

# SPEECHSOFT, INC. CALLMASTER LICENSE AGREEMENT

**IMPORTANT: YOUR USE OF SOFTWARE SUPPLIED BY SPEECHSOFT INDICATES YOUR ACCEPTANCE OF THE FOLLOWING TERMS AND CONDITIONS.**

SpeechSoft agrees to grant you a non-exclusive license to use the enclosed software program only on the terms specified herein.

**Important - Read Carefully:** this SpeechSoft CALLMaster license agreement (CLA) is a legal agreement between you (either an individual or a single entity) and SpeechSoft corporation for the SpeechSoft software product identified above, which includes computer software and may include associated media, printed materials, and online or electronic documentation (software product). The software product also includes any updates and supplements to the original software product provided to you by SpeechSoft. By installing, copying, downloading, accessing or otherwise using the software product, you agree to be bound by the terms of this CLA. If you do not agree to the terms of this CLA, do not install or use the software product; you may, however, return it to your place of purchase for a refund based on their refund policy.

## Permitted Uses

You may use this program on a single computer. That computer must be set-up with a software license file from SpeechSoft. The software license file is locked to one of the dialogic boards installed on the pc to be used by CALLMaster.

You may make a single back up of the program.

You may transfer the program and software license with the associated dialogic board to another party if the party agrees to the terms of this agreement and you destroy all your copies of the program. If you transfer the program or any copy to another party, your license and right to use the program is automatically terminated.

## Software Product License

**The software product is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The software product is licensed, not sold.**

### 1. Grant of license. This CLA grants you the following rights:

- Applications software. You may install, use, access, display, run, or otherwise interact with (run) one copy of the software product, or any prior version for the same operating system, on a single computer, workstation, terminal, or other digital electronic device (computer). The primary user of the computer on which the software product is installed may make a second copy for his or her exclusive use as backup.
- You must acquire and dedicate a license for each separate computer on which the software product is run. A license for the software product may not be shared or used concurrently on different computers.
- Reservation of rights. All rights not expressly granted are reserved by SpeechSoft.

### 2. Description of other rights and limitations.

- Limitations on reverse engineering, de-compilation, and disassembly. You may not reverse engineer, decompile, or disassemble the software product, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation.
- Separation of components. The software product is licensed as a single product. Its component parts may not be separated for use on more than one computer.
- Trademarks. This CLA does not grant you any rights in connection with any trademarks or service marks of SpeechSoft.
- Rental. You may not rent, lease, or lend the software product unless granted that ability in writing from SpeechSoft.
- Support services. SpeechSoft may provide you with support services related to the software product (support services). Use of support services is governed by the SpeechSoft policies and programs described in the user manual, in online documentation, and/or in other SpeechSoft-provided

materials. Any supplemental software code provided to you as part of the support services shall be considered part of the software product and subject to the terms and conditions of this CLA. With respect to technical information you provide to SpeechSoft as part of the support services, SpeechSoft may use such information for its business purposes, including for product support and development. SpeechSoft will not utilize such technical information in a form that personally identifies you.

- **Software transfer.** The initial licensee of the software product may make a one-time permanent transfer of this CLA and software product only directly to an end user. This transfer must include all of the software product (including all component parts, the media and printed materials, any upgrades, this CLA, the software license file and, if applicable, the certificate of authenticity). Such transfer may not be by way of consignment or any other indirect transfer. The transferee of such one-time transfer must agree
- To comply with the terms of this CLA, including the obligation not to further transfer this CLA and software product.
- **Termination.** Without prejudice to any other rights, SpeechSoft may terminate this CLA if you fail to comply with the terms and conditions of this CLA. In such event, you must destroy all copies of the software product and all of its component parts.

**3. Upgrades.** If the software product is labeled as an upgrade, you must be properly licensed to use a product identified by SpeechSoft as being eligible for the upgrade in order to use the software product. A software product labeled as an upgrade replaces and/or supplements (and may disable) the product that formed the basis for your eligibility for the upgrade. You may use the resulting upgraded product only in accordance with the terms of this CLA. If the software product is an upgrade of a component of a package of software programs that you licensed as a single product, the software product may be used and transferred only as part of that single product package and may not be separated for use on more than one computer.

