

# MiCollab Advanced Messaging Web Services API Functions Reference Developer Resources Document

For version 6.1 and above

## Notice

The information contained in this document is believed to be accurate in all respects but is not warranted by Mitel Networks™ Corporation (MITEL®). Mitel makes no warranty of any kind with regards to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The information is subject to change without notice and should not be construed in any way as a commitment by Mitel or any of its affiliates or subsidiaries. Mitel and its affiliates and subsidiaries assume no responsibility for any errors or omissions in this document. Revisions of this document or new editions of it may be issued to incorporate such changes.

No part of this document can be reproduced or transmitted in any form or by any means - electronic or mechanical - for any purpose without written permission from Mitel Networks Corporation.

## Trademarks

The trademarks, service marks, logos and graphics (collectively "Trademarks") appearing on Mitel's Internet sites or in its publications are registered and unregistered trademarks of Mitel Networks Corporation (MNC) or its subsidiaries (collectively "Mitel") or others. Use of the Trademarks is prohibited without the express consent from Mitel. Please contact our legal department at [legal@mitel.com](mailto:legal@mitel.com) for additional information. For a list of the worldwide Mitel Networks Corporation registered trademarks, please refer to the website: <http://www.mitel.com/trademarks>.

© Copyright 2016, Mitel Networks Corporation

All rights reserved

# LICENSE AGREEMENT

## Mitel Networks Coporation LICENSE AGREEMENT

### READ CAREFULLY BEFORE USING MiCollab Advanced Messaging

BY USING THE MiCollab AM SOFTWARE, CUSTOMER ACCEPTS THE TERMS OF THIS LICENSE AGREEMENT. IF CUSTOMER DOES NOT AGREE TO THIS LICENSE AGREEMENT, PROMPTLY RETURN THE MiCollab AM SOFTWARE, DOCUMENTATION AND ALL ITEMS SUPPLIED WITH IT TO THE PERSON WHO SUPPLIED THEM TO YOU. YOUR MONEY WILL BE REFUNDED IN FULL.

#### 1. Definitions

**Software.** *Software* means the computer software and accompanying documentation known as MiCollab AM, and all additions, corrections, updates, and enhancements provided under this agreement, together with copies of such items

#### 2. License.

Mitel Networks Coporation (*Mitel*) grants Customer a nonexclusive, nontransferable license to use the Software in accordance with terms of this agreement. Mitel remains the owner of the Software and of any copy of the Software which Customer may make.

#### 3. Proprietary Rights.

The Software is the valuable trade secret property of Mitel and is protected under U.S. copyright laws and international treaties. UNAUTHORIZED COPYING, USE, OR DISCLOSURE OF THE SOFTWARE IS AN INFRINGEMENT OF Mitel's COPYRIGHT, TRADE SECRET, AND OTHER INTELLECTUAL PROPERTY RIGHTS.

#### 4. Use Restrictions.

4.1 Customer may make copies of the Software for its own internal use on the number of workstations licensed to Customer and as a backup; provided that Customer (a) includes in and on such copies all notices of copyright and proprietary rights appearing in and on the Software, and (b) erases or destroys such copies when they are no longer required. Customer will not permit anyone else to copy any portion of the Software. Customer may not copy the written materials accompanying the Software.

4.2 Customer shall not disassemble, unassemble, reverse engineer, or decode any portion of the Software or attempt to bypass the metering device contained in the Software. Except as expressly set forth in this Agreement, Customer shall not modify any portion of the Software, or merge or embed the Software into another computer program.

#### 5. Distribution.

Customer is licensed to create derivatives of the MiCollab AM Web Services API and to incorporate such derivatives of the MiCollab AM Web Services API into Customer's applications. However, Customer has no right to distribute any components of the MiCollab AM Web Services API except as incorporated into Customer's application as provided in the preceding sentence. Customer may use files included in the MiCollab AM Web Services API to build Customer's applications. However, Customer has no right to

distribute any of such files or any related documentation. Customer may not distribute the Software as a whole or distribute a developer's kit created using the Software.

## **6. Limited Warranty and Disclaimer of Implied Warranties.**

6.1 Mitel warrants that, upon delivery by Mitel or Mitel's Vendor to you, the Software will perform substantially in accordance with the specifications stated in the accompanying written materials. Mitel does not warrant that the Software is free from all bugs, errors, and omissions. This Limited Warranty is void if the failure of the Software has resulted from your improper use or misapplication of it.

6.2 If the Software does not comply with the Limited Warranty, Mitel will use reasonable efforts to correct the noncompliance either by, at Mitel's option: (a) repair or replacement of the Software, or (b) return of a fair portion of the price paid. You must return all copies of the Software to Mitel or Mitel's vendor with a copy of your receipt within 90 days of the date you received the Software to receive this remedy. Any replacement Software will be warranted for the remainder of the original 90 day warranty period or 30 days from the date you received the replacement, whichever is longer.

6.3 THE WARRANTY PROVIDED UNDER SECTION 6.1 OF THIS LIMITED WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED. Mitel MAKES NO OTHER WARRANTIES, EXPRESS OR IMPLIED. Mitel DISCLAIMS ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT.

## **7. Term and Termination**

7.1 Customer may terminate the license at any time.

7.2 If Customer fails to comply with any provision of this agreement, Mitel may, in addition to all other remedies available, terminate the license by notice to Customer. Upon termination of the license, Customer shall cease using the Software. Customer's obligations to hold the Software in confidence and all related obligations shall survive termination of the license.

## **8. Assignment.**

This Software is licensed only to Customer. Customer shall not transfer the Software or this license to anyone. In no event may Customer transfer, assign, rent, lease, sell, or otherwise dispose of the Software on a temporary or permanent basis.

## **9. Miscellaneous**

9.1 Mitel's liability (whether in tort, contract, or otherwise and notwithstanding any fault, negligence, product liability, or strict liability of Mitel) under this agreement or with regard to any Software or documentation will in no event exceed the amount paid by Customer to Mitel for the Software. IN NO EVENT WILL Mitel BE LIABLE (WHETHER IN CONTRACT, WARRANTY, TORT, PRODUCT LIABILITY, STRICT LIABILITY OR OTHERWISE) TO CUSTOMER OR ANY OTHER PERSON FOR ANY INDIRECT, INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES.

9.2 This agreement will be governed by the laws of the State of Washington. Customer consents to the nonexclusive jurisdiction of the federal and state courts in Seattle, Washington, in any action arising out of or in connection with this agreement. No agent, employee, or representative of Mitel has any authority to bind Mitel to any affirmation, representation, or warranty, and, unless such is specifically included within this written agreement, it shall not be enforceable by Customer. If any provision of this agreement is held by a court of competent jurisdiction to be invalid, illegal, or unenforceable, the remainder of this agreement shall remain in full force and effect.

U.S. GOVERNMENT RESTRICTED RIGHTS LEGEND

The Software and documentation are provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subdivision (c)(1)(II) of The Rights in Technical Data and Computer Software clause at 48 C.F.R. Section 252.227-7013 or in subparagraphs (c)(1) and (2) of the Commercial Computer Software Restricted Rights clause at C.F.R. Section 52.227-19, as applicable. Contractor/manufacturer is Mitel Networks Coporation

# Contents

<b>LICENSE AGREEMENT</b>	<b>3</b>
<b>Preface</b>	<b>28</b>
References	28
Documentation	28
Documentation Updates	29
Help	29
Document Conventions	29
<b>MiCollab AM Web Services API Functions List</b>	<b>31</b>
Session Management	31
Mailbox Configuration Functions	31
Subscriber Availability Functions	34
Recording Object Functions	36
Message Management Functions	36
Line Management Functions	37
Miscellaneous Functions	38
<b>MiCollab AM Web Services API Functions</b>	<b>40</b>
CXLogon	40
Syntax	40
Input Parameters	40
Output Parameters	41
XML Return	41
Errors	41
Comments	41
See Also	41
CXLogoff	41
Syntax	42
Input Parameters	42
Output Parameters	42
Errors	42
Comments	42

See Also	42
CXMBSearch	42
Syntax	43
Input Parameters	43
Output Parameters	47
XML Return	47
Errors	47
Comments	48
Example	48
See Also	49
CXMBSearchCountGet	49
Syntax	50
Input Parameters	50
Output Parameters	50
XML Return	50
Errors	51
Comments	51
Example	51
See Also	52
CXMBGet	52
Syntax	52
Input Parameters	52
Output Parameters	53
Errors	53
See Also	53
CXMBAdd	54
Syntax	54
Input Parameters	54
Output Parameters	55
Errors	55
See Also	55
CXMBUpdate	55
Syntax	55

Input Parameters	56
Output Parameters	57
Errors	57
Comments	57
Example	58
See Also	58
CXMBTemplateUpdate	58
Syntax	58
Input Parameters	59
Output Parameters	60
Errors	60
Comments	60
Example	61
See Also	61
CXMBSubIMNCalloutArrayGet	61
Syntax	61
Input Parameters	61
Output Parameters	62
Errors	62
See Also	62
CXMBSubIMNCalloutArrayUpdate	63
Syntax	63
Input Parameters	63
Output Parameters	64
Errors	64
Comments	65
Example	65
See Also	65
CXMBSubEmailGet	65
Syntax	65
Input Parameters	66
Output Parameters	66
Errors	66



See Also	67
CXMBSubEmailUpdate	67
Syntax	67
Input Parameters	67
Output Parameters	67
Errors	67
Comments	68
Example	68
See Also	68
CXMBDistListMemberArrayGet	68
Syntax	68
Input Parameters	69
Output Parameters	70
Errors	70
Comments	70
Example	70
See Also	71
CXMBDistListAllowedMBIDArrayGet	72
Syntax	72
Input Parameters	72
Output Parameters	72
Errors	72
See Also	73
CXMBDistListAllowedMBIDArrayUpdate	73
Syntax	73
Input Parameters	73
Syntax	73
Input Parameters	74
Output Parameters	74
Errors	74
Comments	75
Example	75
See Also	76

CXMBCPActionArrayGet	76
Syntax	76
Input Parameters	76
Output Parameters	76
Errors	79
See Also	79
CXMBCPActionArrayUpdate	79
Syntax	79
Input Parameters	80
Output Parameters	81
Errors	81
Comments	81
Example	81
See Also	83
CXMailboxDefaults	83
Syntax	83
Input Parameters	83
Output Parameters	83
XML Return	85
Errors	85
CXMailboxPasswordChange	85
Syntax	85
Input Parameters	86
Output Parameters	86
Errors	86
Example	87
See Also	87
CXPromptRecordingListGet	87
Syntax	87
Input Parameters	87
Output Parameters	88
XML Return	89
Errors	89

See Also	89
CXPromptRecordingUpdate	89
Syntax	89
Input Parameters	90
Output Parameters	90
Errors	90
Comments	91
Example	91
See Also	92
CXMBGroupAssignmentListGet	92
Syntax	92
Input Parameters	92
Output Parameters	93
XML Return	93
Errors	93
See Also	93
CXMBGroupAssignmentAdd	93
Syntax	94
Input Parameters	94
Output Parameters	94
XML Return	94
Errors	94
See Also	95
CXMBGroupAssignmentDelete	95
Syntax	95
Input Parameters	95
Output Parameters	96
XML Return	96
Errors	96
See Also	96
CXDeviceListGet	96
Syntax	96
Input Parameters	97

Output Parameters	97
XML Return	97
Errors	97
Example	97
See Also	98
CXDeviceAdd	98
Syntax	98
Input Parameters	98
Output Parameters	98
XML Return	99
Errors	99
Example	99
See Also	99
CXDeviceUpdate	99
Syntax	99
Input Parameters	100
Output Parameters	100
XML Return	100
Errors	100
Example	101
Deleting a Device	101
See Also	101
CXDeviceFinalize	101
Syntax	101
Input Parameters	102
Output Parameters	102
XML Return	102
Errors	102
See Also	102
CXMBSpeechAliasListGet	102
Syntax	103
Input Parameters	103
Output Parameters	103

XML Return	103
Errors	104
See Also	104
CXMBSpeechAliasAdd	104
Syntax	104
Input Parameters	104
Output Parameters	105
XML Return	105
Errors	105
See Also	106
CXMBSpeechAliasUpdate	106
Syntax	106
Input Parameters	106
Output Parameters	107
XML Return	107
Errors	107
See Also	107
CXMBSpeechAliasDelete	107
Syntax	107
Input Parameters	108
Output Parameters	108
XML Return	108
Errors	108
See Also	108
CXMBDataGet	109
Syntax	109
Input Parameters	109
Output Parameters	110
Errors	111
See Also	111
CXMBDataAdd	111
Syntax	111
Input Parameters	112

Output Parameters	113
Errors	113
See Also	113
CXMBDataUpdate	113
Syntax	113
Input Parameters	114
Output Parameters	116
Errors	116
Comments	116
Example	116
See Also	117
CXMBDataTemplateUpdate	117
Syntax	117
Input Parameters	117
Output Parameters	118
Errors	119
Comments	119
Example	119
See Also	120
CXMBDataDelete	120
Syntax	120
Input Parameters	120
Return Value	121
XML Return (range delete only)	121
Errors	121
Comments	121
Example	121
See Also	122
CXMBDataRenumber	122
Syntax	122
Input Parameters	122
Return Value	123
Errors	123

Example	123
See Also	124
CXAvailabilityGreetingListGet	124
Syntax	124
Input Parameters	124
Output Parameters	124
XML Return	125
Errors	125
Example	125
See Also	125
CXAvailabilityActivate	125
Syntax	126
Input Parameters	126
Output Parameters	127
Errors	127
Example	128
See Also	128
CXAvailabilityEnableSchedules	128
Syntax	128
Input Parameters	128
Output Parameters	129
Errors	129
Example	129
See Also	129
CXAvailabilityEnable	129
Syntax	129
Input Parameters	130
Output Parameters	130
Errors	130
Example	130
See Also	130
CXAvailabilityDisable	131
Syntax	131

Input Parameters	131
Output Parameters	131
Errors	131
Example	131
See Also	132
CXAvailabilityActiveStateGet	132
Syntax	132
Input Parameters	132
Output Parameters	133
XML Return	135
Example	136
Errors	136
See Also	136
Syntax	137
Input Parameters	137
Output Parameters	137
XML Return	137
Errors	138
Example	138
See Also	138
CXAvailabilityStateDeviceListGet	138
Syntax	138
Input Parameters	139
Output Parameters	139
XML Return	139
Example	140
Errors	140
See Also	141
Syntax	141
Input Parameters	141
Output Parameters	141
XML Return	142
Errors	142



Example	142
See Also	142
CXAvailabilityAutoOverrideListGet	143
Syntax	143
Input Parameters	143
Output Parameters	143
XML Return	144
Errors	144
Example	145
See Also	145
CXAvailabilityDayListGet	145
Syntax	145
Input Parameters	145
Output Parameters	146
XML Return	146
Errors	147
Example	147
See Also	147
CXAvailabilityDayAdd	147
Syntax	148
Input Parameters	148
Output Parameters	148
XML Return	148
Errors	148
Example	149
See Also	149
CXAvailabilityDayUpdate	150
Syntax	150
Input Parameters	150
Output Parameters	151
Errors	151
Example	151
See Also	152

CXAvailabilityDayDelete	153
Syntax	153
Input Parameters	153
Output Parameters	153
Errors	153
Example	153
See Also	154
CXAvailabilityStateListGet	154
Syntax	154
Input Parameters	154
Output Parameters	155
XML Return	155
Errors	156
Example	156
See Also	157
CXAvailabilityStateUpdate	157
Syntax	157
Input Parameters	157
Output Parameters	158
Errors	158
Example	158
See Also	158
CXAvailabilitySubscriberReset	158
Syntax	159
Input Parameters	159
Output Parameters	159
Errors	159
Example	160
CXObjectCreate	160
Syntax	160
Input Parameters	160
Output Parameters	160
XML Return	160

Comments	161
See Also	161
CXObjectOpen	161
Syntax	161
Input Parameters	161
Output Parameters	162
Errors	162
Comments	162
Example	162
See Also	162
CXObjectClose	163
Syntax	163
Input Parameters	163
Output Parameters	163
Comments	164
Errors	164
Example	164
See Also	164
CXObjectUpload	164
Syntax	164
Input Parameters	165
Output Parameters	165
Example	165
See Also	165
CXObjectDownload	165
Syntax	166
Input Parameters	166
Output Parameters	166
XML Return	166
Example	166
See Also	167
CXMessageSearch	167
Syntax	167

Input Parameters	168
Output Parameters	170
XML Return	170
Errors	171
Comments	171
Example	171
See Also	172
Syntax	172
Input Parameters	173
Output Parameters	173
XML Return	173
Example	175
See Also	175
CXMessageUpdate	175
Syntax	176
Input Parameters	176
Output Parameters	176
Example	177
See Also	177
CXMessageRetrieveAttachments	177
Syntax	177
Input Parameters	177
Output Parameters	178
XML Return	178
See Also	178
CXMessageSend	178
Syntax	178
Input Parameters	179
Output Parameters	179
XML Return	182
Errors	182
Example	182
See Also	183

CXMWIClear	183
Syntax	183
Input Parameters	183
Output Parameters	183
Example	183
See Also	184
CXIMNCancel	184
Syntax	184
Input Parameters	184
Output Parameters	184
Comments	184
Example	184
See Also	185
CXLMLineAttach	185
Syntax	185
Input Parameters	185
Output Parameters	185
XML Return	186
Errors	186
Comments	186
Example	186
See Also	186
CXLMLineDetach	186
Syntax	186
Input Parameters	187
Output Parameters	187
Errors	187
Comments	187
See Also	187
CXLMOBJECTCreate	187
Syntax	187
Input Parameters	188
Output Parameters	188

XML Return	188
Errors	188
Comments	188
See Also	188
CXLMOBJECTOpen	188
Syntax	189
Input Parameters	189
Output Parameters	189
Errors	190
Comments	190
Example	190
See Also	190
CXLMOBJECTClose	190
Syntax	190
Input Parameters	191
Output Parameters	191
Comments	191
Errors	192
Example	192
See Also	192
CXLMOBJECTUpload	192
Syntax	192
Input Parameters	192
Output Parameters	193
Errors	193
Example	193
See Also	193
CXLMOBJECTDownload	193
Syntax	194
Input Parameters	194
Output Parameters	194
XML Return	194
Errors	194

Example	195
See Also	195
CXLMLinePlay	195
Syntax	195
Input Parameters	195
Output Parameters	196
Errors	196
Comments	196
Example	196
See Also	197
CXLMLineRecord	197
Syntax	197
Input Parameters	197
Output Parameters	197
Errors	198
Comments	198
Example	198
See Also	198
CXLMLineStatusGet	198
Syntax	198
Input Parameters	198
Output Parameters	199
XML Return	199
Errors	199
Comments	199
See Also	199
CXLMLineWait	199
Syntax	199
Input Parameters	200
Output Parameters	200
XML Return	200
Errors	200
Example	201

See Also	201
CXLMLineTransferStart	201
Syntax	201
Input Parameters	201
Output Parameters	202
Errors	202
Comments	202
See Also	203
CXLMLineTransferEnd	203
Syntax	203
Input Parameters	203
Output Parameters	203
Errors	204
Comments	204
See Also	204
CXGetVersionInfo	204
Syntax	204
Input Parameters	204
Output Parameters	204
XML Return	204
Errors	205
CXPasswordResetRequestCreate	205
Syntax	205
Input Parameters	205
Output Parameters	205
Errors	206
CXPasswordReset	206
Syntax	206
Input Parameters	206
Output Parameters	206
Errors	207
CXGetMgmtData	207
Syntax	207



Input Parameters	207
Output Parameters	207
XML Return	208
Errors	208
CXLanguagePackListGet	208
Syntax	208
Input Parameters	208
Output Parameters	209
XML Return	209
Errors	209
CXSpeechCmdListGet	210
Syntax	210
Input Parameters	210
Output Parameters	210
XML Return	210
Errors	212
CXSpeechCmdGet	212
Syntax	212
Input Parameters	212
Output Parameters	212
XML Return	212
Errors	213
CXSpeechCmdAdd	213
Syntax	213
Input Parameters	214
Output Parameters	214
XML Return	214
Errors	214
CXSpeechCmdUpdate	214
Input Parameters	215
Output Parameters	215
Errors	215
Comments	215

Example	216
CXSpeechCmdDelete	216
Syntax	216
Input Parameters	216
Output Parameters	217
Errors	217
CXSpeechCmdNamesListByCollectionTypeGet	217
Syntax	217
Input Parameters	217
Output Parameters	218
XML Return	218
Errors	219
CXMBGroupTypeListGet	219
Syntax	219
Input Parameters	219
Output Parameters	219
XML Return	220
Errors	220
CXMBGroupListGet	220
Syntax	221
Input Parameters	221
Output Parameters	221
XML Return	221
Errors	222
CXListFirst	222
Syntax	222
Input Parameters	222
Output Parameters	224
XML Return	224
Errors	225
Examples	225
See Also	226
CXListNext	226

Syntax	226
Input Parameters	227
Output Parameters	227
Errors	227
See Also	227
CXListTotal	227
Syntax	227
Input Parameters	228
Output Parameters	229
XML Return	229
Errors	229
Example	229
<b>MiCollab AM Web Services API Structures</b>	<b>230</b>
MBCommon	230
MBSubSpecific	232
MDBListSpecific	244
MBCPSpecific	245
Device	247
Address	250
Attachment	251
Message	251
StoredMessage	255
SpeechAliasCollection	255
SpeechCmd	256
AvailabilityDay	257
AvailabilityState	259
AvailabilityAutoOverride	267
AvailabilityPointInstalled	270
AvailabilityGreeting	271

# Preface

The Web Services API Functions Guide provides the detail information on individual functions that you need to develop applications that integrate with MiCollab Advanced Messaging (MiCollab AM), using the MiCollab AM Simple Object Access Protocol (SOAP) Application Programming Interface (API).

Please refer to the MiCollab AM Web Services API Developer's Guide document for an introduction to the programming model, concepts, error handling and other general aspects of the API. Then use this document for the details regarding a particular function.

The chapter, [MiCollab AM Web Services API Functions](#), describes each function. Any nested structures that is used within a function is either documented inline as part of the function documentation or is documented separately in the [Mailbox Configuration Functions](#) chapter.

For a categorized list of the functions, refer to the chapter [MiCollab AM Web Services API Functions List](#).

## References

A catalog of technical documentation is included on the MiCollab AM Installation Media. If you are installing any advanced applications, such as Networking and Fax Server applications, you should refer to the appropriate technical documentation for application and installation information.

## Documentation

The technical documentation is produced in the PDF format and requires the PDF reader to view it. The documentation set for this MiCollab AM includes the following documents and resources:

- **Developer Resources.** Contains programming guides and API references for developers for integrating the server clients and web applications with MiCollab AM.
- **Integration Technical Notes (ITN).** Contains a set of guides that describe the integration methods and instructions for a variety of phone systems to work with MiCollab AM. The ITNs are generally used by resellers or administrators who are experienced with MiCollab AM and familiar with the integration procedures and terminology.
- **Quick Reference Card (QRC).** Contains shortcuts and quick instructions telling subscribers how to access and use the messaging system.
- **Server Documentation.** Available as a PDF only. Contains administrative guides for administrators about installing, configuring, and administering the messaging system, and user guides for subscribers about accessing the messaging system and checking and sending messages.
- **Spare Parts Documentation.** Contains a set of guides that describe the instructions for installing and configuring hardware parts to work with MiCollab AM. These documents are written for Mitel certified MiCollab AM technicians who are experienced with MiCollab AM and familiar with the procedures and terminology.
- **Software Release Notice (SRN).** This notice introduces the new features, capabilities, and hardware/software requirements for the corresponding MiCollab AM version.

## Documentation Updates

Documentation updates may be available from the following sources:

- Mitel certified technicians can view or download documents and program files from our partner web site: [connect.mitel.com/connect](https://connect.mitel.com/connect)

## Help

The primary source of information about MiCollab AM is the online help available within any of its administrative utilities. You can access **Help** as follows:

- Click the **Help** button in the dialog box or window in which you are working
- Press the **F1** key at any time.

## Document Conventions

The following conventions are used in this document:

- **Key Names.** Names of keys on the keyboard are shown in a box.

Example: **Enter**

When two keys must be pressed simultaneously, they are joined by a + sign.

Example: **Alt** + **Tab**

- **Reference to Document.** *Italics* fonts can also signify the titles of other documents.

Example: Refer to *System Installation Guide*.

- **UI Element Names.** Names of UI elements such as dialog windows, screens, menu items, tabs, buttons, icons, etc. are shown in bold.

Example: On the **Startup** screen, click the **Start** icon.

- **User Input.** Information required to be typed is shown in italics.

Example: Type the password *voicemail*.

- **Warning, Caution, Important, and Notes.** Text for the contents that require attention are shown as follows:

**WARNING** A warning paragraph advises you of circumstances that can result in the loss of data, harm to the system server platform, or personal harm.

**CAUTION** Failure to follow these recommendations can result in unauthorized access to the system and consequent loss of data.

**IMPORTANT** An important paragraph gives decision-making information or informs you of the order in which tasks need to be completed.

**NOTE** A note gives additional information, provides an explanation, or indicates an exception to the information in the preceding text.

# MiCollab AM Web Services API Functions List

This section describes the MiCollab AM Web Services API functions, which are broken into groups of associated functions. Information is provided on how the functions work together, on typical sequencing of functions, and on other programming tips.

## Session Management

Session Management functions allow your application to establish and terminate a session with the server. The server maintains a context for your application's session consisting of the administrator account or mailbox. If your application ends without logging off from the server, the server will terminate the session after it detects no activity for a predetermined length of time. Because each session uses valuable resources on the server, care should be taken within your application to log off the server as soon as a session no longer needs these resources.

Table 1. Session management functions

Function	Description
CXLogon	Logs on to an administrator account or mailbox; application specifies an administrator account or mailbox number, and password.
CXLogoff	Logs off an administrator account or mailbox that was previously logged on to using the <b>CXLogon</b> function.

## Mailbox Configuration Functions

The mailbox functions give your application access to the MiCollab AM directory information and allow for the creation, deletion, editing and viewing of mailbox properties.

Table 2. Mailbox configuration functions

Function	Description
CXMBSearch	Returns the selected attributes of mailboxes that match the specified search criteria.
CXMBSearchCountGet	Returns a count of the number of mailboxes that

match the specified search criteria.

CXMBGet	Loads the basic mailbox structure for the specified mailbox.
CXMBAdd	Adds a new mailbox, with just the basic mailbox structure.
CXMBUpdate	Updates the basic mailbox structure for the specified mailbox.
CXMBTemplateUpdate	Updates the basic mailbox record for the specified mailboxes using a template mailbox (COS mailbox).
CXMBSubIMNCalloutArrayGet	Loads the ordered array of immediate message notification callout entries for the specified Subscriber mailbox.
CXMBSubIMNCalloutArrayUpdate	Updates the ordered array of immediate message notification callout entries for the specified Subscriber mailbox.
CXMBSubEmailGet	Loads the email configuration information for the specified Subscriber mailbox.
CXMBSubEmailUpdate	Updates the email configuration information for the specified Subscriber mailbox.
CXMBDistListMemberArrayGet	Loads the array of members of the specified Distribution List mailbox.
CXMBDistListMemberArrayUpdate	Updates the list of members of the specified Distribution List mailbox.
CXMBDistListAllowedMBIDArrayGet	Loads the array of mailbox numbers which are allowed to address the specified Distribution List mailbox.
CXMBDistListAllowedMBIDArrayUpdate	Updates the array of mailbox numbers which are allowed to address the specified Distribution List mailbox.
CXMBCPActionArrayGet	Loads the array of actions for the specified Call Processor mailbox.
CXMBCPActionArrayUpdate	Updates the list of actions for the specified Call Processor mailbox.



CXMailboxDefaults	Retrieves default and system settings for mailbox properties.
CXMailboxPasswordChange	Changes the password for a specified mailbox.
CXPromptRecordingListGet	Retrieves the list of Recording structures associated with a mailbox.
CXPromptRecordingUpdate	Updates the PromptRecording structure.
CXMBGroupAssignmentListGet	Retrieves the list of Groups to which a specific mailbox is assigned.
CXMBGroupAssignmentAdd	Adds a Group to the list of Groups assigned to the specified Mailbox.
CXMBGroupAssignmentDelete	Deletes a Group from the list of Groups assigned to the specified Mailbox.
CXDeviceListGet	Get the list of devices for a particular subscriber.
CXDeviceAdd	Add a device record.
CXDeviceUpdate	Update properties of a device record. This function is also used to delete a device record.
CXDeviceFinalize	Finalize critical settings for the devices associated with the specified Subscriber. This function insures that important device settings that are interrelated among devices are correctly set.
CXMBSpeechAliasListGet	Get the list of speech aliases for a particular subscriber or a list of all mailbox speech aliases on the system.
CXMBSpeechAliasAdd	Add a new speech alias for a particular subscriber.
CXMBSpeechAliasUpdate	Update a speech alias.
CXMBSpeechAliasDelete	Delete a speech alias.
CXMBDataGet	Loads the requested data categories for the specified mailbox.
CXMBDataAdd	Adds a new mailbox, including the basic mailbox structure plus optional additional mailbox data categories.

CXMBDataUpdate	Updates an existing mailbox. This can include a variety of data categories.
CXMBDataTemplateUpdate	Updates the specified mailboxes using a template mailbox (COS mailbox).
CXMBDataDelete	Deletes one or more mailboxes.
CXMBDataRenummer	Renumbers an existing mailbox.

## Subscriber Availability Functions

This section deals with the changed Subscriber Availability paradigm introduced in 5.1 . This change has had dramatic effects on the related SOAP functions and data structures supported in MiCollab AM 5.1

At the core of the change is a new entity called the Availability state. The Availability state structure contains all the settings that affect the Availability experience of the caller including the list of devices to call. All other entities are only there to support the determination of the active Availability state at the instant of time the incoming call arrives. For example, the new Availability day entity is used to determine which of the subscriber's Availability states is active based on the day schedule. The Availability auto overrides entities are used to process the Availability *In Point* sources (Calendar, Presence, Location, Time, and Settings) to determine if there is an automatically selected Availability state to use instead of the one determined by the Availability day schedules.

The **CXAvailabilityActivate** SOAP function can manually select an Availability state to override both the scheduled and automatically determined Availability state. Conversely, the **CXAvailabilityEnableSchedules** SOAP function turns off the manual override and returns to automatic processing. To determine which Availability state is active for a subscriber and its associated devices, use the **CXAvailabilityActiveStateGet** and **CXAvailabilityStateDeviceListGet** SOAP functions. Use the **CXAvailabilityEnable** and the **CXAvailabilityDisable** SOAP functions to control whether Availability call processing is to be used for a particular subscriber.

The new Availability Class of Service (COS) mailbox is the glue that holds the multiple Availability day entities, the multiple Availability state entities, and the multiple Availability automatic override entities together. Subscriber Availability settings are based on an associated Availability COS mailbox. The subscriber mailboxes initially use copies of the Availability COS mailbox Availability day and Availability state entities. However, since they are copies, the individual settings are allowed to diverge from the Availability COS mailbox settings. Subscriber mailboxes always use the Availability COS's automatic override entities, since they do not have their own Availability Automatic Override entities.

The Availability day, Availability state and Availability automatic overrides are represented by the AvailabilityDay, AvailabilityState and AvailabilityAutoOverride structures described at the end of this document.

The **AvailabilityDay** data is managed by the **CXAvailabilityDayListGet**, **CXAvailabilityDayAdd**, **CXAvailabilityDayUpdate**, and **CXAvailabilityDayDelete** SOAP functions. The **AvailabilityState** data is read using **CXAvailabilityStateListGet** and changed using **CXAvailabilityStateUpdate**. The **AvailabilityAutoOverride** data is read using the **CXAvailabilityAutoOverrideListGet** SOAP function.

Each subscriber using Availability must be based on an Availability COS mailbox. While a subscriber cannot have an Availability state that is not part of the Availability COS Availability state list, the Availability state settings as well as the Availability day schedules can be customized for each subscriber independently.

Use the SOAP function **CXAvailabilitySubscriberReset** to reset all Availability day and Availability state data to match the current Availability COS's Availability day and Availability state data. This can be done for a list of subscribers or for all the current subscribers associated with the Availability COS mailbox.

Following are the Availability SOAP functions that allow management of subscriber Availability:

Table 3.Subscriber Availability Options

Function	Description
CXAvailabilityGreetingListGet	Gets all Availability Greeting records or a specific Availability Greeting record for a Subscriber or Availability Class of Service mailbox.
CXAvailabilityActivate	Sets an Availability State or Availability Day manual override for a specific mailbox for a specified range of time.
CXAvailabilityEnableSchedules	Enables normal automatic Availability processing.
CXAvailabilityEnable	Enables Availability processing for a subscriber.
CXAvailabilityDisable	Disables Availability processing for a subscriber.
CXAvailabilityActiveStateGet	Gets the active Availability State of the specified mailbox for the time and time-zone specified.
CXAvailabilityStateDeviceListGet	Gets the Availability State Device List for the specified Availability State.
CXAvailabilityAutoOverrideListGet	Gets all Availability Auto Override records for a mailbox.
CXAvailabilityDayListGet	Gets all Availability Day records or a specific Availability Day record for a mailbox.
CXAvailabilityDayAdd	Adds a new Availability Day record for an Availability Class of Service mailbox.
CXAvailabilityDayUpdate	Updates an existing Availability Day record for a mailbox.
CXAvailabilityDayDelete	Deletes an Availability Day record for an Availability Class of Service mailbox.

CXAvailabilityStateListGet	Gets Availability State records for a mailbox.
CXAvailabilityStateUpdate	Updates an existing Availability State record for a mailbox.
CXAvailabilitySubscriberReset	Resets a list of subscriber mailboxes to use the current Availability settings of the specified Availability Class of Service mailbox's states and schedules.

## Recording Object Functions

The recording object functions are used for creating, opening, closing and accessing the contents of various types of recording objects.

Table 4. Recording object functions

Function	Description
CXObjectCreate	Creates a new recording object.
CXObjectOpen	Opens an existing recording object.
CXObjectClose	Closes/Saves the recording object.
CXObjectUpload	Uploads binary content into the recording object.
CXObjectDownload	Downloads the binary content in the recording object.

## Message Management Functions

The message management functions are used for enumerating, creating, and manipulating messages.

Table 5. Message management functions

Function	Description
CXMessageSearch	Retrieves important properties for messages that match specific criteria.
CXMessageGet	Retrieves all the properties for a message.
CXMessageUpdate	Updates the properties of a message.

CXMessageRetrieveAttachments	Retrieves the list of attachments for a message.
CXMessageSend	Sends a new message, a reply to an existing message, or forwards an existing message.
CXMWIClear	Clears the Message Waiting Indicator for the subscriber.
CXIMNCancel	Cancels Immediate Message Notification for the subscriber.

## Line Management Functions

The line management functions are used for placing and transferring calls and recording and playing messages and recording objects.

Table 6. Line management functions

Function	Description
CXLMLineAttach	Calls a specified number and establishes a line for playing or recording audio.
CXLMLineDetach	Detaches (hangs up) the attached line.
CXLMLObjectCreate	Creates a new recording object.
CXLMLObjectOpen	Opens an existing recording object.
CXLMLObjectClose	Closes/Saves the recording object.
CXLMLObjectUpload	Uploads binary content into the recording object.
CXLMLObjectDownload	Downloads the binary content in the recording object.
CXLMLLinePlay	Starts playback on the attached line.
CXLMLLineRecord	Starts recording audio on the attached line.
CXLMLLineStatusGet	Returns the status of the attached line.
CXLMLLineWait	Waits for termination of telephone recording or playback.
CXLMLLineTransferStart	Initiates a call transfer to the specified telephone number or extension.

## Miscellaneous Functions

The utility functions provide version and other information about the server. **CXGetVersionInfo** can be used prior to using **CXLogon** to verify that the API version is compatible with the server.

Table 7. Miscellaneous functions

Function	Description
CXGetVersionInfo	Retrieves version information from the server.
CXPasswordResetRequestCreate	Creates a Password Reset Request and sends an e-mail to the user containing a link that allows the user to reset the password.
CXPasswordReset	Resets the password for a user.
CXGetMgmtData	Retrieves information about the server such as site name, software version and serial number.
CXLanguagePackListGet	Retrieves the list of the language packs that are configured on this server.
CXSpeechCmdListGet	Retrieves the list of the configured speech commands.
CXSpeechCmdGet	Retrieves a speech command.
CXSpeechCmdAdd	Adds a new speech command.
CXSpeechCmdUpdate	Updates the specified speech command.
CXSpeechCmdDelete	Delete a speech command.
CXSpeechCmdNamesListByCollectionTypeGet	Retrieves the SpeechCmdID and Name fields for all the speech commands that belong to a particular speech alias collection type.
CXMBGroupTypeListGet	Retrieves the list of configured Group Types.
CXMBGroupListGet	Retrieves the list of Groups assigned to a specific Group Type.
CXListFirst	Retrieves the first item plus a specified number of

additional items in an enumeration. Numerous types of data can be enumerated using this function such as mailbox summary information, distribution list memberships, alternate extensions associated with a mailbox and other enumeration types.

---

CXListNext	Retrieves the successive elements in an enumeration. <b>CXListNext</b> must follow a <b>CXListFirst</b> function call or another <b>CXListNext</b> call.
CXListTotal	Retrieves a count of the number of items in a list.

---

# MiCollab AM Web Services API Functions

This section lists the MiCollab AM Web Services API functions in order by functional group. A complete description is given for each function call's elements, which act as parameters; return values; XML return elements, including any possible returned error values. Some functions also include usage comments.

Some parameters are optional and are marked as such. The description for the parameter describes any default value when an optional parameter is omitted.

The XML in the SOAP response looks like the following:

```
<XMLResponse xmlns="http://www.mitel.com/CXIf">
  <CXResponse>
    <RESPONSE_ELEMENTS/>
  </CXResponse>
</XMLResponse>
```

Where RESPONSE\_ELEMENTS is the XML response to the SOAP request. Since the **XMLResponse** and **CXResponse** elements are returned in the XML of all SOAP responses, neither is included in the **Output Parameters** and **XML Return** sections.

## CXLogon

Logs on to an administrator account or mailbox.

### Syntax

```
<CXLogon>
  <UserID></UserID>
  <Password></Password>
  <LogonType></LogonType>
</CXLogon>
```

## Input Parameters

UserID

- A mailbox number or administrator account.

Password

- The password associated with the account.

LogonType

- The type of logon access.



- 8 – Log on as an administrator.
- 11 – Log on as a Subscriber.

## Output Parameters

### SessionID

- The session identifier that identifies the logon session in all subsequent requests.

### SessionInstanceID

- A unique identifier that identifies the server instance in all subsequent requests.

## XML Return

```
<SessionID></SessionID>
<SessionInstanceID></SessionInstanceID>
```

## Errors

### 2147749809 Invalid credentials

- The UserID or Password value is not valid or the account does not have the privileges required for the Logon Type.

## Comments

A valid logon is required to authenticate the credentials and establish a session with the MiCollab AM SOAP server. All other Web Services API functions must reference a valid SessionID and SessionInstanceID. The use of a valid **SessionID** and **SessionInstanceID** allows the server to apply the appropriate access privileges and also supports functions that maintain session state. A session is terminated after 30 minutes of inactivity.

Each administrator account is configured with a logon limit which is the maximum number of concurrent logons allowed for the administrator account. The default logon limit of the administrator account is 5. When the number of active logons has reached the limit, **CXLogon** returns an error and no more logons are allowed until a logoff has occurred or a session has timed out.

## See Also

[CXLogoff](#)

## CXLogoff

Logs off an administrator account or subscriber mailbox.

## Syntax

```
<CXLogout>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
</CXLogout>
```

## Input Parameters

SessionID

- Identifies the server session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

## Output Parameters

None

## Errors

2147749804     Session is Invalid

- The SessionID provided is invalid, the SessionInstanceID is invalid, the session has been previously logged off, or the server has timed out of the session.

## Comments

When this function is called, the server releases all resources, including any attached lines and open recordings.

## See Also

[CXLogon](#)

## CXMBSearch

Returns the selected attributes of mailboxes that match the specified search criteria. This function may be used to retrieve specific subsets of mailboxes and may also be used to do a *mailbox change* search for determining if any mailboxes have changed since a prior search.

