

## [MS-SFMWA-Diff]:

# Server and File Management Web APIs Protocol

---

### Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation (“this documentation”) for protocols, file formats, data portability, computer languages, and standards as well as overviews of the interaction among each of these technologies support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you maycan make copies of it in order to develop implementations of the technologies that are described in the Open Specifications this documentation and maycan distribute portions of it in your implementations using that use these technologies or in your documentation as necessary to properly document the implementation. You maycan also distribute in your implementation, with or without modification, any schema, IDL's schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications- documentation.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that maymight cover your implementations of the technologies described in the Open Specifications- documentation. Neither this notice nor Microsoft's delivery of the this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specification may Specifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in the Open Specifications this documentation 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 maymight 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 email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

**Tools.** The Open Specifications documentation does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments, you are free to take advantage of them. Certain Open Specifications documents are intended for use in conjunction with publicly available standard standards specifications and network programming art, and assumes, as such, assume 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.~~

## Revision Summary

Date	Revision History	Revision Class	Comments
8/8/2013	1.0	New	Released new document.
11/14/2013	1.0	None	No changes to the meaning, language, or formatting of the technical content.
2/13/2014	1.0	None	No changes to the meaning, language, or formatting of the technical content.
5/15/2014	1.0	None	No changes to the meaning, language, or formatting of the technical content.
6/30/2015	1.0	<del>No</del> <del>Change</del> None	No changes to the meaning, language, or formatting of the technical content.
<u>7/14/2016</u>	<u>1.0</u>	<u>None</u>	<u>No changes to the meaning, language, or formatting of the technical content.</u>

# Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>18</b>
1.1	Glossary .....	18
1.2	References .....	19
1.2.1	Normative References .....	19
1.2.2	Informative References .....	19
1.3	Overview .....	19
1.4	Relationship to Other Protocols .....	20
1.5	Prerequisites/Preconditions .....	20
1.6	Applicability Statement .....	20
1.7	Versioning and Capability Negotiation .....	20
1.8	Vendor-Extensible Fields .....	20
1.9	Standards Assignments.....	20
<b>2</b>	<b>Messages.....</b>	<b>21</b>
2.1	Transport .....	21
2.2	Common Data Types .....	21
2.2.1	Namespaces .....	21
2.2.2	HTTP Headers .....	21
2.2.2.1	AppName.....	22
2.2.2.2	AppPublisherName .....	22
2.2.2.3	AppVersion.....	22
2.2.2.4	Authorization.....	22
2.2.2.5	Canary .....	23
2.2.2.6	Accept .....	23
2.2.2.7	Range.....	23
2.2.3	Common URI Parameters .....	23
2.2.3.1	alertkey .....	25
2.2.3.2	amount.....	25
2.2.3.3	ascending .....	25
2.2.3.4	containerpath .....	26
2.2.3.5	count.....	26
2.2.3.6	deletecontents.....	26
2.2.3.7	description .....	26
2.2.3.8	deviceid .....	26
2.2.3.9	email.....	26
2.2.3.10	emails .....	27
2.2.3.11	filter .....	27
2.2.3.12	firstname .....	27
2.2.3.13	groupguid .....	27
2.2.3.14	groupingdata.....	27
2.2.3.15	groupingmethoddata .....	27
2.2.3.16	heightdata .....	28
2.2.3.17	id.....	28
2.2.3.18	index.....	28
2.2.3.19	isoriginaldata.....	28
2.2.3.20	keywords .....	29
2.2.3.21	language.....	29
2.2.3.22	lastname.....	29
2.2.3.23	localusername .....	29
2.2.3.24	mediatypedata .....	29
2.2.3.25	name.....	29
2.2.3.26	newname .....	30
2.2.3.27	newpath .....	30
2.2.3.28	onlineusername .....	30
2.2.3.29	originalpath.....	30

2.2.3.30	overwrite	30
2.2.3.31	overwritepermissions	30
2.2.3.32	path	31
2.2.3.33	permission	31
2.2.3.34	query	31
2.2.3.35	remotewebaccess	31
2.2.3.36	requestedcountdata	32
2.2.3.37	scope	32
2.2.3.38	servername	32
2.2.3.39	shallowdata	32
2.2.3.40	sharename	32
2.2.3.41	sizedata	33
2.2.3.42	sortbyfield	33
2.2.3.43	sortorderdata	33
2.2.3.44	sortpropertydata	33
2.2.3.45	startingindex	34
2.2.3.46	startingindexdata	34
2.2.3.47	timeoutseconds	34
2.2.3.48	titleonlydata	34
2.2.3.49	usergroups	34
2.2.3.50	userid	34
2.2.3.51	username	35
2.2.3.52	usersid	35
2.2.3.53	vpnaccess	35
2.2.3.54	widthdata	35
2.2.4	Complex Types	35
2.2.4.1	AlertInfo	38
2.2.4.2	ArrayOfAlertInfo	39
2.2.4.3	ArrayOfConnectionInfo	39
2.2.4.4	ArrayOfDeviceInfo	39
2.2.4.5	ArrayOfDriveInfo	40
2.2.4.6	ArrayOfFolderInfo	40
2.2.4.7	ArrayOfGroupInfo	40
2.2.4.8	ArrayOfItemInfo	41
2.2.4.9	ArrayOfLink	41
2.2.4.10	ArrayOfMailbox	41
2.2.4.11	ArrayOfMetadataBase	41
2.2.4.12	ArrayOfMetadataItemStream	42
2.2.4.13	ArrayOfMSODomain	42
2.2.4.14	ArrayOfMSOLicense	42
2.2.4.15	ArrayOfMSOLicenseService	42
2.2.4.16	ArrayOfMSOLicenseSubscription	43
2.2.4.17	ArrayOfMSOLicenseSuite	43
2.2.4.18	ArrayOfMSOUser	43
2.2.4.19	ArrayOfStorageDriveInfo	44
2.2.4.20	ArrayOfStorageServerInfo	44
2.2.4.21	ArrayOfstring	44
2.2.4.22	ArrayOfUserInfo	44
2.2.4.23	ArrayOfWebApiProvider	45
2.2.4.24	CompanyAddress	45
2.2.4.25	ConnectionClientInfo	46
2.2.4.26	ConnectionInfo	46
2.2.4.27	CustomizationInfo	47
2.2.4.28	DeviceInfo	47
2.2.4.29	DriveInfo	49
2.2.4.30	FolderInfo	50
2.2.4.31	GroupInfo	51
2.2.4.32	ItemInfo	52

2.2.4.33	ItemList.....	52
2.2.4.34	ItemThumbnail .....	53
2.2.4.35	Link.....	53
2.2.4.36	Mailbox.....	53
2.2.4.37	MetadataBase.....	54
2.2.4.38	MetadataContainer .....	54
2.2.4.39	MetadataDetailResult.....	55
2.2.4.40	MetadataItem.....	55
2.2.4.41	MetadataItemStream.....	56
2.2.4.42	MetadataResult.....	56
2.2.4.43	MetadataStreamResult.....	57
2.2.4.44	MSODomain .....	57
2.2.4.45	MSOLicense.....	57
2.2.4.46	MSOLicenseService .....	58
2.2.4.47	MSOLicenseSubscription .....	58
2.2.4.48	MSOLicenseSuite .....	59
2.2.4.49	MSOTenantInformation.....	59
2.2.4.50	MSOUser.....	60
2.2.4.51	PartialCollection_Of_AlertInfo .....	60
2.2.4.52	PartialCollection_Of_ConnectionInfo .....	61
2.2.4.53	PartialCollection_Of_DeviceInfo .....	61
2.2.4.54	PartialCollection_Of_DriveInfo .....	62
2.2.4.55	PartialCollection_Of_FolderInfo .....	62
2.2.4.56	PartialCollection_Of_UserInfo .....	62
2.2.4.57	ReadOnlyCollectionOfGroupInfoEV6sb80H .....	63
2.2.4.58	ReadOnlyCollectionOfMSOLicensePGX_Pb6b .....	63
2.2.4.59	ReadOnlyCollectionOfMSOLicenseServicePGX_Pb6b .....	63
2.2.4.60	ReadOnlyCollectionOfMSOLicenseSubscriptionPGX_Pb6b .....	63
2.2.4.61	ReadOnlyCollectionOfMSOLicenseSuitePGX_Pb6b .....	64
2.2.4.62	RemoteConnectionUserInfo .....	64
2.2.4.63	SearchItemList .....	65
2.2.4.64	ServerInfo .....	65
2.2.4.65	SharePointSiteAddressCollection .....	66
2.2.4.66	StorageDriveInfo .....	66
2.2.4.67	StorageServerInfo.....	67
2.2.4.68	UserInfo .....	67
2.2.4.69	WebApiProvider .....	68
2.2.5	Simple Types .....	69
2.2.5.1	ContainerType .....	69
2.2.5.2	guid .....	70
2.2.5.3	MSODomainTypes .....	70
2.2.5.4	MSOLicenseServiceTypes .....	70
2.2.5.5	Permission .....	71
2.2.5.6	ServerFolderType.....	71
2.2.5.7	ThumbnailFormat.....	72
<b>3</b>	<b>Protocol Details.....</b>	<b>74</b>
3.1	SessionService Server Details.....	74
3.1.1	Abstract Data Model.....	74
3.1.1.1	SessionId.....	74
3.1.1.2	Canary .....	74
3.1.2	Timers .....	74
3.1.3	Initialization .....	74
3.1.4	Higher-Layer Triggered Events .....	74
3.1.5	Message Processing Events and Sequencing Rules .....	74
3.1.5.1	login.....	75
3.1.5.1.1	GET .....	75
3.1.5.1.1.1	Request Body.....	76

3.1.5.1.1.2	Response Body .....	76
3.1.5.1.1.3	Processing Details .....	76
3.1.5.2	logout.....	76
3.1.5.2.1	GET .....	76
3.1.5.2.1.1	Request Body .....	77
3.1.5.2.1.2	Response Body .....	77
3.1.5.2.1.3	Processing Details .....	77
3.1.6	Timer Events.....	77
3.1.7	Other Local Events.....	77
3.2	IFileContentAccessService Server Details .....	77
3.2.1	Abstract Data Model.....	77
3.2.2	Timers .....	77
3.2.3	Initialization .....	77
3.2.4	Higher-Layer Triggered Events .....	77
3.2.5	Message Processing Events and Sequencing Rules .....	78
3.2.5.1	filecontent?path={path} .....	78
3.2.5.1.1	GET .....	78
3.2.5.1.1.1	Request Body .....	79
3.2.5.1.1.2	Response Body .....	79
3.2.5.1.1.3	Processing Details .....	79
3.2.5.2	filecontent?path={path}&overwrite={overwrite} .....	79
3.2.5.2.1	POST .....	80
3.2.5.2.1.1	Request Body .....	80
3.2.5.2.1.2	Response Body .....	80
3.2.5.2.1.3	Processing Details .....	80
3.2.5.3	itemthumbnail?path={path} .....	81
3.2.5.3.1	GET .....	81
3.2.5.3.1.1	Request Body .....	81
3.2.5.3.1.2	Response Body .....	81
3.2.5.3.1.3	Processing Details .....	82
3.2.6	Timer Events.....	82
3.2.7	Other Local Events.....	82
3.3	IFileOperationService Server Details .....	82
3.3.1	Abstract Data Model.....	82
3.3.2	Timers .....	82
3.3.3	Initialization .....	82
3.3.4	Higher-Layer Triggered Events .....	82
3.3.5	Message Processing Events and Sequencing Rules .....	82
3.3.5.1	items/index/{index}/count/{count}?path={path}&filter={filter}&sortByfield={sortByfield}&ascending={ascending} .....	84
3.3.5.1.1	GET .....	85
3.3.5.1.1.1	Request Body .....	85
3.3.5.1.1.2	Response Body .....	85
3.3.5.1.1.3	Processing Details .....	86
3.3.5.2	items/index/{index}/count/{count}/search?query={query}&sortByfield={sortByfield}&ascending={ascending}&scope={scope}&timeoutseconds={timeoutseconds} .....	86
3.3.5.2.1	GET .....	86
3.3.5.2.1.1	Request Body .....	87
3.3.5.2.1.2	Response Body .....	87
3.3.5.2.1.3	Processing Details .....	87
3.3.5.3	folder/newsubfoldername?path={path}&language={language}.....	87
3.3.5.3.1	GET .....	88
3.3.5.3.1.1	Request Body .....	88
3.3.5.3.1.2	Response Body .....	88
3.3.5.3.1.3	Processing Details .....	88

3.3.5.4	itemmetadata?path={path}	89
3.3.5.4.1	GET	89
3.3.5.4.1.1	Request Body	89
3.3.5.4.1.2	Response Body	89
3.3.5.4.1.3	Processing Details	90
3.3.5.5	folder?path={path}	90
3.3.5.5.1	POST	90
3.3.5.5.1.1	Request Body	91
3.3.5.5.1.2	Response Body	91
3.3.5.5.1.3	Processing Details	91
3.3.5.6	item/rename?path={path}&newname={newname}	91
3.3.5.6.1	POST	91
3.3.5.6.1.1	Request Body	92
3.3.5.6.1.2	Response Body	92
3.3.5.6.1.3	Processing Details	92
3.3.5.7	item/delete?path={path}	92
3.3.5.7.1	POST	92
3.3.5.7.1.1	Request Body	93
3.3.5.7.1.2	Response Body	93
3.3.5.7.1.3	Processing Details	93
3.3.5.8	accessuri?path={path}	93
3.3.5.8.1	GET	93
3.3.5.8.1.1	Request Body	94
3.3.5.8.1.2	Response Body	94
3.3.5.8.1.3	Processing Details	94
3.3.5.9	item/move?newpath={newpath}&originalpath={originalpath}	94
3.3.5.9.1	POST	95
3.3.5.9.1.1	Request Body	95
3.3.5.9.1.2	Response Body	95
3.3.5.9.1.3	Processing Details	95
3.3.5.10	item/copy?newpath={newpath}&originalpath={originalpath}	96
3.3.5.10.1	POST	96
3.3.5.10.1.1	Request Body	96
3.3.5.10.1.2	Response Body	97
3.3.5.10.1.3	Processing Details	97
3.3.6	Timer Events	97
3.3.7	Other Local Events	97
3.4	IAzureADManagement Server Details	97
3.4.1	Abstract Data Model	97
3.4.1.1	MSOUser	97
3.4.1.2	MSOTenantInformation	97
3.4.1.3	ArrayOfMSODomain	97
3.4.2	Timers	97
3.4.3	Initialization	97
3.4.4	Higher-Layer Triggered Events	97
3.4.5	Message Processing Events and Sequencing Rules	98
3.4.5.1	msouser?localusername={localusername}	99
3.4.5.1.1	GET	99
3.4.5.1.1.1	Request Body	99
3.4.5.1.1.2	Response Body	100
3.4.5.1.1.3	Processing Details	100
3.4.5.2	msousers	100
3.4.5.2.1	GET	100
3.4.5.2.1.1	Request Body	101
3.4.5.2.1.2	Response Body	101
3.4.5.2.1.3	Processing Details	101



3.4.5.3	msouser/create?localusername={localusername}&onlineusername={onlineusername}.....	101
3.4.5.3.1	POST .....	101
3.4.5.3.1.1	Request Body .....	102
3.4.5.3.1.2	Response Body .....	102
3.4.5.3.1.3	Processing Details.....	102
3.4.5.4	msouser/assign?localusername={localusername}&onlineusername={onlineusername}.....	102
3.4.5.4.1	POST .....	103
3.4.5.4.1.1	Request Body .....	103
3.4.5.4.1.2	Response Body .....	103
3.4.5.4.1.3	Processing Details.....	104
3.4.5.5	msouser/unassign?localusername={localusername} .....	104
3.4.5.5.1	POST .....	104
3.4.5.5.1.1	Request Body .....	105
3.4.5.5.1.2	Response Body .....	105
3.4.5.5.1.3	Processing Details.....	105
3.4.5.6	msouser/enable?localusername={localusername} .....	105
3.4.5.6.1	POST .....	105
3.4.5.6.1.1	Request Body .....	106
3.4.5.6.1.2	Response Body .....	106
3.4.5.6.1.3	Processing Details.....	106
3.4.5.7	msouser/disable?localusername={localusername}.....	106
3.4.5.7.1	POST .....	106
3.4.5.7.1.1	Request Body .....	107
3.4.5.7.1.2	Response Body .....	107
3.4.5.7.1.3	Processing Details.....	107
3.4.5.8	msouser/delete?localusername={localusername} .....	107
3.4.5.8.1	POST .....	108
3.4.5.8.1.1	Request Body .....	108
3.4.5.8.1.2	Response Body .....	108
3.4.5.8.1.3	Processing Details.....	108
3.4.5.9	msodomains.....	109
3.4.5.9.1	GET .....	109
3.4.5.9.1.1	Request Body .....	109
3.4.5.9.1.2	Response Body .....	109
3.4.5.9.1.3	Processing Details.....	110
3.4.5.10	msosubscriptioninfo .....	110
3.4.5.10.1	GET .....	110
3.4.5.10.1.1	Request Body .....	111
3.4.5.10.1.2	Response Body .....	111
3.4.5.10.1.3	Processing Details.....	111
3.4.5.11	msolicense/set?localusername={localusername} .....	111
3.4.5.11.1	POST .....	111
3.4.5.11.1.1	Request Body .....	112
3.4.5.11.1.2	Response Body .....	112
3.4.5.11.1.3	Processing Details.....	112
3.4.6	Timer Events.....	112
3.4.7	Other Local Events.....	112
3.5	IMailboxManagement Server Details .....	113
3.5.1	Abstract Data Model.....	113
3.5.1.1	Mailbox.....	113
3.5.1.2	ArrayOfMailbox .....	113
3.5.2	Timers .....	113
3.5.3	Initialization.....	113
3.5.4	Higher-Layer Triggered Events .....	113

3.5.5	Message Processing Events and Sequencing Rules .....	113
3.5.5.1	mailbox?username={username} .....	114
3.5.5.1.1	GET .....	114
3.5.5.1.1.1	Request Body .....	115
3.5.5.1.1.2	Response Body .....	115
3.5.5.1.1.3	Processing Details .....	115
3.5.5.2	mailbox/create?username={username}&email={email} .....	115
3.5.5.2.1	POST .....	115
3.5.5.2.1.1	Request Body .....	116
3.5.5.2.1.2	Response Body .....	116
3.5.5.2.1.3	Processing Details .....	116
3.5.5.3	mailbox/set?username={username}&email={email} .....	116
3.5.5.3.1	POST .....	117
3.5.5.3.1.1	Request Body .....	117
3.5.5.3.1.2	Response Body .....	117
3.5.5.3.1.3	Processing Details .....	117
3.5.5.4	mailbox/unset?username={username} .....	118
3.5.5.4.1	POST .....	118
3.5.5.4.1.1	Request Body .....	118
3.5.5.4.1.2	Response Body .....	119
3.5.5.4.1.3	Processing Details .....	119
3.5.5.5	mailbox/disable?username={username} .....	119
3.5.5.5.1	POST .....	119
3.5.5.5.1.1	Request Body .....	120
3.5.5.5.1.2	Response Body .....	120
3.5.5.5.1.3	Processing Details .....	120
3.5.5.6	mailbox/delete?username={username} .....	120
3.5.5.6.1	POST .....	120
3.5.5.6.1.1	Request Body .....	121
3.5.5.6.1.2	Response Body .....	121
3.5.5.6.1.3	Processing Details .....	121
3.5.5.7	mailbox/enable?username={username} .....	121
3.5.5.7.1	POST .....	121
3.5.5.7.1.1	Request Body .....	122
3.5.5.7.1.2	Response Body .....	122
3.5.5.7.1.3	Processing Details .....	122
3.5.5.8	mailboxes .....	122
3.5.5.8.1	GET .....	122
3.5.5.8.1.1	Request Body .....	123
3.5.5.8.1.2	Response Body .....	123
3.5.5.8.1.3	Processing Details .....	123
3.5.5.9	domains .....	123
3.5.5.9.1	GET .....	124
3.5.5.9.1.1	Request Body .....	124
3.5.5.9.1.2	Response Body .....	124
3.5.5.9.1.3	Processing Details .....	124
3.5.5.10	mailbox/getemailaddresses?username={username} .....	125
3.5.5.10.1	GET .....	125
3.5.5.10.1.1	Request Body .....	125
3.5.5.10.1.2	Response Body .....	126
3.5.5.10.1.3	Processing Details .....	126
3.5.5.11	mailbox/setemailaddresses?username={username}&emails={emails} .....	126
3.5.5.11.1	POST .....	126
3.5.5.11.1.1	Request Body .....	127
3.5.5.11.1.2	Response Body .....	127
3.5.5.11.1.3	Processing Details .....	127
3.5.6	Timer Events .....	127
3.5.7	Other Local Events .....	127

3.6	IAlertManagement Server Details .....	127
3.6.1	Abstract Data Model.....	127
3.6.1.1	AlertInfo .....	127
3.6.2	Timers .....	128
3.6.3	Initialization.....	128
3.6.4	Higher-Layer Triggered Events .....	128
3.6.5	Message Processing Events and Sequencing Rules .....	128
3.6.5.1	alerts/index/{startingindex}/count/{amount} .....	128
3.6.5.1.1	GET .....	128
3.6.5.1.1.1	Request Body .....	129
3.6.5.1.1.2	Response Body .....	129
3.6.5.1.1.3	Processing Details.....	129
3.6.5.2	alert/enable?alertkey={alertkey} .....	130
3.6.5.2.1	POST .....	130
3.6.5.2.1.1	Request Body .....	130
3.6.5.2.1.2	Response Body .....	130
3.6.5.2.1.3	Processing Details.....	131
3.6.5.3	alert/disable?alertkey={alertkey} .....	131
3.6.5.3.1	POST .....	131
3.6.5.3.1.1	Request Body .....	131
3.6.5.3.1.2	Response Body .....	132
3.6.5.3.1.3	Processing Details.....	132
3.6.5.4	alert/clear?alertkey={alertkey} .....	132
3.6.5.4.1	POST .....	132
3.6.5.4.1.1	Request Body .....	133
3.6.5.4.1.2	Response Body .....	133
3.6.5.4.1.3	Processing Details.....	133
3.6.5.5	alert/repair?alertkey={alertkey}.....	133
3.6.5.5.1	POST .....	133
3.6.5.5.1.1	Request Body .....	134
3.6.5.5.1.2	Response Body .....	134
3.6.5.5.1.3	Processing Details.....	134
3.6.6	Timer Events.....	134
3.6.7	Other Local Events.....	134
3.7	IDeviceManagement Server Details .....	134
3.7.1	Abstract Data Model.....	134
3.7.1.1	DeviceInfo .....	134
3.7.2	Timers .....	134
3.7.3	Initialization.....	134
3.7.4	Higher-Layer Triggered Events .....	134
3.7.5	Message Processing Events and Sequencing Rules .....	135
3.7.5.1	devices/index/{startingindex}/count/{amount}.....	135
3.7.5.1.1	GET .....	135
3.7.5.1.1.1	Request Body .....	136
3.7.5.1.1.2	Response Body .....	136
3.7.5.1.1.3	Processing Details.....	136
3.7.5.2	device/{deviceid}/startbackup .....	136
3.7.5.2.1	POST .....	137
3.7.5.2.1.1	Request Body .....	137
3.7.5.2.1.2	Response Body .....	137
3.7.5.2.1.3	Processing Details.....	137
3.7.5.3	device/{deviceid}/stopbackup .....	138
3.7.5.3.1	POST .....	138
3.7.5.3.1.1	Request Body .....	138
3.7.5.3.1.2	Response Body .....	138
3.7.5.3.1.3	Processing Details.....	139
3.7.6	Timer Events.....	139
3.7.7	Other Local Events.....	139

3.8	IServiceManagement Server Details .....	139
3.8.1	Abstract Data Model.....	139
3.8.1.1	WebApiProvider .....	139
3.8.2	Timers .....	139
3.8.3	Initialization.....	139
3.8.4	Higher-Layer Triggered Events .....	139
3.8.5	Message Processing Events and Sequencing Rules .....	139
3.8.5.1	permittedbuiltinservices.....	140
3.8.5.1.1	GET .....	140
3.8.5.1.1.1	Request Body .....	140
3.8.5.1.1.2	Response Body .....	140
3.8.5.1.1.3	Processing Details.....	141
3.8.6	Timer Events.....	141
3.8.7	Other Local Events.....	141
3.9	IServerManagement Server Details .....	141
3.9.1	Abstract Data Model.....	141
3.9.1.1	WebApiProvider .....	141
3.9.2	Timers .....	141
3.9.3	Initialization.....	141
3.9.4	Higher-Layer Triggered Events .....	141
3.9.5	Message Processing Events and Sequencing Rules .....	141
3.9.5.1	serverinformation .....	142
3.9.5.1.1	GET .....	142
3.9.5.1.1.1	Request Body.....	142
3.9.5.1.1.2	Response Body.....	143
3.9.5.1.1.3	Processing Details.....	143
3.9.6	Timer Events.....	143
3.9.7	Other Local Events.....	143
3.10	ICustomizationManagement Server Details.....	143
3.10.1	Abstract Data Model.....	143
3.10.1.1	CustomizationInfo .....	143
3.10.2	Timers .....	143
3.10.3	Initialization.....	143
3.10.4	Higher-Layer Triggered Events .....	143
3.10.5	Message Processing Events and Sequencing Rules .....	144
3.10.5.1	customizationinformation .....	144
3.10.5.1.1	GET .....	144
3.10.5.1.1.1	Request Body.....	144
3.10.5.1.1.2	Response Body.....	144
3.10.5.1.1.3	Processing Details.....	145
3.10.6	Timer Events.....	145
3.10.7	Other Local Events.....	145
3.11	IMediaManagement Server Details .....	145
3.11.1	Abstract Data Model.....	145
3.11.2	Timers .....	145
3.11.3	Initialization.....	145
3.11.4	Higher-Layer Triggered Events .....	145
3.11.5	Message Processing Events and Sequencing Rules .....	145
3.11.5.1	metadata/item/mediatype/{mediatypedata}/groupingmethod/{groupingmetho ddata}/sortproperty/{sortpropertydata}/sortorder/{sortorderdata}/index/{star tingindexdata}/count/{requestedcountdata}?grouping={groupingdata}.....	148
3.11.5.1.1	GET .....	148
3.11.5.1.1.1	Request Body .....	149
3.11.5.1.1.2	Response Body .....	149
3.11.5.1.1.3	Processing Details.....	149

3.11.5.2	metadata/container/mediatype/{mediatypedata}/groupingmethod/{groupingmethoddata}?path={containerpath}.....	149
3.11.5.2.1	GET .....	149
3.11.5.2.1.1	Request Body .....	150
3.11.5.2.1.2	Response Body .....	150
3.11.5.2.1.3	Processing Details.....	150
3.11.5.3	search/mediatype/{mediatypedata}/groupingmethod/{groupingmethoddata}/index/{startingindexdata}/count/{requestedcountdata}?keywords={keywords}&grouping={groupingdata}&titleonly={titleonlydata}&shallow={shallowdata}.....	151
3.11.5.3.1	GET .....	151
3.11.5.3.1.1	Request Body .....	152
3.11.5.3.1.2	Response Body .....	152
3.11.5.3.1.3	Processing Details.....	152
3.11.5.4	photo/{id}/isoriginal/{isoriginaldata}/width/{widthdata}/height/{heightdata}.....	152
3.11.5.4.1	GET .....	153
3.11.5.4.1.1	Request Body .....	153
3.11.5.4.1.2	Response Body .....	153
3.11.5.4.1.3	Processing Details.....	153
3.11.5.5	thumbnail/mediatype/{mediatypedata}/id/{id}/width/{widthdata}/height/{heightdata}.....	154
3.11.5.5.1	GET .....	154
3.11.5.5.1.1	Request Body .....	155
3.11.5.5.1.2	Response Body .....	155
3.11.5.5.1.3	Processing Details.....	155
3.11.5.6	testdata?sizedata={sizedata}.....	155
3.11.5.6.1	POST .....	155
3.11.5.6.1.1	Request Body .....	156
3.11.5.6.1.2	Response Body .....	156
3.11.5.6.1.3	Processing Details.....	156
3.11.5.7	metadata/streams/mediatype/{mediatypedata}/id/{id}/index/{startingindexdata}/count/{requestedcountdata} .....	156
3.11.5.7.1	GET .....	156
3.11.5.7.1.1	Request Body .....	157
3.11.5.7.1.2	Response Body .....	157
3.11.5.7.1.3	Processing Details.....	157
3.11.6	Timer Events.....	158
3.11.7	Other Local Events.....	158
3.12	IStorageManagement Server Details.....	158
3.12.1	Abstract Data Model.....	158
3.12.1.1	FolderInfo .....	158
3.12.1.2	StorageDriveInfo .....	158
3.12.1.3	StorageServerInfo.....	158
3.12.2	Timers .....	158
3.12.3	Initialization.....	158
3.12.4	Higher-Layer Triggered Events .....	158
3.12.5	Message Processing Events and Sequencing Rules .....	159
3.12.5.1	servers .....	160
3.12.5.1.1	GET .....	161
3.12.5.1.1.1	Request Body .....	161
3.12.5.1.1.2	Response Body .....	161
3.12.5.1.1.3	Processing Details.....	161
3.12.5.2	storagedriveinfo?servername={servername} .....	162

3.12.5.2.1	GET .....	162
3.12.5.2.1.1	Request Body .....	162
3.12.5.2.1.2	Response Body .....	162
3.12.5.2.1.3	Processing Details .....	163
3.12.5.3	serverdrives/index/{index}/count/{count} .....	163
3.12.5.3.1	GET .....	163
3.12.5.3.1.1	Request Body .....	164
3.12.5.3.1.2	Response Body .....	164
3.12.5.3.1.3	Processing Details .....	164
3.12.5.4	serverfolders/index/{index}/count/{count}?username={username} .....	164
3.12.5.4.1	GET .....	164
3.12.5.4.1.1	Request Body .....	165
3.12.5.4.1.2	Response Body .....	165
3.12.5.4.1.3	Processing Details .....	165
3.12.5.5	serverfolder/create/overwritepermissions/{overwritepermissions}?sharename={sharename}&path={path}&description={description}&servername={servername} .....	165
3.12.5.5.1	POST .....	166
3.12.5.5.1.1	Request Body .....	167
3.12.5.5.1.2	Response Body .....	167
3.12.5.5.1.3	Processing Details .....	167
3.12.5.6	serverfolder/{id}/delete/deletecontents/{deletecontents} .....	167
3.12.5.6.1	POST .....	167
3.12.5.6.1.1	Request Body .....	168
3.12.5.6.1.2	Response Body .....	168
3.12.5.6.1.3	Processing Details .....	168
3.12.5.7	serverfolder/{id}/rename?newname={newname} .....	168
3.12.5.7.1	POST .....	168
3.12.5.7.1.1	Request Body .....	169
3.12.5.7.1.2	Response Body .....	169
3.12.5.7.1.3	Processing Details .....	169
3.12.5.8	serverfolder/{id}/modify/permission/{permission}?username={username}&name={name}&description={description} .....	169
3.12.5.8.1	POST .....	170
3.12.5.8.1.1	Request Body .....	170
3.12.5.8.1.2	Response Body .....	170
3.12.5.8.1.3	Processing Details .....	170
3.12.5.9	serverfolder/{id}/modify/usersid/{usersid}/permission/{permission}?name={name}&description={description} .....	171
3.12.5.9.1	POST .....	171
3.12.5.9.1.1	Request Body .....	172
3.12.5.9.1.2	Response Body .....	172
3.12.5.9.1.3	Processing Details .....	172
3.12.6	Timer Events .....	172
3.12.7	Other Local Events .....	172
3.13	IUserManagement Server Details .....	172
3.13.1	Abstract Data Model .....	172
3.13.1.1	UserInfo .....	172
3.13.2	Timers .....	172
3.13.3	Initialization .....	172
3.13.4	Higher-Layer Triggered Events .....	173
3.13.5	Message Processing Events and Sequencing Rules .....	173
3.13.5.1	users/index/{startingindex}/count/{amount} .....	174
3.13.5.1.1	GET .....	175
3.13.5.1.1.1	Request Body .....	175

3.13.5.1.1.2	Response Body .....	175
3.13.5.1.1.3	Processing Details .....	176
3.13.5.2	usergroups .....	176
3.13.5.2.1	GET .....	176
3.13.5.2.1.1	Request Body .....	177
3.13.5.2.1.2	Response Body .....	177
3.13.5.2.1.3	Processing Details .....	177
3.13.5.3	users/connection/index/{startingindex}/count/{amount}?username={username}	
	e} .....	177
3.13.5.3.1	GET .....	177
3.13.5.3.1.1	Request Body .....	178
3.13.5.3.1.2	Response Body .....	178
3.13.5.3.1.3	Processing Details .....	178
3.13.5.4	user/{userid}/enable .....	178
3.13.5.4.1	POST .....	178
3.13.5.4.1.1	Request Body .....	179
3.13.5.4.1.2	Response Body .....	179
3.13.5.4.1.3	Processing Details .....	179
3.13.5.5	user/{userid}/disable .....	179
3.13.5.5.1	POST .....	180
3.13.5.5.1.1	Request Body .....	180
3.13.5.5.1.2	Response Body .....	180
3.13.5.5.1.3	Processing Details .....	180
3.13.5.6	user/setpassword/{userid} .....	180
3.13.5.6.1	POST .....	181
3.13.5.6.1.1	Request Body .....	181
3.13.5.6.1.2	Response Body .....	181
3.13.5.6.1.3	Processing Details .....	182
3.13.5.7	user/{id}/update?firstname={firstname}&lastname={lastname}&remotewebac	
	cess={remotewebaccess}&vpnaccess={vpnaccess} .....	182
3.13.5.7.1	POST .....	182
3.13.5.7.1.1	Request Body .....	183
3.13.5.7.1.2	Response Body .....	183
3.13.5.7.1.3	Processing Details .....	183
3.13.5.8	usergroup/{groupguid}/addusers .....	183
3.13.5.8.1	POST .....	183
3.13.5.8.1.1	Request Body .....	184
3.13.5.8.1.2	Response Body .....	184
3.13.5.8.1.3	Processing Details .....	184
3.13.5.9	usergroup/{groupguid}/removeusers .....	184
3.13.5.9.1	POST .....	184
3.13.5.9.1.1	Request Body .....	185
3.13.5.9.1.2	Response Body .....	185
3.13.5.9.1.3	Processing Details .....	185
3.13.5.10	user/add?username={username}&firstname={firstname}&lastname={lastnam	
	e}&remotewebaccess={remotewebaccess}&vpnaccess={vpnaccess}&usergrou	
	ps={usergroups} .....	185
3.13.5.10.1	POST .....	186
3.13.5.10.1.1	Request Body .....	186
3.13.5.10.1.2	Response Body .....	186
3.13.5.10.1.3	Processing Details .....	187
3.13.5.11	user/{id}/delete .....	187
3.13.5.11.1	POST .....	187
3.13.5.11.1.1	Request Body .....	188
3.13.5.11.1.2	Response Body .....	188

3.13.5.11.1.3	Processing Details .....	188
3.13.6	Timer Events.....	188
3.13.7	Other Local Events.....	188
3.14	ISharePointSiteMgmt Server Details .....	188
3.14.1	Abstract Data Model.....	188
3.14.2	Timers .....	188
3.14.3	Initialization .....	188
3.14.4	Higher-Layer Triggered Events .....	188
3.14.5	Message Processing Events and Sequencing Rules .....	188
3.14.5.1	site .....	188
3.14.5.1.1	GET .....	189
3.14.5.1.1.1	Request Body .....	189
3.14.5.1.1.2	Response Body .....	189
3.14.5.1.1.3	Processing Details.....	189
3.14.6	Timer Events.....	189
3.14.7	Other Local Events.....	189
3.15	IWindowsPhoneManagement Server Details.....	190
3.15.1	Abstract Data Model.....	190
3.15.2	Timers .....	190
3.15.3	Initialization.....	190
3.15.4	Higher-Layer Triggered Events .....	190
3.15.5	Message Processing Events and Sequencing Rules .....	190
3.15.5.1	/notification/subscribe?deviceid={deviceid} .....	190
3.15.5.1.1	POST .....	190
3.15.5.1.1.1	Request Body .....	191
3.15.5.1.1.2	Response Body .....	191
3.15.5.1.1.3	Processing Details.....	191
3.15.5.2	/notification/unsubscribe?deviceid={deviceid}.....	191
3.15.5.2.1	POST .....	192
3.15.5.2.1.1	Request Body .....	192
3.15.5.2.1.2	Response Body .....	192
3.15.5.2.1.3	Processing Details.....	192
3.15.6	Timer Events.....	193
3.15.7	Other Local Events.....	193
<b>4</b>	<b>Protocol Examples .....</b>	<b>194</b>
4.1	Login .....	194
4.2	Get Server Information .....	194
4.3	Get Server Folders .....	195
4.4	Retrieve the Metadata for Items within a Folder .....	196
4.5	Create a Folder .....	197
4.6	Upload a File .....	198
4.7	Logout .....	198
<b>5</b>	<b>Security .....</b>	<b>200</b>
5.1	Security Considerations for Implementers .....	200
5.2	Index of Security Parameters .....	200
<b>6</b>	<b>Appendix A: Full Xml Schema .....</b>	<b>201</b>
6.1	http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi Schema	201
6.2	http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.Storage Schema .....	210
6.3	http://schemas.microsoft.com/2003/10/Serialization/Arrays Schema .....	210
6.4	http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADO bjectModel Schema .....	210
6.5	http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel Schema	213
6.6	http://schemas.microsoft.com/2003/10/Serialization/ Schema .....	213



6.7	<a href="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.WebApi.Management.Storage">http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.WebApi.Management.Storage</a> Schema.....	213
6.8	<a href="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.O365Integration">http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.O365Integration</a> Schema .....	214
<b>7</b>	<b>Appendix B: Product Behavior .....</b>	<b>215</b>
<b>8</b>	<b>Change Tracking .....</b>	<b>216</b>
<b>9</b>	<b>Index.....</b>	<b>217</b>

# 1 Introduction

The Server and File Management Web APIs Protocol is designed to enable access to and management of the server. It also allows users to access files through Web APIs over the Internet or intranet.

