



FS – Profile Manager

Content

1	INTRODUCTION	2
1.1	GLOSSARY	2
1.1.1	Profile	2
2	FUNCTION REQUIREMENTS (COMMERCIAL).....	3
3	FUNCTION SPECIFICATION (DESIGN RELATED)	3
3.1	INTRODUCTION	3
3.2	EXPORTED INTERFACES	3
3.2.1	<i>IProfile</i>	4
3.2.2	<i>ICommunity</i>	5
3.2.3	<i>ICOS</i>	5
3.2.4	<i>IDistributionList</i>	5
3.3	IMPORTED INTERFACES	6
3.4	FUNCTIONS	6
3.4.1	<i>LDAP Communication with MUR</i>	6
3.4.2	<i>Referrals and Other LDAP Features</i>	6
3.4.3	<i>MUR Host</i>	7
3.4.4	<i>Profile Lookup</i>	9
3.4.5	<i>Subscriber creation and deletion</i>	9
3.4.6	<i>Authentication</i>	10
3.4.7	<i>Available MUR Attributes</i>	10
3.4.8	<i>Greeting and Spoken Name</i>	10
3.4.9	<i>Available Greeting Properties</i>	11
3.4.10	<i>Distribution Lists</i>	12
3.4.11	<i>Events</i>	15
3.4.12	<i>Thread safe</i>	15
4	EXTERNAL OPERATION CONDITIONS.....	15
4.1	CONFIGURATION	15
4.2	LOGGING	16
5	CAPABILITIES.....	16
6	REFERENCES	16
7	TERMINOLOGY	16
8	APPENDIX: ATTRIBUTE MAPPING	16



History

Version	Date	Adjustments
A	2006-10-03	First version. (MANDE)

1 Introduction

This document specifies the function of the Profile Manager component. The Profile Manager provides an interface that can be used by other components in order to retrieve for example COS settings or a subscriber Profile according to the system data model described in [1] or to create or delete subscribers from the system.

The Profile Manager communicates with MUR using LDAP in order to retrieve subscriber data.

1.1 Glossary

1.1.1 Profile

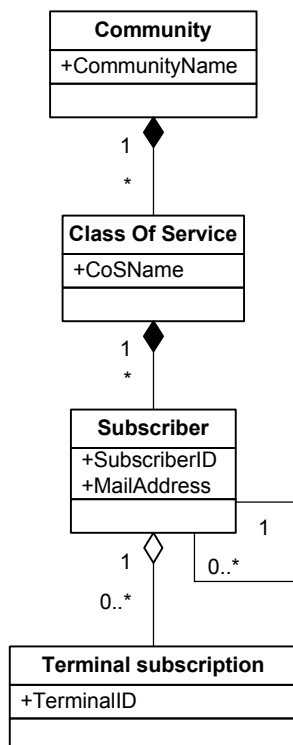


Figure 1 Profile

A profile is a grouping of all the settings applicable for a subscriber (see [1] for a detailed definition).

Given a unique identifier of a subscriber (i.e. SubscriberID or MailAddress) or a terminal subscription (i.e. TerminalID) a profile is built based on the settings in the Terminal Subscription, Subscriber, COS and Community for that subscriber.

According to [1], a subscriber (and therefore also a profile) has zero or more instances of a Mailbox as defined in [3].

2 Function Requirements (Commercial)

This paragraph is intentionally left blank.

3 Function Specification (Design Related)

3.1 Introduction

The Profile Manager exports the IProfile, ICommunity, ICOS and IDistributionList interfaces. These interfaces are used by clients to retrieve a subscriber Profile and the Community, COS or distribution lists of a Profile.

The Profile Manager imports the ILog, IConfig, IMediaObject, ILocateService, IServiceInstance, IMailbox, IEventDispatcher and IProvisioning interfaces.

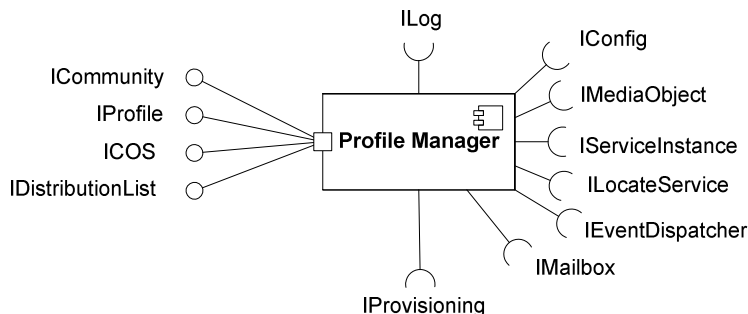


Figure 2 Profile Manager component

The client requests a Profile by specifying unique subscriber identification or terminal subscription. The Profile Manager returns the matching subscriber Profile to the client.

From the IProfile interface it is possible to retrieve separate parts of the profile; the Community or the COS. From the IProfile interface it is also possible to retrieve profile attributes such as distribution lists.

3.2 Exported Interfaces

The Profile Manager exports the following interfaces: IProfile, ICommunity, ICOS, and IDistributionList.



3.2.1 IProfile

The IProfile interface is used by a client to manage a subscriber Profile. IProfile offers the following methods:

- *IProfile[] getProfile(String entryPoint, AttributeFilter filter, Boolean limit)*
A Profile instance is retrieved using a factory. The factory returns a list of the profiles that matches the given filter. The attributes allowed in the filter are configurable in the Profile Manager. The entryPoint is an optional parameter that indicates the LDAP search base used for the profile lookup. The limit parameter indicates if the search could be limited to the user directory in the local sub/domain, e.g. when using iMux (see section 3.4.4.1).
- *Object getAttribute(String attrName)*
Returns the requested attribute. Available attributes are described in section 8.
- *void setAttribute(String attrName, Object attrValue)*
Sets the specified attribute to the given value. Available attributes are described in section 8.
- *void createSubscription(ISubscription sub, String adminUID, String COSName);*
Creates a subscriber with the specified characteristics. COSName is optional. In case the request fails, the client will be notified including the reason why.
- *void deleteSubscription(ISubscription sub, String adminUID);*
Deletes a subscriber with the specified characteristics. In case the request fails, the client will be notified including the reason why.
- *ICommunity getCommunity()*
Returns the community of the profile (see 3.2.2).
- *ICOS getCOS()*
Returns the COS of the profile (see 3.2.3).
- *IMediaObject getGreeting(GreetingFilter filter)*
Returns the greeting that matches the given filter. The properties allowed in the filter are specified in 2. IMediaObject is an imported interface, see chapter 3.3.
- *void setGreeting(GreetingFilter properties, IMediaObject greeting, IMediaContentResource resource)*
Replaces the greeting that matches the given filter. The properties allowed in the filter are specified in 2. IMediaObject and IMediaContentResource are imported interfaces, see chapter 3.3.
- *IMediaObject getSpokenName(GreetingFilter filter)*
Returns the spoken name of the profile. IMediaObject is an imported interface, see chapter 3.3.
- *void setSpokenName(GreetingFilter properties, IMediaObject spokenName, IMediaContentResource resource)*
Sets the spoken name for the profile. IMediaObject and IMediaContentResource are imported interfaces, see chapter 3.3.



