

## [MS-OXCROPS]:

# Remote Operations (ROP) List and Encoding Protocol

---

### Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

**Preliminary Documentation.** This Open Specification provides documentation for past and current releases and/or for the pre-release version of this technology. This Open Specification is final documentation for past or current releases as specifically noted in the document, as applicable; it is preliminary documentation for the pre-release versions. Microsoft will release final documentation in connection with the commercial release of the updated or new version of this technology. As the documentation may change between this preliminary version and the final version of this technology, there are risks in relying on preliminary documentation. To the extent that you incur additional

development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

Preliminary

## Revision Summary

Date	Revision History	Revision Class	Comments
4/4/2008	0.1		Initial Availability.
6/27/2008	1.0		Initial Release.
8/6/2008	1.01		Revised and edited technical content.
9/3/2008	1.02		Revised and edited technical content.
12/3/2008	1.03		Revised and edited technical content.
3/4/2009	1.04		Revised and edited technical content.
4/10/2009	2.0		Updated technical content and applicable 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	7.1	Minor	Clarified the meaning of the technical content.
3/18/2011	8.0	Major	Significantly changed the technical content.
8/5/2011	8.1	Minor	Clarified the meaning of the technical content.
10/7/2011	9.0	Major	Significantly changed the technical content.
1/20/2012	10.0	Major	Significantly changed the technical content.
4/27/2012	11.0	Major	Significantly changed the technical content.
7/16/2012	12.0	Major	Significantly changed the technical content.
10/8/2012	12.1	Minor	Clarified the meaning of the technical content.
2/11/2013	13.0	Major	Significantly changed the technical content.
7/26/2013	13.1	Minor	Clarified the meaning of the technical content.
11/18/2013	14.0	Major	Significantly changed the technical content.
2/10/2014	14.0	No Change	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	14.0	No Change	No changes to the meaning, language, or formatting of the technical content.
7/31/2014	15.0	Major	Significantly changed the technical content.
10/30/2014	15.1	Minor	Clarified the meaning of the technical content.
3/16/2015	16.0	Major	Significantly changed the technical content.

Date	Revision History	Revision Class	Comments
5/26/2015	16.1	Minor	Clarified the meaning of the technical content.

Preliminary

# Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>15</b>
1.1	Glossary .....	15
1.2	References .....	18
1.2.1	Normative References .....	19
1.2.2	Informative References .....	19
1.3	Overview .....	19
1.3.1	Server Objects .....	20
1.3.2	Format of ROP Buffers.....	20
1.4	Relationship to Other Protocols .....	20
1.5	Prerequisites/Preconditions .....	21
1.6	Applicability Statement .....	21
1.7	Versioning and Capability Negotiation .....	21
1.8	Vendor-Extensible Fields .....	21
1.9	Standards Assignments.....	21
<b>2</b>	<b>Messages.....</b>	<b>22</b>
2.1	Transport .....	22
2.2	Message Syntax .....	22
2.2.1	ROP Input and Output Buffers .....	22
2.2.2	The Table of RopIds .....	22
2.2.3	Logon ROPs .....	31
2.2.3.1	RopLogon ROP.....	31
2.2.3.1.1	RopLogon ROP Request Buffer.....	31
2.2.3.1.2	RopLogon ROP Success Response Buffer for Private Mailboxes .....	32
2.2.3.1.2.1	LogonTime Structure .....	34
2.2.3.1.3	RopLogon ROP Success Response Buffer for Public Folders .....	34
2.2.3.1.4	RopLogon ROP Redirect Response Buffer .....	36
2.2.3.1.5	RopLogon ROP Failure Response Buffer .....	36
2.2.3.2	RopGetReceiveFolder ROP.....	37
2.2.3.2.1	RopGetReceiveFolder ROP Request Buffer.....	37
2.2.3.2.2	RopGetReceiveFolder ROP Success Response Buffer.....	37
2.2.3.2.3	RopGetReceiveFolder ROP Failure Response Buffer .....	38
2.2.3.3	RopSetReceiveFolder ROP .....	38
2.2.3.3.1	RopSetReceiveFolder ROP Request Buffer .....	38
2.2.3.3.2	RopSetReceiveFolder ROP Response Buffer .....	39
2.2.3.4	RopGetReceiveFolderTable ROP .....	39
2.2.3.4.1	RopGetReceiveFolderTable ROP Request Buffer .....	39
2.2.3.4.2	RopGetReceiveFolderTable ROP Success Response Buffer .....	40
2.2.3.4.3	RopGetReceiveFolderTable ROP Failure Response Buffer .....	40
2.2.3.5	RopGetStoreState ROP .....	41
2.2.3.5.1	RopGetStoreStage ROP Request Buffer .....	41
2.2.3.5.2	RopGetStoreState ROP Success Response Buffer .....	41
2.2.3.5.3	RopGetStoreState ROP Failure Response Buffer .....	41
2.2.3.6	RopGetOwningServers ROP .....	42
2.2.3.6.1	RopGetOwningServers ROP Request Buffer .....	42
2.2.3.6.2	RopGetOwningServers ROP Success Response Buffer .....	42
2.2.3.6.3	RopGetOwningServers ROP Failure Response Buffer .....	43
2.2.3.7	RopPublicFolderIsGhosed ROP .....	44
2.2.3.7.1	RopPublicFolderIsGhosed ROP Request Buffer .....	44
2.2.3.7.2	RopPublicFolderIsGhosed ROP Success Response Buffer .....	44
2.2.3.7.3	RopPublicFolderIsGhosed ROP Failure Response Buffer.....	45
2.2.3.8	RopLongTermIdFromId ROP .....	45
2.2.3.8.1	RopLongTermIdFromId ROP Request Buffer .....	45

2.2.3.8.2	RopLongTermIdFromId ROP Success Response Buffer .....	46
2.2.3.8.3	RopLongTermIdFromId ROP Failure Response Buffer .....	46
2.2.3.9	RopIdFromLongTermId ROP .....	47
2.2.3.9.1	RopIdFromLongTermId ROP Request Buffer .....	47
2.2.3.9.2	RopIdFromLongTermId ROP Success Response Buffer .....	48
2.2.3.9.3	RopIdFromLongTermId ROP Failure Response Buffer .....	48
2.2.3.10	RopGetPerUserLongTermIds ROP .....	49
2.2.3.10.1	RopGetPerUserLongTermIds ROP Request Buffer .....	49
2.2.3.10.2	RopGetPerUserLongTermIds ROP Success Response Buffer .....	49
2.2.3.10.3	RopGetPerUserLongTermIds ROP Failure Response Buffer .....	50
2.2.3.11	RopGetPerUserGuid ROP .....	50
2.2.3.11.1	RopGetPerUserGuid ROP Request Buffer .....	50
2.2.3.11.2	RopGetPerUserGuid ROP Success Response Buffer .....	51
2.2.3.11.3	RopGetPerUserGuid ROP Failure Response Buffer .....	51
2.2.3.12	RopReadPerUserInformation ROP .....	52
2.2.3.12.1	RopReadPerUserInformation ROP Request Buffer .....	52
2.2.3.12.2	RopReadPerUserInformation ROP Success Response Buffer .....	53
2.2.3.12.3	RopReadPerUserInformation ROP Failure Response Buffer .....	53
2.2.3.13	RopWritePerUserInformation ROP .....	54
2.2.3.13.1	RopWritePerUserInformation ROP Request Buffer .....	54
2.2.3.13.2	RopWritePerUserInformation ROP Response Buffer .....	55
2.2.4	Folder ROPs .....	56
2.2.4.1	RopOpenFolder ROP .....	56
2.2.4.1.1	RopOpenFolder ROP Request Buffer .....	56
2.2.4.1.2	RopOpenFolder ROP Success Response Buffer .....	56
2.2.4.1.3	RopOpenFolder ROP Failure Response Buffer .....	57
2.2.4.2	RopCreateFolder ROP .....	57
2.2.4.2.1	RopCreateFolder ROP Request Buffer .....	57
2.2.4.2.2	RopCreateFolder ROP Success Response Buffer .....	58
2.2.4.2.3	RopCreateFolder ROP Failure Response Buffer .....	59
2.2.4.3	RopDeleteFolder ROP .....	60
2.2.4.3.1	RopDeleteFolder ROP Request Buffer .....	60
2.2.4.3.2	RopDeleteFolder ROP Response Buffer .....	60
2.2.4.4	RopSetSearchCriteria ROP .....	61
2.2.4.4.1	RopSetSearchCriteria ROP Request Buffer .....	61
2.2.4.4.2	RopSetSearchCriteria ROP Response Buffer .....	62
2.2.4.5	RopGetSearchCriteria ROP .....	62
2.2.4.5.1	RopGetSearchCriteria ROP Request Buffer .....	62
2.2.4.5.2	RopGetSearchCriteria ROP Success Response Buffer .....	63
2.2.4.5.3	RopGetSearchCriteria ROP Failure Response Buffer .....	64
2.2.4.6	RopMoveCopyMessages ROP .....	64
2.2.4.6.1	RopMoveCopyMessages ROP Request Buffer .....	64
2.2.4.6.2	RopMoveCopyMessages ROP Response Buffer .....	65
2.2.4.6.3	RopMoveCopyMessages ROP Null Destination Failure Response Buffer .....	65
2.2.4.7	RopMoveFolder ROP .....	66
2.2.4.7.1	RopMoveFolder ROP Request Buffer .....	66
2.2.4.7.2	RopMoveFolder ROP Response Buffer .....	67
2.2.4.7.3	RopMoveFolder ROP Null Destination Failure Response Buffer .....	67
2.2.4.8	RopCopyFolder ROP .....	68
2.2.4.8.1	RopCopyFolder ROP Request Buffer .....	68
2.2.4.8.2	RopCopyFolder ROP Response Buffer .....	68
2.2.4.8.3	RopCopyFolder ROP Null Destination Failure Response Buffer .....	69
2.2.4.9	RopEmptyFolder ROP .....	69
2.2.4.9.1	RopEmptyFolder ROP Request Buffer .....	69
2.2.4.9.2	RopEmptyFolder ROP Response Buffer .....	70
2.2.4.10	RopHardDeleteMessagesAndSubfolders ROP .....	70
2.2.4.10.1	RopHardDeleteMessagesAndSubfolders ROP Request Buffer .....	70

2.2.4.10.2	RopHardDeleteMessagesAndSubfolders ROP Response Buffer.....	71
2.2.4.11	RopDeleteMessages ROP.....	71
2.2.4.11.1	RopDeleteMessages ROP Request Buffer.....	71
2.2.4.11.2	RopDeleteMessages ROP Response Buffer.....	72
2.2.4.12	RopHardDeleteMessages ROP.....	72
2.2.4.12.1	RopHardDeleteMessages ROP Request Buffer.....	73
2.2.4.12.2	RopHardDeleteMessages ROP Response Buffer.....	73
2.2.4.13	RopGetHierarchyTable ROP.....	74
2.2.4.13.1	RopGetHierarchyTable ROP Request Buffer.....	74
2.2.4.13.2	RopGetHierarchyTable ROP Success Response Buffer.....	74
2.2.4.13.3	RopGetHierarchyTable ROP Failure Response Buffer.....	75
2.2.4.14	RopGetContentsTable ROP.....	75
2.2.4.14.1	RopGetContentsTable ROP Request Buffer.....	75
2.2.4.14.2	RopGetContentsTable ROP Success Response Buffer.....	76
2.2.4.14.3	RopGetContentsTable ROP Failure Response Buffer.....	76
2.2.5	Table ROPs.....	77
2.2.5.1	RopSetColumns ROP.....	77
2.2.5.1.1	RopSetColumns ROP Request Buffer.....	77
2.2.5.1.2	RopSetColumns ROP Success Response Buffer.....	77
2.2.5.1.3	RopSetColumns ROP Failure Response Buffer.....	78
2.2.5.2	RopSortTable ROP.....	78
2.2.5.2.1	RopSortTable ROP Request Buffer.....	78
2.2.5.2.2	RopSortTable ROP Success Response Buffer.....	79
2.2.5.2.3	RopSortTable ROP Failure Response Buffer.....	79
2.2.5.3	RopRestrict ROP.....	80
2.2.5.3.1	RopRestrict ROP Request Buffer.....	80
2.2.5.3.2	RopRestrict ROP Success Response Buffer.....	80
2.2.5.3.3	RopRestrict ROP Failure Response Buffer.....	81
2.2.5.4	RopQueryRows ROP.....	81
2.2.5.4.1	RopQueryRows ROP Request Buffer.....	81
2.2.5.4.2	RopQueryRows ROP Success Response Buffer.....	82
2.2.5.4.3	RopQueryRows ROP Failure Response Buffer.....	82
2.2.5.5	RopAbort ROP.....	83
2.2.5.5.1	RopAbort ROP Request Buffer.....	83
2.2.5.5.2	RopAbort ROP Success Response Buffer.....	83
2.2.5.5.3	RopAbort ROP Failure Response Buffer.....	84
2.2.5.6	RopGetStatus ROP.....	84
2.2.5.6.1	RopGetStatus ROP Request Buffer.....	84
2.2.5.6.2	RopGetStatus ROP Success Response Buffer.....	84
2.2.5.6.3	RopGetStatus ROP Failure Response Buffer.....	85
2.2.5.7	RopQueryPosition ROP.....	85
2.2.5.7.1	RopQueryPosition ROP Request Buffer.....	85
2.2.5.7.2	RopQueryPosition ROP Success Response Buffer.....	86
2.2.5.7.3	RopQueryPosition ROP Failure Response Buffer.....	86
2.2.5.8	RopSeekRow ROP.....	87
2.2.5.8.1	RopSeekRow ROP Request Buffer.....	87
2.2.5.8.2	RopSeekRow ROP Success Response Buffer.....	87
2.2.5.8.3	RopSeekRow ROP Failure Response Buffer.....	88
2.2.5.9	RopSeekRowBookmark ROP.....	88
2.2.5.9.1	RopSeekRowBookmark ROP Request Buffer.....	88
2.2.5.9.2	RopSeekRowBookmark ROP Success Response Buffer.....	89
2.2.5.9.3	RopSeekRowBookmark ROP Failure Response Buffer.....	89
2.2.5.10	RopSeekRowFractional ROP.....	90
2.2.5.10.1	RopSeekRowFractional ROP Request Buffer.....	90
2.2.5.10.2	RopSeekRowFractional ROP Response Buffer.....	90
2.2.5.11	RopCreateBookmark ROP.....	91
2.2.5.11.1	RopCreateBookmark ROP Request Buffer.....	91

2.2.5.11.2	RopCreateBookmark ROP Success Response Buffer	91
2.2.5.11.3	RopCreateBookmark ROP Failure Response Buffer	92
2.2.5.12	RopQueryColumnsAll ROP	92
2.2.5.12.1	RopQueryColumnsAll ROP Request Buffer	92
2.2.5.12.2	RopQueryColumnsAll ROP Success Response Buffer	93
2.2.5.12.3	RopQueryColumnsAll ROP Failure Response Buffer	93
2.2.5.13	RopFindRow ROP	94
2.2.5.13.1	RopFindRow ROP Request Buffer	94
2.2.5.13.2	RopFindRow ROP Success Response Buffer	94
2.2.5.13.3	RopFindRow ROP Failure Response Buffer	95
2.2.5.14	RopFreeBookmark ROP	95
2.2.5.14.1	RopFreeBookmark ROP Request Buffer	96
2.2.5.14.2	RopFreeBookmark ROP Response Buffer	96
2.2.5.15	RopResetTable ROP	96
2.2.5.15.1	RopResetTable ROP Request Buffer	96
2.2.5.15.2	RopResetTable ROP Response Buffer	97
2.2.5.16	RopExpandRow ROP	97
2.2.5.16.1	RopExpandRow ROP Request Buffer	97
2.2.5.16.2	RopExpandRow ROP Success Response Buffer	98
2.2.5.16.3	RopExpandRow ROP Failure Response Buffer	98
2.2.5.17	RopCollapseRow ROP	99
2.2.5.17.1	RopCollapseRow ROP Request Buffer	99
2.2.5.17.2	RopCollapseRow ROP Success Response Buffer	99
2.2.5.17.3	RopCollapseRow ROP Failure Response Buffer	100
2.2.5.18	RopGetCollapseState ROP	100
2.2.5.18.1	RopGetCollapseState ROP Request Buffer	100
2.2.5.18.2	RopGetCollapseState ROP Success Response Buffer	101
2.2.5.18.3	RopGetCollapseState ROP Failure Response Buffer	101
2.2.5.19	RopSetCollapseState ROP	102
2.2.5.19.1	RopSetCollapseState ROP Request Buffer	102
2.2.5.19.2	RopSetCollapseState ROP Success Response Buffer	102
2.2.5.19.3	RopSetCollapseState ROP Failure Response Buffer	103
2.2.6	Message ROPs	103
2.2.6.1	RopOpenMessage ROP	103
2.2.6.1.1	RopOpenMessage ROP Request Buffer	104
2.2.6.1.2	RopOpenMessage ROP Success Response Buffer	104
2.2.6.1.2.1	OpenRecipientRow Structure	105
2.2.6.1.3	RopOpenMessage ROP Failure Response Buffer	106
2.2.6.2	RopCreateMessage ROP	106
2.2.6.2.1	RopCreateMessage ROP Request Buffer	106
2.2.6.2.2	RopCreateMessage ROP Success Response Buffer	107
2.2.6.2.3	RopCreateMessage ROP Failure Response Buffer	108
2.2.6.3	RopSaveChangesMessage ROP	108
2.2.6.3.1	RopSaveChangesMessage ROP Request Buffer	108
2.2.6.3.2	RopSaveChangesMessage ROP Success Response Buffer	109
2.2.6.3.3	RopSaveChangesMessage ROP Failure Response Buffer	109
2.2.6.4	RopRemoveAllRecipients ROP	110
2.2.6.4.1	RopRemoveAllRecipients ROP Request Buffer	110
2.2.6.4.2	RopRemoveAllRecipients ROP Response Buffer	110
2.2.6.5	RopModifyRecipients ROP	110
2.2.6.5.1	RopModifyRecipients ROP Request Buffer	110
2.2.6.5.1.1	ModifyRecipientRow Structure	111
2.2.6.5.2	RopModifyRecipients ROP Response Buffer	112
2.2.6.6	RopReadRecipients ROP	112
2.2.6.6.1	RopReadRecipients ROP Request Buffer	112
2.2.6.6.2	RopReadRecipients ROP Success Response Buffer	113
2.2.6.6.2.1	ReadRecipientRow Structure	113



2.2.6.6.3	RopReadRecipients ROP Failure Response Buffer .....	114
2.2.6.7	RopReloadCachedInformation ROP .....	114
2.2.6.7.1	RopReloadCachedInformation ROP Request Buffer .....	114
2.2.6.7.2	RopReloadCachedInformation ROP Success Response Buffer .....	115
2.2.6.7.3	RopReloadCachedInformation ROP Failure Response Buffer .....	116
2.2.6.8	RopSetMessageStatus ROP .....	116
2.2.6.8.1	RopSetMessageStatus ROP Request Buffer .....	116
2.2.6.8.2	RopSetMessageStatus ROP Success Response Buffer .....	117
2.2.6.8.3	RopSetMessageStatus ROP Failure Response Buffer .....	117
2.2.6.9	RopGetMessageStatus ROP .....	118
2.2.6.9.1	RopGetMessageStatus ROP Request Buffer .....	118
2.2.6.9.2	RopGetMessageStatus ROP Response Buffers .....	118
2.2.6.10	RopSetReadFlags ROP .....	119
2.2.6.10.1	RopSetReadFlags ROP Request Buffer .....	119
2.2.6.10.2	RopSetReadFlags ROP Response Buffer .....	119
2.2.6.11	RopSetMessageReadFlag ROP .....	120
2.2.6.11.1	RopSetMessageReadFlag ROP Request Buffer .....	120
2.2.6.11.2	RopSetMessageReadFlag ROP Success Response Buffer .....	121
2.2.6.11.3	RopSetMessageReadFlag ROP Failure Response Buffer .....	121
2.2.6.12	RopOpenAttachment ROP .....	122
2.2.6.12.1	RopOpenAttachment ROP Request Buffer .....	122
2.2.6.12.2	RopOpenAttachment ROP Response Buffer .....	122
2.2.6.13	RopCreateAttachment ROP .....	123
2.2.6.13.1	RopCreateAttachment ROP Request Buffer .....	123
2.2.6.13.2	RopCreateAttachment ROP Success Response Buffer .....	123
2.2.6.13.3	RopCreateAttachment ROP Failure Response Buffer .....	124
2.2.6.14	RopDeleteAttachment ROP .....	124
2.2.6.14.1	RopDeleteAttachment ROP Request Buffer .....	124
2.2.6.14.2	RopDeleteAttachment ROP Response Buffer .....	125
2.2.6.15	RopSaveChangesAttachment ROP .....	125
2.2.6.15.1	RopSaveChangesAttachment ROP Request Buffer .....	125
2.2.6.15.2	RopSaveChangesAttachment ROP Response Buffer .....	126
2.2.6.16	RopOpenEmbeddedMessage ROP .....	126
2.2.6.16.1	RopOpenEmbeddedMessage ROP Request Buffer .....	126
2.2.6.16.2	RopOpenEmbeddedMessage ROP Success Response Buffer .....	127
2.2.6.16.3	RopOpenEmbeddedMessage ROP Failure Response Buffer .....	128
2.2.6.17	RopGetAttachmentTable ROP .....	128
2.2.6.17.1	RopGetAttachmentTable ROP Request Buffer .....	129
2.2.6.17.2	RopGetAttachmentTable ROP Response Buffer .....	129
2.2.6.18	RopGetValidAttachments ROP .....	129
2.2.6.18.1	RopGetValidAttachments ROP Request Buffer .....	129
2.2.6.18.2	RopGetValidAttachments ROP Success Response Buffer .....	130
2.2.6.18.3	RopGetValidAttachments ROP Failure Response Buffer .....	130
2.2.7	Transport ROPs .....	131
2.2.7.1	RopSubmitMessage ROP .....	131
2.2.7.1.1	RopSubmitMessage ROP Request Buffer .....	131
2.2.7.1.2	RopSubmitMessage ROP Response Buffer .....	131
2.2.7.2	RopAbortSubmit ROP .....	132
2.2.7.2.1	RopAbortSubmit ROP Request Buffer .....	132
2.2.7.2.2	RopAbortSubmit ROP Response Buffer .....	132
2.2.7.3	RopGetAddressTypes ROP .....	133
2.2.7.3.1	RopGetAddressTypes ROP Request Buffer .....	133
2.2.7.3.2	RopGetAddressTypes ROP Success Response Buffer .....	133
2.2.7.3.3	RopGetAddressTypes ROP Failure Response Buffer .....	134
2.2.7.4	RopSetSpooler ROP .....	134
2.2.7.4.1	RopSetSpooler ROP Request Buffer .....	134
2.2.7.4.2	RopSetSpooler ROP Response Buffer .....	135

2.2.7.5	RopSpoolerLockMessage ROP .....	135
2.2.7.5.1	RopSpoolerLockMessage ROP Request Buffer .....	135
2.2.7.5.2	RopSpoolerLockMessage ROP Response Buffer .....	136
2.2.7.6	RopTransportSend ROP .....	136
2.2.7.6.1	RopTransportSend ROP Request Buffer .....	136
2.2.7.6.2	RopTransportSend ROP Success Response Buffer .....	136
2.2.7.6.3	RopTransportSend ROP Failure Response Buffer .....	137
2.2.7.7	RopTransportNewMail ROP .....	138
2.2.7.7.1	RopTransportNewMail ROP Request Buffer .....	138
2.2.7.7.2	RopTransportNewMail ROP Response Buffer .....	138
2.2.7.8	RopGetTransportFolder ROP .....	139
2.2.7.8.1	RopGetTransportFolder ROP Request Buffer .....	139
2.2.7.8.2	RopGetTransportFolder ROP Success Response Buffer .....	139
2.2.7.8.3	RopGetTransportFolder ROP Failure Response Buffer .....	140
2.2.7.9	RopOptionsData ROP .....	140
2.2.7.9.1	RopOptionsData ROP Request Buffer .....	140
2.2.7.9.2	RopOptionsData ROP Success Response Buffer .....	141
2.2.7.9.3	RopOptionsData ROP Failure Response Buffer .....	142
2.2.8	Property ROPs .....	142
2.2.8.1	RopGetPropertyIdsFromNames ROP .....	142
2.2.8.1.1	RopGetPropertyIdsFromNames ROP Request Buffer .....	142
2.2.8.1.2	RopGetPropertyIdsFromNames ROP Success Response Buffer .....	143
2.2.8.1.3	RopGetPropertyIdsFromNames ROP Failure Response Buffer .....	143
2.2.8.2	RopGetNamesFromPropertyIds ROP .....	144
2.2.8.2.1	RopGetNamesFromPropertyIds ROP Request Buffer .....	144
2.2.8.2.2	RopGetNamesFromPropertyIds ROP Success Response Buffer .....	144
2.2.8.2.3	RopGetNamesFromPropertyIds ROP Failure Response Buffer .....	145
2.2.8.3	RopGetPropertiesSpecific ROP .....	145
2.2.8.3.1	RopGetPropertiesSpecific ROP Request Buffer .....	145
2.2.8.3.2	RopGetPropertiesSpecific ROP Success Response Buffer .....	146
2.2.8.3.3	RopGetPropertiesSpecific ROP Failure Response Buffer .....	147
2.2.8.4	RopGetPropertiesAll ROP .....	147
2.2.8.4.1	RopGetPropertiesAll ROP Request Buffer .....	147
2.2.8.4.2	RopGetPropertiesAll ROP Success Response Buffer .....	148
2.2.8.4.3	RopGetPropertiesAll ROP Failure Response Buffer .....	148
2.2.8.5	RopGetPropertiesList ROP .....	149
2.2.8.5.1	RopGetPropertiesList ROP Request Buffer .....	149
2.2.8.5.2	RopGetPropertiesList ROP Success Response Buffer .....	149
2.2.8.5.3	RopGetPropertiesList ROP Failure Response Buffer .....	150
2.2.8.6	RopSetProperties ROP .....	150
2.2.8.6.1	RopSetProperties ROP Request Buffer .....	150
2.2.8.6.2	RopSetProperties ROP Success Response Buffer .....	151
2.2.8.6.3	RopSetProperties ROP Failure Response Buffer .....	151
2.2.8.7	RopSetPropertiesNoReplicate ROP .....	152
2.2.8.7.1	RopSetPropertiesNoReplicate ROP Request Buffer .....	152
2.2.8.7.2	RopSetPropertiesNoReplicate ROP Success Response Buffer .....	152
2.2.8.7.3	RopSetPropertiesNoReplicate ROP Failure Response Buffer .....	153
2.2.8.8	RopDeleteProperties ROP .....	153
2.2.8.8.1	RopDeleteProperties ROP Request Buffer .....	153
2.2.8.8.2	RopDeleteProperties ROP Success Response Buffer .....	154
2.2.8.8.3	RopDeleteProperties ROP Failure Response Buffer .....	155
2.2.8.9	RopDeletePropertiesNoReplicate ROP .....	155
2.2.8.9.1	RopDeletePropertiesNoReplicate ROP Request Buffer .....	155
2.2.8.9.2	RopDeletePropertiesNoReplicate ROP Success Response Buffer .....	156
2.2.8.9.3	RopDeletePropertiesNoReplicate ROP Failure Response Buffer .....	156
2.2.8.10	RopQueryNamedProperties ROP .....	157
2.2.8.10.1	RopQueryNamedProperties ROP Request Buffer .....	157

2.2.8.10.2	RopQueryNamedProperties ROP Success Response Buffer .....	157
2.2.8.10.3	RopQueryNamedProperties ROP Failure Response Buffer .....	158
2.2.8.11	RopCopyProperties ROP .....	159
2.2.8.11.1	RopCopyProperties ROP Request Buffer .....	159
2.2.8.11.2	RopCopyProperties ROP Success Response Buffer .....	159
2.2.8.11.3	RopCopyProperties ROP Null Destination Failure Response Buffer .....	160
2.2.8.11.4	RopCopyProperties ROP Failure Response Buffer .....	160
2.2.8.12	RopCopyTo ROP .....	161
2.2.8.12.1	RopCopyTo ROP Request Buffer .....	161
2.2.8.12.2	RopCopyTo ROP Success Response Buffer .....	162
2.2.8.12.3	RopCopyTo ROP Null Destination Failure Response Buffer .....	162
2.2.8.12.4	RopCopyTo ROP Failure Response Buffer .....	163
2.2.8.13	RopProgress ROP .....	163
2.2.8.13.1	RopProgress ROP Request Buffer .....	163
2.2.8.13.2	RopProgress ROP Success Response Buffer .....	164
2.2.8.13.3	RopProgress ROP Failure Response Buffer .....	164
2.2.9	Stream ROPs .....	165
2.2.9.1	RopOpenStream ROP .....	165
2.2.9.1.1	RopOpenStream ROP Request Buffer .....	165
2.2.9.1.2	RopOpenStream ROP Success Response Buffer .....	165
2.2.9.1.3	RopOpenStream ROP Failure Response Buffer .....	166
2.2.9.2	RopReadStream ROP .....	166
2.2.9.2.1	RopReadStream ROP Request Buffer .....	166
2.2.9.2.2	RopReadStream ROP Response Buffer .....	167
2.2.9.3	RopWriteStream ROP .....	167
2.2.9.3.1	RopWriteStream ROP Request Buffer .....	167
2.2.9.3.2	RopWriteStream ROP Response Buffer .....	168
2.2.9.4	RopCommitStream ROP .....	168
2.2.9.4.1	RopCommitStream ROP Request Buffer .....	168
2.2.9.4.2	RopCommitStream ROP Response Buffer .....	169
2.2.9.5	RopGetStreamSize ROP .....	169
2.2.9.5.1	RopGetStreamSize ROP Request Buffer .....	169
2.2.9.5.2	RopGetStreamSize ROP Success Response Buffer .....	170
2.2.9.5.3	RopGetStreamSize ROP Failure Response Buffer .....	170
2.2.9.6	RopSetStreamSize ROP .....	170
2.2.9.6.1	RopSetStreamSize ROP Request Buffer .....	170
2.2.9.6.2	RopSetStreamSize ROP Response Buffer .....	171
2.2.9.7	RopSeekStream ROP .....	171
2.2.9.7.1	RopSeekStream ROP Request Buffer .....	171
2.2.9.7.2	RopSeekStream ROP Success Response Buffer .....	172
2.2.9.7.3	RopSeekStream ROP Failure Response Buffer .....	172
2.2.9.8	RopCopyToStream ROP .....	173
2.2.9.8.1	RopCopyToStream ROP Request Buffer .....	173
2.2.9.8.2	RopCopyToStream ROP Response Buffer .....	173
2.2.9.8.3	RopCopyToStream ROP Null Destination Failure Response Buffer .....	174
2.2.9.9	RopLockRegionStream ROP .....	175
2.2.9.9.1	RopLockRegionStream ROP Request Buffer .....	175
2.2.9.9.2	RopLockRegionStream ROP Response Buffer .....	176
2.2.9.10	RopUnlockRegionStream ROP .....	176
2.2.9.10.1	RopUnlockRegionStream ROP Request Buffer .....	176
2.2.9.10.2	RopUnlockRegionStream ROP Response Buffer .....	177
2.2.9.11	RopWriteAndCommitStream ROP .....	177
2.2.9.11.1	RopWriteAndCommitStream ROP Request Buffer .....	177
2.2.9.11.2	RopWriteAndCommitStream ROP Response Buffer .....	178
2.2.9.12	RopCloneStream ROP .....	178
2.2.9.12.1	RopCloneStream ROP Request Buffer .....	178
2.2.9.12.2	RopCloneStream ROP Response Buffer .....	178

2.2.10	Permission ROPs.....	179
2.2.10.1	RopModifyPermissions ROP.....	179
2.2.10.1.1	RopModifyPermissions ROP Request Buffer .....	179
2.2.10.1.1.1	PermissionData Structure .....	179
2.2.10.1.2	RopModifyPermissions ROP Response Buffer .....	180
2.2.10.2	RopGetPermissionsTable ROP .....	180
2.2.10.2.1	RopGetPermissionsTable ROP Request Buffer .....	180
2.2.10.2.2	RopGetPermissionsTable ROP Response Buffer .....	181
2.2.11	Rule ROPs.....	181
2.2.11.1	RopModifyRules ROP .....	181
2.2.11.1.1	RopModifyRules ROP Request Buffer .....	181
2.2.11.1.1.1	RuleData Structure .....	182
2.2.11.1.2	RopModifyRules ROP Response Buffer .....	182
2.2.11.2	RopGetRulesTable ROP .....	182
2.2.11.2.1	RopGetRulesTable ROP Request Buffer .....	183
2.2.11.2.2	RopGetRulesTable ROP Response Buffer .....	183
2.2.11.3	RopUpdateDeferredActionMessages ROP .....	183
2.2.11.3.1	RopUpdateDeferredActionMessages ROP Request Buffer .....	183
2.2.11.3.2	RopUpdateDeferredActionMessages ROP Response Buffer .....	184
2.2.12	Fast Transfer ROPs .....	185
2.2.12.1	RopFastTransferDestinationConfigure ROP.....	185
2.2.12.1.1	RopFastTransferDestinationConfigure ROP Request Buffer .....	185
2.2.12.1.2	RopFastTransferDestinationConfigure ROP Response Buffer .....	185
2.2.12.2	RopFastTransferDestinationPutBuffer ROP .....	186
2.2.12.2.1	RopFastTransferDestinationPutBuffer ROP Request Buffer .....	186
2.2.12.2.2	RopFastTransferDestinationPutBuffer ROP Response Buffer .....	186
2.2.12.3	RopFastTransferSourceGetBuffer ROP.....	187
2.2.12.3.1	RopFastTransferSourceGetBuffer ROP Request Buffer.....	187
2.2.12.3.2	RopFastTransferSourceGetBuffer ROP Response Buffer.....	187
2.2.12.4	RopFastTransferSourceCopyFolder ROP.....	188
2.2.12.4.1	RopFastTransferSourceCopyFolder ROP Request Buffer .....	188
2.2.12.4.2	RopFastTransferSourceCopyFolder ROP Response Buffer .....	189
2.2.12.5	RopFastTransferSourceCopyMessages ROP.....	189
2.2.12.5.1	RopFastTransferSourceCopyMessages ROP Request Buffer.....	189
2.2.12.5.2	RopFastTransferSourceCopyMessages ROP Response Buffer .....	190
2.2.12.6	RopFastTransferSourceCopyTo ROP.....	191
2.2.12.6.1	RopFastTransferSourceCopyTo ROP Request Buffer.....	191
2.2.12.6.2	RopFastTransferSourceCopyTo ROP Response Buffer .....	191
2.2.12.7	RopFastTransferSourceCopyProperties ROP .....	192
2.2.12.7.1	RopFastTransferSourceCopyProperties ROP Request Buffer .....	192
2.2.12.7.2	RopFastTransferSourceCopyProperties ROP Response Buffer .....	193
2.2.12.8	RopTellVersion ROP .....	193
2.2.12.8.1	RopTellVersion ROP Request Buffer.....	193
2.2.12.8.2	RopTellVersion ROP Response Buffer.....	194
2.2.13	Incremental Change Synchronization ROPs .....	194
2.2.13.1	RopSynchronizationConfigure ROP .....	194
2.2.13.1.1	RopSynchronizationConfigure ROP Request Buffer .....	194
2.2.13.1.2	RopSynchronizationConfigure ROP Response Buffer .....	195
2.2.13.2	RopSynchronizationImportMessageChange ROP.....	196
2.2.13.2.1	RopSynchronizationImportMessageChange ROP Request Buffer.....	196
2.2.13.2.2	RopSynchronizationImportMessageChange ROP Success Response Buffer .....	197
2.2.13.2.3	RopSynchronizationImportMessageChange ROP Failure Response Buffer.....	197
2.2.13.3	RopSynchronizationImportReadStateChanges ROP .....	197
2.2.13.3.1	RopSynchronizationImportReadStateChanges ROP Request Buffer.....	198
2.2.13.3.1.1	MessageReadState Structure.....	198
2.2.13.3.2	RopSynchronizationImportReadStateChanges ROP Response Buffer.....	198

2.2.13.4	RopSynchronizationImportHierarchyChange ROP .....	199
2.2.13.4.1	RopSynchronizationImportHierarchyChange ROP Request Buffer .....	199
2.2.13.4.2	RopSynchronizationImportHierarchyChange ROP Success Response Buffer .....	200
2.2.13.4.3	RopSynchronizationImportHierarchyChange ROP Failure Response Buffer .....	200
2.2.13.5	RopSynchronizationImportDeletes ROP .....	201
2.2.13.5.1	RopSynchronizationImportDeletes ROP Request Buffer .....	201
2.2.13.5.2	RopSynchronizationImportDeletes ROP Response Buffer .....	201
2.2.13.6	RopSynchronizationImportMessageMove ROP .....	202
2.2.13.6.1	RopSynchronizationImportMessageMove ROP Request Buffer .....	202
2.2.13.6.2	RopSynchronizationImportMessageMove ROP Success Response Buffer .....	203
2.2.13.6.3	RopSynchronizationImportMessageMove ROP Failure Response Buffer .....	204
2.2.13.7	RopSynchronizationOpenCollector ROP .....	204
2.2.13.7.1	RopSynchronizationOpenCollector ROP Request Buffer .....	204
2.2.13.7.2	RopSynchronizationOpenCollector ROP Response Buffer .....	205
2.2.13.8	RopSynchronizationGetTransferState ROP .....	205
2.2.13.8.1	RopSynchronizationGetTransferState ROP Request Buffer .....	205
2.2.13.8.2	RopSynchronizationGetTransferState ROP Response Buffer .....	206
2.2.13.9	RopSynchronizationUploadStateStreamBegin ROP .....	206
2.2.13.9.1	RopSynchronizationUploadStateStreamBegin ROP Request Buffer .....	206
2.2.13.9.2	RopSynchronizationUploadStateStreamBegin ROP Response Buffer .....	207
2.2.13.10	RopSynchronizationUploadStateStreamContinue ROP .....	207
2.2.13.10.1	RopSynchronizationUploadStateStreamContinue ROP Request Buffer .....	207
2.2.13.10.2	RopSynchronizationUploadStateStreamContinue ROP Response Buffer .....	208
2.2.13.11	RopSynchronizationUploadStateStreamEnd ROP .....	208
2.2.13.11.1	RopSynchronizationUploadStateStreamEnd ROP Request Buffer .....	208
2.2.13.11.2	RopSynchronizationUploadStateStreamEnd ROP Response Buffer .....	209
2.2.13.12	RopSetLocalReplicaMidsetDeleted ROP .....	209
2.2.13.12.1	RopSetLocalReplicaMidsetDeleted ROP Request Buffer .....	209
2.2.13.12.1.1	LongTermIdRange Structure .....	210
2.2.13.12.2	RopSetLocalReplicaMidsetDeleted ROP Response Buffer .....	211
2.2.13.13	RopGetLocalReplicaIds ROP .....	211
2.2.13.13.1	RopGetLocalReplicaIds ROP Request Buffer .....	211
2.2.13.13.2	RopGetLocalReplicaIds ROP Success Response Buffer .....	211
2.2.13.13.3	RopGetLocalReplicaIds ROP Failure Response Buffer .....	212
2.2.14	Notification ROPs .....	213
2.2.14.1	RopRegisterNotification ROP .....	213
2.2.14.1.1	RopRegisterNotification ROP Request Buffer .....	213
2.2.14.1.2	RopRegisterNotification ROP Response Buffer .....	214
2.2.14.2	RopNotify ROP .....	214
2.2.14.2.1	RopNotify ROP Response Buffer .....	214
2.2.14.3	RopPending ROP .....	215
2.2.14.3.1	RopPending ROP Response Buffer .....	215
2.2.15	Other ROPs .....	215
2.2.15.1	RopBufferTooSmall ROP .....	215
2.2.15.1.1	RopBufferTooSmall ROP Response Buffer .....	215
2.2.15.2	RopBackoff ROP .....	215
2.2.15.2.1	RopBackoff ROP Response Buffer .....	216
2.2.15.2.1.1	BackoffRop Structure .....	216
2.2.15.3	RopRelease ROP .....	217
2.2.15.3.1	RopRelease ROP Request Buffer .....	217
<b>3</b>	<b>Protocol Details .....</b>	<b>218</b>
3.1	Client Details .....	218
3.1.1	Abstract Data Model .....	218
3.1.2	Timers .....	218

3.1.3	Initialization .....	218
3.1.4	Higher-Layer Triggered Events .....	218
3.1.4.1	Creating a ROP Input Buffer .....	218
3.1.4.2	Logging On .....	219
3.1.5	Message Processing Events and Sequencing Rules .....	219
3.1.5.1	Extra ROP Responses .....	219
3.1.5.1.1	Processing the RopBackoff ROP Response .....	219
3.1.5.1.2	Processing the RopBufferTooSmall Response Buffer .....	219
3.1.5.1.3	Processing the RopNotify and RopPending Responses.....	220
3.1.5.2	Subsequent RopProgress ROP Requests .....	220
3.1.5.3	Server Object Dependencies.....	220
3.1.5.4	Code Page for Strings .....	221
3.1.6	Timer Events.....	221
3.1.7	Other Local Events.....	221
3.1.7.1	Shutting Down.....	221
3.2	Server Details.....	221
3.2.1	Abstract Data Model.....	221
3.2.2	Timers .....	222
3.2.3	Initialization.....	222
3.2.4	Higher-Layer Triggered Events .....	222
3.2.4.1	Sending Notifications to the Client.....	222
3.2.4.2	Sending the RopBackoff ROP Response .....	222
3.2.4.3	Responding to Insufficient Output Buffer Space .....	222
3.2.5	Message Processing Events and Sequencing Rules .....	223
3.2.5.1	Processing a ROP Input Buffer .....	223
3.2.5.2	Creating a ROP Output Buffer .....	223
3.2.5.3	Processing the RopRelease ROP Request .....	224
3.2.5.4	Error Codes Returned When an Object Is Invalid.....	224
3.2.6	Timer Events.....	224
3.2.7	Other Local Events.....	224
3.2.7.1	Disconnecting.....	224
<b>4</b>	<b>Protocol Examples .....</b>	<b>225</b>
4.1	Empty ROP Buffer .....	225
4.2	Single ROP Request.....	225
4.3	Multiple ROP Request.....	225
4.4	RopRelease ROP Request .....	226
4.5	RopBufferTooSmall ROP Response.....	227
4.6	Logon RopBackoff ROP Response .....	227
4.7	RopBackoff ROP Response.....	228
<b>5</b>	<b>Security .....</b>	<b>230</b>
5.1	Security Considerations for Implementers .....	230
5.2	Index of Security Parameters .....	230
<b>6</b>	<b>Appendix A: Product Behavior .....</b>	<b>231</b>
<b>7</b>	<b>Change Tracking.....</b>	<b>234</b>
<b>8</b>	<b>Index.....</b>	<b>236</b>

# 1 Introduction

The Remote Operations (ROP) List and Encoding Protocol provides the **remote operations (ROPs)** used to access and modify **mailbox** information on the server.