Sections [1.5](#), [1.8](#), [1.9](#), [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~~This document uses the following terms ~~are specific to this document:~~

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

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

**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).

**Hypertext Transfer Protocol ~~over~~ Secure Sockets Layer (HTTPS):** An extension of HTTP that securely encrypts and decrypts ~~webpageweb page~~ requests. In some older protocols, "Hypertext Transfer Protocol over Secure Sockets Layer" is still used (Secure Sockets Layer has been deprecated). For more information, see [SSL3] and [RFC5246].

**Secure Sockets Layer (SSL):** A security protocol that supports confidentiality and integrity of messages in client and server applications that communicate over open networks. SSL uses two keys to encrypt data—a public key known to everyone and a private or secret key known only to the recipient of the message. SSL supports server and, optionally, client authentication (2) using X.509 certificates (2). For more information, see [X509]. The SSL protocol is precursor to Transport Layer Security (TLS). The TLS version 1.0 specification is based on SSL version 3.0 ~~7~~ [\[SSL3\]](#).

**security identifier (SID):** An identifier for security principals in Windows that is used to identify an account or a group. Conceptually, the **SID** is composed of an account authority portion (typically a domain) and a smaller integer representing an identity relative to the account authority, termed the relative identifier (RID). The **SID** format is specified in [MS-DTYP] section 2.4.2; a string representation of **SIDs** is specified in [MS-DTYP] section 2.4.2 and [MS-AZOD] section 1.1.1.2.

**Stock Keeping Unit (SKU):** A unique code that refers to a particular manufactured object or source of revenue. A **SKU** can refer to a retail product (software in a box that is sold through a channel), a subscription program (such as MSDN), or an online service (such as MSN).

**Uniform Resource Identifier (URI):** A string that identifies a resource. The URI is an addressing mechanism defined in Internet Engineering Task Force (IETF) Uniform Resource Identifier (URI): Generic Syntax [RFC3986].

**user principal name (UPN):** A user account name (sometimes referred to as the user logon name) and a domain name that identifies the domain in which the user account is located. This is the standard usage for logging on to a Windows domain. The format is: someone@example.com (in the form of an email address). In Active Directory, the userPrincipalName attribute (2) of the account object, as described in [MS-ADTS].

**XML:** The Extensible Markup Language, as described in [XML1.0].

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

## 1.2 References

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

### 1.2.1 Normative References

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

[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>

[RFC2616] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, <http://www.rfc-editor.org/rfc/rfc2616.txt>

[RFC4346] Dierks, T., and Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006, <http://www.ietf.org/rfc/rfc4346.txt>

[WSDL] Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S., "Web Services Description Language (WSDL) 1.1", W3C Note, March 2001, <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>

[XMLNS] Bray, T., Hollander, D., Layman, A., et al., Eds., "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation, December 2009, <http://www.w3.org/TR/2009/REC-xml-names-20091208/>

[XMLSCHEMA1] Thompson, H., Beech, D., Maloney, M., and Mendelsohn, N., Eds., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>

[XMLSCHEMA2] Biron, P.V., Ed. and Malhotra, A., Ed., "XML Schema Part 2: Datatypes", W3C Recommendation, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>

### 1.2.2 Informative References

None.

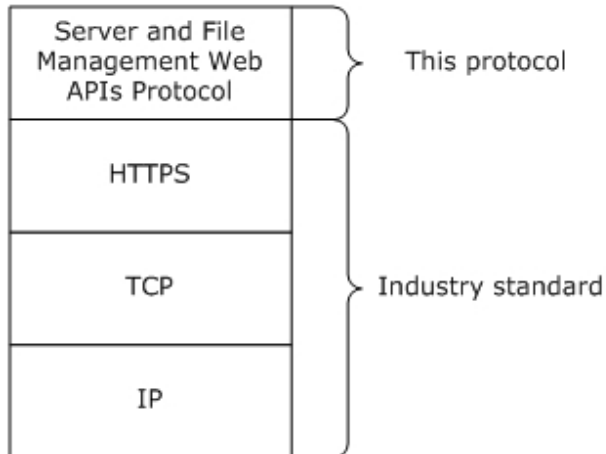
## 1.3 Overview

The Server and File Management Web APIs Protocol is used to access a REST-based server and for file management over the HTTPS transports.

The protocol exposes a set of built-in web services for third-party developers to build applications on different devices that can access files and manage the server remotely. The protocol also allows third-party developers to add their own web services without the need to handle authentication.

#### 1.4 Relationship to Other Protocols

The following figure ~~illustrates~~shows the relationship of this protocol to industry-standard protocols.



**Figure 1: Relationship of Server and File Management Web APIs Protocol to industry-standard protocols**

#### 1.5 Prerequisites/Preconditions

All web services that are exposed in this protocol are hosted in Internet Information Services (IIS) 7.0 so that a user can call web services by using the **Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)** protocol. **Secure Sockets Layer (SSL)** is required for secure communication.

#### 1.6 Applicability Statement

This protocol defines a set of server and file management REST APIs. This protocol is applicable to both Internet and intranet client-server scenarios.

#### 1.7 Versioning and Capability Negotiation

This protocol does not provide any mechanism for capability negotiation.

#### 1.8 Vendor-Extensible Fields

None.

#### 1.9 Standards Assignments

None.

## 2 Messages

### 2.1 Transport

This protocol consists of a set of RESTful (representational state transfer) web services.

HTTPS over TCP/IP, as specified in [RFC2616].

All client messages to the server MUST use HTTPS.

Protocol messages MUST be formatted as specified either in **XML** or in Javascript Object Notation (JSON). Protocol server faults MUST be returned by using HTTP status codes as specified in [RFC2616], section 10, "Status Code Definitions".

### 2.2 Common Data Types

This section contains common definitions that are used by this protocol. The syntax of the definitions uses an XML Schema, as specified in [XMLSCHEMA1] and [XMLSCHEMA2], and WSDL, as specified in [WSDL].

#### 2.2.1 Namespaces

This specification defines and references various XML namespaces by using the mechanisms specified in [XMLNS]. Although this specification associates a specific XML namespace prefix for each XML namespace that is used, the choice of any particular XML namespace prefix is implementation-specific and not significant for interoperability.

Prefix	NameSpaces URI	Reference
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA A]
tns1	http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi	
tns2	http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.Storage	
tns4	http://schemas.microsoft.com/2003/10/Serialization/Arrays	
tns5	http://schemas.microsoft.com/2003/10/Serialization/	
tns6	http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObjectModel	
tns7	http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel	
tns8	http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.WebApi.Management.Storage	
tns9	http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.O365Integration	

#### 2.2.2 HTTP Headers

The following table summarizes the set of HTTP headers defined by this protocol.

The client MUST either pass Canary header or pass AppName, AppPublisherName, AppVersion, and Authorization to a server when the client calls an API that needs authentication information.

Header	Description
Accept	Specifies the acceptable data format for the response. See section 2.2.2.6
AppName	Defines the client application name. See section 2.2.2.1
AppPublisherName	Defines the client application publisher name. See section 2.2.2.2
AppVersion	Defines the client application version. See section 2.2.2.3
Authorization	Defines the user authorization. See section 2.2.2.4
Canary	Defines a user token returned in the Login response header. See section 2.2.2.5
Range	Defines the client request only part of an entity. Bytes are numbered from 0. See section 2.2.2.7.

### 2.2.2.1 AppName

The AppName header defines the client application name.

```
String = *(%x20-7E)
AppName = String
```

### 2.2.2.2 AppPublisherName

The AppPublisherName header defines the client application publisher name.

```
String = *(%x20-7E)
AppPublisherName = String
```

### 2.2.2.3 AppVersion

The AppVersion header defines the client application version.

```
String = *(%x20-7E)
AppVersion = String
```

### 2.2.2.4 Authorization

The Authorization header specifies the user authorization credentials. The client sends requests with an authorization header, the value of which starts with "Basic", followed by a blank space and a **base64** encoded string that represents the user name and password separated by a colon.

```
String = *(%x20-7E)
Authorization = String
```

### 2.2.2.5 Canary

The Canary header is a user token that is returned in the Login response header. The client MUST either pass this header or pass AppName, AppPublisherName, AppVersion, and Authorization to the server.

```
String = *(%x20-7E)
Canary = String
```

### 2.2.2.6 Accept

The Accept request-header field defines the data format that is acceptable for the response.

The format of the Accept header is as follows:

Accept: "application/json" / "application/xml"

```
String = *(%x20-7E)
Accept = String
```

### 2.2.2.7 Range

Defines the client request only part of an entity. Bytes are numbered from 0.

This is a standard http header defined in [RFC2616], section 14.35.1, "Byte Ranges"

```
String = *(%x20-7E)
Range = String
```

## 2.2.3 Common URI Parameters

The following table summarizes the set of Common **URI** parameters defined by this protocol.

URI parameters are not case sensitive.

URI parameter	Description
alertkey	See section 2.2.3.1
amount	See section 2.2.3.2
ascending	See section 2.2.3.3
containerpath	See section 2.2.3.4
count	See section 2.2.3.5
deletecontents	See section 2.2.3.6
description	See section 2.2.3.7
deviceid	See section 2.2.3.8
email	See section 2.2.3.9

<b>URI parameter</b>	<b>Description</b>
emails	See section 2.2.3.10
filter	See section 2.2.3.11
firstname	See section 2.2.3.12
groupguid	See section 2.2.3.13
groupingdata	See section 2.2.3.14
groupingmethoddata	See section 2.2.3.15
heightdata	See section 2.2.3.16
id	See section 2.2.3.17
index	See section 2.2.3.18
isoriginaldata	See section 2.2.3.19
keywords	See section 2.2.3.20
language	See section 2.2.3.21
lastname	See section 2.2.3.22
localusername	See section 2.2.3.23
mediatypedata	See section 2.2.3.24
name	See section 2.2.3.25
newname	See section 2.2.3.26
newpath	See section 2.2.3.27
onlineusername	See section 2.2.3.28
originalpath	See section 2.2.3.29
overwrite	See section 2.2.3.30
overwritepermissions	See section 2.2.3.31
path	See section 2.2.3.32
permission	See section 2.2.3.33
query	See section 2.2.3.34
remotewebaccess	See section 2.2.3.35
requestedcountdata	See section 2.2.3.36
scope	See section 2.2.3.37
servername	See section 2.2.3.38
shallowdata	See section 2.2.3.39
sharename	See section 2.2.3.40
sizedata	See section 2.2.3.41



URI parameter	Description
sortbyfield	See section 2.2.3.42
sortorderdata	See section 2.2.3.43
sortpropertydata	See section 2.2.3.44
startingindex	See section 2.2.3.45
startingindexdata	See section 2.2.3.46
timeoutseconds	See section 2.2.3.47
titleonlydata	See section 2.2.3.48
usergroups	See section 2.2.3.49
userid	See section 2.2.3.50
username	See section 2.2.3.51
usersid	See section 2.2.3.52
vpnaccess	See section 2.2.3.53
widthdata	See section 2.2.3.54

### 2.2.3.1 alertkey

The key of an Alert instance. This parameter MUST be included in alert instance operation.

```
String = *(%x20-7E)
alertkey = String
```

### 2.2.3.2 amount

The *amount* parameter defines the number of items to be retrieved. It SHOULD be a string that can be converted to a positive integer.

```
String = *(%x20-7E)
amount = String
```

### 2.2.3.3 ascending

The *ascending* parameter is a Boolean value that indicates whether the sorting is ascending.

```
String = *(%x20-7E)
ascending = String
```

### 2.2.3.4 containerpath

The *containerpath* parameter describes the path to the container. To specify the root container, leave this parameter empty.

```
String = *(%x20-7E)
containerpath = String
```

### 2.2.3.5 count

The *count* parameter describes the number of items to be retrieved.

```
String = *(%x20-7E)
count = String
```

### 2.2.3.6 deletecontents

This is a flag that indicates whether the contents of the shared folder have been deleted when the folder is no longer shared.

Set the flag to TRUE to end sharing the folder and to delete the folder contents. Set the flag to FALSE to end sharing the folder without deleting its contents.

```
String = *(%x20-7E)
deletecontents = String
```

### 2.2.3.7 description

The *description* parameter defines user-friendly description information.

```
String = *(%x20-7E)
description = String
```

### 2.2.3.8 deviceid

The *deviceid* parameter defines the ID of device. It SHOULD be the **security identifier (SID)** of the device.

```
String = *(%x20-7E)
deviceid = String
```

### 2.2.3.9 email

The *email* parameter defines the user's email address.

```
String = *(%x20-7E)
email = String
```

### 2.2.3.10 emails

This parameter defines an array of email addresses.

```
String = *(%x20-7E)
emails = String
```

### 2.2.3.11 filter

This parameter defines the type of the files to be retrieved. It MUST be All, Multimedia, or Document.

```
String = *(%x20-7E)
filter = String
```

### 2.2.3.12 firstname

Defines the first name of the user.

```
String = *(%x20-7E)
firstname = String
```

### 2.2.3.13 groupguid

Defines the he **GUID** of the Group, which is specified in section 2.2.5.2.

```
String = *(%x20-7E)
groupguid = String
```

### 2.2.3.14 groupingdata

The *groupingdata* parameter describes the container path. To specify the root container, leave this parameter empty.

```
String = *(%x20-7E)
groupingdata = String
```

### 2.2.3.15 groupingmethoddata

The *groupingmethoddata* parameter describes the grouping method. Choose from one of the following values.

Grouping Method Value	Applicable Media Types	Meaning
album	music	Group by album.

Grouping Method Value	Applicable Media Types	Meaning
albumartist	music	Group by album artist.
all	music, photo, or video	Group by media type.
artist	music	Group by artist.
date	photo	Group by date.
folder	photo or video	Group by folder.
genre	music	Group by musical genre (style).
playlist	music	Group by playlist.
rating	music	Group by rating.

```
String = *(%x20-7E)
groupingmethoddata = String
```

### 2.2.3.16 heightdata

The *heightdata* parameter describes the height of an image.

```
String = *(%x20-7E)
heightdata = String
```

### 2.2.3.17 id

The *id* parameter describes the identifier of the item.

```
String = *(%x20-7E)
id = String
```

### 2.2.3.18 index

The *index* parameter defines the numerical position of the first item to be retrieved.

```
String = *(%x20-7E)
index = String
```

### 2.2.3.19 isoriginaldata

The *isoriginaldata* parameter describes whether to return the original data. If this parameter is set, the *widthdata* and *heightdata* parameters will not take effect, although they are still required in the request.

```
String = *(%x20-7E)
isoriginaldata = String
```

### 2.2.3.20 keywords

The *keywords* parameter describes a space-separated list of keywords to search for.

```
String = *(%x20-7E)
keywords = String
```

### 2.2.3.21 language

This parameter defines the language/region code (such as en-us) of the new folder name.

```
String = *(%x20-7E)
language = String
```

### 2.2.3.22 lastname

This parameter defines the last name of the user.

```
String = *(%x20-7E)
lastname = String
```

### 2.2.3.23 localusername

This parameter defines the local name of the user.

```
String = *(%x20-7E)
localusername = String
```

### 2.2.3.24 mediatypedata

The *mediatypedata* parameter describes the media type. Choose from one of the following values.

Media Type Value	Meaning
music	Audio clips.
photo	Photos.
video	Video clips.

```
String = *(%x20-7E)
mediatypedata = String
```

### 2.2.3.25 name

This parameter describes the name of item.

```
String = *(%x20-7E)
```

```
name = String
```

### **2.2.3.26 newname**

The *newname* parameter describes the new name of the item.

```
String = *(%x20-7E)
newname = String
```

### **2.2.3.27 newpath**

The *newpath* parameter describes the network path that begins with the server name, in the format /ServerName/FolderPath.

```
String = *(%x20-7E)
newpath = String
```

### **2.2.3.28 onlineusername**

This parameter defines the online user's name.

```
String = *(%x20-7E)
onlineusername = String
```

### **2.2.3.29 originalpath**

The *originalpath* parameter is a network path that begins with the server name, in the format /ServerName/FolderPath.

```
String = *(%x20-7E)
originalpath = String
```

### **2.2.3.30 overwrite**

The *overwrite* flag indicates whether to overwrite the item. It SHOULD be either TRUE or FALSE.

```
String = *(%x20-7E)
overwrite = String
```

### **2.2.3.31 overwritepermissions**

This is a flag that indicates whether to overwrite the descriptions on the server folder. It SHOULD be a string of TRUE or FALSE.

```
String = *(%x20-7E)
```

```
overwritepermissions = String
```

### 2.2.3.32 path

This parameter is the path of directory or file.

If it is a network path that begins with the server name, it is in this format:

```
\\ServerName\FolderPath
```

If it is a local path, the format is as follows:

```
C:\folderName\ItemName
```

```
String = *(%x20-7E)
path = String
```

### 2.2.3.33 permission

The *permissions* parameter defines the user access permissions on a shared folder.

The value SHOULD be a string that can be converted to one of following integers:

- 0: None, no access.
- 1: ReadOnly, read-only access.
- 2: Full, read/write access.
- 3: Other, unknown or unspecified access.

```
String = *(%x20-7E)
permission = String
```

### 2.2.3.34 query

This parameter defines the string to search for within the names of the files and folders.

```
String = *(%x20-7E)
query = String
```

### 2.2.3.35 remotewebaccess

This flag indicates whether to allow the user to have remote web access. It SHOULD be either TRUE (to allow the user to have remote web access) or FALSE.

```
String = *(%x20-7E)
remotewebaccess = String
```

### 2.2.3.36 requestedcountdata

The *requestedcountdata* parameter describes the requested number of media items to be retrieved.

```
String = *(%x20-7E)
requestedcountdata = String
```

### 2.2.3.37 scope

This is the UNC path of the folder in which the search is performed.

```
String = *(%x20-7E)
scope = String
```

### 2.2.3.38 servername

This parameter identifies the server name.

```
String = *(%x20-7E)
servername = String
```

### 2.2.3.39 shallowdata

The *shallowdata* parameter describes whether the search is shallow or deep.

```
String = *(%x20-7E)
shallowdata = String
```

### 2.2.3.40 sharename

This is the name of the shared folder. It cannot contain the following characters:

- < (less than)
- >(greater than)
- : (colon)
- " (double quote)
- / (forward slash)
- \ (backslash)
- | (vertical bar or pipe)
- ? (question mark)
- \* (asterisk)

```
String = *(%x20-7E)
```



```
sharename = String
```

### 2.2.3.41 sizedata

The *sizedata* parameter describes the size of the test data to be retrieved.

```
String = *(%x20-7E)  
sizedata = String
```

### 2.2.3.42 sortbyfield

This is the field of the file item property by which file items are sorted.

```
String = *(%x20-7E)  
sortbyfield = String
```

### 2.2.3.43 sortorderdata

The *sortorderdata* parameter describes the sort order. The following values are allowed.

Sort Order Value	Meaning
ASC	Sort in ascending order.
default	Sort in the default sort order.
DESC	Sort in descending order.

```
String = *(%x20-7E)  
sortorderdata = String
```

### 2.2.3.44 sortpropertydata

The *sortpropertydata* parameter describes the metadata property on which to sort the media items. The following values are allowed.

Metadata Property Type	Meaning
date	Sort by date.
default	Sort by the default metadata property type in the web service.
title	Sort by title.

```
String = *(%x20-7E)  
sortpropertydata = String
```

### 2.2.3.45 **startingindex**

The *startingindex* parameter is the numerical position of the first item to be retrieved. It SHOULD be a string that can be converted to a positive integer.

```
String = *(%x20-7E)
startingindex = String
```

### 2.2.3.46 **startingindexdata**

The *startingindexdata* parameter describes the numerical position of the first media item to be retrieved.

```
String = *(%x20-7E)
startingindexdata = String
```

### 2.2.3.47 **timeoutseconds**

The *timeoutseconds* parameter defines the number of seconds after which the request times out.

```
String = *(%x20-7E)
timeoutseconds = String
```

### 2.2.3.48 **titleonlydata**

The *titleonlydata* parameter describes whether the search is performed on the title only. If this parameter is not specified, the default value is FALSE.

```
String = *(%x20-7E)
titleonlydata = String
```

### 2.2.3.49 **usergroups**

The *usergroups* parameter is a list of the group GUIDs separated by commas, as specified in section 2.2.5.2.

```
String = *(%x20-7E)
usergroups = String
```

### 2.2.3.50 **userid**

The *userid* identifies the user. It SHOULD be a SID.

```
String = *(%x20-7E)
userid = String
```

### 2.2.3.51 username

The username parameter defines the user name and SHOULD be the logon name of the user.

```
String = *(%x20-7E)
username = String
```

### 2.2.3.52 usersid

This parameter defines the security identifier (SID) of a user.

```
String = *(%x20-7E)
usersid = String
```

### 2.2.3.53 vpnaccess

This is a flag that indicates whether to allow the user to have access to a virtual private network (VPN). It is set to TRUE to allow a user to have access to a VPN; otherwise, it is set to FALSE.

```
String = *(%x20-7E)
vpnaccess = String
```

### 2.2.3.54 widthdata

The *widthdata* parameter describes the width of an image.

```
String = *(%x20-7E)
widthdata = String
```

## 2.2.4 Complex Types

The following table summarizes the set of common XML Schema complex type definitions defined by this specification.

Complex Type	Description
AlertInfo	The AlertInfo type contains informational data about a Network Health Alert.
ArrayOfAlertInfo	This type describes an array of AlertInfo.
ArrayOfConnectionInfo	This type describes an array of ConnectionInfo.
ArrayOfDeviceInfo	This type describes an array of DeviceInfo.
ArrayOfDriveInfo	This type describes an array of DriveInfo.
ArrayOfFolderInfo	This type describes an array of FolderInfo.
ArrayOfGroupInfo	This type describes an array of GroupInfo.

<b>Complex Type</b>	<b>Description</b>
ArrayOfItemInfo	This type describes an array of ItemInfo.
ArrayOfLink	This type describes an array of Link.
ArrayOfMailbox	This type describes an array of Mailbox.
ArrayOfMetadataBase	This type describes an array of MetadataBase.
ArrayOfMetadataItemStream	This type describes an array of MetadataItemStream.
ArrayOfMSODomain	This type describes an array of MSODomain.
ArrayOfMSOLicense	This type describes an array of MSOLicense.
ArrayOfMSOLicenseService	This type describes an array of MSOLicenseService.
ArrayOfMSOLicenseSubscription	This type describes an array of MSOLicenseSubscription.
ArrayOfMSOLicenseSuite	This type contains the license suite information for the online service.
ArrayOfMSOUser	This type describes an array of MSOUser.
ArrayOfStorageDriveInfo	This type describes an array of StorageDriveInfo.
ArrayOfStorageServerInfo	This type describes an array of StorageServerInfo.
ArrayOfstring	This type describes an array of string.
ArrayOfUserInfo	This type describes an array of UserInfo.
ArrayOfWebApiProvider	This type describes an array of WebApiProvider.
CompanyAddress	The CompanyAddress type contains the address of the company.
ConnectionClientInfo	The ConnectionClientInfo type contains client information of the connection instance.
ConnectionInfo	The ConnectionInfo type contains user connection data information.
CustomizationInfo	The CustomizationInfo type contains customization data information.
DeviceInfo	The DeviceInfo type contains informational data about the device that is managed by Windows.
DriveInfo	The DriveInfo type contains informational data about the server drive.
FolderInfo	The FolderInfo type contains informational data about the folder that is managed by the server.
GroupInfo	The GroupInfo type contains informational data about the group.
ItemInfo	The file item metadata.
ItemList	The collection of the file item metadata.
ItemThumbnail	The ItemThumbnail type contains informational data

<b>Complex Type</b>	<b>Description</b>
	about Thumbnail.
Link	The Link type contains informational data about the customization link.
Mailbox	The Mailbox type contains information about the mailbox.
MetadataBase	The MetadataBase type is the base type for all the metadata of all the media content.
MetadataContainer	The MetadataContainer type describes the container for media content.
MetadataDetailResult	The MetadataDetailResult describes the detail information for the metadata container.
MetadataItem	The MetadataItem type contains the definition of the specific media item.
MetadataItemStream	The MetadataItemStream type describes the stream information of the media item.
MetadataResult	The MetadataResult type contains a list of media items and media containers, together with the total number that matches the query request.
MetadataStreamResult	The MetadataStreamResult type contains a list of media stream items, together with the count number of streams for the specified media items.
MSODomain	The MSODomain type contains the domain information for the online service.
MSOLicense	The MSOLicense type contains the license information for the online service
MSOLicenseService	The MSOLicenseService type contains the license type information for the online service
MSOLicenseSubscription	The MSOLicenseSubscription type contains the license subscription information for the online service.
MSOLicenseSuite	The MSOLicenseSuite type contains the license suite information for the online service.
MSOTenantInformation	The MSOTenantInformation type contains the tenant information for the online service.
MSOUser	The MSOUser type contains the user information for the online service.
PartialCollection_Of_AlertInfo	This type describes a partial collection of AlertInfo.
PartialCollection_Of_ConnectionInfo	This type describes a partial collection of ConnectionInfo.
PartialCollection_Of_DeviceInfo	This type describes a partial collection of DeviceInfo.
PartialCollection_Of_DriveInfo	This type describes a partial collection of DriveInfo.
PartialCollection_Of_FolderInfo	This type describes a partial collection of FolderInfo.

Complex Type	Description
PartialCollection_Of_UserInfo	This type describes a partial collection of UserInfo.
ReadOnlyCollectionOfGroupInfoEV6sb80H	This type describes a read-only connection of GroupInfo.
ReadOnlyCollectionOfMSOLicensepGX_Pb6b	This type describes a read-only connection of MSOLicense.
ReadOnlyCollectionOfMSOLicenseServicepGX_Pb6b	This type describes a read-only connection of MSOLicenseService.
ReadOnlyCollectionOfMSOLicenseSubscriptionpGX_Pb6b	This type describes a read-only connection of MSOLicenseSubscription.
ReadOnlyCollectionOfMSOLicenseSuitepGX_Pb6b	This type describes a read-only collection of MSOLicenseSuite as specified in section 2.2.4.48.
RemoteConnectionUserInfo	The RemoteConnectionUserInfo type contains user information about the remote connection.
SearchItemList	The collection of searched file item metadata.
ServerInfo	The ServerInfo type contains informational data about the server.
SharePointSiteAddressCollection	The SharePointSiteAddressCollection type contains the SharePoint site addresses.
StorageDriveInfo	This type contains information about the drive used to create the storage folder (see section 2.2.4.30).
StorageServerInfo	This type contains information about the server used to create the storage folder.
UserInfo	The UserInfo type contains informational data about the user.
WebApiProvider	The WebApiProvider type contains informational data about the WebApi provider.

