoginFailure (section 3.2.5.1.3) in the **ReturnValue** field of the **RopLogon** ROP redirect response (section 2.2.1.1.2), then the client SHOULD use the Autodiscover HTTP Service protocol, as specified in [MS-OXDISCO], to attempt to retrieve updated user and server information. If successful, the client attempts to log on again by releasing the previous Session Context and creating a new Session Context with the information supplied by the Autodiscover HTTP Service protocol. If the client is not successful at retrieving updated information or if no changes are detected, then the client MUST fail the logon.

If the server returns the value ecMailboxInTransit (section 3.2.5.1.3) in the **ReturnValue** field of the **RopLogon** redirect response (section 2.2.1.1.2), then the mailbox being moved is locked for client access for the entire duration of the move. Any attempt to establish a new Store object for such a mailbox results in the ecMailboxInTransit error code.

If the client is unable to establish a Session Context to a public folder message store, then it can request a redirection to an alternative public folder message store from the private message store. The client can use an existing Session Context to the private message store, or create a new one. To request a redirection to an alternative public folder message store, the client issues a **RopLogon** request (section 2.2.1.1.1) to the private message store. The client MUST set the ALTERNATE_SERVER flag in the **OpenFlags** field of the **RopLogon** request. The logon request returns ecWrongServer and redirects the client to an alternate server. When issuing the logon request to the alternate server, the client MUST clear the ALTERNATE_SERVER flag and set the IGNORE_HOME_MDB flag in the **OpenFlags** field.

If the RPC session to the server is lost and then reconnected, then the existing logon is invalid. The client MUST log on again by calling **RopLogon** (the client can reuse the existing **LogonID**). Additionally, all objects (folders, messages, and tables) that were opened on the original logon are now invalid and MUST be re-opened. The new REPLGUID returned by **RopLogon** MUST be compared to the cached value. If the GUIDs are different, then the client MUST dispose of all local caches of server information. This includes any open **Server objects**, caches of data mappings, or caches of special FIDs. The effect MUST be similar to actually exiting the client application and restarting from the beginning of the process.

3.1.5.2 Streaming of Per-User Read/Unread Data

When synchronizing the per-user read/unread data for a public folder, the size of the returned data can exceed the maximum amount of data that can be sent in a single ROP. For this reason, **RopReadPerUserInformation** (section 2.2.1.12) and **RopWritePerUserInformation** (section 2.2.1.13) are designed to stream the data by means of multiple invocations until all data is received or sent. The client MUST complete the streaming of data for one public folder before commencing streaming operations for another public folder on the same server logon.

RopReadPerUserInformation Streaming

If the **HasFinished** field of the **RopReadPerUserInformation** response is set to FALSE, indicating that there is more data to be retrieved for the public folder, the client sends another **RopReadPerUserInformation** request. The client continues to send **RopReadPerUserInformation** requests until all data is retrieved from the server.

If multiple requests are necessary to retrieve all of the data, the client MUST specify an updated value in the **DataOffset** field of the next request so that **DataOffset** always points to the first byte of the next block of data to be returned. The new value is equal to the sum of the value used in the previous **RopReadPerUserInformation** request and the value of the **DataSize** field of the previous **RopReadPerUserInformation** response. In other words, if **HasFinished** is set to FALSE, **DataOffset** is set as follows in the next **RopReadPerUserInformation** request.

DataOffset = DataOffset + DataSize

If multiple requests are necessary to retrieve all of the data, the **MaxDataSize** field can be set to a different value in each invocation of the ROP. This is completely at the client's discretion.

The per-user read/unread data has been completely retrieved when the **HasFinished** field of the response is set to TRUE. The client MUST NOT interpret the value of the **Data** field of the **RopReadPerUserInformation** response. The client simply provides the value unaltered in a future sequence of invocations of **RopWritePerUserInformation**.

RopWritePerUserInformation Streaming

The client sends the data as it was received in the **RopReadPerUserInformation** response. The client continues to send **RopWritePerUserInformation** requests until all of the data is sent to the server. The client sets **HasFinished** to TRUE when the last block of data is sent.

3.1.6 Timer Events

None

3.1.7 Other Local Events

None

3.2 Server Details

3.2.1 Abstract Data Model

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

The server maintains several tables of data in order to satisfy the various ROPs that a client can invoke. These tables include: a REPLID and REPLGUID to-and-from mapping table, named property-to-property ID mapping table, a mailbox table, a per-user data table, and a Receive folder table.

REPLID/REPLGUID mapping table: The REPLID and REPLGUID to-and-from mapping table contains rows of 16-bit REPLID values coupled with 128-bit REPLGUID values. This table is used to map a REPLID to a REPLGUID and vice versa. When a client invokes **RopIdFromLongTermId** (section 2.2.1.9)**,** this table is searched for the REPLGUID portion of the **LongTermID**, as specified in [MS-OXCDATA] section 2.2.1.3.1, passed by the client. When a client invokes **RopLongTermIdFromId** (section 2.2.1.8), this table is searched for the REPLID portion of the **Folder ID** or **Message ID**, as specified in [MS-OXCDATA] sections 2.2.1.1 or 2.2.1.2, passed by the client.

named property/property ID mapping table: The named property-to-property ID mapping table contains the mapping between registered named properties and their server-assigned property identifiers. The server uses this table to look up the named property for a given property ID and vice versa when processing **RopGetNamesFromPropertyIds** and **RopGetPropertyIdsFromNames**, as specified in [MS-OXCPRPT] section 3.2.5.9 and [MS-OXCPRPT] section 3.2.5.10

mailbox table: The mailbox table is used for logging on to a private mailbox. The table contains one row for each mailbox in the database. It contains columns to specify the root folder and other **special folders** of the mailbox, the access permissions to the mailbox, and an identifying GUID that matches the owner of the mailbox, along with other metadata, such as last logon time, various item counts and aggregate sizes within the mailbox, and so on. When a client invokes **RopLogon** (section 2.2.1.1), the client passes an identifying moniker for the mailbox. The server then looks up the moniker in a global directory. The entry in the global directory indicates the proper server to log on to for this user's mailbox, and contains other relevant data used to find the mailbox on that server in the mailbox table. The proper row in the mailbox table is then found, and the user's access is checked. If the logon is

allowed, then the Folder IDs of various special folders are obtained from the table and returned to the client. For more details about special folders, see [MS-OXOSFLD]. A list of the folders that are special folders is provided in [MS-OXOSFLD] section 1.3.

per-user data table: The per-user data table contains the read/unread information for various public folders on a specific public folder replica server. The table maintains the mailbox GUID, the REPLGUID, the folder, and the change number set of read items. The **RopGetPerUserLongTermIds** ([MS-OXCROPS] section 2.2.3.10), **RopGetPerUserGuid** ([MS-OXCROPS] section 2.2.3.11), **RopReadPerUserInformation** ([MS-OXCROPS] section 2.2.3.12), **RopWritePerUserInformation** ([MS-OXCROPS] section 2.2.3.13) ROPs each make use of the data in this table.

Receive folder table: The Receive folder table contains rows of message class strings and associated FIDs. Each FID specifies a Receive folder. The server maintains a single Receive folder table per database. The data within the table are scoped to each mailbox. Conceptually, there is a single Receive folder table per mailbox. The delivery process uses the message class string on the incoming e-mail to look up the appropriate folder to which to deliver that message. The **RopGetReceiveFolder** ([MS-OXCROPS] section 2.2.3.2) and **RopSetReceiveFolder** ([MS-OXCROPS] section 2.2.3.3) ROPs are used to retrieve and set the data in this table.

3.2.2 Timers

None.

3.2.3 Initialization

When a database is created, the database MUST be assigned a new randomly generated REPLGUID. When the REPLID and REPLGUID to-and-from mapping table is created, a single new entry MUST be added, consisting of the database REPLGUID and a newly assigned REPLID.

When a database is restored from backup, the server MUST take steps to ensure that it does not re-issue a REPLGUID that was issued prior to the restoration. These steps are implementation-specific. <17>

When a mailbox is created, the following entries MUST be added to the Receive folder table for the new mailbox:

- "" (empty string) – Inbox in the new mailbox
- "IPM" – Inbox in the new mailbox
- "Report.IPM" – Inbox in the new mailbox
- "IPC" – Root folder of the new mailbox

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

Except for **RopLogon** (section 2.2.1.1), all ROPs listed in the following sections have the client prerequisite of successfully completing a **RopLogon** operation. **RopLogon** requires that the client has successfully connected to the server by establishing a Session Context via either the **EcDoConnectEx** method, as specified in [MS-OXCRPC] section 3.1.4.1, or the **Connect** request type,<18> as specified in [MS-OXCMAPIHTTP] section 2.2.4.1.

3.2.5.1 Receiving a RopLogon ROP Request

If the **LogonFlags** field has the Private bit set, the logon is going to a private mailbox. Otherwise, the logon is going to the public folders.

3.2.5.1.1 Private Mailbox Logon

Look up the ESSDN (specified in the **Essdn** field of the request) in the global directory to get that user's configuration information. If lookup fails specifically because the ESSDN could not be found, the server MUST fail the operation with a **ReturnValue** of ecUnknownUser (section 3.2.5.1.3). If lookup fails for any other reason, the server MUST fail the operation with a **ReturnValue** of ecLoginFailure, as specified in section 3.2.5.1.3.

