



IP Telephony

Contact Centers

Mobility

Services

WHITE  
PAPER

# Avaya Modular Messaging Caller Applications

White Paper and Frequently Asked Questions

May 2006



## Table of Contents

<b>Section 1: Introduction</b> .....	1
TUI Configuration without Caller Applications.....	1
The Need for More Configuration Options.....	1
<b>Section 2: Configuration Requirements</b> .....	2
<b>Section 3: Implementation</b> .....	3
<b>Section 4: Frequently Asked Questions</b> .....	4
General .....	4
Associations.....	5
Storage.....	5
Deployment .....	6
Undeployment.....	8
Security.....	8
“Front Door” Applications .....	8
Application Prompts .....	9
Subscriber Prompts .....	9
Joining Modular Messaging Voice Mail Domains.....	11
Language Support .....	11
Comparison with Aria Mailbox Types.....	13
Other Features .....	15
<b>Section 5: Examples of Caller Application Features</b> .....	15

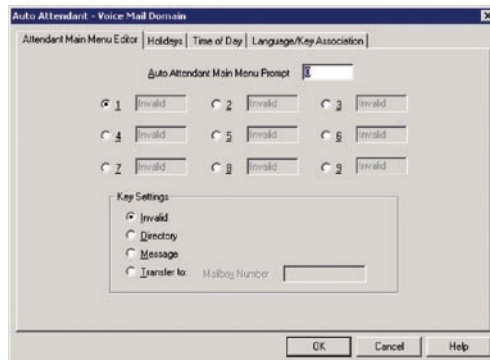
## Section 1: Introduction

As an inherent part of its design, Avaya Modular Messaging solutions provide enterprises with a telephone user interface (TUI) that enables subscribers and callers to access messaging easily regardless of their familiarity with the messaging system.

Within Modular Messaging, a *Caller Application* is a custom extension that can be added to the Modular Messaging TUI to provide a more flexible way to handle calls. These Caller Applications (often shortened to “Caller Apps”) can contain prompts, menus, and other actions (e.g., transfer call, record message) that can be assembled by an administrator or systems integrator to tailor Modular Messaging call answering to the specific needs of the enterprise.

### TUI Configuration without Caller Applications

With Avaya Modular Messaging, Messaging Application Servers (MASs) allow some configuration of the TUI “out of the box.” For example, the system’s Automated Attendant can be configured by using the Voice Mail System Configuration (VMSC) utility. During configuration, the VMSC property page might look like the figure below.



There are separate property pages for scheduled changes to prompting behavior and a multilingual opening menu, but the settings that most change the TUI behavior are on the Attendant Main Menu Editor page. There, administrators can specify the number of the main Automated Attendant prompt. (By default, this is *PLEASE ENTER THE MAILBOX NUMBER OF THE PERSON YOU ARE CALLING.*)

Administrators 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 telephone keypad. For instance, 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. The “directory” action allows a caller to use the dial-by-name functionality of the TUI (*PLEASE SPELL THE LAST NAME...*).

### The Need for More Configuration Options

There are some limitations to “out-of-the-box” configuration:

- **Fixed (single) depth of menu** — The Automated Attendant menu, if configured, has a fixed “depth ” of one 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, but also the maximum depth of menus is a detail that should not be limited by the architecture of any new system.

- **Static configuration** — Any Automated Attendant configuration made through the VMSC utility will only take effect when the various MASs in the Voice Mail Domain (VMD) are stopped and restarted. Where the configuration options are relatively limited, this may not be a problem, as changes will 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 configurable. However, much of the TUI falls outside the Automated Attendant, and some of it is just as suitable for configuration. For example, the behavior when a caller is transferred to a subscriber's mailbox is something that many enterprises might like to customize.
- **Only one Automated Attendant** — There is a single default Automated Attendant per voice mail domain. But different behaviors, depending on the number called, could be highly desirable for some applications. For example, an enterprise might publish different numbers for different caller queries and want to be able to provide a simplified TUI structure to the caller in each case.
- **Not Portable** — Systems integrators and implementation partners who install and set up Modular Messaging may wish to reuse some of their expertise in future installations. However, there is no way to export the default Automated Attendant settings from one VMD into another. As the number and complexity of TUI configurations increases, some mechanism for making these configurations portable and reusable would be highly desirable.