- *DistributionList[] getDistributionLists()*
Returns the profiles distribution lists (see 3.2.4).
- *void addDistributionList(DistributionList distrList)*
Adds a new distribution list (see 3.2.4).
- *void deleteDistributionList(DistributionList distrList)*
Deletes the given distribution list.
- *IMailbox getMailbox(String mailHost, String accountID, String accountPassword)*
Returns the requested Mailbox. The parameters mailHost and accountID uniquely identifies a Mailbox.
- *void close()*
Indicates that resources allocated for the Profile can be freed: closing open mailboxes etc.

3.2.2 ICommunity

This interface is used by a client to get Community attributes. ICommunity offers the following methods:

- *ICommunity getCommunity(String community)*
A community instance is retrieved using a factory.
- *Object getAttribute(String attrName)*
Returns the requested attribute. Available attributes are described in section 8.

3.2.3 ICOS

This interface is used by a client to get COS attributes. ICOS offers the following methods:

- *ICos getCos(String cos)*
A COS instance is retrieved using a factory.
- *Object getAttribute(String attrName)*
Returns the requested attribute. Available attributes are described in section 8.

3.2.4 IDistributionList

This interface is used by a client to modify a distribution list. IDistributionList offers the following methods:

- *DistributionList(String ID)*
Creates a distribution list with the specified identifier.
- *String getID()*
Returns the identifier of the distribution list.
- *void addMember(String member)*
Adds a member.
- *void removeMember(String member)*
Removes the indicated member.



- *String[] getMembers()*
Returns a list of the current members.
- *IMediaObject getSpokenName()*
Returns the distribution list spoken name. IMediaObject is an imported interface, see chapter 3.3.
- *void setSpokenName(IMediaObject spokenName, IMediaContentResource resource)*
Sets the distribution list spoken name. IMediaObject and IMediaContentResource are imported interfaces, see chapter 3.3.

3.3 Imported Interfaces

The Profile Manager uses the following external interfaces:

- ILog for logging
- IConfig for configuration
- IMediaObject for passing media objects such as a greeting
- ILocateService for M3 service lookup
- IServiceInstance for accessing service properties
- IMailbox for initiation of a subscriber Mailbox
- IEventDispatcher for retrieving events
- IProvisioning for create and delete of subscribers

3.4 Functions

3.4.1 LDAP Communication with MUR

The Profile Manager communicates with MUR using LDAP. An internal interface IDirectoryService is used towards the LDAP implementation.

The communication with MUR uses a configurable number of connections (using a connection pool) and does as few read and writes as possible (using data caching). The lifetime of a connection is configurable.

3.4.2 Referrals and Other LDAP Features

The Profile Manager LDAP communication support referrals.

The Profile Manager sets a connection timeout for the LDAP connections to avoid connections that will hang forever if MUR is down. The value of the timeout depends upon if the connection is used for reading or writing.

If an LDAP command fails, the Profile Manager can re-try the command. The decision whether or not to retry the command depends upon the time elapsed since the previous command was initiated and how many retries that have been configured.

3.4.3 MUR Host

The Profile Manager retrieves the MUR host to communicate with using the `ILocateService` interface. The Profile Manager asks the component register for a host and it is the component register that is responsible for determining which MUR hosts that is appropriate depending upon for example replicationid (for the MultiMaster function).

The Profile Manager asks the Component Register for a host each time to read from or write to MUR, see Figure 3.

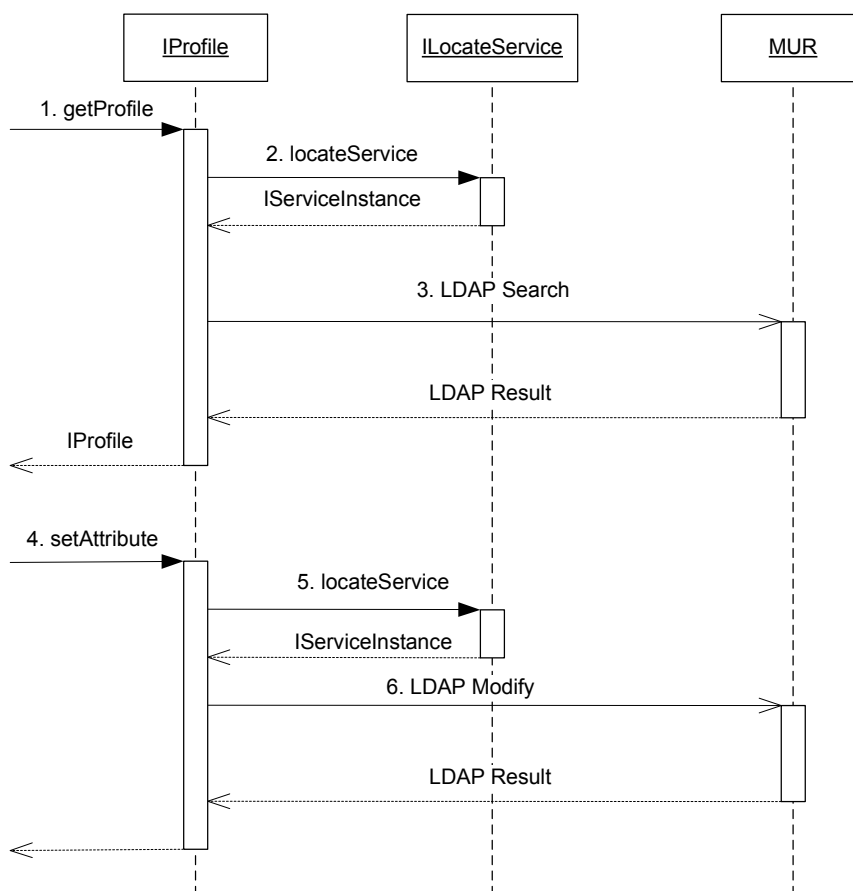


Figure 3 Host lookup in Component Register

1. A profile instance is requested.
2. The MUR host to use is retrieved using the `ILocateService` interface.
3. The LDAP search request is sent to the MUR host.
4. An attribute is changed in the profile.
5. The MUR host to use is retrieved using the `ILocateService` interface.
6. The LDAP modify request is sent to the MUR host.

3.4.3.1 MUR host errors

When communication with a MUR host fails, the Profile Manager reports this to the Component Register and retries the search request, see Figure 4.