## Syntax

```
<CXMBSearch>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <SearchType></SearchType>
  <SearchCriteria>
    <MBIDRange>
      <RangeFrom></RangeFrom>
      <RangeTo></RangeTo>
    </MBIDRange>
    <MBFilters>
      </MBFilters>
    </SearchCriteria>
  <MBFields>
    <Field></Field>
  </MBFields>
  <Count></Count>
</CXMBSearch>
```

## Input Parameters

### SessionID

- Identifies the session. This value is returned by a previous call to CXLogon.

### SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to CXLogon.

### SearchType

- The type of search.
- *Generic* – Currently the only valid value.

### SearchCriteria

- Each of the following fields are optional, but can be grouped together in different combinations to get the precise search criteria desired. The criteria work together to filter down the matching mailboxes. Thus as more fields are populated, the search will be further qualified and result in a narrower result set.
  - COSMBID
    - If populated, the search will be limited to mailboxes that are part of this Class of Service.
  - DisplayName
    - If populated, the search will be limited to mailboxes with a Display Name that either matches this value exactly or is a partial match that includes this value anywhere in the name.
  - FirstName

- If populated, the search will be limited to mailboxes with a First Name that either matches this value exactly or is a partial match that begins with this value.
- LocalizedGroupName
  - If populated, the search will be limited to mailboxes that are part of this Group.
    - GroupName
      - The name of the Group.
    - CultureName
      - The culture corresponding to the **GroupName** e.g. "en-US".
- LocalizedGroupTypeName
  - If populated, the search will be limited to mailboxes that are part of a Group that is part of this Group Type.
    - GroupTypeName
      - The name of the Group Type.
    - CultureName
      - The culture corresponding to the **GroupTypeName** e.g. "en-US".
- LastChangeNumber
  - If populated, enables a *mailbox change* search. This will cause the search to be limited to mailboxes that have had a configuration change after that of the given value. Useful numbers to use for LastChangeNumber include 0 (to search for all mailboxes that have changed since database initialization), and values returned in the ChangeNumber MBField.
- LastName
  - If populated, the search will be limited to mailboxes with a Last Name that either matches this value exactly or is a partial match that begins with this value.
- MiddleName
  - If populated, the search will be limited to mailboxes with a Middle Name that either matches this value exactly or is a partial match that begins with this value.
- MBIDRange
  - If populated, contains the mailbox numbers used in a range search.
    - RangeFrom
      - Starting mailbox number used in a range search.
    - RangeTo
      - Ending mailbox number used in a range search.
- MBFilters
  - The filters to be applied to the search. Multiple filters can be specified. Only mailboxes that meet all the filter criteria will be retrieved.
    - DisplayName

- If populated, the search will be limited to mailboxes with a Display Name that matches this value exactly.
- FirstName
  - If populated, the search will be limited to mailboxes with a First Name that matches this value exactly.
- LastName
  - If populated, the search will be limited to mailboxes with a Last Name that matches this value exactly.
- MBTypeArray
  - If populated, this container will include an array of mailbox types to be used as filters for the search. The array can have a single mailbox type or many mailbox types, and the search will match mailboxes that are of any type in the array. If **MBTypeArray** is not populated, the search will not be filtered by mailbox type.
    - MBType
      - Mailbox type. See MBType in the MBCommon section for valid mailbox types.
- MiddleName
  - If populated, the search will be limited to mailboxes with a Middle Name that matches this value exactly.
- SponsorMBID
  - If populated, the search will be limited to mailboxes that are sponsored by this mailbox number.
- VMDir
  - If populated, the search will be limited based on whether the mailbox is configured to be included in the subscriber directory. The values are 1 for true, 0 for false.

**NOTE** Including 'X' as a mailbox type when doing a *mailbox change* search which will allow the search to match deleted mailboxes (regardless of their original mailbox type).

#### MBFields

- Required field containing the list of mailbox fields to be returned for each of the matching mailboxes.
  - Field
    - Possible values are:
    - *ChangeNumber*
      - For use with *mailbox change* searches. Include this field to return the change number associated with the last change made to the mailbox.

- *COSMBID*
  - Include this field to return the mailbox number of the Class of Service mailbox of which this mailbox is a member.
- *DisplayName*
  - Include this field to return the Display Name of the mailbox.
- *ExternalDirSync*
  - Include this field to return whether the Subscriber mailbox was created through Active Directory. The return value will be 1 for true, 0 for false.
- *IsDeleted*
  - For use with *mailbox change* searches when MBType 'X' (indicating a deleted mailbox) is included in the MBTypeArray. Include this field to return whether the last change to the mailbox was a deletion. The return value will be 1 for true, 0 for false.

**NOTE** If MBIDRange is used, the search will not include deleted mailboxes, so IsDeleted will always have a value of 0.

- *MBID*
  - Include this field to return the mailbox number of the mailbox.
- *MBType*
  - Include this field to return the mailbox type. See MBType in the MBCommon section for valid mailbox types.

**NOTE** When doing a *mailbox change* search, a mailbox type of 'X' indicates a mailbox that has been deleted.

- *PrimaryExt*
  - Include this field to return the primary extension of the Subscriber mailbox.
- *PrimaryGroupDisplayName*
  - Include this field to return the primary Group of the Subscriber mailbox.
- *Propagate*
  - Include this field to return whether the Subscriber or Distribution List mailbox is set to propagate to remote correspondent servers. The return value will be 1 for true, 0 for false.
- *SetupRequired*
  - Include this field to return whether the Subscriber mailbox requires the setup tutorial. The return value will be 1 for true, 0 for false.
- *WasPropagated*
  - Include this field to return whether the Subscriber or Distribution List mailbox was propagated from a remote correspondent server. The return value will be 1 for true, 0 for false.

- *ClassOfServiceType*
  - Include this field to return whether the Class of Service Mailbox is *Standard* or *Availability*. The return value is *Standard* for a Mailbox COS. It is *Availability* for an Availability COS.

#### Count

- Optional. If provided, Count specifies the maximum number of mailboxes to retrieve. Also, if there are more matching mailboxes than Count, **MoreRecordsExist** will be returned with a value of 1. If Count is not provided, all matching mailboxes will be retrieved.

## Output Parameters

#### MBFieldsList

- The top-level container for the list of mailbox fields for each mailbox found in the search.
  - MBFields
    - Contains the fields that are retrieved for a particular mailbox. Only the fields specified in the **MBFields** input parameter are retrieved.

#### MoreRecordsExist

- If Count is provided, and there are more matching mailboxes than Count, **MoreRecordsExist** will be returned with a value of 1.

## XML Return

```
<MBFieldsList>
  <MBFields>
    <MBID>1007</MBID>
    <MBType>A</MBType>
    <DisplayName>Thomas Bulgar</DisplayName>
    <PrimaryExt>1007</PrimaryExt>
  </MBFields>
</MBFieldsList>
```

## Errors

#### 2147749305 Invalid argument

- An invalid argument has been provided.

#### 2147749804 Session is Invalid

- The **SessionID** provided is invalid, the session has been previously logged off, or the server has timed out of the session.

#### 2147750608 Data-layer error: Missing field

- A required element was not provided.

## Comments

Using **LastChangeNumber** changes the behavior of the search to be a *mailbox change* search. A key difference is that it gives meaning to the **ChangeNumber**. A use of **ChangeNumber** is to use the highest value from a mailbox change search as **LastChangeNumber** in a subsequent mailbox change search. Another key difference is that it also allows references to deleted mailboxes to be returned (noted by the **IsDeleted** value having a value of 1). For this to occur, the **MBTypeArray** needs to include the mailbox type 'X', which is a special value indicating deleted mailboxes. For deleted mailboxes, the only MBFields that will be populated are MBID, ChangeNumber, and IsDeleted (with 1).

**NOTE** Although **LastChangeNumber** can be combined with **MBIDRange** to further narrow the scope of the search, that combination will excluded deleted mailboxes, regardless of whether 'X' is included in the **MBTypeArray**.

## Example

The following example searches for all Subscriber and Local Alias mailboxes with mailbox numbers in the range of 0000-9999, returning up to 100 mailboxes. The output will include the mailbox number, mailbox type, display name, and primary group, and be ordered by ascending mailbox number.

```
<CXMBSearch>
  <SessionID>1972306838</SessionID>
  <SessionInstanceID>3d804697-74f6-4826-b8fa-93e98a328a62</SessionInstanceID>
  <SearchType>Generic</SearchType>
  <SearchCriteria>
    <MBIDRange>
      <RangeFrom>0000</RangeFrom>
      <RangeTo>9999</RangeTo>
    </MBIDRange>
    <MBFilters>
      <MBTypeArray>
        <MBType>A</MBType>
        <MBType>F</MBType>
      </MBTypeArray>
    </MBFilters>
  </SearchCriteria>
  <MBFields>
    <Field>MBID</Field>
    <Field>MBType</Field>
    <Field>DisplayName</Field>
    <Field>PrimaryGroupDisplayName</Field>
  </MBFields>
  <Count>100</Count>
</CXMBSearch>
```

The following example searches for all Subscriber mailboxes with a display name that includes the substring *Bo*. The output will include the mailbox number, display name, and primary extension, and be ordered by ascending mailbox number.

```
<CXMBSearch>
```



```

<SessionID>1972306838</SessionID>
<SessionInstanceID>3d804697-74f6-4826-b8fa-93e98a328a62</SessionInstanceID>
<SearchType>Generic</SearchType>
<SearchCriteria>
  <DisplayName>Bo</DisplayName>
  <MBFilters>
    <MBTypeArray>
      <MBType>A</MBType>
    </MBTypeArray>
  </MBFilters>
</SearchCriteria>
<MBFields>
  <Field>MBID</Field>
  <Field>DisplayName</Field>
  <Field>PrimaryExt</Field>
</MBFields>
</CXMBSearch>

```

The following example searches for all Subscriber mailboxes that have had configuration changes since database initialization. The output will include the mailbox number, display name, including the change number, and be ordered by ascending change number.

```

<CXMBSearch>
  <SessionID>1972306838</SessionID>
  <SessionInstanceID>3d804697-74f6-4826-b8fa-93e98a328a62</SessionInstanceID>
  <SearchType>Generic</SearchType>
  <SearchCriteria>
    <LastChangeNumber>0</LastChangeNumber>
    <MBFilters>
      <MBTypeArray>
        <MBType>A</MBType>
      </MBTypeArray>
    </MBFilters>
  </SearchCriteria>
  <MBFields>
    <Field>MBID</Field>
    <Field>DisplayName</Field>
    <Field>ChangeNumber</Field>
  </MBFields>
</CXMBSearch>

```

## See Also

[CXMBSearchCountGet](#), [Mailbox Configuration Functions](#)

## CXMBSearchCountGet

Returns a count of the number of mailboxes that match the specified search criteria. The format of the search criteria is the same as that of **CXMBSearch**, and the intended use of the function is to first get the

total count of mailboxes that match. An example would be for displaying a progress bar during the search. This strategy can also help optimize a large search by splitting it up into sub searches.

## Syntax

```
<CXMBSearchCountGet>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <SearchType></SearchType>
  <SearchCriteria>
    <MBIDRange>
      <RangeFrom></RangeFrom>
      <RangeTo></RangeTo>
    </MBIDRange>
    <MBFilters>
      </MBFilters>
    </SearchCriteria>
  </CXMBSearchCountGet>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

SearchType

- The type of search.
- *Generic* – Currently the only valid value.

SearchCriteria

Refer to **SearchCriteria** in [CXLogon](#)

CXMBSearch.

## Output Parameters

Count

- The total number of mailboxes that match the search criteria.

## XML Return

```
<Count>153</Count>
```

## Errors

2147749305 Invalid argument

- An invalid argument has been provided.

2147749804 Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147750608 Data-layer error: Missing field

- A required element was not provided.

## Comments

**CXMBSearchCountGet** indicates the number of mailboxes that **CXMBSearch** would match given the same **SearchCriteria** values. See Comments in [CXLogon](#)

CXMBSearch for more information.

## Example

The following example gets the count of all Subscriber and Local Alias mailboxes.

```
<CXMBSearchCountGet>
  <SessionID>1972306838</SessionID>
  <SessionInstanceID>3d804697-74f6-4826-b8fa-93e98a328a62</SessionInstanceID>
  <SearchType>Generic</SearchType>
  <SearchCriteria>
    <MBFilters>
      <MBTypeArray>
        <MBType>A</MBType>
        <MBType>F</MBType>
      </MBTypeArray>
    </MBFilters>
  </SearchCriteria>
</CXMBSearchCountGet>
```

The following example searches for all Subscriber mailboxes with a mailbox number in the range of 1000-1500 and has a last name of Jones.

```
<CXMBSearchCountGet>
  <SessionID>1972306838</SessionID>
  <SessionInstanceID>3d804697-74f6-4826-b8fa-93e98a328a62</SessionInstanceID>
  <SearchType>Generic</SearchType>
  <SearchCriteria>
    <MBIDRange>
      <RangeFrom>1000</RangeFrom>
      <RangeTo>1500</RangeTo>
    </MBIDRange>
    <MBFilters>
```

```

        <MBTypeArray>
            <MBType>A</MBType>
        </MBTypeArray>
        <LastName>Jones</LastName>
    </MBFilters>
</SearchCriteria>
</CXMBSearchCountGet>

```

The following example gets the count of all mailboxes that once existed but have since been deleted.

```

<CXMBSearchCountGet>
    <SessionID>1972306838</SessionID>
    <SessionInstanceID>3d804697-74f6-4826-b8fa-93e98a328a62</SessionInstanceID>
    <SearchType>Generic</SearchType>
    <SearchCriteria>
        <LastChangeNumber>0</LastChangeNumber>
        <MBFilters>
            <MBTypeArray>
                <MBType>X</MBType>
            </MBTypeArray>
        </MBFilters>
    </SearchCriteria>
</CXMBSearchCountGet>

```

## See Also

[CXLogin](#)

CXMBSearch, [Mailbox Configuration Functions](#)

## CXMBGet

Loads the basic mailbox structure for the specified mailbox.

## Syntax

```

<CXMBGet>
    <SessionID></SessionID>
    <SessionInstanceID></SessionInstanceID>
    <MBID></MBID>
</CXMBGet>

```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogin**.

SessionInstancelD

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

MBID

- The mailbox number of the mailbox.

## Output Parameters

MB

- The basic mailbox structure.

MBCCommon

- Contains the fields that are common to most mailbox types. See the MBCCommon section.

MBSpecific

- Contains the fields that are specific to the mailbox types, including:
  - MSubSpecific
    - Data specific to Subscriber mailboxes. See the MSubSpecific section.
  - MBDistListSpecific
    - Data specific to Distribution List mailboxes. See the MBDistListSpecific section.
  - MBCPSpecific
    - Data specific to Call Processor mailboxes. See the MBCPSpecific section.

## Errors

2147749804    Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147750608    Data-layer error: Missing field

- A required element was not provided.

2147750611    Data-layer error: Insufficient privileges

- The logged on user does not have sufficient privileges for accessing this mailbox.

2147750612    Data-layer error: Mailbox not found

- MBID does not exist on the local server.

## See Also

[CXMBAdd](#), [CXMBUpdate](#), [Mailbox Configuration Functions](#)

# CXMBAdd

Adds a new mailbox, with just the basic mailbox structure.

## Syntax

```
<CXMBAdd>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <TemplateMBID></TemplateMBID>
  <MB>
    <MBCommon>
      <MBID></MBID>
      <MBType></MBType>
    </MBCommon>
    <MBTypeSpecific>
      </MBTypeSpecific>
    </MB>
  </CXMBAdd>
```

## Input Parameters

### SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

### SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

### TemplateMBID

- This is an optional parameter. If set to a COS MBID, the default values for the subscriber mailbox being added will be derived from this template COS mailbox. Note that derivation of the field values from the COS mailbox specified by TemplateMBID only happens for unspecified fields.

### MB

- The basic mailbox structure.
- MBCommon
  - Contains the fields that are common to most mailbox types. See the MBCommon section.
  - MBID
    - Required. The mailbox number.
  - MBType
    - Required. The type of mailbox.
- MBTypeSpecific
  - Contains the fields that are specific to the mailbox types, including:

- MBSubSpecific
  - Data specific to Subscriber mailboxes. See the MBSubSpecific section.
- MBDistListSpecific
  - Data specific to Distribution List mailboxes. See the MBDistListSpecific section.
- MBCPSpecific
  - Data specific to Call Processor mailboxes. See the MBCPSpecific section.

## Output Parameters

None

## Errors

2147749804 Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147750608 Data-layer error: Missing field

- A required element was not provided.

2147750611 Data-layer error: Insufficient privileges

- The logged on user does not have sufficient privileges for accessing this mailbox.

2147750613 Data-layer error: Mailbox already exists

- MBID already exists on the local server.

2147750615 Data-layer error: The template mailbox was not found

- The mailbox specified by TemplateMBID does not exist.

2147750616 Data-layer error: The template mailbox is of the wrong type

- The mailbox specified by TemplateMBID is of the wrong type.

## See Also

[CXMBGet](#), [CXMBUpdate](#), [Mailbox Configuration Functions](#)

## CXMBUpdate

Updates the basic mailbox structure for the specified mailbox.

## Syntax

```
<CXMBUpdate>
```

```

<SessionID></SessionID>
<SessionInstanceID></SessionInstanceID>
<MB>
  <MBCommon>
    <MBID></MBID>
    <MBType></MBType>
  </MBCommon>
  <MBTypeSpecific>
  </MBTypeSpecific>
</MB>
<OldObject>
  <MB>
  </MB>
</OldObject>
</CXMBUpdate>

```

## Input Parameters

### SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

### SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to CXLogon.

### MB

- The basic mailbox structure.
- MBCommon
  - Contains the fields that are common to most mailbox types. See the MBCommon section.

**NOTE** MBID and MBType cannot be changed through CXMBUpdate.

- MBID
  - Required. The mailbox number.
- MBType
  - Required. The type of mailbox.

### MBTypeSpecific

- Contains the fields that are specific to the mailbox types, including:
- MBSpecific
  - Data specific to Subscriber mailboxes. See the MBSpecific section.
- MBDistListSpecific
  - Data specific to Distribution List mailboxes. See the MBDistListSpecific section.
- MBCPSpecific



- Data specific to Call Processor mailboxes. See the MBCPSpecific section.

#### OldObject

- This optional field represents the baseline from which the update is being made. If populated, the difference between this MB and the update MB determines the updates that are applied.
- MB
  - The basic mailbox structure. This has the same format as MB above.
  - MBCommon
    - See description above.
  - MBTypeSpecific
    - See description above.
    - MBDistListSpecific
      - See description above.
    - MBSubSpecific
      - See description above.

## Output Parameters

None

## Errors

2147749804 Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147750608 Data-layer error: Missing field

- A required element was not provided.

2147750611 Data-layer error: Insufficient privileges

- The logged on user does not have sufficient privileges for accessing this mailbox.

2147750612 Data-layer error: Mailbox not found

- MBID does not exist on the local server.

## Comments

If **OldObject/MB** is not populated, the update is based completely on the existing mailbox. If **OldObject/MB** is populated, the update is based on the difference between **MB** and **OldObject/MB**. This is important when allowing multiple clients to edit the same mailbox such that changes from one client don't inadvertently overwrite changes by the other client.

## Example

The following example demonstrates updating the display name of the subscriber.

```
<CXMBUpdate>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
  <MB>
    <MBCommon>
      <MBID>2001</MBID>
      <MBType>A</MBType>
      <DisplayName>New Display Name</DisplayName>
    </MBCommon>
  </MB>
</CXMBUpdate>
```

## See Also

[CXMBAdd](#), [CXMBGet](#), [CXMBTemplateUpdate](#), [Mailbox Configuration Functions](#)

## CXMBTemplateUpdate

Updates the basic mailbox record for the specified mailboxes using a template mailbox (COS mailbox).

## Syntax

```
<CXMBTemplateUpdate>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <TemplateMBID></TemplateMBID>
  <TemplateMB>
    <MBCommon>
      <MBID></MBID>
      <MBType></MBType>
    </MBCommon>
    <MBTypeSpecific>
    </MBTypeSpecific>
  </TemplateMB>
  <OldTemplateMB>
    <MBCommon>
      <MBID></MBID>
      <MBType></MBType>
    </MBCommon>
    <MBTypeSpecific>
    </MBTypeSpecific>
  </OldTemplateMB>
  <MBIDList>
    <MBID></MBID>
  </MBIDList>
```

```
</CXMBTemplateUpdate>
```

## Input Parameters

### SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

### SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

### TemplateMBID

- The MBID of the template mailbox to be used as the source mailbox for updating the specified mailboxes. Specify either the **TemplateMBID** or the **TemplateMB** parameter.

### TemplateMB

- The template mailbox structure to be used as the source mailbox for updating the specified mailboxes. Specify either the **TemplateMBID** or the **TemplateMB** parameter.
- MBCommon
  - Contains the fields that are common to most mailbox types. See the MBCommon section.
- MBType
  - Required. The type of the template mailbox. This should be set to S for a Class of Service mailbox.
  - MBTypeSpecific
    - Contains the fields that are specific to the mailbox type:
  - MBSubSpecific
    - Data specific to Subscriber and Class of Service mailboxes. See the MBSubSpecific section.

### OldTemplateMB

- This optional field represents the baseline for the template mailbox from which the update is being made. If populated, the difference between this structure and the TemplateMB structure determines the updates that are applied to the specified mailboxes.
- MBCommon
  - See description above.
  - MBTypeSpecific
    - See description above.

### MBIDList

- This is a container of one or more MBID fields. Each MBID field represents one mailbox that needs to be updated.
  - MBID

- MBID of the mailbox to be updated.

## Output Parameters

### MBErrorList

The top-level container for the per-mailbox error values. This contains one or more **MBError** fields. Each **MBError** field holds the error value for exactly one mailbox. This list of **MBError** fields is returned in the same order as the **MBIDList** input parameter.

If a **MBError** field's value is 0 then that means that the mailbox was successfully updated. If it is greater than zero then it indicates that some kind of error happened while updating this mailbox. Here is the table of the error values that can be returned via this field:

Table 8. Returnable error values

Error Value	Error String	Description
2147750612	Data-layer error: Mailbox not found	The mailbox being updated does not exist on the local server.
2147750616	Data-layer error: The template mailbox is of the wrong type	The mailbox specified by TemplateMBID is of the wrong type for updating this mailbox.

## Errors

2147749804 Session is Invalid

- The **SessionID** provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147750608 Data-layer error: Missing field

- A required element was not provided.

2147750611 Data-layer error: Insufficient privileges

- The logged on user does not have sufficient privileges for accessing this mailbox.

2147750615 Data-layer error: The template mailbox was not found

- The mailbox specified by TemplateMBID does not exist.

## Comments

Note that only the inheritable fields in the template mailbox are updated. Non-inheritable fields (such as Name fields, EmailID, Email Server Type etc.) are never updated.

If **OldTemplateMB** is not populated, all the inheritable fields of **TemplateMB** are used to update the subscriber mailbox. If the **OldTemplateMB** is populated, the update is based on the difference between

**TemplateMB** and **OldTemplateMB**. This is important when updating the mailboxes with only the fields that changed in the COS mailbox.

## Example

The following example demonstrates updating all inheritable fields of the subscriber mailbox 5710 with values from the COS mailbox 7000.

```
<CXMBTemplateUpdate>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
  <TemplateMBID>7000</TemplateMBID>
  <MBIDList>
    <MBID>5710</MBID>
  </MBIDList>
</CXMBTemplateUpdate>
```

## See Also

[CXMBAdd](#), [CXMBGet](#), [CXMBUpdate](#), [Mailbox Configuration Functions](#)

## CXMBSubIMNCalloutArrayGet

Loads the ordered array of immediate message notification callout entries for the specified Subscriber mailbox.

## Syntax

```
<CXMBSubIMNCalloutArrayGet>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <MBID></MBID>
</CXMBSubIMNCalloutArrayGet>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

MBID

- The mailbox number of the Subscriber mailbox.

## Output Parameters

MBSUBIMNCalloutArray

- The container for the ordered array of immediate message notification callout entries.
- IMNCallout
  - The notification callout information.
  - CallType
    - The type of phone device.
      - *Normal* – Regular voice line
      - *Radio* – Radio or voice announce pager
      - *Digital* – Digital pager
      - *PIN* – PIN type pager
    - WaitMinutes
      - The number of minutes, between 1 and 99, that the server should wait before trying the next callout entry.
    - Phone
      - The telephone or pager number.
    - Pin
      - The personal identification number if the device is a PIN pager.

## Errors

2147749804      Session is Invalid

- The **SessionID** provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147750608      Data-layer error: Missing field

- A required element was not provided.

2147750611      Data-layer error: Insufficient privileges

- The logged on user does not have sufficient privileges for accessing this mailbox.

## See Also

[CXMBSubIMNCalloutArrayUpdate](#), [Mailbox Configuration Functions](#)

# CXMBSubIMNCalloutArrayUpdate

Updates the ordered array of immediate message notification callout entries for the specified Subscriber mailbox.

## Syntax

```
<CXMBSubIMNCalloutArrayUpdate>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <MBID>
  </MBID>
  <MBSUBIMNCalloutArray>
    <IMNCallout>
      <CallType></CallType>
      <WaitMinutes></WaitMinutes>
      <Phone></Phone>
      <Pin></Pin>
    </IMNCallout>
  </MBSUBIMNCalloutArray>
</CXMBSubIMNCalloutArrayUpdate>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

MBID

- The mailbox number of the Subscriber mailbox.

MBSUBIMNCalloutArray

- The container for the updated ordered array of immediate message notification callout entries.
- IMNCallout
  - The notification callout information.
  - CallType
    - The type of telephone number.
      - *Normal* – Regular telephone number
      - *Radio* – Radio and voice announce pager
      - *Digital* – Digital pager
      - *PIN* – PIN-type pager

- WaitMinutes
  - The number of minutes, between 1 and 99, that the server should wait before trying the next callout entry.
- Phone
  - The telephone number.
- Pin
  - The personal identification number if the subscriber uses a PIN pager.

## Output Parameters

None

## Errors

2147749804    Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147750608    Data-layer error: Missing field

- A required element was not provided.

2147750611    Data-layer error: Insufficient privileges

- The logged on user does not have sufficient privileges for accessing this mailbox.

2147750612    Data-layer error: Mailbox not found

- MBID does not exist on the local server.

2147750703    Data-layer error: Phone number failed dialplan

- Phone either failed dialplan or the Subscriber is restricted from calling this type of telephone number.

2147750716

- Data-layer error: Pin number contains invalid characters
- Pin contains invalid characters.

2147750718

- Data-layer error: Invalid personal call list wait time
- WaitMinutes is outside the valid range.

2147750742

- Data-layer error: Notification phone number is missing
- Phone is missing or blank.



## Comments

The order of the **IMNCallout** entries in **MBSubIMNCalloutArray** XML represent the call order as they will exist for the Subscriber.

## Example

The following example demonstrates replacing the existing array.

```
<CXMBSubIMNCalloutArrayUpdate>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
  <MBID>2001</MBID>
  <MBSubIMNCalloutArray>
    <IMNCallout>
      <CallType>Normal</CallType>
      <WaitMinutes>15</WaitMinutes>
      <Phone>3001</Phone>
      <Pin></Pin>
    </IMNCallout>
    <IMNCallout>
      <CallType>PIN</CallType>
      <WaitMinutes>15</WaitMinutes>
      <Phone>5551212</Phone>
      <Pin>1234</Pin>
    </IMNCallout>
  </MBSubIMNCalloutArray>
</CXMBSubIMNCalloutArrayUpdate>
```

## See Also

[CXMBSubIMNCalloutArrayGet](#), [Mailbox Configuration Functions](#)

## CXMBSubEmailGet

Loads the email configuration information for the specified Subscriber mailbox.

## Syntax

```
<CXMBSubEmailGet>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <MBID></MBID>
</CXMBSubEmailGet>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

MBID

- The mailbox number of the Subscriber mailbox.

## Output Parameters

MBSubEmail

- The email configuration information.
- MBID
  - The mailbox number of the Subscriber mailbox.
- DisplayName
  - The display name for this email account.
- Address
  - The subscriber's email address.
- ServerID
  - The unique identifier for the email server profile. Use the CXListFirst function with a ListType of *EmailServer* to enumerate the configured email profiles.
- LogonID
  - The user name or other ID used to log on to the subscriber's email account.

## Errors

2147749804     Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147750611     Data-layer error: Insufficient privileges

- The logged on user does not have sufficient privileges for accessing this mailbox.

2147750612     Data-layer error: Mailbox not found

- MBID does not exist on the local server.

2147750767     Data-layer error: Mailbox Email Config Entry Not Found

- The server does not have an email config entry for this mailbox.

## See Also

[CXMBSubEmailUpdate](#), [Mailbox Configuration Functions](#)

# CXMBSubEmailUpdate

Updates the email configuration information for the specified Subscriber mailbox.

## Syntax

```
<CXMBSubEmailUpdate>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <MBSubEmail>
    <MBID> </MBID>
    <DisplayName> </DisplayName>
    <Address> </Address>
    <ServerID> </ServerID>
    <LogonID> </LogonID>
  </MBSubEmail> </CXMBSubEmailUpdate>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to CXLogon.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to CXLogon.

MBSubEmail

- The email configuration information. Please refer to [CXMBSubEmailGet](#) for the details of this structure.

## Output Parameters

None

## Errors

2147749804      Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147750608      Data-layer error: Missing field

- A required element was not provided.

2147750609 Data-layer error: Missing object parameter

- The **MBSubEmail** parameter was not provided.

2147750611 Data-layer error: Insufficient privileges

- The logged on user does not have sufficient privileges for accessing this mailbox.

2147750612 Data-layer error: Mailbox not found

- MBID does not exist on the local server.

2147750772 Data-layer error: Email Server Config Profile Not Found

- The email server profile for the ServerID was not found.

## Comments

In order to remove email configuration for a mailbox, please set the ServerID field to the empty value.

## Example

The following example demonstrates updating the existing email configuration for mailbox 5702.

```
<CXMBSubEmailUpdate>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
  <MBSubEmail>
    <MBID>5702</MBID>
    <DisplayName>John Doe</DisplayName>
    <Address>johndoe@mycompany.com</Address>
    <ServerID>{CB28D17C-CB96-4566-89BA-4D7CC68A575F}</ServerID>
    <LogonID></LogonID>
  </MBSubEmail>
</CXMBSubEmailUpdate>
```

## See Also

[CXMBSubEmailGet](#), [Mailbox Configuration Functions](#)

## CXMBDistListMemberArrayGet

Loads the array of members of the specified Distribution List mailbox.

## Syntax

```
<CXMBDistListMemberArrayGet>
  <SessionID></SessionID>
```

```
<SessionInstanceID></SessionInstanceID>
<MBID></MBID>
</CXMBDistListMemberArrayGet>
```

## Input Parameters

### SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

### SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

### MBID

- The mailbox number of the Distribution List mailbox.

### MBDistListMemberArray

- The container for the updated array of members of the Distribution List mailbox.
- Member
  - The member information. The member can be either a Subscriber mailbox or a Distribution List mailbox, and the member must either exist on the local server, or represent an existing mailbox on a remote server.
  - MemberMBID
    - If the member is a local mailbox, **MemberMBID** contains the mailbox number. If the member represents a remote mailbox, MemberMBID contains the mailbox number of the Digital Networking mailbox that represents the remote server on which the remote mailbox exists.
  - RemoteMBNo
    - If the member represents a remote mailbox, **RemoteMBNo** contains the mailbox number of the remote mailbox, otherwise the field is unpopulated.

### OldObject

- This optional field represents the baseline list of members from which the update is being made. If populated, the difference between this array and the update array determines the updates that are applied. See Comments and the examples for further details.
- MBDistListMemberArray
  - The container for the array of members on which the update is based. This has the same format as MBDistListMemberArray above.
  - Member
    - See description above.
    - MemberMBID
      - See description above.

- RemoteMBNo
  - See description above.

## Output Parameters

None

## Errors

2147749804 Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147750608 Data-layer error: Missing field

- A required element was not provided.

2147750611 Data-layer error: Insufficient privileges

- The logged on user does not have sufficient privileges for accessing this mailbox.

2147750612 Data-layer error: Mailbox not found

- Either MBID or one or more instances of MemberMBID does not exist on the local server.

2147750614 Data-layer error: Invalid mailbox Data

- The member cannot be added to the Distribution List mailbox because it violates one of the membership rules. There are a variety of if the Distribution List mailbox is set to propagate or was propagated from a remote server.

## Comments

If **OldObject/MBDistListMemberArray** is not populated, **MBDistListMemberArray** completely replaces the existing array. If **OldObject/MBDistListMemberArray** is populated, the update is based on the difference between **MBDistListMemberArray** and **OldObject/MBDistListMemberArray**. This can be leveraged to add or remove members without specifying the entire array. See the examples below.

## Example

The following example demonstrates completely replacing the existing array of members.

```
<CXMBDistListMemberArrayUpdate>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
  <MBID>2001</MBID>
  <MBDistListMemberArray>
    <Member>
      <MemberMBID>1001</MemberMBID>
      <RemoteMBNo></RemoteMBNo>
    </Member>
  </MBDistListMemberArray>
</CXMBDistListMemberArrayUpdate>
```

```

    <Member>
      <MemberMBID>1002</MemberMBID>
      <RemoteMBNo></RemoteMBNo>
    </Member>
    <Member>
      <MemberMBID>1003</MemberMBID>
      <RemoteMBNo></RemoteMBNo>
    </Member>
  </MBDistListMemberArray>
</CXMBDistListMemberArrayUpdate>

```

The following example demonstrates adding a member to the existing array of members.

```

<CXMBDistListMemberArrayUpdate>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
  <MBID>2001</MBID>
  <MBDistListMemberArray>
    <Member>
      <MemberMBID>1004</MemberMBID>
      <RemoteMBNo></RemoteMBNo>
    </Member>
  </MBDistListMemberArray>
  <OldObject>
    <MBDistListMemberArray>
      </MBDistListMemberArray>
    </OldObject>
  </CXMBDistListMemberArrayUpdate>

```

The following example demonstrates removing a member from the existing array of members.

```

<CXMBDistListMemberArrayUpdate>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
  <MBID>2001</MBID>
  <MBDistListMemberArray>
    </MBDistListMemberArray>
  <OldObject>
    <MBDistListMemberArray>
      <Member>
        <MemberMBID>1001</MemberMBID>
        <RemoteMBNo></RemoteMBNo>
      </Member>
    </MBDistListMemberArray>
  </OldObject>
</CXMBDistListMemberArrayUpdate>

```

## See Also

[CXMBDistListMemberArrayGet](#), [Mailbox Configuration Functions](#)

# CXMBDistListAllowedMBIDArrayGet

Loads the array of mailbox numbers which are allowed to address the specified Distribution List mailbox.

## Syntax

```
<CXMBDistListAllowedMBIDArrayGet>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <MBID></MBID>
</CXMBDistListAllowedMBIDArrayGet>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

MBID

- The mailbox number of the Distribution List mailbox.

## Output Parameters

MBDistListAllowedMBIDArray

- The container for the array of mailbox numbers that are allowed to address the Distribution List mailbox.
  - AllowedMBID
    - A mailbox number that is allowed to address the Distribution List mailbox.

## Errors

2147749804    Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147750608    Data-layer error: Missing field

- A required element was not provided.

2147750611    Data-layer error: Insufficient privileges

- The logged on user does not have sufficient privileges for accessing this mailbox.



## See Also

[CXMBDistListAllowedMBIDArrayUpdate](#), [Mailbox Configuration Functions](#)

# CXMBDistListAllowedMBIDArrayUpdate

Updates the array of mailbox numbers which are allowed to address the specified Distribution List mailbox.

## Syntax

```
<CXMBDistListAllowedMBIDArrayUpdate>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <MBID>
  </MBID>
  <MBDistListAllowedMBIDArray>
    <AllowedMBID></AllowedMBID>
  </MBDistListAllowedMBIDArray>
  <OldObject>
    <MBDistListAllowedMBIDArray>
      <AllowedMBID></AllowedMBID>
    </MBDistListAllowedMBIDArray>
  </OldObject>
</CXMBDistListAllowedMBIDArrayUpdate>
```

## Input Parameters

Updates the array of mailbox numbers which are allowed to address the specified Distribution List mailbox.

## Syntax

```
<CXMBDistListAllowedMBIDArrayUpdate>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <MBID>
  </MBID>
  <MBDistListAllowedMBIDArray>
    <AllowedMBID></AllowedMBID>
  </MBDistListAllowedMBIDArray>
  <OldObject>
    <MBDistListAllowedMBIDArray>
      <AllowedMBID></AllowedMBID>
    </MBDistListAllowedMBIDArray>
  </OldObject>
</CXMBDistListAllowedMBIDArrayUpdate>
```

## Input Parameters

### SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

### SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

### MBID

- The mailbox number of the existing Distribution List mailbox for which to update the list of members.

### MBDistListAllowedMBIDArray

- The container for the updated array of mailbox numbers that are allowed to address the Distribution List mailbox.
- AllowedMBID
  - A mailbox number that is allowed to address the Distribution List mailbox.

### OldObject

- This optional field represents the baseline list of mailbox numbers that are allowed to address the Distribution List mailbox, from which the update is being made. If populated, the difference between this array and the update array determines the updates that are applied. See Comments and the examples for further details.
- MBDistListAllowedMBIDArray
  - The container for the array of members on which the update is based. This has the same format as **MBDistListAllowedMBIDArray** above.
    - AllowedMBID
      - See description above.

## Output Parameters

None

## Errors

2147749804     Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147750608     Data-layer error: Missing field

- A required element was not provided.

2147750611     Data-layer error: Insufficient privileges

- The logged on user does not have sufficient privileges for accessing this mailbox.

2147750612 Data-layer error: Mailbox not found

- MBID does not exist on the local server.

## Comments

If **OldObject/MBDistListAllowedMBIDArray** is not populated, **MBDistListAllowedMBIDArray** completely replaces the existing array. If **OldObject/MBDistListAllowedMBIDArray** is populated, the update is based on the difference between **MBDistListAllowedMBIDArray** and **OldObject/MBDistListAllowedMBIDArray**. This can be leveraged to add or remove entries without specifying the entire array. See the examples below.

## Example

The following example demonstrates completely replacing the existing array.

```
<CXMBDistListAllowedMBIDArrayUpdate>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
  <MBID>2001</MBID>
  <MBDistListAllowedMBIDArray>
    <AllowedMBID>1001</AllowedMBID>
    <AllowedMBID>1002</AllowedMBID>
    <AllowedMBID>1003</AllowedMBID>
  </MBDistListAllowedMBIDArray>
</CXMBDistListAllowedMBIDArrayUpdate>
```

The following example demonstrates adding an entry to the existing array.

```
<CXMBDistListAllowedMBIDArrayUpdate>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
  <MBID>2001</MBID>
  <MBDistListAllowedMBIDArray>
    <AllowedMBID>1004</AllowedMBID>
  </MBDistListAllowedMBIDArray>
  <OldObject>
    <MBDistListAllowedMBIDArray>
      </MBDistListAllowedMBIDArray>
    </OldObject>
</CXMBDistListAllowedMBIDArrayUpdate>
```

The following example demonstrates removing an entry from the existing array.

```
<CXMBDistListAllowedMBIDArrayUpdate>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
  <MBID>2001</MBID>
  <MBDistListAllowedMBIDArray>
  </MBDistListAllowedMBIDArray>
```

```

    <OldObject>
      <MBDistListAllowedMBIDArray>
        <AllowedMBID>1001</AllowedMBID>
      </MBDistListAllowedMBIDArray>
    </OldObject>
  </CXMBDistListAllowedMBIDArrayUpdate>

```

## See Also

[CXMBDistListAllowedMBIDArrayGet](#), [Mailbox Configuration Functions](#)

## CXMBCPActionArrayGet

Loads the array of actions for the specified Call Processor mailbox.

## Syntax

```

<CXMBCPActionArrayGet>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <MBID></MBID>
</CXMBCPActionArrayGet>

```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

MBID

- The mailbox number of the Call Processor mailbox.