If the user has no configured mailbox database, the ROP MUST fail with a **ReturnValue** of ecUnknownUser, as specified in section 3.2.5.1.3. If the database indicated by the user's configured mailbox database is currently offline, the operation SHOULD<19> fail with a **ReturnValue** of ecLoginFailure. If the client attempts to log on to a mailbox that is in transit, the server MUST fail the operation with ecMailboxInTransit, as specified in section 3.2.5.1.3, in the **ReturnValue** field. If the client attempts to log on to a mailbox that is disabled, the server SHOULD<20> fail the operation with ecMailboxDisabled, as specified in section 3.2.5.1.3; ecLoginFailure; or ecUnknownUser in the **ReturnValue** field.

If the user's configured mailbox is not hosted by this server, the server determines the name of the correct server hosting the user's mailbox and fail the ROP with a **ReturnValue** of ecWrongServer, as specified in section 3.2.5.1.3. For details about properly forming the response when a **ReturnValue** of ecWrongServer is sent, see section 2.2.1.1.2.

If the client attempts to log on to a nonhome message store and the USE_ADMIN_PRIVILEGE bit in the **OpenFlags** field is not set, the server MUST fail the operation with ecProfileNotConfigured, as specified in section 3.2.5.1.3, in the **ReturnValue** field.

If the client specified an invalid code page for a string or a Server object, the server SHOULD<21> fail the operation with ecUnknownCodePage, as specified in section 3.2.5.1.3, in the **ReturnValue** field.

If the client has made more than five attempts within a 10-second period to log on to a mailbox that is not hosted on the server, the server SHOULD<22> fail the operation with ecServerPaused, as specified in section 3.2.5.1.3, in the **ReturnValue** field.

If the client sets an undefined flag in either the **LogonFlags** field or the **OpenFlags** field, the server SHOULD<23> fail the operation with ecError, as specified in [MS-OXCDATA] section 2.4, in the **ReturnValue** field.

If the SUPPORT_PROGRESS flag is set, the server responds as specified in [MS-OXCMSG] section 3.2.5.10.<24>

If the **PidTagSortLocaleId** property (section 2.2.2.1) is set, then the server supports the GET/read operation for the property.<25> If the property is not set, then the server returns ecNotFound, as specified in [MS-OXCDATA] section 2.4, in the **ReturnValue** field.

For the USE_PER_MDB_REPLID_MAPPING flag of the **OpenFlags** field, the server SHOULD<26> have the following behavior:

- If the logon is the first on the RPC session, or if the logon is additional on the RPC session and it is to the same mailbox that is associated with the first logon, then the server ignores the USE_PER_MDB_REPLID_MAPPING flag of the **OpenFlags** field.<27>
- If the logon is additional on the RPC session, and it is to a mailbox that is different from the mailbox that is associated with the first logon, then the server inspects the USE_PER_MDB_REPLID_MAPPING flag of the **OpenFlags** field to see if it is set. If the USE_PER_MDB_REPLID_MAPPING flag is not set, then the server SHOULD<28> fail the ROP with

a **ReturnValue** of ecInvalidParameter 0x80070057.<29> If the USE_PER_MDB_REPLID_MAPPING flag is set, then the server takes no action.<30>

If the user does not match the owner of the mailbox, then the return value is implementation-specific;<31> otherwise, the server checks the user's permissions to determine whether the user is an owner of the mailbox or a delegate. An owner is not required to have security settings checked before performing any non-administrative operations on the mailbox. For more details about delegates, see [MS-OXODLGT].

The server then finds the mailbox in the mailbox table. If the mailbox is not present in the table and the user has owner permission on the mailbox, the server creates the mailbox. That process includes creating the default folders and establishing the proper Receive folder values. For details about setting Receive folder values, see section 3.2.3. The server does not create the mailbox if the user does not have owner permission. In that case, the ROP fails with a **ReturnValue** of ecLoginPerm, as specified in section 3.2.5.1.3 or a **ReturnValue** of ecAccessDenied, as specified in section 3.2.5.1.3; the return value is implementation specific.<32> Other failures to find the user in the mailbox table (beyond a "not found" error) MUST fail the operation with a **ReturnValue** of ecLoginFailure.

The server then determines the appropriate FIDs to return to the client. For details, see section 2.2.1.1.3. The server returns the appropriate REPLGUID in the **ReplGuid** field. If the server returns the same REPLGUID for different logons, the server MUST use the same REPLID-to-REPLGUID mapping and named property-to-property ID mapping for those different logons.

3.2.5.1.2 Public Folders Logon

The server confirms that the user logging on to the public folders has a mailbox in the organization. The server performs the following:

1. Determine the mailbox database hosting the connected user's mailbox. The user is determined from the underlying Session Context. The **Essdn** field specifies a mailbox to log on to for private mailbox logons (see section 2.2.1.1.1).

2. Determine from the global directory (for that mailbox container) the preferred public folder container to use (database or mailbox).<33>

3. Determine the server that the public folder container is hosted on, by using the Autodiscover Publishing and Lookup Protocol, as specified in [MS-OXDSCLI]. If the USE_AUTODISCOVER_FOR_PUBLIC_FOLDER_CONFIGURATION flag is set in the AUX_EXORGINFO auxiliary block, as specified in [MS-OXCRPC] section 2.2.2.2.17, then the client SHOULD<34> configure public folders via autodiscover.

If the client sets an undefined flag in either the **LogonFlags** field or the **OpenFlags** field, the server SHOULD<35> fail the operation with 0x80004005 (ecError) in the **ReturnValue** field of the response.

The **OpenFlags** field MUST have the PUBLIC bit set to log on to the public folder container (for either database or mailbox).

If the **OpenFlags** field has the ALTERNATE_SERVER bit set, the server searches for another public folder container server in the organization which is not the configured preferred server. The process by which another public folder container is chosen is up to the implementation. If a suitable server cannot be found, the operation MUST fail with a **ReturnValue** of 0x80040111 (ecLoginFailure). Otherwise, the operation MUST fail with a **ReturnValue** of 0x00000478 (ecWrongServer). For more details about properly forming the return values when a **ReturnValue** of 0x00000478 is sent, see section 2.2.1.1.2.

If the **OpenFlags** field has either the ALTERNATE_SERVER bit or the IGNORE_HOME_MDB bit set, the server ignores the Ghosted bit of the **LogonFlags** field. Otherwise, the server has the following behavior: If the Ghosted bit of the **LogonFlags** field is set, then the server uses the public folder container that is present on the server. If there is no public folder container on the server, then the

server responds with the **ReturnValue** field set to 0x80040111 (ecLoginFailure). If the Ghosted bit is not set, the server uses the default public folder container for the logon. If the server does not host that container, the server MUST fail the operation with a **ReturnValue** of 0x00000478 (ecWrongServer). For details about properly forming the response when a **ReturnValue** of 0x00000478 is sent, see section 2.2.1.1.2.

If this server doesn't host a public folder container at all, or the container is not presently accessible, the operation MUST fail with a **ReturnValue** of 0x80040111 (ecLoginFailure).

The server then determines the appropriate FIDs to return to the user. For more details, see section 2.2.1.1.4.

The server is now ready to accept further ROP commands from the client on behalf of this logon session.

3.2.5.1.3 RopLogon ROP Common Return Codes

The following specific error codes apply to the **RopLogon** ROP. Other possible error codes are specified in [MS-OXCDATA] section 2.4.

Name	Value	Meaning
ecNone	0x00000000	Success.
ecLoginFailure	0x80040111	A login failure occurred.
ecUnknownUser	0x000003EB	The user that is specified by the **Essdn** field is unknown to the system.<36>
ecUnknownCodePage	0x000003EF	The **code page** for this session is unknown.
ecMailboxDisabled	0x0000096C	The user account is marked as disabled.
ecMailboxInTransit	0x0000050C	The mailbox is in transit; logon is not allowed
ecWrongServer	0x00000478	The requested message store for logon is not the user's home message store.
ecInvalidParameter	0x80070057	The client has not set the USE_PER_MDB_REPLID_MAPPING flag.
ecProfileNotConfigured	0x0000011C	A user tries to log on to a non-home message store and the USE_ADMIN_PRIVILEGE bit in the **OpenFlags** field is not set.
ecAccessDenied	0x80070005	The user does not have sufficient permissions to the mailbox.
ecLoginPerm	0x000003F2	A user without owner permission attempted to create a mailbox.
ecServerPaused	0x0000047F	The client has made more than five attempts within a 10-second period to log on to a mailbox that is not hosted on the server.

3.2.5.2 Receiving a RopGetReceiveFolder ROP Request

The server verifies that the operation is being performed against a private mailbox logon, and not a public folders logon. If the operation is performed against a public folders logon, the server MUST fail the operation with the **ReturnValue** field set to 0x80040102.

The server validates the value of the **MessageClass** field, as specified in section 2.2.1.2.1. If the value does not conform to the requirements, then the server MUST fail the operation with the **ReturnValue** field set to 0x80070057.