### 2.2.4.1 AlertInfo

The AlertInfo type contains informational data about a Network Health Alert.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi>

```
<xs:complexType name="AlertInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="CanRepair" type="xs:boolean"/>
    <xs:element minOccurs="0" maxOccurs="1" name="DateAndTime" type="DateTime"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Description" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="IsSuppressed" type="xs:boolean"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Key" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="MachineName" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Severity" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Title" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

```
<xs:element minOccurs="0" maxOccurs="1" name="TroubleshootingSteps" nillable="true"
type="xs:string"/>
</xs:sequence>
</xs:complexType>
```

**CanRepair:** If the alert is repairable, this is set to TRUE; otherwise, FALSE.

**DateAndTime:** The date and time when the alert was raised.

**Description:** The description of the alert.

**IsSuppressed:** If the alert is suppressed, this is set to TRUE; otherwise, FALSE.

**Key:** The key of the alert.

**MachineName:** The name of the machine on which the alert is raised.

**Severity:** The health status.

**Title:** The name of the feature definition for which the alert is raised.

**TroubleshootingSteps:** The troubleshooting steps of the alert.

#### 2.2.4.2 ArrayOfAlertInfo

This type describes an array of AlertInfo.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi>

```
<xs:complexType name="ArrayOfAlertInfo">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded" name="AlertInfo" nillable="true"
type="tns1:AlertInfo"/>
  </xs:sequence>
</xs:complexType>
```

**AlertInfo:** The instance of AlertInfo. See section 2.2.4.1

#### 2.2.4.3 ArrayOfConnectionInfo

This type describes an array of ConnectionInfo.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi>

```
<xs:complexType name="ArrayOfConnectionInfo">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded" name="ConnectionInfo" nillable="true"
type="tns1:ConnectionInfo"/>
  </xs:sequence>
</xs:complexType>
```

**ConnectionInfo:** The connection information instance.

#### 2.2.4.4 ArrayOfDeviceInfo

This type describes an array of DeviceInfo.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi>

```
<xs:complexType name="ArrayOfDeviceInfo">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded" name="DeviceInfo" nillable="true"
type="tns1:DeviceInfo"/>
  </xs:sequence>
</xs:complexType>
```

**DeviceInfo:** The instance of DeviceInfo, as specified in section 2.2.4.28.

### 2.2.4.5 ArrayOfDriveInfo

This type describes an array of DriveInfo.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi>

```
<xs:complexType name="ArrayOfDriveInfo">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded" name="DriveInfo" nillable="true"
type="tns1:DriveInfo"/>
  </xs:sequence>
</xs:complexType>
```

**DriveInfo:** The instance of DriveInfo, as specified in section 2.2.4.29.

### 2.2.4.6 ArrayOfFolderInfo

This type describes an array of FolderInfo.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi>

```
<xs:complexType name="ArrayOfFolderInfo">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded" name="FolderInfo" nillable="true"
type="tns1:FolderInfo"/>
  </xs:sequence>
</xs:complexType>
```

**FolderInfo:** The instance of FolderInfo, as specified in section 2.2.4.30.

### 2.2.4.7 ArrayOfGroupInfo

This type describes an array of GroupInfo.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi>

```
<xs:complexType name="ArrayOfGroupInfo">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded" name="GroupInfo" nillable="true"
type="tns1:GroupInfo"/>
  </xs:sequence>
</xs:complexType>
```

**GroupInfo:** The instance of Group, as specified in section 2.2.4.31.



### 2.2.4.8 ArrayOfItemInfo

This type describes an array of ItemInfo.

**Namespace:** http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi

```
<xs:complexType name="ArrayOfItemInfo">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded" name="ItemInfo" nillable="true"
type="tns1:ItemInfo"/>
  </xs:sequence>
</xs:complexType>
```

**ItemInfo:** The instance of ItemInfo as specified in section 2.2.4.32.

### 2.2.4.9 ArrayOfLink

This type describes an array of Link.

**Namespace:** http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi

```
<xs:complexType name="ArrayOfLink">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded" name="Link" nillable="true"
type="tns1:Link"/>
  </xs:sequence>
</xs:complexType>
```

**Link:** The instance of Link type represents the customization link information, as specified in section 2.2.4.35.

### 2.2.4.10 ArrayOfMailbox

This type describes an array of Mailbox.

**Namespace:** http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi

```
<xs:complexType name="ArrayOfMailbox">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded" name="Mailbox" nillable="true"
type="tns1:Mailbox"/>
  </xs:sequence>
</xs:complexType>
```

**Mailbox:** The instance of Mailbox, as specified in section 2.2.4.36.

### 2.2.4.11 ArrayOfMetadataBase

This type describes an array of MetadataBase.

**Namespace:** http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi

```
<xs:complexType name="ArrayOfMetadataBase">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded" name="MetadataBase" nillable="true"
type="tns1:MetadataBase"/>
  </xs:sequence>
```

```
</xs:complexType>
```

**MetadataBase:** The instance of Metadata, as specified in section 2.2.4.37.

#### 2.2.4.12 ArrayOfMetadataItemStream

This type describes an array of MetadataItemStream as specified in section 2.2.4.41.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebAPI>

```
<xs:complexType name="ArrayOfMetadataItemStream">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded" name="MetadataItemStream" nillable="true"
      type="tns1:MetadataItemStream"/>
  </xs:sequence>
</xs:complexType>
```

**MetadataItemStream:** Describes the stream information of the media item.

#### 2.2.4.13 ArrayOfMSODomain

This type describes an array of MSODomain, specified in section 2.2.4.44.

**Namespace:**

<http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObjectModel>

```
<xs:complexType name="ArrayOfMSODomain">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded" name="MSODomain" nillable="true"
      type="tns6:MSODomain"/>
  </xs:sequence>
</xs:complexType>
```

**MSODomain:** An instance of MSODomain.

#### 2.2.4.14 ArrayOfMSOLicense

This type describes an array of MSOLicense, specified in section 2.2.4.45.

**Namespace:**

<http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObjectModel>

```
<xs:complexType name="ArrayOfMSOLicense">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded" name="MSOLicense" nillable="true"
      type="tns6:MSOLicense"/>
  </xs:sequence>
</xs:complexType>
```

**MSOLicense:** The instance of MSOLicense.

#### 2.2.4.15 ArrayOfMSOLicenseService

This type describes an array of MSOLicenseService specified in section 2.2.4.46.

**Namespace:**

<http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObjectModel>

```
<xs:complexType name="ArrayOfMSOLicenseService">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded" name="MSOLicenseService" nillable="true"
type="tns6:MSOLicenseService"/>
  </xs:sequence>
</xs:complexType>
```

**MSOLicenseService:** The instance of MSOLicenseService.

### 2.2.4.16 ArrayOfMSOLicenseSubscription

This type describes an array of MSOLicenseSubscription, specified in section 2.2.4.47.

**Namespace:**

<http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObjectModel>

```
<xs:complexType name="ArrayOfMSOLicenseSubscription">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded" name="MSOLicenseSubscription"
nillable="true" type="tns6:MSOLicenseSubscription"/>
  </xs:sequence>
</xs:complexType>
```

**MSOLicenseSubscription:** The instance of MSOLicenseSubscription.

### 2.2.4.17 ArrayOfMSOLicenseSuite

This type describes an array of MSOLicenseSuite, specified in section 2.2.4.48.

**Namespace:**

<http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObjectModel>

```
<xs:complexType name="ArrayOfMSOLicenseSuite">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded" name="MSOLicenseSuite" nillable="true"
type="tns6:MSOLicenseSuite"/>
  </xs:sequence>
</xs:complexType>
```

**MSOLicenseSuite:** The instance of MSOLicenseSuite.

### 2.2.4.18 ArrayOfMSOUser

This type describes an array of MSOUser, specified in section 2.2.4.50.

**Namespace:**

<http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObjectModel>

```
<xs:complexType name="ArrayOfMSOUser">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded" name="MSOUser" nillable="true"
type="tns6:MSOUser"/>
  </xs:sequence>
```

```
</xs:complexType>
```

**MSOUser:** The instance of MSOUser.

#### 2.2.4.19      **ArrayOfStorageDriveInfo**

This type describes an array of StorageDriveInfo.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi>

```
<xs:complexType name="ArrayOfStorageDriveInfo">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded" name="StorageDriveInfo" nillable="true"
type="tnsl:StorageDriveInfo"/>
  </xs:sequence>
</xs:complexType>
```

**StorageDriveInfo:** The instance of StorageDriveInfo. See section 2.2.4.66

#### 2.2.4.20      **ArrayOfStorageServerInfo**

This type describes an array of storageServerInfo.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi>

```
<xs:complexType name="ArrayOfStorageServerInfo">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded" name="StorageServerInfo" nillable="true"
type="tnsl:StorageServerInfo"/>
  </xs:sequence>
</xs:complexType>
```

**StorageServerInfo:** The instance of StorageServerInfo. See section 2.2.4.67

#### 2.2.4.21      **ArrayOfstring**

This type describes an array of string.

**Namespace:** <http://schemas.microsoft.com/2003/10/Serialization/Arrays>

```
<xs:complexType name="ArrayOfstring">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded" name="string" nillable="true"
type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

**string:** The instance of string.

#### 2.2.4.22      **ArrayOfUserInfo**

This type describes an array of UserInfo.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi>

```

<xs:complexType name="ArrayOfUserInfo">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded" name="UserInfo" nillable="true"
type="tns1:UserInfo"/>
  </xs:sequence>
</xs:complexType>

```

**UserInfo:** The instance of UserInfo. See section 2.2.4.68

### 2.2.4.23 ArrayOfWebApiProvider

This type describes an array of WebApiProvider, as specified in section 2.2.4.69.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi>

```

<xs:complexType name="ArrayOfWebApiProvider">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded" name="WebApiProvider" nillable="true"
type="tns1:WebApiProvider"/>
  </xs:sequence>
</xs:complexType>

```

**WebApiProvider:** The instance of WebApiProvider. See section 2.2.4.69

### 2.2.4.24 CompanyAddress

This type describes the address of a company.

**Namespace:**

<http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObjectModel>

```

<xs:complexType name="CompanyAddress">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="City" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Country" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="PostalCode" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="State" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Street" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="TelephoneNumber" nillable="true"
type="xs:string"/>
  </xs:sequence>
</xs:complexType>

```

**City:** The city of the company.

**Country:** The country of the company.

**PostalCode:** The postal code of the company.

**State:** The state of the company.

**Street:** The street of the company.

**TelephoneNumber:** The telephone number of the company.

### 2.2.4.25 ConnectionClientInfo

The ConnectionClientInfo type contains client information of the connection instance.

**Namespace:** http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi

```
<xs:complexType name="ConnectionClientInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="ClientApplicationName" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ClientApplicationPublisher" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ClientApplicationVersion" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ClientHostname" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ClientIPAddress" nillable="true"
type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

**ClientApplicationName:** The client application name.

**ClientApplicationPublisher:** The client application publisher information.

**ClientApplicationVersion:** The client application version.

**ClientHostname:** The client host name.

**ClientIPAddress:** The client IP address.

### 2.2.4.26 ConnectionInfo

The ConnectionInfo type contains user connection data information.

**Namespace:** http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi

```
<xs:complexType name="ConnectionInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="ClientInfo" nillable="true"
type="tns1:ConnectionClientInfo"/>
    <xs:element minOccurs="0" maxOccurs="1" name="EndTime" type="DateTime"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ServiceType" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="StartTime" type="DateTime"/>
    <xs:element minOccurs="0" maxOccurs="1" name="UserInfo" nillable="true"
type="tns1:RemoteConnectionUserInfo"/>
  </xs:sequence>
</xs:complexType>
```

**ClientInfo:** The client information of the connection. See section 2.2.4.25.

**EndTime:** The end time of the connection.

**ServiceType:** The service type of the connection.

**StartTime:** The start time of the connection.

**UserInfo:** The user information of the connection. See section 2.2.4.62.

## 2.2.4.27 CustomizationInfo

The CustomizationInfo type contains customization data information.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebAPI>

```
<xs:complexType name="CustomizationInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="BackgroundImage" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Links" nillable="true"
type="tnsl:ArrayOfLink"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Logo" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Title" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

**BackgroundImage:** The URI of the background image.

**Links:** An array that contains customization links.

**Logo:** The URI of the logo.

**Title:** The URI of the title.

## 2.2.4.28 DeviceInfo

The DeviceInfo type contains informational data about a device that is managed by the server.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebAPI>

```
<xs:complexType name="DeviceInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="AntiSpywareStatus" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="AntiVirusStatus" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="BackupProgress" type="xs:int"/>
    <xs:element minOccurs="0" maxOccurs="1" name="BackupStatus" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="CanBackup" type="xs:boolean"/>
    <xs:element minOccurs="0" maxOccurs="1" name="DeviceDescription" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="DeviceName" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="DeviceType" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="FirewallStatus" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Key" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="OSFamily" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="OSName" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="OSServicePack" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="OnlineStatus" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="SecurityCenterStatus" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="SystemType" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="UpdateDetailStatus" nillable="true"
type="xs:string"/>
```

```
<xs:element minOccurs="0" maxOccurs="1" name="UpdateStatus" nillable="true"
type="xs:string"/>
</xs:sequence>
</xs:complexType>
```

**AntiSpywareStatus:** The antispware status of the device.

**AntiVirusStatus:** The antivirus status of the device.

**BackupProgress:** The backup process if the backup status is InProcess.

**BackupStatus:** The backup status of devices. It SHOULD one of these values:

- Unknown
- Configured
- Successful
- Failed
- Incomplete
- InProgress
- NotConfigured
- InQueue
- PreparingForBackup
- FinalizingBackup
- RecoveryInProgress

**CanBackup:** A flag that indicates whether the device can start backup.

**DeviceDescription:** The description of the device.

**DeviceName:** The name of the device.

**DeviceType:** The type of device, which SHOULD be one of these values:

- Unknown
- ArchivedClient
- Client
- SolutionServer

**FirewallStatus:** The firewall status of the device.

**Key:** The key of the device.

**OnlineStatus:** The online status of the device, which SHOULD be one of these values:

- Unknown
- Online
- Offline



**OSFamily:** The operating system information, which SHOULD one of these values:

- Unknown
- WindowsSeverity
- Macintosh

**OSName:** The name of the operating system of the device.

**OSServicePack:** The operating system service pack version of the device.

**SecurityCenterStatus:** The security center status of the device.

**SystemType:** The system type of the device, which SHOULD be one of these values:

- Unknown
- X86
- AMD64

**UpdateDetailStatus:** The Windows update detail information of the device.

**UpdateStatus:** The Windows update status of the device.

#### 2.2.4.29 DriveInfo

The DriveInfo type contains informational data about the server drive.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebAPI>

```
<xs:complexType name="DriveInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Capacity" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="DataStatus" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="DiskID" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="DriveGuid" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="FileSystemType" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ID" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="MountPoint" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Name" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="NextSnapshotTime" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="SnapshotsEnabled" type="xs:boolean"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Status" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="SystemDrive" type="xs:boolean"/>
    <xs:element minOccurs="0" maxOccurs="1" name="UsedSpace" nillable="true"
type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

**Capacity:** The total capacity of this drive in bytes.

**DataStatus:** The data status of this drive.

**DiskID:** The unique ID of the parent disk.

**DriveGuid:** The drive GUID, which persists across reboots.

**FileSystemType:** The file system on this drive, which SHOULD be one of the following:

- Unknown
- Raw
- FAT
- FAT32
- NTFS
- CDFS
- UDF
- EXFAT
- CSVFS
- REFS

**ID:** The unique ID of the drive.

**MountPoint:** The mount point of the drive.

**Name:** The user-friendly name of the drive.

**NextSnapshotTime:** The time at which the drive takes the next snapshot.

**SnapshotsEnabled:** A flag that indicates whether snapshots are enabled for this drive.

**Status:** The status of this drive.

**SystemDrive:** A flag that indicates whether the drive is a system drive.

**UsedSpace:** The total used space on this drive in bytes.

### 2.2.4.30 FolderInfo

The FolderInfo type contains informational data about the folder that is managed by the server.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi>

```
<xs:complexType name="FolderInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="AccessPermission" nillable="true"
      xmlns:q9="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.Storage"
      type="q9:Permission"/>
    <xs:element minOccurs="0" maxOccurs="1" name="CurrentStatus" nillable="true"
      type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Description" nillable="true"
      type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="DriveFreeSpace" nillable="true"
      type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="DriveID" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="FolderType" nillable="true"
      xmlns:q10="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.WebApi.Ma
      nagement.Storage" type="q10:ServerFolderType"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ID" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Name" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Path" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

```

    <xs:element minOccurs="0" maxOccurs="1" name="SharePath" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Shared" type="xs:boolean"/>
    <xs:element minOccurs="0" maxOccurs="1" name="UncPath" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

```

**AccessPermission:** The user access permission on this folder, as specified in section 2.2.5.5.

**CurrentStatus:** The folder status.

**Description:** The user-friendly description of the folder.

**DriveFreeSpace:** The free space of the parent drive.

**DriveID:** The ID of the drive this folder is on.

**FolderType:** The folder type. It SHOULD be one of ServerFolderType, see section 2.2.5.6.

**ID:** The unique ID of the folder.

**Name:** The folder name.

**Path:** The local path of the folder.

**Shared:** A flag that indicates whether the folder is shared.

**SharePath:** The share path of this folder. If the folder is not shared, this is NULL.

**UncPath:** The Universal Naming Convention (UNC) path of the folder.

### 2.2.4.31 GroupInfo

The GroupInfo type contains informational data about the group.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebAPI>

```

<xs:complexType name="GroupInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="CurrentSid" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="GroupName" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Guid" nillable="true"
xmlns:q12="http://schemas.microsoft.com/2003/10/Serialization/" type="q12:guid"/>
    <xs:element minOccurs="0" maxOccurs="1" name="RemoteWebAccess" type="xs:boolean"/>
    <xs:element minOccurs="0" maxOccurs="1" name="VpnAccess" type="xs:boolean"/>
    <xs:element minOccurs="0" maxOccurs="1" name="WindowsAccount" nillable="true"
type="xs:string"/>
  </xs:sequence>
</xs:complexType>

```

**CurrentSid:** The SID of the group.

**GroupName:** The name of the group.

**Guid:** The Guid of the group.

**RemoteWebAccess:** A flag that indicates whether the group has remote web access permission.

**VpnAccess:** A flag that indicates whether the group has VPN access permission.

**WindowsAccount:** The Windows account of the group.

### 2.2.4.32 ItemInfo

This is the file item metadata.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi>

```
<xs:complexType name="ItemInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="AccessPermission" nillable="true"
      xmlns:ql="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.Storage"
      type="ql:Permission"/>
    <xs:element minOccurs="0" maxOccurs="1" name="CreateTime" type="DateTime"/>
    <xs:element minOccurs="0" maxOccurs="1" name="CreateTimeUtc" type="DateTime"/>
    <xs:element minOccurs="0" maxOccurs="1" name="IsDirectory" type="xs:boolean"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ModifiedTime" type="DateTime"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ModifiedTimeUtc" type="DateTime"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Name" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Path" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Size" type="xs:long"/>
  </xs:sequence>
</xs:complexType>
```

**AccessPermission:** This is a number that represents the current user's access permission to the file item. It MUST be one of the following: 0 (no access), 1 (read-only access), 2 (read/write access), or 3 (unknown or unspecific access).

**CreateTime:** The time that the file is created.

**CreateTimeUtc:** The UTC time that the file is created.

**IsDirectory:** A Boolean value that indicates whether the file item is a directory.

**ModifiedTime:** The last modified time of the file item.

**ModifiedTimeUtc:** The last modified UTC time of the file item.

**Name:** The name of the file item.

**Path:** The UNC path of the file item.

**Size:** The size (in bytes) of the file item.

### 2.2.4.33 ItemList

The collection of file item metadata.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi>

```
<xs:complexType name="ItemList">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Items" nillable="true"
      type="tns1:ArrayOfItemInfo"/>
    <xs:element minOccurs="0" maxOccurs="1" name="TotalCount" type="xs:int"/>
  </xs:sequence>
</xs:complexType>
```

**Items:** The collection of file item metadata.

**TotalCount:** The total count of the matched file items regardless of index and count parameter.

#### 2.2.4.34 ItemThumbnail

The ItemThumbnail type contains informational data about the Thumbnail.

**Namespace:** http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi

```
<xs:complexType name="ItemThumbnail">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Buffer" nillable="true"
type="xs:base64Binary"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Format" nillable="true"
type="tns1:ThumbnailFormat"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Height" type="xs:int"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Width" type="xs:int"/>
  </xs:sequence>
</xs:complexType>
```

**Buffer:** The raw image buffer.

**Format:** The image format.

**Height:** The image height.

**Width:** The image width.

#### 2.2.4.35 Link

The Link type contains informational data about the link that can be customized by the administrator.

**Namespace:** http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi

```
<xs:complexType name="Link">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Address" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Title" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

**Address:** The URI of the customizable link.

**Title:** The title of the customizable link.

#### 2.2.4.36 Mailbox

The Mailbox type contains the information about the mailbox.

**Namespace:** http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi

```
<xs:complexType name="Mailbox">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Email" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="UserName" nillable="true"
type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

**Email:** The email address.

**UserName:** The local user name.

### 2.2.4.37 MetadataBase

This is the base type for all the metadata of all the media content, including each specific item and container.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebAPI>

```
<xs:complexType name="MetadataBase">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Date" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Id" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Title" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="TypeName" nillable="true"
type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

**Date:** The date of the item.

**Id:** The ID of the item.

**Title:** The title of the item.

**TypeName:** The type name of the item.

### 2.2.4.38 MetadataContainer

The MetadataContainer type describes the container for media content. According to different grouping methods used for browsing, the meaning of the container will be different. For example, if it is grouped by Music Album, then the media item in each container will be the music from the same album. If it is grouped by the folder, then it can be a regular file, and the media item in each container will be the media in the same folder.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebAPI>

```
<xs:complexType name="MetadataContainer">
  <xs:complexContent mixed="false">
    <xs:extension base="tns1:MetadataBase">
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="AllCount" type="xs:int"/>
        <xs:element minOccurs="0" maxOccurs="1" name="AllDuration" type="xs:long"/>
        <xs:element minOccurs="0" maxOccurs="1" name="AllSize" type="xs:long"/>
        <xs:element minOccurs="0" maxOccurs="1" name="CalculationDone" type="xs:boolean"/>
        <xs:element minOccurs="0" maxOccurs="1" name="ContainerType" nillable="true"
type="tns1:ContainerType"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Count" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="FullPath" nillable="true"
type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

**AllCount:** The count of subcontainers and subitems (recursive).

**AllDuration:** Total duration of subcontainers and subitems (recursive).

**AllSize:** Total size of subcontainers and subitems (recursive).

**CalculationDone:** A flag indicating whether the detail information is available.

**ContainerType:** The type of container.

**Count:** The count of subcontainers and sub-items (nonrecursive).

**FullPath:** The full path for the container when grouping by folder.

#### 2.2.4.39 MetadataDetailResult

The MetadataDetailResult type describes the detail information for the metadata container.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi>

```
<xs:complexType name="MetadataDetailResult">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Result" nillable="true"
type="tnsl:MetadataContainer"/>
  </xs:sequence>
</xs:complexType>
```

**Result:** The result of MetadataDetail.

#### 2.2.4.40 MetadataItem

The MetadataItem type contains the definition of the specific media item.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi>

```
<xs:complexType name="MetadataItem">
  <xs:complexContent mixed="false">
    <xs:extension base="tnsl:MetadataBase">
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="Album" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="AlbumArtist" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Duration" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="ParentFullPath" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Rating" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Resolution" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Size" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Tags" nillable="true"
type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

**Album:** The album of the media item.

**AlbumArtist:** The album artist of the media item.

**Duration:** The duration of the media item.

**ParentFullPath:** The full path of the media item's parent container.

**Rating:** The rating of the media item.

**Resolution:** The resolution of the media item.

**Size:** The size of the media item.

**Tags:** The tags of the media item.

#### 2.2.4.41 MetadataItemStream

The MetadataItemStream type describes the stream information of the media item.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebAPI>

```
<xs:complexType name="MetadataItemStream">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Bitrate" type="xs:int"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Height" type="xs:int"/>
    <xs:element minOccurs="0" maxOccurs="1" name="MimeType" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Profile" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="RelativePath" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="RequiresTranscoding" type="xs:boolean"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Width" type="xs:int"/>
  </xs:sequence>
</xs:complexType>
```

**Bitrate:** The bitrate of the media stream.

**Height:** The height of the media stream.

**MimeType:** The Multipurpose Internet Mail Exchange (MIME) type of the media stream.

**Profile:** The profile of the media stream.

**RelativePath:** The relative path of the media stream, which can be used to generate the full path of the media stream.

**RequiresTranscoding:** Indicates whether this media stream requires transcoding on the server side.

**Width:** The width of the media stream.

#### 2.2.4.42 MetadataResult

The MetadataResult type contains a list of media items and media containers, together with the total number that matches the query request.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebAPI>

```
<xs:complexType name="MetadataResult">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Result" nillable="true"
type="tns:ArrayOfMetadataBase"/>
    <xs:element minOccurs="0" maxOccurs="1" name="TotalMatches" type="xs:unsignedInt"/>
  </xs:sequence>
```



```
</xs:complexType>
```

**Result:** A list of media items and media containers.

**TotalMatches:** The total number that matches the query request.

#### 2.2.4.43 MetadataStreamResult

The MetadataStreamResult type contains a list of media stream items, together with the number of streams for the specified media items.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebAPI>

```
<xs:complexType name="MetadataStreamResult">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Result" nillable="true"
type="tns1:ArrayOfMetadataItemStream"/>
    <xs:element minOccurs="0" maxOccurs="1" name="TotalMatches" type="xs:unsignedInt"/>
  </xs:sequence>
</xs:complexType>
```

**Result:** A list of media stream items.

**TotalMatches:** The count number of streams for the specified media items.

#### 2.2.4.44 MSODomain

The MSODomain type contains the domain information for the online service.

**Namespace:**

<http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObjectModel>

```
<xs:complexType name="MSODomain">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="CapabilityType" nillable="true"
type="tns6:MSODomainTypes"/>
    <xs:element minOccurs="0" maxOccurs="1" name="IsPrimary" type="xs:boolean"/>
    <xs:element minOccurs="0" maxOccurs="1" name="IsVerified" type="xs:boolean"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Name" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

**CapabilityType:** The domain type.

**IsPrimary:** A flag that indicates whether this domain is primary.

**IsVerified:** A flag that indicates whether this domain is verified.

**Name:** The domain name.

#### 2.2.4.45 MSOLicense

The MSOLicense type contains the license information for the online service.

**Namespace:**

<http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObjectModel>

```

<xs:complexType name="MSOLicense">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="AccountName" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Services" nillable="true"
xmlns:q3="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
type="q3:ReadOnlyCollectionOfMSOLicenseServicePGX_Pb6b"/>
    <xs:element minOccurs="0" maxOccurs="1" name="SkuId" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

```

**AccountName:** The account name of the license.

**Services:** The service of the license.

**SkuId:** The **SKU** ID of the license.

#### 2.2.4.46 MSOLicenseService

The MSOLicenseService type contains the license type information for the online service.

**Namespace:**

<http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObjectModel>

```

<xs:complexType name="MSOLicenseService">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Activated" type="xs:boolean"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Description" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ServicePlan" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ServiceType"
type="tns6:MSOLicenseServiceTypes"/>
  </xs:sequence>
</xs:complexType>

```

**Activated:** A flag that indicates whether the service is activated.

**Description:** The description of the license.

**ServicePlan:** The service plan for the license.

**ServiceType:** The service type for the license.

#### 2.2.4.47 MSOLicenseSubscription

The MSOLicenseSubscription type contains the license subscription information for the online service.

**Namespace:**

<http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObjectModel>

```

<xs:complexType name="MSOLicenseSubscription">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="BillingExpirationTime" type="DateTime"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Expired" type="xs:boolean"/>
  </xs:sequence>
</xs:complexType>

```

**BillingExpirationTime:** The billing expiration time of the subscription.

**Expired:** A flag that indicates whether the subscription is expired.

#### 2.2.4.48 MSOLicenseSuite

The MSOLicenseSuite type contains the license suite information for the online service.

**Namespace:**

<http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObjectModel>

```
<xs:complexType name="MSOLicenseSuite">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="ConsumedLicenses" type="xs:int"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Description" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="License" nillable="true"
type="tns6:MSOLicense"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Subscriptions" nillable="true"
xmlns:q5="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
type="q5:ReadOnlyCollectionOfMSOLicenseSubscriptionPGX_Pb6b"/>
    <xs:element minOccurs="0" maxOccurs="1" name="TotalLicenses" type="xs:int"/>
  </xs:sequence>
</xs:complexType>
```

**ConsumedLicenses:** The number of consumed licenses.

**Description:** The description of the license.

**License:** The license type included in the license suite.

**Subscriptions:** The collection of the subscriptions for the license suite.

**TotalLicenses:** The total license number of licenses.

#### 2.2.4.49 MSOTenantInformation

The MSOTenantInformation type contains the tenant information for the online service.

**Namespace:**

<http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObjectModel>

```
<xs:complexType name="MSOTenantInformation">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Activated" type="xs:boolean"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Address" nillable="true"
type="tns6:CompanyAddress"/>
    <xs:element minOccurs="0" maxOccurs="1" name="CompanyName" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="InitialDomain" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="LicenseSuite" nillable="true"
xmlns:q4="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
type="q4:ReadOnlyCollectionOfMSOLicenseSuitePGX_Pb6b"/>
    <xs:element minOccurs="0" maxOccurs="1" name="PrimaryDomain" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="TechnicalContact" nillable="true"
type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

**Activated:** A flag that indicates whether the online account is activated for this tenant.

**Address:** The address of the company.

**CompanyName:** The name of the company.

**InitialDomain:** The initial domain.

**LicenseSuite:** The license suite.

**PrimaryDomain:** The primary domain.

**TechnicalContact:** The technical contact information.

#### 2.2.4.50 MSOUser

The MSOUser type contains the user information for the online service.

**Namespace:**

<http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObjectModel>

```
<xs:complexType name="MSOUser">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Activated" type="xs:boolean"/>
    <xs:element minOccurs="0" maxOccurs="1" name="FirstName" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="LastName" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Licenses" nillable="true"
xmlns:q1="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
type="q1:ReadOnlyCollectionOfMSOLicensePGX_Pb6b"/>
    <xs:element minOccurs="0" maxOccurs="1" name="LocalUserName" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ObjectId" nillable="true"
xmlns:q2="http://schemas.microsoft.com/2003/10/Serialization/" type="q2:guid"/>
    <xs:element minOccurs="0" maxOccurs="1" name="UserPrincipalName" nillable="true"
type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

**Activated:** A flag that indicates whether the user is activated.

**FirstName:** The first name of the local user.

**LastName:** The last name of the local user.

**Licenses:** The licenses assigned to the online user.

**LocalUserName:** The local user name.

**ObjectId:** The object ID of the online user.

**UserPrincipalName:** The principle name for the user.

#### 2.2.4.51 PartialCollection\_Of\_AlertInfo

This type describes a partial collection of AlertInfo (section 2.2.4.1). If **CollectionSize** is less than zero, the collection is not available and the **CollectionChunk** field SHOULD be ignored.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebAPI>

```
<xs:complexType name="PartialCollection_Of_AlertInfo">
  <xs:sequence>
```

```

    <xs:element minOccurs="0" maxOccurs="1" name="CollectionChunk" nillable="true"
type="tns1:ArrayOfAlertInfo"/>
    <xs:element minOccurs="0" maxOccurs="1" name="CollectionModified" type="DateTime"/>
    <xs:element minOccurs="0" maxOccurs="1" name="CollectionSize" type="xs:int"/>
  </xs:sequence>
</xs:complexType>

```

**CollectionChunk:** The items retrieved from the overall collection.

**CollectionModified:** The date and time the collection was last changed.

**CollectionSize:** The total number of items in the complete collection.

#### 2.2.4.52 PartialCollection\_Of\_ConnectionInfo

This type describes a partial collection of ConnectionInfo (section 2.2.4.26). If **CollectionSize** is less than zero, the collection is not available and the **CollectionChunk** field SHOULD be ignored.

**Namespace:** http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebAPI

```

<xs:complexType name="PartialCollection_Of_ConnectionInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="CollectionChunk" nillable="true"
type="tns1:ArrayOfConnectionInfo"/>
    <xs:element minOccurs="0" maxOccurs="1" name="CollectionModified" type="DateTime"/>
    <xs:element minOccurs="0" maxOccurs="1" name="CollectionSize" type="xs:int"/>
  </xs:sequence>
</xs:complexType>

```

**CollectionChunk:** The items retrieved from the overall collection.

**CollectionModified:** The date and time the collection last changed.

**CollectionSize:** The total number of items in the complete collection.

#### 2.2.4.53 PartialCollection\_Of\_DeviceInfo

This type describes a partial collection of DeviceInfo (section 2.2.4.28). If **CollectionSize** is less than zero, the collection is not available and the **CollectionChunk** field SHOULD be ignored.

**Namespace:** http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebAPI

```

<xs:complexType name="PartialCollection_Of_DeviceInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="CollectionChunk" nillable="true"
type="tns1:ArrayOfDeviceInfo"/>
    <xs:element minOccurs="0" maxOccurs="1" name="CollectionModified" type="DateTime"/>
    <xs:element minOccurs="0" maxOccurs="1" name="CollectionSize" type="xs:int"/>
  </xs:sequence>
</xs:complexType>

```

**CollectionChunk:** The items retrieved from the overall collection.

**CollectionModified:** The date and time the collection last changed.

**CollectionSize:** The total number of items in the complete collection.

#### 2.2.4.54 PartialCollection\_Of\_DriveInfo

This type describes a partial collection of DriveInfo (section 2.2.4.29). If **CollectionSize** is less than zero, the collection is not available and the **CollectionChunk** field SHOULD be ignored.

**Namespace:** http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebAPI

```
<xs:complexType name="PartialCollection_Of_DriveInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="CollectionChunk" nillable="true"
type="tns1:ArrayOfDriveInfo"/>
    <xs:element minOccurs="0" maxOccurs="1" name="CollectionModified" type="DateTime"/>
    <xs:element minOccurs="0" maxOccurs="1" name="CollectionSize" type="xs:int"/>
  </xs:sequence>
</xs:complexType>
```

**CollectionChunk:** Items retrieved from the overall collection.

**CollectionModified:** Date and time the collection last changed.

**CollectionSize:** The total number of items in the complete collection.

#### 2.2.4.55 PartialCollection\_Of\_FolderInfo

This Type describes a partial collection of FolderInfo (section 2.2.4.30). If **CollectionSize** is less than zero, the collection is not available and the **CollectionChunk** field SHOULD be ignored.

**Namespace:** http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebAPI

```
<xs:complexType name="PartialCollection_Of_FolderInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="CollectionChunk" nillable="true"
type="tns1:ArrayOfFolderInfo"/>
    <xs:element minOccurs="0" maxOccurs="1" name="CollectionModified" type="DateTime"/>
    <xs:element minOccurs="0" maxOccurs="1" name="CollectionSize" type="xs:int"/>
  </xs:sequence>
</xs:complexType>
```

**CollectionChunk:** The items retrieved from the overall collection.

**CollectionModified:** The date and time the collection last changed.

**CollectionSize:** The total number of items in the complete collection.

#### 2.2.4.56 PartialCollection\_Of\_UserInfo

This type describes an partial collection of UserInfo (section 2.2.4.68). If **CollectionSize** is less than zero, the collection is not available and the **CollectionChunk** field SHOULD be ignored.