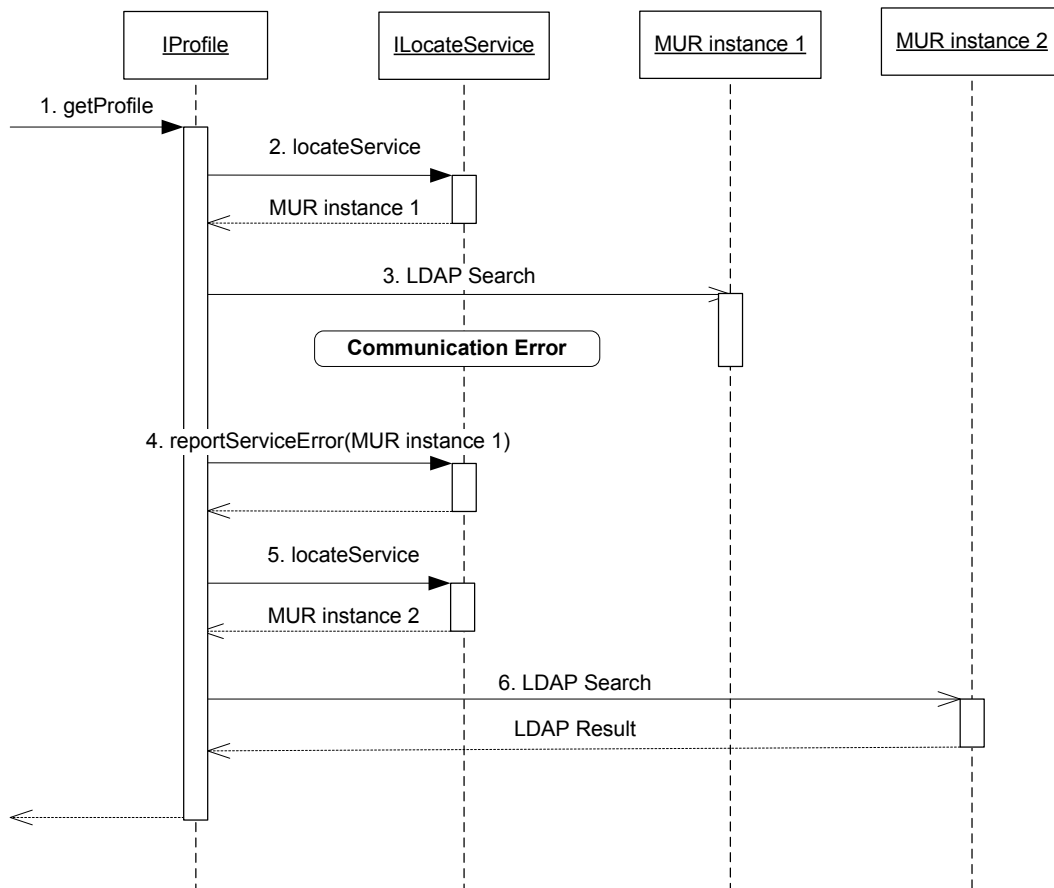


Figure 4 Host failure

1. A profile instance is requested.
2. The MUR host to use is retrieved using the ILocateService interface.
3. The LDAP search request sent to the MUR host fails.
4. The host failure is reported using the ILocateService interface.
5. Another MUR host is retrieved using the ILocateService interface.
6. The LDAP search request is sent to the MUR host.

If no MUR host can be retrieved using the ILocateService interface this is logged as an error and notified to the user of the Profile Manager.



3.4.4 Profile Lookup

The attribute or attributes to search for when finding a profile can be selected from the configured available MUR attributes, see 3.4.7.

A profile consists of the summarized data from Community, COS, Subscriber and Terminal Subscription. If an attribute exists on several levels, only one value is active for the profile.

When a profile is retrieved using method `getProfile`, the lookups of Community, COS, Subscriber and Terminal Subscription are made in a configurable priority order for each attribute.

When an attribute exists on several levels, a value with lower priority will be overwritten. For example, an attribute exists on both COS and Subscriber level. If the Subscriber level has higher priority, this attribute will be returned from the Profile. Of course the COS attribute can be retrieved using the ICOS interface.

If a profile cannot be created due to provisioning errors this is logged as an error and is notified to the user of the Profile Manager.

3.4.4.1 Support for iMux solution

A profile lookup can be marked as limited which means that the search is limited to a user-directory in a local subdomain. This is indicated by adding the LDAP attribute *limitScope* to the search. It is configurable if a limited profile lookup should use the *limitScope* attribute or not.

3.4.5 Subscriber creation and deletion

The Profile Manager provides functionality to create and delete subscribers from the system. The IProvisioning interface is used for this purpose. When a subscriber is to be created the Profile Manager basically performs the following tasks:

1. Adapt the ISubscription object (provided in the create request) using information specified in section 8 to the IProvisioning interface. Basically this means that the attribute name will be mapped to their corresponding names defined in the IProvisioning interface. The attribute value will be left unchanged and passed as is to the IProvisioning interface.
2. Looks-up the password to use for the provisioning administrator uid (provided in the create request). This information is then cached in the Profile Manager and possible to use in consecutive sessions. The cache is updated at the same interval as other cached MUR data (e.g. COS).
3. If the client provided a COSName as well, Profile Manager, by using the community of the provision administrator provided, retrieves the COSDN from MUR. The COSDN will then be used in the IProvisioning interface. If the client provides both COSName and COSDN (as a Subscription parameter), the COSName will be used. If COSName is not found in the request will fail.
If the client provides neither COSName nor COSDN, the administrators own COSDN (an attribute in the administrators profile that explicitly describes the COSDN to use when subscribers are to be created) will be used.

When a subscriber shall be deleted, only task 1 and 2 are performed.

3.4.6 Authentication

The Profile Manager does not do authentication of its clients. It is up to the clients (based on information provided by the Profile Manager) to authenticate end-users that shall have access to the subscriber information.

3.4.7 Available MUR Attributes

The MUR and provisioning attributes that shall be accessible to clients of the Profile Manager are listed in an XML document that is read at startup or when re-loading configuration. The XML document lists all MUR and provisioning attributes that should be available. For each attribute the document also lists the corresponding name in the Profile Manager, the type and format, the attribute read location (COS, Subscriber etc.), if this is a provisioning attribute, read location priority, write location, whether the attribute is read-only, max and min values and possible default values if such exists.

The main reason for keeping the attributes in an XML/configuration file is that it should be easy to extend for example the Profile Manager with support for a new MUR attribute.

3.4.8 Greeting and Spoken Name

Greetings and Spoken Names follow the format described in [2].