The server then searches the Receive folder table to find the entry with the longest case-insensitive prefix string that matches the value of the **MessageClass** field. The search is scoped to the mailbox that is identified by the logon. The server then retrieves the actual message class string from the Receive folder table, and the associated Folder ID, as specified in [MS-OXCDATA] section 2.2.1.1. The Receive folder table is primed for the mailbox at creation time, as specified in section 3.2.3.

If no entry in the table can be matched, the server returns an empty string in the **ExplicitMessageClass** field and the Folder ID for the user's **Inbox folder** in the **FolderId** field. If a match is found, the server returns a string specifying the actual configured message class and the Folder ID of the associated Receive folder. The server can case-fold the string to all uppercase or all lowercase, or leave the string as stored.

For example, if the client sends "IPM.Schedule.Meeting.Request" in the **MessageClass** field, the string returned in the **ExplicitMessageClass** field might be "IPM.Schedule.Meeting", which implies that all messages that are message class "IPM.Schedule.Meeting" (or that are a subclass of "IPM.Schedule.Meeting") will be delivered to the folder that is associated with the "IPM.Schedule.Meeting" message class. In this same example, if a client sends "IPM.Schedule.Meeting" in the **MessageClass** field, the string "IPM.Schedule.Meeting" will be returned in the **ExplicitMessageClass** field.

As a second example, suppose that the client sends a request with either "MY.Class" or "" (an empty string) in the **MessageClass** field. In both cases, the longest prefix substring match is the empty string. Therefore, the server will return an empty string in the **ExplicitMessageClass** field and the FID for the user's Inbox folder in the **FolderId** field. If the client requests "IPM.MY.Class", the server will return "IPM" in the **ExplicitMessageClass** field and the FID for the user's Inbox folder in the **FolderId** field.

As a third example, suppose that the client creates a folder and then uses **RopSetReceiveFolder** to register the message class "MY.Class". If the client's **RopGetReceiveFolder** request specifies the message class "MY.Class.SOMETHING", the server will return the string "MY.Class" and the Folder ID registered for "MY.Class".

The following specific error codes apply to this ROP. Other possible error codes are specified in [MS-OXCDATA] section 2.4.

Name	Value	Meaning
ecNone	0x00000000	Success.
ecInvalidParam	0x80070057	The **MessageClass** value does not conform to the format requirements specified in section 2.2.1.2.1.
ecNotSupported	0x80040102	The ROP was not performed against a private mailbox logon.

3.2.5.3 Receiving a RopSetReceiveFolder ROP Request

The server verifies that the operation is being performed against a private mailbox logon, and not a public folders logon. If the operation is performed against a public folders logon, then the server MUST fail the operation with the **ReturnValue** field set to 0x80040102.

The server validates the value of the **MessageClass** field, as specified in section 2.2.1.3.1. If the value does not conform to the requirements, then the server MUST fail the operation with the **ReturnValue** field set to 0x80070057.

If the value of the **MessageClass** field is a case-insensitive match to either "IPM" or "Report.IPM", then the server MUST fail the operation with the **ReturnValue** field set to 0x80070005. If the

MessageClass field is set to a zero-length string and the **FolderId** field is set to zero, then the server MUST fail the operation with the **ReturnValue** field set to 0x80004005.

The server searches the Receive folder table using a case-insensitive string comparison for an exact match to the value of the **MessageClass** field. The search is scoped to the mailbox that is identified by the logon. If a match is found, the value of the **FolderId** field replaces the Folder ID, as specified in [MS-OXCDATA] section 2.2.1.1, stored in the table on that row. If the **FolderId** field is set to zero, then the table row for the specified message class is deleted from the Receive folder table. (The details about the content of a table row are provided following this paragraph.) If a match is not found, a new row is added with the **MessageClass** and **FolderId** field values. The server can case-fold the value of the **MessageClass** field to upper case or lower case, or leave the value unchanged before storage. After modifying or inserting the new row, the "Last-modification Time" column for that row is set to the current system-time of the server, adjusted to Coordinated Universal Time (UTC).

The Receive folder table is initialized when the mailbox is created, as specified in section 3.2.3. Each row of the table contains at least the following three columns, with each column corresponding to a property. Any other columns included in the table, and the storage format for the table, are determined by the implementer.

1. "Folder ID" column (**PidTagFolderId** property ([MS-OXPROPS] section 2.691)) — Contains the Folder ID, as specified in [MS-OXCDATA] section 2.2.1.1, of the Receive folder, which is the folder to which messages of the specified message class will be delivered. The Receive folder MUST be a folder that is within the user's mailbox.

2. "Message Class" column (**PidTagMessageClass** property ([MS-OXPROPS] section 2.778)) — Contains a string that specifies the message class that is configured for the Receive folder.

3. "Last-modification Time" column (**PidTagLastModificationTime** property ([MS-OXPROPS] section 2.755)) — Contains the current system-time, in UTC, when the entry was created or last modified.

The following specific error codes apply to this ROP. Other possible error codes are specified in [MS-OXCDATA] section 2.4.

Name	Value	Meaning
ecNone	0x00000000	Success.
ecAccessDenied	0x80070005	The client has attempted to change the Receive folder for the "IPM" or "Report.IPM" classes.
ecInvalidParam	0x80070057	The message class string does not conform to the requirements specified in section 2.2.1.3.1.
ecError	0x80004005	The FID (specified in the **FolderId** field) is all zeros AND the message class string (specified in the **MessageClass** field) has a length of zero.
ecNotSupported	0x80040102	The ROP was not performed against a private mailbox logon.

3.2.5.4 Receiving a RopGetReceiveFolderTable ROP Request

The server verifies that the operation is being performed against a private mailbox logon, and not a public folders logon. If the operation is performed against a public folders logon, the server MUST fail the operation with a **ReturnValue** of 0x80040102.

The server MUST return all rows of the Receive folder table for the mailbox identified by the logon. If there are no entries in the Receive folder table, the server MUST fail the operation with 0x00000463 (ecNoReceiveFolder) in the **ReturnValue** field. The **Rows** field of the **RopGetReceiveFolderTable**

response (section 2.2.1.4.2) contains either a **StandardPropertyRow** structure or a **FlaggedPropertyRow** structure for each row of the Receive folder table. If there is an error retrieving any data of a row from the Receive folder table, the server returns the row formatted as a **FlaggedPropertyRow** structure; otherwise, the server returns the row formatted as a **StandardPropertyRow** structure. For more details about these structures, see [MS-OXCDATA] section 2.8.1 and its sub-sections.

The server can convert message class values to all upper case or all lower case or return the value as stored.

The following specific error codes apply to this **ROP**. Other possible error codes are specified in [MS-OXCDATA] section 2.4.

Name	Value	Meaning
ecNone	0x00000000	Success.
ecNoReceiveFolder	0x00000463	There are no configured Receive folder entries.
ecNotSupported	0x80040102	The ROP was not performed against a private mailbox logon.

3.2.5.5 Receiving a RopGetStoreState ROP Request

Servers SHOULD NOT implement this ROP and SHOULD return a value of 0x80040FFF (NotImplemented) in the **ReturnValue** field of the response. Servers MAY implement this ROP as specified in this section.<37>

If the server implements this ROP, it has the following behavior:

- The server verifies that the operation is being performed against a private mailbox logon, and not a public folders logon. If the operation is performed against a public folders logon, then the server MUST fail the operation with a **ReturnValue** of 0x80040102 (NotSupported).
- If the mailbox has any persisted search folders, then the server MUST set the STORE_HAS_SEARCHES flag in the response, as specified in section 2.2.1.5.2. If the mailbox does not have any persisted search folders, then the server MUST NOT set the STORE_HAS_SEARCHES flag in the response.
- The server MUST NOT set any other flags in the response.

The following specific error codes apply to this ROP. Other possible error codes are specified in [MS-OXCDATA] section 2.4.

Name	Value	Meaning
Success	0x00000000	The operation succeeded.
NotSupported	0x80040102	The ROP was not performed against a private mailbox logon.
NotImplemented	0x80040FFF	The server does not implement this ROP.

3.2.5.6 Receiving a RopGetOwningServers ROP Request

If the operation is performed against a private mailbox message store, the server can fail the operation, or it can compute a correct answer for the client. If the public folder specified by the

FolderId field cannot be found in the public folder database, the server MUST fail the operation with 0x8004010F (ecNotFound) in the **ReturnValue** field.

Each public folder has associated configuration information, including which servers are configured to actually hold content of the folder. This specific configuration indicates one of several potential states for each replica server. An "Active" replica contains content and is expected to serve that content to clients. Servers in other replica states do not serve content to clients. These other replica states are implementation-specific, but possible definitions are as follows:

- An "Inactive" replica contains content, but is not going to serve that content to clients.
- A "Deleted" replica once contained content and presently does not.

Each server in the organization's network has a tangible communication cost due to the following implementation-dependent factors: network hardware costs, the cost of the network connectors (various WAN versus LAN costs and so forth), and the perceived cost of using the network for certain applications, and so on.

The server retrieves the current replica information for the specific public folder specified by the **FolderId** field. This replica information is a list, each entry consisting of at least a server identifier and the replication state for this folder on that server (Active/Inactive/Deleted/etc.). The server obtains the network cost for each server in the list, if that cost information isn't already in the list. The source used to determine these network costs can be whatever configuration source the server finds most appropriate.<38> The network cost values are expressed relative to the server servicing the request, not the client making the request.