## Section 2: Configuration Requirements

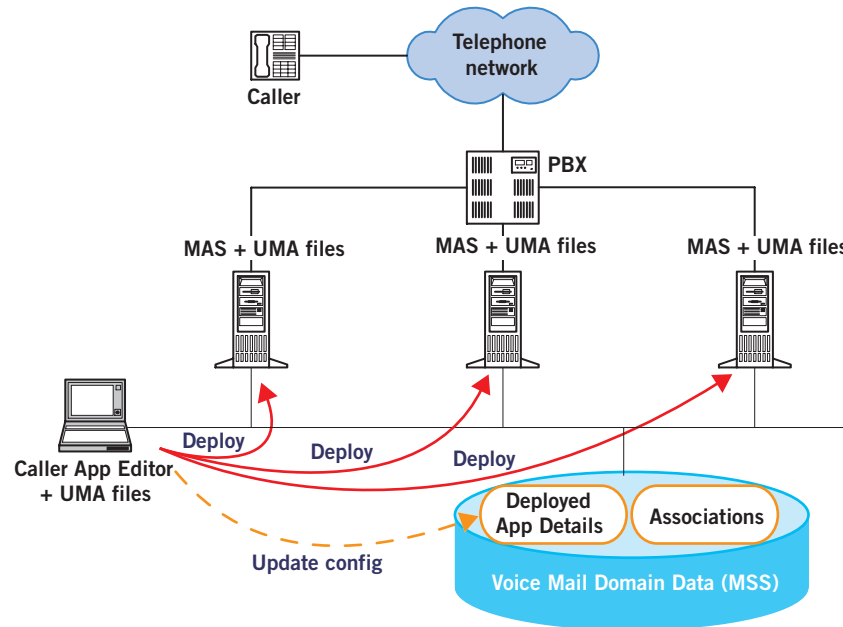
Within Modular Messaging, the term “Caller Applications” refers to the feature set that tries to address the configuration limitations outlined above. Each Caller App is a self-contained part of the overall Modular Messaging TUI and can be added/linked to the TUI to provide a more flexible way to handle callers.

The Caller Apps attempt to satisfy these configuration requirements:

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

### Section 3: Implementation

The figure below shows the way that the data involved with Caller Applications is distributed in Modular Messaging.



The Caller Apps 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 Unified Messaging Application (UMA) files, which are often (but not necessarily) originated and initially stored on the Editor machine. When an administrator needs to make an application available to callers, he/she uses the Editor to transfer a copy of the UMA file to the MASs in the VMD. This process is called deployment. If the deployment succeeds, the VMD configuration data is updated to show that the application is present. The Editor may then be used to create one or more associations (also stored in the VMD configuration data) to specify the conditions under which the MASs will run the particular Caller App.

The implementation of Caller Apps in Modular Messaging meets the requirements (outlined above) in the following ways:

- **Menus of any depth, functionality** — The Caller App 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 Modular Messaging VMD. As long as the administrator 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 application 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 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** — The Editor can include references to prompts recorded by Modular Messaging subscribers in any Caller App. Administrators can bind a Caller App to a subscriber's mailbox, and the App will be executed in place of the normal greeting when control is transferred to the mailbox.

## Section 4: Frequently Asked Questions

### General

#### What is a Caller Application?

A Caller Application (or Caller App) is a custom extension to the Avaya Modular Messaging TUI and used by administrators to customize Modular Messaging call answering.

Caller Apps can extend or even replace the Modular Messaging system Automated Attendant. They can be invoked when callers to an Automated Attendant enter specified numbers, when a call is placed to a particular extension, or in place of a subscriber's normal 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 VMD. Once deployed, a Caller App is available until an administrator decides to replace or remove it. The 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 Editor (a snap-in for Microsoft Management Console) can create a Caller App and save it to a file, only Modular Messaging 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 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, such as when:

- The App refers to one or more Subscriber Prompts (see page 9).
- 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 8).

Note that although a Caller App may refer to (or be associated with) a mailbox, it is in no way part of the mailbox itself. An App is simply a custom-TUI behavior that is stored in a file and 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 caller. Every association must contain the identity of a deployed Caller App. The association must also specify at least one of the following:

- **An association ID** — This is a number that uniquely identifies the association within the VMD. If a caller enters this number on his/her telephone keypad in response to the Automated Attendant 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.
- **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 the call (see above), the number that is entered is checked against Caller App association IDs and if no match is found, only then checked 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 the hidden mailbox's owner. Address resolution when sending messages never checks Caller App associations.

Modular Messaging system administrators may find this technique (hiding mailboxes from callers) useful in some circumstances, but it may cause confusion if used carelessly. An example is hiding an Enhanced List Application (ELA) mailbox so that outside calls are not accidentally sent to a distribution list. In this case, the Caller App might notify callers that it is an invalid extension and either disconnect or send them to the Automated Attendant or another Caller App for further processing.

### 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 VMD to avoid potential problems. For example, if the association ID is shorter than the VMD mailbox number, the Automated Attendant must execute a time-out for each of the “missing digits” when a caller enters what is at first presumed to be a mailbox number. The resulting 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, be specified as to be Provided by runtime. The Mailbox Number field in the association supplies the required mailbox number for these cases.