**Namespace:** http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebAPI

```
<xs:complexType name="PartialCollection_Of_UserInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="CollectionChunk" nillable="true"
type="tns1:ArrayOfUserInfo"/>
    <xs:element minOccurs="0" maxOccurs="1" name="CollectionModified" type="DateTime"/>
    <xs:element minOccurs="0" maxOccurs="1" name="CollectionSize" type="xs:int"/>
  </xs:sequence>
</xs:complexType>
```

**CollectionChunk:** The items retrieved from the overall collection.

**CollectionModified:** The date and time the collection changed lastly.

**CollectionSize:** The total number of items in the complete collection.

#### 2.2.4.57 **ReadOnlyCollectionOfGroupInfoEV6sb80H**

This type describes a read-only collection of GroupInfo (section 2.2.4.31).

**Namespace:** <http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel>

```
<xs:complexType name="ReadOnlyCollectionOfGroupInfoEV6sb80H">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="list" nillable="true"
      xmlns:q5="http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebAPI"
      type="q5:ArrayOfGroupInfo"/>
  </xs:sequence>
</xs:complexType>
```

**list:** The content of the read-only collection, it contains the list of GroupInfo instances.

#### 2.2.4.58 **ReadOnlyCollectionOfMSOLicensePGX\_Pb6b**

This type describes a read-only collection of MSOLicense (section 2.2.4.45).

**Namespace:** <http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel>

```
<xs:complexType name="ReadOnlyCollectionOfMSOLicensePGX_Pb6b">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="list" nillable="true"
      xmlns:q1="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObj
      ectModel" type="q1:ArrayOfMSOLicense"/>
  </xs:sequence>
</xs:complexType>
```

**list:** The content of the read-only collection; it contains the list of MSOLicense instances.

#### 2.2.4.59 **ReadOnlyCollectionOfMSOLicenseServicePGX\_Pb6b**

This type describes a read-only collection of MSOLicenseService (section 2.2.4.46).

**Namespace:** <http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel>

```
<xs:complexType name="ReadOnlyCollectionOfMSOLicenseServicePGX_Pb6b">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="list" nillable="true"
      xmlns:q2="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObj
      ectModel" type="q2:ArrayOfMSOLicenseService"/>
  </xs:sequence>
</xs:complexType>
```

**list:** The content of the read-only collection; it contains the list of MSOLicenseService instances.

#### 2.2.4.60 **ReadOnlyCollectionOfMSOLicenseSubscriptionPGX\_Pb6b**

This type describes a read-only collection of MSOLicenseSubscription (section 2.2.4.47).

**Namespace:** http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel

```
<xs:complexType name="ReadOnlyCollectionOfMSOLicenseSubscriptionPGX_Pb6b">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="list" nillable="true"
xmlns:q4="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObjectModel" type="q4:ArrayOfMSOLicenseSubscription"/>
  </xs:sequence>
</xs:complexType>
```

**list:** The content of the read-only collection; it contains the list of MSOLicenseSubscription instances.

### 2.2.4.61 ReadOnlyCollectionOfMSOLicenseSuitePGX\_Pb6b

This type describes a read-only collection of MSOLicenseSuite as specified in section 2.2.4.48.

**Namespace:** http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel

```
<xs:complexType name="ReadOnlyCollectionOfMSOLicenseSuitePGX_Pb6b">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="list" nillable="true"
xmlns:q3="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObjectModel" type="q3:ArrayOfMSOLicenseSuite"/>
  </xs:sequence>
</xs:complexType>
```

**list:** The content of the read-only collection; it contains the list of MSOLicenseSuite instances.

### 2.2.4.62 RemoteConnectionUserInfo

The RemoteConnectionUserInfo type contains user information for a remote connection (see section 2.2.4.26).

**Namespace:** http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi

```
<xs:complexType name="RemoteConnectionUserInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="AccountName" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="FirstName" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="LastName" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="SidValue" nillable="true"
type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

**AccountName:** The account name of the user.

**FirstName:** The first name of the user.

**LastName:** The last name of the user.

**SidValue:** The SID of the user.



### 2.2.4.63 SearchItemList

The SearchItemList type is a collection of metadata from the search results.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebAPI>

```
<xs:complexType name="SearchItemList">
  <xs:complexContent mixed="false">
    <xs:extension base="tnsl:ItemList">
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="UnsearchedShares" nillable="true"
xmlns:q2="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
type="q2:ArrayOfstring"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

**UnsearchedShares:** A collection of top-level shared folder names that were not successfully searched.

### 2.2.4.64 ServerInfo

The ServerInfo type contains informational data about the server.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebAPI>

```
<xs:complexType name="ServerInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="IsMediaStreamingServiceAvailable"
type="xs:boolean"/>
    <xs:element minOccurs="0" maxOccurs="1" name="IsTsGatewayEnabled" type="xs:boolean"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ServerDomainGuid" nillable="true"
xmlns:q7="http://schemas.microsoft.com/2003/10/Serialization/" type="q7:guid"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ServerGuid" nillable="true"
xmlns:q8="http://schemas.microsoft.com/2003/10/Serialization/" type="q8:guid"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ServerInternetDomainName" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ServerName" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ServerSku" type="xs:int"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ServerUtcNow" type="DateTime"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ServerVersion" nillable="true"
type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

**IsMediaStreamingServiceAvailable:** A flag that indicates whether the media streaming service is available on the server.

**IsTsGatewayEnabled:** A flag that indicates whether the TsGateway service is available on the server.

**ServerDomainGuid:** The globally unique identifier (GUID) of the domain to which the server belongs.

**ServerGuid:** The GUID of the server.

**ServerInternetDomainName:** The Internet domain name of the server.

**ServerName:** The NetBIOS name of the server.

**ServerSku:** The Product Type of the hosting server.

**ServerUtcNow:** The current date and time on the server in coordinated universal time (UTC).

**ServerVersion:** The version of the server.

#### 2.2.4.65 SharePointSiteAddressCollection

The SharePointSiteAddressCollection type contains the SharePoint site address information.

**Namespace:**

<http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.O365Integration>

```
<xs:complexType name="SharePointSiteAddressCollection">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="MySiteAddress" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="TeamSiteAddress" nillable="true"
type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

**MySiteAddress:** The My Site address.

**TeamSiteAddress:** The SharePoint team site address.

#### 2.2.4.66 StorageDriveInfo

The StorageDriveInfo type contains informational data about the drive to be used to create a storage folder (see section 2.2.4.30).

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebAPI>

```
<xs:complexType name="StorageDriveInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="DeviceName" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="FileSystem" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ShortName" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Size" type="xs:unsignedLong"/>
    <xs:element minOccurs="0" maxOccurs="1" name="SizeRemaining" type="xs:unsignedLong"/>
    <xs:element minOccurs="0" maxOccurs="1" name="VolumeName" nillable="true"
type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

**DeviceName:** The name of the device to which the drive belongs.

**FileSystem:** The file system of the drive.

**ShortName:** The short name of the drive.

**Size:** The total size of the drive.

**SizeRemaining:** The free space of the drive.

**VolumeName:** The volume name of the drive.

## 2.2.4.67 StorageServerInfo

The StorageServerInfo type contains informational data about the server that can be used to create a storage folder (see section 2.2.4.30).

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebAPI>

```
<xs:complexType name="StorageServerInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="DeviceName" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="DfsNameSpace" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="DfsPhysicalPath" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="IsSecondServer" type="xs:boolean"/>
    <xs:element minOccurs="0" maxOccurs="1" name="OperatingSystemName" nillable="true"
type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

**DeviceName:** The name of the server.

**DfsNameSpace:** The **DFS** namespace of server. This SHOULD be NULL if the server is the second server.

**DfsPhysicalPath:** The DFS physical path of server. It SHOULD be NULL if the server is the second server.

**IsSecondServer:** A flag that indicates whether the server is second server.

**OperatingSystemName:** The name of the operating system of the server.

## 2.2.4.68 UserInfo

The UserInfo type contains informational data about the user.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebAPI>

```
<xs:complexType name="UserInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="AccessLevel" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="CanChangePassword" type="xs:boolean"/>
    <xs:element minOccurs="0" maxOccurs="1" name="FirstName" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Key" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="LastName" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="RemoteWebAccess" type="xs:boolean"/>
    <xs:element minOccurs="0" maxOccurs="1" name="UserGroups" nillable="true"
type="tns1:ArrayOfGroupInfo"/>
    <xs:element minOccurs="0" maxOccurs="1" name="UserGuid" nillable="true"
xmlns:q1="http://schemas.microsoft.com/2003/10/Serialization/" type="q1:guid"/>
    <xs:element minOccurs="0" maxOccurs="1" name="UserName" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="UserStatus" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="VpnAccess" type="xs:boolean"/>
  </xs:sequence>
</xs:complexType>
```

**AccessLevel:** The access level for the current user account, which SHOULD be one of the following values:

- Unknown
- Guest
- User
- Administrator

**CanChangePassword:** A flag that indicates whether the user's password can be changed.

**FirstName:** The first name of the user that is associated with the user account.

**Key:** The unique key of the user.

**LastName:** The last name of the user that is associated with the user account.

**RemoteWebAccess:** A flag that indicates whether the user has remote web access permission.

**UserGroups:** The list of groups to which the user belongs.

**UserGuid:** The GUID of the user.

**UserName:** The name of the user account.

**UserStatus:** The user status, which SHOULD be one of the following values:

- Disabled
- Enabled
- Error

**VpnAccess:** A flag that indicates whether the user has VPN access permission.

#### 2.2.4.69 WebApiProvider

The WebApiProvider type contains informational data about the WebApi provider.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi>

```
<xs:complexType name="WebApiProvider">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="ContractName" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ContractNameSpace" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="RelativeLocation" type="xs:anyURI"/>
  </xs:sequence>
</xs:complexType>
```

**ContractName:** The name of the provider.

**ContractNameSpace:** The namespace of the provider.

**RelativeLocation:** The location of the provider relative to the application root.

## 2.2.5 Simple Types

The following table summarizes the set of common XML schema simple type definitions defined by this specification.

Simple Type	Description
ContainerType	Describes the type of media container.
guid	Represents a GUID.
MSODomainTypes	Indicates the type of the domain for the online services.
MSOLicenseServiceTypes	Indicates the type of the license services.
Permission	Describes the user access permissions on a folder.
ServerFolderType	Describes the server folder type.
ThumbnailFormat	A class that specifies the preferred thumbnail image format.

### 2.2.5.1 ContainerType

The ContainerType describes the type of media container.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi>

```
<xs:simpleType name="ContainerType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Album"/>
    <xs:enumeration value="Artist"/>
    <xs:enumeration value="Playlist"/>
    <xs:enumeration value="Rating"/>
    <xs:enumeration value="Genre"/>
    <xs:enumeration value="Folder"/>
    <xs:enumeration value="Date"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for **ContainerType**.

Value	Meaning
Album	The type of the media container is "Album".
Artist	The type of the media container is "Artist".
Playlist	The type of the media container is "Playlist".
Rating	The type of the media container is "Rating".
Genre	The type of the media container is "Genre".
Folder	The type of the media container is "Folder".
Date	The type of the media container is "Date".

### 2.2.5.2 guid

This represents a GUID.

**Namespace:** <http://schemas.microsoft.com/2003/10/Serialization/>

```
<xs:simpleType name="guid">
  <xs:restriction base="xs:string">
    <xs:pattern value="\da-fA-F}{8}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{12}"/>
  </xs:restriction>
</xs:simpleType>
```

### 2.2.5.3 MSODomainTypes

Indicates the type of the domain for the online services.

**Namespace:**

<http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObjectModel>

```
<xs:simpleType name="MSODomainTypes">
  <xs:restriction base="xs:string">
    <xs:enumeration value="None"/>
    <xs:enumeration value="Email"/>
    <xs:enumeration value="SharePoint"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for **MSODomainTypes**.

Value	Meaning
None	The type of the domain is None.
Email	The type of the domain is email.
SharePoint	The type of the domain is SharePoint.

### 2.2.5.4 MSOLicenseServiceTypes

Indicates the type of the license services.

**Namespace:**

<http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObjectModel>

```
<xs:simpleType name="MSOLicenseServiceTypes">
  <xs:list>
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="None"/>
        <xs:enumeration value="Unknown"/>
        <xs:enumeration value="ExchangeOnline"/>
        <xs:enumeration value="SharePointOnline"/>
        <xs:enumeration value="LyncOnline"/>
        <xs:enumeration value="IntuneOnline"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:list>
</xs:simpleType>
```

```
</xs:list>
</xs:simpleType>
```

The following table specifies the allowable values for **MSOLicenseServiceTypes**.

Value	Meaning
None	The type of the license service is "None".
Unknown	The type of the license service is "Unknown".
ExchangeOnline	The type of the license service is "ExchangeOnline".
SharePointOnline	The type of the license service is "SharePointOnline".
LyncOnline	The type of the license service is "LyncOnline".
IntuneOnline	The type of the license service is "IntuneOnline".

### 2.2.5.5 Permission

This class describes the user access permissions on a folder.

**Namespace:** <http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.Storage>

```
<xs:simpleType name="Permission">
  <xs:restriction base="xs:string">
    <xs:enumeration value="None"/>
    <xs:enumeration value="ReadOnly"/>
    <xs:enumeration value="Full"/>
    <xs:enumeration value="Other"/>
    <xs:enumeration value="ReadPermissions"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for **Permission**.

Value	Meaning
None	No access permission.
ReadOnly	User has read-only permission on the server.
Full	User has full permission on the server.
Other	User has customized permission on the server.
ReadPermissions	User has a set of read permissions.

### 2.2.5.6 ServerFolderType

This class describes the server folder type.

**Namespace:**

<http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.WebApi.Management.Storage>

```
<xs:simpleType name="ServerFolderType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="NonPredefinedType"/>
    <xs:enumeration value="MusicType"/>
    <xs:enumeration value="PicturesType"/>
    <xs:enumeration value="DocumentsType"/>
    <xs:enumeration value="VideosType"/>
    <xs:enumeration value="BackupsType"/>
    <xs:enumeration value="FileBackupsType"/>
    <xs:enumeration value="FolderRedirectionType"/>
    <xs:enumeration value="CompanyType"/>
    <xs:enumeration value="UserType"/>
    <xs:enumeration value="OtherType"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for **ServerFolderType**.

Value	Meaning
NonPredefinedType	Non-predefined folder type.
MusicType	Music folder type.
PicturesType	Pictures folder type.
DocumentsType	Documents folder type.
VideosType	Videos folder type.
BackupsType	Backups folder type.
FileBackupsType	File backups folder type.
FolderRedirectionType	Folder redirection folder type.
CompanyType	Company folder type.
UserType	User folder type.
OtherType	System or special use folders type.

### 2.2.5.7 ThumbnailFormat

A class that specifies the preferred thumbnail image format.

**Namespace:** <http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi>

```
<xs:simpleType name="ThumbnailFormat">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Jpeg"/>
    <xs:enumeration value="Png"/>
  </xs:restriction>
</xs:simpleType>
```



The following table specifies the allowable values for **ThumbnailFormat**.

<b>Value</b>	<b>Meaning</b>
Jpeg	The thumbnail that presents as .jpg file.
Png	The thumbnail which presents as .png file.

## 3 Protocol Details

### 3.1 SessionService Server Details

#### 3.1.1 Abstract Data Model

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

##### 3.1.1.1 SessionId

The login call creates a session and generates a **SessionId**. This **SessionId** is returned to the client in a response cookie.

The logout call ends the session specified by the SessionId in the request cookie.

##### 3.1.1.2 Canary

The login call creates a **Canary** to indicate the user identity; it will be saved on the server. This **Canary** will be returned to the client in a response header.

The logout call disposes of the Canary token and removes it from the server.

#### 3.1.2 Timers

None.

#### 3.1.3 Initialization

At initialization, the protocol server MUST begin listening for requests at the respective URL addresses given in the message transport (as specified in section 2.1).

#### 3.1.4 Higher-Layer Triggered Events

None.

#### 3.1.5 Message Processing Events and Sequencing Rules

To call a web service, the client MUST perform these steps:

1. Add the caller's credential in an HTTP header field. The caller's credential is encoded text that is based on the user name and the password.
2. Add Appname (section 2.2.2.1), Apppublisher (section 2.2.2.2), and Appversion (section 2.2.2.3) values in the HTTP header fields. These values are also required to log on.
3. Call the logon method to retrieve the authenticated user token that is issued by the server.
4. Send subsequent web service requests with the user token in the header.

Resource	Description
login	Generates a valid user token for a valid user.
logout	Invalidates the session and user token.

The responses to all the operations can result in the following status codes.

Status code	Reason phrase	Description
200	OK	Operation successfully completed.
401	Unauthorized	Invalid user name or password.

### 3.1.5.1 login

Generates a valid user token for a valid user.

HTTP method	Description
GET	Generates a valid user token for a valid user.

#### 3.1.5.1.1 GET

Generates a valid user token for a valid user.

This operation is transported by an HTTP **GET**.

The operation can be invoked through the following URI:

```
https://<serverName>/services/builtin/session.svc/login
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Authorization		The caller's credential is encoded text that is based on user name and password.
AppName		The application name.
AppVersion		The application version.
AppPublisherName		The publisher of the application.

The response message for this operation contains the following HTTP header.

Response header	Usage	Value
Canary		The user token.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
401	Invalid user name or password.

### 3.1.5.1.1.1 Request Body

None.

### 3.1.5.1.1.2 Response Body

None.

### 3.1.5.1.1.3 Processing Details

1. The server MUST only respond to requests that have established TLS 1.1 server authentication as specified in [RFC4346].
2. The client MUST send requests with an authorization header, the value of which starts with "Basic" followed by a blank space and a base64 encoded string of user name and password separated by a colon.
3. The server MUST start a session and generate a user Canary token, as specified in section 2.2.2.5, for the client.
4. The user Canary token MUST be returned to client in Response header.

## 3.1.5.2 logout

Invalidate the session and user token.

HTTP method	Description
GET	Invalidate the session and user token.

### 3.1.5.2.1 GET

Invalidate the session and user token.

This operation is transported by an HTTP **GET**.

The operation can be invoked through the following URI:

```
https://<serverName>/services/builtin/Session.svc/logout
```

The request message for this operation contains the following HTTP header.

Request header	Usage	Value
Canary		The user token.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
401	Invalid user token.

#### **3.1.5.2.1.1 Request Body**

None.

#### **3.1.5.2.1.2 Response Body**

None.

#### **3.1.5.2.1.3 Processing Details**

The server **MUST** only respond to requests that have established TLS 1.1 server authentication [RFC4346].

The user **SHOULD** log out of the API explicitly when the application exits for security purposes. The server ends the session specified in the request.

#### **3.1.6 Timer Events**

None.

#### **3.1.7 Other Local Events**

None.

### **3.2 IFileContentAccessService Server Details**

#### **3.2.1 Abstract Data Model**

None.

#### **3.2.2 Timers**

None.

#### **3.2.3 Initialization**

At initialization time, the protocol server **MUST** begin listening for requests at the respective URL addresses given in the message transport (as specified in section 2.1).

The storage provider **MUST** be initialized and configured correctly.

#### **3.2.4 Higher-Layer Triggered Events**

None.

### 3.2.5 Message Processing Events and Sequencing Rules

Resource	Description
filecontent?path={path}	The file resource can be downloaded.
filecontent?path={path}&overwrite={overwrite}	The file resource will be used to upload.
itemthumbnail?path={path}	The thumbnail of the file item.

The responses to all the operations can result in the following status codes.

Status code	Reason phrase	Description
200	OK	The operation completed successfully.
400	Bad Request	Bad request, the path too long or missing, or the path contains unsupported characters.
401	Unauthorized	Access denied to requested resource.
404	Not Found	The requested resource does not exist.
201	Created	File uploading successfully completed.
409	Conflict	Attempted to create a resource that already exists.

#### 3.2.5.1 filecontent?path={path}

Retrieves the binary content of a file in bytes.

**path:** This is the description of the parameter path.

HTTP method	Description
GET	Retrieves the binary content of a file in bytes.

##### 3.2.5.1.1 GET

This API retrieves the binary content of a file in bytes. The caller reads the response stream. The API supports retrieving partial content of the file range that is specified by the Range header in an HTTP request.

This operation is transported by an HTTP **GET**.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/FileContentAccessService.svc/filecontent?path={path}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Range	Optional	Request only part of an entity, as specified in section 2.2.2.7.
Canary	Optional	The user token, as specified in section 2.2.2.5. Client MUST either pass this header to server or pass all of the following headers to the server.
Authorization	Optional	The caller's credentials, which are encoded text that is based on the user name and password as specified in section 2.2.2.4.
AppName	Optional	The client application name, as specified in section 2.2.2.1.
AppVersion	Optional	The client application version, as specified in section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name, as specified in section 2.2.2.2.

The response message for this operation can result in one of the following status codes.

Status code	Description
200	Download operation successfully completed.
400	Input parameters are invalid.
401	User is not allowed to access resource.
404	Requested resource does not exist.

### 3.2.5.1.1.1 Request Body

None.

### 3.2.5.1.1.2 Response Body

The response body is encoded in the following XML format.

```
<xs:element name="Stream" nillable="true" type="xs:base64Binary"/>
```

### 3.2.5.1.1.3 Processing Details

This operation downloads a file in the form of a file stream from the location specified by the **path**.

### 3.2.5.2 filecontent?path={path}&overwrite={overwrite}

Uploads a file to a location for which the user has access permissions.

**path:** The path of the file.

**overwrite:** This flag indicates whether to overwrite the file if it exists on server. See section 2.2.3.30.

HTTP method	Description
POST	Uploads a file to a location where the user has access permissions.

### 3.2.5.2.1 POST

The file content SHOULD be put into the HTTP request. The protocol does not support resuming uploads.

By default, the maximum size of a file that can be uploaded is 100 MB.

This operation is transported by an HTTP **POST**.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/FileContentAccessService.svc/filecontent?path={path}&overwrite={overwrite}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The value returned in the logon Response Canary header. The client MUST either pass this header to the server or pass all of the following headers to the client. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on a user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The application name, as specified in section 2.2.2.1.
AppVersion	Optional	The application version, as specified in section 2.2.2.3.
AppPublisherName	Optional	The application publisher name, as specified in section 2.2.2.2.

The response message for this operation can result in one of the following status codes.

Status code	Description
201	File uploading successfully completed.
401	User is not allowed to access resource.
404	The specific resource does not exist.
409	An item already exists with the requested new name.

#### 3.2.5.2.1.1 Request Body

The request body is a binary stream.

#### 3.2.5.2.1.2 Response Body

None.

#### 3.2.5.2.1.3 Processing Details

This operation uploads a file to the location specified by the **path**.



### 3.2.5.3 itemthumbnail?path={path}

Retrieves a thumbnail of a single item.

**path:**

HTTP method	Description
GET	Retrieves a thumbnail of a single item.

#### 3.2.5.3.1 GET

Retrieves a thumbnail of a single item. This method supports most image formats.

This operation is transported by an HTTP **GET**.

The operation can be invoked through the following URI:

```
https://<servername>/services/builtin/FileContentAccessService.svc/itemthumbnail?path={path}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The value returned in logon Response canary header. Client MUST either pass this header to server or pass all of the following headers to the client. For more details, see section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, see section 2.2.2.4.
AppName	Optional	The application name. For more details, see section 2.2.2.1.
AppVersion	Optional	The application version. For more details, see section 2.2.2.3.
AppPublisherName	Optional	The application publisher name. For more details, see section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied.
404	The path does not exist.

#### 3.2.5.3.1.1 Request Body

None.

#### 3.2.5.3.1.2 Response Body

The response body is encoded in either XML format or JSON format. The format is controlled by the Accept header defined in section 2.2.2.6. For the detailed definition of response body, refer to section 2.2.4.34.

```
<xs:element name="ItemThumbnail" nillable="true" type="tnsl:ItemThumbnail"/>
```

### 3.2.5.3.1.3 Processing Details

This call will return the item thumbnail to the client.

The server MUST construct a response in either XML or JSON format based on the value of the Accept header (section 2.2.2.6), or in XML format if the Accept header is not present. The response MUST include all of the complex types defined in section 2.2.4.34.

### 3.2.6 Timer Events

None.

### 3.2.7 Other Local Events

None.

## 3.3 IFileOperationService Server Details

### 3.3.1 Abstract Data Model

None.

### 3.3.2 Timers

None.

### 3.3.3 Initialization

At initialization time, the protocol server MUST begin listening for requests at the respective URL addresses given in the message transport (as specified in section 2.1).

The storage provider MUST be initialized and configured correctly.

### 3.3.4 Higher-Layer Triggered Events

None.

### 3.3.5 Message Processing Events and Sequencing Rules

Resource	Description
items/index/{index}/count/{count}?path={path}&filter={filter}&sortByfield={sortByfield}&ascending={ascending}	ItemList object that represents a collection of file

Resource	Description
	items in the specified folder.
items/index/{index}/count/{count}/search?query={query}&orderbyfield={orderbyfield}&ascending={ascending}&scope={scope}&timeoutseconds={timeoutseconds}	SearchItem object that represents a collection of file items that match the specified keyword.
folder/newsubfoldername?path={path}&language={language}	Default available new folder name in specified language.
itemmetadata?path={path}	ItemInfo object that represents the specified file item.
folder?path={path}	Folder being created.
item/rename?path={path}&newname={newname}	File being renamed.
item/delete?path={path}	File item being deleted.
accessuri?path={path}	Temporary URI that can be used to download a file.
item/move?newpath={newpath}&originalpath={originalpath}	File item being moved from

Resource	Description
	another location.
item/copy?newpath={newpath}&originalpath={originalpath}	File item being copied from another location.

The responses to all the operations can result in the following status codes.

Status code	Reason phrase	Description
200	OK	The operation successfully completed.
400	Bad Request	One or more parameters are not valid.
401	Unauthorized	Access to the requested resource is denied.
404	Not Found	The path does not exist.
501	Not Implemented	The server does not have the functionality to support the requested operation.
503	Service Unavailable	The server is currently unable to handle the request.
504	Gateway Timeout	The request does not finish within a given time interval.
201	Created	The request resource is created.
409	Conflict	The request fails due to a conflict with the current resource.

### 3.3.5.1 items/index/{index}/count/{count}?path={path}&filter={filter}&sortbyfield={sortbyfield}&ascending={ascending}

The URI is an ItemList object that represents a collection of file items within the specified folder.

**index:** The numerical position of the first item to be retrieved (see section 2.2.3.18).

**count:** The number of items to be retrieved (see section 2.2.3.5).

**path:** The UNC path of the folder (see section 2.2.3.32).

**filter:** The type of files to be retrieved (see section 2.2.3.11). All file types will be retrieved if this parameter is ignored.

**sortbyfield:** The field of the file item property by which file items are sorted (see section 2.2.3.42).

**ascending:** A Boolean value that indicates whether the sorting is ascending (see section 2.2.3.3). Ascending sorting is performed if this parameter is ignored.

HTTP method	Description
GET	Retrieves the metadata information of items in a folder.

### 3.3.5.1.1 GET

The GET method retrieves the metadata information of items under a folder. The metadata includes basic information such as the creation time, item type, modification time, name, path, and size.

This operation is transported by an HTTP **GET**.

The operation can be invoked through the following URI:

```
items/index/{index}/count/{count}?path={path}&filter={filter}&sortByfield={sortByfield}&ascending={ascending}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The value returned in the logon Response Canary header. The client MUST either pass this header to the server or pass all of following headers to the client. For more details, see section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, see section 2.2.2.4.
AppName	Optional	The application name. For more details, see section 2.2.2.1.
AppVersion	Optional	The application version. For more details, see section 2.2.2.3.
AppPublisherName	Optional	The name of the application publisher. For more details, see section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	The operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied.
404	The path does not exist.

#### 3.3.5.1.1.1 Request Body

None.

#### 3.3.5.1.1.2 Response Body

The response body is encoded in XML or JSON format according to the ACCEPT header of the request.

```
<xs:element name="ItemList" nillable="true" type="tnsl:ItemList"/>
```

### 3.3.5.1.1.3 Processing Details

This operation enumerates the file items in the specified folder and returns a collection of metadata that fulfils the specified parameters.

### 3.3.5.2 items/index/{index}/count/{count}/search?query={query}&sortByfield={sortByfield}&ascending={ascending}&scope={scope}&timeoutseconds={timeoutseconds}

This URI represents a search method that returns a SearchItemList object. The SearchItemList object represents a collection of file items that match the specified keyword.

**index:** The numerical position of the first item to be retrieved (see section 2.2.3.18).

**count:** The number of items to be retrieved (see section 2.2.3.5).

**query:** The start string to search for within the names of the files and folders. The asterisk (\*) and question mark (?) wildcard characters are supported (see section 2.2.3.34).

**sortByfield:** The field of the file item property by which file items are sorted (see section 2.2.3.42).

**ascending:** A Boolean value that indicates whether the sorting is ascending (see section 2.2.3.3). If this parameter is ignored, ascending sorting is applied.

**scope:** The UNC path of the folder where the search is performed (see section 2.2.3.37). All shared folders are searched if this parameter is ignored.

**timeoutseconds:** The number of seconds after which the request times out (see section 2.2.3.47). Timeout interval is set to 10 seconds if this parameter is ignored.

HTTP method	Description
GET	Retrieves the metadata information for search results.

#### 3.3.5.2.1 GET

The GET method retrieves the metadata information for search results. The metadata includes basic information such as the creation time, item type, modification time, name, path, and size. The search occurs across all share folders that users have access to.

This operation is transported by an HTTP **GET**.

The operation can be invoked through the following URI:

```
items/index/{index}/count/{count}/search?query={query}&sortByfield={sortByfield}&ascending={ascending}&scope={scope}&timeoutseconds={timeoutseconds}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The value returned in login Response Canary header.

Request header	Usage	Value
		Client MUST either pass this header to the server or pass all of the following headers to the client. For more details, see section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, see section 2.2.2.4.
AppName	Optional	Application name. For more details, see section 2.2.2.1.
AppVersion	Optional	Application version. For more details, see section 2.2.2.3.
AppPublisherName	Optional	Application publisher name. For more details, see section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied.
501	Windows search service is not available.
503	Windows search service failed to perform the search.
504	Search does not finish within the specified timeout interval.

### 3.3.5.2.1.1 Request Body

None.

### 3.3.5.2.1.2 Response Body

The response body is encoded in XML or JSON format according to the ACCEPT header of the request.

```
<xs:element name="SearchItemList" nillable="true" type="tns1:SearchItemList"/>
```

### 3.3.5.2.1.3 Processing Details

This operation searches file items in the specified folder and returns the collection of metadata that fulfills the specified parameters.

### 3.3.5.3 folder/newsubfoldername?path={path}&language={language}

This URI represents the unique name to use for a new folder.

**path:** The UNC path (specified in section 2.2.3.32) for the new folder.

**language:** The language/region code (such as en-us) (specified in section 2.2.3.21) of the new folder.

HTTP method	Description
GET	Generates an unused name to use for a new folder.

### 3.3.5.3.1 GET

The GET operation generates a unique name to use for a new folder.

This operation is transported by an HTTP **GET**.

The operation can be invoked through the following URI:

```
folder/newsubfoldername?path={path}&language={language}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The value returned in the logon Response Canary header. Client MUST either pass this header to the server or pass all of the following headers to the client. For more details, see section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, see section 2.2.2.4.
AppName	Optional	Application name. For more details, see section 2.2.2.1.
AppVersion	Optional	Application version. For more details, see section 2.2.2.3.
AppPublisherName	Optional	Application publisher name. For more details, see section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied.
404	Folder specified by the path does not exist.

#### 3.3.5.3.1.1 Request Body

None.

#### 3.3.5.3.1.2 Response Body

The response body is encoded in XML or JSON format according to the ACCEPT header of the request.

```
<xs:element name="string" nillable="true" type="xs:string"/>
```

#### 3.3.5.3.1.3 Processing Details

This operation retrieves the next available new folder name with the predefined pattern in the specified folder.



### 3.3.5.4 itemmetadata?path={path}

This URI represents the basic metadata of a single item.

**path:** The UNC path (section 2.2.3.32) of the file item.

HTTP method	Description
GET	Retrieves the basic metadata of a single item.

#### 3.3.5.4.1 GET

The GET operation retrieves the basic metadata of a single item.

This operation is transported by an HTTP **GET**.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/fileoperationsservice.svc/itemmetadata?path={path}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The value returned in the logon Response Canary header. Client MUST either pass this header to the server or pass all of the following headers to the client. For more details, see section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, see section 2.2.2.4.
AppName	Optional	Application name. For more details, see section 2.2.2.1.
AppVersion	Optional	Application version. For more details, see section 2.2.2.3.
AppPublisherName	Optional	Application publisher name. For more details, see section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied.
404	The path does not exist.

#### 3.3.5.4.1.1 Request Body

None.

#### 3.3.5.4.1.2 Response Body

The response body is encoded in XML or JSON format according to ACCEPT header of the request.