The server removes entries from the list that are not active replicas. The server can remove entries from the list which, at its discretion, it determines to be too expensive for the client to reach. The algorithm used to determine the servers that are too expensive is implementation-defined.<39> The server can remove entries if another configuration indicates that the client be prohibited from attempting a connection, if such a configuration exists. If the resulting trimmed list is empty, the operation MUST fail with a **ReturnValue** of 0x00000469.

The server sorts the list according to the cost information, least expensive items sorting to the front of the list. Servers with the same cost can appear in any order, but the server SHOULD ensure that the same list values sort to the same order every time.

The server counts the number of **lowest-cost servers** at the front of the list that all have the same cost value, as described previously in this section. The resultant value is the number of cheapest, equally costed servers (the **CheapServersCount** value returned in the response), in terms of the tangible communication cost, relative to the server servicing the request.

The current total list length constitutes the **OwningServersCount** value returned in the response. The list contents of server identifiers constitute the value in the **OwningServers** field. The server MUST map whatever identifier moniker for each server it has into an ESSDN string to return to the client.

The following specific error codes apply to this ROP. Other possible error codes are specified in [MS-OXCDATA] section 2.4.

Name	Value	Meaning
ecNone	0x00000000	Success.
ecNoReplicaAvailable	0x00000469	There are no active replicas for the folder OR the only available replicas have been deemed "too expensive" to reach or are otherwise deemed "unavailable" by the server implementation.
ecNotFound	0x8004010F	The FID could not be found in the public folder database.

3.2.5.7 Receiving a RopPublicFolderIsGhosted ROP Request

If the operation is issued against a private mailbox message store, the server MUST return FALSE in the **IsGhosted** field of the response. In this case, no replication state data is returned. If the public folder specified by the **FolderId** field cannot be found in the public folder container (either public folder database or public folder mailbox), the server MUST fail the operation with 0x8004010F (ecNotFound) in the **ReturnValue** field.

Each public folder has associated configuration information, including which servers are configured to actually hold content of the folder.<40> This specific configuration indicates one of several potential states for each replica server. An "Active" replica contains content and is expected to serve that content to clients. Servers in other replica states do not serve content to clients. These other replica states are implementation-specific, but possible definitions are as follows:

- An "Inactive" replica contains content, but is not going to serve that content to clients.
- A "Deleted" replica once contained content and presently does not.

Each server in the organization's network has a tangible communication cost due to the following implementation-dependent factors: network hardware costs, the cost of the network connectors (various WAN versus LAN costs, and so forth), and the perceived cost of using the network for certain applications, and so on.

The server retrieves the current replica information for the specific public folder specified by the **FolderId** field. This replica information is a list, each entry consisting of at least a server identifier and the replication state (Active, Inactive, Deleted) for this folder on that server. The server obtains the network cost for each server in the list, if that cost information isn't already in the list. The source used to determine these network costs can be whatever configuration source the server finds most appropriate.<41> The network cost values are expressed relative to the server, not the client making the request.

The server MUST return TRUE in the **IsGhosted** field if the queried server is not listed as an active replica of the folder. The value of the **IsGhosted** field MUST be FALSE if the queried server is listed as an active replica of the folder. If the client issues this operation against the IPM subtree or the non-IPM subtree public folders, the value of the **IsGhosted** field MUST be FALSE.

The server removes entries from the list which are not active replicas. The server can remove entries from the list which, at its discretion, it determines to be too expensive for the client to reach. The algorithm used to determine the servers that are too expensive is implementation-defined.<42> The server can remove entries if another configuration indicates that the client be prohibited from attempting a connection, if such configuration exists. If the resulting trimmed list is empty, the operation MUST be failed with a **ReturnValue** of 0x00000469. A client MUST interpret this **ReturnValue** value as implying an **IsGhosted** value of TRUE.

The server sorts the list according to the cost information, least expensive items sorting to the front of the list. Servers with the same cost can appear in any order, but the server SHOULD ensure that the same list values sort to the same order every time.

The server counts the number of entries at the front of the list that all have the same cost value. The resultant value will be the number of cheapest, equally costed servers (the **CheapServersCount** return value).

The current total list length constitutes the **ServersCount** return value. The list contents of server identifiers constitute the value of the **Servers** field. The server MUST map whatever identifier moniker for each server it has into an ESSDN string to return to the client.

The following specific error codes apply to this ROP. Other possible error codes are specified in [MS-OXCDATA] section 2.4.

Name	Value	Meaning
ecNone	0x00000000	Success.
ecNoReplicaAvailable	0x00000469	There are no active replicas for the folder OR the only available replicas have been deemed "too expensive" to reach or are otherwise deemed "unavailable" by the server implementation.
ecNotFound	0x8004010F	The FID could not be found in the public folder container.

3.2.5.8 Receiving a RopLongTermIdFromId ROP Request

The server searches the REPLID and REPLGUID to-and-from mapping table for the replica ID (REPLID) portion of the given **Folder ID** or **Message ID**, as specified in [MS-OXCDATA] sections 2.2.1.1 or 2.2.1.2. The server does not attempt to confirm that the given **Folder ID** or **Message ID** is a change number for an existing or former object, is an identifier for an existing or former object, or has ever been assigned for any reason.

If the REPLID is not in the REPLID and REPLGUID to-and-from mapping table, the operation SHOULD<43> fail with a **ReturnValue** of 0x8004010F. A REPLID MUST NOT have a value of zero.

The server MUST map the same REPLID to the same **replica GUID (REPLGUID)** every time it is queried. Other servers can map a particular REPLID to a different REPLGUID, but each server MUST map any particular REPLID to the same REPLGUID every time it is queried.

After obtaining the REPLGUID from the REPLID and REPLGUID to-and-from mapping table, the server uses the REPLGUID to construct the 192-bit **LongTermID**, as specified in [MS-OXCDATA] section 2.2.1.3.1, which is returned in the **LongTermId** field. The **LongTermID** consists of the 128-bit REPLGUID, followed by the 48-bit global counter portion of the given **Folder ID** or **Message ID**, followed by 16 bits of padding set to 0x0000.

The following specific error codes apply to this ROP. Other possible error codes are specified in [MS-OXCDATA] section 2.4.

Name	Value	Meaning
ecNone	0x00000000	Success.
ecNotFound	0x8004010F	The REPLID portion of the ID could not be found in the REPLID and REPLGUID to-and-from mapping table.

3.2.5.9 Receiving a RopIdFromLongTermId ROP Request

If the **LongTermId** field of the request contains zeros for the replica GUID (REPLGUID) component, the server SHOULD<44> fail the operation with 0x80070057 (ecInvalidParam) in the **ReturnValue** field. If the **LongTermId** field of the request contains zeros for the global counter component, the server MUST fail the operation with 0 in the **ReturnValue** field.

The server searches the REPLID and REPLGUID to-and-from mapping table for the REPLGUID portion of the **LongTermID**, as specified in [MS-OXCDATA] section 2.2.1.3.1. The server does not attempt to confirm that the **LongTermID** is a change number for an existing or former object, is an identifier for an existing or former object, or has ever been assigned for any reason.

If the REPLGUID is not found, the server adds a new entry consisting of the REPLGUID portion of the **LongTermID** and a newly assigned replica ID (REPLID). The new REPLID MUST be unique in the REPLID and REPLGUID to-and-from mapping table. If the maximum number of REPLIDs have already been registered, the server cannot register a new REPLID. In this case, the server MUST fail the operation with 0x00000450 (ecParameterOverflow) in the **ReturnValue** field.

The server MUST map the same REPLGUID to the same REPLID every time it is queried. Other servers can map a particular REPLGUID to a different REPLID, but each server MUST map any particular REPLGUID to the same REPLID every time it is queried.

The server ignores the content of the padding bytes in the LongTermID.

After obtaining the REPLID from the REPLID and REPLGUID to-and-from mapping table, the server uses the REPLID to construct the 64-bit Folder ID or Message ID, as specified in [MS-OXCDATA] sections 2.2.1.1 or 2.2.1.2, which is returned in the **ObjectId** field. The **Folder ID** or **Message ID** consists of the 16-bit REPLID followed by the 48-bit global counter portion of the given **LongTermID**.

The following error codes are specific to this ROP. Other possible error codes are specified in [MS-OXCDATA] section 2.4.

Name	Value	Meaning
ecNone	0x00000000	Success.
ecInvalidParam	0x80070057	The **LongTermId** field of the request contained zeros for either the replica GUID (REPLGUID) component.<45>
ecParameterOverflow	0x00000450	The number of replica IDs registered is at the maximum. (The maximum is 32,768, but the implementation can impose a lower limit.)

3.2.5.10 Receiving a RopGetPerUserLongTermIds ROP Request

The server verifies that the operation is being performed against a private mailbox logon, not against a public folders logon. If the operation is performed against a public folders logon, the server MUST fail the operation with a **ReturnValue** of 0x80040102.