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in [\[RFC2119\]](#). Sections 1.5 and 1.9 are also normative but do not contain those terms. All other sections and examples in this specification are informative.

## 1.1 Glossary

The following terms are specific to this document:

**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.

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

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

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

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

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

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

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

**entry ID:** See EntryID.

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

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

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

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

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

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

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

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

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

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

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

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

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



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

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

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

**non-read receipt:** A message that is generated when an email message is deleted at the expiration of a time limit or due to other client-specific criteria.

**permissions table:** A Table object whose rows represent entries in a permissions list for a **Folder object**.

**property ID:** A 16-bit numeric identifier of a specific attribute (1). A property ID does not include any property type information.

**property name:** A string that, in combination with a property set, identifies a **named property**.

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

**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.

**recipient:** (1) An entity that can receive email messages.

(2) An entity that is in an address list, can receive email messages, and contains a set of attributes (1). Each attribute has a set of associated values.

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

(2) A copy of the data that is in a user's **mailbox** at a specific point in time.

**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.

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

**ROP buffer:** A structure containing an array of bytes that encode a **remote operation (ROP)**. The first byte in the buffer identifies the ROP. This byte is followed by ROP-specific fields. Multiple ROP buffers can be packed into a single **remote procedure call (RPC)** request or response.

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

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

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

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

**rule:** A condition or action, or a set of conditions or actions, that performs tasks automatically based on events and values.

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

**search criteria:** A criteria used to determine which messages are included in a folder with specific characteristics. It is composed of a restriction, which is the filter to be applied, and a search scope, which are the folders that contain the content to search.

**search folder:** A **Folder object** that provides a means of querying for items that match certain criteria. The search folder includes the search folder definition message and the search folder container.

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

**Server object handle:** A 32-bit value that identifies a **Server object**.

**Server object handle table:** An array of 32-bit handles that are used to identify input and output **Server objects** for **ROP requests** and **ROP responses**.

**server replica:** A copy of a user's **mailbox** that exists on a server.

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

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

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

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

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

[MS-OXCXICS] 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-OXCPerm] Microsoft Corporation, "[Exchange Access and Operation Permissions Protocol](#)".

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

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

[MS-OXCSTOR] Microsoft Corporation, "[Store Object Protocol](#)".

[MS-OXCTABL] Microsoft Corporation, "[Table Object Protocol](#)".

[MS-OXOMSG] Microsoft Corporation, "[Email Object Protocol](#)".

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

[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

[MS-OXIMAP4] Microsoft Corporation, "[Internet Message Access Protocol Version 4 \(IMAP4\) Extensions](#)".

[MS-OXPROTO] Microsoft Corporation, "[Exchange Server Protocols System Overview](#)".

[MS-XWDVSEC] Microsoft Corporation, "[Web Distributed Authoring and Versioning \(WebDAV\) Protocol Security Descriptor Extensions](#)".

## 1.3 Overview

A ROP input buffer is used by the client to send requests to the server. A ROP output buffer is used by the server to send responses back to the client. These requests specify operations, such as the following, that access and modify mailbox data on a server:

- Log on to mailboxes or **public folders** on a server
- Retrieve the list of folders and messages on a mailbox
- Move and copy messages or folders from one location to another

- Sort and filter views of messages or folders
- Submit new messages for delivery to other mailboxes or servers
- Retrieve a list of changes made to messages and folders
- Register for notifications when contents of a mailbox have changed

A client packages one or more **ROP requests** into a ROP input buffer and either makes a method call on the **EMS MDB** interface, as described in [\[MS-OXCRPC\]](#) section 3.2, or uses a request type, as described in [\[MS-OXCMAPIHTTP\]](#) section 3.1, passing the ROP input buffer as part of the payload. The server then parses the ROP requests from the ROP input buffer. Each request is processed and a **ROP response** is added to the ROP output buffer. Once the server has finished processing the ROP input buffer, the method call returns the responses to the client, as described in [\[MS-OXCRPC\]](#) section 3.1 or as described in [\[MS-OXCMAPIHTTP\]](#) section 3.2.

### 1.3.1 Server Objects

A **Server object** refers to an object created during the processing of specific ROPs. Once created, a Server object can then be used as a parameter for other ROPs until it is released. A Server object is typically a view of data on the server, such as a **Folder object** or a **Message object**, but it could also be an object used to control synchronization or change notifications.

A Server object is identified by a **Server object handle**, a 32-bit **handle** value. This handle is assigned by the server when the Server object is created. A client uses a Server object handle to specify a Server object to be used as input for a ROP request.

### 1.3.2 Format of ROP Buffers

A ROP input buffer contains a list of ROP requests and a **Server object handle table**. A ROP output buffer contains a list of ROP responses and a Server object handle table. Each ROP request contains input values to use for an operation. Each ROP response contains output produced during the processing of previous requests. The Server object handle table contains Server object handles that are used as input or output for ROPs.

With few exceptions, a ROP request or ROP response does not directly specify a Server object handle but instead contains an index into the Server object handle table. When the index is used for an input parameter, it specifies the location of the Server object handle identifying the input Server object. When the index is used for an output parameter, it specifies the entry in the Server object handle table where the handle for the created Server object will be stored if the operation is successful. If the operation is unsuccessful, the content of the entry in the Server object handle table is undefined.

The usage of an index allows a client to create a ROP input buffer consisting of ROP requests that reference as an input parameter a Server object created earlier during the server's processing of a previous ROP input buffer. In this way, fewer round-trip calls between the client and server are necessary.

## 1.4 Relationship to Other Protocols

This protocol depends on the Wire Format Protocol, as described in [\[MS-OXCRPC\]](#), or on the MAPI extensions to HTTP, as described in [\[MS-OXCMAPIHTTP\]](#).

This protocol provides details for implementing a ROP parsing layer. It is expected that this protocol is to be implemented in its entirety in order to facilitate working with any of the other protocols built on top of this protocol. Complete implementation is necessary to be able to parse ROPs in the ROP input/output buffer, because there are no markers between **ROP buffers** in the **remote procedure call (RPC)** buffer.

Several protocols can be considered alternatives to the ROP List and Encoding Protocol. These alternatives include the Internet Message Access Protocol Version 4 (IMAP4), as described in [\[MS-OXIMAP4\]](#), and Web Distributed Authoring and Versioning (WebDAV), as described in [\[MS-XWDVSEC\]](#).

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 a connection was already made using the **EMSMDB RPC** interface, as specified in [\[MS-OXCRPC\]](#), or the **EMSMDB HTTP** interface, as specified in [\[MS-OXCMAPIHTTP\]](#). The connection process authenticates and exchanges version information between the server and client.

Successful communication between the client and server requires that the client is already configured with information about the mailbox and server. Several ROPs make use of a **distinguished name (DN)** as an input field to specify a mailbox or a server.

## 1.6 Applicability Statement

This protocol is used to view and modify mailbox and public folder data. It provides broad low-level access to content on the server.

## 1.7 Versioning and Capability Negotiation

None.

## 1.8 Vendor-Extensible Fields

None.

## 1.9 Standards Assignments

None.

## 2 Messages

Unless otherwise specified, buffers and fields in this section are depicted in **little-endian** byte order.

### 2.1 Transport

The ROP input/output buffer, as specified in section 2.2.1, is sent as part of the payload of either the **EcDoRpcExt2** RPC method, as specified in [MS-OXCRPC] section 3.1.4.2, or the **Execute** request type, <1> as specified in [MS-OXCMAPIHTTP] section 2.2.4.2.

### 2.2 Message Syntax

#### 2.2.1 ROP Input and Output Buffers

The ROP input buffer, which is sent by the client, includes an array of **ROP request buffers** to be processed by the server. The ROP output buffer, which is sent by the server, includes an array of **ROP response buffers**. The ROP input and output buffers are transmitted between the client and the server as part of the payload of either the **EcDoRpcExt2** method, as specified in [MS-OXCRPC] section 3.1.4.2, or the **Execute** request type, <2> as specified in [MS-OXCMAPIHTTP] section 2.2.4.2.

The ROP input buffer and ROP output buffer have the following structure.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopSize																RopsList (variable)															
...																															
ServerObjectHandleTable (variable)																															
...																															

**RopSize (2 bytes):** An unsigned integer that specifies the size of both this field and the **RopsList** field.

**RopsList (variable):** An array of ROP buffers. For a ROP input buffer, this field contains an array of ROP request buffers. For a ROP output buffer, this field contains an array of ROP response buffers. The format of each ROP buffer is specified in subsequent sections. The size of this field is 2 bytes less than the value specified in the **RopSize** field.

**ServerObjectHandleTable (variable):** An array of 32-bit values. Each 32-bit value specifies a Server object handle that is referenced by a ROP buffer. The size of this field is equal to the number of bytes of data remaining in the ROP input/output buffer after the **RopsList** field.

#### 2.2.2 The Table of RopIds

Each ROP is identified by a one-byte value, which is contained in the **RopId** field of ROP request buffers and ROP response buffers. The ROPs that a client is allowed to use are listed in the following table. A **RopId** value that is specified as "Reserved" is not used in the communication between the client and server. Therefore, the client **MUST NOT** use the reserved **RopId** values.

When the server encounters a **RopId** value that is associated with a reserved ROP, it SHOULD<3> return an error for the RPC, as specified in [\[MS-OXCRPC\]](#) section 3.1.4.2, or for the **Execute request type**<4> response, as specified in [\[MS-OXCMAPIHTTP\]](#) section 2.2.4.2.2.

<b>RopId value</b>	<b>ROP name</b>	<b>Section</b>
0x00	Reserved	
0x01	<b>RopRelease</b>	<a href="#">2.2.15.3</a>
0x02	<b>RopOpenFolder</b>	<a href="#">2.2.4.1</a>
0x03	<b>RopOpenMessage</b>	<a href="#">2.2.6.1</a>
0x04	<b>RopGetHierarchyTable</b>	<a href="#">2.2.4.13</a>
0x05	<b>RopGetContentsTable</b>	<a href="#">2.2.4.14</a>
0x06	<b>RopCreateMessage</b>	<a href="#">2.2.6.2</a>
0x07	<b>RopGetPropertiesSpecific</b>	<a href="#">2.2.8.3</a>
0x08	<b>RopGetPropertiesAll</b>	<a href="#">2.2.8.4</a>
0x09	<b>RopGetPropertiesList</b>	<a href="#">2.2.8.5</a>
0x0A	<b>RopSetProperties</b>	<a href="#">2.2.8.6</a>
0x0B	<b>RopDeleteProperties</b>	<a href="#">2.2.8.8</a>
0x0C	<b>RopSaveChangesMessage</b>	<a href="#">2.2.6.3</a>
0x0D	<b>RopRemoveAllRecipients</b>	<a href="#">2.2.6.4</a>
0x0E	<b>RopModifyRecipients</b>	<a href="#">2.2.6.5</a>
0x0F	<b>RopReadRecipients</b>	<a href="#">2.2.6.6</a>
0x10	<b>RopReloadCachedInformation</b>	<a href="#">2.2.6.7</a>
0x11	<b>RopSetMessageReadFlag</b>	<a href="#">2.2.6.11</a>
0x12	<b>RopSetColumns</b>	<a href="#">2.2.5.1</a>
0x13	<b>RopSortTable</b>	<a href="#">2.2.5.2</a>
0x14	<b>RopRestrict</b>	<a href="#">2.2.5.3</a>
0x15	<b>RopQueryRows</b>	<a href="#">2.2.5.4</a>
0x16	<b>RopGetStatus</b>	<a href="#">2.2.5.6</a>
0x17	<b>RopQueryPosition</b>	<a href="#">2.2.5.7</a>
0x18	<b>RopSeekRow</b>	<a href="#">2.2.5.8</a>
0x19	<b>RopSeekRowBookmark</b>	<a href="#">2.2.5.9</a>
0x1A	<b>RopSeekRowFractional</b>	<a href="#">2.2.5.10</a>
0x1B	<b>RopCreateBookmark</b>	<a href="#">2.2.5.11</a>
0x1C	<b>RopCreateFolder</b>	<a href="#">2.2.4.2</a>

<b>RopId value</b>	<b>ROP name</b>	<b>Section</b>
0x1D	<b>RopDeleteFolder</b>	<a href="#">2.2.4.3</a>
0x1E	<b>RopDeleteMessages</b>	<a href="#">2.2.4.11</a>
0x1F	<b>RopGetMessageStatus</b>	<a href="#">2.2.6.9</a>
0x20	<b>RopSetMessageStatus</b>	<a href="#">2.2.6.8</a>
0x21	<b>RopGetAttachmentTable</b>	<a href="#">2.2.6.17</a>
0x22	<b>RopOpenAttachment</b>	<a href="#">2.2.6.12</a>
0x23	<b>RopCreateAttachment</b>	<a href="#">2.2.6.13</a>
0x24	<b>RopDeleteAttachment</b>	<a href="#">2.2.6.14</a>
0x25	<b>RopSaveChangesAttachment</b>	<a href="#">2.2.6.15</a>
0x26	<b>RopSetReceiveFolder</b>	<a href="#">2.2.3.3</a>
0x27	<b>RopGetReceiveFolder</b>	<a href="#">2.2.3.2</a>
0x28	Reserved	
0x29	<b>RopRegisterNotification</b>	<a href="#">2.2.14.1</a>
0x2A	<b>RopNotify</b>	<a href="#">2.2.14.2</a>
0x2B	<b>RopOpenStream</b>	<a href="#">2.2.9.1</a>
0x2C	<b>RopReadStream</b>	<a href="#">2.2.9.2</a>
0x2D	<b>RopWriteStream</b>	<a href="#">2.2.9.3</a>
0x2E	<b>RopSeekStream</b>	<a href="#">2.2.9.7</a>
0x2F	<b>RopSetStreamSize</b>	<a href="#">2.2.9.6</a>
0x30	<b>RopSetSearchCriteria</b>	<a href="#">2.2.4.4</a>
0x31	<b>RopGetSearchCriteria</b>	<a href="#">2.2.4.5</a>
0x32	<b>RopSubmitMessage</b>	<a href="#">2.2.7.1</a>
0x33	<b>RopMoveCopyMessages</b>	<a href="#">2.2.4.6</a>
0x34	<b>RopAbortSubmit</b>	<a href="#">2.2.7.2</a>
0x35	<b>RopMoveFolder</b>	<a href="#">2.2.4.7</a>
0x36	<b>RopCopyFolder</b>	<a href="#">2.2.4.8</a>
0x37	<b>RopQueryColumnsAll</b>	<a href="#">2.2.5.12</a>
0x38	<b>RopAbort</b>	<a href="#">2.2.5.5</a>
0x39	<b>RopCopyTo</b>	<a href="#">2.2.8.12</a>
0x3A	<b>RopCopyToStream</b>	<a href="#">2.2.9.8</a>
0x3B	<b>RopCloneStream</b>	<a href="#">2.2.9.12</a>



<b>RopId value</b>	<b>ROP name</b>	<b>Section</b>
0x3C	Reserved	
0x3D	Reserved	
0x3E	<b>RopGetPermissionsTable</b>	<a href="#">2.2.10.2</a>
0x3F	<b>RopGetRulesTable</b>	<a href="#">2.2.11.2</a>
0x40	<b>RopModifyPermissions</b>	<a href="#">2.2.10.1</a>
0x41	<b>RopModifyRules</b>	<a href="#">2.2.11.1</a>
0x42	<b>RopGetOwningServers</b>	<a href="#">2.2.3.6</a>
0x43	<b>RopLongTermIdFromId</b>	<a href="#">2.2.3.8</a>
0x44	<b>RopIdFromLongTermId</b>	<a href="#">2.2.3.9</a>
0x45	<b>RopPublicFolderIsGhosted</b>	<a href="#">2.2.3.7</a>
0x46	<b>RopOpenEmbeddedMessage</b>	<a href="#">2.2.6.16</a>
0x47	<b>RopSetSpooler</b>	<a href="#">2.2.7.4</a>
0x48	<b>RopSpoolerLockMessage</b>	<a href="#">2.2.7.5</a>
0x49	<b>RopGetAddressTypes</b>	<a href="#">2.2.7.3</a>
0x4A	<b>RopTransportSend</b>	<a href="#">2.2.7.6</a>
0x4B	<b>RopFastTransferSourceCopyMessages</b>	<a href="#">2.2.12.5</a>
0x4C	<b>RopFastTransferSourceCopyFolder</b>	<a href="#">2.2.12.4</a>
0x4D	<b>RopFastTransferSourceCopyTo</b>	<a href="#">2.2.12.6</a>
0x4E	<b>RopFastTransferSourceGetBuffer</b>	<a href="#">2.2.12.3</a>
0x4F	<b>RopFindRow</b>	<a href="#">2.2.5.13</a>
0x50	<b>RopProgress</b>	<a href="#">2.2.8.13</a>
0x51	<b>RopTransportNewMail</b>	<a href="#">2.2.7.7</a>
0x52	<b>RopGetValidAttachments</b>	<a href="#">2.2.6.18</a>
0x53	<b>RopFastTransferDestinationConfigure</b>	<a href="#">2.2.12.1</a>
0x54	<b>RopFastTransferDestinationPutBuffer</b>	<a href="#">2.2.12.2</a>
0x55	<b>RopGetNamesFromPropertyIds</b>	<a href="#">2.2.8.2</a>
0x56	<b>RopGetPropertyIdsFromNames</b>	<a href="#">2.2.8.1</a>
0x57	<b>RopUpdateDeferredActionMessages</b>	<a href="#">2.2.11.3</a>
0x58	<b>RopEmptyFolder</b>	<a href="#">2.2.4.9</a>
0x59	<b>RopExpandRow</b>	<a href="#">2.2.5.16</a>
0x5A	<b>RopCollapseRow</b>	<a href="#">2.2.5.17</a>

<b>RopId value</b>	<b>ROP name</b>	<b>Section</b>
0x5B	<b>RopLockRegionStream</b>	<a href="#">2.2.9.9</a>
0x5C	<b>RopUnlockRegionStream</b>	<a href="#">2.2.9.10</a>
0x5D	<b>RopCommitStream</b>	<a href="#">2.2.9.4</a>
0x5E	<b>RopGetStreamSize</b>	<a href="#">2.2.9.5</a>
0x5F	<b>RopQueryNamedProperties</b>	<a href="#">2.2.8.10</a>
0x60	<b>RopGetPerUserLongTermIds</b>	<a href="#">2.2.3.10</a>
0x61	<b>RopGetPerUserGuid</b>	<a href="#">2.2.3.11</a>
0x62	Reserved	
0x63	<b>RopReadPerUserInformation</b>	<a href="#">2.2.3.12</a>
0x64	<b>RopWritePerUserInformation</b>	<a href="#">2.2.3.13</a>
0x65	Reserved	
0x66	<b>RopSetReadFlags</b>	<a href="#">2.2.6.10</a>
0x67	<b>RopCopyProperties</b>	<a href="#">2.2.8.11</a>
0x68	<b>RopGetReceiveFolderTable</b>	<a href="#">2.2.3.4</a>
0x69	<b>RopFastTransferSourceCopyProperties</b>	<a href="#">2.2.12.7</a>
0x6A	Reserved	
0x6B	<b>RopGetCollapseState</b>	<a href="#">2.2.5.18</a>
0x6C	<b>RopSetCollapseState</b>	<a href="#">2.2.5.19</a>
0x6D	<b>RopGetTransportFolder</b>	<a href="#">2.2.7.8</a>
0x6E	<b>RopPending</b>	<a href="#">2.2.14.3</a>
0x6F	<b>RopOptionsData</b>	<a href="#">2.2.7.9</a>
0x70	<b>RopSynchronizationConfigure</b>	<a href="#">2.2.13.1</a>
0x71	Reserved	
0x72	<b>RopSynchronizationImportMessageChange</b>	<a href="#">2.2.13.2</a>
0x73	<b>RopSynchronizationImportHierarchyChange</b>	<a href="#">2.2.13.4</a>
0x74	<b>RopSynchronizationImportDeletes</b>	<a href="#">2.2.13.5</a>
0x75	<b>RopSynchronizationUploadStateStreamBegin</b>	<a href="#">2.2.13.9</a>
0x76	<b>RopSynchronizationUploadStateStreamContinue</b>	<a href="#">2.2.13.10</a>
0x77	<b>RopSynchronizationUploadStateStreamEnd</b>	<a href="#">2.2.13.11</a>
0x78	<b>RopSynchronizationImportMessageMove</b>	<a href="#">2.2.13.6</a>
0x79	<b>RopSetPropertiesNoReplicate</b>	<a href="#">2.2.8.7</a>

<b>RopId value</b>	<b>ROP name</b>	<b>Section</b>
0x7A	<b>RopDeletePropertiesNoReplicate</b>	<a href="#">2.2.8.9</a>
0x7B	<b>RopGetStoreState</b>	<a href="#">2.2.3.5</a>
0x7C	Reserved	
0x7D	Reserved	
0x7E	<b>RopSynchronizationOpenCollector</b>	<a href="#">2.2.13.7</a>
0x7F	<b>RopGetLocalReplicaIds</b>	<a href="#">2.2.13.13</a>
0x80	<b>RopSynchronizationImportReadStateChanges</b>	<a href="#">2.2.13.3</a>
0x81	<b>RopResetTable</b>	<a href="#">2.2.5.15</a>
0x82	<b>RopSynchronizationGetTransferState</b>	<a href="#">2.2.13.8</a>
0x83	<b>Reserved</b>	
0x84	<b>Reserved</b>	
0x85	Reserved	
0x86	<b>RopTellVersion</b>	<a href="#">2.2.12.8</a>
0x87	Reserved	
0x88	<b>Reserved</b>	
0x89	<b>RopFreeBookmark</b>	<a href="#">2.2.5.14</a>
0x8A	Reserved	
0x8B	Reserved	
0x8C	Reserved	
0x8D	Reserved	
0x8E	Reserved	
0x8F	Reserved	
0x90	<b>RopWriteAndCommitStream</b>	<a href="#">2.2.9.11</a>
0x91	<b>RopHardDeleteMessages</b>	<a href="#">2.2.4.12</a>
0x92	<b>RopHardDeleteMessagesAndSubfolders</b>	<a href="#">2.2.4.10</a>
0x93	<b>RopSetLocalReplicaMidsetDeleted</b>	<a href="#">2.2.13.12</a>
0x94	Reserved	
0x95	Reserved	
0x96	Reserved	
0x97	Reserved	
0x98	Reserved	

<b>RopId value</b>	<b>ROP name</b>	<b>Section</b>
0x99	Reserved	
0x9A	Reserved	
0x9B	Reserved	
0x9C	Reserved	
0x9D	Reserved	
0x9E	Reserved	
0x9F	Reserved	
0xA0	Reserved	
0xA1	Reserved	
0xA2	Reserved	
0xA3	Reserved	
0xA4	Reserved	
0xA5	Reserved	
0xA6	Reserved	
0xA7	Reserved	
0xA8	Reserved	
0xA9	Reserved	
0xAA	Reserved	
0xAB	Reserved	
0xAC	Reserved	
0xAD	Reserved	
0xAE	Reserved	
0xAF	Reserved	
0xB0	Reserved	
0xB1	Reserved	
0xB2	Reserved	
0xB3	Reserved	
0xB4	Reserved	
0xB5	Reserved	
0xB6	Reserved	
0xB7	Reserved	

<b>RopId value</b>	<b>ROP name</b>	<b>Section</b>
0xB8	Reserved	
0xB9	Reserved	
0xBA	Reserved	
0xBB	Reserved	
0xBC	Reserved	
0xBD	Reserved	
0xBE	Reserved	
0xBF	Reserved	
0xC0	Reserved	
0xC1	Reserved	
0xC2	Reserved	
0xC3	Reserved	
0xC4	Reserved	
0xC5	Reserved	
0xC6	Reserved	
0xC7	Reserved	
0xC8	Reserved	
0xC9	Reserved	
0xCA	Reserved	
0xCB	Reserved	
0xCC	Reserved	
0xCD	Reserved	
0xCE	Reserved	
0xCF	Reserved	
0xD0	Reserved	
0xD1	Reserved	
0xD2	Reserved	
0xD3	Reserved	
0xD4	Reserved	
0xD5	Reserved	
0xD6	Reserved	

<b>RopId value</b>	<b>ROP name</b>	<b>Section</b>
0xD7	Reserved	
0xD8	Reserved	
0xD9	Reserved	
0xDA	Reserved	
0xDB	Reserved	
0xDC	Reserved	
0xDD	Reserved	
0xDE	Reserved	
0xDF	Reserved	
0xE0	Reserved	
0xE1	Reserved	
0xE2	Reserved	
0xE3	Reserved	
0xE4	Reserved	
0xE5	Reserved	
0xE6	Reserved	
0xE7	Reserved	
0xE8	Reserved	
0xE9	Reserved	
0xEA	Reserved	
0xEB	Reserved	
0xEC	Reserved	
0xED	Reserved	
0xEE	Reserved	
0xEF	Reserved	
0xF0	Reserved	
0xF1	Reserved	
0xF2	Reserved	
0xF3	Reserved	
0xF4	Reserved	
0xF5	Reserved	

RopId value	ROP name	Section
0xF6	Reserved	
0xF7	Reserved	
0xF8	Reserved	
0xF9	<b>RopBackoff</b>	<a href="#">2.2.15.2</a>
0xFA	Reserved	
0xFB	Reserved	
0xFC	Reserved	
0xFD	Reserved	
0xFE	<b>RopLogon</b>	<a href="#">2.2.3.1</a>
0xFF	<b>RopBufferTooSmall</b>	<a href="#">2.2.15.1</a>

## 2.2.3 Logon ROPs

### 2.2.3.1 RopLogon ROP

The **RopLogon** ROP logs on to a mailbox or a public folder. For more details about this operation, see [\[MS-OXCSTOR\]](#) section 2.2.1.1.

#### 2.2.3.1.1 RopLogon ROP Request Buffer

The following descriptions define valid fields for the **RopLogon** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										LogonId										OutputHandleIndex						LogonFlags					
OpenFlags																															
StoreState																															
EssdnSize																Essdn (variable)															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0xFE.

**LogonId (1 byte):** An unsigned integer that specifies the ID that the client requests to have associated with the created logon. Any value is allowed, and the client does not have to use values in a certain numeric order. If the client specifies an active LogonID, the current logon is released and replaced with the new one.

**OutputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the output Server object will be stored. For more information about Server objects, see section [1.3.1](#).

**LogonFlags (1 byte):** A **flags** structure that contains flags that control the behavior of the logon. The possible values are specified in [\[MS-OXCSTOR\]](#) section 2.2.1.1.1.

**OpenFlags (4 bytes):** A flags structure that contains more flags that control the behavior of the logon. The possible values are specified in [\[MS-OXCSTOR\]](#) section 2.2.1.1.1.

**StoreState (4 bytes):** A flags structure. This field is not used and is ignored by the server. This field MUST be set to 0x00000000.

**EssdnSize (2 bytes):** An unsigned integer that specifies the size of the **Essdn** field.

**Essdn (variable):** A null-terminated **ASCII** string that specifies which mailbox to log on to. The number of characters (including the terminating null character) contained in this field is specified by the **EssdnSize** field.

### 2.2.3.1.2 RopLogon ROP Success Response Buffer for Private Mailboxes

The following descriptions define valid fields for the **RopLogon** ROP success response buffer for private mailboxes.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										OutputHandleIndex										ReturnValue											
...										LogonFlags										FolderIds											
...																															
...																															
...																															
...																															
...																															
...																															
...																															
(FolderIds cont'd for 18 rows)																															
...																				ResponseFlags											
MailboxGuid																															



...	
...	
...	
ReplId	ReplGuid
...	
...	
...	
...	LogonTime
...	
...	GwartTime
...	
...	StoreState
...	

**ROPId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0xFE.

**OutputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the output Server object will be stored. For more information about Server objects, see section [1.3.1](#).

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**LogonFlags (1 byte):** A flags structure that contains flags that specify the type of logon. The possible values are specified in [\[MS-OXCSTOR\]](#) section 2.2.1.1.3.

**FolderIds (104 bytes):** 13 64-bit identifiers that specify a set of **special folders** for a mailbox.

**ResponseFlags (1 byte):** A flags structure that contains flags that provide details about the state of the mailbox. The possible values are specified in [\[MS-OXCSTOR\]](#) section 2.2.1.1.3.

**MailboxGuid (16 bytes):** A **GUID** that identifies the mailbox on which the logon was performed.

**ReplId (2 bytes):** An identifier that specifies a **replica ID** for the logon.

**ReplGuid (16 bytes):** A GUID that specifies the **replica GUID** that is associated with the replica ID, which is specified in the **ReplId** field.

**LogonTime (8 bytes):** A **LogonTime** structure that specifies the time at which the logon occurred. The format of this structure is specified in section [2.2.3.1.2.1](#).

**GwartTime (8 bytes):** An unsigned integer that contains a numeric value that tracks the currency of the **Gateway Address Routing Table (GWART)**. The server generates a new numeric value with each change of the GWART. The client can use the value of this field in a comparison to detect a change but does not interpret the value of this field.

**StoreState (4 bytes):** A flags structure. This field **MUST** be set to 0x00000000 by the server and **MUST** be ignored by the client.

### 2.2.3.1.2.1 LogonTime Structure

The following descriptions define valid fields for the **LogonTime** structure.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Seconds								Minutes								Hour								DayOfWeek							
Day								Month								Year															

**Seconds (1 byte):** An unsigned integer that specifies the current second.

**Minutes (1 byte):** An unsigned integer that specifies the current minute.

**Hour (1 byte):** An unsigned integer that specifies the current hour.

**DayOfWeek (1 byte):** An enumeration that specifies the current day of the week (Sunday = 0, Monday = 1, and so on).

**Day (1 byte):** An unsigned integer that specifies the current day of the month.

**Month (1 byte):** An unsigned integer that specifies the current month (January = 1, February = 2, and so on).

**Year (2 bytes):** An unsigned integer that specifies the current year.

### 2.2.3.1.3 RopLogon ROP Success Response Buffer for Public Folders

The following descriptions define valid fields for the **RopLogon** ROP success response buffer for public folders.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId								OutputHandleIndex								ReturnValue															
...								...								LogonFlags								FolderIds							
...																...															
...																...															

...	
...	
...	
...	
...	
...	
(FolderIds cont'd for 18 rows)	
...	ReplId
...	ReplGuid
...	
...	
...	
...	PerUserGuid
...	
...	
...	
...	
...	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0xFE.

**OutputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the output Server object will be stored. For more information about Server objects, see section [1.3.1](#).

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**LogonFlags (1 byte):** A flags structure that contains flags that specify the type of logon. The possible values are specified in [\[MS-OXCSTOR\]](#) section 2.2.1.1.4.

**FolderIds (104 bytes):** 13 64-bit identifiers that specify a set of special folders for a mailbox.

**ReplId (2 bytes):** An identifier that specifies a replica ID for the logon.

**ReplGuid (16 bytes):** A GUID that specifies the replica GUID associated with the replica ID that is specified in the **ReplId** field. <5>

**PerUserGuid (16 bytes):** This field is not used and is ignored by the client. The server SHOULD set this field to all zeroes.

### 2.2.3.1.4 RopLogon ROP Redirect Response Buffer

The following descriptions define valid fields for the **RopLogon** ROP redirect response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										OutputHandleIndex										ReturnValue											
...															LogonFlags										ServerNameSize						
ServerName (variable)																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0xFE.

**OutputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the output Server object will be stored. For more information about Server objects, see section [1.3.1](#).

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000478.

**LogonFlags (1 byte):** A flags structure that contains flags that specify the type of logon. The possible values are specified in [\[MS-OXCSTOR\]](#) section 2.2.1.1.2.

**ServerNameSize (1 byte):** An unsigned integer that specifies the length of the **ServerName** field.

**ServerName (variable):** A null-terminated ASCII string that specifies a different server for the client to connect to. The number of characters (including the terminating null character) contained in this field is specified by the **ServerNameSize** field.

### 2.2.3.1.5 RopLogon ROP Failure Response Buffer

The following descriptions define valid fields for the **RopLogon** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										OutputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0xFE.

**OutputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the output Server object will be stored. For more information about Server objects, see section [1.3.1](#).

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000 or 0x00000478.

### 2.2.3.2 RopGetReceiveFolder ROP

The **RopGetReceiveFolder** ROP gets the **Receive folder** for incoming messages of a particular **message class**. For more details about this operation, see [\[MS-OXCSTOR\]](#) section 2.2.1.2.

#### 2.2.3.2.1 RopGetReceiveFolder ROP Request Buffer

The following descriptions define valid fields for the **RopGetReceiveFolder** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31								
RopId										LogonId										InputHandleIndex										MessageClass (variable)									
...																																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x27.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**MessageClass (variable):** A null-terminated ASCII string that specifies the message class to find the Receive folder for.

#### 2.2.3.2.2 RopGetReceiveFolder ROP Success Response Buffer

The following descriptions define valid fields for the **RopGetReceiveFolder** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...																FolderId															
...																															
...																ExplicitMessageClass (variable)															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x27.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**FolderId (8 bytes):** An identifier that specifies the Receive folder.

**ExplicitMessageClass (variable):** A null-terminated ASCII string that specifies the message class that is actually configured for delivery to the folder.

### 2.2.3.2.3 RopGetReceiveFolder ROP Failure Response Buffer

The following descriptions define valid fields for the **RopGetReceiveFolder** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x27.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.3.3 RopSetReceiveFolder ROP

The **RopSetReceiveFolder** ROP sets the Receive folder for incoming messages of a particular message class. For more details about this operation, see [\[MS-OXCSTOR\]](#) section 2.2.1.3.

#### 2.2.3.3.1 RopSetReceiveFolder ROP Request Buffer

The following descriptions define valid fields for the **RopSetReceiveFolder** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31								
RopId										LogonId										InputHandleIndex										FolderId									
...																																							
...																						MessageClass (variable)																	
...																																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x26.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**FolderId (8 bytes):** An identifier that specifies the Receive folder.

**MessageClass (variable):** A null-terminated ASCII string that specifies which message class to set the Receive folder for.

### 2.2.3.3.2 RopSetReceiveFolder ROP Response Buffer

The following descriptions define valid fields for the **RopSetReceiveFolder** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x26.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.3.4 RopGetReceiveFolderTable ROP

The **RopGetReceiveFolderTable** ROP gets the table of all folders that were established as Receive folders. For more details about this operation, see [\[MS-OXCSTOR\]](#) section 2.2.1.4.

#### 2.2.3.4.1 RopGetReceiveFolderTable ROP Request Buffer

The following descriptions define valid fields for the **RopGetReceiveFolderTable** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								LogonId								InputHandleIndex															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x68.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

### 2.2.3.4.2 RopGetReceiveFolderTable ROP Success Response Buffer

The following descriptions define valid fields for the **RopGetReceiveFolderTable** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...										RowCount																					
...										Rows (variable)																					
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x68.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**RowCount (4 bytes):** An unsigned integer that specifies the number of row structures contained in the **Rows** field.

**Rows (variable):** An array of row structures. This field contains the rows of the Receive folder table. Each row is returned in either a **StandardPropertyRow** structure, as specified in [\[MS-OXCDATA\]](#) section 2.8.1.1, or a **FlaggedPropertyRow** structure, as specified in [\[MS-OXCDATA\]](#) section 2.8.1.2. The number of row structures contained in this field is specified by the **RowCount** field. The **ValueArray** field of either the **StandardPropertyRow** structure or the **FlaggedPropertyRow** structure MUST include only the **PidTagFolderId** ([\[MS-OXCFCID\]](#) section 2.2.2.1.5), **PidTagMessageClass** ([\[MS-OXCMSG\]](#) section 2.2.1.3), and **PidTagLastModificationTime** ([\[MS-OXCMSG\]](#) section 2.2.2.2) properties, in that order, and no other properties.