**4. Copyright.** All title and copyrights in and to the software product (including but not limited to any images, photographs, animations, video, audio, music, text, and applets incorporated into the software product), the accompanying printed materials, and any copies of the software product are owned by SpeechSoft or its suppliers. All title and intellectual property rights in and to the content which may be accessed through use of the software product is the property of the respective content owner and may be protected by applicable copyright or other intellectual property laws and treaties. This CLA grants you no rights to use such content. If this software product contains documentation which is provided only in electronic form, you may print one copy of such electronic documentation.

**5. Dual-media Software.** You may receive the software product in more than one medium. Regardless of the type or size of medium you receive, you may use only one medium that is appropriate for your single computer. You may not run the other medium on another computer. You may not loan, rent, lease, or otherwise transfer the other medium to another user, except as part of the permanent transfer (as provided above) of the software product.

**6. Backup Copy.** After installation of one copy of the software product pursuant to this CLA, you may keep the original media on which the software product was provided by SpeechSoft solely for backup or archival purposes. If the original media is required to use the software product on the computer, you may make one copy of the software product solely for backup or archival purposes. Except as expressly provided in this CLA, you may not otherwise make copies of the software product or the printed material accompanying the software product.

## **Miscellaneous**

**This CLA is governed by the laws of the State of New York.**

Should you have any questions concerning this CLA, or if you desire to contact SpeechSoft for any reason, please contact the SpeechSoft sales

## **Limited Warranty for Software Products**

SpeechSoft warrants that (a) the software product will perform substantially in accordance with the accompanying written materials for a period of thirty (30) days from the date of receipt, and (b) any support services provided by SpeechSoft shall be substantially as described in applicable written materials provided to you by SpeechSoft, and SpeechSoft support engineers will make commercially reasonable efforts to solve

any problem issues. Some states and jurisdictions do not allow limitations on duration of an implied warranty, so the above limitation may not apply to you. To the extent allowed by applicable law, implied warranties on the software product, if any, are limited to thirty (30) days.

**Customer Remedies.** SpeechSoft's and its suppliers' entire liability and your exclusive remedy shall be, at SpeechSoft's option, either (a) return of the price paid minus restocking fee's, if any, or (b) repair or replacement of the software product that does not meet SpeechSoft's limited warranty and which is returned to SpeechSoft with a copy of your receipt. This limited warranty is void if failure of the software product has resulted from accident, abuse, or misapplication. Any replacement software product will be warranted for the remainder of the original warranty period or thirty (30) days, whichever is longer.

### **No Other Warranties**

To the maximum extent permitted by applicable law, SpeechSoft and its suppliers disCLAIM all other warranties and conditions, either express or implied, including, but not limited to, implied warranties of merchantability, fitness for a particular purpose, title, and non-infringement, with regard to the software product, and the provision of or failure to provide support services. This limited warranty gives you specific legal rights. You may have others, which vary from state/jurisdiction to state/jurisdiction.

### **Limitation of Liability**

To the maximum extent permitted by applicable law, in no event shall SpeechSoft or its suppliers be liable for any special, incidental, indirect, or consequential damages whatsoever (including, without limitation, damages for loss of business profits,

Business interruption, loss of business information, or any other pecuniary loss) arising out of the use of or inability to use the software product or the provision of or failure to provide support services, even if SpeechSoft has been advised of the possibility of such damages. In any case, SpeechSoft's entire liability under any provision of this CLA shall be limited to the greater of the amount actually paid by you for the software product or USD5.00; provided, however, if you have entered into a SpeechSoft support services agreement, SpeechSoft's entire liability regarding support services shall be governed by the terms of that agreement. Because some states and jurisdictions do not allow the exclusion or limitation of liability, the above limitation may not apply to you.

### **Whole Agreement**

This agreement constitutes the entire agreement between the two parties, and may be changed only by a written agreement of both parties. The invalidity or unenforceability of any part of this agreement shall not affect the validity or enforceability of any remaining parts, all of which shall remain in full force. This agreement shall be governed by the laws of the state of New York.

