



IP Telephony

Contact Centers

Mobility

Services

WHITE
PAPER

Avaya Modular Messaging Caller Applications

White Paper and Frequently Asked Questions

May 2005

Table of Contents

Telephone User Interface Configuration without Caller Apps	1
Description	1
Need for More Configuration Options	1
Requirements.....	2
Implementation	3
Frequently Asked Questions	4
General.....	4
Associations	5
Storage.....	5
Deployment.....	6
Undeployment	8
Security	8
“Front Door” Applications.....	8
Application Prompts.....	9
Subscriber Prompts.....	9
Joining MM Voice Mail Domains.....	11
Language Support.....	12
Comparison with Aria Mailbox Types	14
Type	14
Description	14
CA.....	14
Notes	14
Other Features.....	15
Examples of Caller Application Features.....	16
I want callers to hear an announcement, then disconnect the call	16
I want callers to hear an announcement, then leave a comment	16
I want callers to hear an announcement or greeting, then be transferred	18
I want give callers the option of hearing the menu again; I want to have one application jump to another application	18
How do I make the app’s behavior depend on the time of day?.....	19
What happens if I try to delete a node that is referenced by another node?.....	20

Telephone User Interface Configuration without Caller Apps



Description

Avaya Modular Messaging Application Servers (MASs) allow some configuration of the telephone user interface (TUI) “out of the box”.

It is possible to configure the MM Automated Attendant with the voice mail system configuration utility (VMSC). During configuration, the VMSC property page might look like the figure at left.

There are separate property pages for scheduled changes to prompting behavior and a multi-lingual opening menu, but the settings that most change the TUI behavior are on the Attendant Main Menu Editor page.

The customer can specify the number of the main Automated Attendant prompt (by default this is simply: PLEASE ENTER THE MAILBOX NUMBER OF THE PERSON YOU ARE CALLING).

They can record a custom prompt to describe the new actions available. They can also choose from a number of actions to associate with the 1 . . . 9 keys on the caller’s telephone keypad. One action type attempts to transfer the caller to a specified mailbox. Another action allows the caller to record a message and then key in a mailbox number to which to send it. Finally, the “directory” action allows the caller to use the dial-by-name functionality of the TUI (PLEASE SPELL THE LAST NAME FIRST...).

Need for More Configuration Options

These arrangements have several limitations:

Fixed (single) depth of menu

The Automated Attendant menu, if configured, has a fixed “depth” of 1 digit. Because all available options are accessed with a single digit, it is not possible to design a system of nested menus, as might be desirable for a large organization. Not only is one digit considered insufficient; the maximum depth of menus is considered a detail that should not be limited by the architecture of any new system.

Static configuration

Any Automated Attendant configuration made through VMSC will only take effect when the various Application Servers in the voice mail domain are stopped and restarted. Where the configuration options are relatively limited, this may not be a problem — changes will only be made rarely (perhaps just once, after initial installation). However, a more flexible scheme might require changes to be made more frequently. Stopping and restarting MASs to make such changes would be highly undesirable.

Areas of TUI that are configurable

Only the (single) Automated Attendant menu is presently configurable. However, much of the TUI falls outside the Automated Attendant and some of this is just as suitable for configuration. For example, the behavior when a caller is transferred to a subscriber's mailbox is something that many customers might like to customize.

Only One Automated Attendant

There is a single default Automated Attendant (however configured) per voice mail domain. Different behaviors, depending on the number called, could be highly desirable for some customers' applications. A company could publish different numbers for different caller queries, and be able to provide a simplified TUI structure to the caller in each case.

Not Portable

Systems integrators and business partners who install and set up MM may wish to reuse some of their expertise in future installations. There is no way to export the default Automated Attendant settings from one voice mail domain, and import them into another. As the number and complexity of TUI configurations increases, some mechanism for making these configurations portable and reusable would be highly desirable.

Requirements

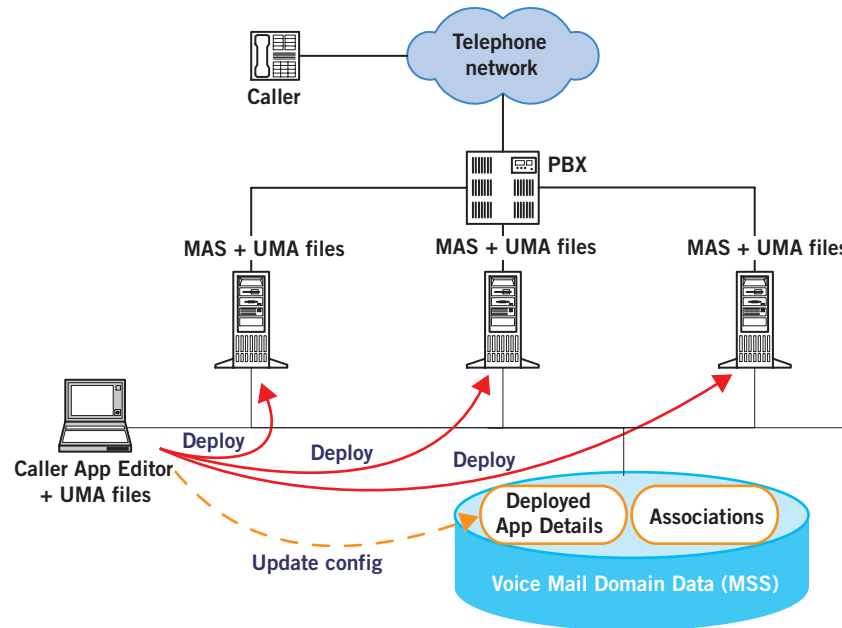
The term *Caller Applications* was given to the feature set in MM 1.0 that tries to address the needs outlined above. Each Caller Application is a self-contained part of the overall MM TUI. Caller Applications (often shortened to *Caller Apps*) can be added and linked in to the MM TUI to provide a more flexible way to handle callers.

The requirements are:

1. Menus of any depth
2. Functionality — at least as many actions possible as the existing Automated Attendant
3. Portability — a Caller App should be easily transferable from one VMD to another
4. Performance — perceived to be as efficient as the existing MM TUI by callers
5. Robustness — if possible, Caller Apps should execute normally if the MAS is offline (i.e. unable to contact the message storage server).
6. Dynamic updates — Caller Apps must be changeable while the MASs run.
7. Multiple Automated Attendants possible
8. Personalization — MM subscribers should be able to provide prompts for Caller Apps

Implementation

The figure shows the way that the data involved with Caller Applications are distributed in MM 1.0.