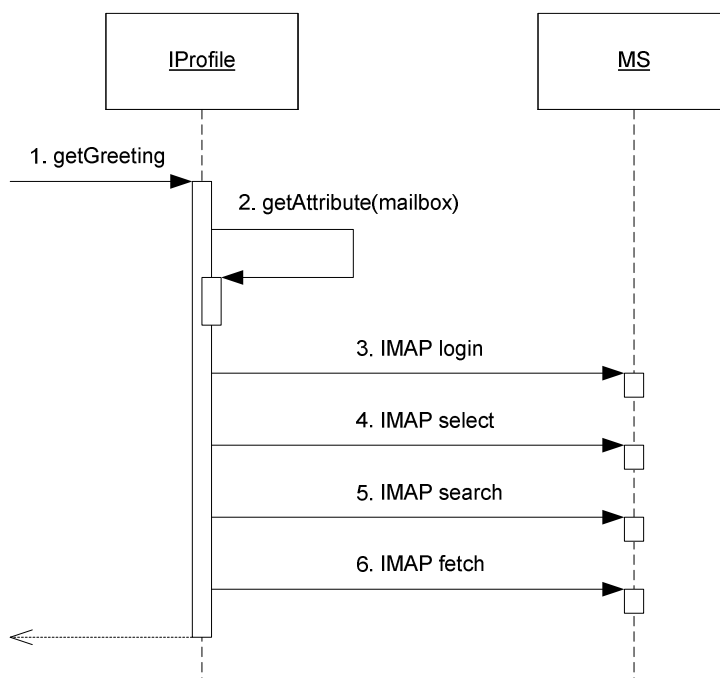


Figure 5 Retrieval of greeting

1. A greeting is requested (a spoken name request is handled in the same way).
2. The mailbox name containing the profile's greetings is retrieved.

3. The mailbox is retrieved.
4. The greeting folder is selected.
5. The greeting message is searched in MS.
6. The greeting message is retrieved from MS.

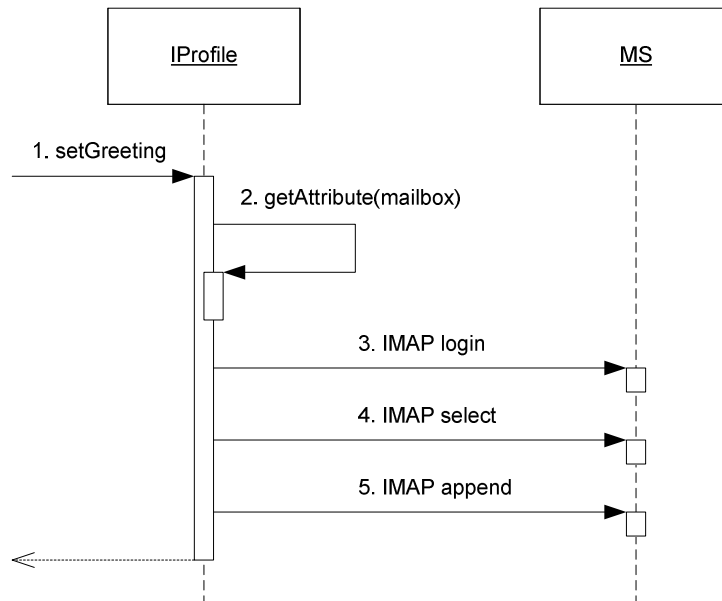


Figure 6 Setting a greeting

1. A greeting is submitted (a spoken name submission is handled in the same way).
2. The mailbox name containing the profile's greetings is retrieved.
3. The mailbox is retrieved.
4. The greeting folder is selected.
5. The greeting message is appended to the folder.

3.4.9 Available Greeting Properties

The client to the Profile Manager can access greetings that match a certain filter. Table 1 lists the properties used in the GreetingFilter.

Table 1 Greeting Properties

Property Name	Type	Valid values	Description
Type	String	"AllCalls", "NoAnswer", "Busy", "OutOfHours", "ExtendedAbsence", "CDG" or "Temporary"	Type of greeting.
Format	String	"voice" or "video"	The greeting format.



CallerId	String	A telephone number	Contains the number of the caller that shall receive a Caller Dependent Greeting. Only interesting to use in filter when accessing greeting of type CDG.
----------	--------	--------------------	---

3.4.10 Distribution Lists

Distribution lists can be added, retrieved, removed and modified, see Figure 7 to Figure 10. Changes made in existing distribution lists are propagated to MUR.

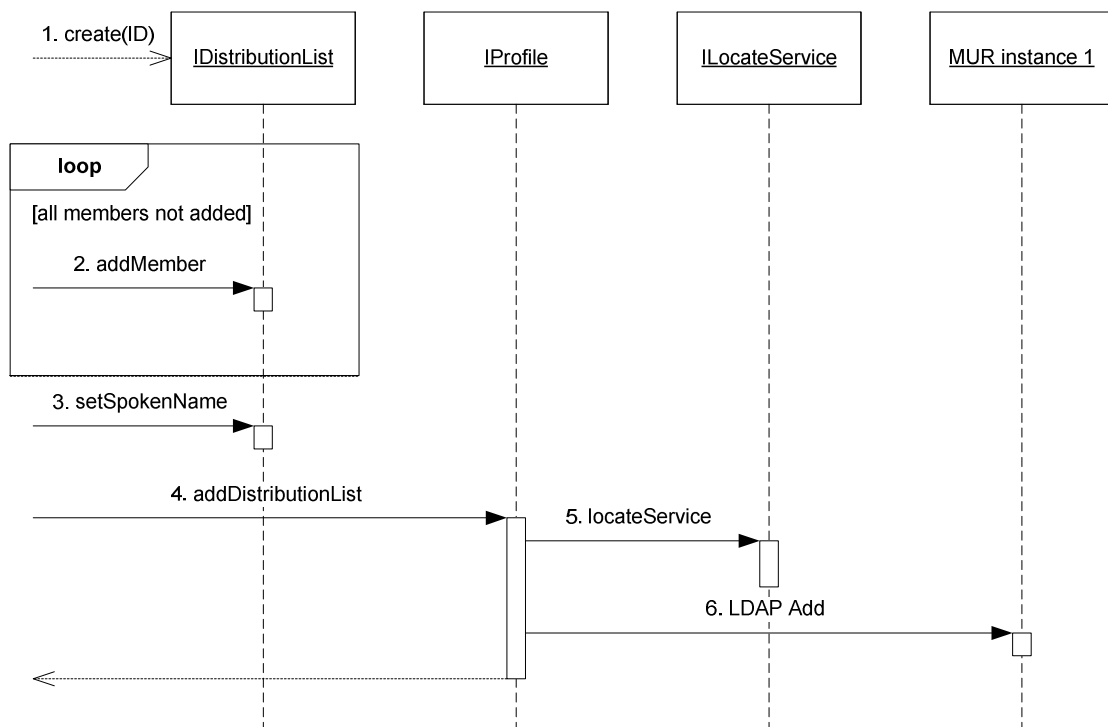


Figure 7 Adding a distribution list

1. A distribution list identified by ID is created.
2. All members are added to the distribution list.
3. The distribution list's spoken name is added to the distribution list, see section 3.4.8 for how the spoken name is stored.
4. The distribution list is added to the Profile.
5. The MUR host to use is retrieved using the ILocateService interface.
6. The LDAP add request is sent to the MUR host.