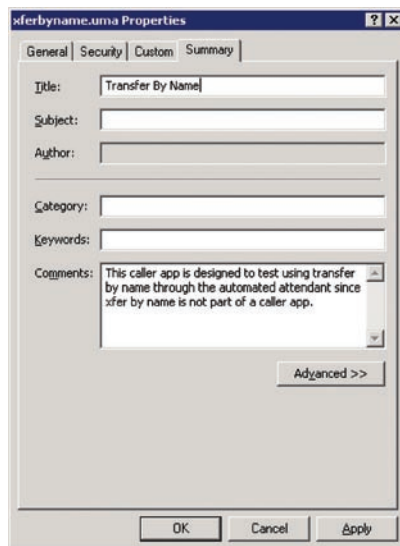
## Storage

### How much space do 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 App prompts are stored in a format that occupies 8,000 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  $(10 \times 5 \times 8,000 =) 400,000$  bytes long.

### Why does the Editor suggest UMA as a file type for the created Apps?

It will be easier to find applications if you stick to this convention. When you install the Caller App Editor, it is registered to be the application that handles UMA (Unified Messaging Application) files if you open them from Windows Explorer.



### Does anything except Modular Messaging 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.

A UMA file 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 the Avaya 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 VMD?

It seems as if it would be more convenient for an administrator to deploy a Caller App to a single MAS and have it replicate from there to other MASs in the domain. However, the VMSC utility changes other VMD settings so that instead, the Caller App Editor goes to each MAS in turn and transfers a copy of the 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 a MAS is disconnected from the network. If an administrator decides to replace the entire Modular Messaging Automated Attendant with a Caller App, it must be as robust as possible in the event that a network problem occurs. In particular, callers should not experience a significant lack of service.

Therefore, Caller App deployment attempts to transfer a copy of the entire application to each MAS. The person performing the deployment can tell whether or not the application has been successfully transferred to each MAS. 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 App is in use 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. An administrator using the Editor to deploy a Caller App will get an error if the MAS with which the Editor 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. Those users/administrators with privileges to view the MAS disk will find the application files stored in folders underneath C:\ProgramFiles\AvayaModularMessaging\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, search under the Caller Apps folder for a UMA file matching the name that used when the App was deployed. If more than one application has been deployed with the same file name, the user/administrator will need to open the UMA files to find the correct one.

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.** This information should not be needed unless some catastrophe occurs (e.g., there is a loss of the machine that holds the only copy of an application, and it needs to be retrieved, edited, and redeployed). Never change the contents of the MAS disk when recovering a UMA file; simply make a copy, and leave the original file in place. Changing the contents of the Caller Apps folder (or any folders beneath it) may have serious (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 a 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 someone tries to deploy an App that has 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, and 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 an administrator undeploys 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 accomplished.

### If an administrator undeploys a Caller App that is referred to by another Caller App (with a “Go to 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 disconnect the call. Check the deployed applications for errors using the Application Analysis feature of the Caller App Editor.

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

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

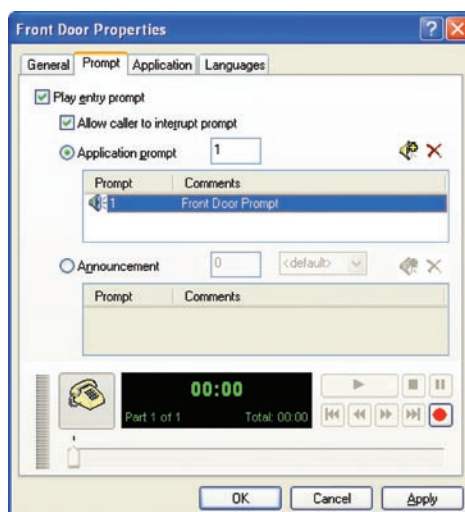
The same permissions model applied to deployment (see above) is applied to Caller App associations.

## “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 Modular Messaging administrator). The caller does not hear the usual behavior of the Modular Messaging 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 a Caller Application Prompt be created from a WAV File?

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

If the encoding is correct, simply drag the WAV file from Explorer (or the desktop, etc.) into *Voice Control* on the Caller Apps Editor Prompt property page, and save the resulting file.

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

- Open the file in *Sound Recorder*.
- Select *File/Save As...*
- Enter a new name for the file.
- Click the *Change...* button.
- Set the format to CCITT  $\mu$ -law.
- Click *OK*.
- Click *Save*.

The resulting file can then be dragged and dropped on to the Voice Control.

## Subscriber Prompts

### What are Subscriber Prompts?

Subscriber Prompts (also called announcements) are segments of audio that are recorded by a Modular Messaging subscriber and stored in that subscriber's mailbox. This is in contrast to Application Prompts, which are recorded by someone using 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 application developer, the Modular Messaging administrator, and the subscriber.

- The application developer needs to use Subscriber Prompts in an appropriate way (see "When should Subscriber Prompts be used...?" below) and tell the administrator which Subscriber Prompts, in which mailboxes, have been referenced.
- The administrator needs to ensure that the subscribers whose mailboxes will contain the Subscriber Prompts have been enabled to record them and 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?