## Output Parameters

MBCPActionArray

- The container of one or more Action structures.
- Action
  - The action information.
  - CmdID

- Unique identifier of the command. A command could be a DTMF key or could be an event like Timeout etc. All speech commands (including Directory speech commands) have the value 19. Valid values are:
  - 0 – This is the DTMF 0 Key.
  - 1 – This is the DTMF 1 Key.
  - 2 – This is the DTMF 2 Key.
  - 3 – This is the DTMF 3 Key.
  - 4 – This is the DTMF 4 Key.
  - 5 – This is the DTMF 5 Key.
  - 6 – This is the DTMF 6 Key.
  - 7 – This is the DTMF 7 Key.
  - 8 – This is the DTMF 8 Key.
  - 9 – This is the DTMF 9 Key.
  - 10 – This is the DTMF A Key.
  - 11 – This is the DTMF B Key.
  - 12 – This is the DTMF C Key.
  - 13 – This is the DTMF D Key.
  - 14 – This is the DTMF \* Key.
  - 15 – This is the DTMF # Key.
  - 16 – This represents Final Timeout.
  - 17 – This represents Fax.
  - 18 – This represents Final No Match.
  - 19 – This indicates that this Action is either a normal speech command or a Directory speech command without any associated DTMF key.
  - 20 – This represents Final Mismatch.
- SpeechCmdTypeID
  - Unique identifier of the type of the speech command (or Directory speech command) that is associated with this Action. Possible values are:
    - 0 – Not a Speech command. This means that CmdID will be set to something other than 19.
    - 1 – Speech Command
    - 2 – Directory: All
    - 3 – Reserved: Do Not Use
    - 4 – Directory: MBGroup
- SpeechCmdForeignID

- The value of this field depends upon the value of the SpeechCmdTypeID field. Here are the different cases:

Table 9. SpeechCmdForeignID values

SpeechCmdTypeID	SpeechCmdForeignID
0	0
1	Is a valid SpeechCmdID that groups all the speech aliases for this command.
2	0
4	Is a valid MBGroupID which identifies the MBGroup to which this Directory command applies. Call CXMBGroupListGet to lookup all MBGroups.

- ActionType
  - Identifies the action type, which is the building block of a call processing application within MiCollab AM. Valid values are:
    - *AccessMessages* – Allows subscribers to log on to their mailboxes from a telephone.
    - *BlindTransfer* – Performs an unsupervised transfer and releases the call.
    - *ConfirmTransfer* – Puts the caller on hold and initiates a transfer to the requested extension.
    - *Directory* – Provides the ability to access an audio list of subscriber extensions.
    - *Execute* – Allows a caller to run a program or start an application outside of MiCollab AM such as MiCollab AM Scheduler, an enhanced ESP application that provides Time of Day capabilities for subscriber mailboxes.
    - *Fax* – Allows callers to retrieve fax documents from a library of documents.
    - *GoToCallProcessor* – Sends the caller to the Call Processor mailbox specified in the argument field.
    - *Hangup* – Directs the Call Server to disconnect the line when the command is initiated.
    - *Interactive* – Routes the caller to an Interactive mailbox to start an audio questionnaire.
    - *GoToAnswerMode* – Sends the caller to the Call Processor mailbox currently active in the answer mode.
    - *DirectoryKey* – Provides an audio list of Subscriber extensions or general department extensions.
    - *Language* – Allows the caller to select the language of the Call Server prompts, announcements, and personal greetings.

- *MonitorTransfer* – Initiates the transfer and monitors the line for busy and reorder call progress tones.
  - *OpenScript* – Shifts control from the Call Server to a UCConnect script.
  - *PlayAnnouncement* – Routes the caller to an Announcement mailbox.
  - *Record* – Lets the caller record a private message for a subscriber.
  - *SubscriberMessage* – Connects callers directly to a subscriber's personal greeting or ESP greeting.
  - *Transfer* – Initiates a transfer, and then supervises the call progress until it is answered before releasing it.
  - *Undefined* – Plays the prompt, *Invalid entry*.
  - *SubscriberFAX* – Allows outside callers to send fax messages directly to a Subscriber mailbox.
  - *LiveRecord* – Allows subscribers to record telephone conversations into Subscriber mailboxes for later reference.
- Template
  - Specifies the template for the input argument.

## Errors

2147749804    Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147750608    Data-layer error: Missing field

- A required element was not provided.

2147750611    Data-layer error: Insufficient privileges

- The logged on user does not have sufficient privileges for accessing this mailbox.

## See Also

[CXMBCPActionArrayUpdate](#), [Mailbox Configuration Functions](#)

## CXMBCPActionArrayUpdate

Updates the list of actions for the specified Call Processor mailbox.

## Syntax

```
<CXMBCPActionArrayUpdate>
  <SessionID></SessionID>
```

```

<SessionInstanceID></SessionInstanceID>
<MBID>
</MBID>
<MBCPActionArray>
  <Action>
    <CmdID></CmdID>
    <SpeechCmdTypeID></SpeechCmdTypeID>
    <SpeechCmdForeignID></SpeechCmdForeignID>
    <ActionType></ActionType>
    <Template></Template>
  </Action>
</MBCPActionArray>
<OldObject>
  <MBCPActionArray>
    <Action>
      <CmdID></CmdID>
      <SpeechCmdTypeID></SpeechCmdTypeID>
      <SpeechCmdForeignID></SpeechCmdForeignID>
      <ActionType></ActionType>
      <Template></Template>
    </Action>
  </MBCPActionArray>
</OldObject>
</CXMBCPActionArrayUpdate>

```

## Input Parameters

### SessionID

- Identifies the session. This value is returned by a previous call to CXLogon.

### SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

### MBID

- The mailbox number of the Call Processor mailbox.

### MBCPActionArray

- The container for the updated array of actions of the Call Processor mailbox.
  - Action
- The action information. See the description in CXMBCPActionArrayGet for the fields of this structure.

### OldObject

- This optional field represents the baseline list of actions from which the update is being made. If populated, the difference between this array and the update array determines the updates that are applied. See Comments and the examples for further details.
  - MBCPActionArray



- The container for the array of actions on which the update is based. This has the same format as MBCPActionArray above.

## Output Parameters

None

## Errors

2147749804 Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147750608 Data-layer error: Missing field

- A required element was not provided.

2147750611 Data-layer error: Insufficient privileges

- The logged on user does not have sufficient privileges for accessing this mailbox.

2147750612 Data-layer error: Mailbox not found

- MBID does not exist on the local server.

2147750614 Data-layer error: Invalid mailbox Data

- The action cannot be added to the Call Processor mailbox because it violates one of the rules.

## Comments

If **OldObject/MBCPActionArray** is not populated, **MBCPActionArray** completely replaces the existing array. If **OldObject/MBCPActionArray** is populated, the update is based on the difference between **MBCPActionArray** and **OldObject/MBCPActionArray**. This can be leveraged to add or remove actions without specifying the entire array. See the examples below.

## Example

The following example demonstrates completely replacing the existing array of actions.

```
<CXMBCPActionArrayUpdate>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
  <MBID>2001</MBID>
  <MBCPActionArray>
    <Action>
      <CmdID>0</CmdID>
      <SpeechCmdTypeID>1</SpeechCmdTypeID>
      <SpeechCmdForeignID>1</SpeechCmdForeignID>
      <ActionType>AccessMessages</ActionType>
      <Template>1111</Template>
    </Action>
```

```

    <Action>
      <CmdID>1</CmdID>
      <SpeechCmdTypeID>0</SpeechCmdTypeID>
      <SpeechCmdForeignID>0</SpeechCmdForeignID>
      <ActionType>BlindTransfer</ActionType>
      <Template>2222</Template>
    </Action>
  </MBCPActionArray>
</CXMBCPActionArrayUpdate>

```

The following example demonstrates adding an action to the existing array of actions.

```

<CXMBCPActionArrayUpdate>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
  <MBID>2001</MBID>
  <MBCPActionArray>
    <Action>
      <CmdID>2</CmdID>
      <SpeechCmdTypeID>1</SpeechCmdTypeID>
      <SpeechCmdForeignID>1</SpeechCmdForeignID>
      <ActionType>AccessMessages</ActionType>
      <Template>1111</Template>
    </Action>
  </MBCPActionArray>
  <OldObject>
    <MBCPActionArray>
      </MBCPActionArray>
    </ OldObject >
  </CXMBCPActionArrayUpdate>

```

The following example demonstrates removing an action from the existing array of actions.

```

<CXMBCPActionArrayUpdate>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
  <MBID>2001</MBID>
  <MBCPActionArray>
  </MBCPActionArray>
  <OldObject>
    <MBCPActionArray>
      <Action>
        <CmdID>1</CmdID>
        <SpeechCmdTypeID>0</SpeechCmdTypeID>
        <SpeechCmdForeignID>0</SpeechCmdForeignID>
        <ActionType>BlindTransfer</ActionType>
        <Template>2222</Template>
      </Action>
    </MBCPActionArray>
  </OldObject>
</CXMBCPActionArrayUpdate>

```

## See Also

[CXMBCPActionArrayGet](#), [Mailbox Configuration Functions](#)

## CXMailboxDefaults

Retrieves system configuration parameters related to mailboxes. The values returned by this function may be used for presenting configuration options for a mailbox. For example, a list of installed languages may be presented for selection, or a message retention field may be initialized to the default value. Some of the values, such as the values of **MailboxIDLength** and **MinSecurityCodeLength**, may be used to validate input.

## Syntax

```
<CXMailboxDefaults>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
</CXMailboxDefaults>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

## Output Parameters

MailboxDefaults

- The top-level container.
- MailboxIDLength
  - The length of mailbox numbers on the system. All mailbox numbers on the system must be this many digits in length.
- MessageRetentionDays
  - The default number of days that a message should be retained prior to auto deletion. This value may be set by an administrator in the MiCollab AM System Configuration/Environment settings.
- MinSecurityCodeLength
  - The minimum length allowed for a security code. This value may be set by an administrator in the MiCollab AM System Configuration/Messaging settings.

- PromptLanguages
  - The top-level container element for the listing of the prompt sets installed on the system. See TTSLanguages for the common fields.
- TTSLanguages
  - The top-level container element for the listing of the text to speech languages installed on the system.
  - Default
    - The default cardinal value to be used for the language selection when a mailbox is configured to use the default language setting. Applies to both PromptLanguages and TTSLanguages sections.
  - Count
    - The number of languages installed on the server. Applies to both PromptLanguages and TTSLanguages sections.
  - Language
    - A container element for the listing the details of a prompt or TTS language that is installed on the server. Applies to both PromptLanguages and TTSLanguages sections.
  - Index
    - The assigned index by which a language selection is referenced. Applies to both PromptLanguages and TTSLanguages sections.
  - Number
    - The cardinal number that uniquely identifies a prompt set or TTS language. This value is associated with a particular prompt set or TTS language and never changes in contrast to the Index value which varies depending on the order of installed languages. Applies to both PromptLanguages and TTSLanguages sections.
  - Name
    - The text name that uniquely identifies a prompt set or TTS language. Applies to both PromptLanguages and TTSLanguages sections.
- LastDBInit
  - The date and time the server database was most recently initialized.
- AllowLockout
  - Whether a mailbox is locked out after too many unsuccessful password entry attempts. If true (1), the mailbox is locked after too many unsuccessful attempts; if false (0), the mailbox is not locked regardless of the number of unsuccessful attempts.
- AllowTrustedLogon
  - Whether a mailbox may be configured with trusted logons. If true (1), the mailbox is configured with trusted logons; if false (0), mailboxes can not be configured for trusted logons.
- FirstNameFirst

- The order of subscriber first and last names. If true (1), subscriber names are entered first name first. If false (0), subscriber names are entered last name first.

## XML Return

```
<MailboxDefaults>
  <MailboxIDLength></MailboxIDLength>
  <MessageRetentionDays></MessageRetentionDays>
  <MinSecurityCodeLength></MinSecurityCodeLength>
  <PromptLanguages>
    <Default></Default>
    <Count></Count>
    <Language>
      <Index></Index>
      <Number></Number>
      <Name></Name>
    </Language>
  </PromptLanguages>
  <TTSLanguages>
    <Default></Default>
    <Count></Count>
    <Language>
      <Index></Index>
      <Number></Number>
      <Name></Name>
    </Language>
  </TTSLanguages>
  <LastDBInit></LastDBInit>
  <AllowLockout></AllowLockout>
  <AllowTrustedLogon></AllowTrustedLogon>
  <FirstNameFirst></FirstNameFirst>
</MailboxDefaults>
```

## Errors

2147749804    Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

## CXMailboxPasswordChange

Changes a subscriber mailbox security code or email password.

## Syntax

```
<CXMailboxPasswordChange>
  <SessionID></SessionID>
```

```
<SessionInstanceID></SessionInstanceID>
<Mailbox></Mailbox>
<Password></Password>
<PWFormat></PWFormat>
</CXMailboxPasswordChange>
```

## Input Parameters

### SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

### SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

### Mailbox

- The mailbox number of the mailbox

### Password

- The new security code or password. The security code for a mailbox consists of up to 15 digits. Passwords for email accounts are subject to the conditions of that account.

### PWFormat

- Whether Password applies to the email password or the mailbox security code
  - 0 (default) – Password applies to the mailbox security code
  - 1 – Password applies to the email password

## Output Parameters

### None

## Errors

### 2147749804 Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

### 2147746505 Logon failed: Invalid password

- The mailbox does not exist or Password is of the wrong format

### 2147746844 This security code does not meet your organization's security standards

- Password is too short or not complex enough

### 2147746316 Invalid parameter

- Password is too long

## Example

The following example sets the security code for mailbox 2002 to 54321.

```
<CXMailboxPasswordChange>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
  <Mailbox>2002</Mailbox>
  <Password>54321</Password>
  <PWFormat>0</PWFormat>
</CXMailboxPasswordChange>
```

## See Also

[CXLogon](#), [CXLogoff](#), [Mailbox Configuration Functions](#)

## CXPromptRecordingListGet

Retrieves the list of **PromptRecording** structures associated with a particular mailbox. Each **PromptRecording** structure holds the information for all the recording objects with a particular **RecordingTypeID** and belonging to a particular mailbox.

Each recording object applies to one prompt language. The recording object's identifier and the prompt language identifier are members of the LangRecording structure. The **PromptRecording** structure holds an array of all the language specific LangRecording structures.

## Syntax

```
<CXPromptRecordingListGet>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <ID></ID>
  <RecordingTypeID></RecordingTypeID>
  <RecordingTypeSpecificID></RecordingTypeSpecificID>
</CXPromptRecordingListGet>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

ID

- Identifier of the mailbox for which the data is to be retrieved. This parameter need not be specified for a Mailbox Logon because the server assumes the ID to be same as that of the logged-in mailbox.

#### RecordingTypeID

- An optional parameter which specifies the identifier of the **RecordingType** for which the data is to be retrieved. If this parameter is not specified, all Prompt recordings for the specified mailbox will be retrieved.

#### RecordingTypeSpecificID

- An optional parameter used when RecordingTypeID is set to "Availability". This specifies the identifier of the availability greeting that is to be retrieved. If this parameter is not specified, all availability greetings for the specified mailbox will be retrieved when the **RecordingTypeID** is missing or has a value of "Availability".

## Output Parameters

#### PromptRecordingList

- Container of one or more PromptRecording structures.
- PromptRecording
  - Structure holding the recording object information for all prompt languages.
  - ID
    - Identifies the mailbox to which the data within this structure applies.
  - RecordingTypeID
    - Identifies the Recording Type to which the data within this structure applies. Valid values are:
      - *Busy* – This is the busy greeting. Note that for a Call Processor mailbox, this recording type represents an Instruction Announcement.
      - *OOF* – This is the Out-of-office greeting.
      - *"Availability"* – This is the Availability set of greetings
      - *Greeting* – This is the normal greeting.
      - *Introduction* – This is an Intro Announcement for a Call Processor mailbox.
  - RecordingTypeSpecificID
    - Identifies the availability greeting retrieved for a **RecordingTypeID** value of "Availability". This is 0 for all other **RecordingTypeID** values.
  - LangRecordingArray
    - Container of one or more LangRecording structures.
    - LangRecording
      - Structure holding the Recording object identifier for a particular Prompt Language.
    - PromptLangID



- Identifies the prompt language to which the recording object applies.
- **MediaFileID**
  - Identifier (ObjectID) of the recording object for the specified mailbox, Recording Type and Prompt Language combination.

## XML Return

```
<PromptRecordingList>
  <PromptRecording>
    <ID>57001</ID>
    <RecordingTypeID>Greeting</RecordingTypeID>
    <RecordingTypeSpecificID>0</RecordingTypeSpecificID>
    <LangRecordingArray>
      <LangRecording>
        <PromptLangID>0</PromptLangID>
        <MediaFileID>f15612b7-327b-4edf-af30-cd8a2fc4a068</MediaFileID>
      </LangRecording>
    </LangRecordingArray>
  </PromptRecording>
</PromptRecordingList>
```

## Errors

2147749804    Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session

## See Also

[CXPromptRecordingUpdate](#), [Mailbox Configuration Functions](#)

## CXPromptRecordingUpdate

Updates the PromptRecording structure.

## Syntax

```
<CXPromptRecordingUpdate>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <PromptRecording>
    <ID></ID>
    <RecordingTypeID></RecordingTypeID>
    <RecordingTypeSpecificID></RecordingTypeSpecificID>
    <LangRecordingArray>
```

```

        <LangRecording>
            <PromptLangID></PromptLangID>
            <MediaFileID></MediaFileID>
        </LangRecording>
    </LangRecordingArray>
</PromptRecording>
<OldObject/>
</CXPromptRecordingUpdate>

```

## Input Parameters

### SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

### SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

### PromptRecording

- Structure holding the recording object information for all prompt languages. Please refer to [CXPromptRecordingListGet](#) for the details of this structure.

### OldObject

- This optional field contains the baseline **PromptRecording** structure. If populated, the difference between this embedded structure and the top-level **PromptRecording** parameter will be used to make the necessary updates.
  - PromptRecording
    - The baseline **PromptRecording** structure. The format of this structure is same as that of the top-level **PromptRecording** parameter.

## Output Parameters

None

## Errors

### 2147749804 Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

### 2147750611 Data-layer error: Insufficient privileges

- The logged on user does not have sufficient privileges for updating this data.

## Comments

If **OldObject/PromptRecording** is not populated, **PromptRecording** parameter completely replaces the existing data. If **OldObject/PromptRecording** is populated, the update is based on the difference between **PromptRecording** and **OldObject/PromptRecording**. This can be leveraged to add or remove entries within the **LangRecordingArray** without specifying the entire array. See the examples below.

## Example

The following example demonstrates completely replacing the existing data.

```
<CXPromptRecordingUpdate>
  <SessionID>229574173</SessionID>
  <SessionInstanceID>2a6ec288-eda5-4327-8db0-638aec22dca2</SessionInstanceID>
  <PromptRecording>
    <ID>57001</ID>
    <RecordingTypeID>Greeting</RecordingTypeID>
    <RecordingTypeSpecificID>0</RecordingTypeSpecificID>
    <LangRecordingArray>
      <LangRecording>
        <PromptLangID>0</PromptLangID>
        <MediaFileID>f15612b7-327b-4edf-af30-cd8a2fc4a068</MediaFileID>
      </LangRecording>
      <LangRecording>
        <PromptLangID>2</PromptLangID>
        <MediaFileID>6d103af4-cd32-41c6-8040-286bf4128395</MediaFileID>
      </LangRecording>
    </LangRecordingArray>
  </PromptRecording>
</CXPromptRecordingUpdate>
```

The following example demonstrates adding an entry with PromptLangID = 3 while simultaneously removing an entry with PromptLangID = 2 from the LangRecordingArray array. Note that this being a partial update, any other existing array elements are left unchanged.

```
<CXPromptRecordingUpdate>
  <SessionID>229574173</SessionID>
  <SessionInstanceID>2a6ec288-eda5-4327-8db0-638aec22dca2</SessionInstanceID>
  <PromptRecording>
    <ID>57001</ID>
    <RecordingTypeID>Greeting</RecordingTypeID>
    <RecordingTypeSpecificID>0</RecordingTypeSpecificID>
    <LangRecordingArray>
      <LangRecording>
        <PromptLangID>3</PromptLangID>
        <MediaFileID>9a03cd71-a5b1-444b-9216-53c317748ee8</MediaFileID>
      </LangRecording>
    </LangRecordingArray>
  </PromptRecording>
  <OldObject>
    <PromptRecording>
      <ID>57001</ID>
```

```

        <RecordingTypeID>Greeting</RecordingTypeID>
        <RecordingTypeSpecificID>0</RecordingTypeSpecificID>
        <LangRecordingArray>
            <LangRecording>
                <PromptLangID>2</PromptLangID>
                <MediaFileID>6d103af4-cd32-41c6-8040-
286bf4128395</MediaFileID>
            </LangRecording>
        </LangRecordingArray>
    </PromptRecording>
</OldObject>
</CXPromptRecordingUpdate>

```

## See Also

[CXPromptRecordingListGet](#), [Mailbox Configuration Functions](#)

## CXMBGroupAssignmentListGet

Retrieves the list of Groups to which a specific mailbox is assigned.

## Syntax

```

<CXMBGroupAssignmentListGet>
    <SessionID></SessionID>
    <SessionInstanceID></SessionInstanceID>
    <MBID></MBID>
    <MBGroupID></MBGroupID>
</CXMBGroupAssignmentListGet>

```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

MBID

- The mailbox number for which the Group assignments are to be retrieved.

MBGroupID

- An optional parameter which specifies the identifier for a specific Group to retrieve. If this parameter is not specified, all Group assignments for the specified mailbox number will be retrieved.

## Output Parameters

### MBGroupAssignmentList

- Container for a list of one or more Group assignments.
- MBGroupAssignment
  - Container for a single Group assignment.
  - MBGroupID
    - Unique identifier of the Group.
  - MBID
    - Mailbox number to which the assignment is made.

## XML Return

```
<CXResponse>
  <MBGroupAssignmentList>
    <MBGroupAssignment>
      <MBID>5410</MBID>
      <MBGroupID>1</MBGroupID>
    </MBGroupAssignment>
    <MBGroupAssignment>
      <MBID>5410</MBID>
      <MBGroupID>6</MBGroupID>
    </MBGroupAssignment>
  </MBGroupAssignmentList>
</CXResponse>
```

## Errors

2147749804    Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session

## See Also

[CXMBGroupTypeListGet](#), [CXMBGroupListGet](#), [CXMBGroupAssignmentAdd](#), [CXMBGroupAssignmentDelete](#), [Mailbox Configuration Functions](#)

## CXMBGroupAssignmentAdd

Adds a group to the list of Groups assigned to the specific mailbox.

## Syntax

```
<CXMBGroupAssignmentAdd>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <MBGroupAssignment>
    <MBID></MBID>
    <MBGroupID></MBGroupID>
  </MBGroupAssignment>
</CXMBGroupAssignmentAdd>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

MBGroupAssignment

- Container for a single Group assignment.
- MBID
  - The mailbox number to which the Group is to be assigned.
- MBGroupID
  - Unique identifier of the Group to be assigned.

## Output Parameters

None

## XML Return

```
<CXResponse>
</CXResponse>
```

## Errors

2147749804      Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session

2147750872      Invalid Mailbox ID

- The MBID provided is invalid

2147750871 Missing Mailbox ID

- The MBID was not provided

2147750870 Invalid MBGroupID

- The MBGroupID provided is invalid

## See Also

[CXMBGroupTypeListGet](#), [CXMBGroupListGet](#), [CXMBGroupAssignmentListGet](#), [CXMBGroupAssignmentDelete](#), [Mailbox Configuration Functions](#)

## CXMBGroupAssignmentDelete

Deletes a Group from the list of Groups assigned to the specified Mailbox.

## Syntax

```
<CXMBGroupAssignmentDelete>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <MBGroupAssignment>
    <MBID></MBID>
    <MBGroupID></MBGroupID>
  </MBGroupAssignment>
</CXMBGroupAssignmentDelete>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

MBGroupAssignment

- Container for a single Group assignment.
- MBID
  - The mailbox number to which the Group is assigned.
- MBGroupID
  - Unique identifier of the assigned Group from which the mailbox is to be removed.

## Output Parameters

None

## XML Return

```
<CXResponse>  
</CXResponse>
```

## Errors

2147749804      Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session

2147750872      Invalid Mailbox

- The MBID provided is invalid

2147750871      Missing Mailbox ID

- The MBID was not provided

2147750870      Invalid MBGroupID

- The MBGroupID provided is invalid

## See Also

[CXMBGroupTypeListGet](#), [CXMBGroupListGet](#), [CXMBGroupAssignmentListGet](#), [CXMBGroupAssignmentAdd](#), [Mailbox Configuration Functions](#)

## CXDeviceListGet

Get the list of configured devices.

## Syntax

```
<CXDeviceListGet>  
  <SessionID></SessionID>  
  <SessionInstanceID></SessionInstanceID>  
  <MBID></MBID>  
  <DevAddr></DevAddr>  
  <DevAddrExactMatch></DevAddrExactMatch>  
  <MaxCount></MaxCount>  
</CXDeviceListGet>
```



## Input Parameters

### SessionID

- Identifies the session. This value is returned by a previous call to CXLogon.

### SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to CXLogon.

### Device

- Container for a single device configuration. Refer to the [Device](#) Properties section for a complete listing of device properties.

## Output Parameters

### DevID

- The device ID assigned to the new device..

## XML Return

```
<XMLResponse xmlns="http://www.mitel.com/CXIf">
  <CXResponse>
    <DevID></DevID>
  </CXResponse>
</XMLResponse>
```

## Errors

### 2147749804 Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

## Example

The following example adds a Personal Mobile type device for mailbox 5410.

```
<CXDeviceListAdd>
  <SessionID>896487536</SessionID>
  <SessionInstanceID>63b254ef-691a-47b2-b854-cb35c7cff52a</SessionInstanceID>
  <Device>
    <MBID>5410</MBID>
    <DevAddr>4255551212</DevAddr>
    <Name>Personal Mobile</Name>
    <DevUsageType>PersonalMobile</DevUsageType>
    <RingTimeoutSec>15</RingTimeoutSec>
    <Enabled>1</Enabled>
```

```
<IsPrimary>1</IsPrimary>
<DerivedDeviceTypeEnum>VoiceCallAutoLogon</DerivedDeviceTypeEnum>
</Device>
</CXDeviceListAdd>
```

## See Also

[CXDeviceListGet](#), [CXDeviceUpdate](#), [CXDeviceFinalize](#), [Mailbox Configuration Functions](#)

## CXDeviceAdd

Add a device record.

## Syntax

```
<CXDeviceAdd>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <Device>
    <MBID></MBID>
    <DevAddr></DevAddr>
  </Device>
</CXDeviceAdd>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

Device

- Container for a single device configuration. Refer to the [Device](#) Properties section for a complete listing of device properties.

## Output Parameters

DevID

- The device ID assigned to the new device..

## XML Return

```
<XMLResponse xmlns="http://www.mitel.com/CXIf">
  <CXResponse>
    <DevID></DevID>
  </CXResponse>
</XMLResponse>
```

## Errors

2147749804      Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

## Example

The following example adds a Personal Mobile type device for mailbox 5410.

```
<CXDeviceListAdd>
  <SessionID>896487536</SessionID>
  <SessionInstanceID>63b254ef-691a-47b2-b854-cb35c7cff52a</SessionInstanceID>
  <Device>
    <MBID>5410</MBID>
    <DevAddr>4255551212</DevAddr>
    <Name>Personal Mobile</Name>
    <DevUsageType>PersonalMobile</DevUsageType>
    <RingTimeoutSec>15</RingTimeoutSec>
    <Enabled>1</Enabled>
    <IsPrimary>1</IsPrimary>
    <DerivedDeviceTypeEnum>VoiceCallAutoLogon</DerivedDeviceTypeEnum>
  </Device>
</CXDeviceListAdd>
```

## See Also

[CXDeviceListGet](#), [CXDeviceUpdate](#), [CXDeviceFinalize](#), [Mailbox Configuration Functions](#)

## CXDeviceUpdate

Update properties of a device record. This function is also used to delete a device record.

## Syntax

```
<CXDeviceUpdate>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
```

```
<Device>
  <MBID></MBID>
  <DevID></DevID>
</Device>
</CXDeviceUpdate>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

Device

- Updated values for one or more device properties. Refer to the section [Device Properties](#) for a listing of device properties.

## Output Parameters

None

## XML Return

```
<XMLResponse xmlns="http://www.mitel.com/CXIf">
  <CXResponse/>
</XMLResponse>
```

## Errors

2147749804    Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147749305    Invalid argument

- The MBID or DevID is missing or invalid.

2147749315    Missing argument

- The MBID or DevID is missing or invalid.

2147749908    DAL error: Missing field

- IsPrimary is missing or invalid.

## Example

The following example updates the Ring Timeout property for device ID 15 which is associated with mailbox 5410.

```
<CXDeviceUpdate>
  <SessionID>206505258</SessionID>
  <SessionInstanceID>4274597c-6656-4bfc-863b-64ec19c840d5</SessionInstanceID>
  <Device>
    <DevID>15</DevID>
    <MBID>5410</MBID>
    <RingTimeoutSec>20</RingTimeoutSec>
  </Device>
</CXDeviceUpdate>
```

## Deleting a Device

**CXDeviceUpdate** is also used to delete a device. To delete a device, provide the **DevID**, **MBID**, and **IsPrimary** properties of the device and set the **DevAddr** to a null value. An example of using **CXDeviceUpdate** to delete a device is shown below:

```
<CXDeviceUpdate>
  <SessionID>1086719013</SessionID>
  <SessionInstanceID></SessionInstanceID>
  <Device>
    <DevID>16</DevID>
    <MBID>5630</MBID>
    <IsPrimary>0</IsPrimary>
    <DevAddr></DevAddr>
  </Device>
</CXDeviceUpdate>
```

## See Also

[CXDeviceListGet](#), [CXDeviceAdd](#), [CXDeviceFinalize](#), [Mailbox Configuration Functions](#)

## CXDeviceFinalize

Finalize critical settings for the devices associated with the specified subscriber. This function insures that important device settings that are interrelated among devices are correctly set and that Availability call lists reference the correct devices. **CXDeviceFinalize** should be called after adding, updating or deleting Devices.

## Syntax

```
<CXDeviceFinalize>
  <SessionID></SessionID>
```

```
<SessionInstanceID></SessionInstanceID>
<MBID></MBID>
</CXDeviceFinalize>
```

## Input Parameters

### SessionID

- Identifies the session. This value is returned by a previous call to CXLogon.

### SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to CXLogon.

### MBID

- The mailbox number for which the device list is to be finalized.

## Output Parameters

None

## XML Return

```
<XMLResponse xmlns="http://www.mitel.com/CXIf">
  <CXResponse/>
</XMLResponse>
```

## Errors

2147749804    Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

## See Also

[CXDeviceAdd](#), [CXDeviceUpdate](#), [Mailbox Configuration Functions](#)

## CXMBSpeechAliasListGet

Get the list of speech aliases for a particular subscriber or a list of all mailbox speech aliases on the system. Speech aliases allow specification of the different ways that a person's name may be spoken for the speech auto attendant or in the VUI application. Aliases can be used to specify nicknames, such as Beth for Elizabeth, maiden names or other names a person goes by and also may be used to specify a phonetic spelling of a person's name that more closely matches how the name is pronounced.

## Syntax

```
<CXMBSpeechAliasListGet>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <MBID></MBID>
</CXMBSpeechAliasListGet>
```

## Input Parameters

### SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

### SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

### MBID

- The mailbox number for which the speech aliases are to be retrieved. If this number is not specified, a list of all speech aliases on the system is retrieved.

## Output Parameters

### MBSpeechAliasList

- Container for a list of one or more speech aliases.
- MBSpeechAlias
  - Container for a single speech alias.
  - ID
    - Unique identifier of the speech alias.
  - MBID
    - Mailbox ID associated with the speech alias.
  - Phrase
    - The phrase that represents the alias.
    - CultureName
      - The culture name for the alias. An empty culture name will designate this alias as a global alias. A global alias is an alias that will be used for all cultures.

## XML Return

```
<CXResponse>
  <MBSpeechAliasList>
```

```

    <MBSpeechAlias>
      <ID>1</ID>
      <MBID>5410</MBID>
      <Phrase>Joe Tester</Phrase>
      <CultureName/>
    </MBSpeechAlias>
    <MBSpeechAlias>
      <ID>2</ID>
      <MBID>5410</MBID>
      <Phrase>Joe Testeur</Phrase>
      <CultureName>fr-ca</CultureName>
    </MBSpeechAlias>
  </MBSpeechAliasList>
</CXResponse>

```

## Errors

2147749804    Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

## See Also

[CXMBSpeechAliasAdd](#), [CXMBSpeechAliasDelete](#), [CXMBSpeechAliasUpdate](#), [Mailbox Configuration Functions](#)

## CXMBSpeechAliasAdd

Add a new speech alias for a particular subscriber.

## Syntax

```

<CXMBSpeechAliasAdd>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <MBSpeechAlias>
    <MBID></MBID>
    <Phrase></Phrase>
    <CultureName></CultureName>
  </MBSpeechAlias>
</CXMBSpeechAliasAdd>

```

## Input Parameters

SessionID



- Identifies the session. This value is returned by a previous call to **CXLogon**.

#### SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

#### MBSpeechAlias

- This is the Speech alias structure. Specify the following fields:
- MBID
  - Mailbox ID associated with the speech alias.
- Phrase
  - The phrase that represents the alias. It must be unique and not blank.
- CultureName
  - The culture name for the alias. An empty culture name will designate this alias as a global alias. A global alias is an alias that will be used for all cultures.

## Output Parameters

#### ID

- Unique identifier of the speech alias.

## XML Return

```
<CXResponse>
  <ID>7</ID>
</CXResponse>
```

## Errors

#### 2147749804 Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

#### 2147749305 Invalid argument

- An invalid or missing argument was specified.

#### 2147750861 Missing parameter

- A parameter was not specified.

#### 2147750862 Duplicate alias phrase

- A duplicate alias phrase was specified.

#### 2147750871 Missing mailbox number

- A mailbox number was not specified.

## See Also

[CXMBSpeechAliasListGet](#), [CXMBSpeechAliasDelete](#), [CXMBSpeechAliasUpdate](#), [Mailbox Configuration Functions](#)

# CXMBSpeechAliasUpdate

Update a speech alias.

## Syntax

```
<CXMBSpeechAliasUpdate>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <MBSpeechAlias>
    <ID></ID>
    <MBID></MBID>
    <Phrase></Phrase>
    <CultureName></CultureName>
  </MBSpeechAlias>
</CXMBSpeechAliasUpdate>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to CXLogon.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to CXLogon.

MBSpeechAlias

- Container for a single speech alias.
- ID
  - The unique identifier of the speech alias to be updated.
- MBID
  - Mailbox ID associated with the speech alias.
- Phrase
  - The phrase that represents the alias. It must be unique and not blank.
- CultureName

- The culture name for the alias. An empty culture name will designate this alias as a global alias. A global alias is an alias that will be used for all cultures.

## Output Parameters

None

## XML Return

```
<CXResponse>
</CXResponse>
```

## Errors

2147749804      Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147750861      Missing parameter

- A parameter was not specified.

2147750862      Duplicate alias phrase

- A duplicate alias phrase was specified.

2147750871      Missing mailbox number

- No MBID was provided.

2147750872      Invalid mailbox number

- The MBID provided is invalid.

## See Also

[CXMBSpeechAliasListGet](#), [CXMBSpeechAliasAdd](#), [CXMBSpeechAliasDelete](#), [Mailbox Configuration Functions](#)

## CXMBSpeechAliasDelete

Delete a speech alias.

## Syntax

```
<CXMBSpeechAliasDelete>
  <SessionID> </SessionID>
```

```
<SessionInstanceID> </SessionInstanceID>  
<ID> </ID>  
</CXMBSpeechAliasDelete>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to CXLogon.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to CXLogon.
- ID
  - The unique identifier of the speech alias to be deleted.

## Output Parameters

None

## XML Return

```
<CXResponse>  
</CXResponse>
```

## Errors

2147749804      Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147749305      Invalid argument

- An invalid or missing argument was specified.

2147750880      Missing parameter

- The ID was not specified.

## See Also

[CXMBSpeechAliasListGet](#), [CXMBSpeechAliasAdd](#), [CXMBSpeechAliasUpdate](#), [Mailbox Configuration Functions](#)

# CXMBDataGet

Loads the requested data categories for the specified mailbox.

## Syntax

```
<CXMBDataGet>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <MBID></MBID>
  <MBDataCategoryList>
    <MBDataCategory></MBDataCategory>
  </MBDataCategoryList>
</CXMBDataGet>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

MBID

- The mailbox number of the mailbox.

MBDataCategoryList

- Container for the data categories.
- MBDataCategory
  - The category of data to load. Some categories are not relevant for certain mailbox types, in which case the returned structure will simply not include an element for that category. Legal values are:
    - *DeviceList*
      - Include the list of devices.
    - *MB*
      - Include the basic mailbox structure.
    - *MBDistListAllowedMBIDArray*
      - Include the list of mailboxes that can address the Distribution List mailbox.
    - *MBDistListMemberArray*
      - Include the members of the Distribution List mailbox.
    - *MBCPAActionArray*

- Include the actions for the Call Processor mailbox.
- *MBGroupAssignmentList*
  - Include the list of groups this Subscriber mailbox is assigned to.
- *MBSpeechAliasList*
  - Include the list of speech aliases for this mailbox.
- *MBSubIMNCalloutArray*
  - Include the list of Immediate Message Notification callout configurations for the Subscriber mailbox.
- *MBSubEmail*
  - Include the email configuration information for the Subscriber mailbox.

## Output Parameters

### MBData

- The entire mailbox structure, containing various data categories. Only the categories requested in **MBDataCategoryList** will be returned, and only those that have a non-empty value.
- DeviceList
  - The list of devices.
  - Device
    - A single device configuration. Refer to the section [Device](#) Properties for a listing of device properties.
- MB
  - The basic mailbox structure. See description in CXMBGet.
- MBDistListAllowedMBIDArray
  - The list of mailboxes that can address the Distribution List. See description in CXMBDistListAllowedMBIDArrayGet.
- MBDistListMemberArray
  - The members of the Distribution List mailbox. See description in CXMBDistListMemberArrayGet.
- MBCPActionArray
  - The array of actions for the Call Processor mailbox. See description in CXMBCPActionArrayGet.
- MBGroupAssignmentList
  - The list of groups this Subscriber is assigned to. See description in CXMBGroupAssignmentListGet.
- MBSpeechAliasList
  - The list of speech aliases for this mailbox. See description in CXMBSpeechAliasListGet.

- **MBSubIMNCalloutArray**
  - The list of Immediate Message Notification callout configurations for the Subscriber mailbox. See description in `CXMBSubIMNCalloutArrayGet`.
- **MBSubEmail**
  - The email configuration information for the Subscriber mailbox. See description in `CXMBSubEmailGet`.

## Errors

2147749305    Invalid argument

- An invalid argument has been provided.

2147749804    Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147750608    Data-layer error: Missing field

- A required element was not provided.

2147750611    Data-layer error: Insufficient privileges

- The logged on user does not have sufficient privileges for accessing this mailbox.

2147750612    Data-layer error: Mailbox not found

- MBID does not exist on the local server.

## See Also

[CXMBDataAdd](#), [CXMBDataUpdate](#), [CXMBDataDelete](#), [CXMBDataRenumber](#), [Mailbox Configuration Functions](#)

## CXMBDataAdd

Adds a new mailbox, including the basic mailbox structure plus optional additional mailbox data categories.

## Syntax

```
<CXMBDataAdd>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <TemplateMBID></TemplateMBID>
  <MBData>
    <MB></MB>
    <DeviceList></DeviceList>
  </MBData>
```

```
</CXMBDataAdd>
```

## Input Parameters

### SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

### SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

### TemplateMBID

- This is an optional parameter. If set to a COS MBID, the default values for the subscriber mailbox being added will be derived from this template COS mailbox. Note that derivation of the field values from the COS mailbox specified by **TemplateMBID** only happens for unspecified fields.

### MBData

- The entire mailbox structure, containing various data categories. MB is always required. DeviceList is required when adding a Subscriber. Otherwise all other fields are optional.
- DeviceList
  - Required for Subscriber mailboxes, otherwise optional. The list of devices.
  - Device
    - A single device configuration. Refer to the section [Device](#) Properties for a listing of device properties.
- MB
  - Required. The basic mailbox structure. See description in CXMBAdd.
- MBDistListAllowedMBIDArray
  - Optional. The list of mailboxes that can address the Distribution List. See description in CXMBDistListAllowedMBIDArrayUpdate.
- MBDistListMemberArray
  - Optional. The members of the Distribution List mailbox. See description in CXMBDistListMemberArrayUpdate.
- MBCPActionArray
  - Optional. The array of actions for the Call Processor mailbox. See description in CXMBCPActionArrayUpdate.
- MBGroupAssignmentList
  - Optional. The list of groups this Subscriber is assigned to. See description in CXMBGroupAssignmentAdd.
- MBSpeechAliasList
  - Optional. The list of speech aliases for this mailbox. See description in CXMBSpeechAliasAdd.



- MSubIMNCalloutArray
  - Optional. The list of Immediate Message Notification callout configurations for the Subscriber mailbox. See description in CXMBSubIMNCalloutArrayUpdate.
- MSubEmail
  - Optional. The email configuration information for the Subscriber mailbox. See description in CXMBSubEmailUpdate.

## Output Parameters

None

## Errors

2147749804 Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147750608 Data-layer error: Missing field

- A required element was not provided.

2147750611 Data-layer error: Insufficient privileges

- The logged on user does not have sufficient privileges for accessing this mailbox.

2147750613 Data-layer error: Mailbox already exists

- MBID already exists on the local server.

## See Also

[CXMBDataGet](#), [CXMBDataUpdate](#), [CXMBDataDelete](#), [CXMBDataRenum](#), [Mailbox Configuration Functions](#)

## CXMBDataUpdate

Updates an existing mailbox. This can include a variety of data categories.

## Syntax

```
<CXMBDataUpdate>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <MBData>
    <MB></MB>
    <DeviceList></DeviceList>
  </MBData>
```

```
<OldObject>
  <MBData>
    </MBData>
  </OldObject>
</CXMBDataUpdate>
```

## Input Parameters

### SessionID

- Identifies the session. This value is returned by a previous call to CXLogon.

### SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to CXLogon.

### MBID

- The mailbox number of the mailbox.

### MBData

- The entire mailbox structure, containing various data categories. All fields are optional. Whichever fields are populated will be updated.
- DeviceList
  - Optional. The list of devices.
  - Device
    - A single device configuration. Refer to the section [Device](#) Properties for a listing of device properties.
- MB
  - Optional. The basic mailbox structure. See description in CXMBUpdate.

**NOTE** MB/MBCommon/MBID and MB/MBCommon/MBType cannot be changed through CXMBDataUpdate.

- MBDistListAllowedMBIDArray
  - Optional. The list of mailboxes that can address the Distribution List. See description in CXMBDistListAllowedMBIDArrayUpdate.
- MBDistListMemberArray
  - Optional. The members of the Distribution List mailbox. See description in CXMBDistListMemberArrayUpdate.
- MBCPActionArray
  - Optional. The array of actions for the Call Processor mailbox. See description in CXMBCPActionArrayUpdate.
  - MBGroupAssignmentList

- Currently not supported in CXMBDataUpdate.
- MBSpeechAliasList
  - Currently not supported in CXMBDataUpdate.
- MBSubIMNCalloutArray
  - Optional. The list of Immediate Message Notification callout configurations for the Subscriber mailbox. See description in CXMBSubIMNCalloutArrayUpdate.
  - MBSubEmail
  - Optional. The email configuration information for the Subscriber mailbox. See description in CXMBSubEmailUpdate.

#### OldObject

- This optional field represents the baseline from which the update is being made. If populated, the difference between this **MBData** and the update **MBData** determines the updates that are applied. See Comments for further details.
- MBData
  - The entire mailbox structure, containing various data categories. All fields are optional. Whichever fields are populated will be used as the corresponding old object.
  - DeviceList
    - Optional. The list of devices.
  - Device
    - Optional. See description above.
  - MB
    - Optional. See description above.
  - MBDistListAllowedMBIDArray
    - Optional. See description above.
  - MBDistListMemberArray
    - Optional. See description above.
  - MBGroupAssignmentList
    - Currently not supported in CXMBDataUpdate.
  - MBSpeechAliasList
    - Currently not supported in CXMBDataUpdate.
  - MBSubIMNCalloutArray
    - Optional. See description above.
  - MBSubEmail
    - Optional. See description above.

## Output Parameters

None

## Errors

2147749804 Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147750608 Data-layer error: Missing field

- A required element was not provided.

2147750611 Data-layer error: Insufficient privileges

- The logged on user does not have sufficient privileges for accessing this mailbox.

2147750613 Data-layer error: Mailbox already exists

- MBID already exists on the local server.

## Comments

**CXMBDataUpdate** encapsulates the *update* functions for each of the contained data structure for the mailbox. For example, if **MBData/MB** is populated, **CXMBUpdate** is called within this function. Similarly if **MBData/MBDistListMemberArray** is populated, **CXMBDistListMemberArrayUpdate** is called within this function. Note that if there is a failure part way through updating the contained data structures, the function will not roll back those updates that have taken place.

If **OldObject/MBData** is not populated, the update is based completely on the existing mailbox. If bis populated, the update is based on the difference between **bMBData** and **OldObject/MBData**. The exact behavior depends on the data type. See the *update* function of the data type for details.

## Example

The following example demonstrates updating the distribution list to be propagated, as well as adding a single member.