# SPEECHSOFT, INC. CALLMASTER LICENSE AGREEMENT

**IMPORTANT: YOUR USE OF SOFTWARE SUPPLIED BY SPEECHSOFT INDICATES YOUR ACCEPTANCE OF THE FOLLOWING TERMS AND CONDITIONS.**

SpeechSoft agrees to grant you a non-exclusive license to use the enclosed software program only on the terms specified herein.

**Important - Read Carefully:** this SpeechSoft CALLMaster license agreement (CLA) is a legal agreement between you (either an individual or a single entity) and SpeechSoft corporation for the SpeechSoft software product identified above, which includes computer software and may include associated media, printed materials, and online or electronic documentation (software product). The software product also includes any updates and supplements to the original software product provided to you by SpeechSoft. By installing, copying, downloading, accessing or otherwise using the software product, you agree to be bound by the terms of this CLA. If you do not agree to the terms of this CLA, do not install or use the software product; you may, however, return it to your place of purchase for a refund based on their refund policy.

## Permitted Uses

You may use this program on a single computer. That computer must be set-up with a software license file from SpeechSoft. The software license file is locked to one of the dialogic boards installed on the pc to be used by CALLMaster.

You may make a single back up of the program.

You may transfer the program and software license with the associated dialogic board to another party if the party agrees to the terms of this agreement and you destroy all your copies of the program. If you transfer the program or any copy to another party, your license and right to use the program is automatically terminated.

## Software Product License

**The software product is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The software product is licensed, not sold.**

### 1. Grant of license. This CLA grants you the following rights:

- Applications software. You may install, use, access, display, run, or otherwise interact with (run) one copy of the software product, or any prior version for the same operating system, on a single computer, workstation, terminal, or other digital electronic device (computer). The primary user of the computer on which the software product is installed may make a second copy for his or her exclusive use as backup.
- You must acquire and dedicate a license for each separate computer on which the software product is run. A license for the software product may not be shared or used concurrently on different computers.
- Reservation of rights. All rights not expressly granted are reserved by SpeechSoft.

### 2. Description of other rights and limitations.

- Limitations on reverse engineering, de-compilation, and disassembly. You may not reverse engineer, decompile, or disassemble the software product, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation.
- Separation of components. The software product is licensed as a single product. Its component parts may not be separated for use on more than one computer.
- Trademarks. This CLA does not grant you any rights in connection with any trademarks or service marks of SpeechSoft.
- Rental. You may not rent, lease, or lend the software product unless granted that ability in writing from SpeechSoft.
- Support services. SpeechSoft may provide you with support services related to the software product (support services). Use of support services is governed by the SpeechSoft policies and programs described in the user manual, in online documentation, and/or in other SpeechSoft-provided

materials. Any supplemental software code provided to you as part of the support services shall be considered part of the software product and subject to the terms and conditions of this CLA. With respect to technical information you provide to SpeechSoft as part of the support services, SpeechSoft may use such information for its business purposes, including for product support and development. SpeechSoft will not utilize such technical information in a form that personally identifies you.

- **Software transfer.** The initial licensee of the software product may make a one-time permanent transfer of this CLA and software product only directly to an end user. This transfer must include all of the software product (including all component parts, the media and printed materials, any upgrades, this CLA, the software license file and, if applicable, the certificate of authenticity). Such transfer may not be by way of consignment or any other indirect transfer. The transferee of such one-time transfer must agree
- To comply with the terms of this CLA, including the obligation not to further transfer this CLA and software product.
- **Termination.** Without prejudice to any other rights, SpeechSoft may terminate this CLA if you fail to comply with the terms and conditions of this CLA. In such event, you must destroy all copies of the software product and all of its component parts.

**3. Upgrades.** If the software product is labeled as an upgrade, you must be properly licensed to use a product identified by SpeechSoft as being eligible for the upgrade in order to use the software product. A software product labeled as an upgrade replaces and/or supplements (and may disable) the product that formed the basis for your eligibility for the upgrade. You may use the resulting upgraded product only in accordance with the terms of this CLA. If the software product is an upgrade of a component of a package of software programs that you licensed as a single product, the software product may be used and transferred only as part of that single product package and may not be separated for use on more than one computer.