A Modular Messaging subscriber's mailbox can contain up to 30 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?**

Subscriber Prompts (announcements) may be recorded by a Modular Messaging subscriber only as permitted by the administrator. For example, Aria® subscribers must use the personal configuration menu (4) and select the announcements option (5) on the TUI to perform the recording. They must also specify the announcement number (1-30). Currently there is no GUI interface (Subscriber Options) for recording Subscriber Prompts. (For Serenade® and INTUITY™ sequences, refer to the respective TUI Guide.)

### **When should Subscriber Prompts be used 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.

That being said, in general Subscriber Prompts should not be used unless absolutely necessary. There are two reasons for this recommendation:

1. Subscriber Prompts are a bit more complicated to set up than Application Prompts, and more people need to be involved in the process.
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, when is it worth the trouble of setting up and using Subscriber Prompts? They do make certain types of application much easier to maintain. For example:

- With Front Door Applications (see page 8), a single Caller App 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 are useful is a bulletin board or information hotline service. Frequently, these have a fixed structure, perhaps rooted on a menu that says something such as *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 can lead to nodes with Subscriber Prompts. The prompts can be associated with mailboxes that belong to subscribers who are authorized to record information and keep it updated frequently. Such a Subscriber Prompt might be sound something such as *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 Modular Messaging Voice Mail Domains**

**With a Modular Messaging MAS that is joining an existing VMD, do all the Caller Apps in the VMD need to be redeployed to this MAS?**

No. This will happen automatically, 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 this process is skipped or it fails to complete successfully, the applications must be manually redeployed so that the new server obtains a copy.

**When moving a 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 App 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. As with the MAS 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.

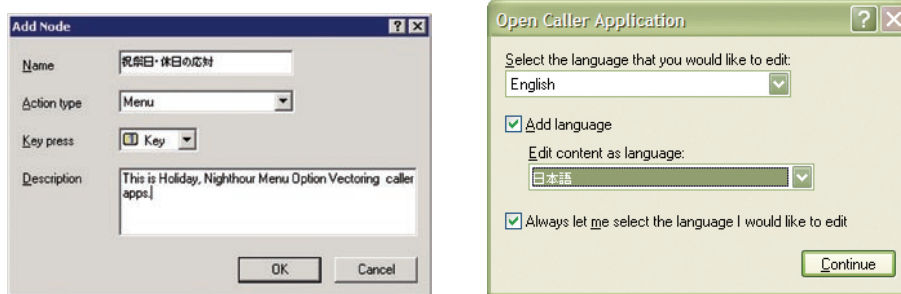
**How does the language of the Caller App get chosen when it's running?**

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

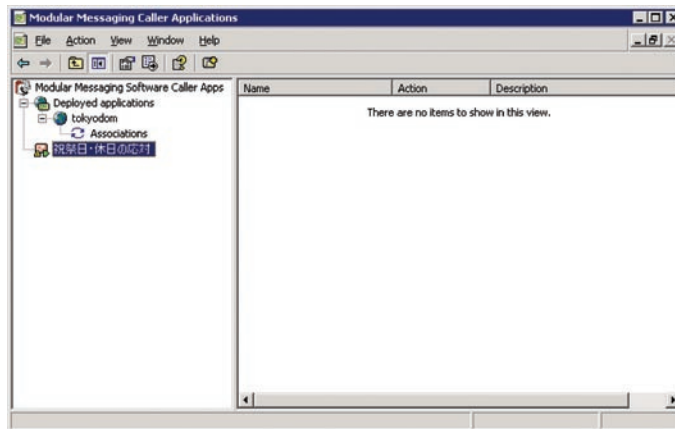
**How many languages can be contained in a Caller App?**

A language can be added for each Modular Messaging language pack that has been installed on the machine running the Caller App Editor. If there is more than one Modular Messaging language pack installed, the administrator will be given a choice of languages when first creating a Caller App.

The figure below 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 easiest to create the App in a single language first, get the structure and prompts right, and then add the second language (and perhaps 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 Modular Messaging 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, the administrator will want to rerecord all the prompts in the new language, but the node descriptions don't have to be changed (although some details in the new language maybe added).

#### 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 information. 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 would, of course, only hear Japanese prompts.