```
<xs:element name="ItemInfo" nillable="true" type="tns1:ItemInfo"/>
```

### 3.3.5.4.1.3 Processing Details

This operation returns the metadata of the specified file item.

### 3.3.5.5 folder?path={path}

This URI represents a new folder in a path where the user has the required permissions.

**path:**The UNC path (section 2.2.3.32) of the new folder.

HTTP method	Description
POST	Creates a new folder in a path where the user has the required permissions.

### 3.3.5.5.1 POST

The POST operation creates a new folder in a path where the user has the required permissions.

This operation is transported by an HTTP POST.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/fileoperationservice.svc/folder?path={path}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The value returned in the logon Response Canary header. Client MUST either pass this header to the server or pass all of the following headers to the client. For more details, see section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, see section 2.2.2.4.
AppName	Optional	Application name. For more details, see section 2.2.2.1.
AppVersion	Optional	Application version. For more details, see section 2.2.2.3.
AppPublisherName	Optional	Application publisher name. For more details, see section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
201	The folder is created.
400	One or more parameters are not valid.
401	Access to the requested resource is denied.

Status code	Description
409	File item with the new name exists.

### 3.3.5.5.1.1 Request Body

None.

### 3.3.5.5.1.2 Response Body

The response body is encoded in XML or JSON format according to the ACCEPT header of the request.

### 3.3.5.5.1.3 Processing Details

This operation creates a new folder with the given name in the specified folder.

### 3.3.5.6 item/rename?path={path}&newname={newname}

This URI represents the name of an existing item.

**path:** UNC path of the file item to be renamed (see section 2.2.3.32).

**newname:** The new name of the item (see section 2.2.3.26).

HTTP method	Description
POST	Changes the name of an existing item.

#### 3.3.5.6.1 POST

The POST operation changes the name of an existing item.

This operation is transported by an HTTP **POST**.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/fileoperationservice.svc/item/rename?path={path}&newname={newname}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The value returned in the logon Response Canary header. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, see section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, see section 2.2.2.4.
AppName	Optional	Application name. For more details, see section 2.2.2.1.
AppVersion	Optional	Application version. For more details, see section 2.2.2.3.

Request header	Usage	Value
AppPublisherName	Optional	Application publisher name. For more details, see section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied.
404	The path does not exist.
409	File item with the new name exists.

### 3.3.5.6.1.1 Request Body

None.

### 3.3.5.6.1.2 Response Body

None.

### 3.3.5.6.1.3 Processing Details

This operation changes name of the specified file item.

### 3.3.5.7 item/delete?path={path}

This URI represents an existing item in a path where the user has the required permissions to delete.

**path:** UNC path of the file to be deleted (see section 2.2.3.32).

HTTP method	Description
POST	Deletes an existing item in a path where the user has the required permissions. If the path is a folder, the method also removes all items in the folder.

#### 3.3.5.7.1 POST

The POST method deletes an existing item in a path where the user has the required permissions. If the path is a folder, the method also removes all items in the folder.

This operation is transported by an HTTP **POST**.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/fileoperationservice.svc/item/delete?path={path}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The value returned in the logon Response Canary header. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, see section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, see section 2.2.2.4.
AppName	Optional	Application name. For more details, see section 2.2.2.1.
AppVersion	Optional	Application version. For more details, see section 2.2.2.3.
AppPublisherName	Optional	Application publisher name. For more details, see section 2.2.2.2.

### 3.3.5.7.1.1 Request Body

None.

### 3.3.5.7.1.2 Response Body

None.

### 3.3.5.7.1.3 Processing Details

This operation deletes the specified file item.

### 3.3.5.8 accessuri?path={path}

This URI represents a temporary URL for a file.

**path:** The UNC path of the file (see section 2.2.3.32).

HTTP method	Description
GET	Retrieves a temporary URL for a file.

#### 3.3.5.8.1 GET

The **GET** method retrieves a temporary URL for a file. The URL can be accessed only twice and only within 30 seconds after retrieval.

This operation is transported by an HTTP GET.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/fileoperationservice.svc/accessuri?path={path}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The value returned in the logon Response Canary header. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, see section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, see section 2.2.2.4.
AppName	Optional	Application name. For more details, see section 2.2.2.1.
AppVersion	Optional	Application version. For more details, see section 2.2.2.3.
AppPublisherName	Optional	Application publisher name. For more details, see section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied.
404	The path does not exist.

### 3.3.5.8.1.1 Request Body

None.

### 3.3.5.8.1.2 Response Body

The response body is encoded in XML or JSON format according to the ACCEPT header of the request.

```
<xs:element name="anyURI" type="xs:anyURI"/>
```

### 3.3.5.8.1.3 Processing Details

This operation retrieves a temporary URL for a file. The URL can be accessed only twice and only within 30 seconds after retrieval.

### 3.3.5.9 item/move?newpath={newpath}&originalpath={originalpath}

This URI represents a file item that can be moved to another location.

**originalpath:** UNC path of the file item (see section 2.2.3.29).

**newpath:** The new UNC path of the file item (see section 2.2.3.27).

HTTP method	Description
POST	Move a file item to another location.

### 3.3.5.9.1 POST

The POST method moves a file item to another location.

This operation is transported by an HTTP **POST**.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/fileoperationservice.svc/item/move?newpath={newpath}&originalpath={originalpath}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The value returned in the logon Response Canary header. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, see section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, see section 2.2.2.4.
AppName	Optional	Application name. For more details, see section 2.2.2.1.
AppVersion	Optional	Application version. For more details, see section 2.2.2.3.
AppPublisherName	Optional	Application publisher name. For more details, see section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied.
404	The path does not exist.
409	File item with the new path exists.

#### 3.3.5.9.1.1 Request Body

None.

#### 3.3.5.9.1.2 Response Body

None.

#### 3.3.5.9.1.3 Processing Details

This operation moves a file item to another location.

### 3.3.5.10 item/copy?newpath={newpath}&originalpath={originalpath}

This URI represents a file item that can be copied to another location.

**originalpath:** UNC path of the file item (see section 2.2.3.29).

**newpath:** The new UNC path of the file item (see section 2.2.3.27).

HTTP method	Description
POST	Copy a file item to another location.

#### 3.3.5.10.1 POST

The **POST** method copies a file item to another location.

This operation is transported by an HTTP **POST**.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/fileoperationservice.svc/item/copy?newpath={newpath}&originalpath={originalpath}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The value returned in the logon Response Canary header. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, see section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, see section 2.2.2.4.
AppName	Optional	Application name. For more details, see section 2.2.2.1.
AppVersion	Optional	Application version. For more details, see section 2.2.2.3.
AppPublisherName	Optional	Application publisher name. For more details, see section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied.
404	The path does not exist.
409	File item with the new path has existed.

#### 3.3.5.10.1.1 Request Body



None.

#### **3.3.5.10.1.2 Response Body**

None.

#### **3.3.5.10.1.3 Processing Details**

This operation copies the file item to another location.

#### **3.3.6 Timer Events**

None.

#### **3.3.7 Other Local Events**

None.

### **3.4 IAzureADManagement Server Details**

#### **3.4.1 Abstract Data Model**

This section describes a conceptual service contract that an implementation maintains in order to participate in this protocol. The described operations of the service contract are provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this service contract as long as their external behavior is consistent with how it is described in this document.

##### **3.4.1.1 MSOUser**

The data model uses the full complex types defined in section 2.2.4.50.

##### **3.4.1.2 MSOTenantInformation**

The data model uses the full complex types defined in section 2.2.4.49.

##### **3.4.1.3 ArrayOfMSODomain**

The data model uses the full complex types defined in section 2.2.4.13.

#### **3.4.2 Timers**

None.

#### **3.4.3 Initialization**

None.

#### **3.4.4 Higher-Layer Triggered Events**

None.

### 3.4.5 Message Processing Events and Sequencing Rules

Resource	Description
msouser?localusername={localusername}	Get the mapped Microsoft online user's information.
msousers	Get all Microsoft online users.
msouser/create?localusername={localusername}&onlineusername={onlineusername}	Create a Microsoft online user.
msouser/assign?localusername={localusername}&onlineusername={onlineusername}	Assign a Microsoft online user to local user.
msouser/unassign?localusername={localusername}	Un-assign the Microsoft online user from local user.
msouser/enable?localusername={localusername}	Enable the Microsoft online user.
msouser/disable?localusername={localusername}	Disable the Microsoft online user.
msouser/delete?localusername={localusername}	Delete the Microsoft online user.
msodomains	Get all accepted Microsoft online domains.
msosubscriptioninfo	Get Microsoft online subscription information.
msolicense/set?localusername={localusername}	Set Microsoft online account license.

The responses to all the operations can result in the following status codes.

Status code	Reason phrase	Description
200	OK	The operation successfully completed.
400	Bad Request	Invalid parameter.
401	Unauthorized	The user is not authorized for web access.
404	Not Found	Cannot find the local user or online user.
500	Internal server error	An internal server error occurred.
409	Conflict	The operation failed because of the status conflict.
501	Not Implemented	The operation is not supported.
403	Forbidden	The operation is forbidden.

### 3.4.5.1 msouser?localusername={localusername}

Retrieves the mapped online user.

**localusername:** Local user name.

HTTP method	Description
GET	Retrieve the mapped online user.

#### 3.4.5.1.1 GET

This API retrieves the online user mapped to the local user.

This operation is transported by an HTTP **GET**.

The operation can be invoked through the following URI:

```
https://<serverName>/services/builtin/AzureADManagement.svc/msouser?localusername={localusername}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	Invalid parameter.
401	The user is not authorized for web access.
404	Cannot find the local user or online user.
500	Internal server error.

#### 3.4.5.1.1.1 Request Body

None.

### 3.4.5.1.1.2 Response Body

The response body is encoded in either XML or JSON format. The format is controlled by the Accept header specified in section 2.2.4.50.

```
<xs:element name="MSOUser" nillable="true"
xmlns:q3="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObjectModel" type="q3:MSOUser"/>
```

### 3.4.5.1.1.3 Processing Details

This call returns the information of the online user. See section 2.2.4.50.

The server MUST construct a response in either XML or JSON format based on the value of the Accept header (section 2.2.2.6), or in XML format if the Accept header is not present.

### 3.4.5.2 msousers

Retrieves all of the online users.

HTTP method	Description
GET	Retrieves all of the online users.

#### 3.4.5.2.1 GET

This API retrieves all of the online users.

This operation is transported by an HTTP GET.

The operation can be invoked through the following URI:

```
https://<serverName>/services/builtin/AzureADManagement.svc/msousers
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.

Status code	Description
400	Invalid parameter.
401	The user is not authorized for web access.
500	Internal server error.

### 3.4.5.2.1.1 Request Body

None.

### 3.4.5.2.1.2 Response Body

The response body is encoded in either XML or JSON format. The format is controlled by the Accept header defined in section 2.2.4.18.

```
<xs:element name="ArrayOfMSOUser" nillable="true"
xmlns:q4="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObjectModel" type="q4:ArrayOfMSOUser"/>
```

### 3.4.5.2.1.3 Processing Details

This call returns the information of all the online users. See section 2.2.4.18.

The server MUST construct a response in either XML or JSON format based on the value of the Accept header (section 2.2.2.6), or in XML format if the Accept header is not present.

### 3.4.5.3 msouser/create?localusername={localusername}&onlineusername={onlineusername}

[CreateCreates](#) an online user and [assignassigns](#) to the local user.

**localusername:** Local user name.

**onlineusername:** Online user name, [shouldrecommended to](#) be a **UPN**.

HTTP method	Description
POST	Create an online user and assign to the local user.

#### 3.4.5.3.1 POST

This API creates an online user and assigns to the local user.

This operation is transported by an HTTP **POST**.

The operation can be invoked through the following URI:

```
https://<serverName>/services/builtin/AzureADManagement.svc/msouser/create?localusername={localusername}&onlineusername={onlineusername}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The value returned in the logon Response Canary header. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	Application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	Application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	Application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	Invalid parameter.
401	The user is not authorized for web access.
404	Cannot find the local user or online user.
409	Operation failed because of the status conflict.
500	Internal server error.

#### 3.4.5.3.1.1 Request Body

The request body is encoded in XML format or JSON format based on the value of the Accept header (section 2.2.2.6), or in XML format if the Accept header is not present.

#### 3.4.5.3.1.2 Response Body

None.

#### 3.4.5.3.1.3 Processing Details

This API creates an online user and assigns to the local user.

#### 3.4.5.4 msouser/assign?localusername={localusername}&onlineusername={onlineusername}

This API assigns the online user to the local user.

**localusername:** Local user name.

**onlineusername:** Online user name.

HTTP method	Description
POST	This API will assign the online user to the local user.

### 3.4.5.4.1 POST

This API assigns the online user to the local user.

This operation is transported by an HTTP **POST**.

The operation can be invoked through the following URI:

```
https://<serverName>/services/builtin/AzureADManagement.svc/msouser/assign?localusername={localusername}&onlineusername={onlineusername}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The value returned in the logon Response Canary header. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	Application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	Application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	Application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	Invalid parameter.
401	The user is not authorized for web access.
404	Cannot find the local user or online user.
409	Operation failed because of the status conflict.
500	Internal server error.
501	Operation is not supported.

#### 3.4.5.4.1.1 Request Body

None.

#### 3.4.5.4.1.2 Response Body

None.

### 3.4.5.4.1.3 Processing Details

This API reassigns the online user to the local user.

### 3.4.5.5 msouser/unassign?localusername={localusername}

This API reassigns the online user to the local user.

**localusername:**

HTTP method	Description
POST	This API reassigns the online user to the local user.

#### 3.4.5.5.1 POST

This API reassigns the online user to the local user.

This operation is transported by an HTTP **POST**.

The operation can be invoked through the following URI:

```
https://<serverName>/services/builtin/AzureADManagement.svc/msouser/unassign?localusername={localusername}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The value returned in the logon Response Canary header. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	Application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	Application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	Application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	Invalid parameter.
401	The user is not authorized for web access.
404	Cannot find the local user or online user.



Status code	Description
409	Operation failed because of the status conflict.
500	Internal server error.
501	Operation is not supported.

### 3.4.5.5.1.1 Request Body

None.

### 3.4.5.5.1.2 Response Body

None.

### 3.4.5.5.1.3 Processing Details

This API reassigns the online user to the local user.

### 3.4.5.6 msouser/enable?localusername={localusername}

This API enables the online user, which is mapped to the local user.

**localusername:**local user name

HTTP method	Description
POST	This API enables the online user, which is mapped to the local user.

### 3.4.5.6.1 POST

This API enables the online user, which is mapped to the local user.

This operation is transported by an HTTP **POST**.

The operation can be invoked through the following URI:

```
https://<serverName>/services/builtin/AzureADManagement.svc/msouser/enable?localusername={localusername}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The value returned in the logon Response Canary header. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	Application name. For more details, refer to section 2.2.2.1.

Request header	Usage	Value
AppVersion	Optional	Application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	Application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	Invalid parameter.
401	The user is not authorized for web access.
404	Cannot find the local user or online user.
500	Internal server error.
501	Operation is not supported.

#### 3.4.5.6.1.1 Request Body

None.

#### 3.4.5.6.1.2 Response Body

None.

#### 3.4.5.6.1.3 Processing Details

This API enables the online user, which is mapped to the local user.

#### 3.4.5.7 msouser/disable?localusername={localusername}

This API disables the online user, which is mapped to the local user.

**localusername:** Local user name.

HTTP method	Description
POST	This API disables the online user, which is mapped to the local user.

#### 3.4.5.7.1 POST

This API disables the online user, which is mapped to the local user.

This operation is transported by an HTTP **POST**.

The operation can be invoked through the following URI:

```
https://<serverName>/services/builtin/AzureADManagement.svc/msouser/disable?localusername={localusername}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The value returned in the logon Response Canary header. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	Application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	Application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	Application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	Invalid parameter.
401	The user is not authorized for web access.
403	Operation is forbidden.
404	Cannot find the local user or online user.
500	Internal server error.
501	Operation is not supported.

### 3.4.5.7.1.1 Request Body

None.

### 3.4.5.7.1.2 Response Body

None.

### 3.4.5.7.1.3 Processing Details

This API disables the online user, which is mapped to the local user.

### 3.4.5.8 msouser/delete?localusername={localusername}

This API deletes the online user, which is mapped to the local user.

**localusername:**

HTTP method	Description
POST	This API deletes the online user, which is mapped to the local user.

### 3.4.5.8.1 POST

This API deletes the online user, which is mapped to the local user.

This operation is transported by an HTTP **POST**.

The operation can be invoked through the following URI:

```
https://<serverName>/services/builtin/AzureADManagement.svc/msouser/delete?localusername={localusername}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The value returned in the logon Response Canary header. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	Application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	Application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	Application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	Invalid parameter.
401	The user is not authorized for web access.
403	Operation is forbidden.
404	Cannot find the local user or online user.
500	Internal server error.

#### 3.4.5.8.1.1 Request Body

None.

#### 3.4.5.8.1.2 Response Body

None.

#### 3.4.5.8.1.3 Processing Details

This API deletes the online user, which is mapped to the local user.

### 3.4.5.9 msodomains

This API retrieves all the accepted online domains.

HTTP method	Description
GET	This API retrieves all the accepted online domains.

#### 3.4.5.9.1 GET

This API retrieves all the accepted online domains.

This operation is transported by an HTTP **GET**.

The operation can be invoked through the following URI:

```
https://<serverName>/services/builtin/AzureADManagement.svc/msodomains
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The value returned in the logon Response Canary header. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	Application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	Application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	Application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	Invalid parameter.
401	The user is not authorized for web access.
500	Internal server error.

#### 3.4.5.9.1.1 Request Body

None.

#### 3.4.5.9.1.2 Response Body

The response body is encoded in either XML or JSON format. The format is controlled by the Accept header defined in section 2.2.4.13.

```
<xs:element name="ArrayOfMSODomain" nillable="true"
xmlns:q5="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObjectModel" type="q5:ArrayOfMSODomain"/>
```

### 3.4.5.9.1.3 Processing Details

This call returns the information of the accepted online domains. See section 2.2.4.13.

The server MUST construct a response in either XML or JSON format based on the value of the Accept header (section 2.2.2.6), or in XML format if the Accept header is not present.

### 3.4.5.10 msosubscriptioninfo

This API retrieves the online subscription information.

HTTP method	Description
GET	This API retrieves the online subscription information.

#### 3.4.5.10.1 GET

This API retrieves the online subscription information.

This operation is transported by an HTTP **GET**.

The operation can be invoked through the following URI:

```
https://<serverName>/services/builtin/AzureADManagement.svc/msosubscriptioninfo
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The value returned in the logon Response Canary header. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	Application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	Application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	Application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.

Status code	Description
400	Invalid parameter.
401	The user is not authorized for web access.
500	Internal server error.

### 3.4.5.10.1.1 Request Body

None.

### 3.4.5.10.1.2 Response Body

The response body is encoded in either XML or JSON format. The format is controlled by the Accept header defined in section 2.2.4.49.

```
<xs:element name="MSOTenantInformation" nillable="true"
xmlns:q6="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObjectModel" type="q6:MSOTenantInformation"/>
```

### 3.4.5.10.1.3 Processing Details

This call returns the information of the online subscription. See section 2.2.4.49.

The server MUST construct a response in either XML or JSON format based on the value of the Accept header (section 2.2.2.6), or in XML format if the Accept header is not present.

## 3.4.5.11 msolicense/set?localusername={localusername}

This API sets the license to the online user, which is mapped to the local user.

**localusername:** Local user name.

HTTP method	Description
POST	This API sets the license to the online user, which is mapped to the local user.

### 3.4.5.11.1 POST

This API sets the license to the online user, which is mapped to the local user.

This operation is transported by an HTTP **POST**.

The operation can be invoked through the following URI:

```
https://<serverName>/services/builtin/AzureADManagement.svc/msolicense/set?localusername={localusername}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The value returned in the logon Response Canary header. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	Application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	Application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	Application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	Invalid parameter.
401	The user is not authorized for web access.
404	Cannot find the local user or online user.
500	Internal server error.

#### 3.4.5.11.1.1 Request Body

The request body is encoded in XML format.

```
<xs:element name="ArrayOfMSOLicense" nillable="true" type="tns6:ArrayOfMSOLicense"/>
```

#### 3.4.5.11.1.2 Response Body

None.

#### 3.4.5.11.1.3 Processing Details

This API sets the license to the online user, which is mapped to the local user.

#### 3.4.6 Timer Events

None.

#### 3.4.7 Other Local Events

None.



## 3.5 IMailboxManagement Server Details

### 3.5.1 Abstract Data Model

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

#### 3.5.1.1 Mailbox

The data model uses full complex types defined in section 2.2.4.36.

#### 3.5.1.2 ArrayOfMailbox

The data model uses full complex types defined in section 2.2.4.10.

### 3.5.2 Timers

None.

### 3.5.3 Initialization

None.

### 3.5.4 Higher-Layer Triggered Events

None.

### 3.5.5 Message Processing Events and Sequencing Rules

Resource	Description
mailbox?username={username}	Retrieve the user's mailbox information.
mailbox/create?username={username}&email={email}	Create a mailbox for a user by using the default profile.
mailbox/set?username={username}&email={email}	Assign an existing mailbox to a user.
mailbox/unset?username={username}	Detach the mailbox from the user.
mailbox/disable?username={username}	Disable the mailbox.
mailbox/delete?username={username}	Delete the mailbox.
mailbox/enable?username={username}	Enable the mailbox.
mailboxes	Get all mailboxes.
domains	Get all accepted domains.
mailbox/getemailaddresses?username={username}	Get all email addresses.
mailbox/setemailaddresses?username={username}&emails={emails}	Set all email addresses.

The responses to all the operations can result in the following status codes.

Status code	Reason phrase	Description
200	OK	The operation successfully completed.
400	Bad request	Invalid parameter.
401	Unauthorized	The user is not authorized for web access.
404	Not Found	Cannot find the local user or online user.
500	Internal Server Error	An internal server error occurred.
409	Conflict	The operation failed because of the status conflict.
501	Not Implemented	The operation is not supported.
403	Forbidden	The operation is forbidden.
402	Payment Required	No available licenses.

### 3.5.5.1 mailbox?username={username}

Retrieves the mailbox assigned to the local user.

**username:** Local user name.

HTTP method	Description
GET	Retrieves the mailbox assigned to the local user.

#### 3.5.5.1.1 GET

Retrieves the mailbox assigned to the local user.

This operation is transported by an HTTP **GET**.

The operation can be invoked through the following URI:

```
https://<serverName>/services/builtin/MailboxManagement.svc/mailbox?username={username}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	Invalid parameter.
401	The user is not authorized for web access.
404	Cannot find the local user or online user.
500	Internal server error.

### 3.5.5.1.1.1 Request Body

None.

### 3.5.5.1.1.2 Response Body

The response body is encoded in either XML or JSON format. The format is controlled by the Accept header defined in section 2.2.4.36.

```
<xs:element name="Mailbox" nillable="true" type="tnsl:Mailbox"/>
```

### 3.5.5.1.1.3 Processing Details

This call returns the information of the mailbox. See section 2.2.4.36.

The server MUST construct a response in either XML or JSON format based on the value of the Accept header (section 2.2.2.6), or in XML format if the Accept header is not present.

## 3.5.5.2 mailbox/create?username={username}&email={email}

Create a mailbox and assign to the local user.

**username:** Local user name.

**email:** Email address to create.

HTTP method	Description
POST	Create a mailbox and assign to the local user.

### 3.5.5.2.1 POST

This operation is transported by an HTTP **POST**.

The operation can be invoked through the following URI:

```
https://<serverName>/services/builtin/MailboxManagement.svc/mailbox/create?username={username}&email={email}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	Invalid parameter.
401	The user is not authorized for web access.
404	Cannot find the local user or online user.
409	Operation failed because of the status conflict.
500	Internal server error.

### 3.5.5.2.1.1 Request Body

The request body is encoded in XML format or JSON format based on the value of the Accept header (section 2.2.2.6), or in XML format if the Accept header is not present.

### 3.5.5.2.1.2 Response Body

The response body is encoded in either XML or JSON format. The format is controlled by the Accept header defined in section 2.2.4.36.

### 3.5.5.2.1.3 Processing Details

This API creates a mailbox and assigns it to the local user.

### 3.5.5.3 mailbox/set?username={username}&email={email}

Assign the mailbox to the local user.

**username:** Local user name.

**email:** Email address to assign to the local user.

HTTP method	Description
POST	Assign the mailbox to the local user.

### 3.5.5.3.1 POST

This operation is transported by an HTTP **POST**.

The operation can be invoked through the following URI:

```
https://<serverName>/services/builtin/MailboxManagement.svc/mailbox/set?username={username}&e  
mail={email}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	Invalid parameter.
401	The user is not authorized for web access.
404	Cannot find the local user or online user.
409	Operation failed because of the status conflict.
500	Internal server error.
501	Operation is not supported.

#### 3.5.5.3.1.1 Request Body

None.

#### 3.5.5.3.1.2 Response Body

None.

#### 3.5.5.3.1.3 Processing Details

This API assigns the mailbox to the local user.

### 3.5.5.4 mailbox/unset?username={username}

Detach the mailbox from the local user.

**username:** Local user name.

HTTP method	Description
POST	Detach the mailbox from the local user.

#### 3.5.5.4.1 POST

This operation is transported by an HTTP **POST**.

The operation can be invoked through the following URI:

```
https://<serverName>/services/builtin/MailboxManagement.svc/mailbox/unset?username={username}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	Invalid parameter.
401	The user is not authorized for web access.
404	Cannot find the local user or online user.
409	Operation failed because of the status conflict.
500	Internal server error.
501	Operation is not supported.

#### 3.5.5.4.1.1 Request Body

None.

### 3.5.5.4.1.2 Response Body

None.

### 3.5.5.4.1.3 Processing Details

This API detaches the mailbox from the local user.

### 3.5.5.5 mailbox/disable?username={username}

Disables the mailbox that is assigned to the local user.

**username:** Local user name.

HTTP method	Description
POST	Disable the mailbox that is assigned to the local user.

### 3.5.5.5.1 POST

This operation is transported by an HTTP **POST**.

The operation can be invoked through the following URI:

```
https://<serverName>/services/builtin/MailboxManagement.svc/mailbox/disable?username={username}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	Invalid parameter.
401	The user is not authorized for web access.
403	Operation is forbidden.
404	Cannot find the local user or online user.

Status code	Description
500	Internal server error.
501	Operation is not supported.

### 3.5.5.5.1.1 Request Body

None.

### 3.5.5.5.1.2 Response Body

None.

### 3.5.5.5.1.3 Processing Details

This API disables the mailbox that is assigned to the local user.

## 3.5.5.6 mailbox/delete?username={username}

Delete the mailbox that is assigned to the local user.

**username:** Local user name.

HTTP method	Description
POST	Delete the mailbox that is assigned to the local user.

### 3.5.5.6.1 POST

This operation is transported by an HTTP **POST**.

The operation can be invoked through the following URI:

```
https://<serverName>/services/builtin/MailboxManagement.svc/mailbox/delete?username={username}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.



The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	Invalid parameter.
401	The user is not authorized for web access.
403	Operation is forbidden.
404	Cannot find the local user or online user.
500	Internal server error.

### 3.5.5.6.1.1 Request Body

The request body is encoded in XML format.

### 3.5.5.6.1.2 Response Body

None.

### 3.5.5.6.1.3 Processing Details

None.

## 3.5.5.7 mailbox/enable?username={username}

Enable the mailbox that is assigned to the local user.

**username:** Local user name.

HTTP method	Description
POST	Enable the mailbox that is assigned to the local user.

### 3.5.5.7.1 POST

This operation is transported by an HTTP **POST**.

The operation can be invoked through the following URI:

```
https://<serverName>/services/builtin/MailboxManagement.svc/mailbox/enable?username={username}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.

Request header	Usage	Value
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	Invalid parameter.
401	The user is not authorized for web access.
404	Cannot find the local user or online user.
500	Internal server error.
501	Operation is not supported.

### 3.5.5.7.1.1 Request Body

None.

### 3.5.5.7.1.2 Response Body

None.

### 3.5.5.7.1.3 Processing Details

This API enables the mailbox that is assigned to the local user.

## 3.5.5.8 mailboxes

Retrieve all the mailboxes.

HTTP method	Description
GET	Retrieve all the mailboxes

### 3.5.5.8.1 GET

This operation is transported by an HTTP **GET**.

The operation can be invoked through the following URI:

```
https://<serverName>/services/builtin/MailboxManagement.svc/mailboxes
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	Invalid parameter.
401	The user is not authorized for web access.
500	Internal server error.

### 3.5.5.8.1.1 Request Body

None.

### 3.5.5.8.1.2 Response Body

The response body is encoded in either XML or JSON format. The format is controlled by the Accept header defined in section 2.2.4.10.

```
<xs:element name="ArrayOfMailbox" nillable="true" type="tns:ArrayOfMailbox"/>
```

### 3.5.5.8.1.3 Processing Details

This call returns the information of all the mailboxes. See section 2.2.4.10.

The server MUST construct a response in either XML or JSON format based on the value of the Accept header (section 2.2.2.6), or in XML format if the Accept header is not present.

## 3.5.5.9 domains

Retrieve all the accepted domains.

HTTP method	Description
GET	Retrieve all the accepted domains.

### 3.5.5.9.1 GET

This operation is transported by an HTTP **GET**.

The operation can be invoked through the following URI:

```
https://<serverName>/services/builtin/MailboxManagement.svc/domains
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	Invalid parameter.
401	The user is not authorized for web access.
500	Internal server error.

#### 3.5.5.9.1.1 Request Body

None.

#### 3.5.5.9.1.2 Response Body

The response body is encoded in either XML or JSON format. The format is controlled by the Accept header defined in section 2.2.4.21.

```
<xs:element name="ArrayOfstring" nillable="true" type="tns4:ArrayOfstring"/>
```

#### 3.5.5.9.1.3 Processing Details

This call returns the information of the SharePoint site. See section 2.2.4.21.

The server MUST construct a response in either XML or JSON format based on the value of the Accept header (section 2.2.2.6), or in XML format if the Accept header is not present.

### 3.5.5.10 mailbox/getemailaddresses?username={username}

Retrieve all the email addresses that are assigned to the user.

**username:**

HTTP method	Description
GET	Retrieve all the email addresses that are assigned to the user.

#### 3.5.5.10.1 GET

This operation is transported by an HTTP **GET**.

The operation can be invoked through the following URI:

```
https://<serverName>/services/builtin/MailboxManagement.svc/mailbox/getemailaddresses?username={username}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	Invalid parameter.
401	The user is not authorized for web access.
404	Cannot find the local user or online user.
500	Internal server error.
501	Operation is not supported.

#### 3.5.5.10.1.1 Request Body

None.

### 3.5.5.10.1.2 Response Body

The response body is encoded in either XML or JSON format. The format is controlled by the Accept header defined in section 2.2.4.21.

### 3.5.5.10.1.3 Processing Details

This call returns the information of the email addresses. See section 2.2.4.21.

The server MUST construct a response in either XML or JSON format based on the value of the Accept header (section 2.2.2.6), or in XML format if the Accept header is not present.

## 3.5.5.11 mailbox/setemailaddresses?username={username}&emails={emails}

Set the email addresses for the user.

**username:** Local user name.

**emails:** Email addresses.

HTTP method	Description
POST	Set the email addresses for the user.

### 3.5.5.11.1 POST

This operation is transported by an HTTP **POST**.

The operation can be invoked through the following URI:

```
https://<serverName>/services/builtin/MailboxManagement.svc/mailbox/setemailaddresses?username={username}&emails={emails}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	Invalid parameter.

Status code	Description
401	The user is not authorized for web access.
402	No available licenses.
404	Cannot find the local user or online user.
500	Internal server error.
501	Operation is not supported.

### 3.5.5.11.1.1 Request Body

None.

### 3.5.5.11.1.2 Response Body

None.

### 3.5.5.11.1.3 Processing Details

This API sets the email addresses for the user.

## 3.5.6 Timer Events

None.

## 3.5.7 Other Local Events

None.

## 3.6 IAlertManagement Server Details

### 3.6.1 Abstract Data Model

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

#### 3.6.1.1 AlertInfo

The AlertInfo data model has a full complex type defined in section 2.2.4.1.

Severity: The health status:

**OK:** There is no existing alert.

**Information:** Informational alert.

**Warning:** Warning alert.

**Critical:** Critical alert.

**Report:** An alert that collects information from the client.

### 3.6.2 Timers

None.

### 3.6.3 Initialization

None.

### 3.6.4 Higher-Layer Triggered Events

None.

### 3.6.5 Message Processing Events and Sequencing Rules

Resource	Description
alerts/index/{startingindex}/count/{amount}	Retrieves all health alerts in the network.
alert/enable?alertkey={alertkey}	Enables an ignored alert.
alert/disable?alertkey={alertkey}	Disables an alert.
alert/clear?alertkey={alertkey}	Clears an alert.
alert/repair?alertkey={alertkey}	Attempts to repair an alert.

The responses to all the operations can result in the following status codes.

Status code	Reason phrase	Description
200	OK	Operation successfully completed.
400	Bad Request	One or more parameters are not valid.
401	Unauthorized	Access to the requested resource is denied.
404	Not Found	The specific resource does not exist.

#### 3.6.5.1 alerts/index/{startingindex}/count/{amount}

Retrieves all health alerts in the network.

**startingindex:** The numerical position of the first item to be retrieved. See section 2.2.3.45.

**amount:** The number of items to be retrieved. See section 2.2.3.2.

HTTP method	Description
GET	Retrieves all health alerts in the network.

#### 3.6.5.1.1 GET



Retrieves all health alerts in the network.

This operation is transported by an HTTP **GET**.

The operation can be invoked through the following URI:

```
alerts/index/{startingindex}/count/{amount}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied.

#### 3.6.5.1.1.1 Request Body

None.

#### 3.6.5.1.1.2 Response Body

The response body is encoded in either XML or JSON format. The format is controlled by the Accept header defined in section 2.2.2.6.

```
<xs:element name="PartialCollection_Of_AlertInfo" nillable="true" type="tns1:PartialCollection_Of_AlertInfo"/>
```

#### 3.6.5.1.1.3 Processing Details

This call returns the PartialCollection, which contains the instances of AlertInfo specified by startingIndex and amount.

The server MUST construct a response in either XML or JSON format based on the value of the Accept header (section 2.2.2.6), or in XML format if the Accept header is not present.