### 2.2.3.4.3 RopGetReceiveFolderTable ROP Failure Response Buffer

The following descriptions define valid fields for the **RopGetReceiveFolderTable** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x68.



**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.3.5 RopGetStoreState ROP

The **RopGetStoreState** ROP gets the current state of the mailbox for the logged on user. For more details about this operation, see [\[MS-OXCSTOR\]](#) section 2.2.1.5.

#### 2.2.3.5.1 RopGetStoreStage ROP Request Buffer

The following descriptions define valid fields for the **RopGetStoreStage** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										LogonId										InputHandleIndex											

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x7B.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

#### 2.2.3.5.2 RopGetStoreState ROP Success Response Buffer

The following descriptions define valid fields for the **RopGetStoreState** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...										StoreState																					
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x7B.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**StoreState (4 bytes):** A flags structure that contains flags that indicate the state of the mailbox for the logged on user. The possible values are specified in [\[MS-OXCSTOR\]](#) section 2.2.1.5.2.

#### 2.2.3.5.3 RopGetStoreState ROP Failure Response Buffer

The following descriptions define valid fields for the **RopGetStoreState** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x7B.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.3.6 RopGetOwningServers ROP

The **RopGetOwningServers** ROP gets the list of servers that host **replicas (1)** of a given public folder. For more details about this operation, see [\[MS-OXCSTOR\]](#) section 2.2.1.6.

#### 2.2.3.6.1 RopGetOwningServers ROP Request Buffer

The following descriptions define valid fields for the **RopGetOwningServers** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1				
RopId										LogonId										InputHandleIndex								FolderId							
...																																			
...																																			

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x42.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**FolderId (8 bytes):** An identifier that specifies the folder for which to get owning servers.

#### 2.2.3.6.2 RopGetOwningServers ROP Success Response Buffer

The following descriptions define valid fields for the **RopGetOwningServers** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										InputHandleIndex										ReturnValue											
...										OwningServersCount																					
CheapServersCount										OwningServers (variable)																					
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x42.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**OwningServersCount (2 bytes):** An unsigned integer that specifies the number of strings in the **OwningServers** field.

**CheapServersCount (2 bytes):** An unsigned integer that specifies the number of strings in the **OwningServers** field that refer to **lowest-cost servers**.

**OwningServers (variable):** A list of null-terminated ASCII strings that specify which servers have replicas (1) of this folder. The number of strings contained in this field is specified by the **OwningServersCount** field.

### 2.2.3.6.3 RopGetOwningServers ROP Failure Response Buffer

The following descriptions define valid fields for the **RopGetOwningServers** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x42.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.3.7 RopPublicFolderIsGhosed ROP

The **RopPublicFolderIsGhosed** ROP determines whether a public folder is ghosed. For more details about this operation, see [\[MS-OXCSTOR\]](#) section 2.2.1.7.

#### 2.2.3.7.1 RopPublicFolderIsGhosed ROP Request Buffer

The following descriptions define valid fields for the **RopPublicFolderIsGhosed** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31								
RopId										LogonId										InputHandleIndex										FolderId									
...																																							
...																																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x45.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**FolderId (8 bytes):** An identifier that specifies the folder to check.

#### 2.2.3.7.2 RopPublicFolderIsGhosed ROP Success Response Buffer

The following descriptions define valid fields for the **RopPublicFolderIsGhosed** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...										IsGhosed										ServersCount (optional)											
...										CheapServersCount (optional)										Servers (optional) (variable)											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x45.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**IsGhosed (1 byte):** A Boolean that specifies whether the folder is a **ghosted folder**.

**ServersCount (2 bytes):** An unsigned integer that is present if **IsGhosed** is nonzero and is not present if **IsGhosed** is zero. This value specifies the number of strings in the **Servers** field.

**CheapServersCount (2 bytes):** An unsigned integer that is present if the value of the **IsGhosed** field is nonzero and is not present if the value of the **IsGhosed** field is zero. This value specifies the number of strings in the **Servers** field that refer to a lowest-cost server.

**Servers (optional) (variable):** A list of null-terminated ASCII strings that specify which servers have replicas (1) of this folder. This field is present if **IsGhosed** is nonzero and is not present if **IsGhosed** is zero.

### 2.2.3.7.3 RopPublicFolderIsGhosed ROP Failure Response Buffer

The following descriptions define valid fields for the **RopPublicFolderIsGhosed** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x45.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value of the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.3.8 RopLongTermIdFromId ROP

The **RopLongTermIdFromId** ROP converts a short-term ID (Folder ID structure or Message ID structure) into a long-term ID (LongTermID structure). For more details about this operation, see [\[MS-OXCSTOR\]](#) section 2.2.1.8. For more details about short-term IDs, see [\[MS-OXCADATA\]](#) section 2.2.1.1 and [\[MS-OXCADATA\]](#) section 2.2.1.2. For more details about long-term IDs, see [\[MS-OXCADATA\]](#) section 2.2.1.3.1.

#### 2.2.3.8.1 RopLongTermIdFromId ROP Request Buffer

The following descriptions define valid fields for the **RopLongTermIdFromId** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31								
RopId										LogonId										InputHandleIndex										ObjectId									
...																																							
...																																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x43.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**ObjectId (8 bytes):** An identifier that specifies the short-term ID to be converted to a long-term ID.

### 2.2.3.8.2 RopLongTermIdFromId ROP Success Response Buffer

The following descriptions define valid fields for the **RopLongTermIdFromId** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...										LongTermId							...														
...										...							...														
...										...							...														
...										...							...														
...										...							...														
...										...							...														

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x43.

**InputHandleIndex (1 byte):** An unsigned integer index that **MUST** be set to the value of the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**LongTermId (24 bytes):** A **LongTermID** structure that specifies the long-term ID that was converted from the short-term ID, which is specified in the **ObjectId** field of the request. The format of the **LongTermID** structure is specified in [\[MS-OXCDATA\]](#) section 2.2.1.3.1.

### 2.2.3.8.3 RopLongTermIdFromId ROP Failure Response Buffer

The following descriptions define valid fields for the **RopLongTermIdFromId** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
RopId										InputHandleIndex										ReturnValue													
...																																	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x43.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.3.9 RopIdFromLongTermId ROP

The **RopIdFromLongTermId** ROP converts a long-term ID into a short-term ID. For more details about this operation, see [\[MS-OXCSTOR\]](#) section 2.2.1.9.

#### 2.2.3.9.1 RopIdFromLongTermId ROP Request Buffer

The following descriptions define valid fields for the **RopIdFromLongTermId** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
RopId										LogonId										InputHandleIndex										LongTermId									
...																																							
...																																							
...																																							
...																																							
...																																							
...																																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x44.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**LongTermId (24 bytes):** A **LongTermID** structure that specifies the long-term ID to be converted to a short-term ID. The format of the **LongTermID** structure is specified in [\[MS-OXCDATA\]](#) section 2.2.1.3.1.

### 2.2.3.9.2 RopIdFromLongTermId ROP Success Response Buffer

The following descriptions define valid fields for the **RopIdFromLongTermId** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										InputHandleIndex										ReturnValue											
...										...										ObjectId											
...										...										...											
...										...										...											

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x44.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**ObjectId (8 bytes):** An identifier that specifies the short-term ID that was converted from the long-term ID, which is specified in the **LongTermId** field of the request.

### 2.2.3.9.3 RopIdFromLongTermId ROP Failure Response Buffer

The following descriptions define valid fields for the **RopIdFromLongTermId** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										InputHandleIndex										ReturnValue											
...										...										...											

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x44.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.



### 2.2.3.10 RopGetPerUserLongTermIds ROP

The **RopGetPerUserLongTermIds** ROP gets the long-term ID of a public folder that is identified by the per-user GUID of the logged on user. For more details about this operation, see [\[MS-OXCSTOR\]](#) section 2.2.1.10.

#### 2.2.3.10.1 RopGetPerUserLongTermIds ROP Request Buffer

The following descriptions define valid fields for the **RopGetPerUserLongTermIds** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										LogonId						InputHandleIndex						DatabaseGuid									
...																															
...																															
...																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x60.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**DatabaseGuid (16 bytes):** A GUID that specifies which database the client is querying data for.

#### 2.2.3.10.2 RopGetPerUserLongTermIds ROP Success Response Buffer

The following descriptions define valid fields for the **RopGetPerUserLongTermIds** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex						ReturnValue															
...																LongTermIdCount															
LongTermIds (variable)																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x60.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**LongTermIdCount (2 bytes):** An unsigned integer that specifies the number of structures in the **LongTermIds** field.

**LongTermIds (variable):** An array of **LongTermID** structures that specifies which folders the user has per-user information about. The format of these structures is specified in [\[MS-OXCDATA\]](#) section 2.2.1.3.1.

### 2.2.3.10.3 RopGetPerUserLongTermIds ROP Failure Response Buffer

The following descriptions define valid fields for the **RopGetPerUserLongTermIds** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
RopId										InputHandleIndex										ReturnValue													
...																																	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x60.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.3.11 RopGetPerUserGuid ROP

The **RopGetPerUserGuid** ROP gets the GUID of a public folder's per-user information. For more details about this operation, see [\[MS-OXCSTOR\]](#) section 2.2.1.11.

#### 2.2.3.11.1 RopGetPerUserGuid ROP Request Buffer

The following descriptions define valid fields for the **RopGetPerUserGuid** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
RopId										LogonId										InputHandleIndex								LongTermId					
...																																	
...																																	
...																																	

...
...
...

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x61.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**LongTermId (24 bytes):** A **LongTermID** structure that specifies the public folder. The format of the **LongTermID** structure is specified in [\[MS-OXCDATA\]](#) section 2.2.1.3.1.

### 2.2.3.11.2 RopGetPerUserGuid ROP Success Response Buffer

The following descriptions define valid fields for the **RopGetPerUserGuid** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...										DatabaseGuid																					
...																															
...																															
...																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x61.

**InputHandleIndex (1 byte):** An unsigned integer index that **MUST** be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**DatabaseGuid (16 bytes):** A GUID that specifies the database for which per-user information was obtained.

### 2.2.3.11.3 RopGetPerUserGuid ROP Failure Response Buffer

The following descriptions define valid fields for the **RopGetPerUserGuid** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
RopId										InputHandleIndex										ReturnValue													
...																																	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x61.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.3.12 RopReadPerUserInformation ROP

The **RopReadPerUserInformation** ROP gets per-user information for a public folder. For more details about this operation, see [\[MS-OXCSTOR\]](#) section 2.2.1.12.

#### 2.2.3.12.1 RopReadPerUserInformation ROP Request Buffer

The following descriptions define valid fields for the **RopReadPerUserInformation** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
RopId										LogonId										InputHandleIndex										FolderId									
...																																							
...																																							
...																																							
...																																							
...																																							
...																														Reserved									
DataOffset																																							
MaxDataSize																																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x63.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**FolderId (24 bytes):** A **LongTermID** structure that specifies the folder for which to get per-user information. The format of the **LongTermID** structure is specified in [\[MS-OXCDATA\]](#) section 2.2.1.3.1.

**Reserved (1 byte):** Reserved. This field is not used and is ignored by the server. This field MUST be set to 0x00.

**DataOffset (4 bytes):** An unsigned integer that specifies the location at which to start reading within the per-user information stream.

**MaxDataSize (2 bytes):** An unsigned integer that specifies the maximum number of bytes of per-user information to be retrieved.

### 2.2.3.12.2 RopReadPerUserInformation ROP Success Response Buffer

The following descriptions define valid fields for the **RopReadPerUserInformation** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...																HasFinished				DataSize											
...										Data (variable)																					
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x63.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**HasFinished (1 byte):** A Boolean that specifies whether this operation reached the end of the per-user information stream.

**DataSize (2 bytes):** An unsigned integer that specifies the size of the **Data** field.

**Data (variable):** An array of bytes. This field contains the per-user data that is returned.

### 2.2.3.12.3 RopReadPerUserInformation ROP Failure Response Buffer

The following descriptions define valid fields for the **RopReadPerUserInformation** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId								InputHandleIndex								ReturnValue															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x63.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.3.13 RopWritePerUserInformation ROP

The **RopWritePerUserInformation** ROP sets per-user information for a public folder. For details about this operation, see [\[MS-OXCSTOR\]](#) section 2.2.1.13.

#### 2.2.3.13.1 RopWritePerUserInformation ROP Request Buffer

The following descriptions define valid fields for the **RopWritePerUserInformation** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId								LogonId								InputHandleIndex								FolderId							
...																															
...																															
...																															
...																															
...																															
...																								HasFinished							
DataOffset																															
DataSize																Data (variable)															
...																															
ReplGuid (optional)																															

...
...
...

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x64.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**FolderId (24 bytes):** A **LongTermID** structure that specifies the folder to set per-user information for. The format of the **LongTermID** structure is specified in [\[MS-OXCDATA\]](#) section 2.2.1.3.1.

**HasFinished (1 byte):** A Boolean that specifies whether this operation specifies the end of the per-user information stream.

**DataOffset (4 bytes):** An unsigned integer that specifies the location in the per-user information stream to start writing.

**DataSize (2 bytes):** An unsigned integer that specifies the size of the **Data** field in bytes.

**Data (variable):** An array of bytes that is the per-user data to write. The size of this field, in bytes, is specified by the **DataSize** field.

**ReplGuid (16 bytes):** A GUID that is present when the **DataOffset** field is 0x00000000 and the logon associated with the **LogonId** field was created with the **Private** flag set in the **RopLogon** ROP request buffer, as specified in [\[MS-OXCSTOR\]](#) section 2.2.1.1.1, and is not present otherwise.

### 2.2.3.13.2 RopWritePerUserInformation ROP Response Buffer

The following descriptions define valid fields for the **RopWritePerUserInformation** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x64.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

## 2.2.4 Folder ROPs

### 2.2.4.1 RopOpenFolder ROP

The **RopOpenFolder** ROP opens an existing folder in a mailbox. For more details about this operation, see [\[MS-OXCFOld\]](#) section 2.2.1.1.

#### 2.2.4.1.1 RopOpenFolder ROP Request Buffer

The following descriptions define valid fields for the **RopOpenFolder** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								LogonId								InputHandleIndex								OutputHandleIndex							
FolderId																															
...																															
OpenModeFlags																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x02.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**OutputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the output Server object will be stored.

**FolderId (8 bytes):** A 64-bit identifier that specifies the folder to be opened.

**OpenModeFlags (1 byte):** An 8-bit flags structure that contains flags that are used to control how the folder is opened. The possible flags values are specified in [\[MS-OXCFOld\]](#) section 2.2.1.1.1.

#### 2.2.4.1.2 RopOpenFolder ROP Success Response Buffer

The following descriptions define valid fields for the **RopOpenFolder** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								OutputHandleIndex								ReturnValue															
...																HasRules								IsGhosted							
ServerCount (optional)																CheapServerCount (optional)															
Servers (variable)																															



...

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x02.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**HasRules (1 byte):** A Boolean that indicates whether the folder has **rules** associated with it.

**IsGhosed (1 byte):** A Boolean that specifies whether the folder is a ghosted folder.

**ServerCount (2 bytes):** An unsigned integer that is present if **IsGhosed** is nonzero and is not present if **IsGhosed** is zero. This value specifies the number of strings in the **Servers** field.

**CheapServerCount (2 bytes):** An unsigned integer that is present if **IsGhosed** is nonzero and is not present if **IsGhosed** is zero. This value specifies the number of values in the **Servers** field that refer to lowest-cost servers.

**Servers (variable):** A list of null-terminated ASCII strings that specify which servers have replicas (2) of this folder. This field is present if **IsGhosed** is nonzero and is not present if **IsGhosed** is zero. The number of strings contained in this field is specified by the **ServerCount** field.

**2.2.4.1.3 RopOpenFolder ROP Failure Response Buffer**

The following descriptions define valid fields for the **RopOpenFolder** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										OutputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x02.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

**2.2.4.2 RopCreateFolder ROP**

The **RopCreateFolder** ROP creates a new subfolder. For more details about this operation, see [\[MS-OXCFLD\]](#) section 2.2.1.2.

**2.2.4.2.1 RopCreateFolder ROP Request Buffer**

The following descriptions define valid fields for the **RopCreateFolder** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId					LogonId					InputHandleIndex					OutputHandleIndex																
FolderType					UseUnicodeStrings					OpenExisting					Reserved																
DisplayName (variable)																															
...																															
Comment (variable)																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x1C.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**OutputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the output Server object will be stored.

**FolderType (1 byte):** An enumeration that specifies what type of folder to create. The possible values for this enumeration are specified in [\[MS-OXCFCM\]](#) section 2.2.1.2.1.

**UseUnicodeStrings (1 byte):** A Boolean that specifies whether the **DisplayName** field and the **Comment** field contain **Unicode** characters or multibyte characters.

**OpenExisting (1 byte):** A Boolean that specifies whether this operation opens a Folder object or fails when the Folder object already exists.

**Reserved (1 byte):** Reserved. This field MUST be set to 0x00.

**DisplayName (variable):** A null-terminated multibyte string that specifies the name of the created folder. If the value of the **UseUnicodeStrings** field is nonzero, the string is composed of Unicode characters. If the value of the **UseUnicodeStrings** field is zero, the string is composed of multibyte characters.

**Comment (variable):** A null-terminated multibyte string that specifies the folder comment that is associated with the created folder. If the value of the **UseUnicodeStrings** field is nonzero, the string is composed of Unicode characters. If the value of the **UseUnicodeStrings** field is zero, the string is composed of multibyte characters.

#### 2.2.4.2.2 RopCreateFolder ROP Success Response Buffer

The following descriptions define valid fields for the **RopCreateFolder** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										OutputHandleIndex										ReturnValue											
...										FolderId																					
...										IsExistingFolder										HasRules (optional)											
IsGhosed (optional)										ServerCount (optional)										CheapServerCount (optional)											
...										Servers (variable)																					
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x1C.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**FolderId (8 bytes):** An identifier that specifies the folder created or opened.

**IsExistingFolder (1 byte):** A Boolean that indicates whether an existing folder was opened or a new folder was created.

**HasRules (1 byte):** A Boolean that indicates whether the folder has rules associated with it. This field is present if the **IsExistingFolder** field is nonzero and is not present otherwise.

**IsGhosed (1 byte):** A Boolean that indicates whether the server is an **active replica** of this folder. This field is present if the value of the **IsExistingFolder** field is nonzero and is not present otherwise.

**ServerCount (2 bytes):** An unsigned integer that is present if the values of both the **IsExistingFolder** and the **IsGhosed** fields are nonzero and is not present otherwise. This value specifies the number of strings in the **Servers** field.

**CheapServerCount (2 bytes):** An unsigned integer that is present if the values of both the **IsExistingFolder** and the **IsGhosed** fields are nonzero and is not present otherwise. This value specifies the number of values in the **Servers** field that refer to lowest-cost servers.

**Servers (variable):** A list of null-terminated ASCII strings that is present if the values of both the **IsExistingFolder** and the **IsGhosed** fields are nonzero and is not present otherwise. The number of strings contained in this field is specified by the **ServersCount** field. These strings specify which servers have replicas (2) of this folder.

#### 2.2.4.2.3 RopCreateFolder ROP Failure Response Buffer

The following descriptions define valid fields for the **RopCreateFolder** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
RopId										OutputHandleIndex										ReturnValue													
...																																	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x1C.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.4.3 RopDeleteFolder ROP

The **RopDeleteFolder** ROP deletes a subfolder. For more details about this operation, see [\[MS-OXCFOLD\]](#) section 2.2.1.3.

#### 2.2.4.3.1 RopDeleteFolder ROP Request Buffer

The following descriptions define valid fields for the **RopDeleteFolder** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
RopId										LogonId										InputHandleIndex										DeleteFolderFlags									
FolderId																																							
...																																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x1D.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**DeleteFolderFlags (1 byte):** A flags structure that contains flags that control how to delete the folder. The possible flags values are specified in [\[MS-OXCFOLD\]](#) section 2.2.1.3.1.

**FolderId (8 bytes):** An identifier that specifies the folder to be deleted.

#### 2.2.4.3.2 RopDeleteFolder ROP Response Buffer

The following descriptions define valid fields for the **RopDeleteFolder** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...																PartialCompletion															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x1D.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

**PartialCompletion (1 byte):** A Boolean that specifies whether the operation was partially completed.

### 2.2.4.4 RopSetSearchCriteria ROP

The **RopSetSearchCriteria** ROP sets the **search criteria** for a **search folder**. For more details about this operation, see [\[MS-OXCFOLD\]](#) section 2.2.1.4.

#### 2.2.4.4.1 RopSetSearchCriteria ROP Request Buffer

The following descriptions define valid fields for the **RopSetSearchCriteria** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31								
RopId										LogonId										InputHandleIndex										RestrictionDataSize									
...																RestrictionData (variable)																							
...																...																							
FolderIdCount																FolderIds (variable)																							
...																...																							
SearchFlags																																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x30.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**RestrictionDataSize (2 bytes):** An unsigned integer that specifies the length of the **RestrictionData** field.

**RestrictionData (variable):** A **restriction** packet, as specified in [\[MS-OXCDATA\]](#) section 2.12, that specifies the filter for this search folder. The size of this field is specified by the **RestrictionDataSize** field.

**FolderIdCount (2 bytes):** An unsigned integer that specifies the number of identifiers in the **FolderIds** field.

**FolderIds (variable):** An array of 64-bit identifiers that specifies which folders are searched. The number of identifiers contained in this field is specified by the **FolderIdCount** field.

**SearchFlags (4 bytes):** A flags structure that contains flags that control the search for a search folder.

#### 2.2.4.4.2 RopSetSearchCriteria ROP Response Buffer

The following descriptions define valid fields for the **RopSetSearchCriteria** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x30.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

#### 2.2.4.5 RopGetSearchCriteria ROP

The **RopGetSearchCriteria** ROP gets the search criteria for a search folder. For more details about this operation, see [\[MS-OXCFOLD\]](#) section 2.2.1.5.

##### 2.2.4.5.1 RopGetSearchCriteria ROP Request Buffer

The following descriptions define valid fields for the **RopGetSearchCriteria** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31								
RopId										LogonId										InputHandleIndex										UseUnicode									
IncludeRestriction										IncludeFolders																													

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x31.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**UseUnicode (1 byte):** A Boolean that specifies whether the client requests the restriction data (returned in the **RestrictionData** field of the response) to be specified with Unicode strings or with ASCII strings.

**IncludeRestriction (1 byte):** A Boolean that specifies whether the server includes the restriction information in the response.

**IncludeFolders (1 byte):** A Boolean that specifies whether the server includes the folders list in the response.

### 2.2.4.5.2 RopGetSearchCriteria ROP Success Response Buffer

The following descriptions define valid fields for the **RopGetSearchCriteria** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...																RestrictionDataSize															
RestrictionData (variable)																															
...																															
LogonId										FolderIdCount										FolderIds (variable)											
...																															
SearchFlags																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x31.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**RestrictionDataSize (2 bytes):** An unsigned integer that specifies the length of the **RestrictionData** field.

**RestrictionData (variable):** A restriction packet, as specified in [\[MS-OXCDATA\]](#) section 2.12, that specifies the filter for this search folder. The size of this field is specified by the **RestrictionDataSize** field.

**LogonId (1 byte):** An unsigned integer that MUST be set to the value of the **LogonId** field in the request.

**FolderIdCount (2 bytes):** An unsigned integer that specifies the number of identifiers in the **FolderIds** field.

**FolderIds (variable):** An array of 64-bit identifiers that specifies which folders are searched. The number of identifiers contained in this field is specified by the **FolderIdCount** field.

**SearchFlags (4 bytes):** A flags structure that contains flags that control the search for a search folder. The possible values for this field are specified in [\[MS-OXCFOLD\]](#) section 2.2.1.5.2.

### 2.2.4.5.3 RopGetSearchCriteria ROP Failure Response Buffer

The following descriptions define valid fields for the **RopGetSearchCriteria** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31				
RopId										InputHandleIndex										ReturnValue															
...																																			

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x31.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.4.6 RopMoveCopyMessages ROP

The **RopMoveCopyMessages** ROP either moves or copies messages from a source folder to a destination folder. For more details about this operation, see [\[MS-OXCFOLD\]](#) section 2.2.1.6.

#### 2.2.4.6.1 RopMoveCopyMessages ROP Request Buffer

The following descriptions define valid fields for the **RopMoveCopyMessages** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31										
RopId										LogonId										SourceHandleIndex										DestHandleIndex											
MessageIdCount																MessageIds (variable)																									
...																																									
WantAsynchronous																WantCopy																									

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x33.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.



**SourceHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the source Server object is stored. For more information about Server objects, see section [1.3.1](#).

**DestHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the destination Server object is stored.

**MessageIdCount (2 bytes):** An unsigned integer that specifies the size of the **MessageIds** field.

**MessageIds (variable):** An array of 64-bit identifiers that specifies which messages to move or copy. The number of identifiers contained in this field is specified by the **MessageIdCount** field.

**WantAsynchronous (1 byte):** A Boolean that specifies whether the operation is to be processed asynchronously with status reported via the **RopProgress** ROP (section [2.2.8.13](#)).

**WantCopy (1 byte):** A Boolean that specifies whether the operation is a copy or a move.

#### 2.2.4.6.2 RopMoveCopyMessages ROP Response Buffer

The following descriptions define valid fields for the **RopMoveCopyMessages** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										SourceHandleIndex										ReturnValue											
...															PartialCompletion																

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x33.

**SourceHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **SourceHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000503.

**PartialCompletion (1 byte):** A Boolean that indicates whether the operation was only partially completed.

#### 2.2.4.6.3 RopMoveCopyMessages ROP Null Destination Failure Response Buffer

The following descriptions define valid fields for the **RopMoveCopyMessages** ROP null destination failure response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										SourceHandleIndex										ReturnValue											
...															DestHandleIndex																
...															PartialCompletion																

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x33.

**SourceHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **SourceHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000503.

**DestHandleIndex (4 bytes):** An unsigned integer index that MUST be set to the value specified in the **DestHandleIndex** field in the request.

**PartialCompletion (1 byte):** A Boolean that indicates whether the operation was only partially completed.

### 2.2.4.7 RopMoveFolder ROP

The **RopMoveFolder** ROP moves a folder. For more details about this operation, see [\[MS-OXCFC\]](#) section 2.2.1.7.

#### 2.2.4.7.1 RopMoveFolder ROP Request Buffer

The following descriptions define valid fields for the **RopMoveFolder** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								LogonId								SourceHandleIndex								DestHandleIndex							
WantAsynchronous								UseUnicode								FolderId															
...																															
...																NewFolderName (variable)															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x35.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**SourceHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the source Server object is stored. For more information about Server objects, see section [1.3.1](#).

**DestHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the destination Server object is stored.

**WantAsynchronous (1 byte):** A Boolean that specifies whether the operation is to be processed asynchronously with status reported via the **RopProgress** ROP (section [2.2.8.13](#)).

**UseUnicode (1 byte):** A Boolean that specifies whether the **NewFolderName** field contains Unicode characters or multibyte characters.

**FolderId (8 bytes):** An identifier that specifies the folder to be moved.

**NewFolderName (variable):** A null-terminated multibyte string that specifies the name for the new moved folder. If the value of the **UseUnicode** field is nonzero, the string is composed of Unicode characters. Otherwise, the string is composed of multibyte characters.

### 2.2.4.7.2 RopMoveFolder ROP Response Buffer

The following descriptions define valid fields for the **RopMoveFolder** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										SourceHandleIndex										ReturnValue											
...										PartialCompletion																					

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x35.

**SourceHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **SourceHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000503.

**PartialCompletion (1 byte):** A Boolean that indicates whether the operation was only partially completed.

### 2.2.4.7.3 RopMoveFolder ROP Null Destination Failure Response Buffer

The following descriptions define valid fields for the **RopMoveFolder** ROP null destination failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										SourceHandleIndex										ReturnValue											
...										DestHandleIndex																					
...										PartialCompletion																					

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x35.

**SourceHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **SourceHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000503.

**DestHandleIndex (4 bytes):** An unsigned integer index that MUST be set to the value specified in the **DestHandleIndex** field in the request.

**PartialCompletion (1 byte):** A Boolean that indicates whether the operation was only partially completed.

### 2.2.4.8 RopCopyFolder ROP

The **RopCopyFolder** ROP copies a folder. For more details about this operation, see [\[MS-OXCFLD\]](#) section 2.2.1.8.

#### 2.2.4.8.1 RopCopyFolder ROP Request Buffer

The following descriptions define valid fields for the **RopCopyFolder** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31								
RopId										LogonId										SourceHandleIndex										DestHandleIndex									
WantAsynchronous										WantRecursive										UseUnicode										FolderId									
...																																							
...																								NewFolderName (variable)															
...																																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x36.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**SourceHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the source Server object is stored. For more information about Server objects, see section [1.3.1](#).

**DestHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the destination Server object is stored.

**WantAsynchronous (1 byte):** A Boolean that specifies whether the operation is to be processed asynchronously with status reported via the **RopProgress** ROP (section [2.2.8.13](#)).

**WantRecursive (1 byte):** A Boolean that specifies that the copy is recursive.

**UseUnicode (1 byte):** A Boolean that specifies whether the **NewFolderName** field contains Unicode characters or multibyte characters.

**FolderId (8 bytes):** An identifier that specifies the folder to be copied.

**NewFolderName (variable):** A null-terminated multibyte string that specifies the name for the new copied folder. If the value of the **UseUnicode** field is nonzero, the string is composed of Unicode characters. Otherwise, the string is composed of multibyte characters.

#### 2.2.4.8.2 RopCopyFolder ROP Response Buffer

The following descriptions define valid fields for the **RopCopyFolder** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										SourceHandleIndex										ReturnValue											
...										PartialCompletion																					

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x36.

**SourceHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **SourceHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000503.

**PartialCompletion (1 byte):** A Boolean that indicates whether the operation was only partially completed.

### 2.2.4.8.3 RopCopyFolder ROP Null Destination Failure Response Buffer

The following descriptions define valid fields for the **RopCopyFolder** ROP null destination failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										SourceHandleIndex										ReturnValue											
...										DestHandleIndex																					
...										PartialCompletion																					

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x36.

**SourceHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **SourceHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000503.

**DestHandleIndex (4 bytes):** An unsigned integer index that MUST be set to the value specified in the **DestHandleIndex** field in the request.

**PartialCompletion (1 byte):** A Boolean that indicates whether the operation was only partially completed.

### 2.2.4.9 RopEmptyFolder ROP

The **RopEmptyFolder** ROP deletes all messages and subfolders from a folder. For more details about this operation, see [\[MS-OXCFOLD\]](#) section 2.2.1.9.

#### 2.2.4.9.1 RopEmptyFolder ROP Request Buffer

The following descriptions define valid fields for the **RopEmptyFolder** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1						
RopId										LogonId										InputHandleIndex						WantAsynchronous											
WantDeleteAssociated																																					

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x58.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**WantAsynchronous (1 byte):** A Boolean that specifies whether the operation is to be processed asynchronously with status reported via the **RopProgress** ROP (section [2.2.8.13](#)).

**WantDeleteAssociated (1 byte):** A Boolean that specifies whether the operation also deletes **folder associated information (FAI)** messages.

#### 2.2.4.9.2 RopEmptyFolder ROP Response Buffer

The following descriptions define valid fields for the **RopEmptyFolder** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										InputHandleIndex										ReturnValue											
...										PartialCompletion																					

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x58.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

**PartialCompletion (1 byte):** A Boolean that indicates whether the operation was only partially completed.

#### 2.2.4.10 RopHardDeleteMessagesAndSubfolders ROP

The **RopHardDeleteMessagesAndSubfolders** ROP **hard deletes** messages and subfolders from a folder. For more details about this operation, see [\[MS-OXCFOld\]](#) section 2.2.1.10.

##### 2.2.4.10.1 RopHardDeleteMessagesAndSubfolders ROP Request Buffer

The following descriptions define valid fields for the **RopHardDeleteMessagesAndSubfolders** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								LogonId								InputHandleIndex								WantAsynchronous							
WantDeleteAssociated																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x92.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**WantAsynchronous (1 byte):** A Boolean that specifies whether the operation is to be processed asynchronously with status reported via the **RopProgress** ROP (section [2.2.8.13](#)).

**WantDeleteAssociated (1 byte):** A Boolean that specifies whether to also delete FAI messages.

#### 2.2.4.10.2 RopHardDeleteMessagesAndSubfolders ROP Response Buffer

The following descriptions define valid fields for the **RopHardDeleteMessagesAndSubfolders** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								InputHandleIndex								ReturnValue															
...																PartialCompletion															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x92.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

**PartialCompletion (1 byte):** A Boolean that indicates whether the operation was only partially completed.

#### 2.2.4.11 RopDeleteMessages ROP

The **RopDeleteMessages** ROP deletes one or more messages in a folder. For more details about this operation, see [\[MS-OXCFCOLD\]](#) section 2.2.1.11.

##### 2.2.4.11.1 RopDeleteMessages ROP Request Buffer

The following descriptions define valid fields for the **RopDeleteMessages** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								LogonId								InputHandleIndex								WantAsynchronous							
NotifyNonRead								MessageIdCount																MessageIds (variable)							
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x1E.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**WantAsynchronous (1 byte):** A Boolean that specifies whether the operation is to be processed asynchronously with status reported via the **RopProgress** ROP (section [2.2.8.13](#)).

**NotifyNonRead (1 byte):** A Boolean that specifies whether the server sends a **non-read receipt** to the message sender when a message is deleted.

**MessageIdCount (2 bytes):** An unsigned integer that specifies the number of identifiers in the **MessageIds** field.

**MessageIds (variable):** An array of 64-bit identifiers that specifies the messages to be deleted. The number of identifiers contained in this field is specified by the **MessageIdCount** field.

#### 2.2.4.11.2 RopDeleteMessages ROP Response Buffer

The following descriptions define valid fields for the **RopDeleteMessages** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								InputHandleIndex								ReturnValue															
...																PartialCompletion															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x1E.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

**PartialCompletion (1 byte):** A Boolean that specifies whether this operation partially completed.

#### 2.2.4.12 RopHardDeleteMessages ROP

The **RopHardDeleteMessages** ROP hard deletes messages in a folder. For more details about this operation, see [\[MS-OXCFCOLD\]](#) section 2.2.1.12.



### 2.2.4.12.1 RopHardDeleteMessages ROP Request Buffer

The following descriptions define valid fields for the **RopHardDeleteMessages** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								LogonId								InputHandleIndex								WantAsynchronous							
NotifyNonRead								MessageIdCount																MessageIds (variable)							
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x91.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**WantAsynchronous (1 byte):** A Boolean that specifies whether the operation is to be processed asynchronously with status reported via the **RopProgress** ROP (section [2.2.8.13](#)).

**NotifyNonRead (1 byte):** A Boolean that specifies whether the server sends a non-read receipt to the message-sender when a message is deleted.

**MessageIdCount (2 bytes):** An unsigned integer that specifies the number of identifiers in the **MessageIds** field.

**MessageIds (variable):** An array of 64-bit identifiers that specifies the messages to be hard deleted. The number of identifiers contained in this field is specified by the **MessageIdCount** field.

### 2.2.4.12.2 RopHardDeleteMessages ROP Response Buffer

The following descriptions define valid fields for the **RopHardDeleteMessages** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								InputHandleIndex								ReturnValue															
...																PartialCompletion															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x91.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

**PartialCompletion (1 byte):** A Boolean that indicates whether the operation was only partially completed.

### 2.2.4.13 RopGetHierarchyTable ROP

The **RopGetHierarchyTable** ROP gets the subfolder **hierarchy table** for a folder. For more details about this operation, see [\[MS-OXCFOLD\]](#) section 2.2.1.13.

#### 2.2.4.13.1 RopGetHierarchyTable ROP Request Buffer

The following descriptions define valid fields for the **RopGetHierarchyTable** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										LogonId										InputHandleIndex						OutputHandleIndex					
TableFlags																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x04.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**OutputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the output Server object will be stored.

**TableFlags (1 byte):** A flags structure. The possible values are specified in [\[MS-OXCFOLD\]](#) section 2.2.1.13.1. These flags control the type of table.

#### 2.2.4.13.2 RopGetHierarchyTable ROP Success Response Buffer

The following descriptions define valid fields for the **RopGetHierarchyTable** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										OutputHandleIndex										ReturnValue											
...																RowCount															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x04.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**RowCount (4 bytes):** An unsigned integer that represents the number of rows in the hierarchy table.

### 2.2.4.13.3 RopGetHierarchyTable ROP Failure Response Buffer

The following descriptions define valid fields for the **RopGetHierarchyTable** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										OutputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x04.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.4.14 RopGetContentsTable ROP

The **RopGetContentsTable** ROP gets the **contents table** of a container. For more details about this operation, see [\[MS-OXCFOld\]](#) section 2.2.1.14.

#### 2.2.4.14.1 RopGetContentsTable ROP Request Buffer

The following descriptions define valid fields for the **RopGetContentsTable** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										LogonId										InputHandleIndex						OutputHandleIndex					
TableFlags																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x05.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**OutputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the output Server object will be stored.

**TableFlags (1 byte):** A flags structure that contains flags control the type of table. The possible values are specified in [\[MS-OXCFOld\]](#) section 2.2.1.14.1.

### 2.2.4.14.2 RopGetContentsTable ROP Success Response Buffer

The following descriptions define valid fields for the **RopGetContentsTable** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										OutputHandleIndex										ReturnValue											
...										RowCount																					
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x05.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**RowCount (4 bytes):** An unsigned integer that represents the number of rows in the contents table.

### 2.2.4.14.3 RopGetContentsTable ROP Failure Response Buffer

The following descriptions define valid fields for the **RopGetContentsTable** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										OutputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x05.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

## 2.2.5 Table ROPs

### 2.2.5.1 RopSetColumns ROP

The **RopSetColumns** ROP sets the properties visible on a table. For more details about this operation, see [\[MS-OXCTABL\]](#) section 2.2.2.2.

#### 2.2.5.1.1 RopSetColumns ROP Request Buffer

The following descriptions define valid fields for the **RopSetColumns** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
RopId										LogonId										InputHandleIndex						SetColumnsFlags							
PropertyTagCount																PropertyTags (variable)																	
...																																	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x12.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**SetColumnsFlags (1 byte):** A flags structure that contains flags that control this operation. The possible values are specified in [\[MS-OXCTABL\]](#) section 2.2.2.2.1.

**PropertyTagCount (2 bytes):** An unsigned integer that specifies the number of tags present in the **PropertyTags** field.

**PropertyTags (variable):** An array of **PropertyTag** structures that specifies the property values that are visible in table rows. The number of structures contained in this field is specified by the **PropertyTagCount** field. The format of the **PropertyTag** structure is specified in [\[MS-OXCDATA\]](#) section 2.9.

#### 2.2.5.1.2 RopSetColumns ROP Success Response Buffer

The following descriptions define valid fields for the **RopSetColumns** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
RopId										InputHandleIndex										ReturnValue													
...																TableStatus																	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x12.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**TableStatus (1 byte):** An enumeration that specifies the status of the table. The possible values for this enumeration are specified in [\[MS-OXCTABL\]](#) section 2.2.2.1.3.