#### Can a multilingual Caller App be used 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 is a choice of languages offered when using 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 branches to other Caller Apps according to that choice. For example, to support English and French in this way, you could 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 it has been saved it (and ideally, deployed and tested), use the *Save As...* option in the Caller App Editor to save the App into a new file called FrenchAA.uma. Now there are two identical Caller Apps on distinct MASs. Then use the Editor to rerecord 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 such as *XYZ CORPORATION. FOR ENGLISH, PRESS ONE. POUR FRANÇAIS, APPUYEZ DEUX.*

Create a node that is of the type *Go to Caller Application* and reached by the 1 key, and make it jump to the EnglishAA application. Next, create a node of the type *Go to Caller Application* and reached by the 2 key, and 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 voice mail systems allows extensive customization of its TUI through the use of special mailbox types. The Information Center Mailbox (ICMB) feature, Enhanced Call Processing (ECP) mailboxes, and other types of mailboxes are familiar to Aria administrators. This section is designed to help them find the Aria features that are supported by Modular Messaging Caller Apps, and how.

**What is 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. But in many cases, the actions that a particular node performs will correspond 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)
- Go to
- Conditional go to
- 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 Modular Messaging 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	√	
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 <i>Send message</i> . Each <i>Send message</i> 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., 2 hours) 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.

**Note:** Avaya DevConnect partners have solutions for transcriber and forms MBXs.



## Other Features

### Is Dial-by-Name supported?

Yes. Callers can use dial-by-name to specify a subscriber's name if the administrator uses a *Menu with mailbox* or *Menu with extension* node in the application. When the 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 Modular Messaging 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 subscriber who matches the spelling. The system can also terminate to dial-by-name.<sup>1</sup>

Standard Modular Messaging system prompts are used after the caller presses the # key, but callers may not be aware of this unless the application prompt for the menu contains some direction such as *TO SPELL THE PERSON'S NAME, PRESS THE #KEY*.

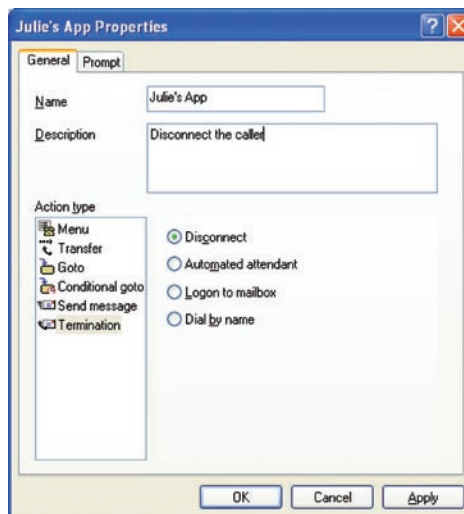
<sup>1</sup> With Modular Messaging 1.1 Service Pack 5 and later.

## Section 5: Examples of Caller Application Features

### I want to provide callers with an announcement and then disconnect the call.

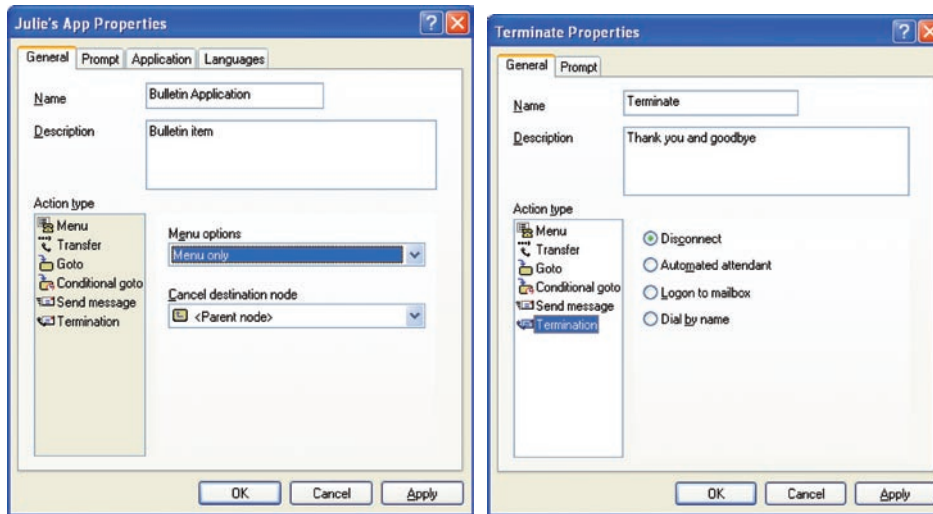
Create an application node that is of type *Termination*. Record the prompt, and select *Disconnect* for the specific action.

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.