The Caller App Editor is often run on a machine that is not a MAS. In the figure, the Editor is on a laptop computer.

The applications are saved as UMA files, which are often (but not necessarily) originated and initially stored on the Editor machine. When the administrator needs to make the application available to callers, they use the Editor to transfer a copy of the UMA file to the MASs in the Voice Mail Domain. This process is called *deployment*. If the deployment succeeds, the Voice Mail Domain configuration data is updated to show that the application is present. The Editor may then create one or more associations (also stored in the Voice Mail Domain configuration data) to specify the conditions under which the MASs will run the app for callers.

The implementation of Caller Apps in MM 1.0 and above, meets the requirements stated above for the following reasons.

Menus of any depth; functionality. The Editor allows menus of any depth to be created. The action types that are supported for menu choices include all the standard automated attendant functions, and add some new possibilities.

Portability. The UMA file in which a Caller App is stored can be used to deploy to any MM Voice Mail Domain. As long as the creator does not include VMD-specific details (e.g. mailbox numbers), the App can be reused directly. Even if some details do need to be changed, it is often easier to start with an existing app and modify it slightly than to create a new one.

Performance, robustness. The MAS runs the Caller App from local storage (i.e. the UMA file is stored on the MAS). No network round trips are required (unless Subscriber Prompts are used). This means that the app executes efficiently, and is fully available, even if the MAS becomes disconnected from the data network.

Dynamic updates. The deployment mechanism fully supports dynamic updates.

Multiple Automated Attendants. Associations can specify a called party ID on which a Caller App can be launched. If this number is mapped by the PBX to MAS ports, the caller will hear the designated App as soon as the call is answered.

Personalization. Editors can include references to prompts recorded by MM subscribers in any Caller App. User Administrators can bind a Caller App to the subscriber's mailbox; the App will be executed (in place of the normal greeting) when control is transferred to the mailbox.

Frequently Asked Questions

General

What is a Caller Application?

A Caller Application is a custom extension to the Avaya Modular Messaging telephone user interface (TUI).

Caller Apps can contain prompts, menus and other actions (e.g. transfer call, record message) that can be assembled by administrators or systems integrators to tailor MM's call answering for special customer needs.

Caller Apps can extend or even replace the MM system automated attendant. They can be invoked when callers to an automated attendant enter specified numbers, or when a call is placed to a particular extension, or in place of the normal subscriber's mailbox greetings.

Caller Apps can be uploaded (deployed) when MASs are running, without any interruption to service. Many Caller Apps can be present simultaneously in the Voice Mail Domain. Once deployed, a Caller App is available to callers until an administrator decides to replace or remove it. Caller Apps are stored on the MASs and are automatically reloaded when the MAS service restarts.

Caller Apps that contain all their own application prompts will be entirely unaffected by unavailability of the messaging or directory servers, which guarantees robust call answering.

Although anyone with access to a copy of the Caller Apps Editor (a snap-in for Microsoft Management Console) can create a Caller App and save that app to a file, only MM System Administrators can deploy the apps. Subscribers can supply prompts for suitably designed Caller Apps through the TUI's personal configuration menu. The mailbox number for the subscriber prompts can be "wired in" to the Caller App, or supplied at runtime when known. The latter option makes it possible for one Caller App to act as a template for customization by many different subscribers.

Do Caller Applications Require Mailboxes?

No. You are not required to create a mailbox, or consume a mailbox license, in order to create or deploy a Caller App. Many Caller Apps can be deployed and used without any reference to mailboxes.

Caller Apps may use mailboxes in a few situations:

- The App refers to one or more Subscriber Prompts (see page 11).
- One of the App's actions is to record and send a message to a mailbox.
- The App is bound to a subscriber's mailbox as a "Front Door" (see page 10).

Note that although an App may refer to (or be associated with) a mailbox, it is in no way part of the mailbox itself. An App is simply some custom TUI behavior, stored in a file, which may use mailboxes to support part of its activity.

Associations

What is a Caller App Association?

A Caller App association specifies one or more conditions under which a given Caller App will be presented to a person calling into Avaya Messaging. Every association must contain the identity of a deployed Caller App. The association must also specify one (or both) of the following:

1. An association ID. This is a number that uniquely identifies the association within the voice mail domain. If a caller enters this number on their telephone keypad in response to the Automated Attendant's prompt PLEASE ENTER THE MAILBOX NUMBER OF THE PERSON THAT YOU ARE CALLING, the associated Caller App will run. From the caller's point of view, this appears to be a mailbox number.
2. A called party ID. If the MAS obtains a call record from the PBX that indicates that the caller was trying to reach this number, the associated Caller App will run.

What happens if an association ID is the same as the number of a mailbox?

When a caller is prompted to enter a mailbox number so that the MAS can transfer their call (see above), the number that they enter is checked first against Caller App association IDs, and only then, if no match is found, against mailbox numbers. Therefore, an association ID will hide any mailbox with the same number from callers. The "hidden" mailbox will still be accessible to subscribers who use the TUI to address messages from their own mailbox to its owner. Address resolution when sending messages never checks Caller App associations. MM System Administrators may find this technique (hiding mailboxes from callers) useful in some circumstances, but it may cause confusion if used carelessly. One example is to hide an Enhanced List Application (ELA) mailbox so outside calls are not accidentally sent to a distribution list.

How many digits can an association ID contain?

An association ID can contain one or more digits. However, most administrators will probably choose to make the association IDs the same length as mailbox numbers in the voice mail domain. If an association ID is not the same length as the mailbox numbers, problems may occur:

If the association ID is shorter than the VMD mailbox numbers, the Automated Attendant must execute a time-out for each of the "missing digits" when the caller enters what is at first presumed to be a mailbox number. This silence can be unsettling for the caller.

What is the meaning of the Mailbox Number field in the association?

Applications that are launched by association may contain Subscriber Prompts that are marked in the app as belonging to the *Default* mailbox, or when *Send Message* or nodes define the mailbox number is specified as to be *Provided by runtime*.

The *Mailbox Number* field in the association supplies the required mailbox number for these cases.