### 2.2.5.1.3 RopSetColumns ROP Failure Response Buffer

The following descriptions define valid fields for the **RopSetColumns** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x12.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.5.2 RopSortTable ROP

The **RopSortTable** ROP defines the order of rows of a table based on sort criteria. For more details about this operation, see [\[MS-OXCTABL\]](#) section 2.2.2.3.

#### 2.2.5.2.1 RopSortTable ROP Request Buffer

The following descriptions define valid fields for the **RopSortTable** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31								
RopId										LogonId										InputHandleIndex										SortTableFlags									
SortOrderCount																CategoryCount																							
ExpandedCount																SortOrders (variable)																							
...																																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x13.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**SortTableFlags (1 byte):** A flags structure that contains flags that control this operation. The possible values are specified in [\[MS-OXCTABL\]](#) section 2.2.2.3.1.

**SortOrderCount (2 bytes):** An unsigned integer that specifies how many **SortOrder** structures are present in the **SortOrders** field. The format of the **SortOrder** structure is specified in [\[MS-OXCDATA\]](#) section 2.13.1.

**CategoryCount (2 bytes):** An unsigned integer that specifies the number of category **SortOrder** structures in the **SortOrders** field.

**ExpandedCount (2 bytes):** An unsigned integer that specifies the number of expanded categories in the **SortOrders** field.

**SortOrders (variable):** An array of **SortOrder** structures that specifies the sort order for the rows in the table. The number of structures contained in this field is specified by the **SortOrderCount** field.

### 2.2.5.2.2 RopSortTable ROP Success Response Buffer

The following descriptions define valid fields for the **RopSortTable** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...															TableStatus																

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x13.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**TableStatus (1 byte):** An enumeration that specifies the status of the table. The possible values for this enumeration are specified in [\[MS-OXCTABL\]](#) section 2.2.2.1.3.

### 2.2.5.2.3 RopSortTable ROP Failure Response Buffer

The following descriptions define valid fields for the **RopSortTable** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x13.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.5.3 RopRestrict ROP

The **RopRestrict** ROP establishes a filter for a table. For more details about this operation, see [\[MS-OXCTABL\]](#) section 2.2.2.4.

#### 2.2.5.3.1 RopRestrict ROP Request Buffer

The following descriptions define valid fields for the **RopRestrict** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31								
RopId										LogonId										InputHandleIndex										RestrictFlags									
RestrictionDataSize																RestrictionData (variable)																							
...																																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x14.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**RestrictFlags (1 byte):** A flags structure that contains flags that control this operation. The possible values are specified in [\[MS-OXCTABL\]](#) section 2.2.2.4.1.

**RestrictionDataSize (2 bytes):** An unsigned integer that specifies the length of the **RestrictionData** field.

**RestrictionData (variable):** A restriction packet, as specified in [\[MS-OXCDATA\]](#) section 2.12, that specifies the filter for this table. The size of this field is specified by the **RestrictionDataSize** field.

#### 2.2.5.3.2 RopRestrict ROP Success Response Buffer

The following descriptions define valid fields for the **RopRestrict** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											



...	TableStatus
-----	-------------

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x14.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**TableStatus (1 byte):** An enumeration that specifies the status of the table. The possible values for this enumeration are specified in [\[MS-OXCTABL\]](#) section 2.2.2.1.3.

### 2.2.5.3.3 RopRestrict ROP Failure Response Buffer

The following descriptions define valid fields for the **RopRestrict** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x14.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.5.4 RopQueryRows ROP

The **RopQueryRows** ROP retrieves rows from a table. For more details about this operation, see [\[MS-OXCTABL\]](#) section 2.2.2.5.

#### 2.2.5.4.1 RopQueryRows ROP Request Buffer

The following descriptions define valid fields for the **RopQueryRows** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										LogonId										InputHandleIndex						QueryRowsFlags					
ForwardRead										RowCount																					

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x15.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**QueryRowsFlags (1 byte):** A flags structure that contains flags that control this operation. The possible values are specified in [\[MS-OXCTABL\]](#) section 2.2.2.5.1.

**ForwardRead (1 byte):** A Boolean that specifies the direction to read rows.

**RowCount (2 bytes):** An unsigned integer that specifies the number of requested rows.

#### 2.2.5.4.2 RopQueryRows ROP Success Response Buffer

The following descriptions define valid fields for the **RopQueryRows** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								InputHandleIndex								ReturnValue															
...																Origin								RowCount							
...								RowData (variable)																							
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x15.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**Origin (1 byte):** An enumeration that specifies current location of the cursor. The possible values for this enumeration are specified in [\[MS-OXCTABL\]](#) section 2.2.2.5.2.

**RowCount (2 bytes):** An unsigned integer that specifies the number of structures in the **RowData** field.

**RowData (variable):** A list of **PropertyRow** structures. The number of structures contained in this field is specified by the **RowCount** field. The format of the **PropertyRow** structure is specified in [\[MS-OXCDATA\]](#) section 2.8. The columns used for these rows were those previously set on this table by a **RopSetColumns** ROP request (section [2.2.5.1](#)).

#### 2.2.5.4.3 RopQueryRows ROP Failure Response Buffer

The following descriptions define valid fields for the **RopQueryRows** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								InputHandleIndex								ReturnValue															

...

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x15.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.5.5 RopAbort ROP

The **RopAbort** ROP aborts an asynchronous table operation in progress. For more details about this operation, see [\[MS-OXCTABL\]](#) section 2.2.2.6.

#### 2.2.5.5.1 RopAbort ROP Request Buffer

The following descriptions define valid fields for the **RopAbort** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										LogonId										InputHandleIndex											

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x38.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

#### 2.2.5.5.2 RopAbort ROP Success Response Buffer

The following descriptions define valid fields for the **RopAbort** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...																TableStatus															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x38.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**TableStatus (1 byte):** An enumeration that specifies the status of the table. The possible values for this enumeration are specified in [\[MS-OXCTABL\]](#) section 2.2.2.1.3.

### 2.2.5.5.3 RopAbort ROP Failure Response Buffer

The following descriptions define valid fields for the **RopAbort** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
RopId										InputHandleIndex										ReturnValue													
...																																	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x38.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.5.6 RopGetStatus ROP

The **RopGetStatus** ROP gets the status of a table. For more details about this operation, see [\[MS-OXCTABL\]](#) section 2.2.2.7.

#### 2.2.5.6.1 RopGetStatus ROP Request Buffer

The following descriptions define valid fields for the **RopGetStatus** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										LogonId										InputHandleIndex											

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x16.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

#### 2.2.5.6.2 RopGetStatus ROP Success Response Buffer

The following descriptions define valid fields for the **RopGetStatus** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										InputHandleIndex										ReturnValue											

...	TableStatus
-----	-------------

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x16.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**TableStatus (1 byte):** An enumeration that specifies the status of the table. The possible values for this enumeration are specified in [\[MS-OXCTABL\]](#) section 2.2.2.1.3.

### 2.2.5.6.3 RopGetStatus ROP Failure Response Buffer

The following descriptions define valid fields for the **RopGetStatus** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
RopId										InputHandleIndex										ReturnValue													
...																																	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x16.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.5.7 RopQueryPosition ROP

The **RopQueryPosition** ROP gets the cursor position. For more details about this operation, see [\[MS-OXCTABL\]](#) section 2.2.2.8.

#### 2.2.5.7.1 RopQueryPosition ROP Request Buffer

The following descriptions define valid fields for the **RopQueryPosition** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										LogonId										InputHandleIndex											

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x17.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

### 2.2.5.7.2 RopQueryPosition ROP Success Response Buffer

The following descriptions define valid fields for the **RopQueryPosition** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										InputHandleIndex										ReturnValue											
...										...										Numerator											
...										...										Denominator											
...										...										...											

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x17.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**Numerator (4 bytes):** An unsigned integer that represents the numerator of the fraction identifying the table position.

**Denominator (4 bytes):** An unsigned integer that represents the denominator of the fraction identifying the table position.

### 2.2.5.7.3 RopQueryPosition ROP Failure Response Buffer

The following descriptions define valid fields for the **RopQueryPosition** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										InputHandleIndex										ReturnValue											
...										...										...											

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x17.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.5.8 RopSeekRow ROP

The **RopSeekRow** ROP moves the cursor to a specific position in a table. For more details about this operation, see [\[MS-OXCTABL\]](#) section 2.2.2.9.

#### 2.2.5.8.1 RopSeekRow ROP Request Buffer

The following descriptions define valid fields for the **RopSeekRow** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								LogonId								InputHandleIndex								Origin							
RowCount																															
WantRowMovedCount																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x18.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**Origin (1 byte):** An enumeration that specifies the origin of this seek operation. The possible values for this enumeration are specified in [\[MS-OXCTABL\]](#) section 2.2.2.5.2.

**RowCount (4 bytes):** A signed integer that specifies the direction and the number of rows to seek.

**WantRowMovedCount (1 byte):** A Boolean that specifies whether the server returns the actual number of rows moved in the response.

#### 2.2.5.8.2 RopSeekRow ROP Success Response Buffer

The following descriptions define valid fields for the **RopSeekRow** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								InputHandleIndex								ReturnValue															
...																HasSoughtLess								RowsSought							
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x18.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**HasSoughtLess (1 byte):** A Boolean that specifies whether the full number of rows sought past was less than the number that was requested.

**RowsSought (4 bytes):** A signed integer that specifies the direction and number of rows sought.

### 2.2.5.8.3 RopSeekRow ROP Failure Response Buffer

The following descriptions define valid fields for the **RopSeekRow** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								InputHandleIndex								ReturnValue															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x18.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.5.9 RopSeekRowBookmark ROP

The **RopSeekRowBookmark** ROP moves the cursor to a location specified relative to a user-defined **bookmark**. For more details about this operation, see [\[MS-OXCTABL\]](#) section 2.2.2.10.

#### 2.2.5.9.1 RopSeekRowBookmark ROP Request Buffer

The following descriptions define valid fields for the **RopSeekRowBookmark** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								LogonId								InputHandleIndex								BookmarkSize							
...								Bookmark (variable)																							
...																															
RowCount																															
WantRowMovedCount																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x19.



**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**BookmarkSize (2 bytes):** An unsigned integer that specifies the size of the **Bookmark** field.

**Bookmark (variable):** An array of bytes that specifies the origin for the seek operation. The size of this field, in bytes, is specified by the **BookmarkSize** field.

**RowCount (4 bytes):** A signed integer that specifies the direction and the number of rows to seek.

**WantRowMovedCount (1 byte):** A Boolean that specifies whether the server returns the actual number of rows sought in the response.

### 2.2.5.9.2 RopSeekRowBookmark ROP Success Response Buffer

The following descriptions define valid fields for the **RopSeekRowBookmark** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31			
RopId										InputHandleIndex										ReturnValue														
...															RowNoLongerVisible										HasSoughtLess									
RowsSought																																		

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x19.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**RowNoLongerVisible (1 byte):** A Boolean that specifies whether the bookmark target is no longer visible.

**HasSoughtLess (1 byte):** A Boolean that specifies whether the full number of rows sought past was less than the number that was requested.

**RowsSought (4 bytes):** An unsigned integer that specifies the direction and number of rows sought.

### 2.2.5.9.3 RopSeekRowBookmark ROP Failure Response Buffer

The following descriptions define valid fields for the **RopSeekRowBookmark** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId								InputHandleIndex								ReturnValue															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x19.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.5.10 RopSeekRowFractional ROP

The **RopSeekRowFractional** ROP moves the cursor to an approximate position in a table. For more details about this operation, see [\[MS-OXCTABL\]](#) section 2.2.2.11.

#### 2.2.5.10.1 RopSeekRowFractional ROP Request Buffer

The following descriptions define valid fields for the **RopSeekRowFractional** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId								LogonId								InputHandleIndex								Numerator							
...																															
Denominator																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x1A.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**Numerator (4 bytes):** An unsigned integer that represents the numerator of the fraction identifying the table position to seek to.

**Denominator (4 bytes):** An unsigned integer that represents the denominator of the fraction identifying the table position to seek to.

#### 2.2.5.10.2 RopSeekRowFractional ROP Response Buffer

The following descriptions define valid fields for the **RopSeekRowFractional** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
RopId										InputHandleIndex										ReturnValue													
...																																	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x1A.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.5.11 RopCreateBookmark ROP

The **RopCreateBookmark** ROP marks the current cursor position in a table. For more details about this operation, see [\[MS-OXCTABL\]](#) section 2.2.2.12.

#### 2.2.5.11.1 RopCreateBookmark ROP Request Buffer

The following descriptions define valid fields for the **RopCreateBookmark** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										LogonId										InputHandleIndex											

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x1B.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

#### 2.2.5.11.2 RopCreateBookmark ROP Success Response Buffer

The following descriptions define valid fields for the **RopCreateBookmark** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
RopId										InputHandleIndex										ReturnValue													
...										BookmarkSize																							
Bookmark (variable)																																	
...																																	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x1B.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**BookmarkSize (2 bytes):** An unsigned integer that specifies the size of the **Bookmark** field.

**Bookmark (variable):** An array of bytes that specifies the bookmark created. The size of this field, in bytes, is specified by the **BookmarkSize** field.

### 2.2.5.11.3 RopCreateBookmark ROP Failure Response Buffer

The following descriptions define valid fields for the **RopCreateBookmark** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x1B.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.5.12 RopQueryColumnsAll ROP

The **RopQueryColumnsAll** ROP gets a list of columns in a table. For more details about this operation, see [\[MS-OXCTABL\]](#) section 2.2.2.13.

#### 2.2.5.12.1 RopQueryColumnsAll ROP Request Buffer

The following descriptions define valid fields for the **RopQueryColumnsAll** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										LogonId										InputHandleIndex											

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x37.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

### 2.2.5.12.2 RopQueryColumnsAll ROP Success Response Buffer

The following descriptions define valid fields for the **RopQueryColumnsAll** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										InputHandleIndex										ReturnValue											
...										PropertyTagCount																					
PropertyTags (variable)																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x37.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**PropertyTagCount (2 bytes):** An unsigned integer that specifies how many tags are present in the **PropertyTags** field.

**PropertyTags (variable):** An array of **PropertyTag** structures that specifies the columns of the table. The number of structures contained in this field is specified by the **PropertyTagCount** field. The format of the **PropertyTag** structure is specified in [\[MS-OXCDATA\]](#) section 2.9.

### 2.2.5.12.3 RopQueryColumnsAll ROP Failure Response Buffer

The following descriptions define valid fields for the **RopQueryColumnsAll** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x37.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.5.13 RopFindRow ROP

The **RopFindRow** ROP moves the cursor to a row in a table that matches specific search criteria. For more details about this operation, see [\[MS-OXCTABL\]](#) section 2.2.2.14.

#### 2.2.5.13.1 RopFindRow ROP Request Buffer

The following descriptions define valid fields for the **RopFindRow** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										LogonId										InputHandleIndex						FindRowFlags					
RestrictionDataSize																RestrictionData (variable)															
...																															
Origin										BookmarkSize																Bookmark (variable)					
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x4F.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**FindRowFlags (1 byte):** A flags structure that contains flags that control this operation. The possible values are specified in [\[MS-OXCTABL\]](#) section 2.2.2.14.1.

**RestrictionDataSize (2 bytes):** An unsigned integer that specifies the length of the **RestrictionData** field.

**RestrictionData (variable):** A restriction packet, as specified in [\[MS-OXCADATA\]](#) section 2.12, that specifies the filter for this operation. The size of this field, in bytes, is specified by the **RestrictionDataSize** field.

**Origin (1 byte):** An enumeration that specifies where this operation begins its search. The possible values for this enumeration are specified in [\[MS-OXCTABL\]](#) section 2.2.2.5.2.

**BookmarkSize (2 bytes):** An unsigned integer that specifies the size of the **Bookmark** field.

**Bookmark (variable):** An array of bytes that specifies the bookmark to use as the origin. The size of this field, in bytes, is specified by the **BookmarkSize** field.

#### 2.2.5.13.2 RopFindRow ROP Success Response Buffer

The following descriptions define valid fields for the **RopFindRow** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
RopId										InputHandleIndex										ReturnValue													
...										RowNoLongerVisible										HasRowData													
RowData (variable)																																	
...																																	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x4F.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**RowNoLongerVisible (1 byte):** A Boolean that specifies whether the bookmark target is no longer visible.

**HasRowData (1 byte):** A Boolean that indicates whether the **RowData** field is present.

**RowData (variable):** A **PropertyRow** structure that is present only when the **HasRowData** field is set to a nonzero value. The format of the **PropertyRow** structure is specified in [\[MS-OXCDATA\]](#) section 2.8. The columns used for these rows were those previously set on this table by a **RopSetColumns** ROP request (section [2.2.5.1](#)).

### 2.2.5.13.3 RopFindRow ROP Failure Response Buffer

The following descriptions define valid fields for the **RopFindRow** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x4F.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.5.14 RopFreeBookmark ROP

The **RopFreeBookmark** ROP releases a bookmark. For more details about this operation, see [\[MS-OXCTABL\]](#) section 2.2.2.15.

### 2.2.5.14.1 RopFreeBookmark ROP Request Buffer

The following descriptions define valid fields for the **RopFreeBookmark** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31								
RopId										LogonId										InputHandleIndex										BookmarkSize									
...										Bookmark (variable)																													
...																																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x89.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**BookmarkSize (2 bytes):** An unsigned integer that specifies the size of the **Bookmark** field.

**Bookmark (variable):** An array of bytes that specifies the bookmark to be freed. The size of this field, in bytes, is specified by the **BookmarkSize** field.

### 2.2.5.14.2 RopFreeBookmark ROP Response Buffer

The following descriptions define valid fields for the **RopFreeBookmark** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x89.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.5.15 RopResetTable ROP

The **RopResetTable** ROP resets a table to its original state. For more details about this operation, see [\[MS-OXCTABL\]](#) section 2.2.2.16.

#### 2.2.5.15.1 RopResetTable ROP Request Buffer

The following descriptions define valid fields for the **RopResetTable** ROP request buffer.



0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId									LogonId									InputHandleIndex													

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x81.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

### 2.2.5.15.2 RopResetTable ROP Response Buffer

The following descriptions define valid fields for the **RopResetTable** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId									InputHandleIndex									ReturnValue													
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x81.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.5.16 RopExpandRow ROP

The **RopExpandRow** ROP expands a categorized row. For more details about this operation, see [\[MS-OXCTABL\]](#) section 2.2.2.17.

#### 2.2.5.16.1 RopExpandRow ROP Request Buffer

The following descriptions define valid fields for the **RopExpandRow** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1				
RopId									LogonId									InputHandleIndex									MaxRowCount								
...									CategoryId																										
...																																			
...																																			

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x59.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**MaxRowCount (2 bytes):** An unsigned integer that specifies the maximum number of expanded rows to return data for.

**CategoryId (8 bytes):** An identifier that specifies the category to be expanded.

### 2.2.5.16.2 RopExpandRow ROP Success Response Buffer

The following descriptions define valid fields for the **RopExpandRow** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...										ExpandedRowCount																					
...										RowCount																					
RowData (variable)																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x59.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**ExpandedRowCount (4 bytes):** An unsigned integer that specifies the total number of rows available in the expanded category.

**RowCount (2 bytes):** An unsigned integer that specifies the number of structures present in the **RowData** field.

**RowData (variable):** A list of **PropertyRow** structures. The number of structures contained in this field is specified by the **RowCount** field. The format of the **PropertyRow** structure is specified in [\[MS-OXCDATA\]](#) section 2.8.1. The columns used for these rows were those previously set on this table by a **RopSetColumns** ROP request (section [2.2.5.1](#)).

### 2.2.5.16.3 RopExpandRow ROP Failure Response Buffer

The following descriptions define valid fields for the **RopExpandRow** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								InputHandleIndex								ReturnValue															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x59.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.5.17 RopCollapseRow ROP

The **RopCollapseRow** ROP collapses a categorized row. For more details about this operation, see [\[MS-OXCTABL\]](#) section 2.2.2.18.

#### 2.2.5.17.1 RopCollapseRow ROP Request Buffer

The following descriptions define valid fields for the **RopCollapseRow** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								LogonId								InputHandleIndex								CategoryId							
...																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x5A.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**CategoryId (8 bytes):** An identifier that specifies the category to be collapsed.

#### 2.2.5.17.2 RopCollapseRow ROP Success Response Buffer

The following descriptions define valid fields for the **RopCollapseRow** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								InputHandleIndex								ReturnValue															

...	CollapsedRowCount
...	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x5A.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**CollapsedRowCount (4 bytes):** An unsigned integer that specifies the total number of rows in the collapsed category.

### 2.2.5.17.3 RopCollapseRow ROP Failure Response Buffer

The following descriptions define valid fields for the **RopCollapseRow** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
RopId										InputHandleIndex										ReturnValue													
...																																	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x5A.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.5.18 RopGetCollapseState ROP

The **RopGetCollapseState** ROP gets the current collapse state of rows in a categorized table. For more details about this operation, see [\[MS-OXCTABL\]](#) section 2.2.2.19.

#### 2.2.5.18.1 RopGetCollapseState ROP Request Buffer

The following descriptions define valid fields for the **RopGetCollapseState** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
RopId										LogonId										InputHandleIndex										RowId									
...																																							

...	RowIndexNumber
...	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x6B.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**RowId (8 bytes):** An identifier that specifies the row to be preserved as the cursor. The cursor is returned as part of the collapse state in the **CollapseState** field of the response.

**RowIndexNumber (4 bytes):** An unsigned integer that specifies the instance number of the row that is to be preserved as the cursor.

### 2.2.5.18.2 RopGetCollapseState ROP Success Response Buffer

The following descriptions define valid fields for the **RopGetCollapseState** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...										CollapseStateSize																					
CollapseState (variable)																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x6B.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**CollapseStateSize (2 bytes):** An unsigned integer that specifies the size of the **CollapseState** field.

**CollapseState (variable):** An array of bytes that specifies a collapse state for a categorized table. The size of this field, in bytes, is specified by the **CollapseStateSize** field.

### 2.2.5.18.3 RopGetCollapseState ROP Failure Response Buffer

The following descriptions define valid fields for the **RopGetCollapseState** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId								InputHandleIndex								ReturnValue															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x6B.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.5.19 RopSetCollapseState ROP

The **RopSetCollapseState** ROP restores the collapse state of rows in a categorized table. For more details about this operation, see [\[MS-OXCTABL\]](#) section 2.2.2.20.

#### 2.2.5.19.1 RopSetCollapseState ROP Request Buffer

The following descriptions define valid fields for the **RopSetCollapseState** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId								LogonId								InputHandleIndex								CollapseStateSize							
...								CollapseState (variable)																							
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x6C.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**CollapseStateSize (2 bytes):** An unsigned integer that specifies the size of the **CollapseState** field.

**CollapseState (variable):** An array of bytes that specifies a collapse state for a categorized table. The size of this field, in bytes, is specified by the **CollapseStateSize** field.

#### 2.2.5.19.2 RopSetCollapseState ROP Success Response Buffer

The following descriptions define valid fields for the **RopSetCollapseState** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										InputHandleIndex										ReturnValue											
...										BookmarkSize																					
Bookmark (variable)																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x6C.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**BookmarkSize (2 bytes):** An unsigned integer that specifies the size of the **Bookmark** field.

**Bookmark (variable):** An array of bytes that specifies the current cursor position. The size of this field, in bytes, is specified by the **BookmarkSize** field.

### 2.2.5.19.3 RopSetCollapseState ROP Failure Response Buffer

The following descriptions define valid fields for the **RopSetCollapseState** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x6C.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

## 2.2.6 Message ROPs

### 2.2.6.1 RopOpenMessage ROP

The **RopOpenMessage** ROP opens an existing message in a mailbox. For more details about this operation, see [\[MS-OXCMSG\]](#) section 2.2.3.1.

### 2.2.6.1.1 RopOpenMessage ROP Request Buffer

The following descriptions define valid fields for the **RopOpenMessage** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								LogonId								InputHandleIndex								OutputHandleIndex							
CodePageId																FolderId															
...																															
...																OpenModeFlags								MessageId							
...																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x03.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**OutputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the output Server object will be stored.

**CodePageId (2 bytes):** An identifier that specifies which **code page** will be used for string values associated with the message.

**FolderId (8 bytes):** An identifier that identifies the parent folder of the message to be opened.

**OpenModeFlags (1 byte):** A flags structure that contains flags that control the access to the message. The possible values are specified in [\[MS-OXCMSG\]](#) section 2.2.3.1.1.

**MessageId (8 bytes):** An identifier that identifies the message to be opened.

### 2.2.6.1.2 RopOpenMessage ROP Success Response Buffer

The following descriptions define valid fields for the **RopOpenMessage** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								OutputHandleIndex								ReturnValue															
...																HasNamedProperties								SubjectPrefix (variable)							
...																															



NormalizedSubject (variable)	
...	
RecipientCount	ColumnCount
RecipientColumns (variable)	
...	
RowCount	RecipientRows (variable)
...	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x03.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**HasNamedProperties (1 byte):** A Boolean that specifies whether the message has **named properties**.

**SubjectPrefix (variable):** A **TypedString** structure that specifies the subject prefix of the message. The format of the **TypedString** structure is specified in [\[MS-OXCDATA\]](#) section 2.11.7.

**NormalizedSubject (variable):** A **TypedString** structure that specifies the normalized subject of the message.

**RecipientCount (2 bytes):** An unsigned integer that specifies the number of **recipients (1)** on the message.

**ColumnCount (2 bytes):** An unsigned integer that specifies the number of structures in the **RecipientColumns** field.

**RecipientColumns (variable):** An array of **PropertyTag** structures that specifies the property values that can be included in each row that is specified in the **RecipientRows** field. The number of structures contained in this field is specified by the **ColumnCount** field. The format of the **PropertyTag** structure is specified in [\[MS-OXCDATA\]](#) section 2.9.

**RowCount (1 byte):** An unsigned integer that specifies the number of structures in the **RecipientRows** field.

**RecipientRows (variable):** A list of **OpenRecipientRow** structures. The number of structures contained in this field is specified by the **RowCount** field. The format of the **OpenRecipientRow** structure is defined in section [2.2.6.1.2.1](#).

### 2.2.6.1.2.1 OpenRecipientRow Structure

The following descriptions define valid fields for the **OpenRecipientRow** structure.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RecipientType										CodePageId										Reserved											
...										RecipientRowSize										RecipientRow (variable)											
...																															

**RecipientType (1 byte):** An enumeration that specifies the type of recipient (2). The possible values for this enumeration are specified in [\[MS-OXCMSG\]](#) section 2.2.3.1.2.

**CodePageId (2 bytes):** An identifier that specifies the code page for the recipient (2).

**Reserved (2 bytes):** Reserved. The server MUST set this field to 0x0000.

**RecipientRowSize (2 bytes):** An unsigned integer that specifies the size of the **RecipientRow** field.

**RecipientRow (variable):** A **RecipientRow** structure. The format of this structure is specified in [\[MS-OXCDATA\]](#) section 2.8.3. The size of this field, in bytes, is specified by the **RecipientRowSize** field.

### 2.2.6.1.3 RopOpenMessage ROP Failure Response Buffer

The following descriptions define valid fields for the **RopOpenMessage** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										OutputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x03.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.6.2 RopCreateMessage ROP

The **RopCreateMessage** ROP creates a Message object in a mailbox. For more details about this operation, see [\[MS-OXCMSG\]](#) section 2.2.3.2.

#### 2.2.6.2.1 RopCreateMessage ROP Request Buffer

The following descriptions define valid fields for the **RopCreateMessage** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								LogonId								InputHandleIndex								OutputHandleIndex							
CodePageId																FolderId															
...																															
...																AssociatedFlag															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x06.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**OutputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the output Server object will be stored.

**CodePageId (2 bytes):** An identifier that specifies the code page for the message.

**FolderId (8 bytes):** An identifier that specifies the parent folder.

**AssociatedFlag (1 byte):** A Boolean that specifies whether the message is an FAI message.

#### 2.2.6.2.2 RopCreateMessage ROP Success Response Buffer

The following descriptions define valid fields for the **RopCreateMessage** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								OutputHandleIndex								ReturnValue															
...																HasMessageId								MessageId (optional)							
...																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x06.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** specified in field the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**HasMessageId (1 byte):** A Boolean that specifies whether the **MessageId** field is present.

**MessageId (8 bytes):** An identifier that is present if **HasMessageId** is nonzero and is not present if it is zero. This value is an identifier that is associated with the created message.

### 2.2.6.2.3 RopCreateMessage ROP Failure Response Buffer

The following descriptions define valid fields for the **RopCreateMessage** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										OutputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x06.

**OutputHandleIndex (1 byte):** An unsigned integer index that **MUST** be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.6.3 RopSaveChangesMessage ROP

The **RopSaveChangesMessage** ROP commits the changes made to a message. For more details about this operation, see [\[MS-OXCMSG\]](#) section 2.2.3.3.

#### 2.2.6.3.1 RopSaveChangesMessage ROP Request Buffer

The following descriptions define valid fields for the **RopSaveChangesMessage** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31								
RopId										LogonId										ResponseHandleIndex										InputHandleIndex									
SaveFlags																																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x0C.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**ResponseHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table that is referenced in the response. For more information about Server objects, see section [1.3.1](#).

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored.

**SaveFlags (1 byte):** A flags structure that contains flags that specify how the save operation behaves. The possible values are specified in [\[MS-OXCMSG\]](#) section 2.2.3.3.1.

### 2.2.6.3.2 RopSaveChangesMessage ROP Success Response Buffer

The following descriptions define valid fields for the **RopSaveChangesMessage** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								ResponseHandleIndex								ReturnValue															
...																InputHandleIndex								MessageId							
...																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x0C.

**ResponseHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **ResponseHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**MessageId (8 bytes):** An identifier that specifies the ID of the message saved.

### 2.2.6.3.3 RopSaveChangesMessage ROP Failure Response Buffer

The following descriptions define valid fields for the **RopSaveChangesMessage** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								ResponseHandleIndex								ReturnValue															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x0C.

**ResponseHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **ResponseHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

## 2.2.6.4 RopRemoveAllRecipients ROP

The **RopRemoveAllRecipients** ROP deletes all recipients (1) from a message. For more details about this operation, see [\[MS-OXCMSG\]](#) section 2.2.3.4.

### 2.2.6.4.1 RopRemoveAllRecipients ROP Request Buffer

The following descriptions define valid fields for the **RopRemoveAllRecipients** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								LogonId								InputHandleIndex								Reserved							
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x0D.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**Reserved (4 bytes):** Reserved. The client SHOULD set this field to 0x00000000. The server MUST ignore this field, regardless of its value.

### 2.2.6.4.2 RopRemoveAllRecipients ROP Response Buffer

The following descriptions define valid fields for the **RopRemoveAllRecipients** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								InputHandleIndex								ReturnValue															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x0D.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

## 2.2.6.5 RopModifyRecipients ROP

The **RopModifyRecipients** ROP adds or modifies recipients (1) on a message. For more details about this operation, see [\[MS-OXCMSG\]](#) section 2.2.3.5.

### 2.2.6.5.1 RopModifyRecipients ROP Request Buffer

The following descriptions define valid fields for the **RopModifyRecipients** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
RopId										LogonId										InputHandleIndex										ColumnCount									
...										RecipientColumns (variable)																													
...																																							
RowCount															RecipientRows (variable)																								
...																																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x0E.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**ColumnCount (2 bytes):** An unsigned integer that specifies the number of structures in the **RecipientColumns** field.

**RecipientColumns (variable):** An array of **PropertyTag** structures that specifies the property values that can be included for each recipient (1). The number of structures contained in this field is specified by the **ColumnCount** field. The format of the **PropertyTag** structure is specified in [\[MS-OXCDATA\]](#) section 2.9.

**RowCount (2 bytes):** An unsigned integer that specifies the number of rows in the **RecipientRows** field.

**RecipientRows (variable):** A list of **ModifyRecipientRow** structures. The number of structures contained in this field is specified by the **RowCount** field. The format of the **ModifyRecipientRow** structure is defined in section [2.2.6.5.1.1](#).

### 2.2.6.5.1.1 ModifyRecipientRow Structure

The following descriptions define valid fields for the **ModifyRecipientRow** structure.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RowId																															
RecipientType										RecipientRowSize															RecipientRow (variable)						
...																															

**RowId (4 bytes):** An unsigned integer that specifies the ID of the recipient (1).

**RecipientType (1 byte):** An enumeration that specifies the type of recipient (1). The possible values for this enumeration are specified in [\[MS-OXCMSG\]](#) section 2.2.3.1.2.

**RecipientRowSize (2 bytes):** An unsigned integer that specifies the size of the **RecipientRow** field.

**RecipientRow (variable):** A **RecipientRow** structure. This field is present when the **RecipientRowSize** field is nonzero and is not present otherwise. The format of the **RecipientRow** structure is specified in [\[MS-OXCDATA\]](#) section 2.8.3. The size of this field, in bytes, is specified by the **RecipientRowSize** field.

### 2.2.6.5.2 RopModifyRecipients ROP Response Buffer

The following descriptions define valid fields for the **RopModifyRecipients** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x0E.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.6.6 RopReadRecipients ROP

The **RopReadRecipients** ROP gets recipient (2) details from a message. For more details about this operation, see [\[MS-OXCMSG\]](#) section 2.2.3.6.

#### 2.2.6.6.1 RopReadRecipients ROP Request Buffer

The following descriptions define valid fields for the **RopReadRecipients** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31								
RopId										LogonId										InputHandleIndex										RowId									
...																														Reserved									
...																																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x0F.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).



**RowId (4 bytes):** An unsigned integer that specifies the starting index for the recipients (2) to be retrieved.

**Reserved (2 bytes):** Reserved. This field MUST be set to 0x0000. Server behavior is undefined if this field is not set to 0x0000.

### 2.2.6.6.2 RopReadRecipients ROP Success Response Buffer

The following descriptions define valid fields for the **RopReadRecipients** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
RopId										InputHandleIndex										ReturnValue													
...										RowCount										RecipientRows (variable)													
...																																	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x0F.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**RowCount (1 byte):** An unsigned integer that specifies the number of structures in the **RecipientRows** field.

**RecipientRows (variable):** A list of **ReadRecipientRow** structures. The number of structures contained in this field is specified by the **RowCount** field. The format of the **ReadRecipientRow** structure is defined in section [2.2.6.6.2.1](#).

#### 2.2.6.6.2.1 ReadRecipientRow Structure

The following descriptions define valid fields for the **ReadRecipientRow** structure.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
RowId																																	
RecipientType										CodePageId										Reserved													
...										RecipientRowSize										RecipientRow (variable)													
...																																	

**RowId (4 bytes):** An unsigned integer that specifies the row ID of the recipient (2).

**RecipientType (1 byte):** An enumeration that specifies the type of recipient (2). The possible values for this enumeration are specified in [\[MS-OXCMSG\]](#) section 2.2.3.1.2.

**CodePageId (2 bytes):** An identifier that specifies the code page for the recipient (2).

**Reserved (2 bytes):** Reserved. The server MUST set this field to 0x0000.

**RecipientRowSize (2 bytes):** An unsigned integer that specifies the size of the **RecipientRow** field.

**RecipientRow (variable):** A **RecipientRow** structure. The format of this structure is specified in [\[MS-OXCDATA\]](#) section 2.8.3. The size of this field, in bytes, is specified by the **RecipientRowSize** field.

### 2.2.6.6.3 RopReadRecipients ROP Failure Response Buffer

The following descriptions define valid fields for the **RopReadRecipients** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x0F.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.6.7 RopReloadCachedInformation ROP

The **RopReloadCachedInformation** ROP gets message and recipient (2) information from a message. For more details about this operation, see [\[MS-OXCMSG\]](#) section 2.2.3.7.

#### 2.2.6.7.1 RopReloadCachedInformation ROP Request Buffer

The following descriptions define valid fields for the **RopReloadCachedInformation** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								LogonId								InputHandleIndex								Reserved							
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x10.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**Reserved (2 bytes):** Reserved. This field MUST be set to 0x0000. Server behavior is undefined if this field is not set to 0x0000.

### 2.2.6.7.2 RopReloadCachedInformation ROP Success Response Buffer

The following descriptions define valid fields for the **RopReloadCachedInformation** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								InputHandleIndex								ReturnValue															
...																HasNamedProperties				SubjectPrefix (variable)											
...																															
NormalizedSubject (variable)																															
...																															
RecipientCount																ColumnCount															
RecipientColumns (variable)																															
...																															
RowCount								RecipientRows (variable)																							
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x10.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** specified field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**HasNamedProperties (1 byte):** A Boolean that specifies whether the message has named properties.

**SubjectPrefix (variable):** A **TypedString** structure that specifies the subject prefix of the message. The format of the **TypedString** structure is specified in [\[MS-OXCDATA\]](#) section 2.11.7.

**NormalizedSubject (variable):** A **TypedString** structure that specifies the normalized subject of the message.

**RecipientCount (2 bytes):** An unsigned integer that specifies the number of recipients (2) on the message.

**ColumnCount (2 bytes):** An unsigned integer that specifies the number of structures in the **RecipientColumns** field.

**RecipientColumns (variable):** An array of **PropertyTag** structures that specifies the property values that can be included for each recipient (2). The number of structures contained in this field is specified by the **ColumnCount** field. The format of the **PropertyTag** structure is specified in [MS-OXCDATA] section 2.9.

**RowCount (1 byte):** An unsigned integer that specifies the number of rows in the **RecipientRows** field.

**RecipientRows (variable):** A list of **OpenRecipientRow** structures. The number of structures contained in this field is specified by the **RowCount** field. The format of the **OpenRecipientRow** structure is defined in section [2.2.6.1.2.1](#).

### 2.2.6.7.3 RopReloadCachedInformation ROP Failure Response Buffer

The following descriptions define valid fields for the **RopReloadCachedInformation** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x10.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.6.8 RopSetMessageStatus ROP

The **RopSetMessageStatus** ROP sets the status of a message in a folder. For more details about this operation, see [\[MS-OXCMSG\]](#) section 2.2.3.8.

#### 2.2.6.8.1 RopSetMessageStatus ROP Request Buffer

The following descriptions define valid fields for the **RopSetMessageStatus** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31								
RopId										LogonId										InputHandleIndex										MessageId									
...																																							

...	MessageStatusFlags
...	MessageStatusMask
...	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x20.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**MessageId (8 bytes):** An identifier that specifies the message for which the status will be changed.

**MessageStatusFlags (4 bytes):** A flags structure that contains status flags to set on the message. The possible values are specified in [\[MS-OXCMSG\]](#) section 2.2.3.8.1.

**MessageStatusMask (4 bytes):** A bitmask that specifies which bits in the **MessageStatusFlags** field are to be changed.

#### 2.2.6.8.2 RopSetMessageStatus ROP Success Response Buffer

The following descriptions define valid fields for the **RopSetMessageStatus** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...										...										MessageStatusFlags											
...										...										...											

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x20.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**MessageStatusFlags (4 bytes):** A flags structure that contains the status flags that were set on the message before this operation. The possible values are specified in [\[MS-OXCMSG\]](#) section 2.2.3.8.2.

#### 2.2.6.8.3 RopSetMessageStatus ROP Failure Response Buffer

The following descriptions define valid fields for the **RopSetMessageStatus** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
RopId										InputHandleIndex										ReturnValue													
...																																	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x20.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.6.9 RopGetMessageStatus ROP

The **RopGetMessageStatus** ROP returns the status of a message in a folder. For more details about this operation, see [\[MS-OXCMSG\]](#) section 2.2.3.9.