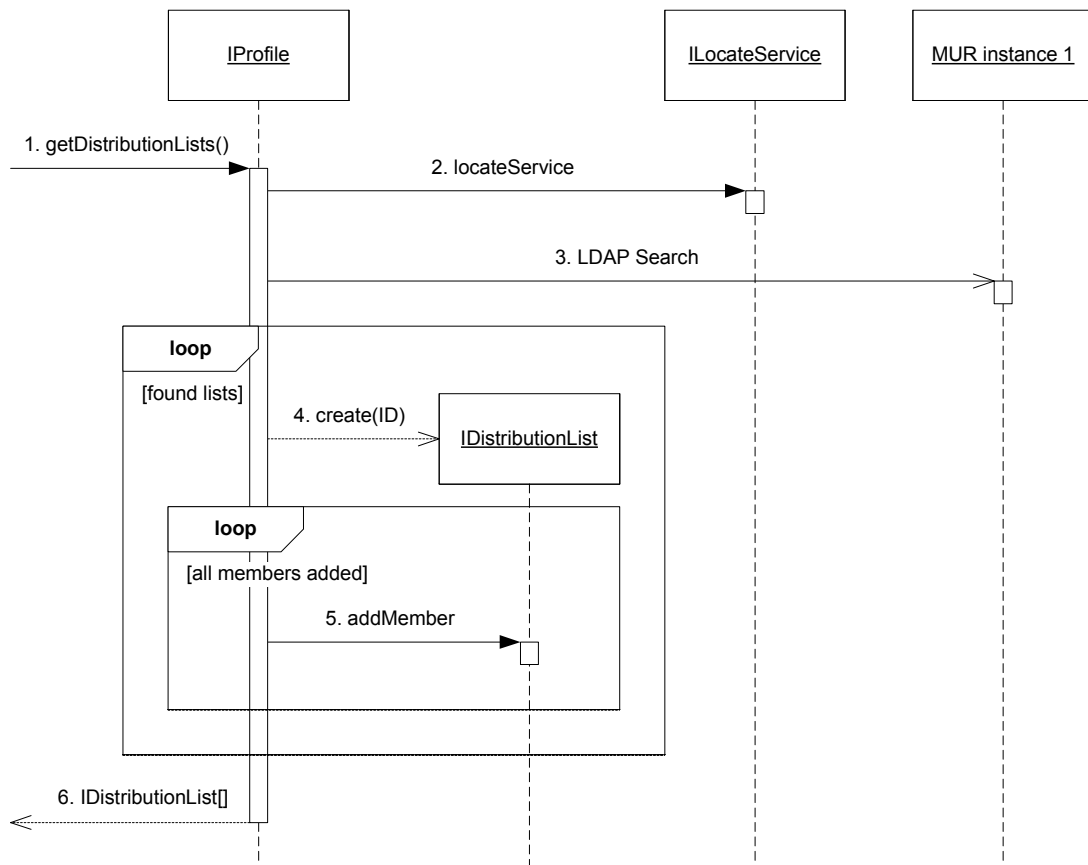


Figure 8 Retrieving distribution lists

1. The distribution lists are requested from the Profile
2. The MUR host to use is retrieved using the ILocateService interface.
3. The LDAP search request is sent to the MUR host.
4. A distribution list is created.
5. All members are added to the distribution list.
6. The distribution lists are returned.

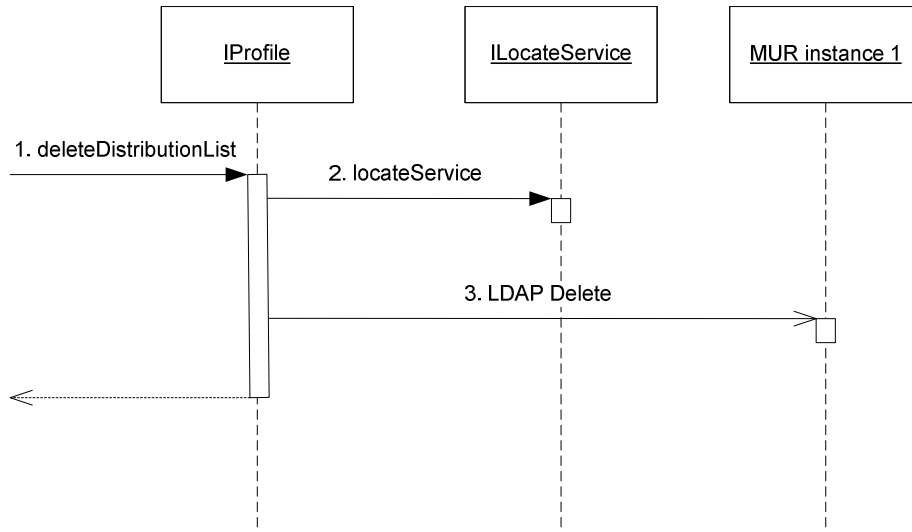


Figure 9 Deleting a distribution list

1. A distribution list is submitted to the Profile for removal.
2. The MUR host to use is retrieved using the ILocateService interface.
3. The LDAP delete request is sent to the MUR host.