Storage

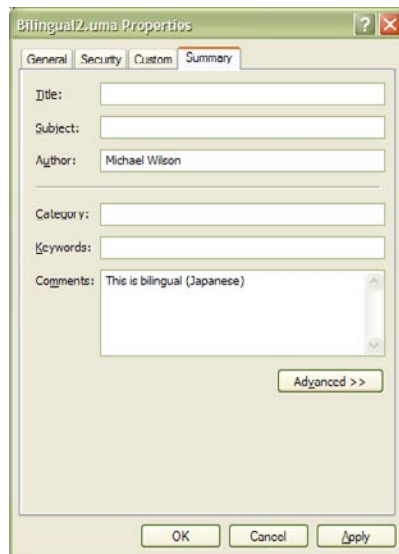
How much space will my Caller Apps occupy?

That depends on their content. A Caller App consists of three types of information: application management data, application structure and application prompts. Of these, prompts will generally occupy by far the largest amount of space. Caller Apps prompts are stored in a format that occupies 8000 bytes for each second of

audio. A Caller App that contains 10 prompts, with an average duration of 5 seconds each, will be at least $(10 \times 5 \times 8000 =) 400,000$ bytes long.

Why does the Editor suggest UMA as a file type for my apps?

UMA stands for Unified Messaging Application. It will be easier to find your apps if you stick to this convention, and when you install the Caller App Editor it is registered to be the application that handles UMA files if you open them from Windows Explorer.



Does anything except MM software understand the UMA file format?

The UMA file format is proprietary, but like many other document formats (e.g. Microsoft Office DOC and XLS files) it is based on OLE structured storage. It contains a *Summary* information set that can be displayed as part of the document properties from the Windows Explorer shell, using the same method that Microsoft Office employs.

If no name is found, the *Author* field will be blank. If no description has been entered, the *Comments* field will be blank.

The *Advanced* property page includes the *Revision* number of the Caller App. This is incremented each time that the Caller App is saved.

The UMA file format is also used by Avaya's Unified Messenger product (for Microsoft Exchange or Lotus Domino).

Deployment

Why does deployment require the Caller App Editor to communicate with all the MASs in the Voice Mail Domain?

It seems as if it would be much more convenient for an administrator to deploy a Caller App to a single MAS, and to have it replicate from there to other MASs in the Voice Mail Domain (VMD). This is how Voice Mail System Configuration utility (VMSC) changes other VMD settings. Instead, the Caller App Editor goes to each MAS in turn, and transfers a copy of the Caller App that is being deployed.

One of the most important requirements for Caller Apps is that they should be able to operate as normally as possible if the MAS is disconnected from the network. If a customer decides to replace the entire MM Automated Attendant with a Caller App, it must be as robust as possible if network problems occur. In particular, callers should not experience a significant lack of service. Therefore, deployment attempts to transfer a copy of the entire application to each MAS. The person performing the deployment can tell whether the application has been successfully transferred to each MAS, or not. The MASs do not need to rely on the network to access the Caller Apps once they have been transferred, and can continue to present a more or less normal telephone user interface.

What happens if a caller is using a Caller App when a new version is deployed?

The caller will not notice that anything has happened. They will continue to hear, and to interact with, the old version of the Caller App, even after the new version has been deployed. Callers who start to interact with the Caller App after the deployment is complete will get the new version. When no callers are using the old version any more, it will automatically be removed.

What happens if two people try to deploy Caller Apps at the same time?

The MASs will only allow one deployment at a time. You will get an error if you try to use the Caller App Editor to deploy an app if the MAS with which it communicates is already busy with another deployment. Because deployments are normally quick (only a few seconds per MAS on a typical network), these “collisions” should be rare.

Where do Caller Apps go when they are deployed?

A Caller App's structure and application prompts are stored in the UMA file that is created by the Caller App Editor. When the App is deployed, the UMA file is copied to each MAS. If you have sufficient privileges to view the MAS's disk, you will find the application files stored in folders underneath C:\Program Files\Avaya Modular Messaging\VServer\CallerApps.

The folders are given names that are the same as the Caller App's internal identifier; an example would be {39ECEB4F-37AA-4b8f-862A-9B119C227CB2}. To find a particular application, it's probably easiest to search under the CallerApps folder for a UMA file matching the name that you used when you deployed it. If you have deployed more than one application with the same file name, you will need to open the UMA files to find the one you want.

Note that the UMA files are found in numbered folders under the “internal identifier” folder. Normally, there will be only one such folder. Where there is more than one folder, always choose the UMA file in the highest-numbered folder.

Note. You should not need to use this information unless some catastrophe occurs (e.g. you lose the machine which holds your only copy of an application, and need to get it back to edit and redeploy it). Never change the contents of the MAS's disk if you need to recover a UMA file. Make a copy, and leave the original file in place. Changing the contents of the CallerApps folder (or any folders beneath it) may have deleterious (and unpredictable) effects on the behavior of one or more Caller Apps.

If a new App is deployed successfully to some MASs in a VMD, but not to others, is it added to the list of Apps in the VMD?

No. The VMD is supposed to act as a call-answering unit that behaves identically from the caller's point of view, no matter which MAS answers the call. Therefore, an App that is present on only some of the MASs in a VMD is not marked as present in the VMD.

What happens if I try to deploy an App that's already been successfully deployed?

If the App has changed, it will be redeployed and new callers will get the updated version. If the App has not changed, then although the Editor will contact each MAS in turn, the MAS will indicate that it already has an up-to-date copy of the App, so no further data transfer will take place. If some MASs in the VMD have out-of-date copies, then the new version will be transferred to them only.

Undeployment***If I undeploy a Caller App, will the copy of its UMA file actually be deleted from the MASs?***

Yes, but not until the last caller using the App (on a given MAS) has finished with it. The App is marked for removal on each MAS, and the Editor operation completes as soon as this has been done.

If I undeploy a Caller App that is referred to by another Caller App (with a "Goto Caller App" node), will this cause problems?

Yes, it will almost certainly cause problems. If a caller goes to the part of the application that tries to jump to the missing App, then the MAS will play an error message and will disconnect the call. You should check the deployed applications for errors with the Caller App Editor's Application Analysis feature.

Security***Who is allowed to create Caller Apps?***

Anyone who has access to the Caller Apps Editor can create a Caller App, and save it as a UMA file.