#### 2.2.6.9.1 RopGetMessageStatus ROP Request Buffer

The following descriptions define valid fields for the **RopGetMessageStatus** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
RopId										LogonId										InputHandleIndex										MessageId									
...																																							
...																																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x1F.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**MessageId (8 bytes):** An identifier that specifies the message for which the status will be returned.

#### 2.2.6.9.2 RopGetMessageStatus ROP Response Buffers

The response buffers for this ROP are the same as those for the **RopSetMessageStatus** ROP, as specified in sections [2.2.6.8.2](#) and [2.2.6.8.3](#). The value of the **RopId** field for **RopGetMessageStatus** responses MUST be 0x20, which is the same as that for the **RopSetMessageStatus** responses.

### 2.2.6.10 RopSetReadFlags ROP

The **RopSetReadFlags** ROP sets the read flag for messages in a folder. For more details about this operation, see [\[MS-OXCMSG\]](#) section 2.2.3.10.

#### 2.2.6.10.1 RopSetReadFlags ROP Request Buffer

The following descriptions define valid fields for the **RopSetReadFlags** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								LogonId								InputHandleIndex								WantAsynchronous							
ReadFlags								MessageIdCount																MessageIds (variable)							
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x66.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**WantAsynchronous (1 byte):** A Boolean that specifies whether the operation is to be processed asynchronously with status reported via the **RopProgress** ROP (section [2.2.8.13](#)).

**ReadFlags (1 byte):** A flags structure that contains flags that specify the flags to set. The possible values for these flags are specified in [\[MS-OXCMSG\]](#) section 2.2.3.10.1.

**MessageIdCount (2 bytes):** An unsigned integer that specifies the number of identifiers in the **MessageIds** field.

**MessageIds (variable):** An array of 64-bit identifiers that specify the messages that are to have their read flags changed. The number of identifiers contained in this field is specified by the **MessageIdCount** field.

#### 2.2.6.10.2 RopSetReadFlags ROP Response Buffer

The following descriptions define valid fields for the **RopSetReadFlags** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								InputHandleIndex								ReturnValue															
...																PartialCompletion															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x66.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

**PartialCompletion (1 byte):** A Boolean that indicates whether the operation was only partially completed. The operation is partially completed if the server was unable to modify one or more of the Message objects that are specified in the **MessageIds** field of the request.

### 2.2.6.11 RopSetMessageReadFlag ROP

The **RopSetMessageReadFlag** ROP sets or clears the message read flag. For more details about this operation, see [\[MS-OXCMSG\]](#) section 2.2.3.11.

#### 2.2.6.11.1 RopSetMessageReadFlag ROP Request Buffer

The following descriptions define valid fields for the **RopSetMessageReadFlag** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId				LogonId								ResponseHandleIndex								InputHandleIndex											
ReadFlags				ClientData (optional)																											
...																															
...																															
...																															
...																															
...																															
...																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x11.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**ResponseHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table that is referenced in the response. For more information about Server objects, see section [1.3.1](#).

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored.

**ReadFlags (1 byte):** A flags structure. The possible values for these flags are specified in [\[MS-OXCMSG\]](#) section 2.2.3.11.1.

**ClientData (24 bytes):** An array of bytes that is present when the logon associated with **LogonId** was created with the **Private** flag, as specified in [\[MS-OXCSTOR\]](#) section 2.2.1.1.1, unset and is



not present otherwise. This value specifies the information that is returned to the client in a successful response.

### 2.2.6.11.2 RopSetMessageReadFlag ROP Success Response Buffer

The following descriptions define valid fields for the **RopSetMessageReadFlag** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										ResponseHandleIndex										ReturnValue											
...															ReadStatusChanged						LogonId (optional)										
ClientData (optional)																															
...																															
...																															
...																															
...																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x11.

**ResponseHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **ResponseHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**ReadStatusChanged (1 byte):** A Boolean that specifies whether the read status of a public folder's message has changed.

**LogonId (1 byte):** An unsigned integer index that is present when the value in the **ReadStatusChanged** field is nonzero and is not present otherwise. This field MUST be set to the value of the **LogonId** field in the request.

**ClientData (24 bytes):** An array of bytes that is present when the value in the **ReadStatusChanged** field is nonzero and is not present otherwise. This field MUST be set to the value of the **ClientData** field in the request.

### 2.2.6.11.3 RopSetMessageReadFlag ROP Failure Response Buffer

The following descriptions define valid fields for the **RopSetMessageReadFlag** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
RopId										ResponseHandleIndex										ReturnValue													
...																																	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x11.

**ResponseHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **ResponseHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

## 2.2.6.12 RopOpenAttachment ROP

The **RopOpenAttachment** ROP opens an attachment to a message. For more details about this operation, see [\[MS-OXCMSG\]](#) section 2.2.3.12.

### 2.2.6.12.1 RopOpenAttachment ROP Request Buffer

The following descriptions define valid fields for the **RopOpenAttachment** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
RopId										LogonId										InputHandleIndex										OutputHandleIndex									
OpenAttachmentFlags										AttachmentID																													
...																																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x22.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**OutputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the output Server object will be stored.

**OpenAttachmentFlags (1 byte):** A flags structure that contains flags for opening attachments. The possible values for these flags are specified in [\[MS-OXCMSG\]](#) section 2.2.3.12.1.

**AttachmentID (4 bytes):** An unsigned integer index that identifies the attachment to be opened. The value of this field is equivalent to the value of the **PidTagAttachNumber** property ([\[MS-OXCMSG\]](#) section 2.2.2.6).

### 2.2.6.12.2 RopOpenAttachment ROP Response Buffer

The following descriptions define valid fields for the **RopOpenAttachment** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										OutputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x22.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.6.13 RopCreateAttachment ROP

The **RopCreateAttachment** ROP creates a new attachment on a message. For more details about this operation, see [\[MS-OXCMSG\]](#) section 2.2.3.13.

#### 2.2.6.13.1 RopCreateAttachment ROP Request Buffer

The following descriptions define valid fields for the **RopCreateAttachment** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31										
RopId										LogonId										InputHandleIndex										OutputHandleIndex											

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x23.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**OutputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the output Server object will be stored.

#### 2.2.6.13.2 RopCreateAttachment ROP Success Response Buffer

The following descriptions define valid fields for the **RopCreateAttachment** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										OutputHandleIndex										ReturnValue											
...																AttachmentID															

...

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x23.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**AttachmentID (4 bytes):** An unsigned integer identifier that refers to the attachment created. The value of this field is equivalent to the value of the **PidTagAttachNumber** property ([\[MS-OXCMSG\]](#) section 2.2.2.6).

### 2.2.6.13.3 RopCreateAttachment ROP Failure Response Buffer

The following descriptions define valid fields for the **RopCreateAttachment** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										OutputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x23.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.6.14 RopDeleteAttachment ROP

The **RopDeleteAttachment** ROP deletes an attachment on a message. For more details about this operation, see [\[MS-OXCMSG\]](#) section 2.2.3.14.

#### 2.2.6.14.1 RopDeleteAttachment ROP Request Buffer

The following descriptions define valid fields for the **RopDeleteAttachment** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31								
RopId										LogonId										InputHandleIndex										AttachmentID									
...																																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x24.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**AttachmentID (4 bytes):** An unsigned integer that identifies the attachment to be deleted. The value of this field is equivalent to the **PidTagAttachNumber** property ([\[MS-OXCMSG\]](#) section 2.2.2.6).

### 2.2.6.14.2 RopDeleteAttachment ROP Response Buffer

The following descriptions define valid fields for the **RopDeleteAttachment** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x24.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.6.15 RopSaveChangesAttachment ROP

The **RopSaveChangesAttachment** ROP commits the changes made to an attachment. For more details about this operation, see [\[MS-OXCMSG\]](#) section 2.2.3.15.

#### 2.2.6.15.1 RopSaveChangesAttachment ROP Request Buffer

The following descriptions define valid fields for the **RopSaveChangesAttachment** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								LogonId								ResponseHandleIndex								InputHandleIndex							
SaveFlags																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x25.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**ResponseHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table that is referenced in the response. For more information about Server objects, see section [1.3.1](#).

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored.

**SaveFlags (1 byte):** A flags structure that contains flags that specify how the save operation behaves. The possible values for these flags are specified in [\[MS-OXCMSG\]](#) section 2.2.3.3.1.

### 2.2.6.15.2 RopSaveChangesAttachment ROP Response Buffer

The following descriptions define valid fields for the **RopSaveChangesAttachment** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
RopId										ResponseHandleIndex										ReturnValue													
...																																	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x25.

**ResponseHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **ResponseHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.6.16 RopOpenEmbeddedMessage ROP

The **RopOpenEmbeddedMessage** ROP opens an attachment as a message. For more details about this operation, see [\[MS-OXCMSG\]](#) section 2.2.3.16.

#### 2.2.6.16.1 RopOpenEmbeddedMessage ROP Request Buffer

The following descriptions define valid fields for the **RopOpenEmbeddedMessage** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										LogonId										InputHandleIndex						OutputHandleIndex					
CodePageId																OpenModeFlags															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x46.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**OutputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the output Server object will be stored.

**CodePageId (2 bytes):** An identifier that specifies which code page is used for string values associated with the message.

**OpenModeFlags (1 byte):** A flags structure that contains flags that control the access to the message. The possible values are specified in [\[MS-OXCMSG\]](#) section 2.2.3.16.1.

### 2.2.6.16.2 RopOpenEmbeddedMessage ROP Success Response Buffer

The following descriptions define valid fields for the **RopOpenEmbeddedMessage** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										OutputHandleIndex										ReturnValue											
...										Reserved										MessageId											
...																															
...																				HasNamedProperties											
SubjectPrefix (variable)																															
...																															
NormalizedSubject (variable)																															
...																															
RecipientCount																ColumnCount															
RecipientColumns (variable)																															
...																															
RowCount								RecipientRows (variable)																							
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x46.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**Reserved (1 byte):** Reserved. This field MUST be set to 0x00.

**MessageId (8 bytes):** An identifier that specifies the ID of the **Embedded Message object**.

**HasNamedProperties (1 byte):** A Boolean that specifies whether the message has named properties.

**SubjectPrefix (variable):** A **TypedString** structure that specifies the subject prefix of the message. The format of the **TypedString** structure is specified in [\[MS-OXCDATA\]](#) section 2.11.7.

**NormalizedSubject (variable):** A **TypedString** structure that specifies the normalized subject of the message.

**RecipientCount (2 bytes):** An unsigned integer that specifies the number of recipients (2) on the message.

**ColumnCount (2 bytes):** An unsigned integer that specifies the number of structures in the **RecipientColumns** field.

**RecipientColumns (variable):** An array of **PropertyTag** structures. The number of structures contained in this field is specified by the **ColumnCount** field. The format of the **PropertyTag** structure is specified in [\[MS-OXCDATA\]](#) section 2.9. This field specifies the property values that can be included for each recipient (2).

**RowCount (1 byte):** An unsigned integer that specifies the number of rows in the **RecipientRows** field.

**RecipientRows (variable):** A list of **OpenRecipientRow** structures. The number of structures contained in this field is specified by the **RowCount** field. The format of the **OpenRecipientRow** structure is defined in section [2.2.6.1.2.1](#).

### 2.2.6.16.3 RopOpenEmbeddedMessage ROP Failure Response Buffer

The following descriptions define valid fields for the **RopOpenEmbeddedMessage** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId								OutputHandleIndex								ReturnValue															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x46.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.6.17 RopGetAttachmentTable ROP

The **RopGetAttachmentTable** ROP gets the attachment table of a message. For more details about this operation, see [\[MS-OXCMSG\]](#) section 2.2.3.17.



### 2.2.6.17.1 RopGetAttachmentTable ROP Request Buffer

The following descriptions define valid fields for the **RopGetAttachmentTable** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								LogonId								InputHandleIndex								OutputHandleIndex							
TableFlags																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x21.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**OutputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the output Server object will be stored.

**TableFlags (1 byte):** A flags structure that contains flags that control the type of table. The possible values are specified in [\[MS-OXCMSG\]](#) section 2.2.3.17.1.

### 2.2.6.17.2 RopGetAttachmentTable ROP Response Buffer

The following descriptions define valid fields for the **RopGetAttachmentTable** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								OutputHandleIndex								ReturnValue															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x21.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.6.18 RopGetValidAttachments ROP

The **RopGetValidAttachments** ROP gets the valid attachment identifiers of a message. For more details about this operation, see [\[MS-OXCMSG\]](#) section 2.2.3.18.

#### 2.2.6.18.1 RopGetValidAttachments ROP Request Buffer

The following descriptions define valid fields for the **RopGetValidAttachments** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId									LogonId									InputHandleIndex													

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x52.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

### 2.2.6.18.2 RopGetValidAttachments ROP Success Response Buffer

The following descriptions define valid fields for the **RopGetValidAttachments** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId									InputHandleIndex									ReturnValue													
...									AttachmentIdCount																						
AttachmentIdArray (variable)																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x52.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**AttachmentIdCount (2 bytes):** An unsigned integer that specifies the number of integers in the **AttachmentIdArray** field.

**AttachmentIdArray (variable):** An array of 32-bit integers that represent the valid attachment identifiers of the message. The number of integer values contained in this field is specified by the **AttachmentIdCount** field.

### 2.2.6.18.3 RopGetValidAttachments ROP Failure Response Buffer

The following descriptions define valid fields for the **RopGetValidAttachments** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId								InputHandleIndex								ReturnValue															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x52.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

## 2.2.7 Transport ROPs

### 2.2.7.1 RopSubmitMessage ROP

The **RopSubmitMessage** ROP submits a message for sending. For more details about this operation, see [\[MS-OXOMSG\]](#) section 2.2.4.1.

#### 2.2.7.1.1 RopSubmitMessage ROP Request Buffer

The following descriptions define valid fields for the **RopSubmitMessage** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId								LogonId								InputHandleIndex								SubmitFlags							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x32.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**SubmitFlags (1 byte):** A flags structure that contains flags that specify special behavior for submitting the message. The possible values are specified in [\[MS-OXOMSG\]](#) section 2.2.4.1.1.

#### 2.2.7.1.2 RopSubmitMessage ROP Response Buffer

The following descriptions define valid fields for the **RopSubmitMessage** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId								InputHandleIndex								ReturnValue															

...

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x32.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.7.2 RopAbortSubmit ROP

The **RopAbortSubmit** ROP aborts a previous message submission. For more details about this operation, see [\[MS-OXOMSG\]](#) section 2.2.4.2.

#### 2.2.7.2.1 RopAbortSubmit ROP Request Buffer

The following descriptions define valid fields for the **RopAbortSubmit** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31								
RopId										LogonId										InputHandleIndex										FolderId									
...																																							
...																								MessageId															
...																																							
...																																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x34.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**FolderId (8 bytes):** An identifier that identifies the folder in which the submitted message is located.

**MessageId (8 bytes):** An identifier that specifies the submitted message.

#### 2.2.7.2.2 RopAbortSubmit ROP Response Buffer

The following descriptions define valid fields for the **RopAbortSubmit** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
RopId										InputHandleIndex										ReturnValue													
...																																	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x34.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.7.3 RopGetAddressTypes ROP

The **RopGetAddressTypes** ROP returns the type of recipient (2) handled by a transport provider. For more details about this operation, see [\[MS-OXOMSG\]](#) section 2.2.4.3.

#### 2.2.7.3.1 RopGetAddressTypes ROP Request Buffer

The following descriptions define valid fields for the **RopGetAddressTypes** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										LogonId										InputHandleIndex											

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x49.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

#### 2.2.7.3.2 RopGetAddressTypes ROP Success Response Buffer

The following descriptions define valid fields for the **RopGetAddressTypes** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
RopId										InputHandleIndex										ReturnValue													
...										AddressTypeCount																							
AddressTypeSize										AddressTypes (variable)																							
...																																	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x49.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**AddressTypeCount (2 bytes):** An unsigned integer that specifies the number of strings in the **AddressTypes** field.

**AddressTypeSize (2 bytes):** An unsigned integer that specifies the length of the **AddressTypes** field.

**AddressTypes (variable):** A list of null-terminated ASCII strings. The number of strings contained in this field is specified by the **AddressTypeCount** field. The size of this field, in bytes, is specified by the **AddressTypeSize** field.

### 2.2.7.3.3 RopGetAddressTypes ROP Failure Response Buffer

The following descriptions define valid fields for the **RopGetAddressTypes** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x49.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.7.4 RopSetSpooler ROP

The **RopSetSpooler** ROP informs the server that the client intends to act as a mail spooler. For more details about this operation, see [\[MS-OXOMSG\]](#) section 2.2.5.1.

#### 2.2.7.4.1 RopSetSpooler ROP Request Buffer

The following descriptions define valid fields for the **RopSetSpooler** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										LogonId										InputHandleIndex											

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x47.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

#### 2.2.7.4.2 RopSetSpooler ROP Response Buffer

The following descriptions define valid fields for the **RopSetSpooler** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								InputHandleIndex								ReturnValue															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x47.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

#### 2.2.7.5 RopSpoolerLockMessage ROP

The **RopSpoolerLockMessage** ROP locks the specified message for spooling. For more details about this operation, see [\[MS-OXOMSG\]](#) section 2.2.5.3.

##### 2.2.7.5.1 RopSpoolerLockMessage ROP Request Buffer

The following descriptions define valid fields for the **RopSpoolerLockMessage** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								LogonId								InputHandleIndex								MessageId							
...																															
...																								LockState							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x48.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**MessageId (8 bytes):** An identifier that specifies the message for which the status will be changed.

**LockState (1 byte):** An enumeration. The possible values for this enumeration are specified in [\[MS-OXOMSG\]](#) section 2.2.5.3.1.

### 2.2.7.5.2 RopSpoolerLockMessage ROP Response Buffer

The following descriptions define valid fields for the **RopSpoolerLockMessage** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x48.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.7.6 RopTransportSend ROP

The **RopTransportSend** ROP sends the specified Message object out for message delivery. For more details about this operation, see [\[MS-OXOMSG\]](#) section 2.2.5.4.

#### 2.2.7.6.1 RopTransportSend ROP Request Buffer

The following descriptions define valid fields for the **RopTransportSend** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										LogonId										InputHandleIndex											

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x4A.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

#### 2.2.7.6.2 RopTransportSend ROP Success Response Buffer

The following descriptions define valid fields for the **RopTransportSend** ROP success response buffer.



0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId								InputHandleIndex								ReturnValue															
...																NoPropertiesReturned				PropertyValueCount (optional)											
...								PropertyValues (variable)																							
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x4A.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**NoPropertiesReturned (1 byte):** A Boolean that specifies whether property values are returned.

**PropertyValueCount (2 bytes):** An unsigned integer that specifies the number of structures returned in the **PropertyValues** field.

**PropertyValues (variable):** An array of **TaggedPropertyValue** structures that specifies the properties to copy. The format of the **TaggedPropertyValue** structure is specified in [\[MS-OXCDATA\]](#) section 2.11.4. The number of structures contained in this field is specified by the **PropertyValueCount** field.

### 2.2.7.6.3 RopTransportSend ROP Failure Response Buffer

The following descriptions define valid fields for the **RopTransportSend** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId								InputHandleIndex								ReturnValue															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x4A.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.7.7 RopTransportNewMail ROP

The **RopTransportNewMail** ROP informs the server of new mail. For more details about this operation, see [\[MS-OXOMSG\]](#) section 2.2.5.5.

#### 2.2.7.7.1 RopTransportNewMail ROP Request Buffer

The following descriptions define valid fields for the **RopTransportNewMail** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								LogonId								InputHandleIndex								MessageId							
...																															
...																								FolderId							
...																															
...																								MessageClass (variable)							
...																															
MessageFlags																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x51.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**MessageId (8 bytes):** An identifier that specifies the new Message object.

**FolderId (8 bytes):** An identifier that specifies the folder of the new Message object.

**MessageClass (variable):** A null-terminated ASCII string that specifies the message class of the new Message object.

**MessageFlags (4 bytes):** A flags structure that contains the message flags of the new Message object. The possible values are specified in [\[MS-OXOMSG\]](#) section 2.2.5.5.1.

#### 2.2.7.7.2 RopTransportNewMail ROP Response Buffer

The following descriptions define valid fields for the **RopTransportNewMail** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								InputHandleIndex								ReturnValue															

...

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x51.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.7.8 RopGetTransportFolder ROP

The **RopGetTransportFolder** ROP retrieves the Folder ID structure, as specified in [\[MS-OXCDATA\]](#) section 2.2.1.1, of the temporary transport folder. For more details about this operation, see [\[MS-OXOMSG\]](#) section 2.2.5.2.

#### 2.2.7.8.1 RopGetTransportFolder ROP Request Buffer

The following descriptions define valid fields for the **RopGetTransportFolder** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										LogonId										InputHandleIndex											

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x6D.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

#### 2.2.7.8.2 RopGetTransportFolder ROP Success Response Buffer

The following descriptions define valid fields for the **RopGetTransportFolder** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...										FolderId																					
...																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x6D.

**InputHandleIndex (1 byte):** An unsigned integer that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

**FolderId (8 bytes):** An identifier that specifies the transport folder.

### 2.2.7.8.3 RopGetTransportFolder ROP Failure Response Buffer

The following descriptions define valid fields for the **RopGetTransportFolder** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								InputHandleIndex								ReturnValue															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x6D.

**InputHandleIndex (1 byte):** An unsigned integer that MUST be set to the **InputHandleIndex** specified in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.7.9 RopOptionsData ROP

The **RopOptionsData** ROP retrieves the options data that is associated with an **address type**.[<6>](#) For more details about this operation, see [\[MS-OXOMSG\]](#) section 2.2.4.4.

#### 2.2.7.9.1 RopOptionsData ROP Request Buffer

The following descriptions define valid fields for the **RopOptionsData** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								LogonId								InputHandleIndex								AddressType (variable)							
...																															
WantWin32																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x6F.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**AddressType (variable):** A null-terminated ASCII string that specifies the address type that options are to be returned for.

**WantWin32 (1 byte):** A Boolean that specifies whether the help file data is to be returned in a format that is suited for 32-bit machines.

### 2.2.7.9.2 RopOptionsData ROP Success Response Buffer

The following descriptions define valid fields for the **RopOptionsData** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...										Reserved							OptionsInfoSize														
...										OptionsInfo (variable)																					
...																															
HelpFileSize																HelpFile (variable)															
...																															
HelpFileName (variable)																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x6F.

**InputHandleIndex (1 byte):** An unsigned integer that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**Reserved (1 byte):** Reserved. This field MUST be set to 0x01.

**OptionsInfoSize (2 bytes):** An unsigned integer that specifies the size of the **OptionsInfo** field.

**OptionsInfo (variable):** An array of bytes that contains opaque data from the server. Clients SHOULD ignore this field. Servers SHOULD return this field as an empty array. <7> The size of this field, in bytes, is specified by the **OptionsInfoSize** field.

**HelpFileSize (2 bytes):** An unsigned integer that specifies the size of the **HelpFile** field.

**HelpFile (variable):** An array of bytes that contains the help file associated with the specified address type. The size of this field, in bytes, is specified by the **HelpFileSize** field. This field is present only when the value of the **HelpFileSize** field is nonzero.

**HelpFileName (variable):** A null-terminated multibyte string that specifies the name of the help file that is associated with the specified address type. The string is composed of multibyte characters. This field is present if **HelpFileSize** is nonzero and is not present otherwise.

### 2.2.7.9.3 RopOptionsData ROP Failure Response Buffer

The following descriptions define valid fields for the **RopOptionsData** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
RopId										InputHandleIndex										ReturnValue													
...																																	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x6F.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field SHOULD be set to a value other than 0x00000000. <8>

## 2.2.8 Property ROPs

### 2.2.8.1 RopGetPropertyIdsFromNames ROP

The **RopGetPropertyIdsFromNames** ROP gets **property IDs** for specified **property names**. For more details about this operation, see [\[MS-OXCPRPT\]](#) section 2.2.12.

#### 2.2.8.1.1 RopGetPropertyIdsFromNames ROP Request Buffer

The following descriptions define valid fields for the **RopGetPropertyIdsFromNames** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1				
RopId										LogonId										InputHandleIndex								Flags							
PropertyNameCount																PropertyNames (variable)																			
...																																			

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x56.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**Flags (1 byte):** A flags structure that contains flags that control the behavior of this operation. The possible values are specified in [\[MS-OXCPRPT\]](#) section 2.2.12.1.

**PropertyNameCount (2 bytes):** An unsigned integer that specifies the number of structures in the **PropertyNames** field.

**PropertyNames (variable):** A list of **PropertyName** structures that specifies the property names requested. The number of structures contained in this field is specified by the **PropertyNameCount** field. The format of the **PropertyName** structure is specified in [\[MS-OXCPRPT\]](#) section 2.2.12.1.

### 2.2.8.1.2 RopGetPropertyIdsFromNames ROP Success Response Buffer

The following descriptions define valid fields for the **RopGetPropertyIdsFromNames** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										InputHandleIndex										ReturnValue											
...																				PropertyIdCount											
PropertyIds (variable)																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x56.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**PropertyIdCount (2 bytes):** An unsigned integer that specifies the number of integers contained in the **PropertyIds** field.

**PropertyIds (variable):** An array of unsigned 16-bit integers. Each integer in the array is the property ID associated with a property name. The number of integers in the array is specified by the **PropertyIdCount** field.

### 2.2.8.1.3 RopGetPropertyIdsFromNames ROP Failure Response Buffer

The following descriptions define valid fields for the **RopGetPropertyIdsFromNames** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x56.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.8.2 RopGetNamesFromPropertyIds ROP

The **RopGetNamesFromPropertyIds** ROP gets property names for specified property IDs. For more details about this operation, see [\[MS-OXCPRPT\]](#) section 2.2.13.

#### 2.2.8.2.1 RopGetNamesFromPropertyIds ROP Request Buffer

The following descriptions define valid fields for the **RopGetNamesFromPropertyIds** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31				
RopId										LogonId						InputHandleIndex						PropertyIdCount													
...										PropertyIds (variable)																									
...																																			

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x55.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**PropertyIdCount (2 bytes):** An unsigned integer that specifies the number of integers contained in the **PropertyIds** field.

**PropertyIds (variable):** An array of unsigned 16-bit integers. Each integer in the array is the property ID associated with a property name to be returned in the response. The number of integers in the array is specified by the **PropertyIdCount** field.

#### 2.2.8.2.2 RopGetNamesFromPropertyIds ROP Success Response Buffer

The following descriptions define valid fields for the **RopGetNamesFromPropertyIds** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31		
RopId										InputHandleIndex						ReturnValue																	



...	PropertyNameCount
PropertyNames (variable)	
...	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x55.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**PropertyNameCount (2 bytes):** An unsigned integer that specifies the number of structures in the **PropertyNames** field.

**PropertyNames (variable):** A list of **PropertyName** structures that specifies the property names for the property IDs requested. The number of structures contained in this field is specified by the **PropertyNameCount** field. The format of the **PropertyName** structure is specified in [\[MS-OXCDATA\]](#) section 2.6.1.

### 2.2.8.2.3 RopGetNamesFromPropertyIds ROP Failure Response Buffer

The following descriptions define valid fields for the **RopGetNamesFromPropertyIds** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
RopId										InputHandleIndex										ReturnValue													
...																																	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x55.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.8.3 RopGetPropertiesSpecific ROP

The **RopGetPropertiesSpecific** ROP gets property values for specified **property tags**. For more details about this operation, see [\[MS-OXCPRPT\]](#) section 2.2.2.

#### 2.2.8.3.1 RopGetPropertiesSpecific ROP Request Buffer

The following descriptions define valid fields for the **RopGetPropertiesSpecific** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1				
RopId									LogonId									InputHandleIndex									PropertySizeLimit								
...									WantUnicode									PropertyTagCount																	
...									PropertyTags (variable)																										
...																																			

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x07.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**PropertySizeLimit (2 bytes):** An unsigned integer that specifies the maximum size allowed for a property value returned.

**WantUnicode (2 bytes):** A Boolean that specifies whether to return string properties in multibyte Unicode.

**PropertyTagCount (2 bytes):** An unsigned integer that specifies the number of tags present in the **PropertyTags** field.

**PropertyTags (variable):** An array of **PropertyTag** structures that specifies the properties requested. The number of structures contained in this field is specified by the **PropertyTagCount** field. The format of the **PropertyTag** structure is specified in [\[MS-OXCDATA\]](#) section 2.9.

### 2.2.8.3.2 RopGetPropertiesSpecific ROP Success Response Buffer

The following descriptions define valid fields for the **RopGetPropertiesSpecific** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1				
RopId									InputHandleIndex									ReturnValue																	
...									RowData (variable)																										
...																																			

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x07.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**RowData (variable):** A **PropertyRow** structure. The format of the **PropertyRow** structure is specified in [\[MS-OXCDATA\]](#) section 2.8. The columns used for these rows were those specified in the **PropertyTags** field in the ROP request.

### 2.2.8.3.3 RopGetPropertiesSpecific ROP Failure Response Buffer

The following descriptions define valid fields for the **RopGetPropertiesSpecific** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								InputHandleIndex								ReturnValue															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x07.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.8.4 RopGetPropertiesAll ROP

The **RopGetPropertiesAll** ROP gets all the property values for an object. For more details about this operation, see [\[MS-OXCPRPT\]](#) section 2.2.3.

#### 2.2.8.4.1 RopGetPropertiesAll ROP Request Buffer

The following descriptions define valid fields for the **RopGetPropertiesAll** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								LogonId								InputHandleIndex								PropertySizeLimit							
...								WantUnicode																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x08.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**PropertySizeLimit (2 bytes):** An unsigned integer that specifies the maximum size allowed for a property value returned.

**WantUnicode (2 bytes):** A Boolean that specifies whether to return string properties in multibyte Unicode.

#### 2.2.8.4.2 RopGetPropertiesAll ROP Success Response Buffer

The following descriptions define valid fields for the **RopGetPropertiesAll** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								InputHandleIndex								ReturnValue															
...																PropertyValueCount															
PropertyValues (variable)																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x08.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**PropertyValueCount (2 bytes):** An unsigned integer that specifies the number of structures present in the **PropertyValues** field.

**PropertyValues (variable):** An array of **TaggedPropertyValue** structures that are the properties defined on the object. The number of structures contained in this field is specified by the **PropertyValueCount** field. The format of the **TaggedPropertyValue** structure is specified in [\[MS-OXCDATA\]](#) section 2.11.4.

#### 2.2.8.4.3 RopGetPropertiesAll ROP Failure Response Buffer

The following descriptions define valid fields for the **RopGetPropertiesAll** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								InputHandleIndex								ReturnValue															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x08.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.8.5 RopGetPropertiesList ROP

The **RopGetPropertiesList** ROP gets the list of property tags for an object. For more details about this operation, see [\[MS-OXCPRPT\]](#) section 2.2.4.

#### 2.2.8.5.1 RopGetPropertiesList ROP Request Buffer

The following descriptions define valid fields for the **RopGetPropertiesList** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId									LogonId									InputHandleIndex													

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x09.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

#### 2.2.8.5.2 RopGetPropertiesList ROP Success Response Buffer

The following descriptions define valid fields for the **RopGetPropertiesList** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId									InputHandleIndex									ReturnValue													
...																		PropertyTagCount													
PropertyTags (variable)																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x09.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**PropertyTagCount (2 bytes):** An unsigned integer that specifies the number of property tags in the **PropertyTags** field.

**PropertyTags (variable):** An array of **PropertyTag** structures that lists the property tags on the object. The number of structures contained in this field is specified by the **PropertyTagCount** field. The format of the **PropertyTag** structure is specified in [\[MS-OXCDATA\]](#) section 2.9.

### 2.2.8.5.3 RopGetPropertiesList ROP Failure Response Buffer

The following descriptions define valid fields for the **RopGetPropertiesList** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x09.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.8.6 RopSetProperties ROP

The **RopSetProperties** ROP sets property values for an object. For more details about this operation, see [\[MS-OXCPRPT\]](#) section 2.2.5.

#### 2.2.8.6.1 RopSetProperties ROP Request Buffer

The following descriptions define valid fields for the **RopSetProperties** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										LogonId										InputHandleIndex								PropertyValueSize			
...										PropertyValueCount										PropertyValues (variable)											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x0A.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**PropertyValueSize (2 bytes):** An unsigned integer that specifies the number of bytes used for the **PropertyValueCount** field and the **PropertyValues** field.

**PropertyValueCount (2 bytes):** An unsigned integer that specifies the number of **PropertyValue** structures listed in the **PropertyValues** field.

**PropertyValues (variable):** An array of **TaggedPropertyValue** structures that specifies the property values to be set on the object. The number of structures contained in this field is specified by the **PropertyValueCount** field. The size of this field, in bytes, is equal to the value of the **PropertyValueSize** field - 2. The format of the **TaggedPropertyValue** structure is specified in [\[MS-OXCDATA\]](#) section 2.11.4.

### 2.2.8.6.2 RopSetProperties ROP Success Response Buffer

The following descriptions define valid fields for the **RopSetProperties** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId								InputHandleIndex								ReturnValue															
...																PropertyProblemCount															
PropertyProblems (variable)																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x0A.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**PropertyProblemCount (2 bytes):** An unsigned integer that specifies the number of **PropertyProblem** structures in the **PropertyProblems** field.

**PropertyProblems (variable):** An array of **PropertyProblem** structures. The number of structures contained in this field is specified by the **PropertyProblemCount** field. The format of the **PropertyProblem** structure is specified in [\[MS-OXCDATA\]](#) section 2.7.

### 2.2.8.6.3 RopSetProperties ROP Failure Response Buffer

The following descriptions define valid fields for the **RopSetProperties** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId								InputHandleIndex								ReturnValue															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x0A.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.8.7 RopSetPropertiesNoReplicate ROP

The **RopSetPropertiesNoReplicate** ROP sets property values for an object without invoking replication. For more details about this operation, see [\[MS-OXCPRPT\]](#) section 2.2.6.

#### 2.2.8.7.1 RopSetPropertiesNoReplicate ROP Request Buffer

The following descriptions define valid fields for the **RopSetPropertiesNoReplicate** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										LogonId						InputHandleIndex						PropertyValueSize									
...										PropertyValueCount						PropertyValues (variable)															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x79.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**PropertyValueSize (2 bytes):** An unsigned integer that specifies the number of bytes used for the **PropertyValueCount** field and the **PropertyValues** field.

**PropertyValueCount (2 bytes):** An unsigned integer that specifies the number of structures listed in the **PropertyValues** field.

**PropertyValues (variable):** An array of **TaggedPropertyValue** structures that specifies the property values to be set on the object. The number of structures contained in this field is specified by the **PropertyValueCount** field. The size of this field, in bytes, is equal to the value of the **PropertyValueSize** field - 2. The format of the **TaggedPropertyValue** structure is specified in [\[MS-OXCDATA\]](#) section 2.11.4.

#### 2.2.8.7.2 RopSetPropertiesNoReplicate ROP Success Response Buffer

The following descriptions define valid fields for the **RopSetPropertiesNoReplicate** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex						ReturnValue															



...	PropertyProblemCount
PropertyProblems (variable)	
...	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x79.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**PropertyProblemCount (2 bytes):** An unsigned integer that specifies the number of **PropertyProblem** structures in the **PropertyProblems** field.

**PropertyProblems (variable):** An array of **PropertyProblem** structures. The number of structures contained in this field is specified by the **PropertyProblemCount** field. The format of the **PropertyProblem** structure is specified in [\[MS-OXCDATA\]](#) section 2.7.

### 2.2.8.7.3 RopSetPropertiesNoReplicate ROP Failure Response Buffer

The following descriptions define valid fields for the **RopSetPropertiesNoReplicate** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
RopId										InputHandleIndex										ReturnValue													
...																																	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x79.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.8.8 RopDeleteProperties ROP

The **RopDeleteProperties** ROP deletes property values for an object. For more details about this operation, see [\[MS-OXCPRPT\]](#) section 2.2.7.

#### 2.2.8.8.1 RopDeleteProperties ROP Request Buffer

The following descriptions define valid fields for the **RopDeleteProperties** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1				
RopId										LogonId										InputHandleIndex						PropertyTagCount									
...										PropertyTags (variable)																									
...																																			

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x0B.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**PropertyTagCount (2 bytes):** An unsigned integer that specifies the number of **PropertyTag** structures in the **PropertyTags** field.

**PropertyTags (variable):** An array of **PropertyTag** structures that specifies the property values to be deleted from the object. The number of structures contained in this field is specified by the **PropertyTagCount** field. The format of the **PropertyTag** structure is specified in [\[MS-OXCDATA\]](#) section 2.9.

#### 2.2.8.8.2 RopDeleteProperties ROP Success Response Buffer

The following descriptions define valid fields for the **RopDeleteProperties** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
RopId										InputHandleIndex										ReturnValue													
...										PropertyProblemCount																							
PropertyProblems (variable)																																	
...																																	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x0B.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**PropertyProblemCount (2 bytes):** An unsigned integer that specifies the number of **PropertyProblem** structures in the **PropertyProblems** field.

**PropertyProblems (variable):** An array of **PropertyProblem** structures. The number of structures contained in this field is specified by the **PropertyProblemCount** field. The format of the **PropertyProblem** structure is specified in [\[MS-OXCDATA\]](#) section 2.7.

### 2.2.8.8.3 RopDeleteProperties ROP Failure Response Buffer

The following descriptions define valid fields for the **RopDeleteProperties** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x0B.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.8.9 RopDeletePropertiesNoReplicate ROP

The **RopDeletePropertiesNoReplicate** ROP deletes property values from an object without invoking replication. For more details about this operation, see [\[MS-OXCPRPT\]](#) section 2.2.8.

#### 2.2.8.9.1 RopDeletePropertiesNoReplicate ROP Request Buffer

The following descriptions define valid fields for the **RopDeletePropertiesNoReplicate** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1				
RopId										LogonId										InputHandleIndex						PropertyTagCount									
...										PropertyTags (variable)																									
...																																			

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x7A.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**PropertyTagCount (2 bytes):** An unsigned integer that specifies the number of **PropertyTag** structures in the **PropertyTags** field.

**PropertyTags (variable):** An array of **PropertyTag** structures that specifies the property values to be deleted from the object. The number of structures contained in this field is specified by the **PropertyTagCount** field. The format of the **PropertyTag** structure is specified in [\[MS-OXCDATA\]](#) section 2.9.

### 2.2.8.9.2 RopDeletePropertiesNoReplicate ROP Success Response Buffer

The following descriptions define valid fields for the **RopDeletePropertiesNoReplicate** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId								InputHandleIndex								ReturnValue															
...																PropertyProblemCount															
PropertyProblems (variable)																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x7A.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**PropertyProblemCount (2 bytes):** An unsigned integer that specifies the number of **PropertyProblem** structures in the **PropertyProblems** field.

**PropertyProblems (variable):** An array of **PropertyProblem** structures. The number of structures contained in this field is specified by the **PropertyProblemCount** field. The format of the **PropertyProblem** structure is specified in [\[MS-OXCDATA\]](#) section 2.7.

### 2.2.8.9.3 RopDeletePropertiesNoReplicate ROP Failure Response Buffer

The following descriptions define valid fields for the **RopDeletePropertiesNoReplicate** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId								InputHandleIndex								ReturnValue															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x7A.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.8.10 RopQueryNamedProperties ROP

The **RopQueryNamedProperties** ROP retrieves all the named properties for an object. For more details about this operation, see [\[MS-OXCPRPT\]](#) section 2.2.9.