### 3.6.5.2 alert/enable?alertkey={alertkey}

This API enables an ignored alert that is identified by the alert key.

**alertkey:** The key of the alert. See section 2.2.3.1.

HTTP method	Description
POST	Enables an ignored alert.

#### 3.6.5.2.1 POST

This API enables an ignored alert. The alert is identified by the alert key.

This operation is transported by an HTTP **POST**.

The operation can be invoked through the following URI:

```
https://<serverName>/services/builtin/AlertManagement.svc/alert/enable?alertkey={alertkey}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	The operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied.
404	The alert specified by the alert key does not exist.

#### 3.6.5.2.1.1 Request Body

None.

#### 3.6.5.2.1.2 Response Body

None.

### 3.6.5.2.1.3 Processing Details

This call enables a network health alert.

### 3.6.5.3 alert/disable?alertkey={alertkey}

This API is for disabling an alert.

**alertkey:** The key of the alert. See section 2.2.3.1.

HTTP method	Description
POST	Disables an alert.

#### 3.6.5.3.1 POST

This operation disables an alert and is transported by an HTTP **POST**.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/AlertManagement.svc/alert/alert/disable?alertkey={alertkey}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, see section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, see section 2.2.2.4.
AppName	Optional	The client application name. For more details, see section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, see section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, see section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied.
404	The alert specified by the alert key does not exist.

#### 3.6.5.3.1.1 Request Body

None.

### 3.6.5.3.1.2 Response Body

None.

### 3.6.5.3.1.3 Processing Details

This call disables a network health alert.

### 3.6.5.4 alert/clear?alertkey={alertkey}

This API clears an alert given the key of the alert.

**alertkey:** The key of the alert. See section 2.2.3.1.

HTTP method	Description
POST	Clears an alert.

#### 3.6.5.4.1 POST

This operation clears an alert and is transported by an HTTP POST.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/AlertManagement.svc/alert/clear?alertkey={alertkey}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied.
404	The alert specified by the alert key does not exist.

### 3.6.5.4.1.1 Request Body

None.

### 3.6.5.4.1.2 Response Body

None.

### 3.6.5.4.1.3 Processing Details

This call clears a network health alert.

### 3.6.5.5 alert/repair?alertkey={alertkey}

This API attempts to repair an alert if the alert is repairable.

**alertkey:** The key of the alert. See section 2.2.3.1.

HTTP method	Description
POST	Attempts to repair an alert.

### 3.6.5.5.1 POST

This operation attempts to repair an alert. It is transported by an HTTP POST.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/AlertManagement.svc/alert/repair?alertkey={alertkey}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied.

Status code	Description
404	The alert specified by the alert key does not exist.

#### **3.6.5.5.1.1 Request Body**

None.

#### **3.6.5.5.1.2 Response Body**

None.

#### **3.6.5.5.1.3 Processing Details**

This call attempts to repair a network health alert.

### **3.6.6 Timer Events**

None.

### **3.6.7 Other Local Events**

None.

## **3.7 IDeviceManagement Server Details**

### **3.7.1 Abstract Data Model**

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

#### **3.7.1.1 DeviceInfo**

The DeviceInfo data model has a full complex type defined in section 2.2.4.28.

#### **3.7.2 Timers**

None.

#### **3.7.3 Initialization**

None.

#### **3.7.4 Higher-Layer Triggered Events**

None.

### 3.7.5 Message Processing Events and Sequencing Rules

Resource	Description
devices/index/{startingindex}/count/{amount}	Retrieves a list of devices that are managed by the server.
device/{deviceid}/startbackup	Starts to back up a device that is managed by the server.
device/{deviceid}/stopbackup	Stops the backup of a device that is managed by the server.

The responses to all the operations can result in the following status codes.

Status code	Reason phrase	Description
200	OK	The operation successfully completed.
400	Bad Request	One or more parameters are not valid.
401	Unauthorized	Access to the requested resource is denied.
404	Not Found	Resource not found.
403	Forbidden	The operation is forbidden.

#### 3.7.5.1 devices/index/{startingindex}/count/{amount}

Retrieves a list of devices that are managed by the server.

**startingindex:** The numerical position of the first item to be retrieved. It SHOULD be a string that can be converted to a positive integer. See section 2.2.3.45.

**amount:** The number of items to be retrieved. It SHOULD be a string that can be converted to a positive integer. See section 2.2.3.2.

HTTP method	Description
GET	Retrieves a list of devices that are managed by the server, or the devices that the current standard user has remote access permission to.

##### 3.7.5.1.1 GET

Retrieves a list of devices that are managed by the server.<1>

This operation is transported by an HTTP **GET**.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/DeviceManagement.svc/devices/index/{startingindex}/count/{amount}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied.
404	The requested resource does not exist.

### 3.7.5.1.1.1 Request Body

None.

### 3.7.5.1.1.2 Response Body

The response body is encoded in either XML or JSON format. The format is controlled by the Accept header defined in section 2.2.2.6.

```
<xs:element name="PartialCollection_Of_DeviceInfo" nillable="true"
type="tns1:PartialCollection_Of_DeviceInfo"/>
```

### 3.7.5.1.1.3 Processing Details

This call returns the PartialCollection, which contains the instances of DeviceInfo specified by startingIndex and amount.

The server MUST construct a response in either XML or JSON format based on the value of the Accept header (section 2.2.2.6), or in XML format if the Accept header is not present.

### 3.7.5.2 device/{deviceid}/startbackup

Starts to back up a device that is managed by the server.

**deviceid:** The ID of the device; see section 2.2.3.8.

HTTP method	Description
POST	Starts to back up a device that is managed by the server.



### 3.7.5.2.1 POST

Starts to back up a device that is managed by the server.

This operation is transported by an HTTP **POST**.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/DeviceManagement.svc/device/{deviceid}/startbackup
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied. This API only supports Administrator calls.
403	Another backup is already in progress.
404	The device specified by DeviceId does not exist.

#### 3.7.5.2.1.1 Request Body

None.

#### 3.7.5.2.1.2 Response Body

None.

#### 3.7.5.2.1.3 Processing Details

The server MUST enable the backup feature to support backup.

This call starts to backup the specific device.

### 3.7.5.3 device/{deviceid}/stopbackup

Stops the backup of a device that is managed by the server.

**deviceid:** The ID of the device; see section 2.2.3.8.

HTTP method	Description
POST	Stops the backup of a device that is managed by the server.

#### 3.7.5.3.1 POST

Stops the backup of a device that is managed by the server.

This operation is transported by an HTTP **POST**.

The operation can be invoked through the following URI:

```
Https://<ServerName>/services/builtin/DeviceManagement.svc/device/{deviceid}/stopbackup
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied. This API only supports Administrator calls.
403	The backup has already been stopped.
404	The device specified by DeviceId does not exist.

#### 3.7.5.3.1.1 Request Body

None.

#### 3.7.5.3.1.2 Response Body

None.

### 3.7.5.3.1.3 Processing Details

Server MUST enable the backup feature to support backup.

This call stops the processing backup for the specific device.

### 3.7.6 Timer Events

None.

### 3.7.7 Other Local Events

None.

## 3.8 IServiceManagement Server Details

### 3.8.1 Abstract Data Model

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

#### 3.8.1.1 WebApiProvider

The WebApiProvider data model has a full complex type defined in section 2.2.4.69.

### 3.8.2 Timers

None.

### 3.8.3 Initialization

None.

### 3.8.4 Higher-Layer Triggered Events

None.

### 3.8.5 Message Processing Events and Sequencing Rules

Resource	Description
permittedbuiltinservices	Retrieves the other services that the current logged-in user has access to.

The responses to all the operations can result in the following status codes.

Status code	Reason phrase	Description
200	OK	The operation successfully completed.

Status code	Reason phrase	Description
401	Unauthorized	Access to the requested resource is denied.

### 3.8.5.1 permittedbuiltinservices

HTTP method	Description
GET	Retrieves the other services that the current logged-in user has access to.

#### 3.8.5.1.1 GET

Retrieves the other services that the current logged-in user has access to.

This operation is transported by an HTTP **GET**.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/servicemanagement.svc/permittedbuiltinservices
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
401	Access to the requested resource is denied.

#### 3.8.5.1.1.1 Request Body

None.

#### 3.8.5.1.1.2 Response Body

The response body is encoded in either XML or JSON format. The format is controlled by the Accept header defined in section 2.2.2.6.

```
<xs:element name="ArrayOfWebApiProvider" nillable="true" type="tns:ArrayOfWebApiProvider"/>
```

### 3.8.5.1.1.3 Processing Details

This call returns a list of other WebApi provider services that the current logged-in user has permission to access.

The server MUST construct a response in either XML or JSON format based on the value of the Accept header (section 2.2.2.6), or in XML format if the Accept header is not present.

### 3.8.6 Timer Events

None.

### 3.8.7 Other Local Events

None.

## 3.9 IServerManagement Server Details

### 3.9.1 Abstract Data Model

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

#### 3.9.1.1 WebApiProvider

The WebApiProvider data model has a full complex type defined in section 2.2.4.64.

#### 3.9.2 Timers

None.

#### 3.9.3 Initialization

None.

#### 3.9.4 Higher-Layer Triggered Events

None.

#### 3.9.5 Message Processing Events and Sequencing Rules

Resource	Description
serverinformation	Retrieves the basic server information to the client, such as the server name and the server SKU.

The responses to all the operations can result in the following status codes.

Status code	Reason phrase	Description
200	OK	The operation successfully completed.
401	Unauthorized	Access to the requested resource is denied.

The request **MUST** contain the authentication information.

### 3.9.5.1 serverinformation

HTTP method	Description
GET	Retrieves the basic server information to the client, such as the server name and the server SKU.

#### 3.9.5.1.1 GET

Retrieves the basic server information to the client, such as the server name and the server SKU.

This operation is transported by an HTTP **GET**.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/servermanagement.svc/serverinformation
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client <b>MUST</b> either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
401	Access to the requested resource is denied.

#### 3.9.5.1.1.1 Request Body

None.

### **3.9.5.1.1.2 Response Body**

The response body is encoded in either XML or JSON format. The format is controlled by the Accept header defined in section 2.2.2.6.

```
<xs:element name="ServerInfo" nillable="true" type="tns1:ServerInfo"/>
```

### **3.9.5.1.1.3 Processing Details**

This call returns an instance of complex type of ServerInfo; see section 2.2.4.64.

The server MUST construct a response in either XML or JSON format based on the value of the Accept header (section 2.2.2.6), or in XML format if the Accept header is not present.

## **3.9.6 Timer Events**

None.

## **3.9.7 Other Local Events**

None.

## **3.10 ICustomizationManagement Server Details**

### **3.10.1 Abstract Data Model**

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

#### **3.10.1.1 CustomizationInfo**

The data model uses full complex types defined in section 2.2.4.27, section 2.2.4.9, and section 2.2.4.35.

#### **3.10.2 Timers**

None.

#### **3.10.3 Initialization**

None.

#### **3.10.4 Higher-Layer Triggered Events**

None.

### 3.10.5 Message Processing Events and Sequencing Rules

Retrieves the customization information.

Resource	Description
customizationinformation	Retrieves the customization information.

The responses to all the operations can result in the following status codes.

Status code	Reason phrase	Description
200	OK	Operation successfully completed.
404	Not Found	Cannot find the customization information from the server.

#### 3.10.5.1 customizationinformation

Retrieves the customization information.

HTTP method	Description
GET	Retrieves the customization information.

##### 3.10.5.1.1 GET

Retrieves the customization information.

This operation is transported by an HTTP **GET**.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/CustomizationManagement.svc/CustomizationInformation
```

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
404	Cannot found the customization information from server.

##### 3.10.5.1.1.1 Request Body

None.

##### 3.10.5.1.1.2 Response Body

The response body is encoded in either XML or JSON format. The format is controlled by the Accept header defined in section 2.2.2.6.



```
<xs:element name="CustomizationInfo" nillable="true" type="tnsl:CustomizationInfo"/>
```

### 3.10.5.1.1.3 Processing Details

1. This call returns an instance of the complex type CustomizationInfo; see section 2.2.4.27.
2. The server MUST construct a response in either XML or JSON format based on the value of the Accept header (section 2.2.2.6), or in XML format if the Accept header is not present.
3. The server initializes the customization information.

### 3.10.6 Timer Events

None.

### 3.10.7 Other Local Events

None.

## 3.11 IMediaManagement Server Details

### 3.11.1 Abstract Data Model

None.

### 3.11.2 Timers

None.

### 3.11.3 Initialization

At initialization time, the protocol server MUST begin listening for requests at the respective URL addresses given in the message transport, as specified in section 2.1.

Media feature MUST be initialized on the server.<2>

### 3.11.4 Higher-Layer Triggered Events

None.

### 3.11.5 Message Processing Events and Sequencing Rules

Resource	Description
metadata/item/mediatype/{mediatypedata}/groupingmethod/{groupingmethoddata}/sortproperty/{sortpropertydata}/sortorder/{sortorderdata}/index/{startingindexdata}/count/{requestedcountdata}?grouping={groupingdata}	The meta data of media items of a certain

Resource	Description
	<p>Media type with the specified sorting and grouping options.</p>
<p>metadata/container/mediatype/{mediatypedata}/groupingmethod/{groupingmethoddata}?path={containerpath}</p>	<p>Detailed information about a metadata container.</p>
<p>search/mediatype/{mediatypedata}/groupingmethod/{groupingmethoddata}/index/{startingindexdata}/count/{requestedcountdata}?keywords={keywords}&amp;grouping={groupingdata}&amp;titleonly={titleonlydata}&amp;shallow={shallowdata}</p>	<p>Search result for media items that contain certain keywords.</p>
<p>photo/{id}/isoriginal/{isoriginaldata}/width/{widthdata}/height/{heightdata}</p>	<p>A photo with a certain size.</p>
<p>thumbnail/mediatype/{mediatypedata}/id/{id}/width/{widthdata}/height/{heightdata}</p>	<p>The thumbnail of picture, video, and album art of</p>

Resource	Description
	the music with a certain size.
testdata?sizedata={sizedata}	Data of a certain size to be sent to the client to help calculate the available bandwidth.
metadata/streams/mediatype/{mediatypedata}/id/{id}/index/{startingindexdata}/count/{requestedcountdata}	Information about the available streams for an audio or video item.

The responses to all the operations can result in the following status codes.

Status code	Reason phrase	Description
200	OK	The operation successfully completed.
400	Bad Request	One or more parameters are not valid.
401	Unauthorized	Access to the requested resource is denied.
404	Not Found	The path does not exist.

### 3.11.5.1 metadata/item/mediatype/{mediatypedata}/groupingmethod/{groupingmethoddata}/sortproperty/{sortpropertydata}/sortorder/{sortorderdata}/index/{startingindexdata}/count/{requestedcountdata}?grouping={groupingdata}

This method retrieves the metadata of media items of a certain media type with the specified sorting and grouping options.

**mediatypedata:** The media type. See section 2.2.3.24.

**groupingmethoddata:** The grouping method. See section 2.2.3.15.

**groupingdata:** The container path. See section 2.2.3.14.

**startingindexdata:** The numerical position of the first media item to be retrieved. See section 2.2.3.46.

**requestedcountdata:** The requested number of media items to be retrieved. See section 2.2.3.36.

**sortpropertydata:** The metadata property on which to sort the media items. See section 2.2.3.44.

**sortorderdata:** The sort order. See section 2.2.3.43.

HTTP method	Description
GET	Retrieves the metadata of media items of a certain media type with the specified sorting and grouping options.

#### 3.11.5.1.1 GET

This operation is transported by an HTTP **GET**.

The operation can be invoked through the following URI:

```
metadata/item/mediatype/{mediatypedata}/groupingmethod/{groupingmethoddata}/sortproperty/{sortpropertydata}/sortorder/{sortorderdata}/index/{startingindexdata}/count/{requestedcountdata}?grouping={groupingdata}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The value returned in the logon Response Canary header. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, see section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, see section 2.2.2.4.
AppName	Optional	Application name. For more details, see section 2.2.2.1.
AppVersion	Optional	Application version. For more details, see section 2.2.2.3.
AppPublisherName	Optional	Application publisher name. For more details, see section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied.
404	The path does not exist.

### 3.11.5.1.1.1 Request Body

None.

### 3.11.5.1.1.2 Response Body

The response body is encoded in either XML or JSON format. The format is controlled by the Accept header.

```
<xs:element name="MetadataResult" nillable="true" type="tns1:MetadataResult"/>
```

### 3.11.5.1.1.3 Processing Details

1. The server MUST only respond to requests that have established TLS 1.1 server authentication [RFC4346].
2. The server MUST construct a response in either XML or JSON format based on the value of the Accept header or XML if the Accept header was not present.
3. The server MUST retrieve the metadata of media items of a certain media type with the specified sorting and grouping options, and fill the response body with type MetadataResult defined in section 2.2.4.42.

## 3.11.5.2 metadata/container/mediatype/{mediatypedata}/groupingmethod/{groupingmethoddata}?path={containerpath}

Retrieves detailed information about a metadata container.

**mediatypedata:** The media type. See section 2.2.3.24.

**groupingmethoddata:** The grouping method. See section 2.2.3.15.

**containerpath:** The path to the container. See section 2.2.3.4.

HTTP method	Description
GET	Retrieves detailed information about a metadata container.

### 3.11.5.2.1 GET

Retrieves detailed information about a metadata container.

This operation is transported by an HTTP **GET**.

The operation can be invoked through the following URI:

```
metadata/container/mediatype/{mediatypedata}/groupingmethod/{groupingmethoddata}?path={containerpath}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The value returned in the logon Response Canary header. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, see section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, see section 2.2.2.4.
AppName	Optional	Application name. For more details, see section 2.2.2.1.
AppVersion	Optional	Application version. For more details, see section 2.2.2.3.
AppPublisherName	Optional	Application publisher name. For more details, see section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied.
404	The path does not exist.

### 3.11.5.2.1.1 Request Body

None.

### 3.11.5.2.1.2 Response Body

The response body is encoded in either Xml or JSON format. The format is controlled by the Accept header.

```
<xs:element name="MetadataDetailResult" nillable="true" type="tns1:MetadataDetailResult"/>
```

### 3.11.5.2.1.3 Processing Details

1. The server MUST only respond to requests that have established TLS 1.1 server authentication [RFC4346].
2. The server MUST construct a response in either XML or JSON format based on the value of the Accept header or XML if the Accept header was not present.

- The server MUST retrieve the metadata of media items of a certain media type with the specified sorting and grouping options, and fill the response body with type MetadataDetailResult defined in section 2.2.4.39.

### 3.11.5.3 search/mediatype/{mediatypedata}/groupingmethod/{groupingmethoddata}/index/{startingindexdata}/count/{requestedcountdata}?keywords={keywords}&grouping={groupingdata}&titleonly={titleonlydata}&shallow={shallowdata}

Get the sub-item count for certain container. This API is deprecated.

**mediatypedata:** The media type. See section 2.2.3.24.

**groupingmethoddata:** The grouping method. See section 2.2.3.15.

**groupingdata:** The path to the container. See section 2.2.3.14.

**keywords:** A space-separated list of keywords to search for. See section 2.2.3.20.

**startingindexdata:** The numerical position of the first media item to be retrieved. See section 2.2.3.46.

**requestedcountdata:** The requested number of media items to be retrieved. See section 2.2.3.36.

**titleonlydata:** Whether the search is performed on title only. See section 2.2.3.48.

**shallowdata:** Whether the search is shallow or deep. See section 2.2.3.39.

HTTP method	Description
GET	Searches for media items that contain certain keywords.

#### 3.11.5.3.1 GET

Searches for media items that contain certain keywords.

This operation is transported by an HTTP **GET**.

The operation can be invoked through the following URI:

```
search/mediatype/{mediatypedata}/groupingmethod/{groupingmethoddata}/index/{startingindexdata}/count/{requestedcountdata}?keywords={keywords}&grouping={groupingdata}&titleonly={titleonlydata}&shallow={shallowdata}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The value returned in the logon Response Canary header. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, see section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, see section 2.2.2.4.
AppName	Optional	Application name. For more details, see section 2.2.2.1.

Request header	Usage	Value
AppVersion	Optional	Application version. For more details, see section 2.2.2.3.
AppPublisherName	Optional	Application publisher name. For more details, see section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied.
404	The path does not exist.

### 3.11.5.3.1.1 Request Body

None.

### 3.11.5.3.1.2 Response Body

The response body is encoded in Xml format.

### 3.11.5.3.1.3 Processing Details

The server MUST only respond to requests that have established TLS 1.1 server authentication as specified in [RFC4346].

The server MUST construct a response in either XML or JSON format based on the value of the Accept header or Xml if the Accept header was not present.

The server MUST search for media items that contain certain keywords, and fill the response body with type MetadataResult defined in section 2.2.4.42.

### 3.11.5.4 photo/{id}/isoriginal/{isoriginaldata}/width/{widthdata}/height/{heightdata}

This method retrieves a photo of a certain size.

**id:** The photo identifier. See section 2.2.3.17.

**isoriginaldata:** Specify whether to return the original data. If yes, **widthdata** and **heightdata** will not take effect, but they're still required in the request. See section 2.2.3.19.

**widthdata:** The width of an image. This value is required, but it is ignored if **isoriginaldata** is TRUE. See section 2.2.3.54.

**heightdata:** The width of the photo. This value is required, but it is ignored if **isoriginaldata** is TRUE. See section 2.2.3.16.

HTTP method	Description
GET	Retrieve a photo of a certain size.



### 3.11.5.4.1 GET

This operation is transported by an HTTP **GET**.

The operation can be invoked through the following URI:

```
photo/{id}/isoriginal/{isoriginaldata}/width/{widthdata}/height/{heightdata}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The value returned in the logon Response Canary header. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, see section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, see section 2.2.2.4.
AppName	Optional	Application name. For more details, see section 2.2.2.1.
AppVersion	Optional	Application version. For more details, see section 2.2.2.3.
AppPublisherName	Optional	Application publisher name. For more details, see section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied.
404	The path does not exist.

#### 3.11.5.4.1.1 Request Body

None.

#### 3.11.5.4.1.2 Response Body

The response body is a binary stream.

#### 3.11.5.4.1.3 Processing Details

1. The server MUST only respond to requests that have established TLS 1.1 server authentication [RFC4346].
2. The server MUST retrieve a photo of a certain size, and fill the response body with a binary stream.

### 3.11.5.5 thumbnail/mediatype/{mediatypedata}/id/{id}/width/{widthdata}/height/{heightdata}

Retrieves the thumbnail of a specified size of the picture, video, and album art of the music.

**mediatypedata:** The media type. See section 2.2.3.24.

**id:** The identifier of the media item. See section 2.2.3.17.

**widthdata:** The desired width of an image.

**heightdata:** The desired height of an image.

HTTP method	Description
GET	Retrieves the thumbnail of a specified size of the picture, video, and album art of the music.

#### 3.11.5.5.1 GET

Retrieves the thumbnail of a specified size of the picture, video, and album art of the music.

This operation is transported by an HTTP **GET**.

The operation can be invoked through the following URI:

```
thumbnail/mediatype/{mediatypedata}/id/{id}/width/{widthdata}/height/{heightdata}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The value returned in the logon Response Canary header. Client <b>MUST</b> either pass this header to the server or pass all of the following headers to the server. For more details, see section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, see section 2.2.2.4.
AppName	Optional	Application name. For more details, see section 2.2.2.1.
AppVersion	Optional	Application version. For more details, see section 2.2.2.3.
AppPublisherName	Optional	Application publisher name. For more details, see section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied.
404	The path does not exist.

### 3.11.5.5.1.1 Request Body

None.

### 3.11.5.5.1.2 Response Body

The response body is a binary stream.

### 3.11.5.5.1.3 Processing Details

1. The server MUST only respond to requests that have established TLS 1.1 server authentication [RFC4346].
2. The server MUST retrieve the thumbnail of a specified size of the picture, video, and album art of the music, and fill the response body with a binary stream.

### 3.11.5.6 testdata?sizedata={sizedata}

Requests data of a certain size to be sent to the client to help calculate the available bandwidth.

**sizedata:** The size of the test data to be retrieved. See section 2.2.3.41.

HTTP method	Description
POST	Requests data of a certain size to be sent to the client to help calculate the available bandwidth.

### 3.11.5.6.1 POST

Requests data of a certain size to be sent to the client to help calculate the available bandwidth.

This operation is transported by an HTTP **POST**.

The operation can be invoked through the following URI:

```
testdata?sizedata={sizedata}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The value returned in the logon Response Canary header. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, see section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, see section 2.2.2.4.
AppName	Optional	Application name. For more details, see section 2.2.2.1.
AppVersion	Optional	Application version. For more details, see section 2.2.2.3.
AppPublisherName	Optional	Application publisher name. For more details, see section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied.
404	The path does not exist.

### 3.11.5.6.1.1 Request Body

The request body contains sample data that can be used to calculate the available bandwidth. The request body can be empty.

```
<xs:element name="input" nillable="true" type="xs:base64Binary"/>
```

### 3.11.5.6.1.2 Response Body

The response body is a binary stream with a certain size specified in the request URI.

### 3.11.5.6.1.3 Processing Details

The server MUST only respond to requests that have established TLS 1.1 server authentication [RFC4346].

The server MUST generate a binary stream of a certain size and fill the response body with it.

### 3.11.5.7 metadata/streams/mediatype/{mediatypedata}/id/{id}/index/{startingindexdata}/count/{requestedcountdata}

Retrieves information about the available streams for an audio or video item.

**mediatypedata:** The media type. See section 2.2.3.24. Only "music" or "video" is valid here.

**id:** The identifier of the media item. See section 2.2.3.17.

**startingindexdata:** The numerical position of the first stream to be retrieved. See section 2.2.3.46.

**requestedcountdata:** The requested number of streams to be retrieved. See section 2.2.3.36.

HTTP method	Description
GET	Retrieves information about the available streams for an audio or video item.

#### 3.11.5.7.1 GET

Retrieves information about the available streams for an audio or video item.

This operation is transported by an HTTP **GET**.

The operation can be invoked through the following URI:

metadata/streams/mediatype/{mediatypedata}/id/{id}/index/{startingindexdata}/count/{requestedcountdata}

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The value returned in the logon Response Canary header. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, see section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, see section 2.2.2.4.
AppName	Optional	Application name. For more details, see section 2.2.2.1.
AppVersion	Optional	Application version. For more details, see section 2.2.2.3.
AppPublisherName	Optional	Application publisher name. For more details, see section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied.
404	The path does not exist.

### 3.11.5.7.1.1 Request Body

None.

### 3.11.5.7.1.2 Response Body

The response body is encoded in XML format.

```
<xs:element name="MetadataStreamResult" nillable="true" type="tns1:MetadataStreamResult"/>
```

### 3.11.5.7.1.3 Processing Details

The server MUST only respond to requests that have established TLS 1.1 server authentication [RFC4346].

The server MUST construct a response in either Xml or JSON format based on the value of the Accept header or XML if the Accept header was not present.

The server MUST retrieve information about the available streams for an audio or video item, and fill the response body with type MetadataStreamResult defined in section 2.2.4.43.

### **3.11.6 Timer Events**

None.

### **3.11.7 Other Local Events**

None.

## **3.12 IStorageManagement Server Details**

### **3.12.1 Abstract Data Model**

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

#### **3.12.1.1 FolderInfo**

FolderInfo represents the server folder managed by the server.

The FolderInfo data model has a full complex type defined in section 2.2.4.30.

#### **3.12.1.2 StorageDriveInfo**

StorageDriveInfo represents the drive info of storage server, which can be used to create the server folder managed by the primary server.

The StorageDriveInfo data model has a full complex type defined in section 2.2.4.66.

#### **3.12.1.3 StorageServerInfo**

StorageServerInfo represents the storage server, which can be used to create the server folder managed by the primary server.

The StorageServerInfo data model has a full complex type defined in section 2.2.4.67.

### **3.12.2 Timers**

None.

### **3.12.3 Initialization**

None.

### **3.12.4 Higher-Layer Triggered Events**

None.

### 3.12.5 Message Processing Events and Sequencing Rules

Resource	Description
servers	Retrieves storage servers' information.
storagedriveinfo?servername={servername}	Retrieves the drives information of storage server.
serverdrives/index/{index}/count/{count}	Retrieves a list of drives on the server.
serverfolders/index/{index}/count/{count}?username={username}	Retrieves a list of shared folders managed by the server.
serverfolder/create/overwritepermissions/{overwritepermissions}?sharename={sharename}&path={path}&description={description}&servername={servername}	Creates a shared folder on specific storage server.
serverfolder/{id}/delete/deletecontents/{deletecontents}	Removes or ends the sharing of a shared folder managed by the server.
serverfolder/{id}/rename?newname={newname}	Renames a shared folder managed by

Resource	Description
	the server.
serverfolder/{id}/modify/permission/{permission}?username={username}&name={name}&description={description}	Modifies the access permissions of a user on a shared folder by specifying a user name.
serverfolder/{id}/modify/usersid/{usersid}/permission/{permission}?name={name}&description={description}	Modifies the access permissions of a user on a shared folder by specifying a security identifier.

The responses to all the operations can result in the following status codes.

Status code	Reason phrase	Description
200	OK	The operation successfully completed.
401	Unauthorized	Access to the requested resource is denied.
400	Bad Request	One or more parameters are not valid.
404	Not Found	The specific resource does not exist.
403	Forbidden	The operation is forbidden.
409	Conflict	Server folder with the new path already exists.

### 3.12.5.1 servers

Retrieves the information of the storage servers.

HTTP method	Description
GET	Retrieves the storage servers' information.



### 3.12.5.1.1 GET

Retrieves the storage servers' information.

This operation is transported by an HTTP GET.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/StorageManagement.svc/Servers
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
401	Access to the requested resource is denied. This API only supports Administrator calls.

#### 3.12.5.1.1.1 Request Body

None.

#### 3.12.5.1.1.2 Response Body

The response body is encoded in either XML or JSON format. The format is controlled by the Accept header defined in section 2.2.2.6.

```
<xs:element name="ArrayOfStorageServerInfo" nillable="true" type="tns:ArrayOfStorageServerInfo"/>
```

#### 3.12.5.1.1.3 Processing Details

This call returns the storage server's information (section 2.2.4.20).

The server MUST construct a response in either XML or JSON format based on the value of the Accept header (section 2.2.2.6), or in XML format if the Accept header is not present.

### 3.12.5.2 storagedriveinfo?servername={servername}

Retrieves the information about the drives on a specific server.

**servername:** The storage server name.

HTTP method	Description
GET	Retrieves the information about the drives on a specific server.

#### 3.12.5.2.1 GET

Retrieves the information about the drives on a specific server.

This operation is transported by an HTTP **GET**.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/StorageManagement.svc/storagedriveinfo?servername={servername}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied. This API only supports Administrator calls.
404	The specific resource does not exist.

#### 3.12.5.2.1.1 Request Body

None.

#### 3.12.5.2.1.2 Response Body

The response body is encoded in either XML or JSON format. The format is controlled by the Accept header defined in section 2.2.2.6.

```
<xs:element name="ArrayOfStorageDriveInfo" nillable="true"
  type="tnsl:ArrayOfStorageDriveInfo"/>
```

### 3.12.5.2.1.3 Processing Details

This call returns the information of drives that belong to a specific storage server (section 2.2.4.19).

The server MUST construct a response in either XML or JSON format based on the value of the Accept header (section 2.2.2.6), or in XML format if the Accept header is not present.

### 3.12.5.3 serverdrives/index/{index}/count/{count}

Retrieves the information about the drives on the server.

**index:** The index in the collection to start retrieving data. See section 2.2.3.18.

**count:** The maximum count of items to return. See section 2.2.3.5.

HTTP method	Description
GET	Retrieves the information about the drives on the server.

#### 3.12.5.3.1 GET

Retrieves the information about the drives on the server.

This operation is transported by an HTTP **GET**.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/StorageManagement.svc/serverdrives/index/{index}/count/
{count}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied. This API only supports Administrator calls.
404	The specific resource does not exist.

### 3.12.5.3.1.1 Request Body

None.

### 3.12.5.3.1.2 Response Body

The response body is encoded in either XML or JSON format. The format is controlled by the Accept header defined in section 2.2.2.6.

```
<xs:element name="PartialCollection_Of_DriveInfo" nillable="true"
type="tns1:PartialCollection_Of_DriveInfo"/>
```

### 3.12.5.3.1.3 Processing Details

This call returns the information about the drives on a specific storage server (section 2.2.4.54).

The server MUST construct a response in either XML or JSON format based on the value of the Accept header (section 2.2.2.6), or in XML format if the Accept header is not present.

### 3.12.5.4 serverfolders/index/{index}/count/{count}?username={username}

Retrieves a list of shared folders managed by the server.

**index:** The index in the collection to start retrieving data. See section 2.2.3.18.

**count:** The maximum count of items to return. See section 2.2.3.5.

**username:** The user name is used to query access permission on the folder. It is an optional parameter. If it is not specified, the API returns the current access permissions for the user. See section 2.2.3.51.

HTTP method	Description
GET	Retrieves a list of shared folders managed by the server.

### 3.12.5.4.1 GET

This operation retrieves a list of shared folders managed by the server. It is transported by an HTTP GET.

The operation can be invoked through the following URI:

https://<ServerName>/services/builtin/StorageManagement.svc/serverfolders/index/{index}/count/{count}?username={username}

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied.
404	The specific resource does not exist.

#### 3.12.5.4.1.1 Request Body

None.

#### 3.12.5.4.1.2 Response Body

The response body is encoded in either XML or JSON format. The format is controlled by the Accept header defined in section 2.2.2.6.