The server searches the per-user data table of the mailbox for entries identified by the **DatabaseGuid** field in the request. For each entry in the table, the server collects the associated public folder **LongTermID**, as specified in [MS-OXCDATA] section 2.2.1.3.1. The total number of **LongTermIDs** collected is specified in the **LongTermIdCount** field and the aggregated list of **LongTermIDs** constitutes the value of the **LongTermIds** field.

The server can return the list of **LongTermIDs** in any order. The server can return an empty list.

The following specific error codes apply to this ROP. Other possible error codes are specified in [MS-OXCDATA] section 2.4.

Name	Value	Meaning
ecNone	0x00000000	Success.
ecNotSupported	0x80040102	The ROP was attempted against a public folders logon.

3.2.5.11 Receiving a RopGetPerUserGuid ROP Request

The server verifies that the operation is being performed against a private mailbox logon, and not a public folders logon. If the operation is performed against a public folders logon, the server MUST fail the operation with a **ReturnValue** of 0x80040102.

The server searches the per-user data table for the mailbox for the only row with an FID that is associated with the public folder specified by the **LongTermId** field in the request. The server returns the associated REPLGUID value in the **DatabaseGuid** field. If the public folder specified by the **LongTermId** field cannot be found, the server MUST fail the operation with 0x8004010F (ecNotFound) in the **ReturnValue** field.

The following specific error codes apply to this ROP. Other possible error codes are specified in [MS-OXCDATA] section 2.4.

Name	Value	Meaning
ecNone	0x00000000	Success.
ecNotSupported	0x80040102	ROP was attempted against a public folders logon.
ecNotFound	0x8004010F	The public folder identified by the value of the **LongTermId** field could not be found

3.2.5.12 Receiving a RopReadPerUserInformation ROP Request

This operation can be issued against either a private mailbox logon or a public folders logon.

3.2.5.12.1 Behavior Common to Both Private Mailbox and Public Folder Logon

Messages that are modified receive a new change number (CN) and hence fall out of the set of read messages. The user will see these modified messages marked as unread. If the user marks a message as read, the current value of that message's **PidTagChangeNumber** property ([MS-OXPROPS] section 2.623) is added to the change number set (CNSET). If the user marks a message as unread, the current value of that message's **PidTagChangeNumber** property is removed from the CNSET.

The change number set MUST be serialized into a binary large object (BLOB) that is formatted as a serialized IDSET with REPLGUID structure, as specified in [MS-OXCFXICS] section 2.2.2.4.2. The server then returns the BLOB in the **Data** field of the response.

The size of the BLOB can potentially exceed the maximum amount of data that can be communicated in a single **RopReadPerUserInformation** response (section 2.2.1.12.2). For this reason, the **RopReadPerUserInformation** ROP ([MS-OXCROPS] section 2.2.3.12) is designed to stream the data to the client by having the client invoke the ROP multiple times. In other words, the client can send multiple **RopReadPerUserInformation** requests (section 2.2.1.12.1) to retrieve the BLOB in segments.

On each invocation of **RopReadPerUserInformation**, the server inspects the value of the **MaxDataSize** field of the **RopReadPerUserInformation** request because the value can be different in each request. In certain cases, the server MUST adjust the value of **MaxDataSize**. For more details about inspecting and adjusting the value, see the summary in this section. After the server has inspected and, if necessary, adjusted the value of **MaxDataSize**, the server compares the value to the size of the remaining BLOB segment. If the adjusted **MaxDataSize** value is less than the size of the remaining BLOB segment, then the server MUST set **HasFinished** field to FALSE to indicate to the client that some data remains to be retrieved.

The **DataOffset** field in the request contains an index into the BLOB. In other words, the value of **DataOffset** specifies the position within the BLOB of the first byte of data to be returned to the client. The value of **DataOffset** is always zero in the first **RopReadPerUserInformation** request. The client updates **DataOffset** based on the number of bytes received in the previous response so that **DataOffset** always points to the first byte of the next BLOB segment to be returned. If the value of the **DataOffset** field is less than zero, the server SHOULD<46> fail the operation with 0x80004005 (ecError) in the **ReturnValue** field. If the value of the **DataOffset** field is greater than the size of the next BLOB segment to be returned, the server MUST fail the operation with 0x80004005 (ecError) in the **ReturnValue** field.

Summary:

1. The MaxDataSize field of the request specifies the maximum number of bytes that can be returned in a single **RopReadPerUserInformation** response. The server MUST adjust the MaxDataSize value in certain cases, as specified in item 2 of this summary.

2. When the client retrieves a BLOB in segments, the client can set MaxDataSize to a different value in each **RopReadPerUserInformation** request that is used to retrieve the BLOB. Therefore, the server examines the value of MaxDataSize on each invocation of **RopReadPerUserInformation** as follows.

    1. The server compares the value of MaxDataSize to zero. If MaxDataSize equals 0, then the server MUST adjust the value of MaxDataSize to a suitable default value, which is determined by the implementation.<47>

    2. The server SHOULD compare the value of MaxDataSize to some suitable maximum value, as determined by the implementation. If MaxDataSize > [server's suitable maximum], then the server SHOULD adjust the value of MaxDataSize to the suitable maximum value.<48>

    3. The server compares the adjusted value of MaxDataSize to the size of the remaining BLOB segment. If [size of remaining BLOB segment] > [adjusted MaxDataSize], then the server MUST set HasFinished to FALSE to indicate to the client that additional requests are necessary to retrieve all of the remaining portions of the BLOB. The size of the remaining BLOB segment is equal to the size of the entire BLOB minus the value of DataOffset.

3. The DataSize field specifies the actual number of bytes that are returned in the response. The value of DataSize MUST NOT exceed the adjusted value of the MaxDataSize field. For details about adjusting MaxDataSize, see item number 2 of this summary. The server MUST set DataSize to the lesser of the following two values:

    1. The adjusted value of MaxDataSize.

    2. The size of the remaining BLOB segment. This is the size of the portion of the BLOB that remains to be sent to the client and is equal to the size of the entire BLOB minus the value of DataOffset.

4. The server MUST set HasFinished to TRUE if DataOffset plus DataSize equals the size of the entire BLOB. In other words, when the server sends the last segment of the BLOB, HasFinished MUST be set to TRUE.

The following specific error codes apply to this ROP. Other possible error codes are specified in [MS-OXCDATA] section 2.4.

Name	Value	Meaning
ecNone	0x00000000	Success.
ecRpcFormat	0x000004B6	The **DataOffset** value was less than zero.
ecError	0x80004005	The **DataOffset** value was greater than the data size.

3.2.5.12.2 Private Mailbox Specific Behavior

The server searches the per-user data table for the mailbox for the only row with an FID equal to the value of the **FolderId** field. If the row exists, then the server retrieves from that row the stored change number set of read items. If the row does not exist, then the server returns an empty array in the **Data** field of the response. After the change number set is retrieved, the server's behavior, as specified in section 3.2.5.12.1, is the same for both private mailboxes and public folders.

3.2.5.12.3 Public Folders Specific Behavior

The server first determines that the persisted read/unread information for the user is up to date. If the server is maintaining any in-memory caches of the per-user read/unread information, the data for the current user MUST now be flushed to disk.

The server searches the per-user data table for the only row with an FID equal to the value of the **FolderId** field and the user ID equal to the logged on user. If the row exists, the server retrieves from that row the stored change number set of read items. If the row does not exist, then the server returns an empty array in the **Data** field of the response. After the change number set is retrieved, the server's behavior, as specified in section 3.2.5.12.1, is the same for both private mailboxes and public folders.

3.2.5.13 Receiving a RopWritePerUserInformation ROP Request

This operation can be issued against either a private mailbox logon or a public folders logon.

3.2.5.13.1 Behavior Common to Both Private Mailbox and Public Folder Logon

Each invocation of this ROP accumulates data from the client until the client makes a final call with **HasFinished** set to TRUE. The server aggregates the data across multiple invocations and it validates the entire data set before persisting to permanent storage.

The server determines whether the current invocation is a continuation of a previous invocation by examining the **FolderId** and **DataOffset** fields. If the FID has changed since the last invocation, or the **DataOffset** value does not equal the amount of data already written, the server MUST assume the previous operation was aborted and MUST dispose of any accumulated data. In addition, if the current invocation's **DataOffset** isn't zero, the ROP MUST fail with a **ReturnValue** of 0x80004005.

Once the client invokes this ROP with **HasFinished** set to TRUE, the server validates the accumulated data and verifies that it is a properly formed serialized IDSET with REPLGUID as specified in [MS-OXCFXICS] section 2.2.2.4.2. If the data is not properly formed, the ROP SHOULD<49> fail with a **ReturnValue** of 0x000004ED.

After performing the specific behavior in the following sections, the server records, in UTC, the current system time on the appropriate row in the table.

The following specific error codes apply to this ROP. Other possible error codes are specified in [MS-OXCDATA] section 2.4.

Name	Value	Meaning
ecNone	0x00000000	Success.
ecError	0x80004005	The **DataOffset** didn't match the size of the data written so far OR the **FolderId** didn't match the value on the previous call, AND THEN **DataOffset** wasn't zero.
ecFmtError	0x000004ED	The data cumulatively written could not be parsed as a proper serialized IDSET with REPLGUIDs, as specified in [MS-OXCFXICS] section 2.2.2.4.2.<50>

3.2.5.13.2 Private Mailbox Specific Behavior

The server searches the per-user data table of the mailbox for the only row with an FID equal to the value of the **FolderId** field. If the row exists, the REPLGUID field and accumulated change number information MUST replace any existing values in the table. If the row does not presently exist, a new row for the mailbox and folder MUST be added, setting the REPLGUID field and accumulated change number information onto that row.