Who is allowed to deploy (or undeploy) Caller Apps?

Only someone who is designated a MM System Administrator (by having their user account, or a group containing it, added to the MM System Administrator access control list by VMSC) can deploy (or undeploy) a Caller App. If someone who is not a MM System Administrator attempts to deploy an App, the operation will fail.

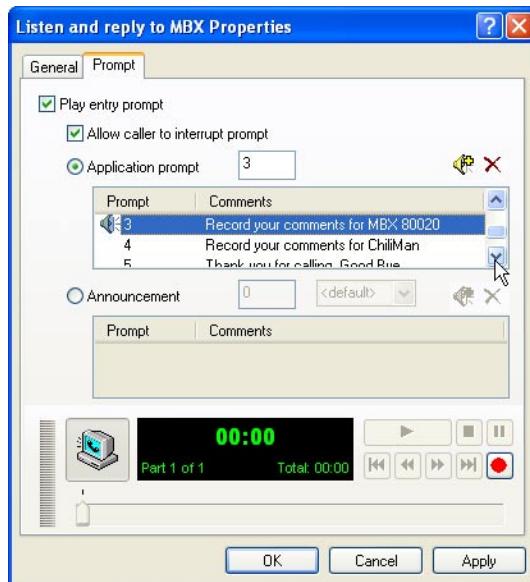
Who is allowed to add (or remove) Caller App Associations?

The same permissions model applied to deployment (see above) is applied to Caller App Associations.

"Front Door" Applications***What is a "Front Door" Caller App?***

A "Front Door" Caller App runs whenever a caller is transferred to a mailbox to which the App has been bound (by a MM User Administrator). The caller does not hear the usual behavior of the MM call answering TUI (probably, the personal greeting of the subscriber whose mailbox has been reached). Instead, the "Front Door" App runs, and the caller hears that.

A "Front Door" Caller App is always run, no matter how the caller happened to be redirected to the mailbox (e.g. ring-no-answer, busy diversion), and no matter what the subscriber may have configured by way of an extended absence greeting.



Application Prompts

Can I Create an Application Prompt from a WAV File?

Yes, but the WAV file must use the same encoding (G.711 PCM μ -law) that is used for all Caller Application prompts. You will need to create an empty application prompt in the usual way.

If the encoding is correct, then you can simply drag the WAV file from Explorer (or the desktop, etc.) into the Voice Control of the Caller Apps Editor's Prompt property page, and save the resulting file.

If the encoding is not correct (i.e. not G.711 PCM μ -law), then you must use an application such as Windows Sound Recorder to create a file with the correct encoding. To do this,

1. Open the file in Sound Recorder
2. Select File | Save As. . .
3. Enter a new name for the file
4. Click the Change. . . Button
5. Set the format to be CCITT μ -law
6. Click OK
7. Click Save

The resulting file can then be dragged and dropped on to the Voice Control (indicated by the red rectangle in the diagram on the left).

Subscriber Prompts

What are Subscriber Prompts?

Subscriber Prompts (also called *announcements*) are segments of audio that are recorded by a MM subscriber and are stored in that subscriber's mailbox. This is in contrast to Application Prompts, which are recorded by someone the Caller Apps Editor, and are stored within the App itself.

Because Subscriber Prompts aren't stored within the App, which only contains references to them, they require some extra attention from the App developer, MM subscriber administrator and MM subscriber.

- The App developer needs to use Subscriber Prompts in an appropriate way (see When should I use. . . ? Below), and to tell the subscriber administrator which Subscriber Prompts, in which mailboxes, have been referenced.
- The subscriber administrator needs to ensure that the subscribers whose mailboxes will contain the Subscriber Prompts have been enabled to record them, that those subscribers have been told which prompts are required, and what they should contain.
- The subscriber needs to record the audio content that is appropriate to the context in which the prompt is to be played to the caller.

How many Subscriber Prompts can a mailbox contain?

An MM subscriber's mailbox can contain up to thirty subscriber prompts.

Are Subscriber Prompts like greetings?

Yes and no. They are stored in the subscriber's mailbox and played to callers, so to that extent they are similar to greetings. But they get played because a Caller App refers to them, not just because a caller has been directed to the subscriber's mailbox. Greetings and Subscriber Prompts are completely separate: they don't overlap (you'll never hear a subscriber's greeting played by a Caller App) and the subscriber edits them in different ways.

How are Subscriber Prompts recorded?

A MM subscriber who has been permitted (by a MM User Administrator) to record Subscriber Prompts (announcements) must use the personal configuration (4) menu and select the announcements (5) option of the subscriber TUI to perform the recording. They must also specify the announcement number (1. . . 30). Currently there is no GUI (Subscriber Options) interface to Subscriber Prompt recording.

When should I use Subscriber Prompts in a Caller App?

The main use of Subscriber Prompts is to allow someone who is not an administrator to personalize the audio content of a Caller App. Let's start with a piece of general advice:

- Don't use Subscriber Prompts unless you need them.

There are two reasons for this recommendation:

1. Subscriber Prompts are a bit more complicated to set up (than Application Prompts); more people need to be involved in the process of getting them to be heard by the caller for the first time.
2. Subscriber Prompts are stored in the subscriber's mailbox (which is usually on a message server elsewhere on the network) rather than locally (on the MAS, as with Application Prompts). This makes them vulnerable to network problems. There is also a performance cost to pay for the round trip to the mailbox (to fetch the audio to play back).

So, why is it ever worth the trouble of setting them up and using them? The reason is that they make certain types of application much easier to maintain.

One kind of application that benefits enormously from Subscriber Prompts is the "Front Door" (see the "Front Door" Applications section). Here, a single Caller Application can be deployed, and used for many different subscribers. The trick is to refer the Subscriber Prompts to the *Default* mailbox when creating the

App. Because *Default* doesn't actually need to be resolved to an actual mailbox number at design time, only a single App is required. The actual mailbox number is supplied at run time, because the MAS knows which mailbox has been reached by the caller. Without Subscriber Prompts, the administrator would need to deploy a different Caller App for each subscriber with a "Front Door". This clearly does not scale well, and would be unacceptable in installations of any significant size.