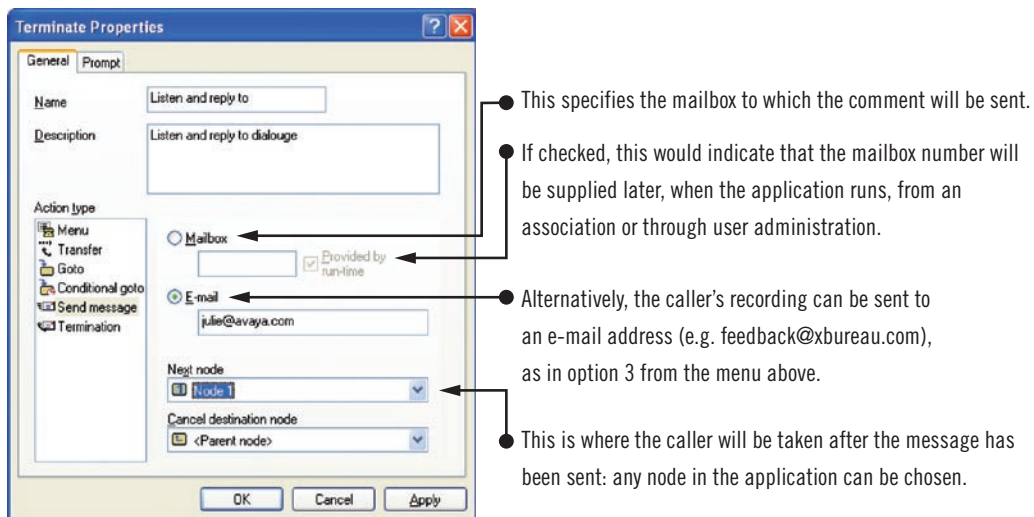
### I want callers to hear an announcement and then leave a comment.

Use the *Send Message* action. It will play the associated prompt and then let 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 a 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 using it here.)

Now, add other nodes below the menu. The example shown 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.



This specifies the mailbox to which the comment will be sent.

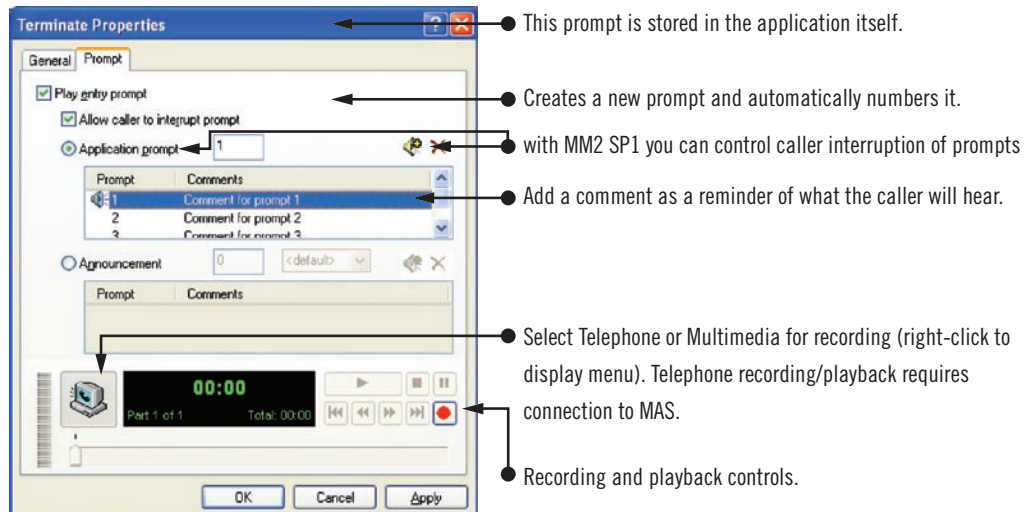
If checked, this would indicate that the mailbox number will be supplied later, when the application runs, from an association or through user administration.

Alternatively, the caller's recording can be sent to an e-mail address (e.g. feedback@xbureau.com), as in option 3 from the menu above.

This is where the caller will be taken after the message has been sent; any node in the application can be chosen.

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.



This prompt is stored in the application itself.

Creates a new prompt and automatically numbers it.

Control caller interruption of prompts.<sup>2</sup>

Add a comment as a reminder of what the caller will hear.

Select *Telephone* or *Multimedia* for recording (right-click to display menu). Telephone recording/playback requires connection to a MAS.

Recording and playback controls.

<sup>2</sup> With Modular Messaging 2 Service Pack 1 and greater.

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

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

The screenshot shows the 'Terminate Properties' dialog box with the following annotations:

- Name:** Location from transfer
- Description:** This should be used for administration notifications before
- Action type:**
  - Operator
  - Mailbox (with a text input field)
  - Extension (with a text input field)
- Provided by run-time:**  (checked)

Annotations with arrows point to the following elements:

- The **Prompt** tab (indicated by an arrow pointing left).
- The **Name** field.
- The **Description** field.
- The **Operator** radio button.
- The **Mailbox** radio button and its associated text input field.
- The **Extension** radio button and its associated text input field.
- The **Provided by run-time** checkbox.

Record a prompt saying “Transferring you to the Sales mailbox” or something appropriate.

Or simply transfer to the operator, with or without a prompt.

Transfer to a specified mailbox (here, selected) will mean that the caller will hear the mailbox greeting and be able to leave a message in the usual way.

An alternative is to transfer the caller to a specified extension, without them needing to know or enter the digits.

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

In the application, create a node of type Goto.