#### 2.2.8.10.1 RopQueryNamedProperties ROP Request Buffer

The following descriptions define valid fields for the **RopQueryNamedProperties** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31								
RopId										LogonId										InputHandleIndex										QueryFlags									
HasGuid										PropertyGuid (optional)																													
...																																							
...																																							
...																																							
...																																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x5F.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**QueryFlags (1 byte):** A flags structure that contains flags control how this ROP behaves. The possible values are specified in [\[MS-OXCPRPT\]](#) section 2.2.9.1.

**HasGuid (1 byte):** A Boolean that specifies whether the **PropertyGuid** field is present.

**PropertyGuid (16 bytes):** A GUID that is present if **HasGuid** is nonzero and is not present if the value of the **HasGuid** field is zero. This value specifies the subset of named properties to be returned.

#### 2.2.8.10.2 RopQueryNamedProperties ROP Success Response Buffer

The following descriptions define valid fields for the **RopQueryNamedProperties** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										InputHandleIndex										ReturnValue											
...										IdCount																					
PropertyIds (variable)																															
...																															
PropertyNames (variable)																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x5F.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000 or 0x00040380.

**IdCount (2 bytes):** An unsigned integer that specifies the number of elements contained in the **PropertyIds** and **PropertyNames** fields.

**PropertyIds (variable):** An array of unsigned 16-bit integers. Each integer in the array is the property ID associated with a property name. The number of integers in the array is specified by the **IdCount** field.

**PropertyNames (variable):** A list of **PropertyName** structures that specifies the property names for the property IDs specified in the **PropertyIds** field. The number of structures contained in this field is specified by the **IdCount** field. The format of the **PropertyName** structure is specified in [\[MS-OXCDATA\]](#) section 2.6.1.

### 2.2.8.10.3 RopQueryNamedProperties ROP Failure Response Buffer

The following descriptions define valid fields for the **RopQueryNamedProperties** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x5F.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000 or 0x00040380.

### 2.2.8.11 RopCopyProperties ROP

The **RopCopyProperties** ROP copies property values from one object to another. For more details about this operation, see [\[MS-OXCPRPT\]](#) section 2.2.10.

#### 2.2.8.11.1 RopCopyProperties ROP Request Buffer

The following descriptions define valid fields for the **RopCopyProperties** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId				LogonId				SourceHandleIndex				DestHandleIndex																			
WantAsynchronous				CopyFlags				PropertyTagCount																							
PropertyTags (variable)																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x67.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**SourceHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the source Server object is stored. For more information about Server objects, see section [1.3.1](#).

**DestHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the destination Server object is stored.

**WantAsynchronous (1 byte):** A Boolean that specifies whether the operation is to be processed asynchronously with status reported via the **RopProgress** ROP (section [2.2.8.13](#)).

**CopyFlags (1 byte):** A flags structure that contains flags that control the operation behavior. The possible values are specified in [\[MS-OXCPRPT\]](#) section 2.2.10.1.

**PropertyTagCount (2 bytes):** An unsigned integer that specifies how many tags are present in the **PropertyTags** field.

**PropertyTags (variable):** An array of **PropertyTag** structures that specifies the properties to copy. The number of structures contained in this field is specified by the **PropertyTagCount** field. The format of the **PropertyTag** structure is specified in [\[MS-OXCADATA\]](#) section 2.9.

#### 2.2.8.11.2 RopCopyProperties ROP Success Response Buffer

The following descriptions define valid fields for the **RopCopyProperties** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										SourceHandleIndex										ReturnValue											
...										PropertyProblemCount																					
PropertyProblems (variable)																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x67.

**SourceHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **SourceHandleIndex** field specified in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**PropertyProblemCount (2 bytes):** An unsigned integer that specifies the number of **PropertyProblem** structures in the **PropertyProblems** field.

**PropertyProblems (variable):** An array of **PropertyProblem** structures. The number of structures contained in this field is specified by the **PropertyProblemCount** field. The format of the **PropertyProblem** structure is specified in [\[MS-OXCDATA\]](#) section 2.7.

### 2.2.8.11.3 RopCopyProperties ROP Null Destination Failure Response Buffer

The following descriptions define valid fields for the **RopCopyProperties** ROP null destination failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										SourceHandleIndex										ReturnValue											
...										DestHandleIndex																					
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x67.

**SourceHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **SourceHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000503.

**DestHandleIndex (4 bytes):** An unsigned integer index that MUST be set to the value specified in the **DestHandleIndex** field in the request.

### 2.2.8.11.4 RopCopyProperties ROP Failure Response Buffer



The following descriptions define valid fields for the **RopCopyProperties** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										SourceHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x67.

**SourceHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **SourceHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000 or 0x00000503.

### 2.2.8.12 RopCopyTo ROP

The **RopCopyTo** ROP copies properties from one Server object to another. For more details about this operation, see [\[MS-OXCPRPT\]](#) section 2.2.11.

#### 2.2.8.12.1 RopCopyTo ROP Request Buffer

The following descriptions define valid fields for the **RopCopyTo** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31										
RopId										LogonId										SourceHandleIndex										DestHandleIndex											
WantAsynchronous										WantSubObjects										CopyFlags										ExcludedTagCount											
...										ExcludedTags (variable)																															
...																																									

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x39.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**SourceHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the source Server object is stored. For more information about Server objects, see section [1.3.1](#).

**DestHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the destination Server object is stored.

**WantAsynchronous (1 byte):** A Boolean that specifies whether the operation is to be processed asynchronously with status reported via the **RopProgress** ROP (section [2.2.8.13](#)).

**WantSubObjects (1 byte):** A Boolean that specifies whether to copy subobjects.

**CopyFlags (1 byte):** A flags structure that contains flags that control the operation behavior. The possible values are specified in [\[MS-OXCPRPT\]](#) section 2.2.11.1.

**ExcludedTagCount (2 bytes):** An unsigned integer that specifies how many tags are present in the **ExcludedTags** field.

**ExcludedTags (variable):** An array of **PropertyTag** structures that specifies the properties to exclude from the copy. The number of structures contained in this field is specified by the **ExcludedTagCount** field. The format of the **PropertyTag** structure is specified in [\[MS-OXCDATA\]](#) section 2.9.

### 2.2.8.12.2 RopCopyTo ROP Success Response Buffer

The following descriptions define valid fields for the **RopCopyTo** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										SourceHandleIndex										ReturnValue											
...															PropertyProblemCount																
PropertyProblems (variable)																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x39.

**SourceHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **SourceHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**PropertyProblemCount (2 bytes):** An unsigned integer that specifies the number of **PropertyProblem** structures in the **PropertyProblems** field.

**PropertyProblems (variable):** An array of **PropertyProblem** structures. The number of structures contained in this field is specified by the **PropertyProblemCount** field. The format of the **PropertyProblem** structure is specified in [\[MS-OXCDATA\]](#) section 2.7.

### 2.2.8.12.3 RopCopyTo ROP Null Destination Failure Response Buffer

The following descriptions define valid fields for the **RopCopyTo** ROP null destination failure response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										SourceHandleIndex										ReturnValue											
...															DestHandleIndex																

...

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x39.

**SourceHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **SourceHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000503.

**DestHandleIndex (4 bytes):** An unsigned integer index that MUST be set to the **DestHandleIndex** specified in the request.

#### 2.2.8.12.4 RopCopyTo ROP Failure Response Buffer

The following descriptions define valid fields for the **RopCopyTo** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
RopId										SourceHandleIndex										ReturnValue													
...																																	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x39.

**SourceHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **SourceHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000 or 0x00000503.

#### 2.2.8.13 RopProgress ROP

The **RopProgress** ROP gets the status of an asynchronous operation. For more details about this operation, see [\[MS-OXCPRPT\]](#) section 2.2.22.

##### 2.2.8.13.1 RopProgress ROP Request Buffer

The following descriptions define valid fields for the **RopProgress** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
RopId										LogonId										InputHandleIndex										WantCancel									

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x50.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**WantCancel (1 byte):** A Boolean that specifies whether to cancel the operation.

### 2.2.8.13.2 RopProgress ROP Success Response Buffer

The following descriptions define valid fields for the **RopProgress** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										InputHandleIndex										ReturnValue											
...																				LogonId				CompletedTaskCount							
										...										TotalTaskCount											
										...																					

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x50.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**CompletedTaskCount (4 bytes):** An unsigned integer that specifies the number of tasks completed.

**TotalTaskCount (4 bytes):** An unsigned integer that specifies the total number of tasks.

### 2.2.8.13.3 RopProgress ROP Failure Response Buffer

The following descriptions define valid fields for the **RopProgress** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x50.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

## 2.2.9 Stream ROPs

### 2.2.9.1 RopOpenStream ROP

The **RopOpenStream** ROP opens a property for streaming access. For more details about this operation, see [\[MS-OXCPRPT\]](#) section 2.2.14.

#### 2.2.9.1.1 RopOpenStream ROP Request Buffer

The following descriptions define valid fields for the **RopOpenStream** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								LogonId								InputHandleIndex								OutputHandleIndex							
PropertyTag																															
OpenModeFlags																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x2B.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**OutputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the output Server object will be stored.

**PropertyTag (4 bytes):** A **PropertyTag** structure that specifies the property of the object to stream. The format of the **PropertyTag** structure is specified in [\[MS-OXCDATA\]](#) section 2.9.

**OpenModeFlags (1 byte):** A flags structure that contains flags that control how the stream is opened. The possible values are specified in [\[MS-OXCPRPT\]](#) section 2.2.14.1.

#### 2.2.9.1.2 RopOpenStream ROP Success Response Buffer

The following descriptions define valid fields for the **RopOpenStream** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								OutputHandleIndex								ReturnValue															
...																StreamSize															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x2B.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**StreamSize (4 bytes):** An unsigned integer that indicates the size of the stream opened.

### 2.2.9.1.3 RopOpenStream ROP Failure Response Buffer

The following descriptions define valid fields for the **RopOpenStream** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										OutputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x2B.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.9.2 RopReadStream ROP

The **RopReadStream** ROP reads bytes from a stream. For more details about this operation, see [\[MS-OXCPRPT\]](#) section 2.2.15.

#### 2.2.9.2.1 RopReadStream ROP Request Buffer

The following descriptions define valid fields for the **RopReadStream** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1				
RopId										LogonId										InputHandleIndex						ByteCount									
...										MaximumByteCount (optional)																									
...																																			

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x2C.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**ByteCount (2 bytes):** An unsigned integer that specifies the maximum number of bytes to read if the value is not equal to 0xBABE.

**MaximumByteCount (4 bytes):** An unsigned integer that specifies the maximum number of bytes to read if the value of the **ByteCount** field is equal to 0xBABE. <9> The **MaximumByteCount** field is present when the **ByteCount** field is equal to 0xBABE and is not present otherwise. If the value of the **MaximumByteCount** field is greater than 0x80000000, then the RPC SHOULD fail with error code 0x000004B6.

### 2.2.9.2.2 RopReadStream ROP Response Buffer

The following descriptions define valid fields for the **RopReadStream** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1				
RopId									InputHandleIndex									ReturnValue																	
...																		DataSize																	
Data (variable)																																			
...																																			

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x2C.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

**DataSize (2 bytes):** An unsigned integer that specifies the size, in bytes, of the **Data** field. The maximum size is specified in the request buffer by one of the following:

- The **ByteCount** field, when the value of the **ByteCount** field is not equal to 0xBABE.
- The **MaximumByteCount** field, when the value of the **ByteCount** field is equal to 0xBABE.

**Data (variable):** An array of bytes that are the bytes read from the stream. The size of this field, in bytes, is specified by the **DataSize** field.

### 2.2.9.3 RopWriteStream ROP

The **RopWriteStream** ROP writes bytes to a stream. For more details about this operation, see [\[MS-OXCPRPT\]](#) section 2.2.16.

#### 2.2.9.3.1 RopWriteStream ROP Request Buffer

The following descriptions define valid fields for the **RopWriteStream** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								LogonId								InputHandleIndex								DataSize							
...								Data (variable)																							
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x2D.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**DataSize (2 bytes):** An unsigned integer that specifies the size of the **Data** field.

**Data (variable):** An array of bytes that specifies the bytes to be written to the stream. The size of this field, in bytes, is specified by the **DataSize** field.

### 2.2.9.3.2 RopWriteStream ROP Response Buffer

The following descriptions define valid fields for the **RopWriteStream** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								InputHandleIndex								ReturnValue															
...								WrittenSize																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x2D.

**InputHandleIndex (1 byte):** An unsigned integer index that **MUST** be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

**WrittenSize (2 bytes):** An unsigned integer that specifies the number of bytes actually written.

### 2.2.9.4 RopCommitStream ROP

The **RopCommitStream** ROP commits stream operations. For more details about this operation, see [\[MS-OXCPRPT\]](#) section 2.2.17.

#### 2.2.9.4.1 RopCommitStream ROP Request Buffer

The following descriptions define valid fields for the **RopCommitStream** ROP request buffer.



0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId									LogonId									InputHandleIndex													

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x5D.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

### 2.2.9.4.2 RopCommitStream ROP Response Buffer

The following descriptions define valid fields for the **RopCommitStream** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId									InputHandleIndex									ReturnValue													
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x5D.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.9.5 RopGetStreamSize ROP

The **RopGetStreamSize** ROP gets the size of a stream. For more details about this operation, see [\[MS-OXCPRPT\]](#) section 2.2.18.

#### 2.2.9.5.1 RopGetStreamSize ROP Request Buffer

The following descriptions define valid fields for the **RopGetStreamSize** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId									LogonId									InputHandleIndex													

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x5E.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

### 2.2.9.5.2 RopGetStreamSize ROP Success Response Buffer

The following descriptions define valid fields for the **RopGetStreamSize** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										InputHandleIndex										ReturnValue											
...										...										StreamSize											
...										...										...											

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x5E.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**StreamSize (4 bytes):** An unsigned integer that is the current size of the stream.

### 2.2.9.5.3 RopGetStreamSize ROP Failure Response Buffer

The following descriptions define valid fields for the **RopGetStreamSize** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										InputHandleIndex										ReturnValue											
...										...										...											

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x5E.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.9.6 RopSetStreamSize ROP

The **RopSetStreamSize** ROP sets the size of a stream. For more details about this operation, see [\[MS-OXCPRPT\]](#) section 2.2.19.

#### 2.2.9.6.1 RopSetStreamSize ROP Request Buffer

The following descriptions define valid fields for the **RopSetStreamSize** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31				
RopId										LogonId										InputHandleIndex								StreamSize							
...																																			
...																																			

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x2F.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**StreamSize (8 bytes):** An unsigned integer that specifies the size of the stream. The server limits the maximum size of the stream to the value specified in a configuration file. If no value is specified in a configuration file, then the maximum size is 2<sup>31</sup> bytes. <10>

#### 2.2.9.6.2 RopSetStreamSize ROP Response Buffer

The following descriptions define valid fields for the **RopSetStreamSize** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x2F.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

#### 2.2.9.7 RopSeekStream ROP

The **RopSeekStream** ROP sets a pointer to a specific offset within a stream. For more details about this operation, see [\[MS-OXCPRPT\]](#) section 2.2.20.

##### 2.2.9.7.1 RopSeekStream ROP Request Buffer

The following descriptions define valid fields for the **RopSeekStream** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								LogonId								InputHandleIndex								Origin							
Offset																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x2E.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**Origin (1 byte):** An enumeration that specifies the origin location for the seek operation. The possible values for this enumeration are specified in [\[MS-OXCPRPT\]](#) section 2.2.20.1.

**Offset (8 bytes):** An unsigned integer that specifies the seek offset.

### 2.2.9.7.2 RopSeekStream ROP Success Response Buffer

The following descriptions define valid fields for the **RopSeekStream** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								InputHandleIndex								ReturnValue															
...																NewPosition															
...																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x2E.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**NewPosition (8 bytes):** An unsigned integer that represents the new position in the stream after the operation.

### 2.2.9.7.3 RopSeekStream ROP Failure Response Buffer

The following descriptions define valid fields for the **RopSeekStream** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								InputHandleIndex								ReturnValue															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x2E.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.9.8 RopCopyToStream ROP

The **RopCopyToStream** ROP copies a specified number of bytes from a source stream to a destination stream. For more details about this operation, see [\[MS-OXCPRPT\]](#) section 2.2.21.

#### 2.2.9.8.1 RopCopyToStream ROP Request Buffer

The following descriptions define valid fields for the **RopCopyToStream** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								LogonId								SourceHandleIndex								DestHandleIndex							
ByteCount																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x3A.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**SourceHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the source Server object is stored. For more information about Server objects, see section [1.3.1](#).

**DestHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the destination Server object is stored.

**ByteCount (8 bytes):** An unsigned integer that specifies the number of bytes to be copied.

#### 2.2.9.8.2 RopCopyToStream ROP Response Buffer

The following descriptions define valid fields for the **RopCopyToStream** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								SourceHandleIndex								ReturnValue															
...																ReadByteCount															
...																															
...																WrittenByteCount															
...																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x3A.

**SourceHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **SourceHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000503.

**ReadByteCount (8 bytes):** An unsigned integer that specifies the number of bytes read from the source object.

**WrittenByteCount (8 bytes):** An unsigned integer that specifies the number of bytes written to the destination object.

### 2.2.9.8.3 RopCopyToStream ROP Null Destination Failure Response Buffer

The following descriptions define valid fields for the **RopCopyToStream** ROP null destination failure response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								SourceHandleIndex								ReturnValue															
...																DestHandleIndex															
...																ReadByteCount															
...																															
...																WrittenByteCount															
...																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x3A.

**SourceHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **SourceHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000503.

**DestHandleIndex (4 bytes):** An unsigned integer index that MUST be set to the value specified in the **DestHandleIndex** field in the request.

**ReadByteCount (8 bytes):** An unsigned integer that, for this response, SHOULD be set to 0x0000000000000000.<11>

**WrittenByteCount (8 bytes):** An unsigned integer that, for this response, SHOULD be set to 0x0000000000000000.<12>

### 2.2.9.9 RopLockRegionStream ROP

The **RopLockRegionStream** ROP locks a specified range of bytes in a stream. For more details about this operation, see [\[MS-OXCPRPT\]](#) section 2.2.23.

#### 2.2.9.9.1 RopLockRegionStream ROP Request Buffer

The following descriptions define valid fields for the **RopLockRegionStream** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										LogonId						InputHandleIndex						RegionOffset									
...																															
...																									RegionSize						
...																															
...																									LockFlags						
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x5B.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**RegionOffset (8 bytes):** An unsigned integer that specifies the byte location in the stream where the region begins.

**RegionSize (8 bytes):** An unsigned integer that specifies the size of the region, in bytes.

**LockFlags (4 bytes):** A flags structure that contains flags specifying the behavior of the lock operation. The possible values for this structure are specified in [\[MS-OXCPRPT\]](#) section 2.2.23.1.

### 2.2.9.9.2 RopLockRegionStream ROP Response Buffer

The following descriptions define valid fields for the **RopLockRegionStream** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x5B.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.9.10 RopUnlockRegionStream ROP

The **RopUnlockRegionStream** ROP unlocks a specified range of bytes in a stream. For more details about this operation, see [\[MS-OXCPRPT\]](#) section 2.2.24.

#### 2.2.9.10.1 RopUnlockRegionStream ROP Request Buffer

The following descriptions define valid fields for the **RopUnlockRegionStream** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31								
RopId										LogonId										InputHandleIndex										RegionOffset									
...																																							
...																								RegionSize															
...																																							
...																								LockFlags															
...																																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x5C.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.



**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**RegionOffset (8 bytes):** An unsigned integer that specifies the byte location in the stream where the region begins.

**RegionSize (8 bytes):** An unsigned integer that specifies the size of the region, in bytes.

**LockFlags (4 bytes):** A flags structure that contains flags specifying the behavior of the lock operation. The possible values for this structure are specified in [\[MS-OXCPRPT\]](#) section 2.2.24.1.

### 2.2.9.10.2 RopUnlockRegionStream ROP Response Buffer

The following descriptions define valid fields for the **RopUnlockRegionStream** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x5C.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.9.11 RopWriteAndCommitStream ROP

The **RopWriteAndCommitStream** ROP writes bytes to a stream and commits the stream. For more details about this operation, see [\[MS-OXCPRPT\]](#) section 2.2.25.

#### 2.2.9.11.1 RopWriteAndCommitStream ROP Request Buffer

The following descriptions define valid fields for the **RopWriteAndCommitStream** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1				
RopId										LogonId										InputHandleIndex						DataSize									
...										Data (variable)																									
...																																			

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x90.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**DataSize (2 bytes):** An unsigned integer that specifies the size of the **Data** field.

**Data (variable):** An array of bytes to be written to the stream. The size of this field, in bytes, is specified by the **DataSize** field.

### 2.2.9.11.2 RopWriteAndCommitStream ROP Response Buffer

The response buffer for this ROP is the same as the response buffer for the **RopWriteStream** ROP, as specified in section [2.2.9.3.2](#), except the the **RopId** field is set to 0x90.

### 2.2.9.12 RopCloneStream ROP

The **RopCloneStream** ROP creates a new **Stream object** based on the same data as another Stream object. For more details about this operation, see [\[MS-OXCPRPT\]](#) section 2.2.26.

#### 2.2.9.12.1 RopCloneStream ROP Request Buffer

The following descriptions define valid fields for the **RopCloneStream** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31								
RopId										LogonId										InputHandleIndex										OutputHandleIndex									

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x3B.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**OutputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the output Server object will be stored.

#### 2.2.9.12.2 RopCloneStream ROP Response Buffer

The following descriptions define valid fields for the **RopCloneStream** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										OutputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x3B.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

## 2.2.10 Permission ROPs

### 2.2.10.1 RopModifyPermissions ROP

The **RopModifyPermissions** ROP modifies the permissions associated with a folder. For more details about this operation, see [\[MS-OXCPERM\]](#) section 2.2.2.

#### 2.2.10.1.1 RopModifyPermissions ROP Request Buffer

The following descriptions define valid fields for the **RopModifyPermissions** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31																																
RopId										LogonId										InputHandleIndex										ModifyFlags																																	
ModifyCount																PermissionsData (variable)																																															
...																																																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x40.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**ModifyFlags (1 byte):** A flags structure that contains flags that control the behavior of this operation. The possible values are specified in [\[MS-OXCPERM\]](#) section 2.2.2.1.

**ModifyCount (2 bytes):** An unsigned integer that specifies specifies the number of structures serialized in the **PermissionsData** field.

**PermissionsData (variable):** A list of **PermissionData** structures. The number of structures contained in this field is specified by the **ModifyCount** field. The format of the **PermissionData** structure is specified in section [2.2.10.1.1.1](#).

#### 2.2.10.1.1.1 PermissionData Structure

The following descriptions define valid fields for the **PermissionData** Structure.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31																																
PermissionDataFlags										PropertyValueCount																PropertyValues (variable)																																					
...																																																															

**PermissionDataFlags (1 byte):** A flags structure that contains flags that specify the type of operation.

**PropertyValueCount (2 bytes):** An unsigned integer that specifies the number of structures present in the **PropertyValues** field.

**PropertyValues (variable):** An array of **TaggedPropertyValue** structures that are used to specify and describe the modify operations. The number of structures contained in this field is specified by the **PropertyValueCount** field. The format of the **TaggedPropertyValue** structure is specified in [\[MS-OXCDATA\]](#) section 2.11.4.

### 2.2.10.1.2 RopModifyPermissions ROP Response Buffer

The following descriptions define valid fields for the **RopModifyPermissions** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
RopId										InputHandleIndex										ReturnValue													
...																																	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x40.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.10.2 RopGetPermissionsTable ROP

The **RopGetPermissionsTable** ROP gets the **permissions table** for a folder. For more details about this operation, see [\[MS-OXCPerm\]](#) section 2.2.1.

#### 2.2.10.2.1 RopGetPermissionsTable ROP Request Buffer

The following descriptions define valid fields for the **RopGetPermissionsTable** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
RopId										LogonId										InputHandleIndex										OutputHandleIndex									
TableFlags																																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x3E.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**OutputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the output Server object will be stored.

**TableFlags (1 byte):** A flags structure that contains flags that control the type of table. The possible values are specified in [\[MS-OXCPERM\]](#) section 2.2.1.1.

### 2.2.10.2.2 RopGetPermissionsTable ROP Response Buffer

The following descriptions define valid fields for the **RopGetPermissionsTable** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										OutputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x3E.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.11 Rule ROPs

#### 2.2.11.1 RopModifyRules ROP

The **RopModifyRules** ROP modifies the rules associated with a folder. For more details about this operation, see [\[MS-OXORULE\]](#) section 2.2.1.

##### 2.2.11.1.1 RopModifyRules ROP Request Buffer

The following descriptions define valid fields for the **RopModifyRules** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										LogonId										InputHandleIndex				ModifyRulesFlags							
RulesCount										RulesData (variable)																					
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x41.

**LogonId (1 byte):** An unsigned integer that specifies the logon on which the operation is performed.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**ModifyRulesFlags (1 byte):** A flags structure that contains flags that specify behavior of this operation. The possible values are specified in [\[MS-OXORULE\]](#) section 2.2.1.1.

**RulesCount (2 bytes):** An unsigned integer that specifies the number of structures serialized in the **RuleData** field.

**RulesData (variable):** A list of **RuleData** structures. The number of structures contained in this field is specified by the **RulesCount** field. The format of the **RuleData** structure is specified in section [2.2.11.1.1.1](#).

### 2.2.11.1.1.1 RuleData Structure

The following descriptions define valid fields for the **RuleData** structure.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
RuleDataFlags										PropertyValueCount										PropertyValues (variable)													
...																																	

**RuleDataFlags (1 byte):** A flags structure that specifies the type of operation.

**PropertyValueCount (2 bytes):** An unsigned integer that specifies the number of structures present in the **PropertyValues** field.

**PropertyValues (variable):** An array of **TaggedPropertyValue** structures that specify and describe the rule operations. The number of structures contained in this field is specified by the **PropertyValueCount** field. The format of the **TaggedPropertyValue** structure is specified in [\[MS-OXCDATA\]](#) section 2.11.4.

### 2.2.11.1.2 RopModifyRules ROP Response Buffer

The following descriptions define valid fields for the **RopModifyRules** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
RopId										InputHandleIndex										ReturnValue													
...																																	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x41.

**InputHandleIndex (1 byte):** An unsigned integer index that refers to the handle in the Server object handle table used as input for this operation. For more information about Server objects, see section [1.3.1](#).

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.11.2 RopGetRulesTable ROP

The **RopGetRulesTable** ROP gets the **rules table** of a folder. For more details about this operation, see [\[MS-OXORULE\]](#) section 2.2.2.

### 2.2.11.2.1 RopGetRulesTable ROP Request Buffer

The following descriptions define valid fields for the **RopGetRulesTable** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								LogonId								InputHandleIndex								OutputHandleIndex							
TableFlags																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x3F.

**LogonId (1 byte):** An unsigned integer that specifies the logon on which the operation is performed.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**OutputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the output Server object will be stored.

**TableFlags (1 byte):** A flags structure that contains flags that control the type of table. The possible values are specified in [\[MS-OXORULE\]](#) section 2.2.2.1.

### 2.2.11.2.2 RopGetRulesTable ROP Response Buffer

The following descriptions define valid fields for the **RopGetRulesTable** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								OutputHandleIndex								ReturnValue															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x3F.

**OutputHandleIndex (1 byte):** An unsigned integer index that refers to the handle in the Server object handle table specified. For more information about Server objects, see section [1.3.1](#).

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.11.3 RopUpdateDeferredActionMessages ROP

The **RopUpdateDeferredActionMessages** ROP updates the **entry IDs** in the **Deferred Action Messages (DAMs)**. For more details about this operation, see [\[MS-OXORULE\]](#) section 2.2.3.

#### 2.2.11.3.1 RopUpdateDeferredActionMessages ROP Request Buffer

The following descriptions define valid fields for the **RopUpdateDeferredActionMessages** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
RopId										LogonId										InputHandleIndex										ServerEntryIdSize									
...										ServerEntryId (variable)																													
...																																							
ClientEntryIdSize															ClientEntryId (variable)																								
...																																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x57.

**LogonId (1 byte):** An unsigned integer that specifies the logon on which the operation is performed.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**ServerEntryIdSize (2 bytes):** An unsigned integer that specifies the size of the **ServerEntryId** field.

**ServerEntryId (variable):** An array of bytes that specifies the ID of the message on the server. The size of this field, in bytes, is specified by the **ServerEntryIdSize** field.

**ClientEntryIdSize (2 bytes):** An unsigned integer that specifies the size of the **ClientEntryId** field.

**ClientEntryId (variable):** An array of bytes that specifies the ID of the downloaded message on the client. The size of this field, in bytes, is specified by the **ClientEntryIdSize** field.

### 2.2.11.3.2 RopUpdateDeferredActionMessages ROP Response Buffer

The following descriptions define valid fields for the **RopUpdateDeferredActionMessages** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x57.

**InputHandleIndex (1 byte):** An unsigned integer index that refers to the handle in the Server object handle table specified as the input handle. For more information about Server objects, see section [1.3.1](#).

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.



## 2.2.12 Fast Transfer ROPs

### 2.2.12.1 RopFastTransferDestinationConfigure ROP

The **RopFastTransferDestinationConfigure** ROP creates a destination fast transfer object. For more details about this operation, see [\[MS-OXCFXICS\]](#) section 2.2.3.1.2.1.

#### 2.2.12.1.1 RopFastTransferDestinationConfigure ROP Request Buffer

The following descriptions define valid fields for the **RopFastTransferDestinationConfigure** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31								
RopId										LogonId										InputHandleIndex										OutputHandleIndex									
SourceOperation										CopyFlags																													

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x53.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**OutputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the output Server object will be stored.

**SourceOperation (1 byte):** An enumeration that indicates how the data stream was created on the source. The possible values for this enumeration are specified in [\[MS-OXCFXICS\]](#) section 2.2.3.1.2.1.1.

**CopyFlags (1 byte):** A flags structure that contains flags that control the behavior of the transfer operation. The possible values are specified in [\[MS-OXCFXICS\]](#) section 2.2.3.1.2.1.1.

#### 2.2.12.1.2 RopFastTransferDestinationConfigure ROP Response Buffer

The following descriptions define valid fields for the **RopFastTransferDestinationConfigure** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										OutputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x53.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.12.2 RopFastTransferDestinationPutBuffer ROP

The **RopFastTransferDestinationPutBuffer** ROP sends a stream of data to a fast transfer destination object. For more details about this operation, see [\[MS-OXCFCICS\]](#) section 2.2.3.1.2.2.

#### 2.2.12.2.1 RopFastTransferDestinationPutBuffer ROP Request Buffer

The following descriptions define valid fields for the **RopFastTransferDestinationPutBuffer** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31								
RopId										LogonId										InputHandleIndex										TransferDataSize									
...										TransferData (variable)																													
...																																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x54.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**TransferDataSize (2 bytes):** An unsigned integer that specifies the size of the **TransferData** field.

**TransferData (variable):** An array of bytes that contains the data to be uploaded to the destination fast transfer object. The size of this field, in bytes, is specified by the **TransferDataSize** field.

#### 2.2.12.2.2 RopFastTransferDestinationPutBuffer ROP Response Buffer

The following descriptions define valid fields for the **RopFastTransferDestinationPutBuffer** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...										TransferStatus																					
InProgressCount																TotalStepCount															
Reserved										BufferUsedSize																					

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x54.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

**TransferStatus (2 bytes):** An enumeration that specifies the current status of the transfer. The possible values for this enumeration are specified in [\[MS-OXCFXICS\]](#) section 2.2.3.1.2.2.

**InProgressCount (2 bytes):** An unsigned integer that specifies the number of steps that have been completed in the current operation.

**TotalStepCount (2 bytes):** An unsigned integer that specifies the approximate total number of steps to be completed in the current operation.

**Reserved (1 byte):** Reserved. The server MUST set this field to 0x00.

**BufferUsedSize (2 bytes):** An unsigned integer that specifies the buffer size that was used.

### 2.2.12.3 RopFastTransferSourceGetBuffer ROP

The **RopFastTransferSourceGetBuffer** ROP retrieves a stream of data from a fast transfer source object. For more details about this operation, see [\[MS-OXCFXICS\]](#) section 2.2.3.1.1.5.

#### 2.2.12.3.1 RopFastTransferSourceGetBuffer ROP Request Buffer

The following descriptions define valid fields for the **RopFastTransferSourceGetBuffer** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31								
RopId										LogonId										InputHandleIndex										BufferSize									
...										MaximumBufferSize (optional)																													

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x4E.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**BufferSize (2 bytes):** An unsigned integer that specifies the buffer size requested.

**MaximumBufferSize (2 bytes):** An unsigned integer that is present when the **BufferSize** field is set to 0xBABE. This value specifies the maximum size limit when the server determines the buffer size.

#### 2.2.12.3.2 RopFastTransferSourceGetBuffer ROP Response Buffer

The following descriptions define valid fields for the **RopFastTransferSourceGetBuffer** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										InputHandleIndex										ReturnValue											
...										TransferStatus																					
InProgressCount										TotalStepCount																					
Reserved										TransferBufferSize										TransferBuffer (variable)											
...																															
BackoffTime (optional)																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x4E.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

**TransferStatus (2 bytes):** An enumeration that specifies the current status of the transfer. The possible values for this enumeration are specified in [\[MS-OXCFXICS\]](#) section 2.2.3.1.1.5.2.

**InProgressCount (2 bytes):** An unsigned integer that specifies the number of steps that have been completed in the current operation.

**TotalStepCount (2 bytes):** An unsigned integer that specifies the approximate number of steps to be completed in the current operation.

**Reserved (1 byte):** Reserved. The server MUST set this field to 0x00.

**TransferBufferSize (2 bytes):** An unsigned integer that specifies the size of the **TransferBuffer** field.

**TransferBuffer (variable):** An array of bytes that is present if the **ReturnValue** is not 0x00000480 and is not present otherwise. If present, the size of this field, in bytes, is specified by the **TransferBufferSize** field.

**BackoffTime (4 bytes):** An unsigned integer that is present if the **ReturnValue** is 0x00000480 and is not present otherwise. This value specifies the number of milliseconds for the client to wait before trying this operation again.

## 2.2.12.4 RopFastTransferSourceCopyFolder ROP

The **RopFastTransferSourceCopyFolder** ROP downloads properties and descendant subobjects of a specified folder. For more details about this operation, see [\[MS-OXCFXICS\]](#) section 2.2.3.1.1.4.

### 2.2.12.4.1 RopFastTransferSourceCopyFolder ROP Request Buffer

The following descriptions define valid fields for the **RopFastTransferSourceCopyFolder** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								LogonId								InputHandleIndex								OutputHandleIndex							
CopyFlags								SendOptions																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x4C.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**OutputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the output Server object will be stored.

**CopyFlags (1 byte):** A flags structure that contains flags that control the type of operation. The possible values are specified in [\[MS-OXCFXICS\]](#) section 2.2.3.1.1.4.1.

**SendOptions (1 byte):** A flags structure that contains flags that control the behavior of the operation. The possible values are specified in [\[MS-OXCFXICS\]](#) section 2.2.3.1.1.1.1.

#### 2.2.12.4.2 RopFastTransferSourceCopyFolder ROP Response Buffer

The following descriptions define valid fields for the **RopFastTransferSourceCopyFolder** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								OutputHandleIndex								ReturnValue															
								...																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x4C.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

#### 2.2.12.5 RopFastTransferSourceCopyMessages ROP

The **RopFastTransferSourceCopyMessages** ROP downloads from a folder the content and descendant subobjects for messages identified by a given set of IDs. For more details about this operation, see [\[MS-OXCFXICS\]](#) section 2.2.3.1.1.3.

##### 2.2.12.5.1 RopFastTransferSourceCopyMessages ROP Request Buffer

The following descriptions define valid fields for the **RopFastTransferSourceCopyMessages** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31								
RopId										LogonId										InputHandleIndex										OutputHandleIndex									
MessageIdCount																				MessageIds (variable)																			
...																																							
CopyFlags																SendOptions																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x4B.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**OutputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the output Server object will be stored.

**MessageIdCount (2 bytes):** An unsigned integer that specifies the number of identifiers in the **MessageIds** field.

**MessageIds (variable):** An array of 64-bit identifiers that specifies the messages to copy. The number of identifiers contained in this field is specified by the **MessageIdCount** field.

**CopyFlags (1 byte):** A flags structure that contains flags that control the type of operation. The possible values are specified in [\[MS-OXCFCICS\]](#) section 2.2.3.1.1.3.1.

**SendOptions (1 byte):** A flags structure that contains flags that control the behavior of the operation. The possible values are specified in [\[MS-OXCFCICS\]](#) section 2.2.3.1.1.1.1.

### 2.2.12.5.2 RopFastTransferSourceCopyMessages ROP Response Buffer

The following descriptions define valid fields for the **RopFastTransferSourceCopyMessages** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										OutputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x4B.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

## 2.2.12.6 RopFastTransferSourceCopyTo ROP

The **RopFastTransferSourceCopyTo** ROP downloads the properties of a given **messaging object** and its descendant subobjects. For more details about this operation, see [\[MS-OXCFXICS\]](#) section 2.2.3.1.1.1.

### 2.2.12.6.1 RopFastTransferSourceCopyTo ROP Request Buffer

The following descriptions define valid fields for the **RopFastTransferSourceCopyTo** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId				LogonId				InputHandleIndex				OutputHandleIndex																			
Level				CopyFlags																											
...				SendOptions				PropertyTagCount																							
PropertyTags (variable)																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x4D.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**OutputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the output Server object will be stored.

**Level (1 byte):** An unsigned integer that specifies whether descendant subobjects are copied.

**CopyFlags (4 bytes):** A flags structure that contains flags that control the behavior of the transfer operation. The possible values are specified in [\[MS-OXCFXICS\]](#) section 2.2.3.1.1.1.1.

**SendOptions (1 byte):** A flags structure that contains flags that control the behavior of the operation. The possible values are specified in [\[MS-OXCFXICS\]](#) section 2.2.3.1.1.1.1.

**PropertyTagCount (2 bytes):** An unsigned integer that specifies the number of structures in the **PropertyTags** field.

**PropertyTags (variable):** An array of **PropertyTag** structures that specifies the properties to exclude during the copy. The format of the **PropertyTag** structure is specified in [\[MS-OXCDATA\]](#) section 2.9. The number of structures contained in this field is specified by the **PropertyTagCount** field.

### 2.2.12.6.2 RopFastTransferSourceCopyTo ROP Response Buffer

The following descriptions define valid fields for the **RopFastTransferSourceCopyTo** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										OutputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x4D.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.12.7 RopFastTransferSourceCopyProperties ROP

The **RopFastTransferSourceCopyProperties** ROP copies properties from a messaging object to a fast transfer object. For more details about this operation, see [\[MS-OXCFICS\]](#) section 2.2.3.1.1.2.

#### 2.2.12.7.1 RopFastTransferSourceCopyProperties ROP Request Buffer

The following descriptions define valid fields for the **RopFastTransferSourceCopyProperties** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1				
RopId										LogonId										InputHandleIndex						OutputHandleIndex									
Level										CopyFlags										SendOptions						PropertyTagCount									
...										PropertyTags (variable)																									
...																																			