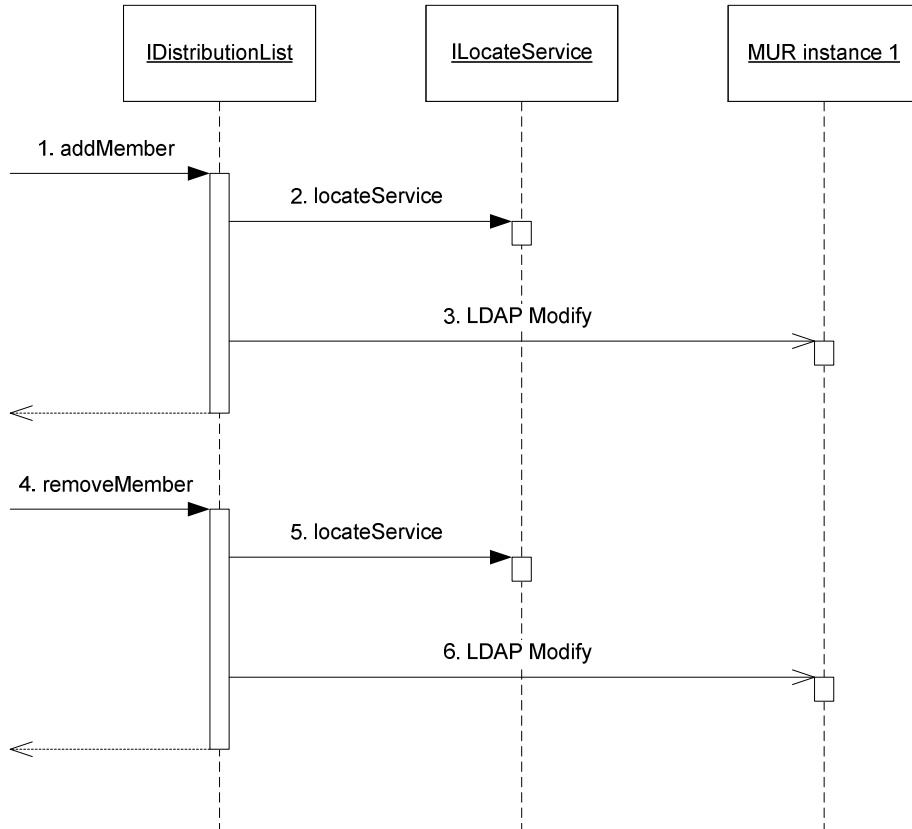


Figure 10 Modifying a distribution list



1. A new member is added to an already existing distribution list.
2. The MUR host to use is retrieved using the ILocateService interface.
3. The LDAP modify request is sent to the MUR host.
4. A member is removed from an already existing distribution list.
5. The MUR host to use is retrieved using the ILocateService interface.
6. The LDAP modify request is sent to the MUR host.

3.4.11 Events

3.4.11.1 Consumed

The Profile Manager registers itself using the IEventDispatcher interface as a receiver of the following events:

- Configuration has changed (indicates that configuration should be re-read)

3.4.11.2 Produced

- The Profile Manager does not produce any events.

3.4.12 Thread safe

The Profile Manager is thread safe.

4 External Operation Conditions

4.1 Configuration

The following are configurable in the Profile Manager:

- Maximum amount of simultaneous LDAP connection towards one MUR instance.
- Idle timeout for a connection, i.e. how long the connection can be unused before it is considered idle and is disconnected.
- Lifetime for a connection, i.e. how long the connection can be used before it is disconnected. This can be used to renew connections behind a load balancer for example.
- Read and write timeouts on a connection
- Number of retries in case of communication error
- The maximum time period during which retries are performed
- Directory administrator credentials
- The available attributes for a profile. These attributes are also possible to use for the search criteria for profile lookup.
- If the additional attribute *limitScope* should be added to profile searches marked as limited.



4.2 Logging

- Hosted service circumstances are logged as described in ref.[7] using the IHostedServiceLogger interface.
- All calls to interface methods are logged at info level. Method name and parameter values are logged. At return returned value is logged.

5 Capabilities

This paragraph is intentionally left blank.

6 References

- [1] Technical Report M3 Data Model
6/0363-PRO049 Uen
- [2] IWD End User Message Format
3/155 19-HDB 101 02 Uen
- [3] FS Mailbox
5/FS-MAS0001 Uen
- [4] FS External Component Register
9/FS-MAS0001 Uen
- [5] MUR Developers Guide
3/1551-CRH 109 086 Uen
- [6] FS Operate and Maintain Manager
17/FS-MAS0001 Uen
- [7] FS-Log Manager
2/FS-MAS0001 Uen
- [8] IWD-Provisioning Interface
2/155 19-CRH 109 086 Uen

7 Terminology

COS	Class of Service
LDAP	Lightweight Directory Access Protocol
MUR	Messaging User Registry
CAI	Customer Administration Interface

8 Appendix: Attribute Mapping

Table 2 specifies the mapping between attributes available in the IProfile interface and their corresponding names used in MUR and the CAI interface (see [8]).



Table 2 Attribute mapping

Application attributes						Name in MUR	Name in CAI	Description
Name	Loc	Type	Valid values/ Examples	Default	Per mission			
activegreetingid	B	String	"SpokenName, NoAnswer, OutOfHours"	-	RW	activegreetingid	-	Holds a list of current active greetings.
addressbookserver	-	String	"mur2@host.domain"	-	RO	-	ADDRESSBOOK_SERVER	The hostname of the address book server.
autoplay	B, C	Boolean	true, false ("yes", "no")	false	RW	autoplay	-	Allows new messages played continuously without input of any DTMF key.
badlogincount	U	Long	Any positive integer number	0	RW	badlogincount	-	Contains the number of invalid login attempts. If value exceeds maxloginlockout, the login is prohibited and the badlogincount need to be reset by a system administrator.
billingnumber	B	String	46701234567	-	RO	billingnumber	TELEPHONENUMBER	Contains the subscriber identifier for the current Profile.
caattr	C	String[]	"sendok"	-	RO	caattr	-	Contains customer adaptation attributes.
callerxfer	C	Boolean	true, false ("yes", "no")	false	RO	callerxfer	-	Enables a caller to transfer a call to another subscriber.
callerxfercoverage	C	Boolean	true, false ("yes", "no")	false	RO	callerxfercoverage	-	Enables a caller to transfer a call to an attendant.
cdgmax	C	Long	A positive integer	10	RO	cdgmax	-	Number of allowed Caller Dependent Greetings.
cn	D	String[]	-	-	RO	cn	-	An array containing the identities of the subscriber's created distribution lists. Currently this id is only a number but in order to be able to integrate this function with the address book in MEC this should be a string. The DistListId shall be sorted numerical. This attribute is RO. This implies that the methods for adding and deleting distribution lists shall be used to modify this attribute.



Application attributes						Name in MUR	Name in CAI	Description
Name	Loc	Type	Valid values/ Examples	Default	Per mission			
cn	U	String	Any string	-	RO	cn	-	The name of the subscriber. In MUR this attribute is UTF-8 encoded but the return value should be encoded using the configuration attribute value for the preferred language for this subscriber.
cosdn	B	String	Any string, e.g. "cos=1,ou=C1,o=ericsson.se"	-	RO	cosdn	COSDN	Use to identify which COS this subscriber belongs to.
cosname	C	String	A valid COS name	-	RO	cosname	-	The name of the COS used for a subscriber.
coverphonenum ber	B	String	A valid telephone number	-	RW	coverph onenum ber	-	The phone number used when a subscriber has activated attendant call coverage.
diskspaceremai ningwarninglev el	C	Float	0.01-0.99	0.15	RO	diskspa cermai ningwar ningleve l	-	Percent setting for warning subscribers of the remaining disk space.
dlentriesmax	C	Long	A positive integer	15	RO	dlentrie smax	-	Number of allowed entries per distribution list.
dlmax	C	Long	A positive integer	10	RO	dlmax	-	Number of allowed distribution lists.
emactivecdg	U	String	"#4660161000#,#46161025#"	""	RW	emactiv ecdg	-	Contains the active Caller Dependent Greeting numbers for a subscriber.
emautoprinfax	U	Boolean	true, false ("yes", "no")	false	RW	emauto prinfax	-	If the subscriber has autoprinfax or not.
emca	U	String	Any string	""	RW	emca	-	Contains user specific attributes to be used in a Customer Adaptation.
emcni	C	Boolean	true, false ("yes", "no")	false	Ro	emcni	-	If the subscriber can login using the original called number (RDNIS) or not.
emcustomerid	?	String	"80000"	-	RO	emcusto merid	CUST- OMERID	?