Another type of application where Subscriber Prompts will almost certainly be useful is a bulletin board or information hotline service. Frequently, these have a fixed structure, perhaps rooted on a menu that says something like:

THANK YOU FOR CALLING THE XYZ INFORMATION SERVICES HOTLINE. TO HEAR ABOUT GENERAL NETWORK STATUS, PRESS 1. FOR E-MAIL SYSTEM STATUS, PRESS 2. FOR INFORMATION ON FUTURE SERVICE AND MAINTENANCE ACTIVITIES, PRESS 3. TO DISCONNECT, PRESS 9.

The 1, 2 and 3 options lead to nodes with Subscriber Prompts. The prompts are associated with mailboxes that belong to subscribers who are authorized to record information, and keep it updated frequently. Such a Subscriber Prompt might sound something like:

FRIDAY, FEBRUARY 8: 9.15 AM. WE HAVE BEEN EXPERIENCING SOME PROBLEMS WITH E-MAIL BETWEEN THE SAN FRANCISCO AND CHICAGO SITES. THIS IS BEING INVESTIGATED AND WE EXPECT TO CLEAR THE BACKLOG QUICKLY.

Different subscribers could be responsible for the various announcements.

There are many other applications that could benefit from the appropriate use of Subscriber Prompts.

Joining MM Voice Mail Domains

I've got a MM MAS that is joining an existing VMD. Do I need to redeploy all the Caller Apps in the VMD to this MAS?

No. This will happen automatically when you join the VMD, either during installation or by using VMSC to move the MAS from another VMD. The installation program (or VMSC) will display a progress dialog, as the applications in the VMD are automatically deployed to the MAS that is joining. If you skip the process, or it fails to complete successfully, you must manually redeploy the applications so that the new server obtains a copy.

If I'm moving an MAS from one VMD to another VMD, will all the old Caller Apps stored on the MAS be deleted?

No. Copies of the old Caller Apps UMA files will still be stored on the MAS. However, because none of these applications may be deployed in the new VMD, the MAS will not use them. Any Caller Applications in the new VMD will automatically be copied to the new MAS (see previous answer).



Language Support

Can a Caller App work in more than one language?

Yes. Just like the MAS's automated attendant and subscriber TUIs, a Caller App can interact with callers in different languages. The basic structure of the Caller App (menu options, transfer details, etc.) is the same for every language. The descriptions that the Editor can give to each node, and the prompts, are usually different for the languages in the App.

How does the language of the Caller App get chosen when it's running?

A multilingual Caller App will be of limited use unless the Voice Mail Domain in which it's running supports the same languages. The Caller Apps runtime will assume that the Caller App must be run in the default TUI language for the voice mail domain, unless a caller has previously interacted with the Automated Attendant and has chosen a different language. (Note: the standard MM Automated Attendant permits the caller to choose one of two or three languages, in a multilingual VMD).

How many languages can be contained in a Caller App?

You can add a language for each MM language pack that has been installed on the machine running the Caller App Editor. If you have more than one MM language pack installed, you will be given a choice of languages when you first create a Caller App.

The figure above shows the dialog box displayed when a new Caller App is created on a machine with English and Japanese language packs installed.

Once the application has been created, all the nodes and prompts will be labeled with the language that the Editor chose: English, in the dialog above. To create a multilingual Caller App, it's probably easiest to create the App in a single language first, get the structure and prompts right, and then add the second language (and maybe other languages) after the App has been tested.



To add a language to an existing Caller App, select the *Add language* checkbox when opening the application, and choose the language to add (from the drop-down list, formed from the installed MM language packs minus the languages already in the app). The node descriptions and prompts for the new language will automatically be populated, using copies of the values in the previous language added. Normally, you will want to re-record all the prompts in the new language, but the node descriptions don't have to be changed—or you may add some details in the new language.

How does the MAS know which languages a Caller App contains?

Caller Apps contain a list of the languages that they contain, so the MAS (and the Caller Apps Editor) can always find this out. However, the Caller Apps Editor does not know that the text entered in a “French” Caller App is actually in French, nor that the audio recorded for the prompts is in French. For example, it's possible to mark a Caller App as English, but record all the prompts in Japanese. Callers will, of course, only hear Japanese prompts.

Can I use a multilingual Caller App as a replacement Automated Attendant?

Yes, but only one language (the default VMD language, assuming it exists in the Caller App) will be played to callers. This is because the caller does not have an opportunity to choose a language before the Caller App starts to run: there is no default Automated Attendant to offer the choice.

So, how do I offer callers a choice of languages and use Caller Apps to provide the interface?

One way to do this is to create a Caller App that collects the caller's choice of language, and which branches to other Caller Apps according to their choice. Suppose that you wanted to support English and French in this way. Create a Caller App (in English) that feels and sounds like the replacement automated attendant for English callers. Let's call it *EnglishAA.uma*. Once you've saved it (and, ideally, deployed and tested it), use the Save As... option in the Caller App Editor and save the App into a new file, called *FrenchAA.uma*. Now you have two identical (but, to, the MASs, distinct) Caller Apps. Use the Editor to re-record all the prompts in *FrenchAA.uma* in French. (You can change the node descriptions if you like, but it's not required). Make sure that both the English and French Caller Apps are deployed successfully.

Now, create a Caller App (let's call it *ChooseLang.uma*) that contains an opening menu with a prompt that says something like:

XYZ CORPORATION. FOR ENGLISH, PRESS ONE. POUR FRANÇAIS, APPUYEZ DEUX.

Create a node, reached by the 1 key that is of type *Goto Caller Application*: make it jump to the *EnglishAA* application.

Create a node, reached by the 2 key, which is of type *Goto Caller Application*: make it jump to the *FrenchAA* application.

Note that you now have two Caller Apps (*EnglishAA*, *FrenchAA*) to change if you need to alter the structure or prompts.

Comparison with Aria Mailbox Types

The Aria series of voicemail systems allows extensive customization of the telephone user interface through the use of special mailbox types. The Information Center Mailbox (ICMB) feature, Enhanced Call Processing (ECP) mailboxes and other types of mailbox are familiar to Aria administrators. This section is designed to help Aria administrators who are trying to find the features that are supported by MM Caller Apps, and how.

What's the difference between a Caller App and an Aria Mailbox Type?