3.2.5.13.3 Public Folders Specific Behavior

The server searches the per-user data table for the only row with a user ID equal to the user ID associated with the session logon and an FID equal to the value of the **FolderId** field. If the row exists, the accumulated change number information MUST replace any existing values in the table. If the row does not exist, a new row for the user and folder MUST be added, setting the accumulated change number information onto that row.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4 Protocol Examples

This section provides a sample sequence of ROP requests and **ROP responses** that a client and a server might exchange as the client logs on to a user mailbox or public folders, reads or writes mailbox-level properties, or determines the availability of content for public folders. Note that the examples listed here only show the relevant portions of the specified ROPs; these portions are not the final byte sequences that get transmitted over the wire. Also note that the data for a multi-byte field appear in **little-endian** format, with the bytes in the field presented from least significant to most significant. Generally speaking, these ROP requests are packed with other ROP requests. These examples assume the client has already successfully connected to the server. For more details, see [MS-OXCRPC] section 4.1.

The byte sequences are shown in the following format with each byte's value expressed as a two-digit hexadecimal number.

```
0080: 45 4d 53 4d 44 42 2e 44-4c 4c 00 00 00 00 00 00
```

The value, 0080, at the far left is the byte sequence's offset from the beginning of the buffer. Following the offset is a colon and then a series of up to 16 byte values. Here, the first byte value (45) in the series is located 0x80 bytes (128 bytes) from the beginning of the buffer. The seventh byte value (2e) in the series is located 0x86 bytes (134 bytes) from the beginning of the buffer. The dash between the eighth byte (44) and ninth byte (4c) has no semantic value, and serves only to distinguish the eight byte boundary for readability purposes.

Each set of byte sequences is followed by one or more lines interpreting it.

The following example shows how a property tag and its property value are represented in a buffer and interpreted directly from it (according to the **TaggedPropertyValue** structure format specified in [MS-OXCDATA] section 2.11.4). The data appears in the buffer in little-endian format.

```
0020: 03 00 76 66 0a 00 00-00
```

[0020-0023] Property tag: 0x66760003 (**PidTagRuleSequence** ([MS-OXPROPS] section 2.953))

[0024-0027] Property value: 10

Generally speaking, interpreted values will be shown in their native format, interpreted appropriately from the raw byte sequence as specified in the appropriate section. Here, the byte sequence "0a 00 00 00" has been interpreted as a **PtypInteger32** with a value of 10 because the type of the **PidTagRuleSequence** property is **PtypInteger32**. Property data types are specified in [MS-OXCDATA] section 2.11.1.

4.1 RopLogon for a Private Mailbox

RopLogon request (section 2.2.1.1.1) for a private mailbox:

```
0000: 01 0C 04 00 01 00 00 00-00 68 00 2F 6F 3D 46 69
0010: 72 73 74 20 4F 72 67 61-6E 69 7A 61 74 69 6F 6E
0020: 2F 6F 75 3D 45 78 63 68-61 6E 67 65 20 41 64 6D
0030: 69 6E 69 73 74 72 61 74-69 76 65 20 47 72 6F 75
0040: 70 20 28 46 59 44 49 42-4F 48 46 32 33 53 50 44
0050: 4C 54 29 2F 63 6E 3D 52-65 63 69 70 69 65 6E 74
0060: 73 2F 63 6E 3D 41 64 6D-69 6E 69 73 74 72 61 74
0070: 6F 72 00
```

[0000-0000] **LogonFlags** — Private

[0001-0004] **OpenFlags** — HOME_LOGON, TAKE_OWNERSHIP, NO_MAIL, USE_PER_MDB_REPLID_MAPPING

[0005-0008] **StoreState** — Value is ignored by the server.

[0009-000A] **EssdnSize** — The size of the **Essdn** field is 0x68 bytes long.

[000B-0072] **Essdn**

RopLogon success response for a private mailbox (section 2.2.1.1.3):

```
0000: 01 01 00 00 00 00 78 27-1A 01 00 00 00 00 78 27
0010: 1C 01 00 00 00 00 78 27-1D 01 00 00 00 00 78 27
0020: 1B 01 00 00 00 00 78 27-1E 01 00 00 00 00 78 27
0030: 1F 01 00 00 00 00 78 27-20 01 00 00 00 00 78 27
0040: 21 01 00 00 00 00 78 27-24 01 00 00 00 00 78 27
0050: 25 01 00 00 00 00 78 27-22 01 00 00 00 00 78 27
0060: 23 01 00 00 00 00 78 27-26 07 F7 F8 91 A5 1C 34
0070: 16 41 8C 48 9D B0 1A 86-F5 0B 01 00 4D 77 D4 64
0080: 83 49 70 4F 9B 8B 46 E6-35 BB 78 AB 0D 10 0F 01
0090: 0A 03 D8 07 60 53 1A C2-BE 82 C8 01 00 00 00 01
```

[0000-0000] **LogonFlags** — Private

[0001-0068] **FolderIds** — As follows:

[0001-0008] Mailbox Root Folder FID

[0009-0010] Deferred Action Folder FID

[0011-0068] <more FIDs>

[0069-0069] **ResponseFlags** — SendAsRight, OwnerRight, Reserved

[006A-0079] **MailboxGuid**

[007A-007B] **ReplId**

[007C-008B] **ReplGuid**

[008C-0093] **LogonTime** — 2008/03/10 Mon 15:10:13

[0094-009B] **GwartTime** — 2008/03/10 14:55:19

[009C-009F] **StoreState** — STORE_HAS_SEARCHES

4.2 RopLogon for Public Folders

RopLogon request (section 2.2.1.1.1) for public folders:

```
0000: 00 04 04 00 01 00 00 00-00 00 00
```

[0000-0000] **LogonFlags** — Log on to public folders

[0001-0004] **OpenFlags** — HOME_LOGON, NO_MAIL, USE_PER_MDB_REPLID_MAPPING

[0005-0008] **StoreState** — Value is ignored.

[0009-000A] **EssdnSize** — No ESSDN is given for public logons.

RopLogon success response for public folders (section 2.2.1.1.4):

```
0000: 00 01 00 00 00 00 00 00-06 01 00 00 00 00 00 00
0010: 01 01 00 00 00 00 00 00-02 01 00 00 00 00 00 00
0020: 03 01 00 00 00 00 00 00-04 01 00 00 00 00 00 00
0030: 05 00 00 00 00 00 00 00-00 03 00 00 00 00 00 00
0040: 07 03 00 00 00 00 00 00-08 00 00 00 00 00 00 00
0050: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0060: 00 00 00 00 00 00 00 00-00 01 00 70 5B CA BF 1E
0070: F9 98 41 89 7D 47 9E 09-45 FD 2F 95 DA FE 74 E0
0080: F7 4C 4C 81 EF 83 BA 85-0B E8 E4
```

[0000-0000] **LogonFlags** — Log on to public folders

[0001-0050] **FolderIds** — As follows:

[0001-0008] Public Folders FID

[0009-0010] IPM Subtree FID

[0011-0050] <other FIDs>

[0051-0068] Unused — Set to zero.

[0069-006A] **ReplId**

[006B-007A] **ReplGuid**

[007B-008A] **PerUserGuid**

4.3 RopGetReceiveFolder

RopGetReceiveFolder request (section 2.2.1.2.1):

```
0000: 00
```

[0000-0000] **MessageClass** <empty string>

RopGetReceiveFolder response (section 2.2.1.2.2):

```
0000: 01 00 00 00 00 78 27 1E-00
```

[0000-0007] **FolderId**

[0008-0008] **ExplicitMessageClass** <empty string>

4.4 RopSetReceiveFolder

RopSetReceiveFolder request (section 2.2.1.3.1):

```
0000: 01 00 00 00 00 78 27 1A-49 50 4D 2E 53 6F 6D 65
0010: 4D 65 73 73 61 67 65 43-6C 61 73 73 00
```

[0000-0007] **FolderId**

[0008-001C] **MessageClass**

RopSetReceiveFolder response (section 2.2.1.3.2):

No response.

4.5 RopGetReceiveFolderTable

RopGetReceiveFolderTable request (section 2.2.1.4.1):

No fields in the request.