```
<CXMBDataUpdate>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
  <MBData>
    <MB>
      <MBCommon>
        <MBID>2001</MBID>
        <MBType>D</MBType>
        <Propagate>1</Propagate>
      </MBCommon>
    </MB>
    <MBDistListMemberArray>
      <Member>
```

```

        <MemberMBID>1004</MemberMBID>
        <RemoteMBNo></RemoteMBNo>
    </Member>
</MBDistListMemberArray>
</MBData>
<OldObject>
    <MBData>
        <MBDistListMemberArray>
        </MBDistListMemberArray>
    </MBData>
</OldObject>
</CXMBDataUpdate>

```

## See Also

[CXMBDataGet](#), [CXMBDataAdd](#), [CXMBDataTemplateUpdate](#), [CXMBDataDelete](#), [CXMBDataRenumber](#), [Mailbox Configuration Functions](#)

## CXMBDataTemplateUpdate

Updates the specified mailboxes using a template mailbox (COS mailbox).

## Syntax

```

<CXMBDataTemplateUpdate>
    <SessionID></SessionID>
    <SessionInstanceID></SessionInstanceID>
    <TemplateMBID></TemplateMBID>    <TemplateMBData>
        <MB>
            <MBCommon>
                <MBType></MBType>
            </MBCommon>
        </MB>
    </TemplateMBData>    <OldTemplateMBData>
        <MB>
            <MBCommon>
                <MBType></MBType>
            </MBCommon>
        </MB>
    </OldTemplateMBData>    <MBIDList>
        <MBID></MBID>
    </MBIDList>
</CXMBDataTemplateUpdate>

```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

#### SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

#### TemplateMBID

- The MBID of the template mailbox to be used as the source mailbox for updating the specified mailboxes. Specify either the **TemplateMBID** or the **TemplateMBData** parameter.

#### TemplateMBData

- The template mailbox structure to be used as the source mailbox for updating the specified mailboxes. Specify either the **TemplateMBID** or the **TemplateMBData** parameter.
- MB
  - The basic mailbox structure. See description in [CXMBUpdate](#).

#### OldTemplateMBData

- This optional field represents the baseline for the template mailbox from which the update is being made. If populated, the difference between this structure and the **TemplateMBData** structure determines the updates that are applied to the specified mailboxes.
- MB
  - The basic mailbox structure. See description in [CXMBUpdate](#).

#### MBIDList

- This is a container of one or more MBID fields. Each MBID field represents one mailbox that needs to be updated.
- MBID
  - MBID of the mailbox to be updated.

## Output Parameters

#### MBErrorList

The top-level container for the per-mailbox error values. This contains one or more **MBError** fields. Each **MBError** field holds the error value for exactly one mailbox. This list of **MBError** fields is returned in the same order as the **MBIDList** input parameter.

If a **MBError** field's value is 0 then that means that the mailbox was successfully updated. If it is greater than zero then it indicates that some kind of error happened while updating this mailbox. Here is the table of the error values that can be returned via this field:

Table 10. Returnable error values

Error Value	Error String	Description
2147750612	Data-layer error: Mailbox not found	The mailbox being updated does not exist on the local

server.

---

2147750616	Data-layer error: The template mailbox is of the wrong type	The mailbox specified by TemplateMBID is of the wrong type for updating this mailbox.
------------	---	---

---

## Errors

2147749804 Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147750608 Data-layer error: Missing field

- A required element was not provided.

2147750611 Data-layer error: Insufficient privileges

- The logged on user does not have sufficient privileges for accessing this mailbox.

2147750615 Data-layer error: The template mailbox was not found

- The mailbox specified by **TemplateMBID** does not exist.

## Comments

Note that only the inheritable fields in the template mailbox are updated. Non-inheritable fields (such as Name fields, EmailID, Email Server Type etc.) are never updated.

If **OldTemplateMBData** is not populated, all the inheritable fields of **TemplateMBData** are used to update the subscriber mailbox. If the **OldTemplateMBData** is populated, the update is based on the difference between **TemplateMBData** and **OldTemplateMBData**. This is important when updating the mailboxes with only the fields that changed in the COS mailbox.

Updating via this function instead of **CXMBTemplateUpdate**, causes the VIM fields to be updated as well provided that the update is made by specifying a **TemplateMBID**.

## Example

The following example demonstrates updating all inheritable fields of the subscriber mailbox 5710 with values from the COS mailbox 7000.

```
<CXMBDataTemplateUpdate>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
  <TemplateMBID>7000</TemplateMBID>
  <MBIDList>
    <MBID>5710</MBID>
  </MBIDList>
</CXMBDataTemplateUpdate>
```

## See Also

[CXMBDataGet](#), [CXMBDataAdd](#), [CXMBDataUpdate](#), [CXMBDataDelete](#), [CXMBDataRenummer](#), [Mailbox Configuration Functions](#)

## CXMBDataDelete

Deletes one or more mailboxes.

## Syntax

```
<CXMBDataDelete>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <MBID></MBID>
</CXMBDataDelete>
<CXMBDataDelete>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <FirstMBID></FirstMBID>
  <LastMBID></LastMBID>
  <MBType></MBType>
</CXMBDataDelete>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

MBID

- The mailbox number to delete.

FirstMBID

- The first mailbox number of the range of mailboxes to delete.

LastMBID

- The last mailbox number of the range of mailboxes to delete.

MBType

- The optional value which specifies a specific mailbox type to delete when doing a range delete. See MBType in the MBCommon section for valid mailbox types.



## Return Value

DeletedCount

- The number of mailboxes deleted (if range delete).

## XML Return (range delete only)

```
<Count></Count>
```

## Errors

2147749305 Invalid argument

- An invalid argument has been provided.

2147749804 Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147749317 Service not available

- The server is not running. The server must be running when deleting mailboxes.

## Comments

Either MBID or the combination of FirstMBID and LastMBID should be provided, but these should never be mixed.

The one exception to the range delete is that unless **MBType** is populated with *P* (Digital Networking mailbox type), correspondent Digital Networking mailboxes that fall within the mailbox number range will not be deleted.

## Example

The following example deletes the mailbox with mailbox number 2001.

```
<CXMBDataDelete>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
  <MBID>2001</MBID>
</CXMBDataDelete>
```

The following example deletes the range of mailboxes from mailbox numbers 2001-2004.

```
<CXMBDataDelete>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
  <FirstMBID>2001</FirstMBID>
  <LastMBID>2004</LastMBID>
```

```
</CXMBDataDelete>
```

The following example deletes all Subscriber mailboxes with mailbox numbers in the range 2001-2004.

```
<CXMBDataDelete>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
  <FirstMBID>2001</FirstMBID>
  <LastMBID>2004</LastMBID>
  <MBType>A</MBType>
</CXMBDataDelete>
```

## See Also

[CXMBDataGet](#), [CXMBDataAdd](#), [CXMBDataUpdate](#), [CXMBDataRenumber](#), [Mailbox Configuration Functions](#)

## CXMBDataRenumber

Renumbers an existing mailbox.

## Syntax

```
<CXMBDataRenumber>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <OldMBID></OldMBID>
  <NewMBID></NewMBID>
  <IncludeExtensions></IncludeExtensions>
</CXMBDataRenumber>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

OldMBID

- The mailbox number of an existing mailbox.

NewMBID

- The new mailbox number. This number must have the correct number of digits, which is returned as the **MailboxIDLength** value by the CXMailboxDefaults function.

## IncludeExtensions

- This optional field is for use when the Subscriber mailbox number matches the number of its primary extension device. In this scenario, if the field is specified, the primary extension device number will also be changed in order to *follow* the mailbox number change.

## Return Value

None

## Errors

2147749305 Invalid argument

- An invalid argument has been provided.

2147749804 Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147750612 Data-layer error: Mailbox not found

- No mailbox with mailbox number OldMBID exists.

2147750613 Data-layer error: Mailbox already exists

- A mailbox with mailbox number NewMBID already exists.

## Example

The following example rennumbers mailbox 2001 to 2002.

```
<CXMailboxRenumber>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
  <OldMBID>2001</OldMBID>
  <NewMBID>2002</NewMBID>
</CXMailboxRenumber>
```

The following example rennumbers mailbox 2001 to 2002. Additionally, if mailbox 2001 is a Subscriber mailbox and has a primary extension device with number 2001, the primary extension device number will change to 2002.

```
<CXMailboxRenumber>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
  <OldMBID>2001</OldMBID>
  <NewMBID>2002</NewMBID>
  <IncludeExtensions>1</IncludeExtensions>
</CXMailboxRenumber>
```

## See Also

[CXMBDataGet](#), [CXMBDataAdd](#), [CXMBDataUpdate](#), [CXMBDataRenum](#), [Mailbox Configuration Functions](#)

## CXAvailabilityGreetingListGet

Gets all Availability Greeting records or a specific Availability Greeting record for a Subscriber or Availability Class of Service mailbox.

## Syntax

```
<CXAvailabilityStateListGet>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <MBID></MBID>
  <AvailGreetingID></AvailGreetingID>
</CXAvailabilityStateListGet>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

MBID

- The mailbox number for which the Availability Greeting information is retrieved.

AvailGreetingID

- A specific Availability Greeting ID for which information is to be retrieved. If this parameter is not specified, information is retrieved for all Availability Greetings for the specified mailbox.

## Output Parameters

AvailabilityGreetingList

- Container for a list of one or more Availability Greeting configurations.
  - AvailabilityGreeting
    - Container for a single Availability Greeting configuration.

## XML Return

```
<XMLResponse xmlns="http://www.mitel.com/CXIf">
  <CXResponse>
    <AvailabilityGreetingList>
      <AvailabilityGreeting>
        <AvailGreetingID>23</AvailGreetingID>
        <MBID>5410</MBID>
        <Name>Out of the Office</Name>
        <SpeechCmdID>1417</SpeechCmdID>
        <TTSNameAlias />
        <SpokenNameFile />
      </AvailabilityGreeting>
    </AvailabilityGreetingList>
  </CXResponse>
</XMLResponse>
```

## Errors

2147749804    Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147749315    Missing argument

- Neither the AvailGreetingID or the MBID are specified. One is required.

## Example

The following example returns the Availability Greeting information for Availability Greeting ID 23 for mailbox 5410.

```
<CXAvailabilityGreetingListGet>
  <SessionID>519196341</SessionID>
  <SessionInstanceID>63b254ef-691a-47b2-b854-cb35c7cff52a</SessionInstanceID>
  <MBID>5410</MBID>
  <AvailGreetingID>23</AvailGreetingID>
</CXAvailabilityGreetingListGet>
```

## See Also

[CXAvailabilityStateUpdate](#)

## CXAvailabilityActivate

Sets an Availability State or Availability Day manual override for a specific mailbox for a specified range of time. Either a single state can be activated or a replacement schedule can be activated.

Note: Only a single manual override may be set at any given time. In most cases it is preferable to set the **ActivationTimeStamp** to the current time to have the override start immediately.

## Syntax

```
<CXAvailabilityActivate>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <TZID></TZID>
  <MBID></MBID>
  <AvailStateID></AvailStateID>
  <AlternateDefaultGreetingType></AlternateDefaultGreetingType>
  <AlternateAvailGreetingID></AlternateAvailGreetingID>
  <AvailDayID></AvailDayID>
  <ActivationTimeStamp></ActivationTimeStamp>
  <ExpirationTimeStamp></ExpirationTimeStamp>
</CXAvailabilityActivate>
```

## Input Parameters

### SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

### SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

### TZID

- The optional Time Zone ID. When no **ExpirationTimeStamp** is specified, an **AvailStateID** is specified, and the state's **DefaultBehavior** parameter is set to **StopTimeOfDay**, the TZID is used to create an **ExpirationTimeStamp** based on the state's **DefaultStopTOD**. If not specified, the Time Zone ID of the server is used.

### MBID

- The mailbox number for which the Availability override will apply.

### AvailStateID

- The ID of the Availability State to activate for the override. If specified, **AvailDayID** is ignored.

### AlternateDefaultGreetingType

- Optional: The greeting to be played prior to presenting the list of instructions instead of the normal default **AnnounceDefaultGreetingType**. Options are as follows.
- "None" – No Alternate so follows the **AnnounceDefaultGreetingType** setting.
- "StandardOrBusy" – Play the standard greeting or the busy greeting depending on conditions.
- "OutOfOffice" – Play the Out of Office greeting.
- "AvailGreeting" – Play the Availability Greeting associated with the AvailGreetingID.

#### AlternateAvailGreetingID

- Optional: The ID of the alternate Availability Greeting associated with the state when AlternateDefaultGreetingType has the value of "AvailGreeting".

#### AvailDayID

- The ID of the Availability Day/Schedule to activate for the override. Only Availability Day records where the **DayIndex** is 0 can be used as a schedule override. In other words, you can override with a day/schedule called *Sick Day* but not with a day/schedule called Tuesday. If **AvailStateID** is specified, **AvailDayID** is ignored.

#### ActivationTimestamp

- The optional UTC timestamp at which the Availability override is to start. When the value is 1970-01-01T00:00:00Z or not specified, the override will start immediately.

#### ExpirationTimestamp

- The optional UTC timestamp at which the Availability override is to end. When an override ends, Availability reverts to the defined schedule. If the value is 1970-01-01T00:00:00Z, the override will not expire.
- If **ExpirationTimestamp** is not specified, an **AvailStatID** is specified, then the behavior depends on the value of the state's **DefaultBehavior** field:
  - **Indefinite** - The override will never expire.
  - **StopTimeOfDay** - The TZID will be used to compute the saved value of ExpirationTimestamp from the DefaultStopTOD parameter.
  - **LengthOfTime** - The ExpirationTimestamp will be the sum of the current time and the state's DefaultDurationSeconds field.

## Output Parameters

None

## Errors

2147749804 Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147750872 Invalid mailbox

- The MBID provided does not exist or is not a subscriber mailbox.

2147750871 MBID missing

- The MBID was not specified.

2147750892 Invalid State ID

- There is no Availability State record for the **AvailStateID** specified.

2147750893 Invalid Day ID

- There is no Availability Day record for the **AvailDayID** specified.

2147750894 Invalid Day

- The **AvailabilityDay** record found for **AvailDayID** specified is not a schedule. It is a Day of the Week.

2147749315 Missing argument

- Neither the **AvailDayID** nor the **AvailStateID** are specified OR neither one is valid. A valid one needs to be specified.

## Example

The following example sets an Availability State override to start immediately with an indefinite duration for mailbox 5410.

```
<CXAvailabilityActivate>
  <SessionID>94935008</SessionID>
  <SessionInstanceID>63b254ef-691a-47b2-b854-cb35c7cff52a</SessionInstanceID>
  <MBID>5410</MBID>
  <AvailStateID>33412</AvailStateID>
  <AlternateDefaultGreetingType>None</AlternateDefaultGreetingType>
  <AlternateAvailGreetingID>0</AlternateAvailGreetingID>
  <ActivationTimestamp>-1</ActivationTimestamp>
  <ExpirationTimestamp>-1</ExpirationTimestamp>
</CXAvailabilityActivate>
```

## See Also

[CXAvailabilityEnableSchedules](#), [CXAvailabilityActiveStateGet](#), [CXAvailabilityStateDeviceListGet](#)

## CXAvailabilityEnableSchedules

Enables normal automatic Availability processing. This function cancels Availability overrides and sets Availability processing to follow the subscriber's weekly schedule and automatic overrides.

## Syntax

```
<CXAvailabilityEnableSchedules>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <MBID></MBID>
</CXAvailabilityEnableSchedules>
```

## Input Parameters

SessionID



- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

MBID

- The mailbox number for which the Availability override is to be removed.

## Output Parameters

None

## Errors

2147749804      Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147750872      Invalid mailbox

- The MBID provided does not exist or is not a subscriber mailbox.

## Example

The following example enables normal automatic Availability processing for mailbox 5410.

```
<CXAvailabilityEnableSchedules>
  <SessionID>94935008</SessionID>
  <SessionInstanceID>63b254ef-691a-47b2-b854-cb35c7cff52a</SessionInstanceID>
  <MBID>5410</MBID>
</CXAvailabilityEnableSchedules>
```

## See Also

[CXAvailabilityActivate](#), [CXAvailabilityActiveStateGet](#), [CXAvailabilityStateDeviceListGet](#)

## CXAvailabilityEnable

Enables Availability processing for a subscriber. When enabled, normal, non-Availability call processing is replaced by Availability call processing.

## Syntax

```
<CXAvailabilityEnable>
  <SessionID></SessionID>
```

```
<SessionInstanceID></SessionInstanceID>
<MBID></MBID>
</CXAvailabilityEnable>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

MBID

- The mailbox number for which the Availability is to be enabled.

## Output Parameters

None

## Errors

2147749804      Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147750872      Invalid mailbox

- The MBID provided does not exist or is not a subscriber mailbox.

## Example

The following example enables Availability processing for mailbox 5410.

```
<CXAvailabilityEnable>
  <SessionID>94935008</SessionID>
  <SessionInstanceID>63b254ef-691a-47b2-b854-cb35c7cff52a</SessionInstanceID>
  <MBID>5410</MBID>
</CXAvailabilityEnable>
```

## See Also

[CXAvailabilityActivate](#), [CXAvailabilityActiveStateGet](#), [CXAvailabilityStateDeviceListGet](#)

# CXAvailabilityDisable

Disables Availability processing for a subscriber. When disabled, normal, non-Availability call processing is used.

## Syntax

```
<CXAvailabilityDisable>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <MBID></MBID>
</CXAvailabilityDisable>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

MBID

- The mailbox number for which the Availability is to be disabled.

## Output Parameters

None

## Errors

2147749804     Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147750872     Invalid mailbox

- The MBID provided does not exist or is not a subscriber mailbox.

## Example

The following example disables Availability processing for mailbox 5410.

```
<CXAvailabilityDisable>
  <SessionID>94935008</SessionID>
  <SessionInstanceID>63b254ef-691a-47b2-b854-cb35c7cff52a</SessionInstanceID>
```

```
<MBID>5410</MBID>  
</CXAvailabilityDisable>
```

## See Also

[CXAvailabilityActivate](#), [CXAvailabilityActiveStateGet](#), [CXAvailabilityStateDeviceListGet](#)

## CXAvailabilityActiveStateGet

Gets the active Availability State of the specified mailbox for the time and time-zone specified.

## Syntax

```
<CXAvailabilityActiveStateGet>  
  <SessionID></SessionID>  
  <SessionInstanceID></SessionInstanceID>  
  <MBID></MBID>  
  <CurrentTimestamp></CurrentTimestamp>  
  <TZID></TZID>  
</CXAvailabilityActiveStateGet>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

MBID

- The mailbox number to determine the Active Availability State.

CurrentTimestamp

- The optional UTC timestamp to use for the Active Availability State determination. If not specified, the current time is used.

TZID

- The optional Time Zone ID. The TZID is used as follows:
  - The CurrentTimestamp (or actual time of the call if CurrentTimestamp is not provided) will be returned in RequestedDisplayTimestamp for the TZID.
  - The TZID is used to compute the time of day when processing a schedule. The time of day is used to select the time range in effect for the currently active schedule. The currently active schedule can be the normal weekly schedule or a Availability Day manual override schedule.

The end time in the time range is converted into the state's expiration timestamp unless the actual ExpirationTimestamp occurs earlier. Then the ExpirationTimestamp is used as the expiration timestamp of the state. The expiration timestamp is returned in ExpirationDisplayTimestamp for the TZID.

- When a manual override State is in effect, the ExpirationTimestamp is returned in ExpirationDisplayTimestamp for the TZID.
- If not specified, the Time Zone ID of the server is used.

## Output Parameters

### ActiveAvailState

- Container for the summary of the Active Availability State.
- The **DeviceListAvailStateID** value should be used in the function CXAvailabilityStateDeviceListGet as the **AvailStateID** in order to request the list of devices to call.
- AvailDayID
  - The ID of the Availability Day/Schedule override currently active.
- DayName
  - The name of the Day determined using the **SpeechCmdID** of the **AvailabilityDay** record.
- AvailStateID
  - The ID of the active Availability State due to a weekly schedule, a manual override, or an automatic override condition. This will be the same as the **NonAutoAvailStateID** unless an automatic override condition is in effect.
- StateName
  - The name of the active Availability State determined using the **SpeechCmdID** of the **AvailabilityState** record.
- TTSNameAlias
  - The TTS Name Alias.
- SpokenNameFile
  - Identifies the Spoken Name file.
- ParentAvailStateID
  - The unique database record id of the Availability Class of Service state the active state is based on.
- UseParentDeviceList
  - A flag indicating whether the find-me device list is locked to the device list of the **ParentAvailStateID** state.
- Priority
  - The priority of the record found. A '2' indicates an override. A '1' is normal priority for a weekday.

- **ActivationTimestamp**
  - The UTC timestamp when a manual override is to begin. The current time or the **CurrentTimestamp** parameter will be greater or equal to this value.
- **ExpirationTimestamp**
  - The UTC timestamp when the active state will no longer be active. The current time or the **CurrentTimestamp** parameter will be less than this value.
- **StartTOD**
  - The starting time of the schedule's current time range in the form XXYY where XX is 00-23 and YY is 00, 15, 30 or 45 for a complete range of 0000 – 2345
- **StopTOD**
  - The starting time of the schedule's current time range in the form XXYY where XX is 00-24 and YY is 00, 15, 30 or 45 for a complete range of 0000 – 2400.
- **RequestedDisplayTimestamp**
  - The interpretation of the current time or the optional **CurrentTimestamp** parameter based on the TZID in effect.
- **ExpirationDisplayTimestamp**
  - The interpretation of the time when the active Availability State will expire based on the TZID in effect.
- **ActivationReason**
  - The reason why the Availability State is active. Values are:
    - *None* – no Availability State is active
    - *Schedule* – Normal Weekly Schedule
    - *AutoOverride* – as a result of an automatic override condition.
    - *ManualOverrideState* – as a result of a **CXAvailabilityActivate** where an AvailStateID was specified.
    - *ManualOverrideSchedule* – as a result of a where an **AvailStateDay** was specified.
- **NonAutoAvailStateID**
  - The ID of the Availability State resulting from a weekly schedule or a manual override. This will be the same as the **AvailStateID** unless an automatic override condition is in effect. So this is the Availability State that would have been active if automatic override condition were not in effect.
- **NonAutoStateName**
  - The name of the Availability State specified by the **NonAutoAvailStateID** above.
- **AnnounceDefaultGreetingType**
  - The normal default greeting to be played prior to presenting the list of instructions. Options are as follows.
  - "None" - No greeting is played

- "StandardOrBusy" – Play the standard greeting or the busy greeting depending on conditions.
- "OutOfOffice" – Play the Out of Office greeting.
- "AvailGreeting" – Play the Availability Greeting associated with the AvailGreetingID.
- AvailGreetingID
  - The ID of the default Availability Greeting associated with the state.
- AvailGreetingName
  - The name of the Availability Greeting specified by the AvailGreetingID above.
- AlternateDefaultGreetingType
  - The greeting to be played prior to presenting the list of instructions. Options are as follows.
  - "None" – No Alternate so follows the AnnounceDefaultGreetingType setting.
  - "StandardOrBusy" – Play the standard greeting or the busy greeting depending on conditions.
  - "OutOfOffice" – Play the Out of Office greeting.
  - "AvailGreeting" – Play the Availability Greeting associated with the AvailGreetingID.
- AlternateAvailGreetingID
  - The ID of the alternate Availability Greeting associated with the state.
- AlternateAvailGreetingName
  - The name of the alternate Availability Greeting specified by the AlternateAvailGreetingID above.

## XML Return

```
<XMLResponse xmlns="http://www.mitel.com/CXIf">
  <CXResponse>
    <ActiveAvailState>
      <AvailDayID>165</AvailDayID>
      <DayName>Manual State Override</DayName>
      <AvailStateID>208</AvailStateID>
      <StateName>Working from Home</StateName>
      <TTSNameAlias/>
      <SpokenNameFile>en-US_AVLN_WorkingFromHome</SpokenNameFile>
      <ParentAvailStateID>169</ParentAvailStateID>
      <UseParentDeviceList>1</UseParentDeviceList>
      <DeviceListAvailStateID>169</DeviceListAvailStateID>
      <Priority>2</Priority>
      <DayIndex>0</DayIndex>
      <ReturnTimeAvailable>1</ReturnTimeAvailable>
      <ActivationTimestamp>1970-01-01T00:00:00Z</ActivationTimestamp>
      <ExpirationTimestamp>2015-10-14T15:00:00Z</ExpirationTimestamp>
      <ReturnTimestamp>2015-10-14T15:00:00Z</ReturnTimestamp>
      <StartTOD>0</StartTOD>
      <StopTOD>2400</StopTOD>
      <IsWorkTime>1</IsWorkTime>
    </ActiveAvailState>
  </CXResponse>
</XMLResponse>
```

```

        <RequestedDisplayTimestamp>10/14/2015
06:57:45</RequestedDisplayTimestamp>
        <ExpirationDisplayTimestamp>10/14/2015
08:00:00</ExpirationDisplayTimestamp>
        <ReturnTimeDisplayTimestamp>10/14/2015
08:00:00</ReturnTimeDisplayTimestamp>
        <ActivationReason>ManualOverrideState</ActivationReason>
        <NonAutoAvailStateID>208</NonAutoAvailStateID>
        <NonAutoStateName>Working from Home</NonAutoStateName>
        <AnnounceDefaultGreetingType>StandardOrBusy</AnnounceDefaultGreetingType>
        <AvailGreetingID>0</AvailGreetingID>
        <AvailGreetingName/>
        <AlternateDefaultGreetingType>None</AlternateDefaultGreetingType>
        <AlternateAvailGreetingID>0</AlternateAvailGreetingID>
        <AlternateAvailGreetingName/>
    </ActiveAvailState>
</CXResponse>
</XMLResponse>

```

## Example

The following example gets the Active Availability State for mailbox 5410 at a specific time in the Pacific time zone.

```

<CXAvailabilityActiveStateGet>
  <SessionID>94935008</SessionID>
  <SessionInstanceID>63b254ef-691a-47b2-b854-cb35c7cff52a</SessionInstanceID>
  <MBID>5410</MBID>
  <CurrentTimestamp>2015-10-14T14:30:00Z</CurrentTimestamp>
  <TZID>60</TZID>
</CXAvailabilityActiveStateGet>

```

## Errors

2147749804    Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147750872    Invalid mailbox

- The MBID provided does not exist or is not a subscriber mailbox.

## See Also

[CXAvailabilityGreetingListGet](#)

Gets all Availability Greeting records or a specific Availability Greeting record for a Subscriber or Availability Class of Service mailbox.



## Syntax

```
<CXAvailabilityStateListGet>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <MBID></MBID>
  <AvailGreetingID></AvailGreetingID>
</CXAvailabilityStateListGet>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

MBID

- The mailbox number for which the Availability Greeting information is retrieved.

AvailGreetingID

- A specific Availability Greeting ID for which information is to be retrieved. If this parameter is not specified, information is retrieved for all Availability Greetings for the specified mailbox.

## Output Parameters

AvailabilityGreetingList

- Container for a list of one or more Availability Greeting configurations.
  - AvailabilityGreeting
    - Container for a single Availability Greeting configuration.

## XML Return

```
<XMLResponse xmlns="http://www.mitel.com/CXIf">
  <CXResponse>
    <AvailabilityGreetingList>
      <AvailabilityGreeting>
        <AvailGreetingID>23</AvailGreetingID>
        <MBID>5410</MBID>
        <Name>Out of the Office</Name>
        <SpeechCmdID>1417</SpeechCmdID>
        <TTSNameAlias />
        <SpokenNameFile />
      </AvailabilityGreeting>
    </AvailabilityGreetingList>
  </CXResponse>
</XMLResponse>
```

```

    </CXResponse>
</XMLResponse>
Errors
2147749804 Session is Invalid
The SessionID provided is invalid, the session has been previously logged off, or the
server has timed out of the session.
2147749315 Missing argument
Neither the AvailGreetingID or the MBID are specified. One is required.
Example
The following example returns the Availability Greeting information for Availability
Greeting ID 23 for mailbox 5410.
<CXAvailabilityGreetingListGet>
    <SessionID>519196341</SessionID>
    <SessionInstanceID>63b254ef-691a-47b2-b854-cb35c7cff52a</SessionInstanceID>
    <MBID>5410</MBID>

```

**<AvailGreetingID>23</AvailGreetingID>**

</CXAvailabilityGreetingListGet>

See Also

- [CXAvailabilityStateUpdate](#)

[CXAvailabilityActivate](#), [CXAvailabilityEnableSchedules](#), [CXAvailabilityStateDeviceListGet](#)

## CXAvailabilityStateDeviceListGet

Gets the Availability State Device List for the specified Availability State.

### Syntax

```

<CXAvailabilityStateDeviceListGet>
    <SessionID></SessionID>
    <SessionInstanceID></SessionInstanceID>
    <MBID></MBID>
    <AvailStateID></AvailStateID>
</CXAvailabilityStateDeviceListGet>

```

### Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

MBID

- The mailbox number to determine the Availability State device list.

AvailStateID

- The ID of the Availability State device list.

## Output Parameters

None

## XML Return

```
<XMLResponse xmlns="http://www.mitel.com/CXIf">
  <CXResponse>
    <AvailabilityStateDeviceList>
      <AvailabilityStateDevice>
        <AvailStateID>19738</AvailStateID>
        <Cardinal>1</Cardinal>
        <Device>
          <Name>Extension</Name>
          <MBID>5410</MBID>
          <RemoteMBNo />
          <IsPrimary>1</IsPrimary>
          <IsSenecaVirtual>0</IsSenecaVirtual>
          <Visibility>2</Visibility>
          <MwiCapable>0</MwiCapable>
          <WasPropagated>0</WasPropagated>
          <DevID>373</DevID>
          <DevAddr>5410</DevAddr>
          <DevTypeID>1</DevTypeID>
          <DevUsageType>Extension</DevUsageType>
          <DevAddrModifier />
          <DevExternalAddr />
          <SwitchSectionID>2</SwitchSectionID>
          <AllowMappingToMB>1</AllowMappingToMB>
          <IsOperator>0</IsOperator>
          <IsSMDI>0</IsSMDI>
          <IsDialable>1</IsDialable>
          <ConsoleDisplayName />
          <TrustedLogon>0</TrustedLogon>
          <RecognitionSensitivity>0</RecognitionSensitivity>
          <RingTimeoutSec>14</RingTimeoutSec>
          <Shared>0</Shared>
          <Enabled>1</Enabled>
          <IsOfficeDefault>1</IsOfficeDefault>
          <IsMobileDefault>0</IsMobileDefault>
          <MblAppNotificationEnabled>0</MblAppNotificationEnabled>
          <MblAppCallAlertType>None</MblAppCallAlertType>
          <MblAppNotifyExpirationSeconds>15</MblAppNotifyExpirationSeconds>
```

```

        <MblAppResponseWaitSeconds>60</MblAppResponseWaitSeconds>
        <MblAppDeviceToken />
        <MblAppPlatform />
        <MblAppPlatformVersion>1</MblAppPlatformVersion>
        <MblAppMfgModel />
        <MblAppCultureName>en-US</MblAppCultureName>
        <MblAppVersion />
        <MblAppInactive>0</MblAppInactive>
        <MblAppLogonTimestamp>0</MblAppLogonTimestamp>
        <DerivedDeviceTypeEnum>VoiceCallLogon</DerivedDeviceTypeEnum>
    </Device>
</AvailabilityStateDevice>
</AvailabilityStateDeviceList>
</CXResponse>
</XMLResponse>

```

## Example

The following example gets the device list for the active Availability State ID 19738 for mailbox 5410. Function **CXAvailabilityActiveStateGet** returns parameter **DeviceListAvailStateID**. The value of **DeviceListAvailStateID** should be used as the **AvailStateID** in this function.

**NOTE** **DeviceListAvailStateID** is provided for convenience. Since the Active Availability State has **UseParentDeviceList** set to 1, the **ParentAvailStateID** should be used instead of the **AvailStateID**. Therefore, 19738 is used instead of 33412 for the **AvailStateID** on the request. **DeviceListAvailStateID** is that value.

```

<CXAvailabilityStateDeviceListGet>
  <SessionID>94935008</SessionID>
  <SessionInstanceID>63b254ef-691a-47b2-b854-cb35c7cff52a</SessionInstanceID>
  <MBID>5410</MBID>
  <AvailStateID>19738</AvailStateID>
</CXAvailabilityStateDeviceListGet>

```

## Errors

2147749804    Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147750872    Invalid mailbox

- The MBID provided does not exist or is not a subscriber mailbox.

## See Also

[CXAvailabilityGreetingListGet](#)

Gets all Availability Greeting records or a specific Availability Greeting record for a Subscriber or Availability Class of Service mailbox.

## Syntax

```
<CXAvailabilityStateListGet>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <MBID></MBID>
  <AvailGreetingID></AvailGreetingID>
</CXAvailabilityStateListGet>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

MBID

- The mailbox number for which the Availability Greeting information is retrieved.

AvailGreetingID

- A specific Availability Greeting ID for which information is to be retrieved. If this parameter is not specified, information is retrieved for all Availability Greetings for the specified mailbox.

## Output Parameters

AvailabilityGreetingList

- Container for a list of one or more Availability Greeting configurations.
  - AvailabilityGreeting
    - Container for a single Availability Greeting configuration.

## XML Return

```
<XMLResponse xmlns="http://www.mitel.com/CXIf">
  <CXResponse>
    <AvailabilityGreetingList>
      <AvailabilityGreeting>
        <AvailGreetingID>23</AvailGreetingID>
        <MBID>5410</MBID>
        <Name>Out of the Office</Name>
        <SpeechCmdID>1417</SpeechCmdID>
      </AvailabilityGreeting>
    </AvailabilityGreetingList>
  </CXResponse>
</XMLResponse>
```

```

        <TTSNameAlias />
        <SpokenNameFile />
    </AvailabilityGreeting>
</AvailabilityGreetingList>
</CXResponse>
</XMLResponse>
Errors
2147749804 Session is Invalid
The SessionID provided is invalid, the session has been previously logged off, or the
server has timed out of the session.
2147749315 Missing argument
Neither the AvailGreetingID or the MBID are specified. One is required.
Example
The following example returns the Availability Greeting information for Availability
Greeting ID 23 for mailbox 5410.
<CXAvailabilityGreetingListGet>
    <SessionID>519196341</SessionID>
    <SessionInstanceID>63b254ef-691a-47b2-b854-cb35c7cff52a</SessionInstanceID>
    <MBID>5410</MBID>

```

**<AvailGreetingID>23</AvailGreetingID>**

</CXAvailabilityGreetingListGet>

See Also

- CXAvailabilityStateUpdate

CXAvailabilityActivate, [CXAvailabilityEnableSchedules](#), [CXAvailabilityActiveStateGet](#)

## CXAvailabilityAutoOverrideListGet

Gets all Availability Auto Override records or a specific Availability Auto Override record for a Subscriber or Availability Class of Service mailbox.

Subscriber mailbox Availability states reference Availability Class of Service automatic overrides since subscribers cannot have private Automatic Override records. Therefore, the Automatic Override records returned for a subscriber mailbox are actually part of the parent Availability Class of Service.

### Syntax

```

<CXAvailabilityAutoOverrideListGet>
    <SessionID></SessionID>
    <SessionInstanceID></SessionInstanceID>
    <MBID></MBID>
    <AvailAutoOverrideID></AvailAutoOverrideID>
</CXAvailabilityAutoOverrideListGet>

```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

MBID

- The Subscriber or Availability Class of Service mailbox number for which the Availability Auto Override information is retrieved.

AvailAutoOverrideID

- A specific Availability Auto Override ID for which information is to be retrieved. If this parameter is not specified, information is retrieved for all Availability Auto Override settings for the specified mailbox.

## Output Parameters

AvailabilityAutoOverrideList

- Container for a list of one or more Availability Auto Override configurations.
- AvailabilityAutoOverride
  - Container for a single Availability Auto Override configuration.

## XML Return

```
<XMLResponse xmlns="http://www.mitel.com/CXIf">
  <CXResponse>
    <AvailabilityAutoOverrideList>
      <AvailabilityAutoOverride>
        <AvailAutoOverrideID>295</AvailAutoOverrideID>
        <MBID>8001</MBID>
        <Priority>1</Priority>
        <Name>When Do Not Disturb</Name>
        <RuleType>AND</RuleType>
        <AvailAutoOverrideType>9</AvailAutoOverrideType>
        <TempBuildAvailAutoOverrideID>0</TempBuildAvailAutoOverrideID>
        <AvailAggregationRowArray>
          <AvailAggregationRow>
            <AvailAggregationRowID>1119</AvailAggregationRowID>
            <RuleType>OR</RuleType>
            <AvailInPointConnectionArray>
              <AvailInPointConnection>
                <AvailInPointType>105</AvailInPointType>
                <SourceEngineType>Presence</SourceEngineType>
                <ServerType>Presence</ServerType>
              </AvailInPointConnection>
            </AvailInPointConnectionArray>
          </AvailAggregationRow>
        </AvailAggregationRowArray>
      </AvailabilityAutoOverride>
    </AvailabilityAutoOverrideList>
  </CXResponse>
</XMLResponse>
```

```

        <Negate>0</Negate>
        <RequiredSource>0</RequiredSource>
    </AvailInPointConnection>
    <AvailInPointConnection>
        <AvailInPointType>302</AvailInPointType>
        <SourceEngineType>Time</SourceEngineType>
        <ServerType>CXWorkHours</ServerType>
        <Negate>1</Negate>
        <RequiredSource>0</RequiredSource>
    </AvailInPointConnection>
    <AvailInPointConnection>
        <AvailInPointType>402</AvailInPointType>
        <SourceEngineType>Setting</SourceEngineType>
        <ServerType>CXSettings</ServerType>
        <Negate>0</Negate>
        <RequiredSource>0</RequiredSource>
    </AvailInPointConnection>
</AvailInPointConnectionArray>
</AvailAggregationRow>
</AvailAggregationRowArray>
</AvailabilityAutoOverride>
</CXResponse>
</XMLResponse>

```

## Errors

2147749804    Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147749315    Missing argument

- Neither the AvailAutoOverrideID or the MBID are specified. One is required.

## Example

The following example returns the Availability Auto Override information for Availability Auto Override ID 295 for mailbox 5410. Even though requested for MBID 5410, the returned record is actually an Automatic Override record from the parent Availability Class of Service mailbox 8001.