Application attributes						Name in MUR	Name in CAI	Description
Name	Loc	Type	Valid values/ Examples	Default	Per mission			
emdeliveryprofile	U	String	A valid delivery profile string	-	RW	emdeliveryprofile	DELIVERY_PROFILE	The delivery profile allows a user to have different notification numbers for each notification type.
emfilter	U	String	A valid filter string	-	RW	emfilter	-	A filter used for notification.
emftl	U	String	"00,T,PIN:M"	-	RW	emftl	-	First time login functions not already performed.
emftlfunctions	C	String	"00,T,PIN:M"	"00,F,PIN:M, SPO:M,ACG:M"	RO	emftlfunctions	-	Functions that shall be performed during First time login.
emgroupmember	B	String[]	billingnumber=1234, uniqueidentifier=um66, ou=C16, o=mobeon.com	-	RO	emgroupmember	-	The members in a family mailbox.
emlogindate	B	String	yyyy-mm-dd	-	RW	emlogindate	-	The date until which the user can login to the old voice mail system.
emmaxgroupmember	C	Long	10	10	RO	emmaxgroupmember	-	The maximum number of family mailbox members.
emmin	U	String	"123456"	-	RO	emmin	MIN	Contains the MIN number for the current profile.
emmmscenterid	U	String	"mms1@host.domain"	-	RO	emmmscenterid	MMSCENTER	Component name of the MMS Center
emmsglenmaxvideo	C	Long	A positive integer	180	RO	emmsglenmaxvideo	-	The max length of a video message.
emmsgplayorder	U	String	"LIFO,LIFO"	"LIFO,FIFO"	RW	emmsgplayorder	-	Defines the order in which new and saved messages are played.
emnoofmailquota	C	Long	Any integer	-1	RO	emnoofmailquota	-	Value of -1 indicates unlimited. Value of >=0 indicates number of messages in quota.
emnotifnumber	U	String	A valid telephone number	-	RW	emnotifnumber	NOTIFICATION_NUMBER	The phone number used for SMS notification.



Application attributes						Name in MUR	Name in CAI	Description
Name	Loc	Type	Valid values/ Examples	Default	Per mission			
emntd	C, B	String	"SMS,MWI,ODL"	""	RW	emntd	DISABLER_NOTIFICATION_TYPES	Contains a comma separated list of disabled notification types. One or several of the following: SMS, MWI, CMW, MMS, WAP, ODL (ODL = Outdial).
emodlpinskip	C, U	Boolean	true, false ("yes", "no")	false	RW	emodlpinskip	-	If the subscriber must enter pin code for an outdial notification call.
emoutboundcalltl	C	Long	30	0	RO	emoutboundcalltl	-	Number of seconds allowed for an outbound call.
empnc	U	String	"00471234++070123456#++h"	-	RO	empnc	PAGER_NOTIFICATION_STRING	The string containing pager number, callers number and pause signs. (h=hangup) This attribute is used to toggle pager notification on/off.
empreferreddateformat	U	String	"mm/dd/yy"		RO	empreferreddateformat	DATEFORMAT	Preferred date format.
empreferredtimeformat	U	String	"12" or "24"	"12"	RW	empreferredtimeformat	TIMEFORMAT	If the subscriber has 12 or 24 hours time presentation.
emreadlevel	C	Long	-1 to infinity	-1	RO	emreadlevel	-	Defines the number of seconds a message must have been listened to before it is treated as read.
emsecnb	U	String[]	A list of valid phone numbers	-	RO	emsecnb	-	Secondary phone numbers used for multi line purposes.
emservicedn	C	String[]	"emservicename=webmail," "ou=services" , "o=mvastest.se"	-	RO	emservicedn	-	Contains the DN of the services this COS have.
emsmppcenterid	U	String	"smpp1@host.domain"	-	RO	emsmppcenterid	SMPPCENTER	Component name of the SMPP Center



Application attributes						Name in MUR	Name in CAI	Description
Name	Loc	Type	Valid values/ Examples	Default	Per mission			
emtmpgprt	U	String	[activationTime]; [deactivationTime] E.g. ";;" or ";2004-08-26 15:20:30"	-	RW	emtmpgprt	-	Contains a semicolon if the temporary greeting is active. activationTime and deactivationTime may specify a validity time.
emtmpgprtaavailable	C	Boolean	true, false true, false ("yes", "no")	false	RO	emtmpgprtaavailable	-	If the subscriber can use temporary greeting.
emtuiaccess	C	String	"none" or "std"	"std"	RO	emtuiaccess	-	Which telephony service to run at this access.
emtuiblockedmenu	C	String[]	"AutoPlay"	-	RO	emtuiblockedmenu	-	This attribute specifies which menus in the TUI that NOT shall be used.
emuserntd	U	String	"PAG"	""	RW	emuserntd	-	Contains a comma separated list of disabled notification types.
emusersd	U	String	"slamdown_notification"	-	RO	emusersd	-	Disables services for a subscriber. The services mentioned here will act as if disabled for the subscriber. Currently the only service to disable is slamdown notification.
emvuiaccess	C	String	"none" or "std"	"none"	RO	emvuiaccess	-	Which video service to run at this access.
eomsgwarning	C	Long	15, 30, 45 or 60	30	RO	eomsgwarning	-	End of message warning threshold in seconds.
facsimiletelephonenum	U	String	A valid telephone number	-	RW	facsimiletelephonenum	FAX_NUMBER	The default Fax telephone number
familymembers	-	String[]	"123456","1242535"	-	RO	-	FAMILY_MEMBERS	The subscribers that belong to a family mailbox
fastloginavailable	C	Boolean	true, false ("yes", "no")	false	RO	fastloginavailable	-	If the subscriber has access to fast login or not.
fastloginenabled	C, B	Boolean	true, false ("yes", "no")	false	RW	fastloginenabled	-	If the subscriber will be able to retrieve fax or not from his/her mailbox.