RopGetReceiveFolderTable response (section 2.2.1.4.2):

```
0000: 04 00 00 00 00 01 00 00-00 00 78 27 1E 00 5E FF
0010: 54 5F C0 82 C8 01 00 01-00 00 00 00 78 27 1A 49
0020: 50 43 00 32 EF 56 5F C0-82 C8 01 00 01 00 00 00
0030: 00 78 27 1E 49 50 4D 00-32 EF 56 5F C0 82 C8 01
0040: 00 01 00 00 00 00 78 27-1E 52 45 50 4F 52 54 2E
0050: 49 50 4D 00 32 EF 56 5F-C0 82 C8 01
```

[0000-0003] **RowCount** (4 rows being returned)

[0004-005B] **Rows** — As follows:

[0004-0004], [0016-0016], [002B-002B], [0040-0040] Error Flag Indicator (no error)

[0005-000C], [0017-001E], [002C-0033], [0041-0048] **PidTagFolderId** property ([MS-OXPROPS] section 2.691)

[000D-000D], [001F-0022], [0034-0037], [0049-0053] **PidTagMessageClass** property ([MS-OXPROPS] section 2.778)

[000E-0015], [0023-002A], [0038-003F], [0054-005B] **PidTagLastModificationTime** property ([MS-OXPROPS] section 2.755)

4.6 RopIdFromLongTermId

RopIdFromLongTermId request (section 2.2.1.9.1):

```
0000: 70 5B CA BF 1E F9 98 41-89 7D 47 9E 09 45 FD 2F
0010: 00 00 00 00 00 12 00 00
```

[0000-000F] **LongTermID REPLGUID**

[0010-0015] **LongTermID** counter

[0016-0017] **LongTermID** padding

RopIdFromLongTermId response (section 2.2.1.9.2):

```
0000: 05 00 00 00 00 00 00 12
```

[0000-0001] **ObjectId REPLID**

[0002-0007] **ObjectId** counter

4.7 RopGetPerUserLongTermIds

RopGetPerUserLongTermIds request (section 2.2.1.10.1):

```
0000: 4D 77 D4 64 83 49 70 4F-9B 8B 46 E6 35 BB 78 AB
```

[0000-000F] **DatabaseGuid**

RopGetPerUserLongTermIds response (section 2.2.1.10.2):

```
0000: 00 00
```

[0000-0001] **LongTermIdCount** (no IDs being returned)

4.8 RopReadPerUserInformation

RopReadPerUserInformation request (section 2.2.1.12.1):

```
0000: 70 5B CA BF 1E F9 98 41-89 7D 47 9E 09 45 FD 2F
0010: 00 00 00 00 00 12 00 00-00 00 00 00 00 00 00
```

[0000-0017] **FolderId**

[0018-0018] **Reserved**

[0019-001C] **DataOffset**

[001D-001E] **MaxDataSize**

RopReadPerUserInformation response (section 2.2.1.12.2):

```
0000: 01 18 00 D8 44 AE 73 F9-61 5D 4F B3 C6 9A 7C 31
0010: FE C1 23 06 00 00 00 78-2B 33 00
```

[0000-0000] **HasFinished**

[0001-0002] **DataSize**

[0003-001A] **Data**

4.9 RopWritePerUserInformation

RopWritePerUserInformation request (section 2.2.1.13.1) :

```
0000: 70 5B CA BF 1E F9 98 41-89 7D 47 9E 09 45 FD 2F
0010: 00 00 00 00 00 12 00 00-01 00 00 00 00 18 00 D8
0020: 44 AE 73 F9 61 5D 4F B3-C6 9A 7C 31 FE C1 23 06
0030: 00 00 00 78 2B 33 00 D8-44 AE 73 F9 61 5D 4F B3
0040: C6 9A 7C 31 FE C1 23
```

[0000-0017] **FolderId**

[0018-0018] **HasFinished**

[0019-001C] **DataOffset**

[001D-001E] **DataSize**

[001F-0036] **Data**

[0037-0046] **ReplGuid**

RopWritePerUserInformation response (section 2.2.1.13.2):

No response.

5 Security

5.1 Security Considerations for Implementers

There are no special security considerations specific to the Store Object protocol. General security considerations pertaining to the underlying RPC-based transport apply. For details, see [MS-OXCROPS].

5.2 Index of Security Parameters

None.

6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs.

- Microsoft Exchange Server 2003
- Microsoft Exchange Server 2007
- Microsoft Exchange Server 2010
- Microsoft Exchange Server 2013
- Microsoft Exchange Server 2016
- Microsoft Office Outlook 2003
- Microsoft Office Outlook 2007
- Microsoft Outlook 2010
- Microsoft Outlook 2013
- Microsoft Outlook 2016

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

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

<1> Section 2.2.1.1.1: In Exchange 2003, Exchange 2007, and Exchange 2010, if the **Private** flag is set, the **Ghosted** flag MUST NOT be set by the client and MUST be ignored by the server; if the **OpenFlags** field has either the **ALTERNATE_SERVER** flag or the **IGNORE_HOME_MDB** flag set, the **Ghosted** flag MUST be ignored by the server; if the **Ghosted** flag is set, the client is requesting a logon to the public folder database that is present on the server. If the **Ghosted** flag is not set, the client is requesting a logon to the default public folder database. In Exchange 2013 and Exchange 2016 the **Ghosted** flag is deprecated.

<2> Section 2.2.1.1.1: Exchange 2003 and Exchange 2007 use the **HOME_LOGON** flag as follows: When the flag is set in a public folder logon, per-user read/unread information is tracked. This flag is ignored in a private **mailbox** logon.

<3> Section 2.2.1.1.1: Exchange 2003 and Exchange 2007 use the **TAKE_OWNERSHIP** flag as follows: If set, then the server checks to determine whether the user can act as an owner of the mailbox. If not set, then the user is considered a delegate.

<4> Section 2.2.1.1.1: Office Outlook 2003 does not set the **USE_PER_MDB_REPLID_MAPPING** flag. Office Outlook 2007 uses this flag to control whether the server maintains one REPLID-to-REPLGUID mapping and one named property-to-property ID mapping for all logon sessions.

<5> Section 2.2.1.1.1: Microsoft Exchange Server 2010 Service Pack 1 (SP1), Exchange 2013, Exchange 2016, Microsoft Outlook 2010 Service Pack 1 (SP1), Outlook 2013, and Outlook 2016 implement this flag. Exchange 2003, Exchange 2007, Exchange 2010, Office Outlook 2003, Office Outlook 2007, and Outlook 2010 do not implement this flag.

<6> Section 2.2.1.1.1: Exchange 2003, Exchange 2007, Exchange 2010, Office Outlook 2003, Office Outlook 2007, and Outlook 2010 require passing 0x00 in the **EssdnSize** field for public folder logon and that the **Essdn** field be empty.

<7> Section 2.2.1.1.3: If the mailbox currently has any active search folders, then Exchange 2003 and Exchange 2007 set this field to 0x01000000; otherwise, this file is set to 0x00000000.

<8> Section 2.2.1.1.4: Exchange 2013 and Exchange 2016 return the empty Folder ID structures for the following folders: Free/Busy Data, Offline Address Book Data, Local Site's Free/Busy Data, Local Site's Offline Address Book Data, and NNTP Article Index.

<9> Section 2.2.1.1.4: Exchange 2007 does not set the **PerUserGuid** field to an empty GUID.

<10> Section 2.2.1.5: Exchange 2010, Exchange 2013, and Exchange 2016 do not implement the **RopGetStoreState** ROP ([MS-OXCROPS] section 2.2.3.5), but it is implemented in Exchange 2003 and Exchange 2007.

<11> Section 2.2.1.6.1: Exchange 2003, Exchange 2007, and Exchange 2010 successfully complete a **RopGetOwningServers** ROP ([MS-OXCROPS] section 2.2.3.6) when issued against a private mailbox logon, but the results are undefined.

<12> Section 2.2.1.7: In Exchange 2013 and Exchange 2016, each public folder has exactly one replica, the folder's content mailbox.

<13> Section 2.2.2.1.2.1: The **PidTagComment** property is read-only in Microsoft Exchange Server 2013 Service Pack 1 (SP1).

<14> Section 2.2.2.1.2.2: Exchange 2013 and Exchange 2016 return 0x80070005 (ecAccessDenied) when the client attempts to set this property by using the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6).

<15> Section 2.2.2.1.2.3: In Exchange 2013 SP1 this property is read-only.

<16> Section 3.1.4.1: Exchange 2003, Exchange 2007, Exchange 2010, the initial release of Exchange 2013, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the initial release of Outlook 2013 do not support the **Connect** request type. The **Connect** request type was introduced in Microsoft Outlook 2013 Service Pack 1 (SP1) and Exchange 2013 SP1.

<17> Section 3.2.3: When a database is restored from backup, Exchange 2003 and Exchange 2007 assign a new randomly-generated REPLGUID to the database and then add this new REPLGUID, along with a new REPLID, to the REPLID and REPLGUID to-and-from mapping table.

<18> Section 3.2.5: Exchange 2003, Exchange 2007, Exchange 2010, the initial release version of Exchange 2013, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the initial release version of Outlook 2013 do not support the **Connect** request type. The **Connect** request type was introduced in Outlook 2013 SP1 and Exchange 2013 SP1.

<19> Section 3.2.5.1.1: Exchange 2010 and Exchange 2013 return RPC fault 0x6ba.

<20> Section 3.2.5.1.1: Exchange 2003 returns ecMailboxDisabled. Exchange 2007 returns ecLoginFailure, then, after 5 minutes, the server returns ecUnknownUser. Exchange 2010, Exchange 2013, and Exchange 2016 return ecUnknownUser. Microsoft Exchange Server 2010 Service Pack 3 (SP3) fails with RPC fault (0x000006BA).

<21> Section 3.2.5.1.1: Exchange 2010 does not return ecUnknownCodePage.