```

<CXAvailabilityAutoOverrideListGet>
    <SessionID>519196341</SessionID>
    <SessionInstanceID>63b254ef-691a-47b2-b854-cb35c7cff52a</SessionInstanceID>
    <MBID>5410</MBID>
    <AvailAutoOverrideID>295</AvailAutoOverrideID>
</CXAvailabilityAutoOverrideListGet>

```



## See Also

[CXAavailabilityStateListGet](#), [CXAavailabilityStateUpdate](#)

## CXAavailabilityDayListGet

Gets all Availability Day records or a specific Availability Day record for a Subscriber or Availability Class of Service mailbox.

## Syntax

```
<CXAavailabilityDayListGet>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <MBID></MBID>
  <AvailDayID></AvailDayID>
</CXAavailabilityDayListGet>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

MBID

- The mailbox number for which the Availability Day information is retrieved. This parameter is optional if **AvailDayID** is specified.

AvailDayID

- A specific Availability Day ID for which information is to be retrieved. If this parameter is not specified, information is retrieved for all Availability Day records for the specified mailbox.

## Output Parameters

AvailabilityDayList

- Container for a list of one or more Availability Day configurations.
  - AvailabilityDay
    - Container for a single Availability Day configuration.

## XML Return

```
<XMLResponse xmlns="http://www.mitel.com/CXIf">
  <CXResponse>
    <AvailabilityDayList>
      <AvailabilityDay>
        <AvailDayID>27541</AvailDayID>
        <MBID>5410</MBID>
        <Name>Monday</Name>
        <SpeechCmdID>0</SpeechCmdID>
        <Priority>1</Priority>
        <ActivationTimestamp>1970-01-01T00:00:00Z</ActivationTimestamp>
        <ExpirationTimestamp>1970-01-01T00:00:00Z</ExpirationTimestamp>
        <Enabled>1</Enabled>
        <DayIndex>2</DayIndex>
        <TemporaryOverride>0</TemporaryOverride>
        <TempBuildAvailDayID>0</TempBuildAvailDayID>
        <MBAvailTimeArray>
          <MBAvailTime>
            <Cardinal>1</Cardinal>
            <StartTOD>800</StartTOD>
            <StopTOD>1200</StopTOD>
            <AvailStateID>33412</AvailStateID>
            <IsWorkTime>1</IsWorkTime>
          </MBAvailTime>
          <MBAvailTime>
            <Cardinal>2</Cardinal>
            <StartTOD>1200</StartTOD>
            <StopTOD>1300</StopTOD>
            <AvailStateID>33419</AvailStateID>
            <IsWorkTime>0</IsWorkTime>
          </MBAvailTime>
          <MBAvailTime>
            <Cardinal>3</Cardinal>
            <StartTOD>1300</StartTOD>
            <StopTOD>1700</StopTOD>
            <AvailStateID>33412</AvailStateID>
            <IsWorkTime>1</IsWorkTime>
          </MBAvailTime>
        </MBAvailTimeArray>
      </AvailabilityDay>
    </AvailabilityDayList>
  </CXResponse>
</XMLResponse>
```

## Errors

2147749804    Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147749315      Missing argument

- Neither the AvailDayID or the MBID are specified. One is required.

## Example

The following example returns the Availability Day information (above) for Availability Day ID 27541 for mailbox 5410. The MBID is optional in this case.

```
<CXAvailabilityDayListGet>
  <SessionID>519196341</SessionID>
  <SessionInstanceID>63b254ef-691a-47b2-b854-cb35c7cff52a</SessionInstanceID>
  <MBID>5410</MBID>
  <AvailDayID>27541</AvailDayID>
</CXAvailabilityDayListGet>
```

## See Also

[CXAvailabilityDayAdd](#), [CXAvailabilityDayUpdate](#), [CXAvailabilityDayDelete](#), [CXAvailabilityStateListGet](#), [CXAvailabilityStateUpdate](#), [CXAvailabilityAutoOverrideListGet](#)

## CXAvailabilityDayAdd

Adds a new Availability Day record for an Availability Class of Service mailbox.

This function should only be used under advanced configuration situations since it is intended only for low level, primitive Day schedule additions for Availability Class of Service mailboxes.

Function **CXAvailabilitySubscriberReset** should be used for adding initial Day schedules to a subscriber based on an Availability Class of Service mailbox.

## Syntax

```
<CXAvailabilityDayAdd>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <AvailabilityDay>
  </AvailabilityDay>
</CXAvailabilityDayAdd>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

AvailabilityDay

- Container for a single Availability Day configuration.

## Output Parameters

AvailDayID

- The Availability Day ID for the added record.

## XML Return

```
<XMLResponse xmlns="http://www.mitel.com/CXIf">
  <CXResponse>
    <AvailDayID>27547</AvailDayID>
  </CXResponse>
</XMLResponse>
```

## Errors

2147749804    Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147749315    Missing argument

- The AvailabilityDay record is not specified.

## Example

The following adds an Availability Day record for mailbox 8001. The value of the AvailDayID is ignored and can be omitted. If **IsWorkTime** is not specified in an **MBAvailTime** record, it is assumed to be equal to 0.

```
<CXAvailabilityDayAdd>
  <SessionID>519196341</SessionID>
  <SessionInstanceID>63b254ef-691a-47b2-b854-cb35c7cff52a</SessionInstanceID>
  <AvailabilityDay>
    <AvailDayID>0</AvailDayID>
    <MBID>8001</MBID>
    <Name>Work Day</Name>
    <SpeechCmdID>1221</SpeechCmdID>
    <Priority>2</Priority>
    <ActivationTimestamp>1970-01-01T00:00:00Z</ActivationTimestamp>
    <ExpirationTimestamp>1970-01-01T00:00:00Z</ExpirationTimestamp>
    <Enabled>0</Enabled>
    <DayIndex>0</DayIndex>
    <TemporaryOverride>0</TemporaryOverride>
```

```

    <TempBuildAvailDayID>0</TempBuildAvailDayID>
    <MBAvailTimeArray>
      <MBAvailTime>
        <Cardinal>1</Cardinal>
        <StartTOD>800</StartTOD>
        <StopTOD>1200</StopTOD>
        <AvailStateID>33412</AvailStateID>
        <IsWorkTime>1</IsWorkTime>
      </MBAvailTime>
      <MBAvailTime>
        <Cardinal>2</Cardinal>
        <StartTOD>1200</StartTOD>
        <StopTOD>1300</StopTOD>
        <AvailStateID>33419</AvailStateID>
        <IsWorkTime>0</IsWorkTime>
      </MBAvailTime>
      <MBAvailTime>
        <Cardinal>3</Cardinal>
        <StartTOD>1300</StartTOD>
        <StopTOD>1700</StopTOD>
        <AvailStateID>33412</AvailStateID>
        <IsWorkTime>1</IsWorkTime>
      </MBAvailTime>
    </MBAvailTimeArray>
  </AvailabilityDay>
</CXAvailabilityDayAdd>

```

## See Also

[CXAvailabilityDayListGet](#), [CXAvailabilityDayUpdate](#), [CXAvailabilityDayDelete](#), [CXAvailabilityStateListGet](#), [CXAvailabilityStateUpdate](#), [CXAvailabilityAutoOverrideListGet](#)

## CXAvailabilityDayUpdate

Updates an existing Availability Day record for an Availability Class of Service or Subscriber mailbox.

## Syntax

```

<CXAvailabilityDayUpdate>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <ScheduleUpdateMode></ScheduleUpdateMode>
  <AvailabilityDay>
  </AvailabilityDay>
  <OldObject>
    <AvailabilityDay>
    </AvailabilityDay>
  </OldObject>

```

```
</CXAvailabilityDayUpdate>
```

## Input Parameters

### SessionID

- Identifies the session. This value is returned by a previous call to CXLogon.

### SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to CXLogon.

### ScheduleUpdateMode

- Optional. The Day Schedule Update Mode. If not specified, a value of "StateOnly" is assumed. It can be one of the following values:
- "StateOnly" – Use only the **AvailStateID** values specified in the **MBAvailTime** containers. Current **IsWorkTime** values are retained.
- "WorkTimeOnly" – Use only the **IsWorkTime** values specified in the **MBAvailTime** containers. Current **AvailStateID** values are retained.
- "All" – Uses both the **IsWorkTime** and the **AvailStateID** values specified in the **MBAvailTime** containers.

### AvailabilityDay

- Container for the updated Availability Day configuration.

### OldObject

- This field represents the baseline from which the update is being made. The difference between this AvailabilityDay and the updated AvailabilityDay determines the updates that are applied.
- AvailabilityDay
  - Container for the original Availability Day configuration.

## Output Parameters

None

## Errors

### 2147749804 Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

### 2147749315 Missing argument

- Either the updated AvailabilityDay record or the original AvailabilityDay record in the OldObject is not specified. Both are required.

## Example

The following updates Availability Day ID 27547 record for mailbox 8001.

```
<CXAvailabilityDayUpdate>
  <SessionID>519196341</SessionID>
  <SessionInstanceID>63b254ef-691a-47b2-b854-cb35c7cff52a</SessionInstanceID>
  <ScheduleUpdateMode>All</ScheduleUpdateMode>
  <AvailabilityDay>
    <AvailDayID>27547</AvailDayID>
    <MBID>8001</MBID>
    <Name>Work Day</Name>
    <SpeechCmdID>1221</SpeechCmdID>
    <Priority>2</Priority>
    <ActivationTimestamp>1970-01-01T00:00:00Z</ActivationTimestamp>
    <ExpirationTimestamp>1970-01-01T00:00:00Z</ExpirationTimestamp>
    <Enabled>0</Enabled>
    <DayIndex>0</DayIndex>
    <TemporaryOverride>0</TemporaryOverride>
    <TempBuildAvailDayID>0</TempBuildAvailDayID>
    <MBAvailTimeArray>
      <MBAvailTime>
        <Cardinal>1</Cardinal>
        <StartTOD>800</StartTOD>
        <StopTOD>1100</StopTOD>
        <AvailStateID>33412</AvailStateID>
        <IsWorkTime>1</IsWorkTime>
      </MBAvailTime>
      <MBAvailTime>
        <Cardinal>2</Cardinal>
        <StartTOD>1100</StartTOD>
        <StopTOD>1200</StopTOD>
        <AvailStateID>33419</AvailStateID>
        <IsWorkTime>0</IsWorkTime>
      </MBAvailTime>
      <MBAvailTime>
        <Cardinal>3</Cardinal>
        <StartTOD>1200</StartTOD>
        <StopTOD>1700</StopTOD>
        <AvailStateID>33412</AvailStateID>
        <IsWorkTime>1</IsWorkTime>
      </MBAvailTime>
    </MBAvailTimeArray>
  </AvailabilityDay>
  <OldObject>
    <AvailabilityDay>
      <AvailDayID>27547</AvailDayID>
      <MBID>8001</MBID>
      <Name>Work Day</Name>
      <SpeechCmdID>1221</SpeechCmdID>
      <Priority>2</Priority>
      <ActivationTimestamp>1970-01-01T00:00:00Z</ActivationTimestamp>
      <ExpirationTimestamp>1970-01-01T00:00:00Z</ExpirationTimestamp>
```

```

    <Enabled>0</Enabled>
    <DayIndex>0</DayIndex>
    <TemporaryOverride>0</TemporaryOverride>
    <TempBuildAvailDayID>0</TempBuildAvailDayID>
    <MBAvailTimeArray>
      <MBAvailTime>
        <Cardinal>1</Cardinal>
        <StartTOD>800</StartTOD>
        <StopTOD>1200</StopTOD>
        <AvailStateID>33412</AvailStateID>
        <IsWorkTime>1</IsWorkTime>
      </MBAvailTime>
      <MBAvailTime>
        <Cardinal>2</Cardinal>
        <StartTOD>1200</StartTOD>
        <StopTOD>1300</StopTOD>
        <AvailStateID>33419</AvailStateID>
        <IsWorkTime>0</IsWorkTime>
      </MBAvailTime>
      <MBAvailTime>
        <Cardinal>3</Cardinal>
        <StartTOD>1300</StartTOD>
        <StopTOD>1700</StopTOD>
        <AvailStateID>33412</AvailStateID>
        <IsWorkTime>1</IsWorkTime>
      </MBAvailTime>
    </MBAvailTimeArray>
  </AvailabilityDay>
</OldObject>
</CXAvailabilityDayUpdate>

```

## See Also

[CXAvailabilityDayListGet](#), [CXAvailabilityDayAdd](#), [CXAvailabilityDayDelete](#), [CXAvailabilityStateListGet](#), [CXAvailabilityStateUpdate](#), [CXAvailabilityAutoOverrideListGet](#)

## CXAvailabilityDayDelete

Deletes an Availability Day record for an Availability Class of Service mailbox.

This function should only be used under advanced configuration situations since it is intended only for low level, primitive Day schedule deletions for Availability Class of Service Mailboxes. Day of the Week records should never be deleted unless the entire Availability Class of Service is being removed.

## Syntax

```

<CXAvailabilityDayDelete>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>

```



```
<AvailDayID></AvailDayID>
</CXAvailabilityDayDelete>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to CXLogon.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to CXLogon.

AvailDayID

- A specific Availability Day ID for the record to be deleted.

## Output Parameters

None

## Errors

2147749804     Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147749315     Missing argument

- The AvailDayID is not specified.

## Example

The following example deletes the Availability Day information for Availability Day ID 27547 for mailbox 5410.

```
<CXAvailabilityDayDelete>
  <SessionID>519196341</SessionID>
  <SessionInstanceID>63b254ef-691a-47b2-b854-cb35c7cff52a</SessionInstanceID>
  <AvailDayID>27547</AvailDayID>
</CXAvailabilityDayDelete>
```

## See Also

[CXAvailabilityDayListGet](#), [CXAvailabilityDayAdd](#), [CXAvailabilityDayUpdate](#), [CXAvailabilityStateListGet](#), [CXAvailabilityStateUpdate](#), [CXAvailabilityAutoOverrideListGet](#)

# CXAvailabilityStateListGet

Gets all Availability State records or a specific Availability State record for a Subscriber or Availability Class of Service mailbox.

## Syntax

```
<CXAvailabilityStateListGet>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <MBID></MBID>
  <AvailStateID></AvailStateID>
  <IncludeEmptyDevices></IncludeEmptyDevices>
</CXAvailabilityStateListGet>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

MBID

- The mailbox number for which the Availability State information is retrieved.

AvailStateID

- A specific Availability State ID for which information is to be retrieved. If this parameter is not specified, information is retrieved for all Availability State settings for the specified mailbox.

IncludeEmptyDevices

- A flag used to determine whether to return place holder MBAvailDevice items in the MBAvailDeviceArray for devices that do not exist for the subscriber but are referenced by the subscriber's Availability COS. By definition, since an Availability COS has no actual devices, this flag must be set in order to list the devices associated with the Availability COS States. The returned data for Subscriber Availability States that not have the UseParentDeviceList set to 1 are not affected by this flag.

## Output Parameters

AvailabilityStateList

- Container for a list of one or more Availability State configurations.
- AvailabilityState
  - Container for a single Availability State configuration.

## XML Return

```
<XMLResponse xmlns="http://www.mitel.com/CXIf">
  <CXResponse>
    <AvailabilityStateList>
      <AvailabilityState>
        <AvailStateID>208</AvailStateID>
        <MBID>6000</MBID>
        <Name>Working from Home</Name>
        <ParentAvailStateID>169</ParentAvailStateID>
        <UseParentDeviceList>1</UseParentDeviceList>
        <Locked>1</Locked>
        <AvailAutoOverrideID>13</AvailAutoOverrideID>
        <Hidden>0</Hidden>
        <PlayGeneralGreeting>1</PlayGeneralGreeting>
        <AnnounceEnable>0</AnnounceEnable>
        <AnnounceFirst>0</AnnounceFirst>
        <AnnouncePrivacyOptions>NameOnly</AnnouncePrivacyOptions>
        <AvailAnnouncementID>15</AvailAnnouncementID>
        <AnnounceReturnEnable>0</AnnounceReturnEnable>
        <AnnounceReturnDate>0</AnnounceReturnDate>
        <AnnounceReturnTime>0</AnnounceReturnTime>
        <ReturnTimeAvailable>1</ReturnTimeAvailable>

      <AnnounceDefaultGreetingType>StandardOrBusy</AnnounceDefaultGreetingType>
        <AvailGreetingID>0</AvailGreetingID>
        <AnnounceCustomAllowed>1</AnnounceCustomAllowed>
        <PlayGreetingBeforeRouting>0</PlayGreetingBeforeRouting>
        <LocateMode>ConfirmOnce</LocateMode>
        <FinalRoutingOptionEnable>1</FinalRoutingOptionEnable>
        <TimeoutAction>TakeMessage</TimeoutAction>
        <TimeoutRouteMBID/>
        <DoNotAcceptMessages>0</DoNotAcceptMessages>
        <SuppressNotAcceptMsgPrompt>0</SuppressNotAcceptMsgPrompt>
        <AutoLocateEnable>0</AutoLocateEnable>
        <AutoLocateLocateFirst>0</AutoLocateLocateFirst>
        <MobileNotificationAllow>1</MobileNotificationAllow>
        <CallScreeningEnable>1</CallScreeningEnable>

        <ContinueCallAfterDisconnect>0</ContinueCallAfterDisconnect>
        <DefaultBehavior>Indefinite</DefaultBehavior>
        <DefaultStopTOD>0</DefaultStopTOD>
        <DefaultDurationSeconds>3600</DefaultDurationSeconds>
        <SpeechCmdID>10027</SpeechCmdID>
        <TTSNameAlias/>
        <SpokenNameFile>en-US_AVLN_WorkingFromHome</SpokenNameFile>
        <TempBuildAvailStateID>169</TempBuildAvailStateID>
        <MBAvailDeviceArray>
          <MBAvailDevice>
            <Cardinal>1</Cardinal>
            <RecordType>1</RecordType>
            <DevUsageTypeID>1</DevUsageTypeID>
            <DevUsageType>CompanyMobile</DevUsageType>
          </MBAvailDevice>
        </MBAvailDeviceArray>
      </AvailabilityState>
    </AvailabilityStateList>
  </CXResponse>
</XMLResponse>
```

```

        <DeviceID>553</DeviceID>
      </MBAvailDevice>
    </MBAvailDeviceArray>
  </AvailabilityState>
</AvailabilityStateList>
</CXResponse>
</XMLResponse>

```

## Errors

2147749804    Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147749315    Missing argument

- Neither the AvailStateID or the MBID are specified. One is required.

## Example

The following example returns the Availability State information for Availability State ID 196 for mailbox 5410.

```

<CXAvailabilityStateListGet>
  <SessionID>519196341</SessionID>
  <SessionInstanceID>63b254ef-691a-47b2-b854-cb35c7cff52a</SessionInstanceID>
  <MBID>5410</MBID>
  <AvailStateID>33412</AvailStateID>
  <IncludeEmptyDevices>1</IncludeEmptyDevices>
</CXAvailabilityStateListGet>

```

## See Also

[CXAvailabilityDayListGet](#), [CXAvailabilityDayAdd](#), [CXAvailabilityDayUpdate](#), [CXAvailabilityDayDelete](#), [CXAvailabilityStateUpdate](#), [CXAvailabilityAutoOverrideListGet](#)

## CXAvailabilityStateUpdate

Updates an existing Availability State record for an Availability Class of Service or Subscriber mailbox.

Function **CXAvailabilitySubscriberStateUpdate** can be used to update Subscriber mailbox states with Availability Class of Service state changes. Function **CXAvailabilitySubscriberStateRefresh** can be used to restore Subscriber mailbox states to the current Availability Class of Service state settings.

## Syntax

```

<CXAvailabilityStateUpdate>

```

```

<SessionID></SessionID>
<SessionInstanceID></SessionInstanceID>
<AvailabilityState>
</AvailabilityState>
<OldObject>
  <AvailabilityState>
  </AvailabilityState>
</OldObject>
</CXAvailabilityStateUpdate>

```

## Input Parameters

### SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

### SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

### AvailabilityState

- Container for the updated Availability State configuration.

### OldObject

- This field represents the baseline from which the update is being made. The difference between this AvailabilityState and the updated AvailabilityState determines the updates that are applied.
- AvailabilityState
  - Container for the original Availability State configuration.

## Output Parameters

None

## Errors

2147749804    Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147749315    Missing argument

- Either the updated AvailabilityState record or the original AvailabilityState record in the OldObject is not specified. Both are required.

## Example

The following updates Availability State ID 208 record for mailbox 6000.

```

<CXAvailabilityStateUpdate>
  <SessionID>519196341</SessionID>
  <SessionInstanceID>63b254ef-691a-47b2-b854-cb35c7cff52a</SessionInstanceID>
  <AvailabilityState>
    <AvailStateID>208</AvailStateID>
    <MBID>6000</MBID>
    ...
  </AvailabilityState>
  <OldObject>
    <AvailabilityState>
      <AvailStateID>208</AvailStateID>
      <MBID>6000</MBID>
      ...
    </AvailabilityState>
  </OldObject>
</CXAvailabilityStateUpdate>

```

## See Also

[CXAvailabilityDayListGet](#), [CXAvailabilityDayAdd](#), [CXAvailabilityDayUpdate](#), [CXAvailabilityDayDelete](#), [CXAvailabilityStateListGet](#), [CXAvailabilityAutoOverrideListGet](#)

## CXAvailabilitySubscriberReset

Resets a list of subscriber mailboxes to use the current Availability settings of the specified Availability Class of Service mailbox's states and schedules. All the current Availability settings of each subscriber are discarded. Replicas of all the Availability Class of Service mailbox's states and schedules are generated for each subscriber. If not locked to the Availability Class of Service state, the Availability state's find-me device list is generated.

**CXAvailabilitySubscriberReset** does not modify the subscriber work hours. Use **CXMBUpdate** to change the **StartWorkTOD** and **StopWorkTOD** to the desired values.

**CXAvailabilitySubscriberReset** does not assign the subscriber to the **COSMBID** Availability Class of Service. Use **CXMBUpdate** to change the **ProfileAutoSourceMBID** to the **COSMBID** value.

## Syntax

```

<CXAvailabilitySubscriberReset>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <COSMBID></COSMBID>
  <MBIDList>
    <MBID></MBID>
    ...
    <MBID></MBID>
  </MBIDList>
</CXAvailabilitySubscriberReset>

```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

COSMBID

- The Availability Class of Service mailbox number.

MBIDList

- MBID list of subscriber member mailboxes requiring a reset. If not specified, all members of the Availability Class of Service are reset.

## Output Parameters

None

## Errors

2147749804      Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147749315      Missing argument

- Either the **COSMBID** or the **MBIDList** are not specified. Both are required.

## Example

The following example resets the Availability state and schedule settings to replicate the Availability Class of Service mailbox 8001 settings for mailbox 5410.

```
<CXAvailabilitySubscriberReset>
  <SessionID>519196341</SessionID>
  <SessionInstanceID>63b254ef-691a-47b2-b854-cb35c7cff52a</SessionInstanceID>
  <COSMBID>8001</COSMBID>
  <MBIDList>
    <MBID>5410</MBID>
  </MBIDList>
</CXAvailabilitySubscriberReset>
```

# CXObjectCreate

Creates a new recording object and returns the ID. **CXObjectOpen** or **CXObjectCreate** must be called before any other **CXObject** operations are called on the object.

## Syntax

```
<CXObjectCreate>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
</CXObjectCreate>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

## Output Parameters

ObjectID

- The unique identifier for the object.

## XML Return

```
<ObjectID>ab52ed7b-87c1-49a8-8f2a-a85578365ffc</ObjectID>
```

## Comments

Objects are created in a session-specific temporary storage folder on the server. Objects need to be closed using **CXObjectClose**. If this call is not used to close the objects then the objects will get cleaned up whenever the session is logged off.

## See Also

[CXObjectOpen](#), [CXObjectClose](#), [Recording Object Functions](#)



# CXObjectOpen

Opens an existing recording object with the specified type and ID. **CXObjectOpen** or **CXObjectCreate** must be called before any other **CXObject** operations are called on the object.

## Syntax

```
<CXObjectOpen>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <ObjectID></ObjectID>
  <RecordingTypeID></RecordingTypeID>
  <RecordingTypeSpecificID></RecordingTypeSpecificID>
</CXObjectOpen>
```

## Input Parameters

### SessionID

- Identifies the session. This value is returned by a previous call to CXLogon.

### SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to CXLogon.

### ObjectID

- The unique identifier for the object. A valid ObjectID obtained from the following fields can be specified here:
  - The MBCommon/NameMediaFileID field for the recorded name of a mailbox.
  - The PromptRecording/RecordingTypeID field for a Normal, Busy or Out-of-office greeting.
  - The Attachment/FileName field for a message attachment.

### RecordingTypeID

- The type of the Recording. Valid values are:
  - *Busy* – This is the busy greeting. Note that for a Call Processor mailbox, this recording type represents an Instruction Announcement.
  - *OOF* - This is the Out-of-office greeting.
  - *Name* - This is the recorded Name.
  - *Greeting* - This is the normal greeting.
  - *Message* - This is a message attachment.
  - *Introduction* - This is an Intro Announcement for a Call Processor mailbox.
  - *Availability* - This is the Availability set of greetings.

## Output Parameters

None

## Errors

2147751612     Local file copy error

- The object file could not be copied into the temporary folder. Likely cause of this error is that the source object ID provided is wrong.

## Comments

Opened objects are stored in a session-specific temporary storage folder on the server. Objects need to be closed using **CXObjectClose**. If this call is not used to close the objects then the objects will get cleaned up whenever the session is logged off.

## Example

The following example opens a greeting object.

```
<CXObjectOpen>
  <SessionID>1736771367</SessionID>
  <SessionInstanceID>2a6ec288-eda5-4327-8db0-638aec22dca2</SessionInstanceID>
  <ObjectID>f15612b7-327b-4edf-af30-cd8a2fc4a068</ObjectID>
  <RecordingTypeID>Greeting</RecordingTypeID>
</CXObjectOpen>
```

## See Also

[CXObjectCreate](#), [CXObjectClose](#), [Recording Object Functions](#)

## CXObjectClose

Closes the specified object. A RecordingTypeID parameter needs to be specified if the object needs to be saved permanently.

## Syntax

```
<CXObjectClose>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <ObjectID></ObjectID>
  <RecordingTypeID></RecordingTypeID>
</CXObjectClose>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

ObjectID

- The identifier of the object that was previously created or opened by calling **CXObjectCreate** or **CXObjectOpen**.

RecordingTypeID

- The type of the Recording that this object should be saved as. This can be left empty/missing to indicate that the server should clean up the object without saving it permanently. Valid values for saving the object are:
  - *Busy* – This is the busy greeting. Note that for a Call Processor mailbox, this recording type represents an Instruction Announcement.
  - *OOF* - This is the Out-of-office greeting.
  - *Name* - This is the recorded Name.
  - *Greeting* - This is the normal greeting.
  - *Message* - This is a message attachment.
  - *Introduction* - This is an Intro Announcement for a Call Processor mailbox.
  - *Availability* - This is the Availability set of greetings.

## Output Parameters

None

## Comments

Objects created by calling **CXObjectCreate** may need to be saved to the permanent storage. This can be done by calling **CXObjectClose** with a valid RecordingTypeID parameter.

Objects opened by calling **CXObjectOpen** already exist within the permanent storage. Thus they need not be saved. **CXObjectClose** can be called for these objects with a missing or empty RecordingTypeID parameter.

## Errors

2147751612     Local file copy error

- The object file could not be copied into the permanent storage. Possible causes are: wrong source object ID or an object which already exists in the permanent storage.

## Example

The following example closes the greeting object that was previously opened by calling CXObjectOpen.

```
<CXObjectClose>
  <SessionID>1736771367</SessionID>
  <SessionInstanceID>2a6ec288-eda5-4327-8db0-638aec22dca2</SessionInstanceID>
  <ObjectID>f15612b7-327b-4edf-af30-cd8a2fc4a068</ObjectID>
  <RecordingTypeID></RecordingTypeID>
</CXObjectClose>
```

## See Also

[CXObjectCreate](#), [CXObjectOpen](#), [Recording Object Functions](#)

## CXObjectUpload

Uploads the specified binary content into the recording object.

## Syntax

```
<CXObjectUpload>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <ObjectID></ObjectID>
  <FileData></FileData>
  <StartPosByte></StartPosByte>
</CXObjectUpload>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

ObjectID

- The identifier of the object that was previously created by calling **CXObjectCreate**.

FileData

- Specifies the binary data in the base64-encoded format that needs to be uploaded into the object file. The binary data, without the base64-encoding, is persisted into the object file.

StartPosByte

- Optional. The zero-based byte offset within the object's file at which to start adding the new data. If not specified, this is assumed to be 0 meaning the beginning of the file.

## Output Parameters

None

## Example

The following example uploads binary data into the object file by specifying the base-64 encoded format of that data in the **FileData** parameter. Only a portion of the content in the FileData parameter has been shown in the example.

```
<CXObjectUpload>
  <SessionID>1736771367</SessionID>
  <SessionInstanceID>2a6ec288-eda5-4327-8db0-638aec22dca2</SessionInstanceID>
  <ObjectID>ab52ed7b-87c1-49a8-8f2a-a85578365ffc</ObjectID>
  <FileData>UklGRvJYAABXQVZFZm1...+f3///79/f4=</FileData>
</CXObjectUpload>
```

## See Also

[CXObjectCreate](#), [CXObjectDownload](#), [Recording Object Functions](#),

## CXObjectDownload

Downloads the binary content in the specified recording object.

## Syntax

```
<CXObjectDownload>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <ObjectID></ObjectID>
  <StartPosByte></StartPosByte>
  <LengthBytes></LengthBytes>
</CXObjectDownload>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

ObjectID

- The identifier of the object that was previously created or opened by calling **CXObjectCreate** or **CXObjectOpen**.

StartPosByte

- Optional. The zero-based byte offset within the object's file from which to start fetching the data. If not specified, this is assumed to be 0 meaning the beginning of the file.

LengthBytes

- Optional. The count of the bytes to download. If not specified, this is assumed to be -1 meaning that the data till the end of the file needs to be retrieved.

## Output Parameters

FileData

- This is the binary data in the file in the base64-encoded format.

## XML Return

Only a portion of the content in the FileData parameter has been shown below.

```
<FileData>UklGRvJYAABXQVZFZm1...+f3///79/f4=</FileData>
```

## Example

The following example downloads the binary data within an object file.

```
<CXObjectDownload>
  <SessionID>1736771367</SessionID>
  <SessionInstanceID>2a6ec288-eda5-4327-8db0-638aec22dca2</SessionInstanceID>
  <ObjectID>f15612b7-327b-4edf-af30-cd8a2fc4a068</ObjectID>
</CXObjectDownload>
```

## See Also

[CXObjectCreate](#), [CXObjectOpen](#), [CXObjectUpload](#), [Recording Object Functions](#)

## CXMessageSearch

This function can be used to search for messages meeting specified criteria and to retrieve the few most-important fields about them.

## Syntax

```
<CXMessageSearch>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <StoreType>Local</StoreType>
  <Folder>New</Folder>
  <Style>LIFO</Style>
  <Count>100</Count>
  <StartTimestamp>2011-01-01T00:00:00Z</StartTimestamp>
  <EndTimestamp>2012-01-01T00:00:00Z</EndTimestamp>
  <StartChangeNumber>0</StartChangeNumber>
  <IncludeDeleted>0</IncludeDeleted>      <MsgFields>
    <MsgID/>
    <PermanentMsgID/>
    <StoreType/>
    <Type/>
    <Read/>
    <SoftDeleted/>
    <Deleted/>      <Priority/>
    <DeliveryTimestamp/>
    <ChangeNumber/>      <ChangedTimestamp/>
    <SenderMBID/>
    <SenderRemoteMBID/>
    <SenderName/>
  </MsgFields>
  <MsgFilters>
    <Read></Read>
    <Priority></Priority>
    <VoiceMsgSubTypes>
      <VoiceMsgSubType>Normal</VoiceMsgSubType>
      <VoiceMsgSubType>MissedCall</VoiceMsgSubType>
    </VoiceMsgSubTypes>
    <Types>
      <Type>Voice</Type>
      <Type>Fax</Type>
    </Types>
  <Sender>
    <MBID></MBID>
    <RemoteMBID></RemoteMBID>
    <EmailAddress></EmailAddress>
  </Sender>
</MsgFilters>
</CXMessageSearch>
```

## Input Parameters

### SessionID

- Identifies the session. This value is returned by a previous call to CXLogin.

#### SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to CXLogon.

#### StoreType

- The type of store.
  - *Local* – Local store.
  - *External* – An external store such as Exchange/Domino/IMAP UM

#### Folder

- The folder from which to retrieve.
  - *New* – The messages are from the New folder
  - *Saved* – The messages are from the Saved folder

#### Style

- Sort order of the results.
  - *LIFO* – Last In First Out
  - *FIFO* – First In First Out

#### Count

- Maximum number of messages to retrieve. If the maximum number is returned, adjust StartTimestamp and EndTimestamp to determine whether additional messages are available.

#### StartTimestamp

- This is the starting time filter in UTC. This follows the xml dateTime format. Only those messages that were delivered since the StartTimestamp (including those delivered at this time boundary) will be retrieved.

#### EndTimestamp

- This is the ending time filter in UTC. This follows the xml dateTime format. Only those messages that were delivered before the EndTimestamp (not including this time boundary) will be retrieved.

#### StartChangeNumber

- This parameter allows retrieving messages that have changed since a particular *Start Change Number*. All messages will be retrieved when this parameter is either unspecified or set to 0.

#### IncludeDeleted

- This parameter only applies if the StartChangeNumber parameter has been specified and set to a non-zero value. If so, and provided that this parameter is set to the value 1, messages that have been deleted will also be retrieved in addition to messages that haven't been deleted. This allows a client to discover message deletions as a change and accordingly update its cache.

#### MsgFields

- This structure specifies the list of fields to be returned. Possible values are:
  - <MsgID/>



- <PermanentMsgID/>
- <StoreType/>
- <Type/>
- <Read/>
- <SoftDeleted/>
- <Deleted/>
- <Priority/>
- <DeliveryTimestamp/>
- <ChangeNumber/>
- <ChangedTimestamp/>
- <SenderMBID/>
- <SenderRemoteMBID/>
- <SenderName/>

#### MsgFilters

- This structure specifies the list of filters to be applied to the search. Multiple filter items can be specified. Only messages that meet all the filter criteria will be retrieved. If a field is not specified (by leaving the field out of the xml) then that field will not be used to perform the search. Thus, if the Read field is unspecified, then all messages (both Read and Unread) are returned.
- Read
  - Specify 0 for unread messages, 1 for read messages.
- SoftDeleted
  - Specify 0 for messages marked for deletion, 1 for messages not marked for deletion.
- Deleted
  - Specify 0 for messages that have been deleted, 1 for messages that have not been deleted. This filter can only be specified if the **StartChangeNumber** parameter is a non-zero value and the **IncludeDeleted** parameter is set to the value 1.
- Priority
  - Specify *N* for normal priority messages, *U* for urgent priority messages.
- Types
  - Container for one or more message type (Type) specifications.
- Type
  - May be *Voice*, *Fax*, *Email* or *Calendar*.
- VoiceMsgSubTypes
  - Container for one or more voice message sub type (VoiceMsgSubType) specifications.
- VoiceMsgSubType

- May be *Normal*, *MissedCall*, *Acknowledgement* or *RecordedConversation*.
- Sender
  - Structure for specification of a specific sender.
- MBID
  - Mailbox number of the sender.
- RemoteMBID
  - Remote mailbox number of the sender.
- EmailAddress
  - Email address of the sender.

## Output Parameters

### MsgFieldsList

- The top-level container for the list of message fields (MsgFields).
- MsgFields
  - The structure with fields that are retrieved for a particular message. Only the fields specified in the MsgFields input parameter are retrieved.

## XML Return

```
<MsgFieldsList>
  <MsgFields>
    <MsgID>1000</MsgID>
    <PermanentMsgID>LS\1235090777</PermanentMsgID>
    <StoreType>Local</StoreType>
    <Type>Voice</Type>
    <Read>0</Read>

    <SoftDeleted>0</SoftDeleted>
    <Deleted>0</Deleted>          <Priority>N</Priority>
    <DeliveryTimestamp>2011-02-01T17:45:40Z</DeliveryTimestamp>
    <ChangeNumber>39</ChangeNumber>          <ChangedTimestamp>2011-02-
01T17:45:41Z</ChangedTimestamp>
    <SenderMBID>59999</SenderMBID>
    <SenderRemoteMBID/>
    <SenderName>AUDIO ADMINISTRATOR</SenderName>
  </MsgFields>
</MsgFieldsList>
```

## Errors

### 2147749712 User not logged on to MTA

- The user, for whom the mailbox messages are retrieved, is not logged onto their mail server.

2147749752 MTA error: The Start Change Number provided is too old.

- The specified Start Change Number is too old. This can only happen when the IncludeDeleted parameter is 1 and the server doesn't have message deletion changes that are this old.

## Comments

This function can be used in many different ways. One use of this function is to detect new messages since a given time. Another use is to discover message changes that have happened within the particular mailbox since a specified StartChangeNumber.

## Example

The following example retrieves messages that were delivered to the Inbox since the beginning of the year 2011 for the PST time-zone. Note the 8 hour time-adjustment to get the equivalent UTC time.

```
<CXMessageSearch>
  <SessionID>1972306838</SessionID>
  <SessionInstanceID>3d804697-74f6-4826-b8fa-93e98a328a62</SessionInstanceID>
  <StoreType>Local</StoreType>
  <Folder>New</Folder>
  <Style>LIFO</Style>
  <Count>100</Count>
  <StartTimestamp>2011-01-01T08:00:00Z </StartTimestamp>
  <MsgFields>
    <MsgID/>
    <PermanentMsgID/>
    <Read/>
    <Priority/>
    <DeliveryTimestamp/>
    <SenderMBID/>
    <RemoteSenderMBID/>
  </MsgFields>
</CXMessageSearch>
```

The following example retrieves message changes that have happened in the mailbox (including messages that have been deleted) since a **ChangeNumber** value of 10. This means that the client was up-to-date with all changes before 10 and it wants to discover any changes (messages added, updated or deleted) since that reference change number.

After the function returns, the client should iterate through each change and apply it to its local cache. It would also need to maintain and update an internal **LastChangeNumber** state to reflect the most recent change that it has received from the server. Then, in a subsequent request to **CXMessageSearch**, it should supply the value of **LastChangeNumber** + 1 for the **StartChangeNumber** parameter so that it can retrieve the next set of changes.

When making such periodic calls to **CXMessageSearch** to discover changes, it is possible that the server returns an error that indicates that the supplied Start Change Number is too old. This can happen if the last time that the client called **CXMessageSearch** was at least a day ago and there were deleted messages since then that have been erased out from the server's log. In such a case, the server is not able to reliably

provide the changes requested and so the client should throw its cache and start afresh. It can do that either by calling **CXMessageSearch** with a **StartChangeNumber** set to 0 or by calling **CXMessageSearch** without specifying any restrictive search criteria.