The screenshot shows the 'Goto Properties' dialog box with the following annotations:

- Name:** Goto
- Description:** Points back to the beginning of the application
- Action type:**
  - Node (with a dropdown menu showing 'Julie's App')
  - Caller application (with a dropdown menu showing '<Select application>')

Annotations with arrows point to the following elements:

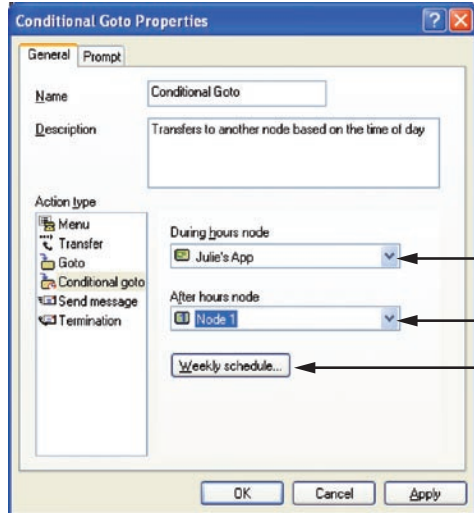
- The **Node** radio button.
- The dropdown menu for the **Node** action, currently showing 'Julie's App'.
- The **Caller application** radio button.
- The dropdown menu for the **Caller application** action, currently showing '<Select application>'.

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

An alternative is to cause the caller to jump to the start of a different Caller App. The destination application must already be deployed so it can be selected from a drop-down list of the Apps deployed in the VMD.

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

In the application, create a node of type Conditional goto .

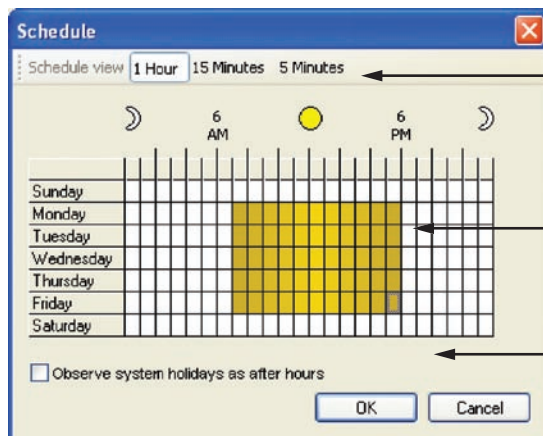


- When the schedule is active (“during hours”), the caller is transferred to one particular behaviour (e.g. live help, via an extension).
- When the schedule is inactive (“after hours”), the caller is transferred to a mailbox where they can record a message.
- This button brings up the schedule editor (see below) to allow you to specify the hours during which the schedule is active, in a week.

When the schedule is active (“during hours”), the caller is transferred to one particular behavior (e.g., live help, via an extension).

When the schedule is inactive (“after hours”), the caller is transferred to a mailbox where he/she can record a message.

This button brings up the Schedule Editor (see below) 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 position.

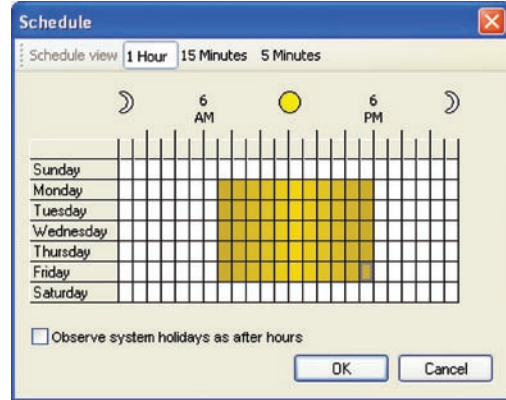
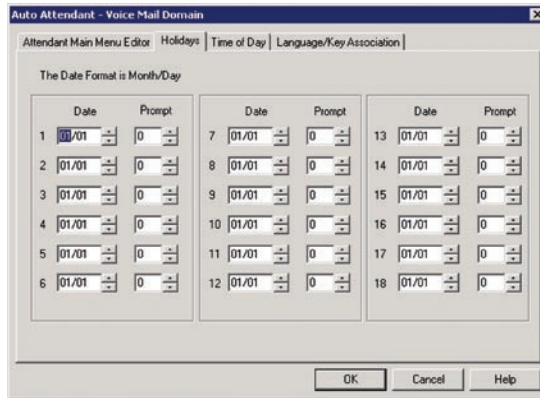
Finer control of the settings is possible with different time scale views.

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.