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x69.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**OutputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the output Server object will be stored.

**Level (1 byte):** An unsigned integer that specifies whether descendant subobjects are copied.



**CopyFlags (1 byte):** A flags structure behavior of the transfer operation. The possible values are specified in [\[MS-OXCFXICS\]](#) section 2.2.3.1.1.2.1.

**SendOptions (1 byte):** A flags structure that contains flags that control the behavior of the operation. The possible values are specified in [\[MS-OXCFXICS\]](#) section 2.2.3.1.1.2.1.

**PropertyTagCount (2 bytes):** An unsigned integer that specifies the number of structures in the **PropertyTags** field.

**PropertyTags (variable):** An array of **PropertyTag** structures that specifies the properties to copy. The format of the **PropertyTag** structure is specified in [\[MS-OXCDATA\]](#) section 2.9. The number of structures contained in this field is specified by the **PropertyTagCount** field.

### 2.2.12.7.2 RopFastTransferSourceCopyProperties ROP Response Buffer

The following descriptions define valid fields for the **RopFastTransferSourceCopyProperties** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										OutputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x69.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.12.8 RopTellVersion ROP

The **RopTellVersion** ROP provides the version of the other server in a server-to-client-to-server upload. For more details about this operation, see [\[MS-OXCFXICS\]](#) section 2.2.3.1.1.6.

#### 2.2.12.8.1 RopTellVersion ROP Request Buffer

The following descriptions define valid fields for the **RopTellVersion** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										LogonId										InputHandleIndex						Version					
...																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x86.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**Version (6 bytes):** An array of three unsigned 16-bit integers that contains the version information for the other server. The format of the version number is specified in [\[MS-OXCRPC\]](#) section 3.1.4.1.3.1.

### 2.2.12.8.2 RopTellVersion ROP Response Buffer

The following descriptions define valid fields for the **RopTellVersion** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x86.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.13 Incremental Change Synchronization ROPs

#### 2.2.13.1 RopSynchronizationConfigure ROP

The **RopSynchronizationConfigure** ROP configures a synchronization object. For more details about this operation, see [\[MS-OXCFXICS\]](#) section 2.2.3.2.1.1.

##### 2.2.13.1.1 RopSynchronizationConfigure ROP Request Buffer

The following descriptions define valid fields for the **RopSynchronizationConfigure** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31								
RopId										LogonId										InputHandleIndex										OutputHandleIndex									
SynchronizationType										SendOptions										SynchronizationFlags																			
RestrictionDataSize																RestrictionData (variable)																							
...																																							

SynchronizationExtraFlags	
PropertyTagCount	PropertyTags (variable)
...	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x70.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**OutputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the output Server object will be stored.

**SynchronizationType (1 byte):** An enumeration that controls the type of synchronization. The possible values for this enumeration are specified in [\[MS-OXCFXICS\]](#) section 2.2.3.2.1.1.1.

**SendOptions (1 byte):** A flags structure that controls the behavior of the operation. The possible values are specified in [\[MS-OXCFXICS\]](#) section 2.2.3.1.1.1.1.

**SynchronizationFlags (2 bytes):** A flags structure that contains flags that control the behavior of the synchronization. The possible values are specified in [\[MS-OXCFXICS\]](#) section 2.2.3.2.1.1.1.

**RestrictionDataSize (2 bytes):** An unsigned integer that specifies the length, in bytes, of the **RestrictionData** field.

**RestrictionData (variable):** A restriction packet, as specified in [\[MS-OXCADATA\]](#) section 2.12, that specifies the filter for this synchronization object. The size of this field, in bytes, is specified by the **RestrictionDataSize** field.

**SynchronizationExtraFlags (4 bytes):** A flags structure that contains flags control the additional behavior of the synchronization. The possible values are specified in [\[MS-OXCFXICS\]](#) section 2.2.3.2.1.1.1.

**PropertyTagCount (2 bytes):** An unsigned integer that specifies how many **PropertyTag** structures are present in the **PropertyTags** field.

**PropertyTags (variable):** An array of **PropertyTag** structures that specifies the property tags to be used for the synchronization process. The number of structures contained in this field is specified by the **PropertyTagCount** field. The format of the **PropertyTag** structure is specified in [\[MS-OXCADATA\]](#) section 2.9.

### 2.2.13.1.2 RopSynchronizationConfigure ROP Response Buffer

The following descriptions define valid fields for the **RopSynchronizationConfigure** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										OutputHandleIndex										ReturnValue											

...

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x70.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.13.2 RopSynchronizationImportMessageChange ROP

The **RopSynchronizationImportMessageChange** ROP imports new messages or full changes to existing messages into the **server replica**. For more details about this operation, see [\[MS-OXCFXICS\]](#) section 2.2.3.2.4.2.

#### 2.2.13.2.1 RopSynchronizationImportMessageChange ROP Request Buffer

The following descriptions define valid fields for the **RopSynchronizationImportMessageChange** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31				
RopId				LogonId				InputHandleIndex				OutputHandleIndex																							
ImportFlag				PropertyValueCount												PropertyValues (variable)																			
...																																			

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x72.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**OutputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the output Server object will be stored.

**ImportFlag (1 byte):** A flags structure that contains flags that control the behavior of the synchronization. The possible values are specified in [\[MS-OXCFXICS\]](#) section 2.2.3.2.4.2.1.

**PropertyValueCount (2 bytes):** An unsigned integer that specifies the number of structures present in the **PropertyValues** field.

**PropertyValues (variable):** An array of **TaggedPropertyValue** structures that specify extra properties on the message. The number of structures contained in this field is specified by the **PropertyValueCount** field. The format of the **TaggedPropertyValue** structure is specified in [\[MS-OXCDATA\]](#) section 2.11.4, and possible properties to be set are specified in [\[MS-OXCFXICS\]](#) section 2.2.3.2.4.2.1.

### 2.2.13.2.2 RopSynchronizationImportMessageChange ROP Success Response Buffer

The following descriptions define valid fields for the **RopSynchronizationImportMessageChange** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								OutputHandleIndex								ReturnValue															
...																MessageId															
...																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x72.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**MessageId (8 bytes):** An identifier. This field MUST be set to 0x0000000000000000.

### 2.2.13.2.3 RopSynchronizationImportMessageChange ROP Failure Response Buffer

The following descriptions define valid fields for the **RopSynchronizationImportMessageChange** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								OutputHandleIndex								ReturnValue															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x72.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.13.3 RopSynchronizationImportReadStateChanges ROP

The **RopSynchronizationImportReadStateChanges** ROP synchronizes a change in the read status for messages. More detailed information about this operation can be found in [\[MS-OXCFCICS\]](#) section 2.2.3.2.4.6.

### 2.2.13.3.1 RopSynchronizationImportReadStateChanges ROP Request Buffer

The following descriptions define valid fields for the **RopSynchronizationImportReadStateChanges** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31								
RopId										LogonId										InputHandleIndex										MessageReadStatesSize									
...										MessageReadStates (variable)																													
...																																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x80.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**MessageReadStatesSize (2 bytes):** An unsigned integer that specifies the size, in bytes, of the **MessageReadStates** field.

**MessageReadStates (variable):** A list of **MessageReadState** structures that specify the messages and associated read states to be changed. The format of the **MessageReadState** structure is defined in section [2.2.13.3.1.1](#). The size of this field, in bytes, is specified by the **MessageReadStatesSize** field.

#### 2.2.13.3.1.1 MessageReadState Structure

The following descriptions define valid fields for the **MessageReadState** structure.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
MessageIdSize																MessageId (variable)															
...																															
MarkAsRead																															

**MessageIdSize (2 bytes):** An unsigned integer that specifies the size of the **MessageId** field.

**MessageId (variable):** An array of bytes that identifies the message to be marked as read or unread. The size of this field, in bytes, is specified by the **MessageIdSize** field.

**MarkAsRead (1 byte):** A Boolean that specifies whether to mark the message as read or not.

### 2.2.13.3.2 RopSynchronizationImportReadStateChanges ROP Response Buffer

The following descriptions define valid fields for the **RopSynchronizationImportReadStateChanges** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x80.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.13.4 RopSynchronizationImportHierarchyChange ROP

The **RopSynchronizationImportHierarchyChange** ROP synchronizes a change to the folder hierarchy. For more details about this operation, see [\[MS-OXCFCICS\]](#) section 2.2.3.2.4.3.

#### 2.2.13.4.1 RopSynchronizationImportHierarchyChange ROP Request Buffer

The following descriptions define valid fields for the **RopSynchronizationImportHierarchyChange** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31								
RopId										LogonId										InputHandleIndex										HierarchyValueCount									
...										HierarchyValues (variable)																													
...																																							
PropertyValueCount																PropertyValues (variable)																							
...																																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x73.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**HierarchyValueCount (2 bytes):** An unsigned integer that specifies the number of structures present in the **HierarchyValues** field.

**HierarchyValues (variable):** An array of **TaggedPropertyValue** structures that specify hierarchy-related properties of the folder. The number of structures contained in this field is specified by the **HierarchyValueCount** field. The format of the **TaggedPropertyValue** structure is specified in [\[MS-OXCDATA\]](#) section 2.11.4, and possible properties to be set are specified in [\[MS-OXCFXICS\]](#) section 2.2.3.2.4.3.1.

**PropertyValueCount (2 bytes):** An unsigned integer that specifies the number of structures present in the **PropertyValues** field.

**PropertyValues (variable):** An array of **TaggedPropertyValue** structures that specify folder properties. The number of structures contained in this field is specified by the **PropertyValueCount** field. The format of the **TaggedPropertyValue** structure is specified in [\[MS-OXCDATA\]](#) section 2.11.4.

#### 2.2.13.4.2 RopSynchronizationImportHierarchyChange ROP Success Response Buffer

The following descriptions define valid fields for the **RopSynchronizationImportHierarchyChange** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...																				FolderId											
...																															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x73.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**FolderId (8 bytes):** An identifier. This field MUST be set to 0x0000000000000000.

#### 2.2.13.4.3 RopSynchronizationImportHierarchyChange ROP Failure Response Buffer

The following descriptions define valid fields for the **RopSynchronizationImportHierarchyChange** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...																															



**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x73.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.13.5 RopSynchronizationImportDeletes ROP

The **RopSynchronizationImportDeletes** ROP synchronizes deleted messages or folders. For more details about this operation, see [\[MS-OXCFXICS\]](#) section 2.2.3.2.4.5.

#### 2.2.13.5.1 RopSynchronizationImportDeletes ROP Request Buffer

The following descriptions define valid fields for the **RopSynchronizationImportDeletes** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										LogonId						InputHandleIndex						ImportDeleteFlags									
PropertyValueCount																PropertyValues (variable)															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x74.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**ImportDeleteFlags (1 byte):** A flags structure that contains flags that specify options for the imported deletions. The possible values for this field are specified in [\[MS-OXCFXICS\]](#) section 2.2.3.2.4.5.1.

**PropertyValueCount (2 bytes):** An unsigned integer that specifies the number of structures present in the **PropertyValues** field.

**PropertyValues (variable):** An array of **TaggedPropertyValue** structures that specify the folders or messages to delete. The number of structures contained in this field is specified by the **PropertyValueCount** field. The format of the **TaggedPropertyValue** structure is specified in [\[MS-OXCDATA\]](#) section 2.11.4, and possible properties to be set are specified in [\[MS-OXCFXICS\]](#) section 2.2.3.2.4.5.1.

#### 2.2.13.5.2 RopSynchronizationImportDeletes ROP Response Buffer

The following descriptions define valid fields for the **RopSynchronizationImportDeletes** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x74.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.13.6 RopSynchronizationImportMessageMove ROP

The **RopSynchronizationImportMessageMove** ROP synchronizes a move of a message from one folder to another. For more details about this operation, see [\[MS-OXCFCICS\]](#) section 2.2.3.2.4.4.

#### 2.2.13.6.1 RopSynchronizationImportMessageMove ROP Request Buffer

The following descriptions define valid fields for the **RopSynchronizationImportMessageMove** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31								
RopId										LogonId										InputHandleIndex										SourceFolderIdSize									
...																				SourceFolderId (variable)																			
...																																							
SourceMessageIdSize																																							
SourceMessageId (variable)																																							
...																																							
PredecessorChangeListSize																																							
PredecessorChangeList (variable)																																							
...																																							
DestinationMessageIdSize																																							
DestinationMessageId (variable)																																							

...
ChangeNumberSize
ChangeNumber (variable)
...

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x78.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**SourceFolderIdSize (4 bytes):** An unsigned integer that specifies the size of the **SourceFolderId** field.

**SourceFolderId (variable):** An array of bytes that identifies the parent folder of the source message. The size of this field, in bytes, is specified by the **SourceFolderIdSize** field.

**SourceMessageIdSize (4 bytes):** An unsigned integer that specifies the size of the **SourceMessageId** field.

**SourceMessageId (variable):** An array of bytes that identifies the source message. The size of this field, in bytes, is specified by the **SourceMessageIdSize** field.

**PredecessorChangeListSize (4 bytes):** An unsigned integer that specifies the size of the **PredecessorChangeList** field.

**PredecessorChangeList (variable):** An array of bytes. The size of this field, in bytes, is specified by the **PredecessorChangeListSize** field. This field is specified in [\[MS-OXCFXICS\]](#) section 2.2.2.3.

**DestinationMessageIdSize (4 bytes):** An unsigned integer that specifies the size of the **DestinationMessageId** field.

**DestinationMessageId (variable):** An array of bytes that identifies the destination message. The size of this field, in bytes, is specified by the **DestinationMessageIdSize** field.

**ChangeNumberSize (4 bytes):** An unsigned integer that specifies the size of the **ChangeNumber** field.

**ChangeNumber (variable):** An array of bytes that specifies the change number of the message. The size of this field, in bytes, is specified by the **ChangeNumberSize** field.

#### 2.2.13.6.2 RopSynchronizationImportMessageMove ROP Success Response Buffer

The following descriptions define valid fields for the **RopSynchronizationImportMessageMove** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								InputHandleIndex								ReturnValue															
...																MessageId															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x78.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**MessageId (8 bytes):** An identifier. This field MUST be set to 0x0000000000000000.

### 2.2.13.6.3 RopSynchronizationImportMessageMove ROP Failure Response Buffer

The following descriptions define valid fields for the **RopSynchronizationImportMessageMove** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								InputHandleIndex								ReturnValue															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x78.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

### 2.2.13.7 RopSynchronizationOpenCollector ROP

The **RopSynchronizationOpenCollector** ROP creates a new incremental change synchronization upload context. For more details about this operation, see [\[MS-OXCFCICS\]](#) section 2.2.3.2.4.1.

#### 2.2.13.7.1 RopSynchronizationOpenCollector ROP Request Buffer

The following descriptions define valid fields for the **RopSynchronizationOpenCollector** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId								LogonId								InputHandleIndex								OutputHandleIndex							
IsContentsCollector																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x7E.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**OutputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the output Server object will be stored.

**IsContentsCollector (1 byte):** A Boolean that specifies whether this synchronization upload context is for contents or for hierarchy.

### 2.2.13.7.2 RopSynchronizationOpenCollector ROP Response Buffer

The following descriptions define valid fields for the **RopSynchronizationOpenCollector** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId								OutputHandleIndex								ReturnValue															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x7E.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.13.8 RopSynchronizationGetTransferState ROP

The **RopSynchronizationGetTransferState** ROP opens a synchronization transfer object to retrieve the storage state properties. For more details about this operation, see [\[MS-OXCFXICS\]](#) section 2.2.3.2.3.1.

#### 2.2.13.8.1 RopSynchronizationGetTransferState ROP Request Buffer

The following descriptions define valid fields for the **RopSynchronizationGetTransferState** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId								LogonId								InputHandleIndex								OutputHandleIndex							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x82.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**OutputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the output Server object will be stored.

### 2.2.13.8.2 RopSynchronizationGetTransferState ROP Response Buffer

The following descriptions define valid fields for the **RopSynchronizationGetTransferState** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId								OutputHandleIndex								ReturnValue															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x82.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.13.9 RopSynchronizationUploadStateStreamBegin ROP

The **RopSynchronizationUploadStateStreamBegin** ROP begins an operation to upload stream data. For more details about this operation, see [\[MS-OXCFCICS\]](#) section 2.2.3.2.2.1.

#### 2.2.13.9.1 RopSynchronizationUploadStateStreamBegin ROP Request Buffer

The following descriptions define valid fields for the **RopSynchronizationUploadStateStreamBegin** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId								LogonId								InputHandleIndex								StateProperty							
...																								TransferBufferSize							

...

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x75.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**StateProperty (4 bytes):** A **PropertyTag** structure. The possible values for this field are specified in [\[MS-OXCFXICS\]](#) section 2.2.3.2.2.1.1. The format of the **PropertyTag** structure is specified in [\[MS-OXCDATA\]](#) section 2.9.

**TransferBufferSize (4 bytes):** An unsigned integer that specifies the size of the stream to be uploaded.

### 2.2.13.9.2 RopSynchronizationUploadStateStreamBegin ROP Response Buffer

The following descriptions define valid fields for the **RopSynchronizationUploadStateStreamBegin** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x75.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.13.10 RopSynchronizationUploadStateStreamContinue ROP

The **RopSynchronizationUploadStateStreamContinue** ROP uploads storage state property values. For more details about this operation, see [\[MS-OXCFXICS\]](#) section 2.2.3.2.2.2.

#### 2.2.13.10.1 RopSynchronizationUploadStateStreamContinue ROP Request Buffer

The following descriptions define valid fields for the **RopSynchronizationUploadStateStreamContinue** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								LogonId								InputHandleIndex								StreamDataSize							

...	StreamData (variable)
...	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x76.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**StreamDataSize (4 bytes):** An unsigned integer that specifies the size, in bytes, of the **StreamData** field.

**StreamData (variable):** An array of bytes that contains the state stream data to be uploaded. The size of this field, in bytes, is specified by the **StreamDataSize** field.

### 2.2.13.10.2 RopSynchronizationUploadStateStreamContinue ROP Response Buffer

The following descriptions define valid fields for the **RopSynchronizationUploadStateStreamContinue** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										InputHandleIndex										ReturnValue											
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x76.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.13.11 RopSynchronizationUploadStateStreamEnd ROP

The **RopSynchronizationUploadStateStreamEnd** ROP marks the end of a storage state property upload operation. For more details about this operation, see [\[MS-OXCFCICS\]](#) section 2.2.3.2.2.3.

#### 2.2.13.11.1 RopSynchronizationUploadStateStreamEnd ROP Request Buffer

The following descriptions define valid fields for the **RopSynchronizationUploadStateStreamEnd** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										LogonId										InputHandleIndex											



**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x77.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

### 2.2.13.11.2 RopSynchronizationUploadStateStreamEnd ROP Response Buffer

The following descriptions define valid fields for the **RopSynchronizationUploadStateStreamEnd** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
RopId										InputHandleIndex										ReturnValue													
...																																	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x77.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.13.12 RopSetLocalReplicaMidsetDeleted ROP

The **RopSetLocalReplicaMidsetDeleted** ROP marks a set of messages in a given folder as deleted. For more details about this operation, see [\[MS-OXCFCICS\]](#) section 2.2.3.2.4.8.

#### 2.2.13.12.1 RopSetLocalReplicaMidsetDeleted ROP Request Buffer

The following descriptions define valid fields for the **RopSetLocalReplicaMidsetDeleted** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
RopId										LogonId										InputHandleIndex										DataSize									
...										LongTermIdRangeCount																													
...										LongTermIdRanges (variable)																													
...																																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x93.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**DataSize (2 bytes):** An unsigned integer that specifies the size of both the **LongTermIdRangeCount** and **LongTermIdRanges** fields.

**LongTermIdRangeCount (4 bytes):** An unsigned integer that specifies the number of structures in the **LongTermIdRanges** field.

**LongTermIdRanges (variable):** An array of **LongTermIdRange** structures that specify the ranges of message identifiers that have been deleted. The number of structures contained in this field is specified by the **LongTermIdRangeCount** field. The format of the **LongTermIdRange** structure is specified in section [2.2.13.12.1.1](#).

### 2.2.13.12.1.1 LongTermIdRange Structure

The following descriptions define valid fields for the **LongTermIdRange** structure.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
MinLongTermId																															
...																															
...																															
...																															
...																															
...																															
...																															
MaxLongTermId																															
...																															
...																															
...																															
...																															
...																															

**MinLongTermId (24 bytes):** A **LongTermId** structure that specifies the beginning of a range. The format of the **LongTermId** structure is specified in [\[MS-OXCDATA\]](#) section 2.2.1.3.1.

**MaxLongTermId (24 bytes):** A **LongTermId** structure that specifies the end of a range. The format of the **LongTermId** structure is specified in [\[MS-OXCDATA\]](#) section 2.2.1.3.1.

### 2.2.13.12.2 RopSetLocalReplicaMidsetDeleted ROP Response Buffer

The following descriptions define valid fields for the **RopSetLocalReplicaMidsetDeleted** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								InputHandleIndex								ReturnValue															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x93.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.13.13 RopGetLocalReplicaIds ROP

The **RopGetLocalReplicaIds** ROP reserves a range of IDs to be used by a **local replica**. For more details about this operation, see [\[MS-OXCFCIS\]](#) section 2.2.3.2.4.7.

#### 2.2.13.13.1 RopGetLocalReplicaIds ROP Request Buffer

The following descriptions define valid fields for the **RopGetLocalReplicaIds** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								LogonId								InputHandleIndex								IdCount							
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x7F.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**IdCount (4 bytes):** An unsigned integer that specifies the number of IDs to reserve.

#### 2.2.13.13.2 RopGetLocalReplicaIds ROP Success Response Buffer

The following descriptions define valid fields for the **RopGetLocalReplicaIds** ROP success response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										InputHandleIndex										ReturnValue											
...										...										ReplGuid											
...										...										...											
...										...										GlobalCount											
...										...										...											

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x7F.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to 0x00000000.

**ReplGuid (16 bytes):** GUID. This field contains the replica GUID that is shared by the IDs.

**GlobalCount (6 bytes):** An array of bytes that specifies the first value in the reserved range.

### 2.2.13.13.3 RopGetLocalReplicaIds ROP Failure Response Buffer

The following descriptions define valid fields for the **RopGetLocalReplicaIds** ROP failure response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										InputHandleIndex										ReturnValue											
...										...										...											

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x7F.

**InputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **InputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP. For this response, this field is set to a value other than 0x00000000.

## 2.2.14 Notification ROPs

### 2.2.14.1 RopRegisterNotification ROP

The **RopRegisterNotification** ROP registers for notification events. For more details about this operation, see [\[MS-OXCNOTIF\]](#) section 2.2.1.2.1.

#### 2.2.14.1.1 RopRegisterNotification ROP Request Buffer

The following descriptions define valid fields for the **RopRegisterNotification** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
RopId										LogonId										InputHandleIndex										OutputHandleIndex									
NotificationTypes															Reserved										WantWholeStore														
FolderId (optional)																																							
...																																							
MessageId (optional)																																							
...																																							

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x29.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

**OutputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the output Server object will be stored.

**NotificationTypes (2 bytes):** A flags structure that contains flags that specify the types of events to register for. The possible values are specified in [\[MS-OXCNOTIF\]](#) section 2.2.1.2.1.1.

**Reserved (1 byte):** A flags structure. This field MUST be set to zero. This field is optional and will only be present if the **Extended** (0x0400) flag is set in the **NotificationTypes** field.

**WantWholeStore (1 byte):** A Boolean that specifies whether the notification is scoped to the mailbox instead of a specific folder or message.

**FolderId (8 bytes):** An identifier that is present when the value of the **WantWholeStore** field is zero and is not present when it is nonzero. This value specifies the folder to register notifications for.

**MessageId (8 bytes):** An identifier that is present when the value of the **WantWholeStore** field is zero and is not present when it is nonzero. This value specifies the message to register notifications for.

### 2.2.14.1.2 RopRegisterNotification ROP Response Buffer

The following descriptions define valid fields for the **RopRegisterNotification** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								OutputHandleIndex								ReturnValue															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x29.

**OutputHandleIndex (1 byte):** An unsigned integer index that MUST be set to the value specified in the **OutputHandleIndex** field in the request.

**ReturnValue (4 bytes):** An unsigned integer that specifies the status of the ROP.

### 2.2.14.2 RopNotify ROP

The **RopNotify** ROP provides notification event data to the client. There is no request buffer for this ROP. For more details about this operation, see [\[MS-OXCNOTIF\]](#) section 2.2.1.4.1.

#### 2.2.14.2.1 RopNotify ROP Response Buffer

The following descriptions define valid fields for the **RopNotify** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId								NotificationHandle																							
...								LogonId								NotificationData (variable)															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x2A.

**NotificationHandle (4 bytes):** A Server object handle that specifies the notification Server object associated with this notification event.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this notification event.

**NotificationData (variable):** Various structures. The various structures for this field are specified in [\[MS-OXCNOTIF\]](#) section 2.2.1.4.1.1.

### 2.2.14.3 RopPending ROP

The **RopPending** ROP notifies the client that there are pending notifications on a session. There is no request buffer for this ROP. For more details about this operation, see [\[MS-OXCNOTIF\]](#) section 2.2.1.3.4.

#### 2.2.14.3.1 RopPending ROP Response Buffer

The following descriptions define valid fields for the **RopPending** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										SessionIndex																					

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x6E.

**SessionIndex (2 bytes):** An unsigned integer index that specifies which session has pending notifications.

### 2.2.15 Other ROPs

#### 2.2.15.1 RopBufferTooSmall ROP

The **RopBufferTooSmall** ROP notifies the client that there is insufficient space to return all ROP responses.

##### 2.2.15.1.1 RopBufferTooSmall ROP Response Buffer

The following descriptions define valid fields for the **RopBufferTooSmall** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
RopId										SizeNeeded												RequestBuffers (variable)											
...																																	

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0xFF.

**SizeNeeded (2 bytes):** An unsigned integer that specifies the size required for the ROP output buffer.

**RequestBuffers (variable):** An array of bytes that contains the section of the ROP input buffer that was not executed because of the insufficient size of the ROP output buffer. The size of the array is equal to the size of the space remaining in the ROP output buffer. The layout of the ROP input buffer and ROP output buffer is specified in section [2.2.1](#).

#### 2.2.15.2 RopBackoff ROP

The **RopBackoff** ROP notifies the client that the server is busy and is requesting that the client retry later.

### 2.2.15.2.1 RopBackoff ROP Response Buffer

The following descriptions define valid fields for the **RopBackoff** ROP response buffer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopId										LogonId										Duration											
...															BackoffRopCount					BackoffRopData (variable)											
...																															
AdditionalDataSize																AdditionalData (variable)															
...																															

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0xF9.

**LogonId (1 byte):** An unsigned integer that specifies the logon to which the **RopBackoff** ROP response applies.

**Duration (4 bytes):** An unsigned integer that specifies the number of milliseconds to apply a logon backoff.

**BackoffRopCount (1 byte):** An unsigned integer that specifies the number of structures in the **BackoffRopData** field.

**BackoffRopData (variable):** An array of **BackoffRop** structures. The format of the **BackoffRop** structure is specified in section [2.2.15.2.1.1](#). This array specifies the operations to be backed off and also the backoff duration for each.

**AdditionalDataSize (2 bytes):** An unsigned integer that specifies the size of the **AdditionalData** field.

**AdditionalData (variable):** An array of bytes that specifies additional information about the backoff response. The size of this field, in bytes, is specified by the **AdditionalDataSize** field.

#### 2.2.15.2.1.1 BackoffRop Structure

The following descriptions define valid fields for the **BackoffRop** structure.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RopIdBackoff										Duration																					
...																															

**RopIdBackoff (1 byte):** An unsigned integer index that identifies the ROP to apply the ROP backoff to.



**Duration (4 bytes):** An unsigned integer that specifies the number of milliseconds to apply a ROP backoff.

### 2.2.15.3 RopRelease ROP

The **RopRelease** ROP releases all resources associated with a Server object. For more details about the dependencies of Server objects, see section [3.1.5.3](#).

#### 2.2.15.3.1 RopRelease ROP Request Buffer

The following descriptions define valid fields for the **RopRelease** ROP request buffer.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
RopId										LogonId										InputHandleIndex											

**RopId (1 byte):** An unsigned integer that specifies the type of ROP. For this operation this field is set to 0x01.

**LogonId (1 byte):** An unsigned integer that specifies the logon associated with this operation.

**InputHandleIndex (1 byte):** An unsigned integer index that specifies the location in the Server object handle table where the handle for the input Server object is stored. For more information about Server objects, see section [1.3.1](#).

Preliminary

## 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.

The following abstract data model (ADM) elements are maintained by the client.

**Logon Collection:** A collection of logon IDs in use.

**Server Object Handle Collection:** A collection of valid Server object handles received from the server but not yet released.

#### 3.1.2 Timers

None.

#### 3.1.3 Initialization

None.

#### 3.1.4 Higher-Layer Triggered Events

The client creates a ROP input buffer consisting of ROP requests for operations to be performed on the server. The client sends the ROP input buffer to the server and receives a ROP output buffer back from the server.

##### 3.1.4.1 Creating a ROP Input Buffer

The layout of the ROP input buffer and the ROP output buffer is specified in section [2.2.1](#).

A ROP input buffer is constructed by the client and sent to the server. The client packages its ROP requests together in the intended processing order and creates an associated Server object handle table.

When assembling a ROP input buffer, the client MUST use a Server object handle table large enough to include an entry for the largest index used by the ROP requests. Each entry that is referenced only as input for ROPs MUST be filled in with the handle of the Server object that is intended to be the input of the operation. Each entry that is referenced only as output SHOULD be filled in with the 0xFFFFFFFF value. Each entry that is referenced first as an output index and then as an input index SHOULD also be initialized to the 0xFFFFFFFF value.

Multiple ROPs in a ROP input buffer can use the same Server object handle table index. For example, a **RopOpenFolder** ROP (section [2.2.4.1](#)) can specify index 1 as the location to place the handle for the folder Server object. In the same ROP input buffer, a **RopGetContentsTable** ROP (section [2.2.4.14](#)) can specify index 1 as the location for the input Server object handle.

### 3.1.4.2 Logging On

The client logs on to a **message store** by using the **RopLogon** ROP (section [2.2.3.1](#)) before attempting any additional ROPs on the message store. Prior to logging on, the client begins an RPC session by calling the **EcDoConnectEx** method, as specified in [\[MS-OXCRPC\]](#) section 3.1.4.1, or the **Connect request type<13>**, as specified in [\[MS-OXCMAPIHTTP\]](#) section 2.2.4.1. Once the client has successfully connected to the server, the client begins a logon session by sending a **RopLogon** ROP request (section [2.2.3.1.1](#)).

Each logon session is identified by a logon ID created by the client. The logon ID is associated with a **Logon object**, which is created by the server during its processing of a **RopLogon** ROP request. The logon ID is initially used in the **RopLogon** ROP request and is then used in all subsequent ROPs that are issued on the logon session. The client specifies the logon ID in the **LogonId** field of the ROP request buffer of each ROP. Any 8-bit integer value is allowed for the logon ID, and the client does not have to specify logon IDs in a certain numeric order. The logon ID MUST be unique per RPC session (that is, per call to the **EcDoConnectEx** method).

The client can issue multiple **RopLogon** ROP requests per RPC session. If the client specifies an active logon ID, then the current logon session is released and replaced with the new one. The client manages the active logon ID by use of a **Logon Collection** abstract data model element.

### 3.1.5 Message Processing Events and Sequencing Rules

The client receives the ROP responses in the ROP output buffer. These ROP responses appear in the ROP output buffer in the same order as they were sent in the ROP input buffer. The ROP output buffer can also include ROP responses that do not have corresponding ROP requests in the ROP input buffer. These extra ROP responses include the **RopBackoff** ROP (section [2.2.15.2](#)), the **RopBufferTooSmall** ROP (section [2.2.15.1](#)), the **RopNotify** ROP (section [2.2.14.2](#)), and the **RopPending** ROP (section [2.2.14.3](#)).

#### 3.1.5.1 Extra ROP Responses

##### 3.1.5.1.1 Processing the RopBackoff ROP Response

Any client reporting its version as 12.00.4228.0000 or later, as specified in [\[MS-OXCRPC\]](#) section 3.2.4.1.3.2, MUST support processing the **RopBackoff** ROP response buffer. The layout of this ROP is specified in section [2.2.15.2](#).

The **RopBackoff** ROP can appear at any location within the **RopsList** field of the ROP output buffer. This ROP response indicates that the server requests the client delay the resending of ROP requests for the specified logon or for a type of ROP for a specified length of time. When this response contains a nonzero value in the **RopIdBackoff** field, as specified in section [2.2.15.2.1.1](#), it specifies the ROP request that is required to be delayed. The ROP response that was delayed and all subsequent ROP responses will not be in the buffer. When the **BackoffRopCount** field, as specified in section [2.2.15.2.1](#), is set to 0x00, all ROP requests for that logon are to be delayed.

##### 3.1.5.1.2 Processing the RopBufferTooSmall Response Buffer

The layout of the **RopBufferTooSmall** ROP is specified in section [2.2.15.1](#).

The **RopBufferTooSmall** ROP response is the last ROP response within the **RopsList** field of the ROP output buffer, as specified in section [2.2.1](#). The **RopBufferTooSmall** ROP response indicates that the size of the ROP output buffer is insufficient to return responses for all of the ROP requests sent. The **RequestBuffers** field of the **RopBufferTooSmall** ROP response includes all ROP requests that were not processed by the server. The client SHOULD<14> resend the unprocessed ROP requests in a new call to the server via the **EcDoRpcExt2** method or the **Execute** request type<15>. The value that the

client specifies in the *pcbOut* parameter of the **EcDoRpcExt2** method MUST be greater than or equal to the value of the **SizeNeeded** field of the **RopBufferTooSmall** ROP response. For more details about the **EcDoRpcExt2** method and the *pcbOut* parameter, see [\[MS-OXCRPC\]](#) section 3.1.4.2. For more details about the **Execute** request type, see [\[MS-OXCMAPIHTTP\]](#) section 2.2.4.2.

### 3.1.5.1.3 Processing the RopNotify and RopPending Responses

The **RopNotify** (section [2.2.14.2](#)) and **RopPending** (section [2.2.14.3](#)) ROP responses appear at the end of the **RopsList** field of the ROP output buffer, as specified in section [2.2.1](#). A **RopPending** ROP response SHOULD [<16>](#) be present only if the ROP output buffer does not contain all queued **RopNotify** ROP responses. For more details about how the client processes these ROPs, see [\[MS-OXCNOTIF\]](#) section 3.2.5.

### 3.1.5.2 Subsequent RopProgress ROP Requests

Once the client receives a **RopProgress** ROP response (section [2.2.8.13](#)), all subsequent **RopProgress** ROP requests MUST use the same Server object handle table values as those used in the first **RopProgress** ROP request.

### 3.1.5.3 Server Object Dependencies

The client MUST use the **RopRelease** ROP (section [2.2.15.3](#)) to release an object. The client MUST release an owned object before releasing the object's owner. For example, a Stream object that is owned by a folder is released before the owning folder is released. A summary of object owners and the objects that they can own is provided below.

An attachment owns the following:

- A Stream object that is opened on properties of the attachment
- An Embedded Message object that is created from the attachment
- A **FastTransfer download context** that is opened by a **RopFastTransferSourceCopyTo** ROP (section [2.2.12.6](#)) or a **RopFastTransferSourceCopyProperties** ROP (section [2.2.12.7](#))
- A **FastTransfer upload context** that is opened by a **RopFastTransferDestinationConfigure** ROP (section [2.2.12.1](#))

A message owns the following:

- An attachment of the message
- A Stream object that is opened on properties of the message
- An **attachments table** that is opened on the message
- A FastTransfer download context that is opened by a **RopFastTransferSourceCopyTo** ROP or a **RopFastTransferSourceCopyProperties** ROP
- A FastTransfer upload context that is opened by a **RopFastTransferDestinationConfigure** ROP (section [2.2.12.1](#))

A folder owns the following:

- A Stream object that is opened on properties of the folder
- A rules table that is opened on the folder

- A permissions table that is opened on the folder
- A fast-transfer download context that is opened by a **RopFastTransferSourceCopyTo** ROP or **RopFastTransferSourceCopyProperties** ROP
- A FastTransfer download context that is opened by a **RopFastTransferSourceCopyMessages** ROP (section [2.2.12.5](#)) or a **RopFastTransferSourceCopyFolder** ROP (section [2.2.12.4](#))
- A synchronization download context that is opened by a **RopSynchronizationConfigure** ROP (section [2.2.13.1](#))
- A FastTransfer upload context that is opened by a **RopFastTransferDestinationConfigure** ROP
- A synchronization upload context that is opened by a **RopSynchronizationOpenCollector** ROP (section [2.2.13.7](#))

#### 3.1.5.4 Code Page for Strings

In a ROP request, the code page for strings MUST match the code page associated with the affected Server object. The code page for a Server object is determined as follows:

- If the Server object is owned by a message, then the Server object uses the same code page that the message uses. The message's properties and the Server objects owned by the message always use the same code page that the message uses.
- If the Server object is not owned by a message, then the Server object uses the code page specified in the **EcDoConnectEx** method or a similar RPC method, or the **Connect** request type [<17>](#). For more details about the **EcDoConnectEx** method and RPC, see [\[MS-OXCRPC\]](#) section 3.1.4.1. For more details about the **Connect** request type, see [\[MS-OXCMAPIHTTP\]](#) section 2.2.4.1.

#### 3.1.6 Timer Events

None.

#### 3.1.7 Other Local Events

##### 3.1.7.1 Shutting Down

When the client shuts down, before it disconnects from the server, it SHOULD send **RopRelease** ROP requests (section [2.2.15.3](#)) to the server to clean up all Server objects it had created but had not yet released.

### 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 specified in this document.

The following ADM elements are maintained by the server.

**Logon Map:** A mapping of logon IDs and logon Server objects.

**Server Object Map:** A mapping of Server object handles and Server objects. There is one map for each logon Server object.

### 3.2.2 Timers

None.

### 3.2.3 Initialization

When a connection is first set up between a client and server via either the **ExDoConnectEx** 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, the **Server Object Map** and **Logon Map** ADM elements are empty.

### 3.2.4 Higher-Layer Triggered Events

#### 3.2.4.1 Sending Notifications to the Client

When events that the client has registered for occur, the server sends one or more **RopNotify** ROP responses (section 2.2.14.2). If not all queued **RopNotify** ROP responses can fit in the ROP output buffer, the server SHOULD <19> include a **RopPending** ROP response (section 2.2.14.3.1) in the ROP output buffer. More detailed information about notifications can be found in [MS-OXCNOTIF].

#### 3.2.4.2 Sending the RopBackoff ROP Response

When the server determines that a ROP request is required to be delayed, the server MUST either respond with a **RopBackoff** ROP response (section 2.2.15.2) or fail the **ExDoRpcExt2** method, as specified in [MS-OXCRPC] section 3.1.4.2, or the **Execute** request type, <20> as specified in [MS-OXCMAPIHTTP] section 2.2.4.2.