```
<CXMessageSearch>
  <SessionID>1972306838</SessionID>
  <SessionInstanceID>3d804697-74f6-4826-b8fa-93e98a328a62</SessionInstanceID>
  <StoreType>Local</StoreType>
  <Folder>New</Folder>
  <Style>LIFO</Style>
  <StartChangeNumber>10</StartChangeNumber>
  <IncludeDeleted>1</IncludeDeleted>
  <MsgFields>
    <MsgID/>
    <PermanentMsgID/>
    <Deleted/>          <ChangeNumber/>
  </MsgFields>
</CXMessageSearch>
```

## See Also

[CXMessageGet](#), [CXMessageUpdate](#)

## CXMessageGet

This function retrieves the **StoredMessage** structure for a message given its **MsgID** or **PermanentMsgID**. Only one of these parameters needs to be specified; the other one can be empty or missing.

## Syntax

```
<CXMessageGet>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>

  <MsgID></MsgID>
  <PermanentMsgID></PermanentMsgID>
</CXMessageGet>
```

## Input Parameters

SessionID

- Identifies the server session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

MsgID

- The session specific unique identifier for the message

PermanentMsgID

- The persistent unique identifier for the message

## Output Parameters

StoredMessage

- Refer to [StoredMessage](#) for details of this structure.

## XML Return

```
<StoredMessage>
  <StoreType>Local</StoreType>
  <Folder>New</Folder>
  <Read>1</Read>
  <Deleted>0</Deleted>
  <SoftDeleted>0</SoftDeleted>
  <Purged>0</Purged>
  <Message>
    <OkToFwd>1</OkToFwd>
    <ReturnReceiptRequested>0</ReturnReceiptRequested>
    <ReceiptType>Unknown</ReceiptType>
    <DeliveryError>0</DeliveryError>
    <SubdueReply>0</SubdueReply>
    <TotalVoiceMsec>2250</TotalVoiceMsec>
    <MsgID>1000</MsgID>
    <PermanentMsgID>LS\431</PermanentMsgID>
    <Type>Voice</Type>
    <VoiceMsgSubType>Normal</VoiceMsgSubType>
    <EmailMsgSubType>Normal</EmailMsgSubType>
    <Priority>N</Priority>
    <SentTimestamp>2014-01-09T15:08:36Z</SentTimestamp>

    <DeliveryTimestamp>2014-01-09T15:08:36Z</DeliveryTimestamp>
    <UserDeliveryTimestamp>2014-01-09T07:08:36-08:00</UserDeliveryTimestamp>
    <ChangeNumber>527</ChangeNumber>
    <ChangedTimestamp>2014-01-09T15:09:05Z</ChangedTimestamp>
    <CalendarStartTimestamp />
    <Subject>Voice Message from AUDIO ADMINISTRATOR Mailbox: 9999</Subject>
    <OriginalAddresseeList>
      <OriginalAddressee>
        <MBID>5001</MBID>
        <MBType />
        <Location />
        <RemoteMBID />
        <Extension />
        <Name>One, Local</Name>
        <EmailName />
        <EmailAddress />
      </OriginalAddressee>
    </OriginalAddresseeList>
  </Message>
</StoredMessage>
```

```

        <PhoneNumber />
        <MobileNumber />
        <AnnounceMsg>0</AnnounceMsg>
        <RecordMsg>0</RecordMsg>
    </OriginalAddressee>
</OriginalAddresseeList>
<OriginalSender>
    <MBID />
    <MBType />
    <Location />
    <RemoteMBID />
    <Extension />
    <Name />
    <EmailName />
    <EmailAddress />
    <PhoneNumber />
    <MobileNumber />
    <AnnounceMsg>0</AnnounceMsg>
    <RecordMsg>0</RecordMsg>
</OriginalSender>
<Sender>
    <MBID>9999</MBID>
    <MBType>A</MBType>
    <Location />
    <RemoteMBID />
    <Extension />
    <Name>AUDIO ADMINISTRATOR</Name>
    <EmailName />
    <EmailAddress />
    <PhoneNumber />
    <MobileNumber />
    <AnnounceMsg>0</AnnounceMsg>
    <RecordMsg>0</RecordMsg>
</Sender>
<Recipients>
    <Recipient>

```

```

        <MBID>5001</MBID>
        <MBType />
        <Location />
        <RemoteMBID />
        <Extension />
        <Name />
        <EmailName />
        <EmailAddress />
        <PhoneNumber />
        <MobileNumber />
        <AnnounceMsg>0</AnnounceMsg>
        <RecordMsg>0</RecordMsg>
    </Recipient>
</Recipients>
<Attachments>
    <Attachment>

```

```

    <ID>0</ID>
    <FileType>4</FileType>
    <FileName>2269bf77-2e9e-44d1-900c-4c5bae86a93c</FileName>
    <IsBody>0</IsBody>
    <FilePath>\CX\speech\msgs\2269bf77-2e9e-44d1-900c-4c5bae86a93c.WAV</FilePath>
    <PageCount>0</PageCount>
  </Attachment>
</Attachments>
</Message>
</StoredMessage>

```

## Example

The following example gets the full properties of a message by specifying its **PermanentMsgID** identifier.

```

<CXMessageGet>
  <SessionID>146538496</SessionID>
  <SessionInstanceID>7a5a9ccf-e3fc-4a2b-8f2e-84aa071d047a</SessionInstanceID>
  <PermanentMsgID>LS\1235090777</PermanentMsgID>
</CXMessageGet>

```

## See Also

[CXMessageSearch](#)

## CXMessageUpdate

Updates the properties of a message given its **MsgID** or **PermanentMsgID**. Only one of these parameters needs to be specified; the other one can be empty or missing. The **StoredMessage** parameter specifies the new properties for the message.

## Syntax

```

<CXMessageUpdate>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <MsgID></MsgID>
  <PermanentMsgID></PermanentMsgID>
  <StoredMessage>
  </StoredMessage>
</CXMessageUpdate>

```

## Input Parameters

**SessionID**

- Identifies the server session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

MsgID

- The session specific unique identifier for the message

PermanentMsgID

- The persistent unique identifier for the message

StoredMessage

- The structure that specifies the new properties for the message. Refer to [StoredMessage](#) for the details of this structure. It is not necessary to specify all the fields. The minimum requirement is that the fields that are being updated should be specified. See the example below where only the Read field is specified.
- StoredMessage fields that can be updated include:
- Folder – Specify *Saved* to move the message to the Saved folder.
- Read – Specify 1/0 to set/clear the message read property.
- Deleted – Specify 1 to delete the message.
- SoftDeleted – Specify 1/0 to set/clear the soft deleted property.
- Message/Subject – Specify the text for the new Subject Line.

## Output Parameters

None

## Example

The following example marks the message as read.

```
<CXMessageUpdate>
  <SessionID>218955855</SessionID>
  <SessionInstanceID>71d8ba2e-acf7-4661-8f9a-235bbb46710d</SessionInstanceID>
  <PermanentMsgID>LS\1235090652</PermanentMsgID>
  <StoredMessage>
    <Read>1</Read>
  </StoredMessage>
</CXMessageUpdate>
```

## See Also

[CXMessageSearch](#), [CXMessageGet](#)



## CXMessageRetrieveAttachments

Retrieves the list of attachments for a message. Also downloads the attachments onto the file-system. This happens when, for example, the message is stored in an External store. It could also happen for Local store messages when the SOAP connection is established to a Call Server. Calling this function, will then cause the call server to retrieve the attachment files from the System Server onto its local file system.

Specify one of the two parameters: MsgID and PermanentMsgID; the other one can be empty or missing.

### Syntax

```
<CXMessageRetrieveAttachments>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <MsgID></MsgID>
  <PermanentMsgID></PermanentMsgID>
</CXMessageRetrieveAttachments>
```

### Input Parameters

SessionID

- Identifies the server session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

MsgID

- The session specific unique identifier for the message

PermanentMsgID

- The persistent unique identifier for the message

### Output Parameters

Attachments

- The top-level container for the attachments. This contains one or more Attachment structures. Refer to [Attachment](#) for the details of the Attachment structure.

### XML Return

```
<Attachments>
  <Attachment>
    <ID>0</ID>
    <FileType>4</FileType>
    <FileName>c568838a-ef1f-4a6f-b7e9-b0de5bcfeca7</FileName>
```

```

        <IsBody>0</IsBody>
        <FilePath>\CX\speech\msgs\c568838a-ef1f-4a6f-b7e9-b0de5bcfeca7.WAV</FilePath>
        <PageCount>0</PageCount>
    </Attachment>
</Attachments>

```

## See Also

[CXMessageSearch](#), [CXMessageGet](#)

## CXMessageSend

Sends a new message, replies to an existing message, or forwards an existing message. The **Message** parameter specifies the properties of the message to be sent.

The **SendType** is specified as *New* to send New messages. Alternatively, the **SendType** can be specified as *Forward* or *Reply*.

For Forward and Reply, the **OriginalMsgIDDef** parameter needs to be specified. It is a structure containing the IDs of the original message which is being forwarded or replied to. Specify one of its two fields: **MsgID** and **PermanentMsgID**; the other one can be empty or missing.

If this function fails then it means that the message was not sent to any recipient. The error code returned will show the reason for the failure.

On the other hand, if this function succeeds, it can still fail for one or more recipients. In such a case, the per-recipient error value is returned via the **RecipientErrorList** output parameter.

## Syntax

```

<CXMessageSend>
    <SessionID>1298857990</SessionID>

    <SessionInstanceID>7a5a9ccf-e3fc-4a2b-8f2e-84aa071d047a</SessionInstanceID>
    <SendType>New</SendType>
    <OriginalMsgIDDef>
        <MsgID></MsgID>
        <PermanentMsgID></PermanentMsgID>
    </OriginalMsgIDDef>
    <Message>
    </Message>
    <ValidateMode></ValidateMode>
</CXMessageSend>

```

## Input Parameters

**SessionID**

- Identifies the server session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

SendType

- The Send type. This can be one of the following:
  - *New* – Sends a new message.
  - *Forward* – Forwards an existing message.
  - *Reply* – Replies to an existing message.

OriginalMsgIDDef

- This is a structure containing the **MsgID** and **PermanentMsgID** fields. Required for **SendType** Forward or Reply; ignored otherwise.

Message

- This is the structure containing the properties of the message. Refer to [Message](#) for the details of this structure.

ValidateMode

- This is an optional parameter which is always defaulted to the value 0 – which means that the **ValidateMode** is OFF. If this parameter is specified as 1 then it means that the **ValidateMode** is ON. If so, the function does all the necessary validations without actually sending the message. The **ValidateMode** is useful for applications to first validate the send operation when the user picks a recipient to send the message to, for example.

## Output Parameters

RecipientErrorList

The top-level container for the per-recipient error values. This contains one or more **RecipientError** fields. Each **RecipientError** field holds the error value for exactly one Recipient. This list of **RecipientError** fields is returned in the same order as the list of Recipients specified within the Message input parameter.

If a **RecipientError** field's value is 0 then that means that the message was successfully sent to that recipient. If it is greater than zero then it indicates that some kind of error happened and the message wasn't sent to that recipient. Here is the table of the error values that can be returned via this field:

Table 11. Error values

Error Value	Error String	Description
2147749707	Invalid MTA User	The recipient mailbox is invalid for sending this message.
2147749711	Invalid MTA Recipient	The recipient info was not provided or the recipient could not be located.

2147749718	Invalid Recipient Mailbox Type	This happens when trying to send a message with a type that cannot be accepted by the recipient mailbox type.
2147749730	MTA error: Recipient mailbox does not exist	The recipient mailbox could not be located.
2147749736	MTA error: Network access is not authorized.	The mailbox is targeting a NET/AMIS location but the sender does not have network access.
2147749737	MTA error: That network node is currently not available.	The network access on the local node is disabled and the selected recipient is in a remote NET/ AMIS network node.
2147749738	MTA error: A remote mailbox number is required.	The recipient mailbox provided is a AMIS/NET/VPIM mailbox but the RemoteMBID field associated with the recipient is empty.
2147749739	MTA error: The recipient is out of the office and is not accepting messages.	The recipient is not enabled to accept messages and has the extended absence greeting enabled.
2147749740	MTA error: The recipient mailbox is not accepting messages.	The recipient is not enabled to accept messages.
2147749741	MTA error: That mailbox has not been activated and is not accepting messages.	The recipient mailbox is not enabled.
2147749742	MTA error: That recipient mailbox requires a fax message.	This happens while sending a voice message to a FXD mailbox.
2147749743	MTA error: You are not allowed to send messages to that distribution list.	The sender does not have the rights to send to the DL selected as a recipient.
2147749744	MTA error: Recipient remote mailbox does not exist.	The remote mailbox provided as recipient does not exists in the directory list of the NET or VPIM mailbox specified as the recipient and mailbox validation

		is enabled on the NET/VPIM mailbox.
2147749745	MTA error: The sender cannot send message to that type of recipient mailbox.	The sender mailbox type cannot send to the mailbox type of the recipient provided. For instance AMIS mailbox can only send to Subscribers and DLs.
2147749747	MTA error: The sender is not allowed to dial that phone number.	The phone number cannot be dialed on behalf of the sender mailbox.
2147749748	MTA error: The recipient mailbox FXD/OUT/AMIS does not have a phone number. The user must provide one.	A phone number was not specified.
2147749749	MTA error: The recipient DL is empty.	The DL provided as a recipient does not have any member.
2147749750	MTA error: The required NET/VPIM/FAX feature flag is not enabled in the security file.	One of the required features flags is not enabled .
2147749751	MTA error: The length of the provided remote mailbox does not match the length configured on the NET/VPIM node.	The length of the provided remote mailbox number is different than the length configured on the specified VPIM or NET mailbox.

## XML Return

```
<RecipientErrorList>
  <RecipientError>0</RecipientError>
  <RecipientError>2147749730</RecipientError>
</RecipientErrorList>
```

## Errors

2147749809 Invalid credentials

- The logged-on user doesn't have the privileges to send this message.

2147749707 Invalid MTA User

- The sender mailbox is invalid.

2147749708 MTA No Message Access

- The specified Message structure has missing information.

2147749710 Mail Store Error

- General Mail Store Error.

2147749714 Invalid Send Type specified

- The specified SendType is invalid.

2147749715 Invalid Parameters

- One or more input parameters are invalid.

## Example

The following example sends the message with ObjectID 2269bf77-2e9e-44d1-900c-4c5bae86a93c to mailbox 59999. The sender is the logged on mailbox.

```
<CXMessageSend>
  <SessionID>1298857990</SessionID>
  <SessionInstanceID>7a5a9ccf-e3fc-4a2b-8f2e-84aa071d047a</SessionInstanceID>
  <SendType>New</SendType>
  <Message>
    <Type>Voice</Type>
    <VoiceMsgSubType>Normal</VoiceMsgSubType>
    <Priority>N</Priority>
    <OkToFwd>1</OkToFwd>
    <ReturnReceiptRequested>0</ReturnReceiptRequested>
    <Recipients>
      <Recipient>
        <MBID>59999</MBID>
      </Recipient>
    </Recipients>
    <Attachments>
      <Attachment>
        <FileType>4</FileType>
        <FileName>2269bf77-2e9e-44d1-900c-4c5bae86a93c</FileName>
      </Attachment>
    </Attachments>
  </Message>
</CXMessageSend>
```

## See Also

[CXLMObjectClose](#), [CXLMObjectCreate](#), [CXMessageSearch](#), [CXMessageGet](#)

## CXMWIClear

Clears the Message Waiting Indicator for the subscriber.

## Syntax

```
<CXMWIClear>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
</CXMWIClear>
```

## Input Parameters

SessionID

- Identifies the server session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

## Output Parameters

None

## Example

The following example clears the Message Waiting Indicator.

```
<CXMWIClear>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
</CXMWIClear>
```

## See Also

[CXIMNCancel](#)

## CXIMNCancel

Cancels Immediate Message Notification for the subscriber.

## Syntax

```
<CXIMNCancel>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
</CXIMNCancel>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

## Output Parameters

None

## Comments

This function cancels the Immediate Message Notification (IMN) for messages that the subscriber has received up to the point when this function is called. This function does not cancel the IMN for messages received after the function finishes.

## Example

The following example cancels Immediate Message Notification for the subscriber.

```
<CXIMNCancel>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
</CXIMNCancel>
```

## See Also

[CXMWIClear](#)

## CXLMLineAttach

Calls the specified number after verifying that the number passes the dial plan and that the subscriber mailbox has sufficient permissions. If the call is successful, the line is open and ready to play or record audio.

## Syntax

```
<CXLMLineAttach>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <Phone></Phone>
```



```
</CXLMLineAttach>
```

## Input Parameters

### SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

### SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

### Phone

- The telephone number to which the call is made.

## Output Parameters

### CallProgress

- Returns a numeric value indicating the call progress state. Values for CallProgress are:
  - 0 – The telephone is ringing (no answer)
  - 1 – The telephone line is busy
  - 2 – The number has either a Do Not Disturb (DND) setting, is disconnected, or is a misdialed number
  - 3 – The call has a dial tone
  - 4 – The line is dead
  - 5 – The line has connected (success)
  - 6 – The status is unknown

## XML Return

```
<CallProgress>5</CallProgress>
```

## Errors

2147751607     Line Manager error: Line already attached

- A line is already attached on this session. **CXLMLineDetach** needs to be called before **CXLMLineAttach** can be called again.

2147751608     Line Manager error: No line available

- No physical lines are available on the server to place the call or the user is not allowed to make this call.

## Comments

Other **CXLM** functions (including the **CXLMObject** functions) can only be called after a successful call to **CXLMLineAttach** has been made.

## Example

The following example establishes a line to the number represented by NNNN.

```
<CXLMLineAttach>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
  <Phone>NNNN</Phone>
</CXLMLineAttach>
```

## See Also

[CXLMLineAttach](#), [Line Management Functions](#)

## CXLMLineDetach

Hangs up the attached line. Also frees resources that are associated with the attached line.

## Syntax

```
<CXLMLineDetach>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
</CXLMLineDetach>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

## Output Parameters

None

## Errors

2147751606 Line Manager error: Line not attached

- A line is not attached. Likely cause of this error is that a **CXLMLineAttach** call was not made on this session.

## Comments

You should detach the line even if a call to **CXLMLineStatusGet** returns a 0 value for the LineStatus parameter, indicating that the called phone has been disconnected.

CXLM functions, other than **CXLMLineAttach**, cannot be called after a call to **CXLMLineDetach** has been made.

**NOTE** When CXLogoff is called, all resources are released, including any attached lines.

## See Also

[CXLMLineAttach](#), [Line Management Functions](#)

## CXLMOBJECTCreate

Creates a new recording object and returns the ID. CXLMObjectOpen or CXLMObjectCreate must be called before any other CXLMObject operations are called on the object.

## Syntax

```
<CXLMObjectCreate>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
</CXLMObjectCreate>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

## Output Parameters

ObjectID

- The unique identifier for the object.

## XML Return

```
<ObjectID>ab52ed7b-87c1-49a8-8f2a-a85578365ffc</ObjectID>
```

## Errors

2147751606      Line Manager error: Line not attached

- A line is not attached. Likely cause of this error is that a **CXLMLineAttach** call was not made on this session.

## Comments

This function has the same syntax as that of **CXObjectCreate**. The main difference in behavior is that this function also keeps track of the Node that has the attached line. So, use this function when you are dealing with objects that need to be played / recorded to over a Phone line using one of the **CXLM** functions.

Objects are created in a session-specific temporary storage folder on the server. Objects need to be closed using **CXLMOBJECTClose**. If this call is not used to close the objects then the objects will get cleaned up whenever the session is logged off.

## See Also

[CXObjectCreate](#), [CXLMOBJECTOpen](#), [CXLMOBJECTClose](#), [Line Management Functions](#)

## CXLMOBJECTOpen

Opens an existing recording object with the specified type and ID. CXLMOBJECTOpen or CXLMOBJECTCreate must be called before any other CXLMOBJECT operations are called on the object.

## Syntax

```
<CXLMOBJECTOpen>  
  <SessionID></SessionID>  
  <SessionInstanceID></SessionInstanceID>  
  <ObjectID></ObjectID>  
  <RecordingTypeID></RecordingTypeID>  
</CXLMOBJECTOpen>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

#### SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

#### ObjectID

- The unique identifier for the object. A valid ObjectID obtained from the following fields can be specified here:
  - The **MBCommon/NameMediaFileID** field for the recorded name of a mailbox.
  - The **PromptRecording/RecordingTypeID** field for a Normal, Busy or Out-of-office greeting.
  - The **Attachment/FileName** field for a message attachment.

#### RecordingTypeID

- The type of the Recording. Valid values are:
  - *Busy* – This is the busy greeting. Note that for a Call Processor mailbox, this recording type represents an Instruction Announcement.
  - *OOF* - This is the Out-of-office greeting.
  - *Name* - This is the recorded Name.
  - *Greeting* - This is the normal greeting.
  - *Message* - This is a message attachment.
  - *Introduction* - This is an Intro Announcement for a Call Processor mailbox.
  - *Availability* - This is the Availability set of greetings.

## Output Parameters

None

## Errors

2147751606 Line Manager error: Line not attached

- A line is not attached. Likely cause of this error is that a **CXLMLineAttach** call was not made on this session.

2147751612 Local file copy error

- The object file could not be copied into the temporary folder. Likely cause of this error is that the source object ID provided is wrong.

## Comments

This function has the same syntax as that of **CXObjectOpen**. The main difference in behavior is that this function also keeps track of the Node that has the attached line. So, use this function when you are dealing with objects that need to be played over a Phone line using one of the **CXLM** functions.

Opened objects are stored in a session-specific temporary storage folder on the server. Objects need to be closed using **CXLMOBJECTClose**. If this call is not used to close the objects then the objects will get cleaned up whenever the session is logged off.

## Example

The following example opens a greeting object.

```
<CXLMObjectOpen>
  <SessionID>1736771367</SessionID>
  <SessionInstanceID>2a6ec288-eda5-4327-8db0-638aec22dca2</SessionInstanceID>
  <ObjectID>f15612b7-327b-4edf-af30-cd8a2fc4a068</ObjectID>
  <RecordingTypeID>Greeting</RecordingTypeID>
</CXLMObjectOpen>
```

## See Also

[CXObjectOpen](#), [CXLMObjectCreate](#), [CXLMObjectClose](#), [Line Management Functions](#)

## CXLMOBJECTClose

Closes the specified object. A RecordingTypeID parameter needs to be specified if the object needs to be saved permanently.

## Syntax

```
<CXLMObjectClose>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <ObjectID></ObjectID>
  <RecordingTypeID></RecordingTypeID>

</CXLMObjectClose>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogin**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

ObjectID

- The identifier of the object that was previously created or opened by calling **CXLMOBJECTCreate** or **CXLMOBJECTOpen**.

RecordingTypeID

- The type of the Recording that this object should be saved as. This can be left empty/missing to indicate that the server should clean up the object without saving it permanently. Valid values for saving the object are:
  - *Busy* – This is the busy greeting. Note that for a Call Processor mailbox, this recording type represents an Instruction Announcement.
  - *OOF* - This is the Out-of-office greeting.
  - *Name* - This is the recorded Name.
  - *Greeting* - This is the normal greeting.
  - *Message* - This is a message attachment.
  - *Introduction* - This is an Intro Announcement for a Call Processor mailbox.
  - *Availability* - This is the Availability set of greetings.

## Output Parameters

None

## Comments

Objects created by calling **CXLMOBJECTCreate** may need to be saved to the permanent storage. This can be done by calling **CXLMOBJECTClose** with a valid RecordingTypeID parameter.

Objects opened by calling **CXLMOBJECTOpen** already exist within the permanent storage. Thus they need not be saved. **CXLMOBJECTClose** can be called for these objects with a missing or empty **RecordingTypeID** parameter.

## Errors

2147751606 Line Manager error: Line not attached

- A line is not attached. Likely cause of this error is that a **CXLMLineAttach** call was not made on this session.

2147751612 Local file copy error

- The object file could not be copied into the permanent storage. Possible causes are: wrong source object ID or an object which already exists in the permanent storage.

## Example

The following example closes the greeting object that was previously opened by calling **CXLMOBJECTOPEN**.

```
<CXLMOBJECTCLOSE>
  <SessionID>1736771367</SessionID>
  <SessionInstanceID>2a6ec288-eda5-4327-8db0-638aec22dca2</SessionInstanceID>
  <ObjectID>f15612b7-327b-4edf-af30-cd8a2fc4a068</ObjectID>
  <RecordingTypeID></RecordingTypeID>
</CXLMOBJECTCLOSE>
```

## See Also

[CXOBJECTCLOSE](#), [CXLMOBJECTCREATE](#), [CXLMOBJECTOPEN](#), [Line Management Functions](#)

## CXLMOBJECTUPLOAD

Uploads the specified binary content into the recording object.

## Syntax

```
<CXLMOBJECTUPLOAD>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <ObjectID></ObjectID>
  <FileData></FileData>
  <StartPosByte></StartPosByte>
</CXLMOBJECTUPLOAD>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLOGON**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLOGON**.

ObjectID

- The identifier of the object that was previously created by calling **CXLMOBJECTCREATE**.

FileData

- Specifies the binary data in the base64-encoded format that needs to be uploaded into the object file. The binary data, without the base64-encoding, is persisted into the object file.



StartPosByte

- Optional. The zero-based byte offset within the object's file at which to start adding the new data. If not specified, this is assumed to be 0 meaning the beginning of the file.

## Output Parameters

None

## Errors

2147751606      Line Manager error: Line not attached

- A line is not attached. Likely cause of this error is that a **CXLMLineAttach** call was not made on this session.

## Example

The following example uploads binary data into the object file by specifying the base-64 encoded format of that data in the **FileData** parameter. Only a portion of the content in the FileData parameter has been shown in the example.

```
<CXLMObjectUpload>
  <SessionID>1736771367</SessionID>
  <SessionInstanceID>2a6ec288-eda5-4327-8db0-638aec22dca2</SessionInstanceID>
  <ObjectID>ab52ed7b-87c1-49a8-8f2a-a85578365ffc</ObjectID>
  <FileData>UklGRvJYAABXQVZFZm1...+f3///79/f4=</FileData>
</CXLMObjectUpload>
```

## See Also

[CXObjectUpload](#), [CXLMObjectCreate](#), [CXLMObjectDownload](#), [Line Management Functions](#)

## CXLMObjectDownload

Downloads the binary content in the specified recording object.

## Syntax

```
<CXLMObjectDownload>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <ObjectID></ObjectID>
  <StartPosByte></StartPosByte>
  <LengthBytes></LengthBytes>
</CXLMObjectDownload>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

ObjectID

- The identifier of the object that was previously created or opened by calling **CXLMOBJECTCreate** or **CXLMOBJECTOpen**.

StartPosByte

- Optional. The zero-based byte offset within the object's file from which to start fetching the data. If not specified, this is assumed to be 0 meaning the beginning of the file.

LengthBytes

- Optional. The count of the bytes to download. If not specified, this is assumed to be -1 meaning that the data till the end of the file needs to be retrieved.

## Output Parameters

FileData

- This is the binary data in the file in the base64-encoded format.

## XML Return

Only a portion of the content in the **FileData** parameter has been shown below.

```
<FileData>UklGRvJYAABXQVZFZm1...+f3////79/f4=</FileData>
```

## Errors

2147751606      Line Manager error: Line not attached

- A line is not attached. Likely cause of this error is that a **CXLMLineAttach** call was not made on this session.

## Example

The following example downloads the binary data within an object file.

```
<CXLMObjectDownload>
  <SessionID>1736771367</SessionID>
  <SessionInstanceID>2a6ec288-eda5-4327-8db0-638aec22dca2</SessionInstanceID>
  <ObjectID>f15612b7-327b-4edf-af30-cd8a2fc4a068</ObjectID>
```

```
</CXLMObjectDownload>
```

## See Also

[CXObjectDownload](#), [CXLMObjectCreate](#), [CXLMObjectOpen](#), [CXLMObjectUpload](#), [Line Management Functions](#)

## CXLMLinePlay

Starts playback of the specified recording on the attached line.

## Syntax

```
<CXLMLinePlay>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <ObjectID></ObjectID>
  <PermanentMsgID></PermanentMsgID>
  <StartPos>0</StartPos>
  <EndPos>-1</EndPos>
</CXLMLinePlay>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

ObjectID

- The identifier of the object that was previously created or opened by calling **CXLMObjectCreate** or **CXLMObjectOpen**. Specify this parameter or the **PermanentMsgID** parameter.

PermanentMsgID

- The Permanent ID of a message that was obtained by calling **CXMessageSearch** or **CXMessageGet**. Specify this parameter or the **ObjectID** parameter.

StartPos

- Specifies the position (in milliseconds) in the recording at which to start playback. Use a value of zero to start playback at the beginning, non-zero values cause playback to start at a position in-between the recording object.

EndPos

- Optional. This parameter specifies the position (in milliseconds) at which to end the playback. Specify a value of -1 for no limit.

## Output Parameters

None

## Errors

2147751606 Line Manager error: Line not attached

- A line is not attached. Likely cause of this error is that a **CXLMLineAttach** call was not made on this session.

2147751613 Line Manager error: Object does not exist

- The specified ObjectID parameter is incorrect.

## Comments

Specify an ObjectID if playing a recording object. Specify a **PermanentMsgID** if playing a message.

This function returns immediately (with error if playback could not be started). During playback, CXLMLineStatusGet can be used to check the state of the line and CXLMLineWait can be used to either stop the playback or wait until playback completes.

## Example

The following example plays the specified recording object beginning at one second into the recording.

```
<CXLMLinePlay>
  <SessionID>1736771367</SessionID>
  <SessionInstanceID>2a6ec288-eda5-4327-8db0-638aec22dca2</SessionInstanceID>
  <ObjectID>f15612b7-327b-4edf-af30-cd8a2fc4a068</ObjectID>
  <StartPos>1000</StartPos>
</CXLMLinePlay>
```

## See Also

[CXLMLineRecord](#), [Line Management Functions](#)

## CXLMLineRecord

Starts recording audio on the attached line.

## Syntax

```
<CXLMLineRecord>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <ObjectID></ObjectID>
  <StartPos>0</StartPos>
  <EndPos>-1</EndPos>
</CXLMLineRecord>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

ObjectID

- The identifier of the object that was previously created by calling **CXLMOBJECTCreate**.

StartPos

- Specifies the position (in milliseconds) in the recording at which to start recording. Use a value of zero to start recording at the beginning, non-zero values cause new audio to be appended starting at the specified location in the recording object.

EndPos

- Optional. This parameter specifies the position (in milliseconds) at which to end the recording. Specify a value of -1 for no limit.

## Output Parameters

None

## Errors

2147751606     Line Manager error: Line not attached

- A line is not attached. Likely cause of this error is that a **CXLMLineAttach** call was not made on this session.

2147751613     Line Manager error: Object does not exist

- The specified ObjectID parameter is incorrect.

## Comments

This function returns immediately (with error if recording could not be started). During recording, **CXLMLineStatusGet** can be used to check the state of the line and **CXLMLineWait** can be used to either stop the recording or wait until recording completes due to a silence timeout or hang-up.

## Example

The following example records up to 10 seconds into the specified object.

```
<CXLMLineRecord>
  <SessionID>1736771367</SessionID>
  <SessionInstanceID>2a6ec288-eda5-4327-8db0-638aec22dca2</SessionInstanceID>
  <ObjectID>ab52ed7b-87c1-49a8-8f2a-a85578365ffc</ObjectID>
  <StartPos>0</StartPos>
  <EndPos>10000</EndPos>
</CXLMLineRecord>
```

## See Also

[CXLMLinePlay](#), [Line Management Functions](#)

## CXLMLineStatusGet

Returns the status of the attached line.

## Syntax

```
<CXLMLineStatusGet>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
</CXLMLineStatusGet>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

## Output Parameters

LineStatus

- The status of the line. Possible values are:
  - 0 – The line is on the hook (not in use).
  - 1 – The line is off the hook (in use).

## XML Return

```
<LineStatus>1</LineStatus>
```

## Errors

2147751606      Line Manager error: Line not attached

- A line is not attached. Likely cause of this error is that a **CXLMLineAttach** call was not made on this session.

## Comments

You should call this function periodically while recording or playing to determine if the connection has been broken by a hang up. If a disconnect has occurred, **CXLMLineDetach** must be called to release the resources associated with the line.

## See Also

[CXLMLineWait](#), [CXLMLineDetach](#), [Line Management Functions](#)

## CXLMLineWait

Waits for termination of telephone recording or playback.

## Syntax

```
<CXLMLineWait>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <Terminate></Terminate>
```

```
</CXLMLineWait>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

#### Terminate

- This parameter specifies whether termination should be immediate (Terminate=1), or should wait for external termination (Terminate=0). External termination may be caused by end of recording, maximum recording time reached or hang-up.
- If Terminate is 0, this function does not return until the current action on the line (playback or recording) is complete or the wait times out. If Terminate is 1, the function terminates the action and returns immediately.

#### Timeout

- This parameter specifies the timeout interval in milliseconds for which to wait for the recording or playback operation to complete. If this parameter is missing or set to 0, the default timeout of 1 minute is used.

## Output Parameters

#### Length

- The length, in milliseconds, of the recording or playback.

## XML Return

```
<Length>2418</Length>
```

## Errors

2147751606     Line Manager error: Line not attached

- A line is not attached. Likely cause of this error is that a **CXLMLineAttach** call was not made on this session.

2147751621     Line Manager error: Operation timed out

- The operation timed out. This means that the recording or playback operation is still in progress. **CXLMLineWait** needs to be called again.

## Example

The following example terminates the active playback operation. No timeout is necessary for this scenario.

```
<CXLMLineWait>
  <SessionID>1736771367</SessionID>
  <SessionInstanceID>2a6ec288-eda5-4327-8db0-638aec22dca2</SessionInstanceID>
  <Terminate>1</Terminate>
</CXLMLineWait>
```



## See Also

[CXLMLineStatusGet](#), [Line Management Functions](#)

## CXLMLineTransferStart

Initiates a call transfer to the specified telephone number or extension.

## Syntax

```
<CXLMLineTransferStart>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <Phone></Phone>
  <TransferType></TransferType>
  <DialPlanConversion></DialPlanConversion>
  <RingCount></RingCount>
</CXLMLineTransferStart>
```

## Input Parameters

### SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

### SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

### Phone

- Phone number or extension to which to transfer the call.

### TransferType

- Type of transfer. Valid values are:
  - 1 – Blind: The transfer is initiated and the server immediately drops the line.
  - 3 – Monitored: The transfer is initiated and the line is dropped as soon as the phone is answered or is ringing.
  - 4 – Supervised: Control of the line is maintained until a call is made to CXLMLineTransferEnd.
  - 5 – Default: The default Transfer Type specified in the environment settings on the server is used.

### RingCount

- The number of times to allow the called number to ring before returning a no-answer condition for a supervised transfer.

## Output Parameters

### CallProgress

- A value indicating the call progress state:
  - 0 – Ring
  - 1 – Busy
  - 2 – Reorder
  - 3 – Dialtone
  - 4 – DeadLine
  - 5 – Connected (Success)
  - 6 – Unknown

## Errors

2147751606      Line Manager error: Line not attached

- A line is not attached. Likely cause of this error is that a **CXLMLineAttach** call was not made on this session.

2147750703      Data-layer error: Phone number failed dialplan

- The Phone parameter could not be validated against the dial plan.

2147750704      Data-layer error: Phone number failed permissions

- The logged-on user doesn't have permissions to call the number specified by the Phone parameter.

## Comments

**CXLMLineTransferStart** retains control of execution until it has determined the call progress status or, in the case of a blind transfer, has completed the transfer.

Blind transfers should only be made to numbers where the call will be handled correctly if the line is busy. This includes some operator consoles, hunt groups, and numbers that are automatically forwarded to voice mail or an operator console.

Monitored transfers are usually used where a blind transfer is desirable, but the switching system does not handle a blind transfer to a busy line correctly.

A supervised transfer gives the most control. The call progress determination is set in the **CallProgress** parameter. Prior to calling the **CXLMLineTransferEnd** function, a script can play an announcement that will be heard by the called party.

Call the **CXLMLineTransferEnd** method to complete a supervised transfer (TransferType= 4). Blind and monitored transfers are completed automatically without requiring a call to **CXLMLineTransferEnd**.

## See Also

[CXLMLineTransferEnd](#), [Line Management Functions](#)

# CXLMLineTransferEnd

Completes or aborts a supervised transfer started with **CXLMLineTransferStart**.

## Syntax

```
<CXLMLineTransferEnd>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <TransferAction></TransferAction>
</CXLMLineTransferEnd>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

TransferAction

- Indicates the action to take on the transfer. This method must be called on supervised transfers. Valid values are:
  - 1 – Abort
  - 2 – Complete
  - 3 – Reject

## Output Parameters

None

## Errors

2147751606     Line Manager error: Line not attached

- A line is not attached. Likely cause of this error is that a **CXLMLineAttach** call was not made on this session.

## Comments

This function must be called on supervised transfers.

Applications using supervised transfers should examine the **CallProgress** parameter returned by **CXLMLineTransferStart** to determine whether or not to complete or abort the transfer. Usually a transfer should be completed if the call progress result is Connected or DeadLine.

## See Also

[CXLMLineTransferStart](#), [Line Management Functions](#)

## CXGetVersionInfo

Gets the version information for the server. This call does not require a valid session to execute.

## Syntax

```
<CXGetVersionInfo></CXGetVersionInfo>
```

## Input Parameters

None

## Output Parameters

Version

- The version number of the server

BuildNumber

- The build number of the server

## XML Return

```
<Version></Version>  
<BuildNumber></BuildNumber>
```

## Errors

None

# CXPasswordResetRequestCreate

Creates a Password Reset Request and sends an e-mail to the user containing a link that allows the user to reset the password. This call does not require a valid session to execute.

## Syntax

```
<CXPasswordResetRequestCreate>
  <MBID></MBID>
  <EmailAddress></EmailAddress>
  <ServerAddr></ServerAddr>
</CXPasswordResetRequestCreate>
```

## Input Parameters

### MBID

- The mailbox number of the user whose password needs to be reset. This is used by the system to validate the request. This parameter can be left unspecified when the EmailAddress parameter is specified.

### EmailAddress

- The e-mail address of the user whose password needs to be reset. This is used by the system to validate the request. This parameter can be left unspecified when the MBID parameter is specified.

### ServerAddress

- Optional. The address of the System. This parameter is not used by the System. It is only included in the link within the Password Reset e-mail so that the application parsing this link can find the right server to connect to in order to make a call to [CXPasswordReset](#).

## Output Parameters

### MBID

- The user's mailbox number.

### DisplayName

- The user's name.

### PasswordResetMessageExpirationMinutes

- The time in minutes after which the created Request will expire. The user needs to click the link within this time in order to successfully reset the password.

## Errors

2147750754     Data-layer error: Password Reset not possible

- Password Reset is not possible for this user.

2147750755      Data-layer error: Password Reset Request data invalid

- Password Reset Request data is invalid.

2147750764      Data-layer error: Password Reset E-mail failed

- Failed to send Password Reset E-mail.

## CXPasswordReset

Resets the password for a user. This call does not require a valid session to execute.

### Syntax

```
<CXPasswordReset>
  <MBID></MBID>
  <RequestID></RequestID>
  <Password></Password>
  <ValidateMode></ValidateMode>
</CXPasswordReset>
```

### Input Parameters

MBID

- The mailbox number of the user whose password needs to be reset.

RequestID

- This parameter identifies the Reset Request created for the user.

Password

- The user's new password.

ValidateMode

- Optional. If this parameter is set to 1, the system only validates the RequestID and does not actually reset the password.

### Output Parameters

None

### Errors

2147750756      Data-layer error: Password Reset Request invalid

- Password Reset Request is invalid.

2147750757      Data-layer error: Password Reset Request expired

- Password Reset Request has expired.

2147750761      Data-layer error: Password invalid

- Password is invalid.

2147750762      Data-layer error: Password failed security policy

- Password failed security policy.

2147750763      Data-layer error: Password cannot be reused

- Password cannot be reused.

## CXGetMgmtData

Gets the MgmtData record for the server to which the session is attached.

### Syntax

```
<CXGetMgmtData>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
</CXGetMgmtData>
```

### Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

### Output Parameters

CXMGMTDATA

- Following are the fields of this parameter:
- SiteName
  - The name of the site as specified during installation
- APIVersion
  - The version of the API used to access the server information
- ServerVersion
  - The version, build, and service pack number of the server
- SerialNumber

- The serial number for the system

## XML Return

```
<CXMGMTDATA>
  <SiteName></SiteName>
  <APIVersion></APIVersion>
  <ServerVersion></ServerVersion>
  <SerialNumber></SerialNumber>
  <CXDomain></CXDomain>
</CXMGMTDATA>
```

## Errors

2147749804     Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session

## CXLanguagePackListGet

Retrieves the list of the language packs that are configured on this server.

## Syntax

```
<CXLanguagePackListGet>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
</CXLanguagePackListGet>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

## Output Parameters

LanguagePackList



## Container of one or more LanguagePack structures.

### LanguagePack

- Structure holding the information for one language pack.