Application attributes						Name in MUR	Name in CAI	Description
Name	Loc	Type	Valid values/ Examples	Default	Per mission			
faxenabled	C	Boolean	true, false ("yes", "no")	false	RO	faxenabled	-	If the subscriber has access to retrieving fax from his/her mailbox.
gender	U	String	"F" or "M"	"F"	RO	gender	GENDER	The gender of the subscriber.
greetingsecmax	C	Long	Any positive integer	60	RO	greetingsecmax	-	Maximum length of time allowed for a recorded greeting.
inboundfaxno	U	String	A valid phone number		RO	inboundfaxno	-	Used in the creation of a subscriber's mail alternate address.
includespokename	C	Boolean	true, false ("yes", "no")	true	RO	includespokename	-	If spoken name should be included in voice messages.
inhoursdow	C, B	String	"12345"	"12345"	RW	inhoursdow	-	The valid days of the work week for the subscriber. (1=Monday)
inhoursend	C, B	String	A valid time, e.g. "1700"	"1700"	RW	inhoursend	-	The end of the business hours during the business days of the week. The time is in the subscriber's time zone.
inhoursstart	C, B	String	A valid time, e.g. "0800"	"0800"	RW	inhoursstart	-	The start of the business hours during the business days of the week. The time is in the subscriber's time zone.
mail	U	String	A valid email address	-	RO	mail	MAILADDRESS	The email address of the subscriber.
mailalternatedress	U	String[]	A list of valid email addresses	-	RO	mailalternatedress	-	An array containing the subscriber's alternate email addresses. Used for e.g. specifying the email address that faxes are sent to, and as from address when fax and emails are printed to fax.
mailhost	U	String	"ms.mobeon.se"	-	RO	mailhost	MAILHOST	The subscriber's storage host to use.
mailquota	U	Long	Any integer	-	RO	mailquota	-	Value of -1 indicates unlimited. Value of 0 takes the default value from object "class=nsmsgDefaultMailBoxQuota". Value of >=1 indicates quota in bytes.



Application attributes						Name in MUR	Name in CAI	Description
Name	Loc	Type	Valid values/ Examples	Default	Per mission			
mailuserstatus	U	String	"inactive" or "active"	-	RO	mailuserstatus	-	Indicates if this mailbox is in use or not.
maxloginlockout	C	Long	Any positive integer	6	RO	maxloginlockout	-	Max number of allowed failed login attempts. Used in conjunction with badlogincount.
messageinventory	C, B	Boolean	true, false true, false ("yes", "no")	true	RO	messageinventory	-	If message inventory should be played or not at retrieval.
messageplayemail	C, B	String	"body", "header" or "both"	"both"	RW	messageplayemail	-	Defines what part(s) of an email that is sent at retrieval.
messageplayvideo	C, B	String	"body", "header" or "both"	both	RW	messageplayvideo	-	Defines what part(s) of a videomail is played at retrieval.
messageplayvoice	C, B	String	"body", "header" or "both"	both	RW	messageplayvoice	-	Defines what part(s) of a voicemail is played at retrieval.
msglenmaxvoice	C	Long	A positive integer	180	RO	msglenmaxvoice	-	The max length of a voice message.
mslocation	-	String	"host.domain"	-	RO	-	MSLOCATION	Specifies the location of MS.
noforfirstname	B	String	A valid DTFM sequence, e.g. 18729372	-	RO	noforfirstname	-	DTMF equivalent of the person's first name followed by the last name.
noforlastname	B	String	A valid DTFM sequence, e.g. 18729372	-	RO	noforlastname	-	DTMF equivalent of the person's last name followed by the first name.
passwdlenmax	C, B	Long	A positive integer	4	RO	passwdlenmax	-	The max length of the PIN code (umpassword).
passwdlenmin	C, B	Long	A positive integer	4	RO	passwdlenmin	-	The min length of the PIN code (umpassword).
password	U	String	"abcd"	-	RO	password	MAILBOX_PW	



Application attributes						Name in MUR	Name in CAI	Description
Name	Loc	Type	Valid values/ Examples	Default	Per mission			
passwordskipavailable	C	Boolean	true, false ("yes", "no")	false	RO	passwordskipavailable	-	If the subscriber has access to PIN skip or not.
passwordskipenabled	B	Boolean	true, false ("yes", "no")	false	RW	passwordskipenabled	-	If PIN skip has been enabled by the subscriber.
preferredlanguage	U	String	A valid language code according to ISO 639-1.	-	RW	preferredlanguage	LANGUAGE	The subscriber's preferred language.
phonenumbercalleraccesssection	C	String	Any string	"CALLEROUT DIAL"	RO	phonenumbercalleraccesssection	-	The section to use to determine what numbers are allowed for non-subscribers to dial.
phonenumberdialingsection	C	String	Any string	SUBSCRIBER OUTDIAL	RO	phonenumberdialingsection	-	The section to use to determine what numbers are allowed for subscriber to dial.
reinventor	C, B	Boolean	true, false ("yes", "no")	false	RO	reinventor	-	If the re-inventory is performed each time the subscriber returns to the main menu.
sn	U	String	"Modano"	-	RO	sn	SN	Last name of the subscriber.
subscribertimezone	B	String	Any valid time zone	-	RO	subscribertimezone	TIMEZONE	Subscriber's time zone.
subscriberxfer	C	Boolean	true, false ("yes", "no")	false	RO	subscriberxfer	-	If call transfer is enabled for the subscriber.
subsearchbase	C	String	"ou=C1, o=mvastest.se"	-	RO	subsearchbase	-	The Community the subscriber belongs to.
telephonenumber	U	String	Any valid phone number	-	RO	telephonenumber	TELEPHONENUMBER	The subscriber's telephone number. This is normally the same as billingnumber.



Application attributes						Name in MUR	Name in CAI	Description
Name	Loc	Type	Valid values/ Examples	Default	Per mission			
ttsemailenable d	C	Boolean	true, false true, false ("yes", "no")	false	RO	ttsemail enabled	-	If TTS for reading text is enabled for the subscriber.
uid	U	String	"jmodano"	-	RO	uid	UID	
umpassword	U	String	Any valid sequence of numbers: "1234"	-	RW	umpass word	PIN	The PIN code used when login through TUI. Contains the decrypted PIN code. Profile is responsible for the encryption/decryption.
urgentmsgplay	C, B	Boolean	true, false true, false ("yes", "no")	true	RW	urgent msgplay	-	If urgent messages should be played before normal messages.
userlevel	U	Long	Any positive integer	-	RW	userleve l	-	0 - subscriber has not logged in 70 - subscriber has logged in and has set up their mailbox. Only used if emftl is not used.
welcomegrt	C	String	Any string	-	RO	welcom egrt	-	Indicates the name (without path) of the prompt file (without extension) to be played when a subscriber has logged in.
xamtevatitle	U	String	"GrtAdmin"	-	RO	xamtev atitle	-	Kind of administrator. If this attribute is set, no broadcast messages will be sent to this subscriber's storage.