A Caller App is not a mailbox. Specifically, it does not require that a mailbox be created on the host e-mail system, or dedicated to its use. A Caller App contains one or more nodes, each of which can interact with the caller, and pass control to another node. The actions which a particular node performs will correspond, in many cases, to actions performed by certain Aria mailbox type.

Are all Aria Mailbox Types emulated by Caller Apps?

No. A Caller App is best thought of as a single-digit menu, comprising a collection of actions of the various types supported, that is:

- Menu (simple, or with extension, mailbox or caller app)
- Transfer (operator, extension or mailbox)
- Goto
- Conditional goto
- Send message (to mailbox or e-mail address)
- Termination (disconnection, default automated attendant, to mailbox logon, or dial by name)

These actions can be combined as required to produce useful applications. There is enough flexibility to duplicate, or at least approximate, the functionality of certain Aria mailbox types.

The following table lists Aria mailbox types and descriptions, and notes where MM Caller Applications can be used to produce the same (or a similar) effect.

Type	Description	CA	Notes
4	Bulletin broadcast	•	Bulletin broadcast mailboxes are not supported. Use an ELA to approximate this feature.
5	Announcement	•	An announcement can be created with a Caller App by using subscriber or application prompts and updating the audio content as required.
7	Fax delivery	•	
10	Forms	•	Send message actions can be used to record voice messages comprising a single audio segment, but they are sent separately to the recipient(s).
11	Transcriber	•	

Type	Description	CA	Notes
12	DTMF forms	•	
20	Hotel guest (DTMF)	•	
21	Hotel guest (rotary)	•	
22	Hotel front desk	•	
28	Extension	–	This can be approximated with an application that is a simple menu, with choices leading to actions of type Send message. Each Send message node is configured to send to a different subscriber's mailbox, and the prompt is provided by an announcement in the relevant subscriber's mailbox.
31	Caller's menu only	–	Simple menu action
32	Caller's menu + extension	–	Menu with extension action
33	Caller's menu + mailbox	–	Menu with mailbox action
40	Information processing	•	
41	InfoTex	•	
45	General fax message	–	Fax reception is supported for any mailbox.
58	Conference	–	Maximum message length can be set to a suitably large value (e.g. 2hr) for any mailbox.
60	Monitored	–	Mailboxes can be shared by sharing the credentials, but password changes are possible.
61	AMIS Loop Back Test	•	

Because a Caller Application can contain many actions, a single Caller App can provide the same experience to callers as several special Aria mailboxes.

Other Features

Is Dial-by-Name supported?

Yes. Callers will be able to use dial-by-name to specify a subscriber's name if you use either a *Menu with mailbox* or *Menu with extension* node in the application. When your application prompts the caller to enter the mailbox (or extension) number, they can press the # key and then spell the name just as they would in the MM Automated Attendant. As with the Automated Attendant, the system will attempt to match the name as they spell it, presenting the spoken names (or, if a spoken name is not recorded, a text-to-speech rendering of the display name) of the subscribers who match the spelling. With MM 1.1 SP5 and later you can also terminate to Dial by Name.

Standard MM system prompts are used after the caller presses the # key, but they may not be aware that this is possible unless your application's prompt for the menu contains some direction, such as: TO SPELL THE PERSON'S NAME, PRESS THE # KEY.

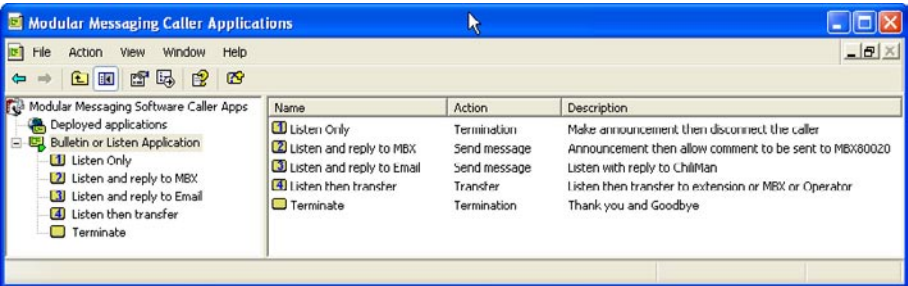
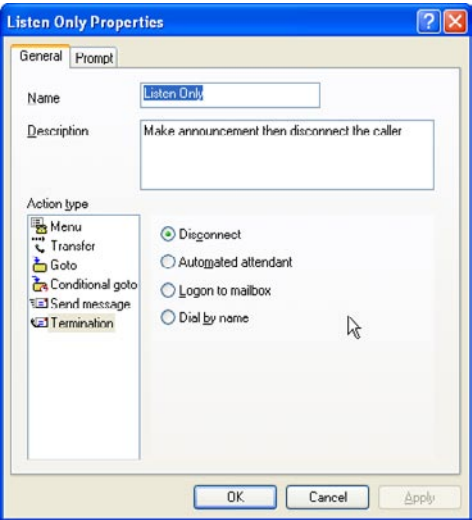
Examples of Caller Application Features

I want callers to hear an announcement, then disconnect the call

Create an application node that is of type **Termination**. Record the prompt, and select **Disconnect** for the specific action.

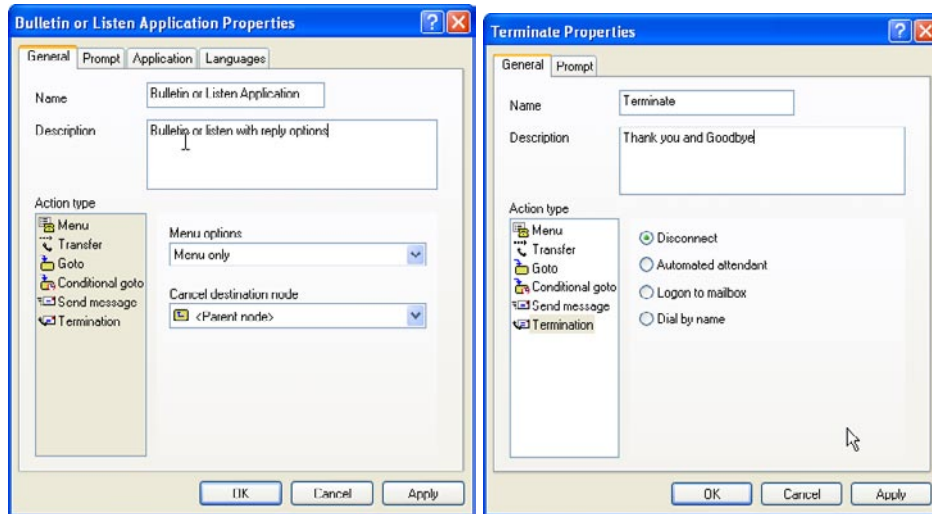
Alternatively, you could choose to have the caller route to the auto attendant after hearing the prompt.