<22> Section 3.2.5.1.1: Exchange 2010 and Exchange 2013 return Success with ecServerPaused.

<23> Section 3.2.5.1.1: The behavior of Exchange 2003, Exchange 2007, Exchange 2010, Exchange 2013, and Exchange 2016 is undefined if the client sets an undefined flag in either the **LogonFlags** field or the **OpenFlags** field.

<24> Section 3.2.5.1.1: Exchange 2010 SP1, Exchange 2013, and Exchange 2016 implement the SUPPORT_PROGRESS flag. Exchange 2003, Exchange 2007, and Exchange 2010 do not implement the flag.

<25> Section 3.2.5.1.1: Exchange 2003 and Exchange 2007 do not support the GET/read operation for the **PidTagSortLocaleId** property. Exchange 2010, Exchange 2013, and Exchange 2016 support the GET/read operation for this property.

<26> Section 3.2.5.1.1: Exchange 2003 ignores the USE_PER_MDB_REPLID_MAPPING flag, and therefore, the behavior of Exchange 2003 is not affected by this flag. Exchange 2003 always maintains one REPLID-to-REPLGUID mapping and one named property-to-property ID mapping per RPC session, and these mappings are shared by all logons on the RPC session.

<27> Section 3.2.5.1.1: In Exchange 2013, if the logon is the first on the RPC session, or if the logon is additional on the RPC session and it is to the same mailbox that is associated with the first logon, then the server fails with an RPC fault.

<28> Section 3.2.5.1.1: If the USE_PER_MDB_REPLID_MAPPING flag is not set, Exchange 2007 does not fail the ROP and instead has the following behavior: Exchange 2007 maintains one REPLID-to-REPLGUID mapping and one named property-to-property ID mapping per RPC session, and these mappings are shared by all logons on the TRPC session.

<29> Section 3.2.5.1.1: Exchange 2003 and Exchange 2007 allow the logon to proceed. Exchange 2010 returns ecWrongServer (0x00000478). For more information about properly forming the response when a **ReturnValue** of 0x00000478 is sent, see section 2.2.1.1.2.

<30> Section 3.2.5.1.1: If the USE_PER_MDB_REPLID_MAPPING flag is set, Exchange 2007 maintains one REPLID-to-REPLGUID mapping and one named property-to-property ID mapping for each logon session.

<31> Section 3.2.5.1.1: Exchange 2003 and Exchange 2007 return ecAccessDenied. Exchange 2010 returns ecLoginPerm. Exchange 2013 and Exchange 2016 return ecNone.

<32> Section 3.2.5.1.1: Exchange 2010 returns ecLoginPerm **ReturnValue**. Exchange 2007 returns ecAccessDenied **ReturnValue**.

<33> Section 3.2.5.1.2: Exchange 2003, Exchange 2007, and Exchange 2010 do not support public folders contained in a mailbox.

<34> Section 3.2.5.1.2: Exchange 2013 and Exchange 2016 always set the USE_AUTODISCOVER_FOR_PUBLIC_FOLDER_CONFIGURATION flag. Exchange 2003, Exchange 2007, and Exchange 2010 do not support this flag. Office Outlook 2003, Office Outlook 2007, and Outlook 2010 ignore this flag.

<35> Section 3.2.5.1.2: The behavior of Exchange 2003, Exchange 2007, Exchange 2010, Exchange 2013, and Exchange 2016 is undefined if the client sets an undefined flag in either the **LogonFlags** field or the **OpenFlags** field.

<36> Section 3.2.5.1.3: If the user doesn't exist in the **Active Directory** forest, Exchange 2003 returns ecLoginFailure.

<37> Section 3.2.5.5: Exchange 2003 and Exchange 2007 implement the **RopGetStoreState** ROP ([MS-OXCROPS] section 2.2.3.5).

<38> Section 3.2.5.6: Exchange 2003 queries the transport engine for cost information. Exchange 2007 and Exchange 2010 query Active Directory for cost information.

<39> Section 3.2.5.6: Exchange 2003 removes servers that have a connection cost of "infinite". Exchange 2007, Exchange 2010, Exchange 2013, and Exchange 2016 remove servers that have a connection cost greater than 500.

<40> Section 3.2.5.7: In Exchange 2013 and Exchange 2016, each public folder has exactly one replica, the folder's content mailbox.

<41> Section 3.2.5.7: Exchange 2003 queries the transport engine for cost information. Exchange 2007, Exchange 2010, Exchange 2013, and Exchange 2016 query Active Directory for cost information.

<42> Section 3.2.5.7: Exchange 2003 removes servers that have a connection cost of "infinite". Exchange 2007, Exchange 2010, Exchange 2013, and Exchange 2016 remove servers that have a connection cost greater than 500.

<43> Section 3.2.5.8: If the **ObjectId** field is set to zero, Exchange 2013 and Exchange 2016 return ecNone.

<44> Section 3.2.5.9: Exchange 2010 returns 0 for this condition.

<45> Section 3.2.5.9: Exchange 2010 returns 0 for this condition.

<46> Section 3.2.5.12.1: Exchange 2003, Exchange 2007, and Exchange 2010 fail the operation with 0x000004B6 (ecRpcFormat).

<47> Section 3.2.5.12.1: Exchange 2003, Exchange 2007, Exchange 2010, Exchange 2013, and Exchange 2016 use 4096 for the default value.

<48> Section 3.2.5.12.1: Exchange 2003, Exchange 2007, Exchange 2010, Exchange 2013, and Exchange 2016 use 4096 for the maximum value.

<49> Section 3.2.5.13.1: Exchange 2010 returns 0x80070057; Exchange 2013 returns 0x00000000.

<50> Section 3.2.5.13.1: Returned in Exchange 2007 only.

7 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as New, Major, Minor, Editorial, or No change.

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

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

- A document revision that incorporates changes to interoperability requirements or functionality.
- The removal of a document from the documentation set.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

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

The revision class **No change** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the technical content of the document is identical to the last released version.

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

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

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

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

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

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
2.2.1.1.1 RopLogon ROP Request Buffer	Clarified product applicability.	Y	Product behavior note updated.
2.2.1.1.1 RopLogon ROP Request Buffer	Clarified product applicability.	Y	Product behavior note updated.
2.2.1.1.4 RopLogon ROP Success Response Buffer for Public Folders	Clarified product applicability.	Y	Product behavior note updated.
2.2.1.5 RopGetStoreState ROP	Clarified product applicability.	Y	Product behavior note updated.
2.2.1.7 RopPublicFolderIsGhosted ROP	Clarified product applicability.	Y	Product behavior note updated.
2.2.2.1.2.1 PidTagComment Property	Clarified product applicability.	Y	Product behavior note updated.
2.2.2.1.2.2 PidTagDeleteAfterSubmit Property	Clarified product applicability.	Y	Product behavior note updated.
2.2.2.1.2.3 PidTagDisplayName Property	Clarified product applicability.	Y	Product behavior note updated.
2.2.2.1.2.3 PidTagDisplayName Property	Added behavior note regarding this property.	Y	New product behavior note added.
3.1.4.1 Logging on to a Store	Clarified product applicability.	Y	Product behavior note updated.
3.1.4.1 Logging on to a Store	Clarified product applicability.	Y	Product behavior

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
			note updated.
3.2.5.1.1 Private Mailbox Logon	Updated description for the return value when user does not match the owner of mailbox.	N	Content update.
3.2.5.1.1 Private Mailbox Logon	Clarified product applicability.	Y	Product behavior note updated.
3.2.5.1.1 Private Mailbox Logon	Clarified product applicability.	Y	Product behavior note updated.
3.2.5.1.1 Private Mailbox Logon	Added behavior note regarding client log on attempts.	Y	New product behavior note added.
3.2.5.1.1 Private Mailbox Logon	Added behavior note regarding offline mailboxes.	Y	New product behavior note added.
3.2.5.1.1 Private Mailbox Logon	Added behavior note regarding the USE_PER_MBD_REPLID_MAPPING flag.	Y	New product behavior note added.
3.2.5.1.2 Public Folders Logon	Clarified product applicability.	Y	Product behavior note updated.
3.2.5.6 Receiving a RopGetOwningServers ROP Request	Clarified product applicability.	Y	Product behavior note updated.
3.2.5.7 Receiving a RopPublicFolderIsGhosted ROP Request	Clarified product applicability.	Y	Product behavior note updated.
3.2.5.8 Receiving a RopLongTermIdFromId ROP Request	Clarified product applicability.	Y	Product behavior note updated.
3.2.5.9 Receiving a RopIdFromLongTermId ROP Request	Added behavior note regarding a zero-valued REPLGUID component.	Y	New product behavior note added.
3.2.5.9 Receiving a RopIdFromLongTermId ROP Request	Added description regarding a zero-valued global counter component.	Y	Content update.
3.2.5.13.1 Behavior Common to Both Private Mailbox and Public Folder Logon	Added behavior note regarding malformed data.	Y	New product behavior note added.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
3.2.5.13.1 Behavior Common to Both Private Mailbox and Public Folder Logon	Added behavior note regarding ecFmtError error code.	Y	New product behavior note added.
6 Appendix A: Product Behavior	Updated list of applicable products.	Y	Content update.

8 Index

S

T

V