### LanguagePackID

- Identifies the language pack to which the data within this structure applies.

### Name

- The display name for this language pack.

### PromptLangID

- The numerical identifier of the prompt language.

### TTSCultureName

- Identifies the TTS language using culture name format such as *en-US* for US English.

### ASRCultureName

- Identifies the ASR language using culture name format such as *en-US* for US English.

### IsDefault

- This is a Boolean field that tells whether this language pack is the default for this system. If the value for this field is set to 1 then it means that this language pack is the default for this system. Otherwise, this value would be set to 0.

## XML Return

```
<CXResponse>
  <LanguagePackList>
    <LanguagePack>
      <LanguagePackID>1</LanguagePackID>
      <Name>FEMALE ENGLISH NA</Name>
      <PromptLangID>0</PromptLangID>
      <TTSCultureName>en-US</TTSCultureName>
      <ASRCultureName>en-US</ASRCultureName>
      <IsDefault>1</IsDefault>
    </LanguagePack>
  </LanguagePackList>
</CXResponse>
```

## Errors

### 2147749804 Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session

# CXSpeechCmdListGet

Retrieves the list of the configured speech commands.

## Syntax

```
<CXSpeechCmdListGet>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <LastSpeechCmdID></LastSpeechCmdID>
  <Count></Count>
</CXSpeechCmdListGet>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

LastSpeechCmdID

- The identifier of the last speech command retrieved from the server. Only speech commands with higher IDs than the specified ID will be retrieved.

Count

- Maximum number of speech commands to be retrieved.

## Output Parameters

SpeechCmdList

- List of one or more speech commands.

SpeechCmd

- Refer to [SpeechCmd](#) for details of this structure.

## XML Return

```
<CXResponse>
  <SpeechCmdList>
    <SpeechCmd>
      <SpeechCmdID>4</SpeechCmdID>

      <CollectionType>CallProcessorCmd</CollectionType>
      <SpeechAliasCollectionArray>
```

```

    <SpeechAliasCollection>
      <SpeechAliasCollectionID>4</SpeechAliasCollectionID>
      <CultureName>en-US</CultureName>
      <TTSNameAlias />
      <SpokenNameFile />
      <Name>User logon</Name>
      <SpeechAliasArray>
        <SpeechAlias>
          <SpeechAliasID>151</SpeechAliasID>
          <Phrase>User login</Phrase>
        </SpeechAlias>
        <SpeechAlias>
          <SpeechAliasID>150</SpeechAliasID>
          <Phrase>User logon</Phrase>
        </SpeechAlias>
      </SpeechAliasArray>
    </SpeechAliasCollection>
  </SpeechAliasCollectionArray>
</SpeechCmd>
<SpeechCmd>
  <SpeechCmdID>5</SpeechCmdID>
  <CollectionType>CallProcessorCmd</CollectionType>
  <SpeechAliasCollectionArray>
    <SpeechAliasCollection>
      <SpeechAliasCollectionID>5</SpeechAliasCollectionID>
      <CultureName>en-US</CultureName>
      <TTSNameAlias />
      <SpokenNameFile />
      <Name>Operator</Name>
      <SpeechAliasArray>
        <SpeechAlias>
          <SpeechAliasID>152</SpeechAliasID>
          <Phrase>Operator</Phrase>
        </SpeechAlias>
        <SpeechAlias>
          <SpeechAliasID>154</SpeechAliasID>
          <Phrase>seero</Phrase>
        </SpeechAlias>
        <SpeechAlias>
          <SpeechAliasID>153</SpeechAliasID>
          <Phrase>Zero</Phrase>
        </SpeechAlias>
      </SpeechAliasArray>
    </SpeechAliasCollection>
  </SpeechAliasCollectionArray>
</SpeechCmd>
</SpeechCmdList>
</CXResponse>

```

## Errors

2147749804      Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session

## CXSpeechCmdGet

This function retrieves the SpeechCmd structure for a speech command given its SpeechCmdID.

## Syntax

```
<CXSpeechCmdGet>  
  <SpeechCmdID></SpeechCmdID>  
</CXSpeechCmdGet>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

SpeechCmdID

- The identifier of the speech command to be retrieved from the server.

## Output Parameters

SpeechCmd

- Refer to [SpeechCmd](#) for details of this structure.

## XML Return

```
<CXResponse>  
  <SpeechCmd>  
    <SpeechCmdID>10005</SpeechCmdID>  
    <CollectionType>Group</CollectionType>  
    <SpeechAliasCollectionArray>  
      <SpeechAliasCollection>  
        <SpeechAliasCollectionID>21</SpeechAliasCollectionID>  
        <CultureName>en-us</CultureName>  
        <TTSNameAlias />  
      </SpeechAliasCollection>  
    </SpeechAliasCollectionArray>  
  </SpeechCmd>  
</CXResponse>
```

```

    <SpokenNameFile />
    <Name>Engineering</Name>
    <SpeechAliasArray>
      <SpeechAlias>
        <SpeechAliasID>372</SpeechAliasID>
        <Phrase>Development</Phrase>
      </SpeechAlias>
      <SpeechAlias>
        <SpeechAliasID>371</SpeechAliasID>
        <Phrase>Engineering</Phrase>
      </SpeechAlias>
    </SpeechAliasArray>
  </SpeechAliasCollection>
</SpeechAliasCollectionArray>
</SpeechCmd>
</CXResponse>

```

## Errors

2147749804    Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session

2147750605    Data-layer error: Object not found

- No speech command found with the specified ID.

## CXSpeechCmdAdd

Adds a new speech command. While calling this function, the **SpeechCmdID**, **SpeechAliasCollectionID** and the **SpeechAliasID** fields within the **SpeechCmd** structure need not be specified by the client.

The ID of the added speech command is returned as the output parameter. The server returns an error if the client attempts to add a new speech command that has the same Name, culture and collection type as an existing speech command.

## Syntax

```

<CXSpeechCmdAdd>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <SpeechCmd>
    <CollectionType></CollectionType>
    <SpeechAliasCollectionArray>
      <SpeechAliasCollection>
      </SpeechAliasCollection>
    </SpeechAliasCollectionArray>
  </SpeechCmd>
</CXSpeechCmdAdd>

```

```
</CXSpeechCmdAdd>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

SpeechCmd

- The speech command to be added. Refer to [SpeechCmd](#) for details of this structure.

## Output Parameters

SpeechCmdID

- The ID of the newly created speech command.

## XML Return

```
<CXResponse>
  <SpeechCmdID>10007</SpeechCmdID>
</CXResponse>
```

## Errors

2147749804    Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147750858    Speech error: Name already used

- A Speech command with the same name, culture and collection type already exists.

## CXSpeechCmdUpdate

Updates the specified speech command.

```
Syntax
<CXSpeechCmdUpdate>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <SpeechCmd>
    <CollectionType></CollectionType>
```

```

        <SpeechAliasCollectionArray>
            <SpeechAliasCollection>
            </SpeechAliasCollection>
        </SpeechAliasCollectionArray>
    </SpeechCmd>
</CXSpeechCmdUpdate>

```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

SpeechCmd

- The speech command to be updated. Refer to [SpeechCmd](#) for details of this structure.

OldObject

- This optional field represents the baseline from which the update is being made. If populated, the difference between this SpeechCmd and the update SpeechCmd determines the updates that are applied.
- SpeechCmd
  - The baseline speech command structure. This has the same format as the SpeechCmd above.

## Output Parameters

None

## Errors

2147749804      Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147750858      Speech error: Name already used

- A Speech command with the same name, culture and collection type already exists.

## Comments

If **OldObject/SpeechCmd** is not populated, the update is based completely on the existing mailbox. If **OldObject/SpeechCmd** is populated, the update is based on the difference between **SpeechCmd** and **OldObject/SpeechCmd**. This is important when allowing multiple clients to edit the same mailbox such that changes from one client don't inadvertently overwrite changes by the other client.

## Example

The following example demonstrates adding a new Speech Alias phrase to recognize the phrase *Quality Assurance* to an existing speech command's US English Speech Alias collection that is named as *QA*.

```
<CXSpeechCmdUpdate>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
  <SpeechCmd>
    <SpeechCmdID>10007</SpeechCmdID>
    <CollectionType>Group</CollectionType>
    <SpeechAliasCollectionArray>
      <SpeechAliasCollection>
        <SpeechAliasCollectionID>22</SpeechAliasCollectionID>
        <CultureName>en-us</CultureName>
        <TTSNameAlias />
        <SpokenNameFile />
        <Name>QA</Name>
        <SpeechAliasArray>
          <SpeechAlias>
            <SpeechAliasID>373</SpeechAliasID>
            <Phrase>QA</Phrase>
          </SpeechAlias>
          <SpeechAlias>
            <Phrase>Quality Assurance</Phrase>
          </SpeechAlias>
        </SpeechAliasArray>
      </SpeechAliasCollection>
    </SpeechAliasCollectionArray>
  </SpeechCmd>
</CXSpeechCmdUpdate>
```

## CXSpeechCmdDelete

Delete a speech command.

## Syntax

```
<CXSpeechCmdDelete>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <SpeechCmdID></SpeechCmdID>
</CXSpeechCmdDelete>
```

## Input Parameters

SessionID



- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

SpeechCmdID

- The unique identifier of the speech command to be deleted.

## Output Parameters

None

## Errors

2147749804      Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

2147750865      Speech error: Collection ID missing

- The ID of the speech command was not specified.

## CXSpeechCmdNamesListByCollectionTypeGet

Retrieves the **SpeechCmdID** and Name fields for all the speech commands that belong to a particular speech alias collection type. This function is useful for populating a UI control that allows the user to select a speech command. Only the ID and name information is required when initially populating the control and this function supplies that information. The retrieval of the complete **SpeechCmd** structure is an expensive operation and should be deferred until the time when the user actually chooses the speech command to work with. This function helps build this necessary optimization.

## Syntax

```
<CXSpeechCmdNamesListByCollectionTypeGet>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <SpeechAliasCollectionType></SpeechAliasCollectionType>
</CXSpeechCmdNamesListByCollectionTypeGet>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

#### SpeechAliasCollectionType

- Specifies the type of the speech command. See `CollectionType` in the `SpeechCmd` section for valid collection types.

## Output Parameters

#### SpeechCmdNamesList

- List of one or more speech commands.
- `SpeechCmdNames`
  - This is a reduced view of the `SpeechCmd` structure. It only contains the ID of the speech command and the localized names.
  - `SpeechCmdID`
    - The ID of the speech command.
  - `NameArray`
    - The container that holds all the localized name information structures for this speech command.
  - `Name`
    - The structure that holds a single localized name information.

#### SpeechAliasCollectionID

- The identifier of the speech alias collection to which the localized name applies.

#### CultureName

- The identifier of the culture to which this structure applies.

#### Name

- The localized name of the speech command for the culture specified by **CultureName**.

## XML Return

```
<CXResponse>
  <SpeechCmdNamesList>
    <SpeechCmdNames>
      <SpeechCmdID>10004</SpeechCmdID>
      <NameArray>
        <Name>
          <SpeechAliasCollectionID>20</SpeechAliasCollectionID>
          <CultureName>en-us</CultureName>
          <Name>Sales</Name>
        </Name>
      </NameArray>
    </SpeechCmdNames>
  </SpeechCmdNamesList>
</CXResponse>
```

```

</SpeechCmdNames>
<SpeechCmdNames>
  <SpeechCmdID>10005</SpeechCmdID>
  <NameArray>
    <Name>
      <SpeechAliasCollectionID>21</SpeechAliasCollectionID>
      <CultureName>en-us</CultureName>
      <Name>Engineering</Name>
    </Name>
  </NameArray>
</SpeechCmdNames>
</SpeechCmdNamesList>
</CXResponse>

```

## Errors

2147749804    Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session

## CXMBGroupTypeListGet

Retrieves the list of the configured Group Types. Group Types define the categories under which Groups are defined. Example Group Types are Department and Location.

## Syntax

```

<CXMBGroupTypeListGet>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
</CXMBGroupTypeListGet>

```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

## Output Parameters

MBGroupTypeList

- List of all the Group Type configurations. The nested xml structure for this parameter has the following fields:
- MBGroupType
  - Container for a single Group Type configuration.
  - MBGroupTypeID
    - Unique identifier of the Group Type
  - Priority
    - The priority for the use of this Group Type for disambiguation
  - CreateGrammar
    - Whether the Group Type should be added to speech recognition grammars
  - SpeechCmdID
    - Unique identifier of the speech command associated with this group type.

## XML Return

```
<CXResponse>
  <MBGroupTypeList>
    <MBGroupType>
      <MBGroupTypeID>1</MBGroupTypeID>
      <Priority>1</Priority>
      <CreateGrammar>1</CreateGrammar>
      <SpeechCmdID>1</SpeechCmdID>
    </MBGroupType>
    <MBGroupType>
      <MBGroupTypeID>2</MBGroupTypeID>
      <Priority>2</Priority>
      <CreateGrammar>1</CreateGrammar>
      <SpeechCmdID>2</SpeechCmdID>
    </MBGroupType>
  </MBGroupTypeList>
</CXResponse>
```

## Errors

2147749804    Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session

## CXMBGroupListGet

Retrieves the list of Groups assigned to a specific Group Type.

## Syntax

```
<CXMBGroupListGet>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <MBGroupTypeID></MBGroupTypeID>
</CXMBGroupListGet>
```

## Input Parameters

### SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

### SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

### MBGroupTypeID

- A unique identifier that identifies the Group Type to be enumerated. Group Type IDs are returned by a call to **CXMBGroupTypeListGet**.

## Output Parameters

### MBGroupList

- List of all the Group configurations. The nested xml structure for this parameter has the following fields:
- MBGroup
  - Container for a single Group configuration
  - MBGroupID
    - Unique identifier of the Group
  - MBGroupTypeID
    - A unique identifier that identifies the Group Type that the Group is a member of.
  - SpeechCmdID
    - Unique identifier of the speech command associated with this group.

## XML Return

```
<CXResponse>
  <MBGroupList>
    <MBGroup>
      <MBGroupID>1</MBGroupID>
      <MBGroupTypeID>1</MBGroupTypeID>
```

```

    <SpeechCmdID>10004</SpeechCmdID>
  </MBGroup>
  <MBGroup>
    <MBGroupID>2</MBGroupID>
    <MBGroupTypeID>1</MBGroupTypeID>
    <SpeechCmdID>10090</SpeechCmdID>
  </MBGroup>
</MBGroupList>
</CXResponse>

```

## Errors

2147749804      Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session

## CXListFirst

**NOTE** **CXListFirst**, **CXListNext**, and **CXListTotal** will be deprecated in the future and will be replaced by other functions.

Lists information about various categories, such as administrators, distribution list members, email servers, and time zones..Since the number of items in each category can vary widely, use this function to retrieve the first Count items in the list and the CXListNext function to retrieve the remaining items in the list.

## Syntax

```

<CXListFirst>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <ListType></ListType>
  <MailboxType></MailboxType>
  <Directory></Directory>
  <ParentMailbox></ParentMailbox>
  <Count></Count>
</CXListFirst>

```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to CXLogon.

## ListType

- The type of list generated. Depending upon the type of list, additional elements may be included that affect the XML output.
  - *Administrator* – Lists administrator accounts. Information includes access-level assignments for each account.
  - *AltExt* – Lists the alternative extensions for the mailbox identified by ParentMailbox
  - *AnswerMode* – Lists the answer modes available
  - *AnswerModeLine* – Lists the lines available
  - *AnswerModeSchedule* – Lists the scheduled answer mode entries
  - *CallRoutingEntry* – Lists the call routing entries
  - *Contact* – Lists any contacts for the Subscriber identified by ParentMailbox.
  - *DistListWithMembership* – Lists the distribution lists to which the mailbox identified by ParentMailbox belongs
  - *DLMember* – Lists the member of the distribution list mailbox identified by ParentMailbox
  - *EmailAddrSearch* – Lists email addresses.
  - *EmailServer* – Lists the email servers
  - *Language* – Lists the available languages
  - *MailboxReference* – Lists all references, such as membership in a distribution list and sponsored mailboxes associated with the mailbox identified by ParentMailbox
  - *MailboxSummary* – Lists a summary of the mailboxes. MailboxType can also be specified.
  - *NetDirEntry* – Lists the Remote Directory entries associated with a Network or Digital Network mailbox identified by ParentMailbox
  - *SMSProvider* – Lists the SMS providers
  - *SwitchSection* – Lists the switches on the system
  - *TimeZone* – Lists the defined time zones

## MailboxType

- The type of mailbox listed. If this value is not supplied or is not one of the following values, all mailboxes are listed.
  - *All* (default) – All mailboxes are listed
  - *CallProcessor* – Lists only Call Processor mailboxes
  - *DigitalNetworking* – Lists only Digital Networking mailboxes
  - *DistributionList* – Lists only Distribution List mailboxes
  - *FaxDelivery* – Lists only Fax Delivery mailboxes
  - *Interactive* – Lists only Interactive mailboxes
  - *LocalAlias* – Lists only Local Alias mailboxes

- *Messagable* – Lists all mailboxes to which a message may be addressed, such as Subscriber, Distribution List, AMIS Networking, Digital Networking, Interactive, Local Alias, and Network. This element is only valid when the ListType element is MailboxSummary.
- *Subscriber* – Lists only Subscriber mailboxes
- *Visitor* – Lists only Visitor mailboxes

#### Directory

- Optional. Whether all Messagable mailboxes are listed or only mailboxes that have the property *Include in Subscriber Directory* set are listed. This element is only valid when the ListType element is **MailboxSummary** and the MailboxType element is **Messagable**..

#### ParentMailbox

- The mailbox number for the associated mailbox when required by the **ListType**

#### Count

- The maximum number of records returned. If there are fewer than Count records found, all records are returned. If there are more than Count records found, use **CXListNext** to enumerate the remaining records.

## Output Parameters

Output Parameters vary widely depending upon the list type and options. A **MailboxSummary** of a Subscriber mailbox returns the following.

#### ListData

- The information associated with the list
- AltExt
  - An alternative extension entry. There can be more than one of these elements.
  - PhoneNumber
    - The alternative telephone number associated with the mailbox
  - Type
    - The type of telephone number
  - Comment
    - Any comment associated with the telephone number

#### Found

- The number of records in the list

## XML Return

```
<ListData>
  <AltExt>
    <PhoneNumber></PhoneNumber>
```



```

        <Type></Type>
        <Comment></Comment>
    </AltExt>
</ListData>
<Found></Found>

```

## Errors

2147749804      Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

## Examples

The following example returns summary information on the first 50 mailboxes configured on the system.

```

<CXListFirst>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
  <ListType>MailboxSummary</ListType>
  <MailboxType>All</MailboxType>
  <Count>50</Count>
</CXListFirst>

```

The following example returns the alternative extensions for the mailbox with number 2002.

```

<CXListFirst>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
  <ListType>AltExt</ListType>
  <MailboxType>Subscriber</MailboxType>
  <ParentMailbox>2002</ParentMailbox>
  <Count>10</Count>
</CXListFirst>

```

The following example lists the members for the distribution list with mailbox number 2100.

```

<CXListFirst>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
  <ListType>DLMember</ListType>
  <MailboxType>Subscriber</MailboxType>
  <ParentMailbox>2100</ParentMailbox>
  <Count>50</Count>
</CXListFirst>

```

The following example lists the email servers.

```

<CXListFirst>
  <SessionID>1234567890</SessionID>

```

```
<SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
<ListType>EmailServer</ListType>
<Count>50</Count>
</CXListFirst>
```

The following example lists the defined time zones.

```
<CXListFirst>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
  <ListType>TimeZone</ListType>
  <Count>50</Count>
</CXListFirst>
```

The following example lists the available languages.

```
<CXListFirst>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
  <ListType>Language</ListType>
  <Count>50</Count>
</CXListFirst>
```

The following example lists the SMS providers.

```
<CXListFirst>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
  <ListType>SMSProvider</ListType>
  <Count>50</Count>
</CXListFirst>
```

## See Also

[CXListNext](#), [CXListTotal](#)

## CXListNext

**NOTE** **CXListFirst**, **CXListNext**, and **CXListTotal** will be deprecated in the future and will be replaced by other functions.

Returns the next group of records as defined by a previous CXListFirst call.

## Syntax

```
<CXListNext>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
```

```
<Count></Count>
</CXListNext>
```

## Input Parameters

SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

Count

- The maximum number of records returned. If there are fewer than **Count** records found, all records are returned.

## Output Parameters

See CXListFirst for an example of the Output Parameters.

## Errors

2147749804     Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

## See Also

[CXListFirst](#)

## CXListTotal

**NOTE** **CXListFirst**, **CXListNext**, and **CXListTotal** will be deprecated in the future and will be replaced by other functions.

Returns the total number of elements in the specified list.

## Syntax

```
<CXListTotal>
  <SessionID></SessionID>
  <SessionInstanceID></SessionInstanceID>
  <ListType></ListType>
```

```
<ParentMailbox></ParentMailbox>  
</CXListTotal>
```

## Input Parameters

### SessionID

- Identifies the session. This value is returned by a previous call to **CXLogon**.

### SessionInstanceID

- A unique identifier that identifies the server instance. This value is returned by a previous call to **CXLogon**.

### ListType

- The type of list for which the total is generated.
  - *Administrator* – Returns the number of administrators
  - *AltExt* – Returns the number of alternative extensions for the mailbox identified by ParentMailbox
  - *AnswerMode* – Returns the number of answer modes available
  - *AnswerModeLine* – Returns the number of lines available
  - *AnswerModeSchedule* – Returns the number of scheduled answer mode entries
  - *CallRoutingEntry* – Returns the number of call routing entries
  - *Contact* – Returns the number of contacts for the Subscriber identified by ParentMailbox.
  - *DistListWithMembership* – Returns the number of distribution lists to which the mailbox identified by ParentMailbox belongs
  - *DLMember* – Returns the number of members of the distribution list mailbox identified by ParentMailbox
  - *EmailAddrSearch* – Returns the number of email addresses.
  - *EmailServer* – Returns the number of email servers
  - *Language* – Returns the number of available languages
  - *MailboxReference* – Returns the number of references, such as membership in a distribution list, to the mailbox identified by ParentMailbox
  - *MailboxSummary* – Returns the number of mailboxes. MailboxType can also be specified.
  - *NetDirEntry* – Returns the number of networks or VPIMs identified by ParentMailbox
  - *SenecaCOS* – Returns the number of Seneca servers
  - *SMSProvider* – Returns the number of SMS providers
  - *SwitchSection* – Returns the number of switches on the system
  - *TimeZone* – Returns the number of defined time zones

### ParentMailbox

- The mailbox number for the associated mailbox when required by the **ListType**

## Output Parameters

### Count

- The number of items in the list

## XML Return

```
<Total></Total>
```

## Errors

2147749804    Session is Invalid

- The SessionID provided is invalid, the session has been previously logged off, or the server has timed out of the session.

## Example

The following example returns the number of members in the Distribution List identified by mailbox number 2100.

```
<CXListTotal>
  <SessionID>1234567890</SessionID>
  <SessionInstanceID>12345678-abcd-1234-abcd-1234567890ab</SessionInstanceID>
  <ListType>DLMember</ListType>
  <ParentMailbox>2100</ParentMailbox>
</CXListTotal>
```

# MiCollab AM Web Services API Structures

## MBCommon

The following table describes the mailbox properties that are common to most mailbox types. These are part of the MB/MBCommon structure.

Table 12. MBCommon mailbox properties

Field Name	Default	Valid Data	Comment
MBID	N/A	Digits	The mailbox number. Mailboxes must be of the system mailbox length, with leading zeros if necessary. Use <b>CXMailboxDefaults</b> to determine the system mailbox length (returned as <b>MailboxIDLength</b> ). System mailbox length is always between 2 and 10 digits.
DisplayName	Empty	Text	The display name associated with the mailbox. Can be up to 45 characters.
FirstName	Empty	Text	The first name associated with the mailbox. Can be up to 25 characters.
MiddleName	Empty	Text	The middle name associated with the mailbox. Can be up to 25 characters.
LastName	Empty	Text	The last name associated with the mailbox. Can be up to 25 characters.
MBType	N/A	A – Subscriber C – Fax Center	The mailbox type.

*D* – Distribution List  
*F* – Local Alias  
*L* – Announcement  
*N* – Network  
*P* – Digital Networking  
*S* – Class of Service  
*V* – Visitor  
*Y* – Fax Delivery  
*Z* – AMIS networking  
 2 – Message Center  
 3 – Call Processor  
 4 – Interactive  
 5 – Outbound

AllowICA	0	0, 1	Whether the subscriber has ICA or WPM message access enabled.
NameMediaFileID	Empty	GUID	Name of file containing the Subscriber's recorded name.
PromptLangID	255	Valid Prompt Language ID. Installed prompt languages may be retrieved using <b>CXMailboxDefaults</b> .	Identifies the prompt language. This is a read-only field. The value of this field is derived from the <b>LanguagePackID</b> field. A default value of 255 specifies that the system default Prompt Language is used.
SponsorMBID	Empty	Mailbox number	The mailbox number of the sponsor mailbox.
SwitchSectionID	First Switch Section ID	Valid Switch Section ID. Switch Section IDs may be enumerated using CXListFirst with a ListType of <i>SwitchSection</i>	The ID of the Switch Section to which this mailbox is associated.
TTSNameAlias	Empty	Text	Text to speech name. Can be up to 45 characters.
VMDir	1	0, 1	When set, Subscriber is listed in the voice mail directory.

WasPropagated	0	0, 1	Whether the mailbox was propagated from a remote correspondent. This only applies to Subscriber and Distribution List mailboxes.
Propagate	0	0, 1	When set, mailbox is propagated to the network directory. This only applies to Subscriber and Distribution List mailboxes.
LanguagePackID	0	Valid Language Pack ID. Installed language packs may be retrieved using <b>CXLanguagePackListGet</b> .	Identifies the language pack that is to be used for this mailbox.

## MBSubSpecific

The following table describes the subscriber mailbox properties. These are part of the MB/MBTypeSpecific/MBSubSpecific structure.

Table 13. MBSubSpecific subscriber mailbox properties

Field Name	Default	Valid Data	Comment
CDAUSTimeZone	0	Number	A valid time zone ID. The list of time zones can be enumerated using <b>CXListFirst</b> with a ListType of <i>TimeZone</i> . Default = 0.
COSMBID	Empty	Mailbox number	The mailbox number of the Class of Service to which the mailbox is assigned.
EnableCondition	Yes	No, Yes, <i>AfterSetupTutorial</i>	Specifies if and when the mailbox is enabled.
SetupRequired	1	0, 1	Whether the subscriber is prompted through setup.
PersonalAssistanceOn	0	0, 1	When set, personal assistance features are enabled.
AcceptMsgCondition	<i>WhenEnabled</i>	<i>Never, WhenEnabled, AfterSetupTutorial</i>	Specifies the required state of the mailbox before it can receive messages.



PasswordResetStatus	OK	OK, ExpiredGracePeriod, Expired, ForceReset	Specifies whether the subscriber is forced to reset his password on the next logon, and the reason why.
UseSystemSecurityPolicy	0	0, 1	When set, advanced security policy is in effect.
AllowPasswordReset	1	0, 1	When set, allows the subscriber to reset the password using a client application like Web PhoneManager.
EmailAddress	Empty	Valid email address	The email address at which the subscriber receives business related messages. It is used by the System to send messages like the Security Code Reset Message.
<b>Language Properties</b>			
CallerLangHandling	None	None, Messaging, Persists	The circumstances under which the subscriber's caller hears voice prompts in the language selected in the subscriber's mailbox.
TTSLangID	0	Valid Locale ID (LCID). Example: 1033 for US English.	The language used for TTS. This is a read-only field. The value of this field is derived from the <b>MBCommon/LanguagePackID</b> field. A value of 0 specifies that the system default language is used.
ClientCodePage	1033	Valid Locale ID (LCID). Example: 1033 for US English.	The language in which this subscriber sees certain elements of Unified Messaging that are generated by the server.
<b>Answering Properties</b>			
PersonalOPPhone	Empty	Digits	Telephone number of the subscriber's personal operator.
HoldResponse	N	N – None Q – Queue A – Announce	Specifies what hold actions are to take place when the subscriber's line is busy.
HoldAnncMBID	Empty	Mailbox number	Mailbox number of hold announcement mailbox.

HoldInterval	15	1 – 300	Hold retry interval.
HoldAttempts	4	1- 99	Maximum hold retry attempts.
GenGreetType	<i>System</i>	<i>None, Announcement, System</i>	Type of greeting introduction to use.
AlwaysPlayGenGreet	0	0, 1	If set, the general greeting is always played.
GenGreetAnncMBID	Empty	Mailbox number	Mailbox number of general greeting announcement mailbox.
TakeMsgDuringOOF	1	0, 1	When set, messages are taken when out of office is enabled.
AllowOOF	1	0, 1	Whether the subscriber is allowed to activate out of office.
OOFOn	0	0, 1	Whether the subscriber has activated out of office.
OOFIntroType	<i>Default</i>	<i>None, Announcement, Default</i>	Type of announcement to play for the out of office greeting introduction.
OOFIntroAnncMBID	Empty	Mailbox number	Mailbox number of out of office greeting announcement mailbox.
DisableDTMFDuringPersonalGreet	<i>DuringOOF</i>	<i>Never, Always, DuringOOF</i>	Specified the conditions under which DTMF input is disabled when a personal greeting is being played.
ESPOn	0	0, 1	Whether extension-specific call processing is enabled.
GreetBeforeESPCP	0	0, 1	When set, the ESP greeting is active.
ESPCPMBID	Empty	Mailbox number	Mailbox number of the ESP Call Processor.
BusyESPCPMBID	Empty	Mailbox number	Mailbox number of the ESP Busy Call Processor.
UseSubTUIMenuOnFwd	0	0, 1	When set, the subscriber interface DTMF command keys is made available to callers forwarded for message taking.

Auto Attendant and Call Handling and other Features Properties			
AAAllowCallScreen	0	0, 1	Whether the subscriber is allowed to activate and use call screening.
AACallScreenOn	0	0, 1	Whether the subscriber has activated call screening.
AAAllowCallBlock	0	0, 1	Whether the subscriber is allowed to activate and use call blocking.
AACallBlockOn	0	0, 1	Whether the subscriber has activated call blocking.
NetAccessPriority	<i>Normal</i>	<i>Normal, None, Priority</i>	Allowed priority level for sending network messages.
AllowExtCall	1	0, 1	When set, callouts to extensions are allowed.
AllowLocalCall	1	0, 1	When set, callouts to local numbers are allowed.
TypesOfLDCallsAllowed	<i>No</i>	<i>Yes, No, Specific</i>	Type of long distance callouts allowed.
LDAreaCode1	Empty	Digits	A specific area code to which callouts may be made.
LDAreaCode2	Empty	Digits	A specific area code to which callouts may be made.
LDAreaCode3	Empty	Digits	A specific area code to which callouts may be made.
AADir	1	0, 1	When set, subscriber is listed in the auto attendant directory.
PurgePersonalGreetDaily	0	0, 1	When set, the subscriber's personal greeting is purged daily.
AllowMsgToDistList	1	0, 1	When set, subscriber is allowed to address messages to distribution lists.
AAXferType		<i>B – Blind</i> <i>M – Monitored</i> <i>T – Transfer</i> <i>C – Confirmed</i>	The type of auto attendant transfers to be used when calling the subscriber.

AAXferRingsToWait	Default comes from the system configuration	1 - 9	Number of rings to wait before inferring a ring-no-answer condition.
AAXferAnncType	<i>Short</i>	<i>Long, Short, None</i>	The type of auto attendant screened call transfer announcement to play.
AAPlayRecordingHelpPrompt	0	0, 1	When set, the recording help prompt is played when the auto attendant takes a message.
SendOriginalAttachmentsWithReply	0	0, 1	When set, the original message is included when the subscriber sends a reply.
AllowTrunkToTrunkReply	0	0, 1	When set, the subscriber is allowed to use the Live Reply feature with telephone numbers outside the local system.
MsgRetentionPeriodDays	Default comes from the system configuration	1 – 99	Specifies how long, in days, that messages are retained. A value of 255 specifies unlimited retention.
MaxNonSubMsgInboxLimit	0	0 – 32767	Specifies the maximum number of messages from outside callers that the subscriber mailbox can hold. A value of 0 specifies unlimited.
MaxNonSubMsgSeconds	0	0 – 2700	Specifies the maximum length, in seconds, or messages from outside callers. A value of 0 specifies that the system default will be used.
<b>Message Waiting Indicator Properties</b>			
MWIClearType	<i>F</i>	<i>E</i> – Empty <i>F</i> – First <i>L</i> – All	Specifies the MWI clear mode.
MWILogicallyOn	0	0, 1	Whether the MWI is on.
<b>Email Access and Store Properties</b>			
TUIMsgTypeFilter/Email	1	0, 1	Whether the subscriber is allowed email access over the telephone.

EmailAccessType	None	None, Access, Store	None – Voicemail/Fax stored locally. No email access. Access – Voicemail/Fax stored locally with email access via external store. Store – Voicemail/fax stored in external store. Includes access to emails.
MsgEnumLimitOn	0	0, 1	Whether partial message enumeration is enabled. This feature can only be enabled when using an external store.
MsgEnumLimitUnitType	Days	MsgCount, Days	Message enumeration unit type.
MsgEnumLimit	10	Number > 0	The number of messages or days of messages to enumerate.
ICAREplyAddr	Empty	Valid email address	When using ICA, the reply-to-address.
<b>Simple UM</b>			
AllowSimpleUM	0	0, 1	When set, subscriber is allowed to enable or disable simple UM.
SimpleUMOn	0	0, 1	When set, simple UM is enabled.
SimpleUMSMTPProviderVisible	0	0, 1	When set, subscriber is allowed to view and set the provider.
CDAUSSimpleUMSMTPProvider	0	Valid SMS Provider ID. Use the <b>CXListFirst</b> function with a ListType of <i>SMSPProvider</i> to enumerate SMS providers that are configured.	The ID of the SMS provider.
SimpleUMSMTPAddrVisible	0	0, 1	When set, subscriber is allowed to view and set the SMTP address.
SimpleUMSMTPAddr	Empty	Valid email address	SMTP address to which the simple UM message is sent.
SimpleUMSendAttachmentsVisible	0	0, 1	When set, subscriber is allowed to view and set the Attach Voice checkbox.

SimpleUMSendAttachments	0	0, 1	When set, the voice mail recording is included in the simple UM message as an attachment.
-------------------------	---	------	---

Availability Properties			
AllowProfiles	0	0, 1	When set, availability processing is allowed for the subscriber.
ProfilesEnabled	0	0, 1	When set, availability processing is enabled for the subscriber.
LocateMode	C	S – Silent C – Confirm O – Confirm Once A – Announce X – Announce with Cancel	Specifies the form prompts played to a caller when locating the subscriber.
ProfileAutoSourceType	None	None, ClassOfService, Subscriber	Used to restrict the mailbox types for ProfileAutoSourceMBID in the UI.
ProfileAutoSourceMBID	Empty	Mailbox number	Mailbox used as a template for Availability settings.
Speech Properties			
VUIStyle	None	None, SubscriberSession, CallCompletion, Full	Type of Voice User Interface enabled for the subscriber.
VUIStyleVisible	0	0, 1	When set, allows the subscriber to change the <b>VUIStyle</b> field for enabling/disabling Speech for the Subscriber Session.
AllowCallback	0	0, 1	When set, subscriber session call back is enabled.
AllowCallRecording	0	0, 1	When set, call recording is allowed.
AllowTotalHandsFree	0	0, 1	When set, total hands free feature is enabled.
AllowCalendaring	0	0, 1	When set, calendar access is enabled.
WhisperCallWaitingOn	0	0, 1	When set, whisper call waiting is

			enabled.
BargeInDisabled	0	0, 1	When set, speech recognition barge in is disabled.
RecordMissedCallsOn	0	0, 1	When set, missed call recording is enabled.
ContactsConfirmBeforeDialing	0	0, 1	When set, contacts are confirmed prior to dialing.
ContactsRefreshMode	<i>Disable</i>	<i>Disable, Automatic, Nightly</i>	Refresh mode for contacts.
ContactStoreType	<i>Local</i>	<i>Local, External</i>	Type of contact store in use.
<b>Immediate Message Notification Properties</b>			
AllowIMN	1	0, 1	Whether the subscriber is allowed to activate and use Immediate Message Notification.
IMNOn	0	0, 1	Whether the subscriber has activated Immediate Message Notification.
IMNForSpecificSender	0	0, 1	Whether notification of messages from a specific sender is enabled.
IMNSpecificSenderMBID	Empty	Mailbox number	Mailbox number of the specific sender for which message notifications are to be made.
IMNMsgTypeFilter/Voice	1	0, 1	Whether notifications are sent when voice messages are received by the subscriber.
IMNMsgTypeFilter/Fax	0	0, 1	Whether notifications are sent when fax messages are received by the subscriber.
IMNMsgTypeFilter/Email	0	0, 1	Whether notifications are sent when email messages are received by the subscriber.
IMNStartTOD	<i>0000</i>	HHMM	Starting time for the when notification is enabled.
IMNStopTOD	<i>2359</i>	HHMM	Ending time for the when notification is enabled.

IMNDOW/Sunday	0	0, 1	Whether message notification is enabled for Sundays.
IMNDOW/Monday	1	0, 1	Whether message notification is enabled for Mondays.
IMNDOW/Tuesday	1	0, 1	Whether message notification is enabled for Tuesdays.
IMNDOW/Wednesday	1	0, 1	Whether message notification is enabled for Wednesdays.
IMNDOW/Thursday	1	0, 1	Whether message notification is enabled for Thursdays.
IMNDOW/Friday	1	0, 1	Whether message notification is enabled for Fridays.
IMNDOW/Saturday	0	0, 1	Whether message notification is enabled for Saturdays.
IMNAllPriorities	<i>All</i>	<i>Urgent, All</i>	Whether to send a message notification for urgent messages only, or for all messages.
IMNBusyAttempts	Default comes from the system configuration	0-99	The number of times that the server tries to reach a number that was busy before trying the next telephone number in the sequence.
IMNBusyWait	Default comes from the system configuration	0-99	The number of minutes that the server waits before retrying a notification number that was busy.
IMNTraverseListCount	1	1-99	The number of times that the server goes through the list trying to deliver the message notification.
<b>Daily Message Reminder Properties</b>			
AllowDMR	1	0, 1	Whether the subscriber is allowed to activate and use Daily Message Reminder.
DMROn	0	0, 1	Whether the subscriber has activated the Daily Message Reminder.
DMRTOD	<i>0000</i>	HHMM	The time at which the daily reminder call is made.



DMRPhone	Empty	Phone number	The phone number to call to deliver the daily message reminder.
<b>SMS Notification Properties</b>			
AllowSMS	0	0, 1	Whether the subscriber is allowed to activate and use SMS notification.
SMSOn	0	0, 1	Whether the subscriber has activated SMS notification.
SMSMsgTypeFilter/Voice	1	0, 1	Whether SMS notifications are sent when voice messages are received by the subscriber.
SMSMsgTypeFilter/Fax	1	0, 1	Whether SMS notifications are sent when fax messages are received by the subscriber.
SMSStartTOD	0000	HHMM	Starting time for the when notification is enabled.
SMSStopTOD	2359	HHMM	Ending time for the when notification is enabled.
SMSDOW/Sunday	0	0, 1	Whether SMS notification is enabled for Sundays.
SMSDOW/Monday	1	0, 1	Whether SMS notification is enabled for Mondays.
SMSDOW/Tuesday	1	0, 1	Whether SMS notification is enabled for Tuesdays.
SMSDOW/Wednesday	1	0, 1	Whether SMS notification is enabled for Wednesdays.
SMSDOW/Thursday	1	0, 1	Whether SMS notification is enabled for Thursdays.
SMSDOW/Friday	1	0, 1	Whether SMS notification is enabled for Fridays.
SMSDOW/Saturday	0	0, 1	Whether SMS notification is enabled for Saturdays.
SMSAllPriorities	<i>Urgent</i>	<i>Urgent, All</i>	Whether to send an SMS notification for urgent messages only, or for all messages.