If the server version, as defined in [MS-OXCRPC] section 3.1.4.1.3.2, is greater than 08.00.0525.0 and the client version, as defined in [MS-OXCRPC] section 3.2.4.1.3.2, is greater than 12.0.4228.0, the server SHOULD substitute a **RopBackoff** ROP response in place of the response for the delayed ROP request. If either the client or the server is an earlier version, then the server MUST instead fail either the **ExDoRpcExt2** method by raising the `RPC_S_SERVER_TOO_BUSY` exception, as specified in [MS-OXCRPC], or the **Execute** request type with an **HTTP** status code of 200, a value of 0 (success) in the **X-ResponseCode** header, and a value of 0x000006BB in the **StatusCode** field of the **Execute** request type failure response body, as specified in [MS-OXCMAPIHTTP].

If ROPs are specified in a **RopBackoff** ROP response, all subsequent ROP requests SHOULD NOT be processed.

#### 3.2.4.3 Responding to Insufficient Output Buffer Space

When the server determines that there is not enough space in the ROP output buffer, as specified in section 2.2.1, for the remaining ROP responses, it MUST NOT process the remaining ROPs. The server responds in one of the following ways.

- If the ROP output buffer can be resized to accommodate the remaining ROPs, then the server MUST send a **RopBufferTooSmall** ROP response (section 2.2.15.1). In this case, the client can resubmit the ROP requests in either a new **ExDoRpcExt2** method call with the `pcbOut` parameter set to an appropriate value, as specified in [MS-OXCRPC] section 3.1.4.2, or a new **Execute** request type with the **MaxRopOut** field set to an appropriate value, <21> as specified in [MS-

[OXCMAPIHTTP](#) section 2.2.4.2.1. For details about how the client processes a **RopBufferTooSmall** ROP response, see section [3.1.5.1.2](#).

- If one of the ROP responses will not fit in the ROP output buffer when either the *pcbOut* parameter of the **EcDoRpcExt2** response, as specified in [MS-OXCRPC] section 3.1.4.2, or the **RopBufferSize** field of the **Execute** request type success response body, as specified in [MS-OXCMAPIHTTP] section 2.2.4.2.2, is set to the maximum value, then the server SHOULD [<22>](#) fail the **EcDoRpcExt2** method with a return value of 0x0000047D or fail the **Execute** request type with a value of 0x0000047D in the **StatusCode** field.

Some ROP response buffers are dynamically sized based on the amount of available space in the ROP output buffer. If the ROP output buffer is not large enough for the minimum size of a dynamically sized ROP response buffer followed by subsequent ROP response buffers, then the server sends the **RopBufferTooSmall** ROP.

## 3.2.5 Message Processing Events and Sequencing Rules

### 3.2.5.1 Processing a ROP Input Buffer

The layout of the ROP input buffer is specified in section [2.2.1](#).

When processing a ROP input buffer received from a client, the server MUST process the ROP requests in the order they are specified in the **RopsList** field and MUST NOT process more than one ROP input buffer concurrently for a connection. The ROP responses in the ROP output buffer MUST be in the same order in which they were processed.

If the server is unable to parse the ROP requests in the ROP input buffer, the RPC MUST fail by returning 0x000004B6.

During processing of a ROP request, the server resolves the Server object handle table index to a Server object. If the index is invalid, the server SHOULD [<23>](#) fail the ROP with the **ReturnValue** field set to 0x000004B9. Any index used for input is converted into the corresponding Server object by looking up the handle in the Server object handle table and then looking up the Server object in the **Server Object Map** ADM element associated with the logon for the ROP request. If any lookup in either the **Logon Map** ADM element or the **Server Object Map** ADM element fails, the server SHOULD [<24>](#) fill in a failure ROP response with a nonzero return value.

After successful processing of a ROP request that created a Server object, the server MUST assign an unused Server object handle to the object and record the mapping in the **Server Object Map** ADM element for the logon associated with the ROP. The handle assigned is then set in the Server object handle table at the location specified by the output index in the ROP request and can be used by subsequent ROP requests in the same ROP input buffer.

Because the Server object handle value 0xFFFFFFFF is used to initialize unused entries of a Server object handle table, a server MUST NOT assign that value to a created Server object.

### 3.2.5.2 Creating a ROP Output Buffer

The format of the ROP output buffer is specified in section [2.2.1](#).

The server constructs the ROP output buffer after processing the ROP input buffer. The ROP output buffer includes a list of ROP responses and a modified Server object handle table. The ROP responses in the ROP output buffer MUST be in the same order in which they were processed.

The Server object handle table MUST be large enough to contain an entry for the highest Server object handle table index specified in the ROP responses. The table can be a smaller size compared to the table in the ROP input buffer if entries at the end of the table were not referenced. The server

MUST preserve the order of entries in the Server object handle table between the ROP input buffer and the ROP output buffer.

### 3.2.5.3 Processing the RopRelease ROP Request

When the server receives a **RopRelease** ROP request, as specified in section [2.2.15.3](#), the server MUST remove the mapping for the Server object handle from the **Server Object Map** ADM element and release the resources associated with it. [<25>](#) The same Server object handle can now be reused for another Server object. If the Server object specified is a logon Server object, then the server MUST also remove it from the **Logon Map** ADM element.

Unlike all other ROP requests, the **RopRelease** ROP does not have any associated ROP response. The server MUST not return any response for a **RopRelease** ROP request. Any errors resulting from the processing of this ROP MUST be ignored and not sent back to the client.

### 3.2.5.4 Error Codes Returned When an Object Is Invalid

The error codes returned by the server when the object on the server has been released or becomes invalid are contained in the following table. These codes are valid for whatever object is represented by the Server object handle.

Name	Value	Meaning
ecInvalidObject	0x80040108	Returned when the client attempts to reuse any Server object handle that the server has previously issued a <b>RopRelease</b> ROP on and that hasn't been recycled for a new object.
ecNullObject	0x000004B9	Returned when the client attempts to use a Server object handle value that was never assigned to an open object.
ecAccessDenied	0x80070005	Returned when the client attempts to use the Server object handle from a different logon.

### 3.2.6 Timer Events

None.

### 3.2.7 Other Local Events

#### 3.2.7.1 Disconnecting

When the client disconnects, the server MUST release all Server objects, **Logon Map** ADM elements, and **Server Object Map** ADM elements associated with the connection.



## 4 Protocol Examples

### 4.1 Empty ROP Buffer

In the empty buffer scenario, the ROP request buffer contains no ROPs and no Server object handles. This type of buffer can be used by a client when it expects to receive extra information in the ROP output buffer, such as pending notifications.

```
02 00
```

**RopSize:** 02 00

**Rops:** <empty>

**ServerObjectHandleTable:** <empty>

### 4.2 Single ROP Request

In the single ROP request scenario, the buffer contains a single ROP that has an input index. The **ServerObjectHandleTable** field is offset from the beginning of the buffer by the number of bytes specified in the **RopSize** field. The Server object handle table in this buffer contains an unused entry.

```
09 00 15 01 01 02 01 FF 0F 6D 00 00 00 56 00 00 00
```

**RopSize:** 09 00

**Rop:**

**RopId:** 15 (**RopQueryRows** (section [2.2.5.4](#)))

**LogonId:** 01

**InputHandleIndex:** 01

**QueryRowsFlags:** 02

**ForwardRead:** 01

**RowCount:** FF 0F

**ServerObjectHandleTable:**

6D 00 00 00 (Handle 0, unused)

56 00 00 00 (Handle 1, input of **RopQueryRows**)

### 4.3 Multiple ROP Request

In the multiple ROP request scenario, the buffer consists of two ROP requests. The first is a **RopOpenFolder** ROP request (section [2.2.4.1](#)), and the second is a **RopGetHierarchyTable** ROP request (section [2.2.4.13](#)). The input for the second ROP is the output for the first ROP. All output handles are initialized to 0xFFFFFFFF.

```
14 00 02 00 00 01 01 00 59 65 73 73 69 72 00 04 00 01 02 04 6E 00 00 00 FF FF FF FF FF FF FF
FF
```

**RopSize:** 14 00

**Rops:**

**RopId:** 02 (**RopOpenFolder**)

**LogonId:** 00

**InputHandleIndex:** 00

**OutputHandleIndex:** 01

**FolderId:** 01 00 59 65 73 73 69 72 (ID of the folder to be opened)

**OpenModeFlags:** 00

**RopId:** 04 (**RopGetHierarchyTable**)

**LogonId:** 00

**InputHandleIndex:** 01

**OutputHandleIndex:** 02

**TableFlags:** 04

**ServerObjectHandleTable:**

6E 00 00 00 (Handle 0, input of **RopOpenFolder**)

FF FF FF FF (Handle 1, output of **RopOpenFolder**, input of **RopGetHierarchyTable**)

FF FF FF FF (Handle 2, output of **RopGetHierarchyTable**)

#### 4.4 RopRelease ROP Request

In the **RopRelease** ROP request scenario, the buffer contains a pair of **RopRelease** ROP requests (section [2.2.15.3](#)). These two ROPs are releasing two different Server objects, based on the different Server object handles that they reference.

```
08 00 01 00 00 01 00 01 6F 00 00 00 6E 00 00 00
```

**RopSize:** 08 00

**Rops:**

**RopId:** 01 (**RopRelease**)

**LogonId:** 00

**InputHandleIndex:** 00

**RopId:** 01 (**RopRelease**)

**LogonId:** 00

**InputHandleIndex:** 01

**ServerObjectHandleTable:**

6F 00 00 00 (Handle 0, input of first **RopRelease**)

6E 00 00 00 (Handle 1, input of second **RopRelease**)

#### 4.5 RopBufferTooSmall ROP Response

In the **RopBufferTooSmall** ROP response scenario, when a **RopOpenMessage** ROP (section [2.2.6.1](#)) call produces a response that won't fit in the output buffer (the output buffer being much smaller than usual for this example), the **RopBufferTooSmall** ROP (section [2.2.15.1](#)) indicates that an output buffer with at least 0x002C bytes is required to return the ROP response buffer from the **RopOpenMessage** ROP in the ROP request. The **RopOpenMessage** ROP request and the Server object handle table are the same as those specified in the input ROP buffer. In this scenario, the **RopBufferTooSmall** ROP is the first ROP, which indicates that no ROPs were processed before running out of room.

```
1C 00 FF 2C 00 03 00 00 01 FF 0F 01 00 15 89 00 78 27 1E 03 01 00 15 89 00 78 2F BB 12 00 00
00 FF FF FF FF
```

**RopSize:** 1C 00

**Rops:**

**RopId:** FF (**RopBufferTooSmall**)

**SizeNeeded:** 2C 00 (0x002C bytes)

**RopId:** 03 (**RopOpenMessage**)

**LogonId:** 00

**InputHandleIndex:** 00

**OutputHandleIndex:** 01

**CodePageID:** FF 0F

**FolderId:** 01 00 15 89 00 78 27 1E

**OpenModeFlags:** 03

**MessageId:** 01 00 15 89 00 78 2F BB

**ServerObjectHandleTable:**

12 00 00 00 (Handle 0, input of **RopOpenMessage**)

FF FF FF FF (Handle 1, output of **RopOpenMessage**)

#### 4.6 Logon RopBackoff ROP Response

In the logon **RopBackoff** ROP response scenario, the response buffer contains a **RopBackoff** ROP response (section [2.2.15.2](#)) in the same buffer as a **RopSetColumns** ROP response (section [2.2.5.1](#)). The **RopBackoff** ROP contains no information specific to a ROP but instead a suggested duration of the delay before the logon be retried.

12 00 12 00 00 00 00 00 00 00 F9 00 34 12 00 00 00 00 00 28 00 00 00

**RopSize:** 12 00

**Rops:**

**RopId:** 12 (**RopSetColumns**)

**InputHandleIndex:** 00

**ReturnValue:** 00 00 00 00

**TableStatus:** 00

**RopId:** F9 (**RopBackoff**)

**LogonId:** 00

**Duration:** 34 12 00 00 (suggested number of milliseconds before retry of logon (section [3.1.5.1.1](#)))

**BackoffRopCount:** 00

**AdditionalDataSize:** 00 00

**ServerObjectHandleTable:**

28 00 00 00 (Handle 0, input of **RopSetColumns**)

#### 4.7 RopBackoff ROP Response

In the **RopBackoff** ROP response scenario, the client has sent a **RopOpenFolder** ROP request (section [2.2.4.1](#)) and a **RopCreateFolder** ROP request (section [2.2.4.2](#)) in the same input buffer. The output buffer contains a **RopBackoff** ROP response (section [2.2.15.2](#)) and a successful **RopOpenFolder** ROP response. The **RopBackoff** ROP response contains information about only the **RopCreateFolder** ROP. The server is including the **RopBackoff** ROP response because it is too busy to complete processing of the **RopCreateFolder** ROP request.

18 00 02 01 00 00 00 00 00 00 F9 00 00 00 00 00 01 1C 17 4F 04 00 00 00 0A 00 00 00 24 00 00 00

**RopSize:** 18 00

**Rops:**

**RopId:** 02 (**RopOpenFolder**)

**OutputHandleIndex:** 01

**ReturnValue:** 00 00 00 00 (success response for **RopOpenFolder**)

**HasRules:** 00

**IsGhosted:** 00

**RopId:** F9 (**RopBackoff**)

**LogonId:** 00

**Duration:** 00 00 00 00

**BackoffRopCount:** 01

**RopIdBackoff:** 1C (**RopCreateFolder**)

**Duration:** 17 4F 04 00 (Suggested time in milliseconds before retrying **RopCreateFolder**)

**AdditionalDataSize:** 00 00

**ServerObjectHandleTable:**

0A 00 00 00 (Handle 0, unused)

24 00 00 00 (Handle 1, output of **RopOpenFolder**)

Preliminary

## **5 Security**

### **5.1 Security Considerations for Implementers**

None.

### **5.2 Index of Security Parameters**

None.

Preliminary

## 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 Preview
- Microsoft Office Outlook 2003
- Microsoft Office Outlook 2007
- Microsoft Outlook 2010
- Microsoft Outlook 2013
- Microsoft Outlook 2016 Preview

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 2.1](#): 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 **Execute** request type. The **Execute** request type was introduced in Microsoft Outlook 2013 Service Pack 1 (SP1) and Microsoft Exchange Server 2013 Service Pack 1 (SP1).

<2> [Section 2.2.1](#): 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 **Execute** request type. The **Execute** request type was introduced in Outlook 2013 SP1 and Exchange 2013 SP1.

<3> [Section 2.2.2](#): Exchange 2003 and Exchange 2007 behavior is undefined when they encounter a reserved value in a **RopId** field.

<4> [Section 2.2.2](#): 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 **Execute** request type. The **Execute** request type was introduced in Outlook 2013 SP1 and Exchange 2013 SP1.

<5> [Section 2.2.3.1.3](#): Exchange 2007 does not set the **PerUserGuid** field to an empty GUID.

<6> [Section 2.2.7.9](#): Outlook 2010 does not use the **RopOptionsData** ROP (section [2.2.7.9](#)).

<7> [Section 2.2.7.9.2](#): Exchange 2003, Exchange 2007, and Exchange 2010 do not return an empty array in the **OptionsInfo** field.

<8> [Section 2.2.7.9.3](#): Exchange 2007 sets the **ReturnValue** field for the **RopOptionsData** ROP response (section 2.2.7.9) to 0x00000000 regardless of the failure of the ROP.

<9> [Section 2.2.9.2.1](#): Exchange 2003 limits the stream size to 32 kilobytes. Therefore, if the value of the **ByteCount** field in the **RopReadStream** ROP request buffer (section [2.2.9.2](#)) is greater than 0x8000, the server returns the **RopBufferTooSmall** ROP (section [2.2.15.1](#)). The **MaximumByteCount** field is never present in the request buffer because the **ByteCount** field is never set to 0xBABE.

<10> [Section 2.2.9.6.1](#): Exchange 2003 and Exchange 2007 do not use a configuration setting and always limit the maximum size of the stream to 2<sup>31</sup> bytes.

<11> [Section 2.2.9.8.3](#): Exchange 2003 and Exchange 2007 do not set the field to 0x0000000000000000, but the nonzero value has no meaning.

<12> [Section 2.2.9.8.3](#): Exchange 2003 and Exchange 2007 do not set the field to 0x0000000000000000, but the nonzero value has no meaning.

<13> [Section 3.1.4.2](#): 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 Outlook 2013 SP1 and Exchange 2013 SP1.

<14> [Section 3.1.5.1.2](#): In Update Rollup 3 for Exchange Server 2010 Service Pack 2 (SP2), if the first ROP in the RPC buffer creates a response that exceeds the maximum ROP buffer size and there are no additional ROPs in the RPC buffer, then the server returns the **ecBufferTooSmall** error to fail the ROP instead of returning the **RopBufferTooSmall** ROP (section 2.2.15.1). Failing the ROP when it satisfies these conditions indicates to the client that there is no client action possible to make that ROP response fit in the ROP buffer.

<15> [Section 3.1.5.1.2](#): 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 **Execute** request type. The **Execute** request type was introduced in Outlook 2013 SP1 and Exchange 2013 SP1.

<16> [Section 3.1.5.1.3](#): Exchange 2003 and Exchange 2007 can include a **RopPending** ROP response (section [2.2.14.3](#)) even though the ROP output buffer contains all queued **RopNotify** ROP responses (section [2.2.14.2](#)).

<17> [Section 3.1.5.4](#): 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 Outlook 2013 SP1 and Exchange 2013 SP1.

<18> [Section 3.2.3](#): 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 Outlook 2013 SP1 and Exchange 2013 SP1.

<19> [Section 3.2.4.1](#): Exchange 2003 and Exchange 2007 sometimes include a **RopPending** ROP response (section 2.2.14.3) even though all queued **RopNotify** ROP responses (section 2.2.14.2) fit in the ROP output buffer.

<20> [Section 3.2.4.2](#): 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 **Execute** request type. The **Execute** request type was introduced in Outlook 2013 SP1 and Exchange 2013 SP1.



<21> [Section 3.2.4.3](#): 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 **Execute** request type. The **Execute** request type was introduced in Outlook 2013 SP1 and Exchange 2013 SP1.

<22> [Section 3.2.4.3](#): Exchange 2003 and Exchange 2007 do not fail the **EcDoRpcExt2** RPC and instead fail the individual ROP with the **ReturnValue** field of the ROP response buffer set to 0x0000047D.

<23> [Section 3.2.5.1](#): For some ROPs, Exchange 2003 and Exchange 2007 use different methods to resolve the Server object and, therefore, do not fail the ROP if the index is invalid.

<24> [Section 3.2.5.1](#): Exchange 2003 and Exchange 2007 fail the RPC.

<25> [Section 3.2.5.3](#): In Exchange 2010, in instances where multiple ROPs are being processed in one RPC and the **RopRelease** ROP is the last ROP in the call, the **RopRelease** ROP (section [2.2.15.3](#)) replaces the handle of the released Server object handle table entry with a value that is considered invalid by both server and client and is not 0xFFFFFFFF. This invalid handle value is then returned as the Server object handle in the ROP responses for the other ROPs in the RPC.

Preliminary

## 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](mailto:dochelp@microsoft.com).

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
<a href="#">2</a> Messages	Updated product behavior notes for the "Messages" section to include behavior of Exchange 2016 and Outlook 2016.	Y	Product behavior note updated.
<a href="#">2.2.6.9.2</a> RopGetMessageStatus ROP Response Buffers	Specified that the value of the RopId field for RopGetMessageStatus responses MUST be 0x20, which is the same as that for the RopSetMessageStatus responses.	N	Content update.
<a href="#">2.2.14.1.1</a> RopRegisterNotification ROP Request Buffer	Specified that NotificationTypes is 2 bytes and Reserved is optional. Updated table.	N	Content update.
<a href="#">3</a> Protocol Details	Updated product behavior notes for the "Protocol Details" section to include behavior of Exchange 2016 and Outlook 2016.	Y	Product behavior note updated.
<a href="#">6</a> Appendix A: Product Behavior	Added Exchange 2016 and Outlook 2016 to the list of applicable products.	Y	Content update.

## 8 Index

### A

Abstract data model

[client](#) 218

[server](#) 221

[Applicability](#) 21

### C

[Capability negotiation](#) 21

[Change tracking](#) 234

Client

[abstract data model](#) 218

[higher-layer triggered events](#) 218

[initialization](#) 218

[message processing](#) 219

[sequencing rules](#) 219

[timer events](#) 221

[timers](#) 218

Client - local events

[shutting down](#) 221

Client - message processing events

[code page for strings](#) 221

[Server object dependencies](#) 220

[subsequent RopProgress ROP requests](#) 220

Client - sequencing rules

[code page for strings](#) 221

[Server object dependencies](#) 220

[subsequent RopProgress ROP requests](#) 220

### D

Data model - abstract

[client](#) 218

[server](#) 221

### E

[Empty ROP buffer example](#) 225

Examples

[empty ROP buffer](#) 225

[Logon RopBackoff ROP response](#) 227

[multiple ROP request](#) 225

[ROP RopBackoff response](#) 228

[RopBufferTooSmall ROP response](#) 227

[RopRelease request](#) 226

[single ROP request](#) 225

### F

Fast transfer ROPs

[RopFastTransferDestinationConfigure ROP](#) 185

[RopFastTransferDestinationPutBuffer ROP](#) 186

[RopFastTransferSourceCopyFolder ROP](#) 188

[RopFastTransferSourceCopyMessages ROP](#) 189

[RopFastTransferSourceCopyProperties ROP](#) 192

[RopFastTransferSourceCopyTo ROP](#) 191

[RopFastTransferSourceGetBuffer ROP](#) 187

[RopTellVersion ROP](#) 193

[Fields - vendor-extensible](#) 21

Folder ROPs

[RopCopyFolder ROP](#) 68

[RopCreateFolder ROP](#) 57

[RopDeleteFolder ROP](#) 60

[RopDeleteMessages ROP](#) 71

[RopEmptyFolder ROP](#) 69

[RopGetContentsTable ROP](#) 75

[RopGetHierarchyTable ROP](#) 74

[RopGetSearchCriteria ROP](#) 62

[RopHardDeleteMessages ROP](#) 72

[RopHardDeleteMessagesAndSubfolders ROP](#) 70

[RopMoveCopyMessages ROP](#) 64

[RopMoveFolder ROP](#) 66

[RopOpenFolder ROP](#) 56

[RopSetSearchCriteria ROP](#) 61

[Format of ROP buffers](#) 20

### G

[Glossary](#) 15

### H

Higher-layer triggered events

[client](#) 218

Higher-layer triggered events - server

[responding to insufficient output buffer space](#) 222

[sending notifications to the client](#) 222

[sending the RopBackoff ROP response](#) 222

### I

[Implementer - security considerations](#) 230

Incremental change synchronization ROPs

[RopGetLocalReplicaIds ROP](#) 211

[RopSetLocalReplicaMidsetDeleted ROP](#) 209

[RopSynchronizationConfigure ROP](#) 194

[RopSynchronizationGetTransferState ROP](#) 205

[RopSynchronizationImportDeletes ROP](#) 201

[RopSynchronizationImportHierarchyChange ROP](#)

199

[RopSynchronizationImportMessageChange ROP](#)

196

[RopSynchronizationImportMessageMove ROP](#) 202

[RopSynchronizationImportReadStreamChanges ROP](#)

197

[RopSynchronizationOpenCollector ROP](#) 204

[RopSynchronizationUploadStateStreamBegin ROP](#)

206

[RopSynchronizationUploadStateStreamContinue](#)

ROP 207

[RopSynchronizationUploadStateStreamEnd ROP](#)

208

[Index of security parameters](#) 230

[Informative references](#) 19

Initialization

[client](#) 218

[server](#) 222

[Introduction](#) 15

## L

Local events - server

[disconnecting](#) 224

[Logon RopBackoff ROP response example](#) 227

Logon ROPs

[RopGetOwningServers ROP](#) 42

[RopGetPerUserGuid ROP](#) 50

[RopGetPerUserLongTermIds ROP](#) 49

[RopGetReceiveFolder ROP](#) 37

[RopGetReceiveFolderTable ROP](#) 39

[RopGetStoreState ROP](#) 41

[RopIdFromLongTermId ROP](#) 47

[RopLogon ROP](#) 31

[RopLongTermIdFromId ROP](#) 45

[RopPublicFolderIsGhosted ROP](#) 44

[RopReadPerUserInformation ROP](#) 52

[RopSetReceiveFolder ROP](#) 38

[RopWritePerUserInformation ROP](#) 54

## M

Message processing

[client](#) 219

Message processing - client

[code page for strings](#) 221

[Server object dependencies](#) 220

[subsequent RopProgress ROP requests](#) 220

Message processing - server

[creating a ROP output buffer](#) 223

[error codes returned when an object is invalid](#) 224

[processing a ROP input buffer](#) 223

[RopRelease](#) 224

Message ROPs

[RopCreateAttachment ROP](#) 123

[RopCreateMessage ROP](#) 106

[RopDeleteAttachment ROP](#) 124

[RopGetAttachmentTable ROP](#) 128

[RopGetMessageStatus ROP](#) 118

[RopGetValidAttachments ROP](#) 129

[RopModifyRecipients ROP](#) 110

[RopOpenAttachment ROP](#) 122

[RopOpenEmbeddedMessage ROP](#) 126

[RopOpenMessage ROP](#) 103

[RopReadRecipients ROP](#) 112

[RopReloadCachedInformation ROP](#) 114

[RopRemoveAllRecipients ROP](#) 110

[RopSaveChangesAttachment ROP](#) 125

[RopSaveChangesMessage ROP](#) 108

[RopSetMessageReadFlag ROP](#) 120

[RopSetMessageStatus ROP](#) 116

[RopSetReadFlags ROP](#) 119

Messages

[overview](#) 22

[ROP Input and Output Buffers](#) 22

[The Table of RopIds](#) 22

[transport](#) 22

[Multiple ROP request example](#) 225

## N

[Normative references](#) 19

Notification ROPs

[RopNotify ROP](#) 214

[RopPending ROP](#) 215

[RopRegisterNotification ROP](#) 213

## O

Other local events - client

[Shutting down](#) 221

Other ROPs

[RopBackoff ROP](#) 215

[RopBufferTooSmall ROP](#) 215

[RopRelease ROP](#) 217

Overview

[format of ROP buffers](#) 20

[Server objects](#) 20

[Overview \(synopsis\)](#) 19

## P

[Parameters - security index](#) 230

Permission ROPs

[RopGetPermissionsTable ROP](#) 180

[RopModifyPermissions ROP](#) 179

[Preconditions](#) 21

[Prerequisites](#) 21

[Product behavior](#) 231

Property ROPs

[RopCopyProperties ROP](#) 159

[RopCopyTo ROP](#) 161

[RopDeleteProperties ROP](#) 153

[RopDeletePropertiesNoReplicate ROP](#) 155

[RopGetNamesFromPropertyIds ROP](#) 144

[RopGetPropertiesAll ROP](#) 147

[RopGetPropertiesList ROP](#) 149

[RopGetPropertiesSpecific ROP](#) 145

[RopGetPropertyIdsFromNames ROP](#) 142

[RopProgress ROP](#) 163

[RopQueryNamedProperties ROP](#) 157

[RopSetProperties ROP](#) 150

[RopSetPropertiesNoReplicate ROP](#) 152

## R

[References](#) 18

[informative](#) 19

[normative](#) 19

[Relationship to other protocols](#) 20

[ROP Input and Output Buffers message](#) 22

[ROP RopBackoff response example](#) 228

[RopAbort ROP table ROP](#) 83

[RopAbortSubmit ROP transport ROP](#) 132

[RopBackoff ROP other ROP](#) 215

[RopBufferTooSmall ROP other ROP](#) 215

[RopBufferTooSmall ROP response example](#) 227

[RopCloneStream ROP stream ROP](#) 178

[RopCollapseRow ROP table ROP](#) 99

[RopCommitStream ROP stream ROP](#) 168

[RopCopyFolder ROP folder ROP](#) 68

[RopCopyProperties ROP property ROP](#) 159

[RopCopyTo ROP property ROP](#) 161

[RopCopyToStream ROP stream ROP](#) 173

[RopCreateAttachment ROP message ROP](#) 123

[RopCreateBookmark ROP table ROP](#) 91

[RopCreateFolder ROP folder ROP](#) 57  
[RopCreateMessage ROP message ROP](#) 106  
[RopDeleteAttachment ROP message ROP](#) 124  
[RopDeleteFolder ROP folder ROP](#) 60  
[RopDeleteMessages ROP folder ROP](#) 71  
[RopDeleteProperties ROP property ROP](#) 153  
[RopDeletePropertiesNoReplicate ROP property ROP](#) 155  
[RopEmptyFolder ROP folder ROP](#) 69  
[RopExpandRow ROP table ROP](#) 97  
[RopFastTransferDestinationConfigure ROP fast transfer ROP](#) 185  
[RopFastTransferDestinationPutBuffer ROP fast transfer ROP](#) 186  
[RopFastTransferSourceCopyFolder ROP fast transfer ROP](#) 188  
[RopFastTransferSourceCopyMessages ROP fast transfer ROP](#) 189  
[RopFastTransferSourceCopyProperties ROP fast transfer ROP](#) 192  
[RopFastTransferSourceCopyTo ROP fast transfer ROP](#) 191  
[RopFastTransferSourceGetBuffer ROP fast transfer ROP](#) 187  
[RopFindRow ROP table ROP](#) 94  
[RopFreeBookmark ROP table ROP](#) 95  
[RopGetAddressTypes ROP transport ROP](#) 133  
[RopGetAttachmentTable ROP message ROP](#) 128  
[RopGetCollapseState ROP table ROP](#) 100  
[RopGetContentsTable ROP folder ROP](#) 75  
[RopGetFolderIsGhosted ROP logon ROP](#) 44  
[RopGetHierarchyTable ROP folder ROP](#) 74  
[RopGetLocalReplicaIds ROP incremental change synchronization ROP](#) 211  
[RopGetMessageStatus ROP message ROP](#) 118  
[RopGetNamesFromPropertyIds ROP property ROP](#) 144  
[RopGetOwningServers ROP logon ROP](#) 42  
[RopGetPermissionsTable ROP permission ROP](#) 180  
[RopGetPerUserGuid ROP logon ROP](#) 50  
[RopGetPerUserLongTermIds ROP logon ROP](#) 49  
[RopGetPropertiesAll ROP property ROP](#) 147  
[RopGetPropertiesList ROP property ROP](#) 149  
[RopGetPropertiesSpecific ROP property ROP](#) 145  
[RopGetPropertyIdsFromNames ROP property ROP](#) 142  
[RopGetReceiveFolder ROP logon ROP](#) 37  
[RopGetReceiveFolderTable ROP logon ROP](#) 39  
[RopGetRulesTable ROP rule ROP](#) 182  
[RopGetSearchCriteria ROP folder ROP](#) 62  
[RopGetStatus ROP table ROP](#) 84  
[RopGetStoreState ROP logon ROP](#) 41  
[RopGetStreamSize ROP stream ROP](#) 169  
[RopGetTransportFolder ROP transport ROP](#) 139  
[RopGetValidAttachments ROP message ROP](#) 129  
[RopHardDeleteMessages ROP folder ROP](#) 72  
[RopHardDeleteMessagesAndSubfolders ROP folder ROP](#) 70  
[RopIdFromLongTermId ROP logon ROP](#) 47  
[RopLockRegionStream ROP stream ROP](#) 175  
[RopLogon ROP logon ROP](#) 31  
[RopLongTermIdFromId ROP logon ROP](#) 45  
[RopModifyPermissions ROP permission ROP](#) 179  
[RopModifyRecipients ROP message ROP](#) 110  
[RopModifyRules ROP rule ROP](#) 181  
[RopMoveCopyMessages ROP folder ROP](#) 64  
[RopMoveFolder ROP folder ROP](#) 66  
[RopOpenAttachment ROP message ROP](#) 122  
[RopOpenEmbeddedMessages ROP message ROP](#) 126  
[RopOpenFolder ROP folder ROP](#) 56  
[RopOpenMessage ROP message ROP](#) 103  
[RopOpenStream ROP stream ROP](#) 165  
[RopOptionsData ROP transport ROP](#) 140  
[RopPending ROP notification ROP](#) 215  
[RopProgress ROP property ROP](#) 163  
[RopQueryColumnsAll ROP table ROP](#) 92  
[RopQueryNamedProperties ROP property ROP](#) 157  
[RopQueryPosition ROP table ROP](#) 85  
[RopQueryRows ROP table ROP](#) 81  
[RopReadPerUserInformation ROP logon ROP](#) 52  
[RopReadRecipients ROP message ROP](#) 112  
[RopReadStream ROP stream ROP](#) 166  
[RopRegisterNotification ROP notification ROP](#) 213  
[RopRegisterNotify ROP notification ROP](#) 214  
[RopRelease request example](#) 226  
[RopRelease ROP other ROP](#) 217  
[RopReloadCachedInformation ROP message ROP](#) 114  
[RopRemoveAllRecipients ROP message ROP](#) 110  
[RopResetTable ROP table ROP](#) 96  
[RopRestrict ROP table ROP](#) 80  
[RopSaveChangesAttachment ROP message ROP](#) 125  
[RopSaveChangesMessage ROP message ROP](#) 108  
[RopSeekRow ROP table ROP](#) 87  
[RopSeekRowBookmark ROP table ROP](#) 88  
[RopSeekRowFractional ROP table ROP](#) 90  
[RopSeekStream ROP stream ROP](#) 171  
[RopSetCollapseState ROP table ROP](#) 102  
[RopSetColumns ROP table ROP](#) 77  
[RopSetLocalReplicaMidsetDeleted ROP incremental change synchronization ROP](#) 209  
[RopSetMessageReadFlag ROP message ROP](#) 120  
[RopSetMessageStatus ROP message ROP](#) 116  
[RopSetProperties ROP property ROP](#) 150  
[RopSetPropertiesNoReplicate ROP property ROP](#) 152  
[RopSetReadFlags ROP message ROP](#) 119  
[RopSetReceiveFolder ROP logon ROP](#) 38  
[RopSetSearchCriteria ROP folder ROP](#) 61  
[RopSetSpooler ROP transport ROP](#) 134  
[RopSetStreamSize ROP stream ROP](#) 170  
[RopSortTable ROP table ROP](#) 78  
[RopSpoolerLockMessage ROP transport ROP](#) 135  
[RopSubmitMessage ROP transport ROP](#) 131  
[RopSynchronizationConfigure ROP incremental change synchronization ROP](#) 194  
[RopSynchronizationGetTransferState ROP incremental change synchronization ROP](#) 205  
[RopSynchronizationImportDeletes ROP incremental change synchronization ROP](#) 201  
[RopSynchronizationImportHierarchyChange ROP incremental change synchronization ROP](#) 199  
[RopSynchronizationImportMessageChange ROP incremental change synchronization ROP](#) 196  
[RopSynchronizationImportMessageMove ROP incremental change synchronization ROP](#) 202  
[RopSynchronizationImportReadStateChanges ROP incremental change synchronization ROP](#) 197  
[RopSynchronizationOpenCollector ROP incremental change synchronization ROP](#) 204

[RopSynchronizationUploadStateStreamBegin ROP incremental change synchronization ROP](#) 206  
[RopSynchronizationUploadStateStreamContinue ROP incremental change synchronization ROP](#) 207  
[RopSynchronizationUploadStateStreamEnd ROP incremental change synchronization ROP](#) 208  
[RopTellVersion ROP fast transfer ROP](#) 193  
[RopTransportNewMail ROP transport ROP](#) 138  
[RopTransportSend ROP transport ROP](#) 136  
[RopUnlockRegionStream ROP stream ROP](#) 176  
[RopUpdateDeferredActionMessages ROP rule ROP](#) 183

[RopWriteandCommitStream ROP stream ROP](#) 177  
[RopWritePerUserInformation ROP logon ROP](#) 54  
[RopWriteStream ROP stream ROP](#) 167

#### Rule ROPs

[RopGetRulesTable ROP](#) 182  
[RopModifyRules ROP](#) 181  
[RopUpdateDeferredActionMessages ROP](#) 183

## S

### Security

[implementer considerations](#) 230  
[parameter index](#) 230

### Sequencing rules

[client](#) 219

### Sequencing rules - client

[code page for strings](#) 221  
[Server object dependencies](#) 220  
[subsequent RopProgress ROP requests](#) 220

### Sequencing rules - server

[creating a ROP output buffer](#) 223  
[error codes returned when an object is invalid](#) 224  
[processing a ROP input buffer](#) 223  
[RopRelease](#) 224

### Server

[abstract data model](#) 221  
[initialization](#) 222  
[timer events](#) 224  
[timers](#) 222

### Server - higher-layer triggered events

[responding to insufficient output buffer space](#) 222  
[sending notifications to the client](#) 222  
[sending the RopBackoff ROP response](#) 222

### Server - local events

[disconnecting](#) 224

### Server - message processing events

[creating a ROP output buffer](#) 223  
[error codes returned when an object is invalid](#) 224  
[processing a ROP input buffer](#) 223  
[RopRelease](#) 224

### Server - sequencing rules

[creating a ROP output buffer](#) 223  
[error codes returned when an object is invalid](#) 224  
[processing a ROP input buffer](#) 223  
[RopRelease](#) 224

### Server objects

[Single ROP request example](#) 225

### Standards assignments

### Stream ROPs

[RopCloneStream ROP](#) 178  
[RopCommitStream ROP](#) 168  
[RopCopyToStream ROP](#) 173

[RopGetStreamSize ROP](#) 169  
[RopLockRegionStream ROP](#) 175  
[RopOpenStream ROP](#) 165  
[RopReadStream ROP](#) 166  
[RopSeekStream ROP](#) 171  
[RopSetStreamSize ROP](#) 170  
[RopUnlockRegionStream ROP](#) 176  
[RopWriteAndCommitStream ROP](#) 177  
[RopWriteStream ROP](#) 167

## T

### Table ROPs

[RopAbort ROP](#) 83  
[RopCollapseRow ROP](#) 99  
[RopCreateBookmark ROP](#) 91  
[RopExpandRow ROP](#) 97  
[RopFindRow ROP](#) 94  
[RopFreeBookmark ROP](#) 95  
[RopGetCollapseState ROP](#) 100  
[RopGetStatus ROP](#) 84  
[RopQueryColumnsAll ROP](#) 92  
[RopQueryPosition ROP](#) 85  
[RopQueryRows ROP](#) 81  
[RopResetTable ROP](#) 96  
[RopRestrict ROP](#) 80  
[RopSeekRow ROP](#) 87  
[RopSeekRowBookmark ROP](#) 88  
[RopSeekRowFractional ROP](#) 90  
[RopSetCollapseState ROP](#) 102  
[RopSetColumns ROP](#) 77  
[RopSortTable ROP](#) 78  
[The Table of RopIds message](#) 22

### Timer events

[client](#) 221  
[server](#) 224

### Timers

[client](#) 218  
[server](#) 222

### Tracking changes

### Transport

### Transport ROPs

[RopAbortSubmit ROP](#) 132  
[RopGetAddressTypes ROP](#) 133  
[RopGetTransportFolder ROP](#) 139  
[RopOptionsData ROP](#) 140  
[RopSetSpooler ROP](#) 134  
[RopSpoolerLockMessage ROP](#) 135  
[RopSubmitMessage ROP](#) 131  
[RopTransportNewMail ROP](#) 138  
[RopTransportSend ROP](#) 136

### Triggered events - higher-layer

[client](#) 218

### Triggered events - server

[responding to insufficient output buffer space](#) 222  
[sending notifications to the client](#) 222  
[sending the RopBackoff ROP response](#) 222

## V

### Vendor-extensible fields

### Versioning

Preliminary