```
<xs:element name="PartialCollection_Of_FolderInfo" nillable="true" type="tns1:PartialCollection_Of_FolderInfo"/>
```

#### 3.12.5.4.1.3 Processing Details

This call returns the information about drives on a specific storage server. See section 2.2.4.55.

The server MUST construct a response in either XML or JSON format based on the value of the Accept header (section 2.2.2.6), or in XML format if the Accept header is not present.

#### 3.12.5.5 serverfolder/create/overwritepermissions/{overwritepermissions}?sharename={sharename}&path={path}&description={description}&servername={servername}

This method creates a shared folder on the specific server.

**sharename:** The name of the shared folder to be created. See section 2.2.3.40.

**overwritepermissions:** This flag indicates whether to overwrite the folder descriptions. See section 2.2.3.31.

**path:** The local path of the folder to share on the specific server. See section 2.2.3.32.

**description:** The description of the shared folder. See section 2.2.3.7.

**servername:** The name of the specific server. This is an optional parameter; if not specified, the server folder will be created on the primary server. See section 2.2.3.38.

This is an optional parameter.

HTTP method	Description
POST	Creates a shared folder on the specific server.

### 3.12.5.5.1 POST

Creates a shared folder on the specific server.

This operation is transported by an HTTP POST.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/StorageManagement.svc/serverfolder/create/overwritepermissions/{overwritepermissions}?sharename={sharename}&path={path}&description={description}&servername={servername}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied. This API only supports Administrator calls.
403	The target share path is invalid.

Status code	Description
404	The specific resource does not exist.

### 3.12.5.5.1.1 Request Body

None.

### 3.12.5.5.1.2 Response Body

None.

### 3.12.5.5.1.3 Processing Details

This call creates a shared folder on the specific server.

### 3.12.5.6 serverfolder/{id}/delete/deletecontents/{deletecontents}

Removes or ends the sharing of a shared folder managed by the server.

**id:** The identifier for the shared folder. See section 2.2.3.17.

**deletecontents:** A flag that indicates whether the contents of the shared folder are deleted when the folder is no longer shared. See section 2.2.3.6.

HTTP method	Description
POST	Removes or ends the sharing of a shared folder managed by the server.

#### 3.12.5.6.1 POST

Removes or ends the sharing of a shared folder managed by the server.

This operation is transported by an HTTP POST.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/StorageManagement.svc/serverfolder/{id}/delete/deletecontents/{deletecontents}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.

Request header	Usage	Value
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied. This API only supports Administrator calls.
404	The specific resource does not exist.

### 3.12.5.6.1.1 Request Body

None.

### 3.12.5.6.1.2 Response Body

None.

### 3.12.5.6.1.3 Processing Details

This call stops the sharing of a shared folder managed by the server, and can delete the contents of the folder.

### 3.12.5.7 serverfolder/{id}/rename?newname={newname}

Rename the specified shared folder that is managed by the server.

**id:** The identifier for the shared folder. See section 2.2.3.17.

**newname:** The new name of the shared folder. See section 2.2.3.26.

HTTP method	Description
POST	Rename the specified shared folder that is managed by the server.

#### 3.12.5.7.1 POST

Rename the specified shared folder that is managed by the server.

This operation is transported by an HTTP POST.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/StorageManagement.svc/serverfolder/{id}/rename?newname={newname}
```

The request message for this operation contains the following HTTP headers.



Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied. This API only supports Administrator calls.
404	The specific resource does not exist.
409	The shared folder with new name has existed.

### 3.12.5.7.1.1 Request Body

None.

### 3.12.5.7.1.2 Response Body

None.

### 3.12.5.7.1.3 Processing Details

This call renames the shared folder.

### 3.12.5.8 `serverfolder/{id}/modify/permission/{permission}?username={username}&name={name}&description={description}`

Modifies the access permissions of a user on a shared folder by specifying a user name.

**id:** The identifier for the shared folder. See section 2.2.3.17.

**username:** The logon name of a user need set access permission. See section 2.2.3.51.

**permission:** The permission level of specific user. See section 2.2.3.33.

**name:** The name of the shared folder. See section 2.2.3.25.

**description:** The description of the shared folder. See section 2.2.3.7.

HTTP method	Description
POST	Modifies the access permissions of a user on a shared folder by specifying a user name.

### 3.12.5.8.1 POST

Modifies the access permissions of a user on a shared folder by specifying a user name.

This operation is transported by an HTTP **POST**.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/StorageManagement.svc/serverfolder/{id}/modify/permission/{permission}?username={username}&name={name}&description={description}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	Operation successfully completed.
401	Access to the requested resource is denied. This API only supports Administrator calls.
404	The specific resource does not exist.

#### 3.12.5.8.1.1 Request Body

None.

#### 3.12.5.8.1.2 Response Body

None.

#### 3.12.5.8.1.3 Processing Details

This call modifies the user access permission on the specific shared folder.

The user access permission MAY also be affected by access permission settings of the user group.

### 3.12.5.9 **serverfolder/{id}/modify/usersid/{usersid}/permission/{permission} ?name={name}&description={description}**

Modifies the access permissions of a user on a shared folder by specifying a security identifier.

**id:** The identifier for the shared folder. See section 2.2.3.17.

**usersid:** The security identifier (SID) of a user. See section 2.2.3.52.

**permission:** The permission level of specific user. See section 2.2.3.33.

**name:** The name of the shared folder. See section 2.2.3.25.

**description:** The description of the shared folder. See section 2.2.3.7.

HTTP method	Description
POST	Modifies the access permissions of a user on a shared folder by specifying a security identifier.

#### 3.12.5.9.1 **POST**

Modifies the access permissions of a user on a shared folder by specifying a security identifier.

This operation is transported by an HTTP **POST**.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/StorageManagement.svc/serverfolder/{id}/modify/usersid/{usersid}/permission/{permission}?name={name}&description={description}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.

Status code	Description
401	Access to the requested resource is denied. This API only supports Administrator calls.
404	The specific resource does not exist.

### 3.12.5.9.1.1 Request Body

None.

### 3.12.5.9.1.2 Response Body

None.

### 3.12.5.9.1.3 Processing Details

This call modifies the user access permission on the specific shared folder.

The user access permission MAY also be affected by access permission settings of the user group.

### 3.12.6 Timer Events

None.

### 3.12.7 Other Local Events

None.

## 3.13 IUserManagement Server Details

### 3.13.1 Abstract Data Model

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

#### 3.13.1.1 UserInfo

The UserInfo represents the information of a user.

The UserInfo data model has a full complex type defined in section 2.2.4.68.

### 3.13.2 Timers

None.

### 3.13.3 Initialization

None.

### 3.13.4 Higher-Layer Triggered Events

None.

### 3.13.5 Message Processing Events and Sequencing Rules

Resource	Description
users/index/{startingindex}/count/{amount}	Retrieves a list of users that have been created on the server.
usergroups	Retrieves a list of groups that have been created on the server.
users/connection/index/{startingindex}/count/{amount}?username={username}	Retrieves the connection information for a specific user.
user/{userid}/enable	Activates a user based on a specified security identifier.
user/{userid}/disable	Deactivates a user in the domain.
user/setpassword/{userid}	Changes the password of a

Resource	Description
	domain user.
user/{id}/update?firstname={firstname}&lastname={lastname}&remotewebaccess={remotewebaccess}&vpnaccess={vpnaccess}	Updates user information and permissions.
usergroup/{groupguid}/addusers	Add users to a specific group.
usergroup/{groupguid}/removeusers	Removes users from a specific group.
user/add?username={username}&firstname={firstname}&lastname={lastname}&remotewebaccess={remotewebaccess}&vpnaccess={vpnaccess}&usergroups={usergroups}	Creates a new standard user on the domain.
user/{id}/delete	Removes a standard user.

The responses to all the operations can result in the following status codes.

Status code	Reason phrase	Description
200	OK	Operation successfully completed.
400	Bad Request	One or more parameters are not valid.
401	Unauthorized	Access to the requested resource is denied.
404	Not Found	The specific resource does not exist.
403	Forbidden	Operation is forbidden.

### 3.13.5.1 users/index/{startingindex}/count/{amount}

Retrieves a list of users that have been created on the server's Active Directory.

**startingindex:** The numerical position of the first user to be retrieved. See section 2.2.3.45.

**amount:** The number of users to be retrieved. See section 2.2.3.2.

HTTP method	Description
GET	Retrieves a list of users that have been created on the Active Directory.

### 3.13.5.1.1 GET

Retrieves a list of users that have been created on the AD DS server.

This operation is transported by an HTTP GET.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/UserManagement.svc/users/index/{startingindex}/count/{amount}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied. This API only supports Administrator calls.

For more detail on the UserInfo type, please see section 2.2.4.68.

#### 3.13.5.1.1.1 Request Body

None.

#### 3.13.5.1.1.2 Response Body

The response body is encoded in either XML or JSON format. The format is controlled by the Accept header defined in section 2.2.2.6.

```
<xs:element name="PartialCollection_Of_UserInfo" nillable="true"
type="tns1:PartialCollection_Of_UserInfo"/>
```

### 3.13.5.1.1.3 Processing Details

This call returns the information of users that were created on the AD DS server. See section 2.2.4.68.

The server **MUST** construct a response in either XML or JSON format based on the value of the Accept header (section 2.2.2.6), or in XML format if the Accept header is not present.

### 3.13.5.2 usergroups

Retrieves a list of groups that have been created on the server's Active Directory.

HTTP method	Description
GET	Retrieves a list of groups that have been created on the server.

#### 3.13.5.2.1 GET

Retrieves a list of groups that have been created on the server.

This operation is transported by an HTTP **GET**.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/UserManagement.svc/usergroups
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client <b>MUST</b> either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
401	Access to the requested resource is denied. This API only supports Administrator calls.



### 3.13.5.2.1.1 Request Body

None.

### 3.13.5.2.1.2 Response Body

The response body is encoded in XML format.

```
<xs:element name="ReadOnlyCollectionOfGroupInfoEV6sb80H" nillable="true"
xmlns:q13="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
type="q13:ReadOnlyCollectionOfGroupInfoEV6sb80H"/>
```

### 3.13.5.2.1.3 Processing Details

This call returns the information about groups created on the AD DS server. See section 2.2.4.31.

The server MUST construct a response in either XML or JSON format based on the value of the Accept header (section 2.2.2.6), or in XML format if the Accept header is not present.

### 3.13.5.3 users/connection/index/{startingindex}/count/{amount}?username={username}

Retrieves the connection information related to a specific user.

**startingindex:** The numerical position of the first user to be retrieved. See section 2.2.3.45.

**amount:** The number of items to be retrieved. See section 2.2.3.2.

**username:** The user name of a specific user. It is an optional parameter; if the caller doesn't specify this parameter, the API returns all of the user's connection information.

HTTP method	Description
GET	Retrieves the connection information related to a specific user.

#### 3.13.5.3.1 GET

Retrieves the connection information related to a specific user.

This operation is transported by an HTTP GET.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/UserManagement.svc/users/connection/index/{startingindex}/count/{amount}?username={username}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.

Request header	Usage	Value
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied. This API only supports Administrator calls.
404	The specific resource does not exist.

### 3.13.5.3.1.1 Request Body

None.

### 3.13.5.3.1.2 Response Body

The response body is encoded in either XML or JSON format. The format is controlled by the Accept header defined in section 2.2.2.6.

```
<xs:element name="PartialCollection_Of_ConnectionInfo" nillable="true"
type="tns1:PartialCollection_Of_ConnectionInfo"/>
```

### 3.13.5.3.1.3 Processing Details

This call returns the connection information of a specific user. See section 2.2.4.26.

The server MUST construct a response in either XML or JSON format based on the value of the Accept header (section 2.2.2.6), or in XML format if the Accept header is not present.

## 3.13.5.4 user/{userid}/enable

Activates a user based on a specified security identifier.

**userid:** The ID of the user; it shouldis recommended to be the security identifier of the user. See section 2.2.3.50.

HTTP method	Description
POST	Activates a user based on a specified security identifier.

### 3.13.5.4.1 POST

Activates a user based on a specified security identifier.

This operation is transported by an HTTP POST.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/UserManagement.svc/user/{userid}/enable
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied. This API only supports Administrator calls.
404	The specific resource does not exist.

#### 3.13.5.4.1.1 Request Body

None.

#### 3.13.5.4.1.2 Response Body

None.

#### 3.13.5.4.1.3 Processing Details

This call activates the user specified by `userid`.

#### 3.13.5.5 `user/{userid}/disable`

Deactivates a user in the domain.

**userid:** The SID of the user. See section 2.2.3.50.

HTTP method	Description
POST	Deactivates a user in the domain.

### 3.13.5.5.1 POST

Deactivates a user in the domain.

This operation is transported by an HTTP POST.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/UserManagement.svc/user/{userid}/disable
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied. This API only supports Administrator calls.
404	The specific resource does not exist.

#### 3.13.5.5.1.1 Request Body

None.

#### 3.13.5.5.1.2 Response Body

None.

#### 3.13.5.5.1.3 Processing Details

This call disables the user specified by the user ID (userid).

### 3.13.5.6 user/setpassword/{userid}

Changes the password of a domain user.

**userid:** The ID of the user; it should be recommended to be the security identifier of the user. See section 2.2.3.50.

HTTP method	Description
POST	Changes the password of a domain user.

### 3.13.5.6.1 POST

Changes the password of a domain user.

This operation is transported by an HTTP **POST**.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/UserManagement.svc/user/setpassword/{userid}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied. This API only supports Administrator calls.
404	The specific resource does not exist.

#### 3.13.5.6.1.1 Request Body

The request body is encoded in XML format or JSON format based on the value of the Accept header (section 2.2.2.6), or in XML format if the Accept header is not present.

#### 3.13.5.6.1.2 Response Body

None.

### 3.13.5.6.1.3 Processing Details

This call changes the password of the user specified by the user ID (userid).

The new password MUST be specified in the body of the request, in the text of the string element.

### 3.13.5.7 user/{id}/update?firstname={firstname}&lastname={lastname}&remotewebaccess={remotewebaccess}&vpnaccess={vpnaccess}

Updates user information and permissions.

**id:** The SID of a user. See section 2.2.3.17.

**firstname:** The first name of the user. See section 2.2.3.12.

**lastname:** The last name of the user. See section 2.2.3.22.

**remotewebaccess:** A flag indicates whether the user has remote web access permission. See section 2.2.3.35.

**vpnaccess:** A flag indicates whether the user has vpn access permission. See section 2.2.3.53.

HTTP method	Description
POST	Updates user information and permissions.

### 3.13.5.7.1 POST

This operation updates user information and permissions. It is transported by an HTTP POST.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/UserManagement.svc/user/{id}/update?firstname={firstname}&lastname={lastname}&remotewebaccess={remotewebaccess}&vpnaccess={vpnaccess}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.

Status code	Description
400	One or more parameters are not valid.
401	Access to the requested resource is denied. This API only supports Administrator calls.
404	The specific resource does not exist.

### 3.13.5.7.1.1 Request Body

None.

### 3.13.5.7.1.2 Response Body

None.

### 3.13.5.7.1.3 Processing Details

This call updates the specified user information and permissions.

## 3.13.5.8 usergroup/{groupguid}/addusers

Adds users to the specified user group.

**groupguid:** The GUID of the group. See section 2.2.3.13.

HTTP method	Description
POST	Adds users to the specified user group.

### 3.13.5.8.1 POST

Adds users to the specified user group.

This operation is transported by an HTTP POST.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/UserManagement.svc/usergroup/{groupguid}/addusers
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.

Request header	Usage	Value
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied. This API only supports Administrator calls.
404	The specific resource does not exist.

### 3.13.5.8.1.1 Request Body

The request body is encoded in XML format or JSON format based on the value of the Accept header (section 2.2.2.6), or in XML format if the Accept header is not present.

### 3.13.5.8.1.2 Response Body

None.

### 3.13.5.8.1.3 Processing Details

This call will add the users specified by the GUIDs in the request body to the group specified by GroupGuid.

## 3.13.5.9 usergroup/{groupguid}/removeusers

Removes users from the specified user group.

**groupguid:** The GUID of the group. See section 2.2.3.13.

HTTP method	Description
POST	Removes users from the specified user group.

### 3.13.5.9.1 POST

Removes users from the specified user group.

This operation is transported by an HTTP POST.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/UserManagement.svc/usergroup/{groupguid}/removeusers
```

The request message for this operation contains the following HTTP headers.



Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied. This API only supports Administrator calls.
404	The specific resource does not exist.

### 3.13.5.9.1.1 Request Body

The request body is encoded in XML format or JSON format based on the value of the Accept header (section 2.2.2.6), or in XML format if the Accept header is not present.

### 3.13.5.9.1.2 Response Body

None.

### 3.13.5.9.1.3 Processing Details

This call removes the users specified by the GUIDs in the request body from the group specified by GroupGuid.

### 3.13.5.10 user/add?username={username}&firstname={firstname}&lastname={lastname}&remotewebaccess={remotewebaccess}&vpnaccess={vpnaccess}&usergroups={usergroups}

Creates a new standard user on the domain.

**username:** The name of the user. It SHOULD be the logon name of the user. See section 2.2.3.51.

**firstname:** The first name of the user. See section 2.2.3.12.

**lastname:** The last name of the user. See section 2.2.3.22.

**remotewebaccess:** A flag indicating whether the user has remote web access permission. See section 2.2.3.35.

**vpnaccess:** A flag indicating whether the user has vpn access permission. See section 2.2.3.53.

**usergroups:** A comma-separated list of group GUIDs. It is an optional parameter<3>. See section 2.2.3.49.

HTTP method	Description
POST	Creates a new standard user on the domain.

### 3.13.5.10.1 POST

Creates a new standard user on the domain.

This operation is transported by an HTTP POST.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/UserManagement.svc/user/add?username={username}&firstna
me={firstname}&lastname={lastname}&remotewebaccess={remotewebaccess}&vpnaccess={vpnaccess}&us
ergroups={usergroups}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied. This API only supports Administrator calls.
403	A user with the requested name already exists.
404	The specific resource does not exist.

#### 3.13.5.10.1.1 Request Body

The request body is encoded in XML format or JSON format based on the value of the Accept header (section 2.2.2.6), or in XML format if the Accept header is not present.

#### 3.13.5.10.1.2 Response Body

None.

### 3.13.5.10.1.3 Processing Details

This call creates a standard user with specified user information and permission settings.

A user with domain administrator permission cannot be created with this method.

#### 3.13.5.11 user/{id}/delete

Removes a standard user.

**id:** The security identifier (SID) of a user.

HTTP method	Description
POST	Removes a standard user.

#### 3.13.5.11.1 POST

This operation removes a standard user and is transported by an HTTP POST.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/UserManagement.svc/user/{id}/delete
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied. This API only supports Administrator calls.
404	The specific resource does not exist.

### 3.13.5.11.1.1 Request Body

None.

### 3.13.5.11.1.2 Response Body

None.

### 3.13.5.11.1.3 Processing Details

This call removes the user specified by id.

### 3.13.6 Timer Events

None.

### 3.13.7 Other Local Events

None.

## 3.14 ISharePointSiteMgmt Server Details

### 3.14.1 Abstract Data Model

None.

### 3.14.2 Timers

None.

### 3.14.3 Initialization

None.

### 3.14.4 Higher-Layer Triggered Events

None.

### 3.14.5 Message Processing Events and Sequencing Rules

Resource	Description
site	Retrieves the team site and my site URI.

#### 3.14.5.1 site

HTTP method	Description
GET	Retrieves the team site and my site URI.

### 3.14.5.1.1 GET

Retrieves the team site and my site URI.

This operation is transported by an HTTP **GET**.

The operation can be invoked through the following URI:

```
https://<serverName>/services/builtin/SharePointSiteMgmt.svc/site
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

#### 3.14.5.1.1.1 Request Body

The request body is encoded in XML format.

#### 3.14.5.1.1.2 Response Body

The response body is encoded in XML format.

```
<xs:element name="SharePointSiteAddressCollection" nillable="true"
xmlns:q14="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.0365Integ
ration" type="q14:SharePointSiteAddressCollection"/>
```

#### 3.14.5.1.1.3 Processing Details

This call returns information about the SharePoint site. See section 2.2.4.65.

The server MUST construct a response in either XML or JSON format based on the value of the Accept header (section 2.2.2.6), or in XML format if the Accept header is not present.

### 3.14.6 Timer Events

None.

### 3.14.7 Other Local Events

None.

## 3.15 IWindowsPhoneManagement Server Details

### 3.15.1 Abstract Data Model

None.

### 3.15.2 Timers

None.

### 3.15.3 Initialization

None.

### 3.15.4 Higher-Layer Triggered Events

None.

### 3.15.5 Message Processing Events and Sequencing Rules

Resource	Description
/notification/subscribe?deviceid={deviceid}	Registers or updates registration for a device to receive Windows phone tile updates notifications.
/notification/unsubscribe?deviceid={deviceid}	Unsubscribes a device from Windows phone tile updates.

The responses to all the operations can result in the following status codes.

Status code	Reason phrase	Description
200	OK	Operation successfully completed.
400	Bad Request	One or more parameters are not valid.
401	Unauthorized	Access to the requested resource is denied.
404	Not Found	The specific resource does not exist.

#### 3.15.5.1 /notification/subscribe?deviceid={deviceid}

Registers or updates the registration for a device to receive Windows Phone tile updates notifications.

**deviceid:** The ID of the device.

HTTP method	Description
POST	Registers or updates the registration for a device to receive Windows Phone tile updates notifications.

##### 3.15.5.1.1 POST

Registers or updates the registration for a device to receive Windows Phone tile updates notifications.

This operation is transported by an HTTP **POST**.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/WindowsPhoneManagement.svc/notification/subscribe?deviceid={deviceid}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied.

### 3.15.5.1.1.1 Request Body

The request body is encoded in XML format or JSON format based on the value of the Accept header (section 2.2.2.6), or in XML format if the Accept header is not present.

The request content ~~should be~~ the push notification channel URI obtained from the Microsoft Push Notification Service.

### 3.15.5.1.1.2 Response Body

None.

### 3.15.5.1.1.3 Processing Details

This call removes the user specified by the ID.

## 3.15.5.2 /notification/unsubscribe?deviceid={deviceid}

Unsubscribes a device from the Windows Phone tile updates.

**deviceid:**

HTTP method	Description
POST	Unsubscribes a device from Windows Phone tile updates.

### 3.15.5.2.1 POST

Unsubscribes a device from Windows Phone tile updates.

This operation is transported by an HTTP POST.

The operation can be invoked through the following URI:

```
https://<ServerName>/services/builtin/WindowsPhoneManagement.svc/notification/unsubscribe?deviceid={deviceid}
```

The request message for this operation contains the following HTTP headers.

Request header	Usage	Value
Canary	Optional	The user token. Client MUST either pass this header to the server or pass all of the following headers to the server. For more details, refer to section 2.2.2.5.
Authorization	Optional	The caller's credential is encoded text that is based on user name and password. For more details, refer to section 2.2.2.4.
AppName	Optional	The client application name. For more details, refer to section 2.2.2.1.
AppVersion	Optional	The client application version. For more details, refer to section 2.2.2.3.
AppPublisherName	Optional	The client application publisher name. For more details, refer to section 2.2.2.2.

The response message for this operation can result in the following status codes.

Status code	Description
200	Operation successfully completed.
400	One or more parameters are not valid.
401	Access to the requested resource is denied.
404	The device specified by the device ID is not registered.

#### 3.15.5.2.1.1 Request Body

None.

#### 3.15.5.2.1.2 Response Body

None.

#### 3.15.5.2.1.3 Processing Details

This call unsubscribes a device from Windows Phone tile updates.



### **3.15.6 Timer Events**

None.

### **3.15.7 Other Local Events**

None.

## 4 Protocol Examples

### 4.1 Login

#### Client Request:

```
GET https://www.contoso.com/services/builtin/session.svc/login HTTP/1.1
User-Agent: Sample App Name/1.0.0.0
Accept: application/xml
AppName: Sample App Name
AppVersion: 1.0.0.0
AppPublisherName: Publisher
Authorization: Basic VXNlcjpwYXNzd29yZCE=
Host: domainName
```

#### Server Response (XML):

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Server: Microsoft-IIS/8.5
Set-Cookie: ASP.NET_SessionId=hincf5pfrry4etux3rfe4n5k; path=/; HttpOnly
Canary: 7a10f945-91af-0597-14fd-a03bbdec2420
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
X-Content-Type-Options: nosniff
Date: Fri, 14 Jun 2013 10:56:44 GMT
Content-Length: 0
```

### 4.2 Get Server Information

#### Client Request:

```
GET https://www.contoso.com/services/builtin/servermanagement.svc/serverinformation HTTP/1.1
User-Agent: Sample App Name/1.0.0.0
Accept: application/xml
AppName: Sample App Name
AppVersion: 1.0.0.0
AppPublisherName: Publisher
Authorization: Basic VXNlcjpwYXNzd29yZCE=
Host: domainName
```

#### Server Response (XML):

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Content-Length: 633
Content-Type: application/xml; charset=utf-8
Server: Microsoft-IIS/8.5
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
X-Content-Type-Options: nosniff
Date: Fri, 14 Jun 2013 10:56:44 GMT
<ServerInfo xmlns="http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebAPI"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <IsMediaStreamingServiceAvailable>false</IsMediaStreamingServiceAvailable>
  <IsTsGatewayEnabled>false</IsTsGatewayEnabled>
  <ServerDomainGuid>82d53067-a76d-4d22-aa64-c8ff19fe22d3</ServerDomainGuid>
  <ServerGuid>b82fde5b-4289-4a8f-913a-06e887a3c528</ServerGuid>
  <ServerInternetDomainName>www.contoso.com</ServerInternetDomainName>
```

```
<ServerName>domainName</ServerName>
<ServerSkus>50</ServerSkus>
<ServerUtcNow>2013-12-02T03:09:57.3817387Z</ServerUtcNow>
<ServerVersion>6.3.9660.0</ServerVersion>
</ServerInfo>
```

### 4.3 Get Server Folders

#### Client Request:

```
GET
https://www.contoso.com/services/builtin/storagemanagement.svc/serverfolders/index/0/count/21
47483647 HTTP/1.1
User-Agent: Sample App Name/1.0.0.0
Accept: application/xml
Canary: 7a10f945-91af-0597-14fd-a03bbdec2420
Host: domainName
Cookie: ASP.NET_SessionId=hincf5pfrry4etux3rfe4n5k
Host: domainName
```

#### Server Response (XML):

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Content-Length: 1791
Content-Type: application/xml; charset=utf-8
Server: Microsoft-IIS/8.5
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
X-Content-Type-Options: nosniff
Date: Fri, 14 Jun 2013 11:08:10 GMT
<PartialCollection_Of_FolderInfo
xmlns="http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebAPI"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <CollectionChunk>
    <FolderInfo>
      <AccessPermission>Full</AccessPermission>
      <CurrentStatus>Healthy</CurrentStatus>
      <Description>Users</Description>
      <DriveFreeSpace>96528044032</DriveFreeSpace>
      <DriveID>75f98280-cbbf-409a-a2f1-dff1de8af145</DriveID>
      <FolderType>UserType</FolderType>
      <ID>6b2c4cd0-9a34-493f-939c-f7641491b4ee</ID>
      <Name>Users</Name>
      <Path>C:\ServerFolders\Users2</Path>
      <SharePath>\\domainName\Shared Folders\Users</SharePath>
      <Shared>true</Shared>
      <UncPath>\\domainName\Users</UncPath>
    </FolderInfo>
    <FolderInfo>
      <AccessPermission>Full</AccessPermission>
      <CurrentStatus>Healthy</CurrentStatus>
      <Description>Company</Description>
      <DriveFreeSpace>96528044032</DriveFreeSpace>
      <DriveID>75f98280-cbbf-409a-a2f1-dff1de8af145</DriveID>
      <FolderType>CompanyType</FolderType>
      <ID>3f2ff46c-031e-4ade-bdeb-e101e9c3d33a</ID>
      <Name>Company</Name>
      <Path>C:\ServerFolders\Company2</Path>
      <SharePath>\\domainName\Shared Folders\Company</SharePath>
      <Shared>true</Shared>
      <UncPath>\\domainName\Company</UncPath>
    </FolderInfo>
  </CollectionChunk>
</PartialCollection_Of_FolderInfo>
```

```

    <CurrentStatus>Healthy</CurrentStatus>
    <Description/>
    <DriveFreeSpace>96538091520</DriveFreeSpace>
    <DriveID>75f98280-cbbf-409a-a2f1-dff1de8af145</DriveID>
    <FolderType>NonPredefinedType</FolderType>
    <ID>084c36d8-dbaf-4ec0-945e-dcf6684bda96</ID>
    <Name>Software</Name>
    <Path>C:\ServerFolders\software</Path>
    <SharePath>\\domainName\Shared Folders\software</SharePath>
    <Shared>true</Shared>
    <UncPath>\\domainName\software</UncPath>
  </FolderInfo>
</CollectionChunk>
<CollectionModified>2013-12-01T19:24:17.12413-08:00</CollectionModified>
<CollectionSize>3</CollectionSize>
</PartialCollection_Of_FolderInfo>

```

## 4.4 Retrieve the Metadata for Items within a Folder

### Client Request:

```

GET
https://www.contoso.com/services/builtin/fileoperationservice.svc/items/index/0/count/510?path=%5C%5CdomainName%5CCompany&filter=All&sortByField=Name&ascending=True HTTP/1.1
User-Agent: Sample App Name/1.0.0.0
Accept: application/xml
Canary: 7a10f945-91af-0597-14fd-a03bbdec2420
Host: domainName
Cookie: ASP.NET_SessionId=hincf5pfrry4etux3rfe4n5k

```

### Server Response (XML):

```

HTTP/1.1 200 OK
Cache-Control: no-cache
Content-Length: 1173
Content-Type: application/xml; charset=utf-8
Server: Microsoft-IIS/8.5
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
X-Content-Type-Options: nosniff
Date: Mon, 02 Dec 2013 03:46:55 GMT
<ItemList xmlns="http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebAPI"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <Items>
    <ItemInfo>
      <AccessPermission>Full</AccessPermission>
      <CreateTime>2013-12-01T19:42:06.7225931-08:00</CreateTime>
      <CreateTimeUtc>2013-12-02T03:42:06.7225931Z</CreateTimeUtc>
      <IsDirectory>true</IsDirectory>
      <ModifiedTime>2013-12-01T19:42:06.7225931-08:00</ModifiedTime>
      <ModifiedTimeUtc>2013-12-02T03:42:06.7225931Z</ModifiedTimeUtc>
      <Name>DesignDocuments</Name>
      <Path>\\domainName\Company</Path>
      <Size>0</Size>
    </ItemInfo>
    <ItemInfo>
      <AccessPermission>Full</AccessPermission>
      <CreateTime>2013-12-01T19:42:12.8161694-08:00</CreateTime>
      <CreateTimeUtc>2013-12-02T03:42:12.8161694Z</CreateTimeUtc>
      <IsDirectory>true</IsDirectory>
      <ModifiedTime>2013-12-01T19:42:12.8161694-08:00</ModifiedTime>
      <ModifiedTimeUtc>2013-12-02T03:42:12.8161694Z</ModifiedTimeUtc>
      <Name>Specifications</Name>
      <Path>\\domainName\Company</Path>
      <Size>0</Size>

```

```

</ItemInfo>
<ItemInfo>
  <AccessPermission>Full</AccessPermission>
  <CreateTime>2013-12-01T19:43:57.2274112-08:00</CreateTime>
  <CreateTimeUtc>2013-12-02T03:43:57.2274112Z</CreateTimeUtc>
  <IsDirectory>false</IsDirectory>
  <ModifiedTime>2013-11-20T01:06:45.980719-08:00</ModifiedTime>
  <ModifiedTimeUtc>2013-11-20T09:06:45.980719Z</ModifiedTimeUtc>
  <Name>MaterialsList.xls</Name>
  <Path>\\DomainName\Company</Path>
  <Size>74240</Size>
</ItemInfo>
<ItemInfo>
  <AccessPermission>Full</AccessPermission>
  <CreateTime>2013-12-01T19:43:08.8831761-08:00</CreateTime>
  <CreateTimeUtc>2013-12-02T03:43:08.8831761Z</CreateTimeUtc>
  <IsDirectory>false</IsDirectory>
  <ModifiedTime>2013-11-20T00:43:35.967-08:00</ModifiedTime>
  <ModifiedTimeUtc>2013-11-20T08:43:35.967Z</ModifiedTimeUtc>
  <Name>ProductsIntroduction.pptx</Name>
  <Path>\\domainName\Company</Path>
  <Size>65803</Size>
</ItemInfo>
<ItemInfo>
  <AccessPermission>Full</AccessPermission>
  <CreateTime>2013-12-01T19:44:44.3645806-08:00</CreateTime>
  <CreateTimeUtc>2013-12-02T03:44:44.3645806Z</CreateTimeUtc>
  <IsDirectory>false</IsDirectory>
  <ModifiedTime>2013-12-01T19:45:16.7316226-08:00</ModifiedTime>
  <ModifiedTimeUtc>2013-12-02T03:45:16.7316226Z</ModifiedTimeUtc>
  <Name>ReadMe.txt</Name>
  <Path>\\domainName\Company</Path>
  <Size>5865</Size>
</ItemInfo>
<ItemInfo>
  <AccessPermission>Full</AccessPermission>
  <CreateTime>2013-12-01T19:45:47.0763661-08:00</CreateTime>
  <CreateTimeUtc>2013-12-02T03:45:47.0763661Z</CreateTimeUtc>
  <IsDirectory>false</IsDirectory>
  <ModifiedTime>2013-11-20T00:43:35.967-08:00</ModifiedTime>
  <ModifiedTimeUtc>2013-11-20T08:43:35.967Z</ModifiedTimeUtc>
  <Name>Structure.pptx</Name>
  <Path>\\domainName\Company</Path>
  <Size>65803</Size>
</ItemInfo>
</Items>
<TotalCount>6</TotalCount>
</ItemList>

```

## 4.5 Create a Folder

### Client Request:

```

POST
https://www.contoso.com/services/builtin/fileoperationservice.svc/folder?path=%5C%5CDomainName%5CCompany%5CDemo+Folder HTTP/1.1
User-Agent: Sample App Name/1.0.0.0
Accept: application/xml
Canary: 7a10f945-91af-0597-14fd-a03bbdec2420
Host: DomainName
Cookie: ASP.NET_SessionId=hincf5pfrry4etux3rfe4n5k
Content-Length: 0

```

### Server Response (XML):

```
HTTP/1.1 201 Created
Cache-Control: no-cache
Server: Microsoft-IIS/8.5
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
X-Content-Type-Options: nosniff
Date: Fri, 14 Jun 2013 11:35:21 GMT
Content-Length: 0
```

## 4.6 Upload a File

### Client Request:

```
POST
https://www.contoso.com/services/builtin/filecontentaccessservice.svc/filecontent?path=%5C%5C
DomainName%5CCompany%5CDemo+Folder%5Ctext.txt&overwrite=True HTTP/1.1
Accept: */*
AppName: Sample App Name
AppVersion: 1.0.0.0
AppPublisherName: Publisher
Authorization: Basic VXNlcjpwYXNzd29yZCE=
UA-CPU: AMD64
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2; Win64; x64; Trident/6.0)
Host: DomainName
Content-Length: 24
Connection: Keep-Alive
Cache-Control: no-cache
<File Content>
```

### Server Response (XML):

```
HTTP/1.1 201 Created
Cache-Control: no-cache
Server: Microsoft-IIS/8.5
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
X-Content-Type-Options: nosniff
Date: Fri, 14 Jun 2013 11:38:19 GMT
Content-Length: 0
```

## 4.7 Logout

### Client Request:

```
GET https://www.contoso.com/services/builtin/session.svc/logout HTTP/1.1
User-Agent: MSampleAppName/1.0.0.0
Accept: application/xml
Canary: 7a10f945-91af-0597-14fd-a03bbdec2420
Host: domainName
Cookie: ASP.NET_SessionId=hincf5pfrry4etux3rfe4n5k
```

### Server Response (XML):

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Server: Microsoft-IIS/8.5
X-AspNet-Version: 4.0.30319
Set-Cookie: ASP.NET_SessionId=; path=/  

```

X-Powered-By: ASP.NET  
X-Content-Type-Options: nosniff  
Date: Fri, 14 Jun 2013 11:47:07 GMT  
Content-Length: 0

## 5 Security

### 5.1 Security Considerations for Implementers

The Server and File Management Web APIs Protocol requires that all requests except GetCustomizationInfo (see section 3.10.5.1.1) from the client be authenticated. The client is expected to submit requests with user credential headers or use an implementation-dependent authentication mechanism to obtain a security token and include that token in the HTTP Canary header. The server will validate the user credential information or security token and use it to authorize the request.