If you want subscribers to hear something before they log on, then the **Logon to mailbox** action is the correct terminating action.



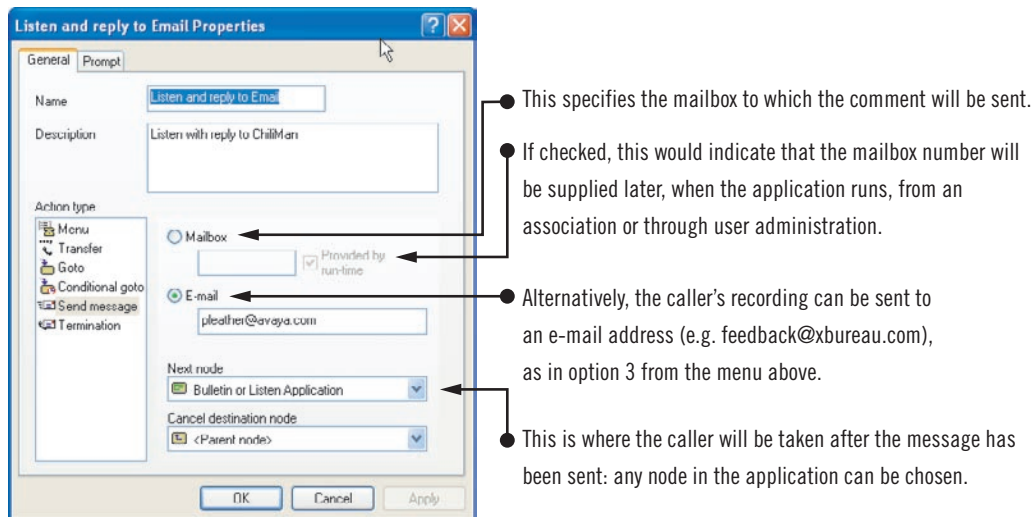
I want callers to hear an announcement, then leave a comment

Use the **Send Message** action: that will play the associated prompt, and then lets the caller record a voice response or comment, which is sent to a mailbox, or even an e-mail address.



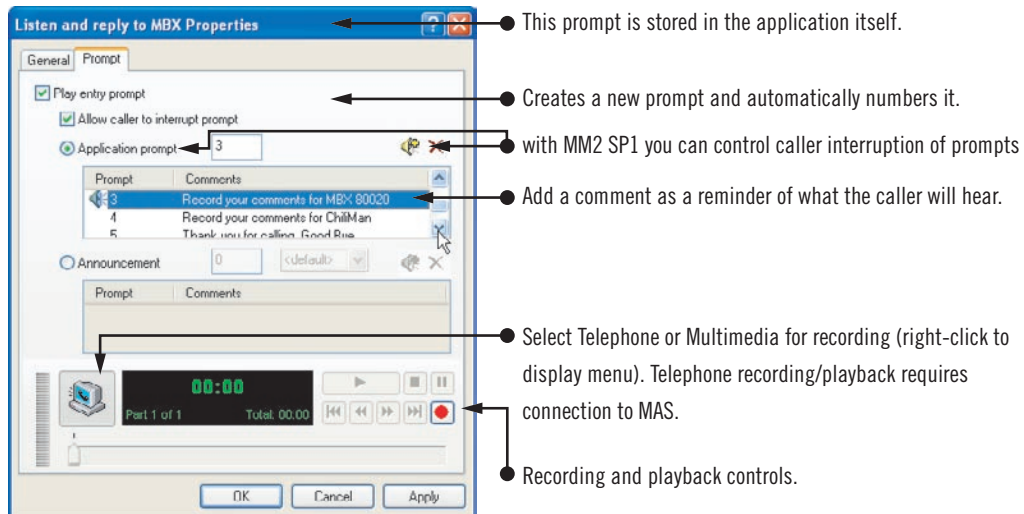
First, create a new application. The root node will be of type **Menu**, by default. Now add a **Terminate** node of specific type **Disconnect**. The example uses blank key, but you could use any available key not used for the menu. By convention, * is used to cancel, so it's probably best to avoid this.

Now, add other nodes below the menu. This is the node reached by pressing 2. It plays a prompt to the caller, then records a message and sends it to the specified mailbox number.



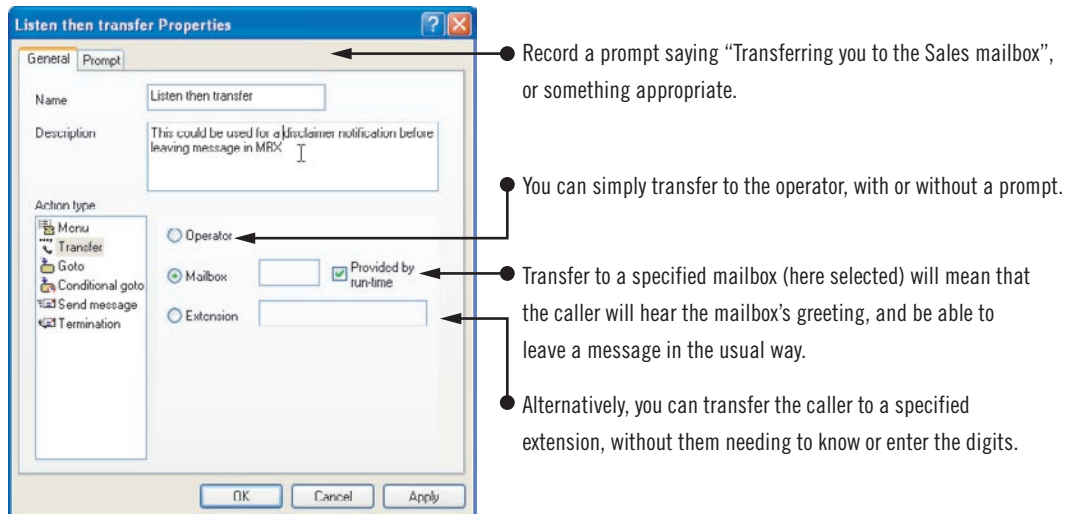
Go to the **Prompt** tab on the property page and record the prompt for this node (see below).