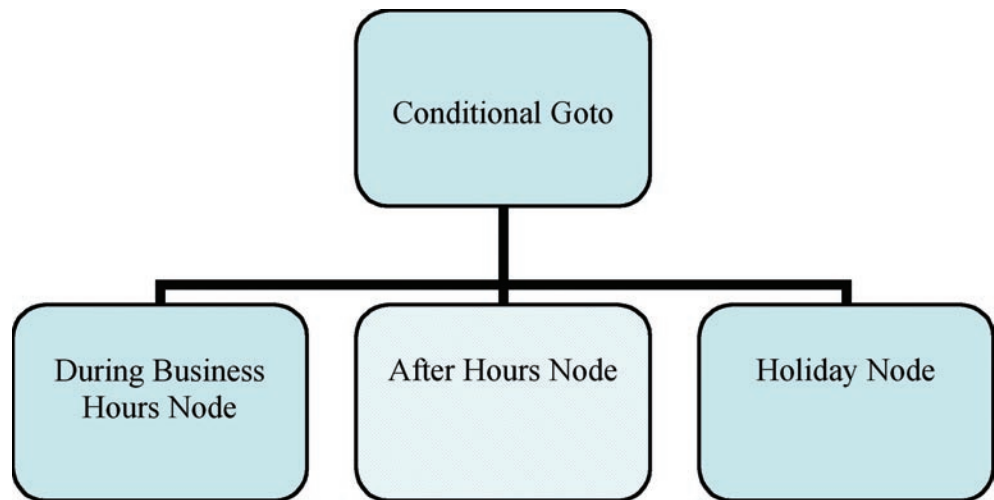
The “toolhelp caption” (1:00 PM) shows the current mouse position.

### How do I incorporate holidays into my Conditional Goto?

First, enter the holiday dates on the Auto Attendant using the VMSC. Next, use the Holidays checkbox on the *Conditional goto*.

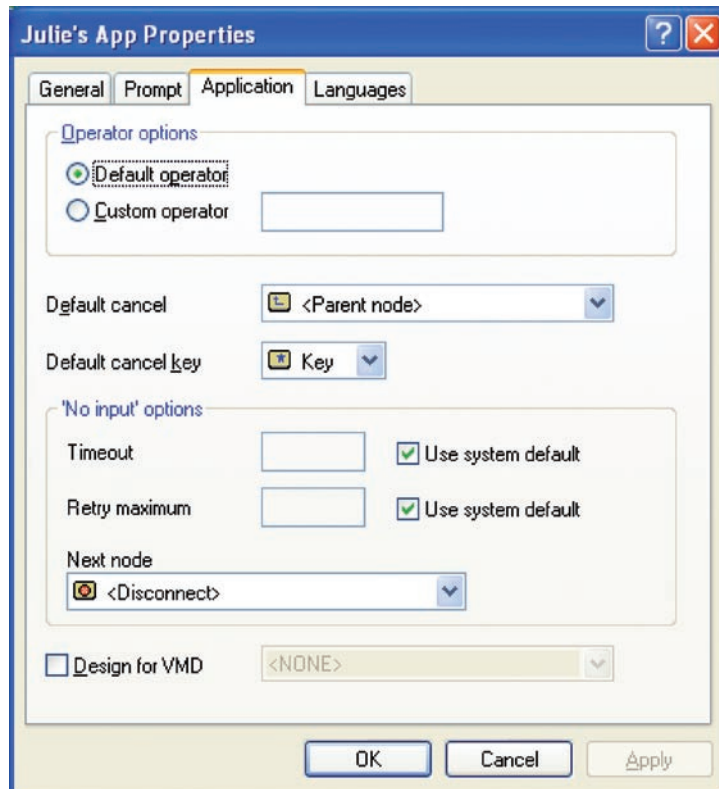


The basic logic of during hours, after hours, and holidays can be daisy-chained together to create more combinations, such as different behavior on the weekends.



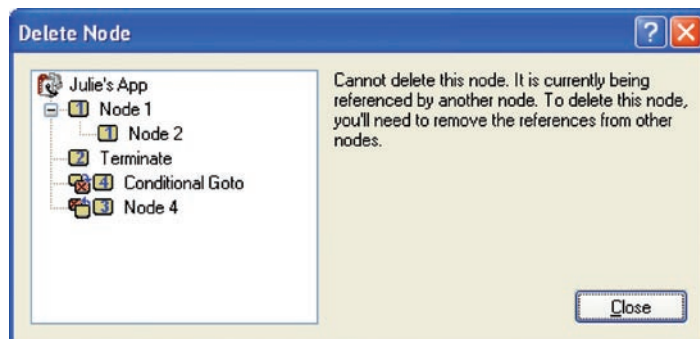
### What happens if a caller doesn't respond to a prompt?

Caller Apps will use a system default, which is to repeat the prompt three times and then disconnect. You can set the number of times to repeat, how long to wait for a response, and where to go if the caller doesn't provide any input/response.



### 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 if so, will disallow the deletion. In the example below, the user tried to delete the *Transfer to MBX* node that is used as a *Conditional goto* destination in the example below.



## 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](http://avaya.com)

© 2006 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.

09/06 • MES2141-03