### 5.2 Index of Security Parameters

None.



## 6 Appendix A: Full Xml Schema

Schema name	Prefi x	Sectio n
http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi	tns1	6.1
http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.Storage	tns2	6.2
http://schemas.microsoft.com/2003/10/Serialization/Arrays	tns4	6.3
http://schemas.microsoft.com/2003/10/Serialization/	tns5	6.6
http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObjectModel	tns6	6.4
http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel	tns7	6.5
http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.WebApi.Management.Storage	tns8	6.7
http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.O365Integration	tns9	6.8

### 6.1 http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi Schema

```

<xs:schema xmlns:tns1="http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi"
targetNamespace="http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="ItemThumbnail">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="Buffer" nillable="true"
type="xs:base64Binary"/>
      <xs:element minOccurs="0" maxOccurs="1" name="Format" nillable="true"
type="tns1:ThumbnailFormat"/>
      <xs:element minOccurs="0" maxOccurs="1" name="Height" type="xs:int"/>
      <xs:element minOccurs="0" maxOccurs="1" name="Width" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="ThumbnailFormat">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Jpeg"/>
      <xs:enumeration value="Png"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="ItemThumbnail" nillable="true" type="tns1:ItemThumbnail"/>
  <xs:complexType name="ItemList">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="Items" nillable="true"
type="tns1:ArrayOfItemInfo"/>
      <xs:element minOccurs="0" maxOccurs="1" name="TotalCount" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ArrayOfItemInfo">
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="unbounded" name="ItemInfo" nillable="true"
type="tns1:ItemInfo"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ItemInfo">
    <xs:sequence>

```

```

    <xs:element minOccurs="0" maxOccurs="1" name="AccessPermission" nillable="true"
xmlns:q1="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.Storage"
type="q1:Permission"/>
    <xs:element minOccurs="0" maxOccurs="1" name="CreateTime" type="DateTime"/>
    <xs:element minOccurs="0" maxOccurs="1" name="CreateTimeUtc" type="DateTime"/>
    <xs:element minOccurs="0" maxOccurs="1" name="IsDirectory" type="xs:boolean"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ModifiedTime" type="DateTime"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ModifiedTimeUtc" type="DateTime"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Name" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Path" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Size" type="xs:long"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ItemList" nillable="true" type="tns1:ItemList"/>
<xs:complexType name="SearchItemList">
  <xs:complexContent mixed="false">
    <xs:extension base="tns1:ItemList">
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="UnsearchedShares" nillable="true"
xmlns:q2="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
type="q2:ArrayOfstring"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="SearchItemList" nillable="true" type="tns1:SearchItemList"/>
<xs:element name="ItemInfo" nillable="true" type="tns1:ItemInfo"/>
<xs:element name="MSOUser" nillable="true"
xmlns:q3="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObj
ectModel" type="q3:MSOUser"/>
  <xs:element name="ArrayOfMSOUser" nillable="true"
xmlns:q4="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObj
ectModel" type="q4:ArrayOfMSOUser"/>
  <xs:element name="ArrayOfMSODomain" nillable="true"
xmlns:q5="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObj
ectModel" type="q5:ArrayOfMSODomain"/>
  <xs:element name="MSOTenantInformation" nillable="true"
xmlns:q6="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObj
ectModel" type="q6:MSOTenantInformation"/>
  <xs:complexType name="Mailbox">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="Email" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="UserName" nillable="true"
type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Mailbox" nillable="true" type="tns1:Mailbox"/>
  <xs:complexType name="ArrayOfMailbox">
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="unbounded" name="Mailbox" nillable="true"
type="tns1:Mailbox"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="ArrayOfMailbox" nillable="true" type="tns1:ArrayOfMailbox"/>
  <xs:complexType name="PartialCollection_Of_AlertInfo">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="CollectionChunk" nillable="true"
type="tns1:ArrayOfAlertInfo"/>
      <xs:element minOccurs="0" maxOccurs="1" name="CollectionModified" type="DateTime"/>
      <xs:element minOccurs="0" maxOccurs="1" name="CollectionSize" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ArrayOfAlertInfo">
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="unbounded" name="AlertInfo" nillable="true"
type="tns1:AlertInfo"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="AlertInfo">
    <xs:sequence>

```

```

        <xs:element minOccurs="0" maxOccurs="1" name="CanRepair" type="xs:boolean"/>
        <xs:element minOccurs="0" maxOccurs="1" name="DateAndTime" type="DateTime"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Description" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="IsSuppressed" type="xs:boolean"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Key" nillable="true" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="MachineName" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Severity" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Title" nillable="true" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="TroubleshootingSteps" nillable="true"
type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="PartialCollection_Of_AlertInfo" nillable="true"
type="tns1:PartialCollection_Of_AlertInfo"/>
<xs:complexType name="PartialCollection_Of_DeviceInfo">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="CollectionChunk" nillable="true"
type="tns1:ArrayOfDeviceInfo"/>
        <xs:element minOccurs="0" maxOccurs="1" name="CollectionModified" type="DateTime"/>
        <xs:element minOccurs="0" maxOccurs="1" name="CollectionSize" type="xs:int"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfDeviceInfo">
    <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="unbounded" name="DeviceInfo" nillable="true"
type="tns1:DeviceInfo"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="DeviceInfo">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="AntiSpywareStatus" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="AntiVirusStatus" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="BackupProgress" type="xs:int"/>
        <xs:element minOccurs="0" maxOccurs="1" name="BackupStatus" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="CanBackup" type="xs:boolean"/>
        <xs:element minOccurs="0" maxOccurs="1" name="DeviceDescription" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="DeviceName" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="DeviceType" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="FirewallStatus" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Key" nillable="true" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="OSFamily" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="OSName" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="OSServicePack" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="OnlineStatus" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="SecurityCenterStatus" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="SystemType" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="UpdateDetailStatus" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="UpdateStatus" nillable="true"
type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="PartialCollection_Of_DeviceInfo" nillable="true"
type="tns1:PartialCollection_Of_DeviceInfo"/>

```

```

<xs:complexType name="ArrayOfWebApiProvider">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded" name="WebApiProvider" nillable="true"
type="tns1:WebApiProvider"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="WebApiProvider">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="ContractName" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ContractNameSpace" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="RelativeLocation" type="xs:anyURI"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfWebApiProvider" nillable="true"
type="tns1:ArrayOfWebApiProvider"/>
<xs:complexType name="ServerInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="IsMediaStreamingServiceAvailable"
type="xs:boolean"/>
    <xs:element minOccurs="0" maxOccurs="1" name="IsTsGatewayEnabled" type="xs:boolean"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ServerDomainGuid" nillable="true"
xmlns:q7="http://schemas.microsoft.com/2003/10/Serialization/" type="q7:guid"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ServerGuid" nillable="true"
xmlns:q8="http://schemas.microsoft.com/2003/10/Serialization/" type="q8:guid"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ServerInternetDomainName" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ServerName" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ServerSku" type="xs:int"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ServerUtcNow" type="DateTime"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ServerVersion" nillable="true"
type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ServerInfo" nillable="true" type="tns1:ServerInfo"/>
<xs:complexType name="CustomizationInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="BackgroundImage" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Links" nillable="true"
type="tns1:ArrayOfLink"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Logo" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Title" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfLink">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded" name="Link" nillable="true"
type="tns1:Link"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Link">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Address" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Title" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="CustomizationInfo" nillable="true" type="tns1:CustomizationInfo"/>
<xs:complexType name="MetadataResult">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Result" nillable="true"
type="tns1:ArrayOfMetadataBase"/>
    <xs:element minOccurs="0" maxOccurs="1" name="TotalMatches" type="xs:unsignedInt"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfMetadataBase">
  <xs:sequence>

```

```

    <xs:element minOccurs="1" maxOccurs="unbounded" name="MetadataBase" nillable="true"
type="tns1:MetadataBase"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="MetadataBase">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Date" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Id" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Title" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="TypeName" nillable="true"
type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="MetadataContainer">
  <xs:complexContent mixed="false">
    <xs:extension base="tns1:MetadataBase">
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="AllCount" type="xs:int"/>
        <xs:element minOccurs="0" maxOccurs="1" name="AllDuration" type="xs:long"/>
        <xs:element minOccurs="0" maxOccurs="1" name="AllSize" type="xs:long"/>
        <xs:element minOccurs="0" maxOccurs="1" name="CalculationDone" type="xs:boolean"/>
        <xs:element minOccurs="0" maxOccurs="1" name="ContainerType" nillable="true"
type="tns1:ContainerType"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Count" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="FullPath" nillable="true"
type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:simpleType name="ContainerType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Album"/>
    <xs:enumeration value="Artist"/>
    <xs:enumeration value="Playlist"/>
    <xs:enumeration value="Rating"/>
    <xs:enumeration value="Genre"/>
    <xs:enumeration value="Folder"/>
    <xs:enumeration value="Date"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="MetadataItem">
  <xs:complexContent mixed="false">
    <xs:extension base="tns1:MetadataBase">
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="Album" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="AlbumArtist" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Duration" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="ParentFullPath" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Rating" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Resolution" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Size" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Tags" nillable="true"
type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="MetadataResult" nillable="true" type="tns1:MetadataResult"/>
<xs:complexType name="MetadataDetailResult">
  <xs:sequence>

```

```

        <xs:element minOccurs="0" maxOccurs="1" name="Result" nillable="true"
type="tns1:MetadataContainer"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="MetadataDetailResult" nillable="true" type="tns1:MetadataDetailResult"/>
<xs:complexType name="MetadataStreamResult">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="Result" nillable="true"
type="tns1:ArrayOfMetadataItemStream"/>
        <xs:element minOccurs="0" maxOccurs="1" name="TotalMatches" type="xs:unsignedInt"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfMetadataItemStream">
    <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="unbounded" name="MetadataItemStream"
nillable="true" type="tns1:MetadataItemStream"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="MetadataItemStream">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="Bitrate" type="xs:int"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Height" type="xs:int"/>
        <xs:element minOccurs="0" maxOccurs="1" name="MimeType" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Profile" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="RelativePath" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="RequiresTranscoding" type="xs:boolean"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Width" type="xs:int"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="MetadataStreamResult" nillable="true" type="tns1:MetadataStreamResult"/>
<xs:complexType name="ArrayOfStorageServerInfo">
    <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="unbounded" name="StorageServerInfo"
nillable="true" type="tns1:StorageServerInfo"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="StorageServerInfo">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="DeviceName" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="DfsNameSpace" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="DfsPhysicalPath" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="IsSecondServer" type="xs:boolean"/>
        <xs:element minOccurs="0" maxOccurs="1" name="OperatingSystemName" nillable="true"
type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfStorageServerInfo" nillable="true"
type="tns1:ArrayOfStorageServerInfo"/>
<xs:complexType name="ArrayOfStorageDriveInfo">
    <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="unbounded" name="StorageDriveInfo" nillable="true"
type="tns1:StorageDriveInfo"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="StorageDriveInfo">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="DeviceName" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="FileSystem" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="ShortName" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Size" type="xs:unsignedLong"/>
        <xs:element minOccurs="0" maxOccurs="1" name="SizeRemaining" type="xs:unsignedLong"/>
    </xs:sequence>
</xs:complexType>

```

```

        <xs:element minOccurs="0" maxOccurs="1" name="VolumeName" nillable="true"
type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfStorageDriveInfo" nillable="true"
type="tns1:ArrayOfStorageDriveInfo"/>
<xs:complexType name="PartialCollection_Of_DriveInfo">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="CollectionChunk" nillable="true"
type="tns1:ArrayOfDriveInfo"/>
        <xs:element minOccurs="0" maxOccurs="1" name="CollectionModified" type="DateTime"/>
        <xs:element minOccurs="0" maxOccurs="1" name="CollectionSize" type="xs:int"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfDriveInfo">
    <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="unbounded" name="DriveInfo" nillable="true"
type="tns1:DriveInfo"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="DriveInfo">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="Capacity" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="DataStatus" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="DiskID" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="DriveGuid" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="FileSystemType" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="ID" nillable="true" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="MountPoint" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Name" nillable="true" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="NextSnapshotTime" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="SnapshotsEnabled" type="xs:boolean"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Status" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="SystemDrive" type="xs:boolean"/>
        <xs:element minOccurs="0" maxOccurs="1" name="UsedSpace" nillable="true"
type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="PartialCollection_Of_DriveInfo" nillable="true"
type="tns1:PartialCollection_Of_DriveInfo"/>
<xs:complexType name="PartialCollection_Of_FolderInfo">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="CollectionChunk" nillable="true"
type="tns1:ArrayOfFolderInfo"/>
        <xs:element minOccurs="0" maxOccurs="1" name="CollectionModified" type="DateTime"/>
        <xs:element minOccurs="0" maxOccurs="1" name="CollectionSize" type="xs:int"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfFolderInfo">
    <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="unbounded" name="FolderInfo" nillable="true"
type="tns1:FolderInfo"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="FolderInfo">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="AccessPermission" nillable="true"
xmlns:q9="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.Storage"
type="q9:Permission"/>
        <xs:element minOccurs="0" maxOccurs="1" name="CurrentStatus" nillable="true"
type="xs:string"/>

```

```

    <xs:element minOccurs="0" maxOccurs="1" name="Description" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="DriveFreeSpace" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="DriveID" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="FolderType" nillable="true"
xmlns:q10="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.WebApi.Ma
nagement.Storage" type="q10:ServerFolderType"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ID" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Name" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Path" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="SharePath" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Shared" type="xs:boolean"/>
    <xs:element minOccurs="0" maxOccurs="1" name="UncPath" nillable="true"
type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="PartialCollection_Of_FolderInfo" nillable="true"
type="tns1:PartialCollection_Of_FolderInfo"/>
<xs:complexType name="PartialCollection_Of_UserInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="CollectionChunk" nillable="true"
type="tns1:ArrayOfUserInfo"/>
    <xs:element minOccurs="0" maxOccurs="1" name="CollectionModified" type="DateTime"/>
    <xs:element minOccurs="0" maxOccurs="1" name="CollectionSize" type="xs:int"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfUserInfo">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded" name="UserInfo" nillable="true"
type="tns1:UserInfo"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="UserInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="AccessLevel" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="CanChangePassword" type="xs:boolean"/>
    <xs:element minOccurs="0" maxOccurs="1" name="FirstName" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Key" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="LastName" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="RemoteWebAccess" type="xs:boolean"/>
    <xs:element minOccurs="0" maxOccurs="1" name="UserGroups" nillable="true"
type="tns1:ArrayOfGroupInfo"/>
    <xs:element minOccurs="0" maxOccurs="1" name="UserGuid" nillable="true"
xmlns:q11="http://schemas.microsoft.com/2003/10/Serialization/" type="q11:guid"/>
    <xs:element minOccurs="0" maxOccurs="1" name="UserName" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="UserStatus" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="VpnAccess" type="xs:boolean"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfGroupInfo">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded" name="GroupInfo" nillable="true"
type="tns1:GroupInfo"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="GroupInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="CurrentSid" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="GroupName" nillable="true"
type="xs:string"/>

```



```

    <xs:element minOccurs="0" maxOccurs="1" name="Guid" nillable="true"
xmlns:q12="http://schemas.microsoft.com/2003/10/Serialization/" type="q12:guid"/>
    <xs:element minOccurs="0" maxOccurs="1" name="RemoteWebAccess" type="xs:boolean"/>
    <xs:element minOccurs="0" maxOccurs="1" name="VpnAccess" type="xs:boolean"/>
    <xs:element minOccurs="0" maxOccurs="1" name="WindowsAccount" nillable="true"
type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="PartialCollection_Of_UserInfo" nillable="true"
type="tns1:PartialCollection_Of_UserInfo"/>
  <xs:element name="ReadOnlyCollectionOfGroupInfoEV6sb80H" nillable="true"
xmlns:q13="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
type="q13:ReadOnlyCollectionOfGroupInfoEV6sb80H"/>
  <xs:complexType name="PartialCollection_Of_ConnectionInfo">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="CollectionChunk" nillable="true"
type="tns1:ArrayOfConnectionInfo"/>
      <xs:element minOccurs="0" maxOccurs="1" name="CollectionModified" type="DateTime"/>
      <xs:element minOccurs="0" maxOccurs="1" name="CollectionSize" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
<xs:complexType name="ArrayOfConnectionInfo">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded" name="ConnectionInfo" nillable="true"
type="tns1:ConnectionInfo"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ConnectionInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="ClientInfo" nillable="true"
type="tns1:ConnectionClientInfo"/>
    <xs:element minOccurs="0" maxOccurs="1" name="EndTime" type="DateTime"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ServiceType" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="StartTime" type="DateTime"/>
    <xs:element minOccurs="0" maxOccurs="1" name="UserInfo" nillable="true"
type="tns1:RemoteConnectionUserInfo"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ConnectionClientInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="ClientApplicationName" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ClientApplicationPublisher"
nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ClientApplicationVersion" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ClientHostname" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ClientIPAddress" nillable="true"
type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="RemoteConnectionUserInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="AccountName" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="FirstName" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="LastName" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="SidValue" nillable="true"
type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="PartialCollection_Of_ConnectionInfo" nillable="true"
type="tns1:PartialCollection_Of_ConnectionInfo"/>
  <xs:element name="SharePointSiteAddressCollection" nillable="true"
xmlns:q14="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.0365Integ
ration" type="q14:SharePointSiteAddressCollection"/>

```

```
</xs:schema>
```

## 6.2 <http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.Storage> Schema

```
<xs:schema
xmlns:tns2="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.Storage"
targetNamespace="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.Storage"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="Permission">
    <xs:restriction base="xs:string">
      <xs:enumeration value="None"/>
      <xs:enumeration value="ReadOnly"/>
      <xs:enumeration value="Full"/>
      <xs:enumeration value="Other"/>
      <xs:enumeration value="ReadPermissions"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

## 6.3 <http://schemas.microsoft.com/2003/10/Serialization/Arrays> Schema

```
<xs:schema xmlns:tns4="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="ArrayOfstring">
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="unbounded" name="string" nillable="true"
type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="ArrayOfstring" nillable="true" type="tns4:ArrayOfstring"/>
</xs:schema>
```

## 6.4 <http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObjectModel> Schema

```
<xs:schema
xmlns:tns6="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObjectModel"
targetNamespace="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObjectModel"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="MSOUser">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="Activated" type="xs:boolean"/>
      <xs:element minOccurs="0" maxOccurs="1" name="FirstName" nillable="true"
type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="LastName" nillable="true"
type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="Licenses" nillable="true"
xmlns:q1="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
type="q1:ReadOnlyCollectionOfMSOLicensePGX_Pb6b"/>
      <xs:element minOccurs="0" maxOccurs="1" name="LocalUserName" nillable="true"
type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="ObjectId" nillable="true"
xmlns:q2="http://schemas.microsoft.com/2003/10/Serialization/" type="q2:guid"/>
      <xs:element minOccurs="0" maxOccurs="1" name="UserPrincipalName" nillable="true"
type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ArrayOfMSOLicense">
    <xs:sequence>
```

```

        <xs:element minOccurs="1" maxOccurs="unbounded" name="MSOLicense" nillable="true"
type="tns6:MSOLicense"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="MSOLicense">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="AccountName" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Services" nillable="true"
xmlns:q3="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
type="q3:ReadOnlyCollectionOfMSOLicenseServicePGX_Pb6b"/>
        <xs:element minOccurs="0" maxOccurs="1" name="SkuId" nillable="true" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfMSOLicenseService">
    <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="unbounded" name="MSOLicenseService"
nillable="true" type="tns6:MSOLicenseService"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="MSOLicenseService">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="Activated" type="xs:boolean"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Description" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="ServicePlan" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="ServiceType"
type="tns6:MSOLicenseServiceTypes"/>
    </xs:sequence>
</xs:complexType>
<xs:simpleType name="MSOLicenseServiceTypes">
    <xs:list>
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration value="None"/>
                <xs:enumeration value="Unknown"/>
                <xs:enumeration value="ExchangeOnline"/>
                <xs:enumeration value="SharePointOnline"/>
                <xs:enumeration value="LyncOnline"/>
                <xs:enumeration value="IntuneOnline"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:list>
</xs:simpleType>
<xs:complexType name="ArrayOfMSOUser">
    <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="unbounded" name="MSOUser" nillable="true"
type="tns6:MSOUser"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfMSODomain">
    <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="unbounded" name="MSODomain" nillable="true"
type="tns6:MSODomain"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="MSODomain">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="CapabilityType" nillable="true"
type="tns6:MSODomainTypes"/>
        <xs:element minOccurs="0" maxOccurs="1" name="IsPrimary" type="xs:boolean"/>
        <xs:element minOccurs="0" maxOccurs="1" name="IsVerified" type="xs:boolean"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Name" nillable="true" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:simpleType name="MSODomainTypes">
    <xs:restriction base="xs:string">
        <xs:enumeration value="None"/>
        <xs:enumeration value="Email"/>
    </xs:restriction>
</xs:simpleType>

```

```

        <xs:enumeration value="SharePoint"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="MSOTenantInformation">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="Activated" type="xs:boolean"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Address" nillable="true"
type="tns6:CompanyAddress"/>
        <xs:element minOccurs="0" maxOccurs="1" name="CompanyName" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="InitialDomain" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="LicenseSuite" nillable="true"
xmlns:q4="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
type="q4:ReadOnlyCollectionOfMSOLicenseSuitePGX_Pb6b"/>
        <xs:element minOccurs="0" maxOccurs="1" name="PrimaryDomain" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="TechnicalContact" nillable="true"
type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="CompanyAddress">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="City" nillable="true" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Country" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="PostalCode" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="State" nillable="true" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Street" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="TelephoneNumber" nillable="true"
type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfMSOLicenseSuite">
    <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="unbounded" name="MSOLicenseSuite" nillable="true"
type="tns6:MSOLicenseSuite"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="MSOLicenseSuite">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="ConsumedLicenses" type="xs:int"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Description" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="License" nillable="true"
type="tns6:MSOLicense"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Subscriptions" nillable="true"
xmlns:q5="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
type="q5:ReadOnlyCollectionOfMSOLicenseSubscriptionPGX_Pb6b"/>
        <xs:element minOccurs="0" maxOccurs="1" name="TotalLicenses" type="xs:int"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfMSOLicenseSubscription">
    <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="unbounded" name="MSOLicenseSubscription"
nillable="true" type="tns6:MSOLicenseSubscription"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="MSOLicenseSubscription">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="BillingExpirationTime" type="DateTime"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Expired" type="xs:boolean"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfMSOLicense" nillable="true" type="tns6:ArrayOfMSOLicense"/>
</xs:schema>

```

## 6.5 <http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel.Schema>

```
<xs:schema
xmlns:tns7="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
targetNamespace="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="ReadOnlyCollectionOfMSOLicensePGX_Pb6b">
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="list" nillable="true"
xmlns:q1="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObj
ectModel" type="q1:ArrayOfMSOLicense"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ReadOnlyCollectionOfMSOLicenseServicePGX_Pb6b">
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="list" nillable="true"
xmlns:q2="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObj
ectModel" type="q2:ArrayOfMSOLicenseService"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ReadOnlyCollectionOfMSOLicenseSuitePGX_Pb6b">
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="list" nillable="true"
xmlns:q3="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObj
ectModel" type="q3:ArrayOfMSOLicenseSuite"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ReadOnlyCollectionOfMSOLicenseSubscriptionPGX_Pb6b">
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="list" nillable="true"
xmlns:q4="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.AzureADObj
ectModel" type="q4:ArrayOfMSOLicenseSubscription"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ReadOnlyCollectionOfGroupInfoEV6sb80H">
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="list" nillable="true"
xmlns:q5="http://contracts.microsoft.com/WindowsServerEssentials/2011/09/WebApi"
type="q5:ArrayOfGroupInfo"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

## 6.6 <http://schemas.microsoft.com/2003/10/Serialization/Schema>

```
<xs:schema xmlns:tns5="http://schemas.microsoft.com/2003/10/Serialization/"
targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="string" nillable="true" type="xs:string"/>
  <xs:element name="anyURI" type="xs:anyURI"/>
  <xs:simpleType name="guid">
    <xs:restriction base="xs:string">
      <xs:pattern value="[\da-fA-F]{8}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{12}"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

## 6.7 <http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.WebApi.Management.Storage.Schema>

```
<xs:schema
xmlns:tns8="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.WebApi.M
```

```

anagement.Storage"
targetNamespace="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.Web
Api.Management.Storage" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="ServerFolderType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="NonPredefinedType"/>
      <xs:enumeration value="MusicType"/>
      <xs:enumeration value="PicturesType"/>
      <xs:enumeration value="DocumentsType"/>
      <xs:enumeration value="VideosType"/>
      <xs:enumeration value="BackupsType"/>
      <xs:enumeration value="FileBackupsType"/>
      <xs:enumeration value="FolderRedirectionType"/>
      <xs:enumeration value="CompanyType"/>
      <xs:enumeration value="UserType"/>
      <xs:enumeration value="OtherType"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

## 6.8 <http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.0365Integration> Schema

```

<xs:schema
xmlns:tns9="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.0365Inte
gration"
targetNamespace="http://schemas.datacontract.org/2004/07/Microsoft.WindowsServerSolutions.036
5Integration" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SharePointSiteAddressCollection">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="MySiteAddress" nillable="true"
type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="TeamSiteAddress" nillable="true"
type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>

```

## 7 Appendix B: 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.

- Windows Server 2012 R2 operating system
- Windows Server 2016 ~~Technical Preview~~ operating system

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 3.7.5.1.1: In Windows, this API only supports administrator calls and standard user calling. It returns a list of devices that are managed by the server. If the caller is a standard user, this API returns a list of devices for which the standard user has remote access permission.

<2> Section 3.11.3: When using the Media Web APIs on Windows Server 2012 R2, an HTTP 500 error occurs unless the Windows Server Essentials Media Pack is installed.

<3> Section 3.13.5.10: The **usergroups** parameter is supported only in Windows Server 2012 operating system.

## 8 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.



## 9 Index

### A

Applicability 20

### C

Capability negotiation 20  
Change tracking 216  
Common data types 21

### E

Examples  
  Create a Folder example 197  
  Get Server Folders example 195  
  Get Server Information example 194  
  Login example 194  
  Logout example 198  
  Retrieve the Metadata for Items within a Folder example 196  
  Upload a File example 198

### F

Fields - vendor-extensible 20

### G

Glossary 18

### I

Ialertmanagement server  
  Abstract data model 127  
  Higher-layer triggered events 128  
  Initialization 128  
  Message processing events and sequencing rules 128  
  Other local events 134  
  Timer events 134  
  Timers 128  
Iazureadmanagement server  
  Abstract data model 97  
  Higher-layer triggered events 97  
  Initialization 97  
  Message processing events and sequencing rules 98  
  Other local events 112  
  Timer events 112  
  Timers 97  
Icustomizationmanagement server  
  Abstract data model 143  
  Higher-layer triggered events 143  
  Initialization 143  
  Message processing events and sequencing rules 144  
  Other local events 145  
  Timer events 145  
  Timers 143  
Idevicemanagement server  
  Abstract data model 134  
  Higher-layer triggered events 134  
  Initialization 134  
  Message processing events and sequencing rules 135  
  Other local events 139  
  Timer events 139

- Timers 134
- Ifilecontentaccessservice server
  - Abstract data model 77
  - Higher-layer triggered events 77
  - Initialization 77
  - Message processing events and sequencing rules 78
  - Other local events 82
  - Timer events 82
  - Timers 77
- Ifileoperationservice server
  - Abstract data model 82
  - Higher-layer triggered events 82
  - Initialization 82
  - Message processing events and sequencing rules 82
  - Other local events 97
  - Timer events 97
  - Timers 82
- Imailboxmanagement server
  - Abstract data model 113
  - Higher-layer triggered events 113
  - Initialization 113
  - Message processing events and sequencing rules 113
  - Other local events 127
  - Timer events 127
  - Timers 113
- Imediamanagement server
  - Abstract data model 145
  - Higher-layer triggered events 145
  - Initialization 145
  - Message processing events and sequencing rules 145
  - Other local events 158
  - Timer events 158
  - Timers 145
- Implementer - security considerations 200
- Index of security parameters 200
- Informative references 19
- Introduction 18
- Iservermanagement server
  - Abstract data model 141
  - Higher-layer triggered events 141
  - Initialization 141
  - Message processing events and sequencing rules 141
  - Other local events 143
  - Timer events 143
  - Timers 141
- I servicemanagement server
  - Abstract data model 139
  - Higher-layer triggered events 139
  - Initialization 139
  - Message processing events and sequencing rules 139
  - Other local events 141
  - Timer events 141
  - Timers 139
- Isharepointitemgmt server
  - Abstract data model 188
  - Higher-layer triggered events 188
  - Initialization 188
  - Message processing events and sequencing rules 188
  - Other local events 189
  - Timer events 189
  - Timers 188
- Istoragemanagement server
  - Abstract data model 158
  - Higher-layer triggered events 158
  - Initialization 158

- Message processing events and sequencing rules 159
- Other local events 172
- Timer events 172
- Timers 158
- Iusermanagement server
  - Abstract data model 172
  - Higher-layer triggered events 173
  - Initialization 172
  - Message processing events and sequencing rules 173
  - Other local events 188
  - Timer events 188
  - Timers 172
- Iwindowsphonemanagement server
  - Abstract data model 190
  - Higher-layer triggered events 190
  - Initialization 190
  - Message processing events and sequencing rules 190
  - Other local events 193
  - Timer events 193
  - Timers 190

## **M**

- Messages
  - transport 21

## **N**

- Namespaces 21
- Normative references 19

## **O**

- Overview (synopsis) 19

## **P**

- Parameters - security index 200
- Preconditions 20
- Prerequisites 20
- Product behavior 215
- Protocol examples
  - Create a Folder 197
  - Get Server Folders 195
  - Get Server Information 194
  - Login 194
  - Logout 198
  - Retrieve the Metadata for Items within a Folder 196
  - Upload a File 198

## **R**

- References
  - informative 19
  - normative 19
- Relationship to other protocols 20

## **S**

- Security
  - implementer considerations 200
  - parameter index 200
- Sessionservice server
  - Abstract data model 74

- Higher-layer triggered events 74
- Initialization 74
- Message processing events and sequencing rules 74
- Other local events 77
- Timer events 77
- Timers 74
- Standards assignments 20

## **T**

- Tracking changes 216
- Transport 21
  - common data types 21
  - namespaces 21

## **V**

- Vendor-extensible fields 20
- Versioning 20