Repeat as required for the other nodes in the application.



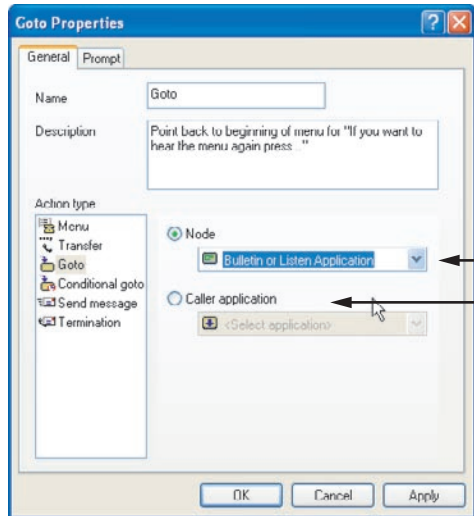
I want callers to hear an announcement or greeting, then be transferred

This is the node reached by pressing 4. It plays a prompt to the caller, then transfers them to the specified mailbox number.



I want give callers the option of hearing the menu again; I want to have one application jump to another application

In your application, create a node of type **Goto**. An example is shown below.

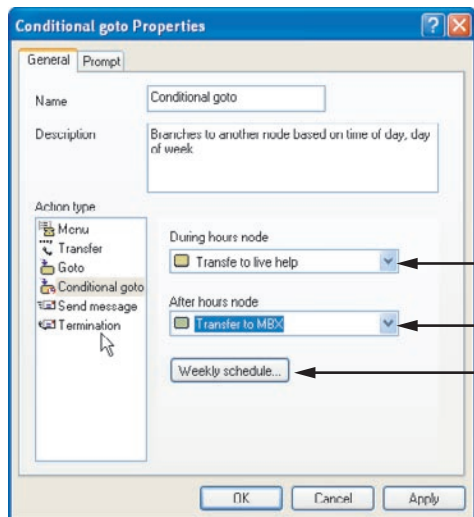


• The Goto Node action is used to route the caller to another node in the application, e.g repeating a top level menu.

• Alternatively, you can cause the caller to jump to the start of a different caller application. The destination application must already be deployed: you will need to select it from a drop-down list of apps deployed in the voice mail domain.

How do I make the app's behavior depend on the time of day?

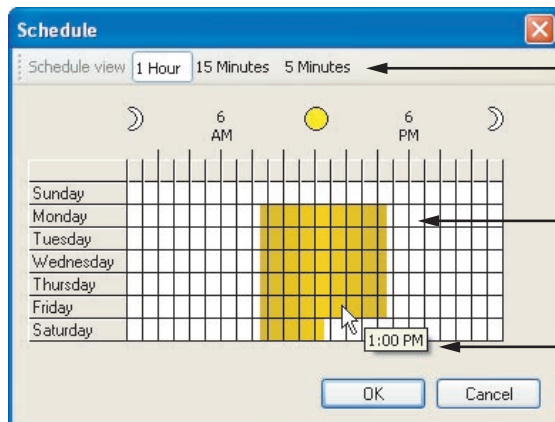
In your application, create a node of type **Conditional goto**. An example is shown below.



• When the schedule is active ("during hours"), the caller is transferred to one particular behaviour (e.g. live help, via an extension).

• When the schedule is inactive ("after hours"), the caller is transferred to a mailbox where they can record a message

• This button brings up the schedule editor (see below) to allow you to specify the hours during which the schedule is active, in a week.



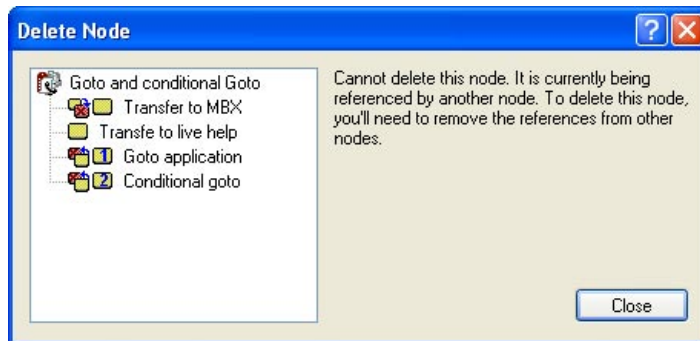
• Finer control of the settings is possible with different time scale views.

• This schedule is active from 8:30 AM to 4:30 PM, Monday to Friday, and from 8:30 AM to 12:30 PM on Saturday. Click (and drag) the left mouse button to change the settings.

• The "toolhelp caption" (1:00 PM) shows the current mouse pos

What happens if I try to delete a node that is referenced by another node?

The Caller Apps Editor will check to see if the node is referenced, and will disallow the deletion if so. Here's what happens if the user tries to delete the **Transfer to MBX** node that was used as a conditional goto destination in the example above.



About Avaya

Avaya enables businesses to achieve superior results by designing, building and managing their communications infrastructure and solutions. For over one million businesses worldwide, including more than 90 percent of the FORTUNE 500®, Avaya's embedded solutions help businesses enhance value, improve productivity and create competitive advantage by allowing people to be more productive and create more intelligent processes that satisfy customers.

For businesses large and small, Avaya is a world leader in secure, reliable IP telephony systems, communications applications and full life-cycle services. Driving the convergence of embedded voice and data communications with business applications, Avaya is distinguished by its combination of comprehensive, world-class products and services. Avaya helps customers across the globe leverage existing and new networks to achieve superior business results.

AVAYA

COMMUNICATIONS
AT THE HEART OF BUSINESS

avaya.com

© 2005 Avaya Inc.

All Rights Reserved. Avaya and the Avaya Logo are trademarks of Avaya Inc. and may be registered in certain jurisdictions. All trademarks identified by the ®, SM or TM are registered trademarks, service marks or trademarks, respectively, of Avaya Inc., with the exception of FORTUNE 500 which is a registered trademark of Time Inc. All other trademarks are the property of their respective owners.

Printed in the U.S.A.

04/05 • EF-MES2141-02