CDAUSSMSProvider	0	Valid SMS Provider ID. Use the <b>CXListFirst</b> function with a ListType of <i>SMSPProvider</i> to enumerate SMS providers that are configured.	The ID of the SMS provider.
SMSMobileNo	Empty	Phone Number	The telephone number to which a notification is sent when the SMS provider is an SMS message type.
SMSSMTPAddr	Empty	SMTP email address	The SMTP email address to which a notification is sent when the SMS provider is an SMTP message type.
SMSSendAttachments	0	0, 1	Whether the voice message recording is attached to the message. Applies to SMTP email notifications only.
PrimaryMessageTemplate	Empty	UTF-8	Name of the primary message template.
<b>Message Presentation Properties</b>			
PresentMsgsByType	0	0, 1	Whether messages are separated by type (voice, fax, email). If false (0) all message types are intermingled.
AutoEnvelopeOccurrence	<i>None</i>	<i>None, Before, After</i>	Whether the envelope information plays automatically, and if so, whether it plays before or after the message is played.
AutoEnvelopePresentation	<i>DateTimeOnly</i>	<i>Complete, DateTimeOnly</i>	Whether only data and time information is presented, or the complete envelope is presented.
MsgRetrievalFIFO	0	0, 1	When partial enumeration is enabled, whether the sort order is FIFO. If false (0), LIFO is used.
AutoPlayFirstMsg	0	0, 1	Whether the subscriber can hear the first new message immediately after logging on.
PresentUrgentMsgsFirst	0	0, 1	Whether the subscriber can hear

			urgent messages first when playing messages. If Partial enumeration is enabled, sorting urgent first is not allowed and this field is set to a value of 0.
PlayBackSpeed	0	-4 to +4	Relative playback speed.
PlayBackSpeedPersistence	S	S – Session Only P – Persist All Sessions M – Message Only	The type of playback speed persistence.
TUIStyle	OriginalAltAddr	Original, OctelAria, Centigram, OctelSerenade, Audix, OriginalAltAddr, OctelSerenadeAlt, OctelAriaAlt, MeridianMail, Kinesis, Repartee	TUI style for the Subscriber.
<b>Message Forwarding Properties</b>			
AllowAFwd	1	0, 1	Whether the subscriber is allowed to activate and use message forwarding.
AFwdOn	0	0, 1	Whether the subscriber has activated message forwarding.
AFwdForSpecificSender	0	0, 1	Whether the subscriber has activated message forwarding for messages received only from a specific sender.
AFwdSpecificSenderMBID	Empty	Mailbox number	The mailbox number of the specific sender whose messages are to be forwarded.
AFwdMsgTypeFilter/Voice	1	0, 1	Whether message forwarding is performed when voice messages are received by the subscriber.
AFwdMsgTypeFilter/Fax	0	0, 1	Whether message forwarding is performed when fax messages are received by the subscriber.
AFwdStartTOD	0000	HHMM	Starting time for the when message

			forwarding is enabled.
AFwdStopTOD	2359	HHMM	Ending time for the when message forwarding is enabled.
AFwdDOW/Sunday	0	0, 1	Whether message forwarding is enabled for Sundays.
AFwdDOW/Monday	1	0, 1	Whether message forwarding is enabled for Mondays.
AFwdDOW/Tuesday	1	0, 1	Whether message forwarding is enabled for Tuesdays.
AFwdDOW/Wednesday	1	0, 1	Whether message forwarding is enabled for Wednesdays.
AFwdDOW/Thursday	1	0, 1	Whether message forwarding is enabled for Thursdays.
AFwdDOW/Friday	1	0, 1	Whether message forwarding is enabled for Fridays.
AFwdDOW/Saturday	0	0, 1	Whether message forwarding is enabled for Saturdays.
AFwdAllPriorities	<i>All</i>	<i>Urgent, All</i>	Whether to message forwarding applies to urgent messages only, or for all messages.
AFwdTargetMBID	Empty	Mailbox number	The mailbox number of the mailbox to which messages are to be forwarded.
<b>Fax Properties</b>			
FaxPrintPhone	Empty	Phone Number	The telephone number or extension to which the subscriber can print fax messages by default.

## MDBListSpecific

The following table describes the Distribution List Mailbox properties. These are part of the MB/MBTypeSpecific/MBDistListSpecific structure.

Table 14. Distribution List Mailbox Properties

Field Name	Default	Valid Data	Comment
DigitalNetMBID	Empty	Mailbox number	The associated Digital Networking mailbox, identifying the remote correspondent server from which this Distribution List was propagated.
AcceptMsgFromType	<i>Anyone</i>	<i>Anyone, Sponsor, Membership, Specified</i>	The category of subscriber from which messages addressed to this Distribution List will be accepted.
SuppressNotificationSMS	0	0, 1	When set, SMS notifications are suppressed.
SuppressNotificationMWI	0	0, 1	When set, MWI notifications are suppressed.
SuppressNotificationIMN	0	0, 1	When set, IMN notifications are suppressed.
SuppressNotificationAFwd	0	0, 1	When set, auto message forwarding is suppressed.

## MBCPSpecific

The following table describes the Call Processor mailbox properties. These are part of the MB/MBTypeSpecific/MBCPSpecific structure.

Table 15. Call Processor Mailbox Properties

Field Name	Default	Valid Data	Comment
TimeoutSec	3	0 to 9	Length of time in seconds the system waits after the Call Processor's instruction announcement ends before invoking the Timeout action.

UseAMCPOpXferAction	1	0, 1	When set, callers that press 0 in this Call Processor mailbox invoke the 0 key action defined in the Answer Mode Call Processor mailbox.
MaxMsgRecLengthSec	2700	1 to 2700	Length of a non-subscriber message when this Call Processor mailbox has control of the call.
NextCPMBID	Empty	Mailbox number	Mailbox number of a Call Processor mailbox to which the system routes a caller after a previously selected action in the Call Processor mailbox that initiated the action has completed.  When set to <i>J</i> , Go to Answer Mode will be enabled.
TwoPartGreetOn	0	0, 1	When set, the recording and the playback of two separate announcement records (Introduction and Instruction) is allowed.
UseSpeechRecognitionTimeout Rules	1	0,1	Defines how the system treats a call when a speech command is not given. When set, multiple timeouts can occur before going to the Final Timeout action of the Call Processor. Otherwise, the traditional one timeout occurrence triggers the Timeout action.
NoASRFallbackCPMBID	Empty	Mailbox number	Mailbox number of a Call Processor mailbox. When a Call Processor mailbox that is speech enabled is unable to acquire an ASR Speech license it passes control to the Call Processor defined in this field.
AlwaysConfirmNames	0	0, 1	When set, the system requires the caller to confirm the recognition match for any subscriber name spoken in this Call Processor.
ASRNoMatchMax	0	0, 9	The maximum number of consecutive unsuccessful attempts while matching a speech command to the caller's spoken request. Once this maximum is reached, the system executes the Final

No Match action.

ASRMismatchMax	0	0, disabled and 1 to 9	The maximum number of consecutive times the caller rejects the recognition result after answering the confirmation dialogue. Once this maximum is reached, the system executes the Final Mismatch Recognition action.
----------------	---	------------------------	---

## Device

The following table describes the Device structure.

Table 16. Device Structure

Field Name	Default	Valid Data	Comment
Name	Auto-generated based on the DevUsageType	Text	Name of the device.
MBID	N/A	Valid mailbox number	Mailbox number of the associated mailbox.
RemoteMBNo	Empty	Valid mailbox number	Mailbox number of the remote mailbox to which the device is associated. This parameter applies to devices related to a network node mailbox.
IsPrimary	0	0, 1	Indicates that the device is the primary device of the device usage type.
MwiCapable	0	0, 1	Whether the server will set and clear the message waiting indicator on the device.
WasPropagated	0	0, 1	Whether the device was propagated from a remote network node.
DevID	Auto-generated	Numeric	The unique identifier of the device.

DevAddr	N/A	Phone Number	The device telephone number.
DevUsageType	N/A	<i>Other</i> <i>CompanyMobile</i> <i>PersonalMobile</i> <i>RemoteOffice</i> <i>HomeNumber</i> <i>Extension</i> <i>Temporary</i>	This string enumeration identifies the usage type of the device.
DevAddrModifier	Empty	Numeric	The SMDI Prefix of an extension type device.
DevExternalAddr	Empty	Numeric	The DID, DIL or other external number. Generally, PBX extension devices may also have an external address.
SwitchSectionID	First Switch Section ID	Valid Switch Section ID. Switch Section IDs may be enumerated using <b>CXListFirst</b> with a ListType of <i>SwitchSection</i>	The ID of the Switch Section to which this device belongs.
IsSMDI	0	0, 1	When set, indicates that the device is an extension for a PBX which requires an SMDI prefix.
RecognitionSensitivity	0	-3 to +5	The sensitivity setting for speech recognition barge in. A positive value increases the sensitivity for situations such as a device that has a highly attenuated signal. A negative value decreases sensitivity for such situations such as the device being in an environment with



background noise.

RingTimeoutSec	14	Numeric	The timeout to wait for a ring no answer condition when calling the device. The value of this property should be long enough for the device to ring a sufficient number of times but should be less than the time at which the device forwards elsewhere such as a mobile service provider's voice mail system.
Shared	Calculated	0, 1	When set, indicates that the device is shared with another user.
Enabled	1	0, 1	When set, indicates that the device is active and that behaviors defined by the various device properties are in effect.
IsOfficeDefault	0	0, 1	When set, indicates that the device is the primary office device. Usually, this identifies the user's primary office extension device
IsMobileDefault	0	0, 1	When set, indicates that the device is the primary mobile device. Set this flag for the mobile device to which most business calls will be directed
DerivedDeviceTypeEnum	N/A	<i>VoiceCallLogon</i> – Phone: logon, can receive calls	This string enumeration specifies the general type of the device. This determines whether

<i>VoiceCallAutoLogon</i> – Phone: auto logon, can receive calls	the device is a Voice or a Fax device, whether logon or auto-logon is allowed from this device and whether this device can receive calls.
<i>VoiceNoCallLogon</i> – Phone: logon, cannot receive calls	
<i>VoiceNoCallAutoLogon</i> – Phone: auto logon, cannot receive calls	
<i>VoiceCallNoLogon</i> – Phone: no logon, can receive calls	
<i>VoiceCallOperator</i> – Phone: operator, can receive calls	
<i>VoiceNoCallOperator</i> – Phone: operator, cannot receive calls	
<i>FaxCallAutoLogon</i> – Fax	

## Address

The Address structure is used within various Messaging functions, both as parameters to functions like **CXMessageSend** and also as embedded structures within the Message structure (as Sender and Recipient structures).

The following table describes the fields of the Address structure.

Table 17. Address structure fields

Field Name	Default	Valid Data	Comment
MBID	N/A	Digits	The unique identifier of the mailbox. In network messaging, the local mailbox representing the remote node.
RemoteMBID	N/A	Digits	Only valid for network messaging. It is the mailbox number of the mailbox on the remote node.
Extension	N/A	Digits	The telephone extension.
Name	N/A	Text	The display name.

EmailName	N/A	Text	The email name.
EmailAddress	N/A	Email address	The email address.
PhoneNumber	N/A	Phone Number	The phone number.
MobileNumber	N/A	Phone Number	The mobile number.

## Attachment

The Attachment structure is used within various Messaging functions, both as parameters to functions like **CXMessageRetrieveAttachments** and also as embedded structure within the Message structure (as Attachment structures).

The following table describes the fields of the Attachment structure.

Table 18. Attachment structure

Field Name	Default	Valid Data	Comment
ID	N/A	Numeric	The unique identifier of the Attachment.
FileType	N/A	4 – The attachment is a WAV (voice) file 5 – The attachment is a TIFF (FAX) file 6 – The attachment is a Text file	The type of file
FileName	N/A	GUID	The name of the file, without path or suffix.
FilePath	N/A	File Path	The full file path to the attachment.
IsBody	0	0, 1	Whether the attachment is the text body part of an email message.
PageCount	0	Numeric	The page count found in the TIFF file for a fax attachment.

## Message

The following table describes the fields of the Message structure.

Table 19. Message structure

Field Name	Default	Valid Data	Comment
MsgID	N/A	Numeric	The session specific unique identifier for the message.
PermanentMsgID	N/A	Text with no spaces	The persistent unique identifier for the message.
Type	N/A	<i>Voice</i> - The message is a voice message <i>Fax</i> - The message is a facsimile message <i>Email</i> - The message is an email <i>Calendar</i> - The message is a calendar item	The type of the message.
VoiceMsgSubType	N/A	<i>Normal</i> - The message is not of a special type <i>MissedCall</i> - A missed call message <i>Acknowledgement</i> - An Acknowledgement message <i>RecordedConversation</i> - A recorded conversation	The type of voice message.
Priority	<i>N</i>	<i>N</i> - The message has normal priority. <i>U</i> - The message has urgent priority.	The message priority.
OkToFwd	1	0, 1	Whether the message may be forwarded to another mailbox.  <div> <b>NOTE:</b> The server does not enforce this setting. It is up to the client to decide whether to honor the setting.         </div>
TotalVoiceMsec	N/A	Numeric	The total length of all voice attachments in milliseconds.

ReceiptType	<i>None</i>	<p><i>None</i> – Not a return receipt. Any other value indicates that this message is a Receipt.</p> <p><i>NonDelivery</i> – A non-delivery receipt, refer to DeliveryError for reason.</p> <p><i>Delivery</i> – A receipt for successful delivery of a message.</p> <p><i>Read</i> – A receipt for a message that was read.</p>	The type of return receipt message.
DeliveryError	0	0 – No error.	An internal error code indicating the reason for non-delivery. Only applies to messages with ReceiptType field set to <i>NonDelivery</i> .
ReturnReceiptRequested	0	0, 1	<p>Whether a return receipt is requested for this message. Only applies to messages that have ReceiptType set to <i>None</i>.</p> <p><b>NOTE:</b> The server automatically generates the receipt when the message is marked <b>Read</b> or <b>Deleted</b>.</p>
SentTimestamp	Current Time	Timestamp	The date and time in UTC when the sender sent the message.
DeliveryTimestamp	N/A	Timestamp	The date and time in UTC when the message was delivered by the server.
UserDeliveryTimestamp	N/A	Timestamp	DeliveryTimestamp expressed in the subscriber's time zone.
ChangedTimestamp	N/A	Timestamp	The date and time in UTC when the message was last changed (such as when marked as Read).
Subject	Generated	Text	The message subject.

SubdueReply	0	0, 1	Whether to disallow replies for a message addressed to an Outbound mailbox
Sender	N/A		This is a structure of the <a href="#">Address</a> type. It represents the sender of the message.
OriginalSender	N/A		This is a structure of the <a href="#">Address</a> type. It represents the original sender of the message. 'Original' here means the first sender that sent the message. The OriginalSender structure is retained across all forwards and replies of the message. If the message originated from an outside caller, then this structure will have the outside caller's phone number in the PhoneNumber field. If the message originated from a subscriber, then the PhoneNumber field will have the original sender subscriber's phone number.
Recipients	N/A		This is a container that contains one or more Recipient structures. Recipient structure is of the <a href="#">Address</a> type. It represents a Recipient of the message.
OriginalAddresseeList	N/A		This is a container that contains one or more OriginalAddressee structures. OriginalAddressee structure is of the <a href="#">Address</a> type. OriginalAddresseeList captures the list of Recipients specified when this message was sent. 'Original' here means an unaltered list, that is to say that even when a recipient is a DL mailbox, this list would track that DL mailbox instead of breaking the DL down to its constituent member mailboxes.

Attachments	N/A	This is a container that contains one or more <a href="#">Attachment</a> structures, each representing one attachment of the message.
-------------	-----	---

## StoredMessage

The following table describes the fields of the StoredMessage structure.

Table 20. StoredMessage structure

Field Name	Default	Valid Data	Comment
StoreType	N/A	<i>Local</i> – Local store <i>External</i> – External store	The type of the data store within which the message is stored.
FolderType	N/A	<i>New</i> – New or Inbox folder <i>Saved</i> – Saved folder	The type of folder within which the message is stored.
Read	N/A	0, 1	Whether the message has been read.
Deleted	0	0, 1	Whether the message has been deleted. Normally, this field should be set to 0.
SoftDeleted	N/A	0, 1	Whether the message has been marked for deletion.
Purged	N/A	0, 1	Whether the message files have been purged. If this is 1, then the message is no longer available even though the message envelope is still present in the database.
Message	N/A		This is the <a href="#">Message</a> structure representing the properties of the message that are not storage related.

## SpeechAliasCollection

A SpeechAliasCollection structure is a container of various speech phrases that have been grouped together because these phrases represent various ways of saying the name of the same entity such as a

Group, GroupType etc. All the phrases that are part of this group belong to one and the same locale/culture.

The following table describes the fields of the SpeechAliasCollection structure.

Table 21. SpeechAliasCollection structure

Field Name	Default	Valid Data	Comment
SpeechAliasCollectionID	N/A	Numeric – 64-bit value	The identifier of the speech alias collection.
Name	N/A	Text	The display name for this collection. This name should be the localized culture-specific name for this collection.
TTSNameAlias	N/A	Text	The TTS Name Alias.
SpokenNameFile	N/A	Text	Identifies the Spoken Name file.
CultureName	N/A	Text	Identifies the locale/culture to which this collection belongs. E.g. <i>en-US</i> for English language used in United States.
SpeechAliasArray	N/A		This is a container that holds one or more SpeechAlias structures. Each SpeechAlias structure has a Phrase field whose value is the text phrase that is used for speech recognition. Each SpeechAlias structure also has a SpeechAliasID field that is the numerical identifier of the Speech Alias.

## SpeechCmd

A SpeechCmd structure represents a single speech command. It groups several speech alias collections together. Each speech alias collection in a SpeechCmd structure has a different culture and provides the localization of the speech command for that particular culture.

The following table describes the fields of the SpeechCmd structure.

Table 22. SpeechCmd Structure

Field Name	Default	Valid Data	Comment
SpeechCmdID	N/A	Numeric – 64-bit	The identifier of the speech



		value	command.
CollectionType	N/A	<i>CallProcessorCmd</i> – Call Processor command <i>Group</i> – Group <i>GroupType</i> – Group Type <i>AvailabilityState</i> – An Availability State <i>AvailabilityDay</i> – An Availability Day	Specifies the type of the speech command. E.g. A value of <i>Group</i> means that this Speech command holds speech alias collections that hold phrases that are alternate ways of saying a particular Group's Name.
SpeechAliasCollectionArray	N/A		This is a container that holds one or more SpeechAliasCollection structures. Each <a href="#">SpeechAliasCollection</a> structure has a different culture and provides the localization of the speech command for that particular culture.

## AvailabilityDay

The following table describes the fields of the AvailabilityDay structure.

Table 23. AvailabilityDay structure

Field Name	Default	Valid Data	Comment
AvailDayID	N/A	Number > 0	The unique database record id. Ignored for a record Add.
MBID	N/A	Digits	The mailbox number of the Subscriber or Availability Class of Service mailbox. The mailbox length is always between 2 and 10 digits.
Name	N/A	AlphaNumeric	This is a read only value determined using the SpeechCmdID
SpeechCmdID	N/A	Number	The ID of the speech command used to select the schedule using speech recognition. For days of the week, this will have a value of 0 since days of the week are not selected by speech

recognition. For all other day schedules, the SpeechCmdID is common to all SpeechAliasCollections, one per configured language, that are used to recognize this day schedule.

ActivationTimestamp	immediate	String of form yyyy-mm-ddThh:mm:ssZ where Z indicates Zulu	The UTC time when activation is to occur.  yyyy-mm-dd = The year (yyyy), month (mm), and day (dd)  hh:mm:ss = The hour (hh), minute (mm), and second (ss)  If not specified, activation is immediate.
ExpirationTimestamp	forever	String of form yyyy-mm-ddThh:mm:ssZ	The UTC time when expiration is to occur.  yyyy-mm-dd = The year (yyyy), month (mm), and day (dd)  hh:mm:ss = The hour (hh), minute (mm), and second (ss)  If not specified, activation is forever.
Enabled	N/A	0 or 1	A flag indicating if the record is active. Days of the week should always be active (1). Other day schedules should always be disabled (0) unless a manual override is in effect.
DayIndex	N/A	0 - 7	The type of schedule.  <b>0</b> non day of week schedule <b>1</b> Sunday <b>2</b> Monday <b>3</b> Tuesday <b>4</b> Wednesday <b>5</b> Thursday <b>6</b> Friday <b>7</b> Saturday
IsWorkday	0	0 or 1	Indicates whether the record specifies a day that is considered to be a work day. Typically, DayIndex = 1,7 (Sunday

and Saturday) are not workdays and DayIndex=2-6 (Monday through Friday) are work days. A DayIndex of 0 (an override schedule) should use a value of 1.

The start and stop work hours are specified in the subscriber mailbox's StartWorkTOD and StopWorkTOD fields.

TemporaryOverride	N/A	(for internal use only)	(for internal use only)
TempBuildAvailDayID	N/A	(for internal use only)	(for internal use only)
MBAvailTimeArray	N/A	container	Container for optional MBAvailTime time period definition records
MBAvailTime	N/A	container	Container for time period parameters
MBAvailTime/Cardinal	N/A	Number > 0	Miss-named unique ordinal value specifying record order.
MBAvailTime/StartTOD	N/A	Digits HHMM 0 - 2400	Starting time for the time period. HH should be 00 through 24. MM should be 15, 30, 45 or 00.
MBAvailTime/StopTOD	N/A	Digits HHMM 0 - 2400	Ending time for the time period. HH should be 00 through 24. MM should be 15, 30, 45 or 00, not 14, 29, 44, or 59, so that the end is announced without a gap before the next time period.
MBAvailTime/AvailStateID	N/A		The unique database record id of the state that should be in effect during the time period.

## AvailabilityState

The following table describes the fields of the AvailabilityState structure.

Table 24. AvailabilityState structure

Field Name	Default	Valid Data	Comment
AvailStateID	N/A	Number > 0	The unique database record id. Ignored for a record Add.
MBID	N/A	Digits	The mailbox number of the Subscriber or Availability Class of Service mailbox. The mailbox length is always between 2 and 10 digits.
Name	N/A	AlphaNumeric	This is a read only value determined using the <b>SpeechCmdID</b>
ParentAvailStateID	N/A	Number	The unique database record id of the Availability Class of Service state this record is replicated from.  Use 0 for Availability Class of Service states.
UseParentDeviceList	N/A	0 or 1	A flag indicating whether the find-me device list is lock to the device list of the ParentAvailStateID state.  0 means the list is unlocked and editable. In this case, the find-me device list is specified using the MBDeviceList described below.  1 means the list is locked.
Locked	N/A	0 or 1	A flag indicating whether the subscriber is allowed to edit any setting of the state. 1 indicates the state is read only and cannot be edited. This should not be controllable by the subscriber.
AvailAutoOverrideID	N/A	Number	The unique database record id of the Availability Class of Service Automatic Override record that will automatically select this state. An <b>AvailAutoOverrideID</b> must only be used by one state for each subscriber otherwise the state to activate is indeterminate. All automatic overrides do not need to be used.

0 indicates none.

Hidden	N/A	0 or 1	A flag that indicates whether the subscriber want the State to appear in any of their spoken or drop down lists for selection.
PlayGeneralGreeting	N/A	0 or 1	A flag indicating whether the system wide general greeting should be played.
AnnounceEnable	N/A	0 or 1	A flag indicating whether the subscriber's availability should be announced. Must be a 1 for <b>AvailAnnouncementID</b> , <b>AnnounceReturnEnable</b> , <b>AnnounceReturnDate</b> , and <b>AnnounceReturnTime</b> to be usable.
AnnounceFirst	N/A	0 or 1	A flag indicating whether the subscriber's Availability should be announced before attempting to reach them or giving them any options.
AnnouncePrivacyOptions	N/A	String	<p>The method to announce the specific subscriber. Options are as follows.</p> <p><i>NameOnly</i> – Only speak the subscriber's name.</p> <p><i>PrimaryDeviceOnly</i> - Only speak the subscriber's primary number (example: <i>Extension 1234</i>).</p> <p><i>NameandPrimary</i> - Speak subscriber's name and primary number (example: <i>John Doe at extension 1234</i>).</p> <p><i>None</i> – refrain from speaking name or primary number.</p>
AvailAnnouncementID	N/A	Number	The unique database record id of the recorded Announcement to play. Examples are <i>is not available</i> , <i>is no longer with the company</i> . Use SOAP function <b>CXAvailabilityAnnouncementListGet</b>

to obtain the list of options.  
0 indicates none.

AnnounceReturnEnable	N/A	0 or 1	A flag indicating whether the subscriber wants to use the AnnounceReturnDate and AnnounceReturnTime setting.
AnnounceReturnDate	N/A	0 or 1	A flag indicating whether the Return Date is announced. This should not be controllable by the subscriber.
AnnounceReturnTime	N/A	0 or 1	A flag indicating whether the Return Time is announced. This should not be controllable by the subscriber.
AnnounceDefaultGreetingType	N/A	0 or 1	<p>The greeting to be played prior to presenting the list of instructions. Options are as follows.</p> <p><i>None</i> - No greeting is played</p> <p><i>StandardOrBusy</i> – Play the standard greeting or the busy greeting depending on conditions.</p> <p><i>OutOfOffice</i> – Play the Out of Office greeting.</p> <p>Note that if this option is used, the recording should not present options or instructions such as taking a message since these are presented in the remainder of the dialog.</p> <p><i>AvailGreeting</i> – Play the Availability Greeting specified by AvailGreetingID.</p>
AvailGreetingID	N/A	Number	<p>The unique database record id of the recorded custom Availability Greeting to play. Use the SOAP function CXAvailabilityGreetingListGet to obtain the list of options.</p> <p>0 indicates none.</p>
AnnounceCustomAllowed	N/A	0 or 1	Future – not implemented

PlayGreetingBeforeRouting	N/A	0 or 1	A flag indicating whether the greeting is to be played prior to routing the caller.
LocateMode	N/A	String	<p>When the caller says <i>Locate</i>, this option controls the behavior they experience. Options are as follows.</p> <p><i>Silent</i> – At most says <i>One moment please</i> and does not interact with caller between numbers</p> <p><i>Confirm</i> – Asks for confirmation before each number is called.</p> <p><i>Confirm Once</i> – Asks for confirmation before the first number only.</p> <p><i>Announce</i> – Announces prior to each number that is dialing the next number.</p> <p><i>AnnounceWithCancel</i> – Announce prior to each number that it is going to try the number unless they say <i>Cancel</i>.</p>
FinalRoutingOptionEnable	N/A	0 or 1	A flag controlling whether the TimeoutAction is used.
TimeoutAction	N/A	String	<p>The action to take after timeout or error. Options are as follows.</p> <p><i>TakeMessage</i> – take a message</p> <p><i>RouteToOperator</i> – transfer to the operator</p> <p><i>RouteToAnswerModeCP</i> – route to the answer mode Call Processor mailbox.</p> <p><i>HangUp</i> – Cordially end the conversation such as <i>Thank you for calling</i></p> <p><i>RouteToSubscriber</i> – route to the subscriber specified in the TimeoutRouteMBID parameter.</p> <p><i>RouteToCP</i> – route to a Call Processor mailbox specified in the</p>

			TimeoutRouteMBID parameter.
TimeoutRouteMBID			The mailbox number of the Subscriber or Call Processor mailbox used for the TimeoutAction parameter <i>RouteToSubscriber</i> or <i>RouteToCP</i> options. The mailbox length is always between 2 and 10 digits
DoNotAcceptMessages	N/A	0 or 1	A flag indicating that a message should not be taken.
SuppressNotAcceptMsgPrompt	N/A	0 or 1	A flag indication to the caller that the subscriber is not accepting messages.
AutoLocateEnable	N/A	0 or 1	A flag indicating that Locate should begin automatically without recognizing "locate" or without selecting locate using DTMF.
MobileNotificationAllow	N/A	0 or 1	A flag indicating that mobile notification should be made when in this state.
CallScreeningEnable	N/A	0 or 1	A flag indicating that call screening callouts should be made when in this state.
ContinueCallAfterDisconnect	N/A	0 or 1	A flag indicating that the next number should be dialed after a callout is answered but then hung up. Only applied when Call Screening is enabled. Essentially, do not treat a called party hang up as a rejection. Treat it as a no answer.
DefaultBehavior	N/A	0 or 1	When the state is selected as a manual override ( <b>CXAvailabilityActivate</b> ) without being given any termination conditions, the behavior defaults in this manner. Options are as follows.  <i>Indefinite</i> – continues on until a replacement manual override is selected ( <b>CXAvailabilityActivate</b> )



or a return to automatic schedule processing  
(**CXAvailabilityEnableSchedules**) is requested.

*StopTimeOfDay* – continue on until the time of day specified in the DefaultStopTOD parameter.

*LengthOfTime* - continue on for the number of seconds specified in the DefaultDurationSeconds parameter.

DefaultStopTOD	N/A	Digits HHMM range 0 - 2400	The stop time of day used for DefaultBehavior option <b>StopTimeOfDay</b> . 2400 indicates the next day. All other values are in the current day.
DefaultDurationSeconds	N/A	Number (1 – 86399)	The number of seconds used for DefaultBehavior option <b>LengthOfTime</b> . The maximum length is 86399 (23 hours, 59 minutes, 59 seconds). One hour is represented by 3600 (60 * 60).
SpeechCmdID	N/A	Number	The ID of the speech command used to select the state using speech recognition. The <b>SpeechCmdID</b> is common to all <b>SpeechAliasCollections</b> , one per configured language, that are used to recognize this state.
TTSNameAlias	N/A	AlphaNumeric	This is a read only value determined using the SpeechCmdID which should be used as input to the TTS player instead of the Name field if not empty.
SpokenNameFile	N/A	AlphaNumeric	This is a read only value determined using the SpeechCmdID which is the file name of the name recording. If empty, use TTS to play the Name or the TTSNameAlias as appropriate.
TempBuildAvailStateID	N/A	(for internal use only)	(for internal use only)

MBAvailDeviceArray	N/A	container	Container for optional <b>MBAvailDevice</b> definition records
MBAvailDevice	N/A	container	Container for the device parameters
MBAvailDevice/Cardinal	N/A	Number > 0	Miss-named unique ordinal value specifying record order.
MBAvailDevice/RecordType	N/A	Number	<p>Indicates the type of <b>MBAvailDevice</b> record. Options are as follows.</p> <p>0 – Actual Device (used for the state of subscribers when the field, UseParentDeviceList, is a value of 0).</p> <p>1 – Reference to a subscriber device of the type DevUsageType. Availability Class of Service records will always have a value of 1 since they are always references. Subscriber records will have a value of 1 if the field UseParentDeviceList is a value of 1, since these are references and do not exist in the database since they are built from the Availability Class of Service.</p>
MBAvailDevice/DevUsageType	N/A	String	<p>This string enumeration indicates the usage type of the device. Options are as follows.</p> <p><i>Other</i> – General Number</p> <p><i>CompanyMobile</i> – Company Mobile</p> <p><i>PersonalMobile</i> – Personal Mobile</p> <p><i>RemoteOffice</i> – Remote Office</p> <p>Number</p> <p><i>HomeNumber</i> – Home Number</p> <p><i>Extension</i> – PBX Extension</p> <p><i>Temporary</i> – Temporary Number</p> <p><i>PrimaryDevice</i> - The Primary Device (Any device can be the primary device)</p> <p><i>PrimaryMobile</i> - The Primary Mobile Device (Only the Company or Personal Mobile Only can be the</p>

primary mobile)

MBAvailDevice/DeviceID	N/A	Number	The unique database record id of the subscriber's device. For Availability Class of Service records, this value will be 0 since the Availability Class of Service has no devices.
------------------------	-----	--------	---

## AvailabilityAutoOverride

The following table describes the fields of the AvailabilityDay structure.

Table 25. AvailabilityDay Structure

Field Name	Default	Valid Data	Comment
AvailAutoOverrideID	N/A	Number > 0	The unique database record id. Ignored for a record Add.
MBID	N/A	Digits	The mailbox number of the Availability Class of Service mailbox. The mailbox length is always between 2 and 10 digits.
Priority	N/A	Number > 0	The records are process in priority order with the lowest number (1) record having the highest priority. Each record should have a unique priority. While ordered, the values do not have to be sequential and do not have to begin with 1.
Name	N/A	AlphaNumeric	This is a the displayable name of the Automatic Override. For clarity, they should resemble the descriptions associated with the parameter AvailAutoOverrideType.
RuleType	N/A	String	The results of each row are related logically based on the RuleType. Options are as follows. AND – All row values must be true for the Automatic Override to be activated. OR – If any row value is true, the Automatic Override is Activated.
AvailAutoOverrideType	N/A	Number > 0	Automatic overrides are classified to allow interchanging set of Overrides

and maintaining the relationship to the Availability States. The following types are currently defined.

- 1 UserDefined
- 2 When At Desk
- 3 When Away
- 4 When In A Meeting
- 5 When Presenting
- 6 When Mobile
- 7 When Out Of Office
- 8 When At Home Office
- 9 When Do Not Disturb
- 10 When In An Accepted Meeting

TempBuildAvailAutoO verrideID	N/A	(for internal use only)	(for internal use only)
AvailAggregationRow Array	N/A	container	Container for optional <b>AvailAggregationRow</b> records
AvailAggregationRow	N/A	container	Container for row parameters
AvailAggregationRow/ AvailAggregationRowI D	N/A	Number > 0	The unique database record id. Ignored for a record <b>Add</b> .
AvailAggregationRow/ RuleType	N/A	String	The results of each <i>In Point</i> in the row are related logically based on the RuleType. Options are as follows.  AND – All items must be true for the row value to be true.  OR – If any item is true, the row value is true.
AvailAggregationRow/ AvailInPointConnectio n	N/A	container	Container for optional <b>AvailInPointConnection</b> records
AvailAggregationRow/ AvailInPointConnectio n	N/A	container	Container for <i>In Point</i> connection items
AvailAggregationRow/ AvailInPointConnectio n/ AvailInPointType	N/A	number	<i>In Point</i> type. Options are as follows. 1 – Calendar - Free 2 – Calendar - Tentative 3 – Calendar - Busy

4 – Calendar - Out Of Office  
 101 – Presence - OffLine  
 102 – Presence - Available  
 103 – Presence - Away  
 104 – Presence - Busy  
 105 – Presence - Do Not Disturb  
 106 – Presence - Be Right Back  
 201 – Location - Other  
 202–220 - Reserved locations  
 221-299 - User defined locations  
 302 – Time – Working Hours  
 303 – Time – Non Working Hours  
 401 – Setting – Out Of Office  
 402 – Setting - Do Not Disturb

AvailAggregationRow/ AvailInPointConnection/ SourceEngineType	N/A	String	<i>In Point</i> connection source engine type. Options are as follows. <i>Calendar</i> – Calendar server <i>In Point</i> <i>Presence</i> – Presence server <i>In Point</i> <i>Location</i> – Location server <i>In Point</i> <i>Time</i> – Time <i>In Point</i> <i>Setting</i> – Setting <i>In Point</i>
AvailAggregationRow/ AvailInPointConnection/ ServerType	N/A	String	<i>In Point</i> connection source engine server type. Options are as follows. <i>Calendar</i> – Generic calendar <i>In Point</i> <i>Exchange</i> – Microsoft Exchange <i>In Point</i> (Don't use. Use <i>Calendar</i> instead) <i>IMAP</i> – Internet Message Access Protocol <i>In Point</i> (Don't use. Use <i>Calendar</i> instead) <i>Presence</i> – Generic Presence <i>In Point</i> <i>OCSLync</i> – Microsoft Lync or OCS <i>In</i> <i>Point</i> (Don't use. Use <i>Calendar</i> instead) <i>CXMobileClient</i> – MiCollab AM Mobile Client <i>In Point</i> <i>CXWorkHours</i> – Mitel MiCollab AM Work Hours <i>In Point</i> <i>CXSettings</i> – Mitel MiCollab AM Settings <i>In Point</i>
AvailAggregationRow/	N/A	0 or 1	Indicates whether to use the binary <i>In</i>

AvailInPointConnectio n/ Negate			<i>Point</i> value or the complement of the <i>In Point</i> value.
AvailAggregationRow/ AvailInPointConnectio n/ RequiredSource	N/A	0 or 1	Future – not implemented

## AvailabilityPointInstalled

The following table describes the fields of the AvailabilityPointInstalled structure.

Table 26. AvailabilityPointInstalled Structure

Field Name	Default	Valid Data	Comment
SourceEngineType	N/A	String	<i>In Point</i> installed source engine type. Options are as follows. <i>Calendar</i> – Calendar server <i>In Point</i> <i>Presence</i> – Presence server <i>In Point</i> <i>Location</i> – Location server <i>In Point</i> <i>Time</i> – Time <i>In Point</i> <i>Setting</i> – Setting <i>In Point</i>
ServerType	N/A	String	<i>In Point</i> installed source engine server type. Options are as follows. <i>Calendar</i> – Generic calendar <i>In Point</i> <i>Exchange</i> – Microsoft Exchange Calendar <i>In Point</i> (don't use. Use <i>Calendar</i> instead) <i>IMAP</i> – Internet Message Access Protocol <i>In Point</i> (don't use. Use <i>Calendar</i> instead) <i>Presence</i> – Generic Presence <i>In Point</i> <i>OCSLync</i> – Microsoft Lync or OCS <i>In Point</i> (don't use. Use <i>Presence</i> instead) <i>CXMobileClient</i> – MiCollab AM Mobile Client <i>In Point</i> <i>CXWorkHours</i> – Mitel MiCollab AM Work Hours <i>In Point</i> <i>CXSettings</i> – Mitel MiCollab AM Settings <i>In Point</i>

AvailInPointType	N/A	number	<i>In Point</i> type. Options are as follows. 1 – Calendar - Free 2 – Calendar - Tentative 3 – Calendar - Busy 4 – Calendar - Out Of Office 5 – Calendar – Working Elsewhere 101 – Presence - OffLine 102 – Presence - Available 103 – Presence - Away 104 – Presence - Busy 105 – Presence - Do Not Disturb 106 – Presence - Be Right Back 201 – Location - Other 202–220 - Reserved locations 221-299 - User defined locations 302 – Time – Working Hours 303 – Time – Non Working Hours 401 – Setting – Out Of Office 402 – Setting - Do Not Disturb
Name	N/A	Alphanumeric	This is a the displayable name of the <i>In Point</i> . For clarity, they should resemble the descriptions associated with the parameter <i>AvailInPointType</i> such as <i>Free</i> , <i>Busy</i> , <i>Do Not Disturb</i> , etc.
Enabled	N/A	0 or 1	Indicates whether to use the <i>In Point</i> value.

## AvailabilityGreeting

The following table describes the fields of the AvailabilityGreeting structure.

Table 27. AvailabilityGreeting Structure

Field Name	Default	Valid Data	Comment
AvailGreetingID	N/A	Number > 0	The unique database record id. Ignored

for a record Add.

MBID	N/A	Digits	The mailbox number of the Subscriber or Availability Class of Service mailbox. The mailbox length is always between 2 and 10 digits.
Name	N/A	Alphanumeric	This is a read only value determined using the SpeechCmdID.
SpeechCmdID	N/A	Number	The ID of the speech command used to select the state using speech recognition. The SpeechCmdID is common to all SpeechAliasCollections, one per configured language, that are used to recognize this state.
Hidden	N/A	0 or 1	Whether this greeting is hidden from the clients. If set to 1, it won't appear in client screens and this greeting cannot be used by clients.
TTSNameAlias	N/A	AlphaNumeric	This is a read only value determined using the SpeechCmdID which should be used as input to the TTS player instead of the Name field if not empty.
SpokenNameFile	N/A	AlphaNumeric	This is a read only value determined using the SpeechCmdID which is the file name of the name recording. If empty, use TTS to play the Name or the TTSNameAlias as appropriate.