**4. Copyright.** All title and copyrights in and to the software product (including but not limited to any images, photographs, animations, video, audio, music, text, and applets incorporated into the software product), the accompanying printed materials, and any copies of the software product are owned by SpeechSoft or its suppliers. All title and intellectual property rights in and to the content which may be accessed through use of the software product is the property of the respective content owner and may be protected by applicable copyright or other intellectual property laws and treaties. This CLA grants you no rights to use such content. If this software product contains documentation which is provided only in electronic form, you may print one copy of such electronic documentation.

**5. Dual-media Software.** You may receive the software product in more than one medium. Regardless of the type or size of medium you receive, you may use only one medium that is appropriate for your single computer. You may not run the other medium on another computer. You may not loan, rent, lease, or otherwise transfer the other medium to another user, except as part of the permanent transfer (as provided above) of the software product.

**6. Backup Copy.** After installation of one copy of the software product pursuant to this CLA, you may keep the original media on which the software product was provided by SpeechSoft solely for backup or archival purposes. If the original media is required to use the software product on the computer, you may make one copy of the software product solely for backup or archival purposes. Except as expressly provided in this CLA, you may not otherwise make copies of the software product or the printed material accompanying the software product.

## **Miscellaneous**

**This CLA is governed by the laws of the State of New York.**

Should you have any questions concerning this CLA, or if you desire to contact SpeechSoft for any reason, please contact the SpeechSoft sales

## **Limited Warranty for Software Products**

SpeechSoft warrants that (a) the software product will perform substantially in accordance with the accompanying written materials for a period of thirty (30) days from the date of receipt, and (b) any support services provided by SpeechSoft shall be substantially as described in applicable written materials provided to you by SpeechSoft, and SpeechSoft support engineers will make commercially reasonable efforts to solve

any problem issues. Some states and jurisdictions do not allow limitations on duration of an implied warranty, so the above limitation may not apply to you. To the extent allowed by applicable law, implied warranties on the software product, if any, are limited to thirty (30) days.

**Customer Remedies.** SpeechSoft's and its suppliers' entire liability and your exclusive remedy shall be, at SpeechSoft's option, either (a) return of the price paid minus restocking fee's, if any, or (b) repair or replacement of the software product that does not meet SpeechSoft's limited warranty and which is returned to SpeechSoft with a copy of your receipt. This limited warranty is void if failure of the software product has resulted from accident, abuse, or misapplication. Any replacement software product will be warranted for the remainder of the original warranty period or thirty (30) days, whichever is longer.

### **No Other Warranties**

To the maximum extent permitted by applicable law, SpeechSoft and its suppliers disCLAIM all other warranties and conditions, either express or implied, including, but not limited to, implied warranties of merchantability, fitness for a particular purpose, title, and non-infringement, with regard to the software product, and the provision of or failure to provide support services. This limited warranty gives you specific legal rights. You may have others, which vary from state/jurisdiction to state/jurisdiction.

### **Limitation of Liability**

To the maximum extent permitted by applicable law, in no event shall SpeechSoft or its suppliers be liable for any special, incidental, indirect, or consequential damages whatsoever (including, without limitation, damages for loss of business profits,

Business interruption, loss of business information, or any other pecuniary loss) arising out of the use of or inability to use the software product or the provision of or failure to provide support services, even if SpeechSoft has been advised of the possibility of such damages. In any case, SpeechSoft's entire liability under any provision of this CLA shall be limited to the greater of the amount actually paid by you for the software product or USD5.00; provided, however, if you have entered into a SpeechSoft support services agreement, SpeechSoft's entire liability regarding support services shall be governed by the terms of that agreement. Because some states and jurisdictions do not allow the exclusion or limitation of liability, the above limitation may not apply to you.

### **Whole Agreement**

This agreement constitutes the entire agreement between the two parties, and may be changed only by a written agreement of both parties. The invalidity or unenforceability of any part of this agreement shall not affect the validity or enforceability of any remaining parts, all of which shall remain in full force. This agreement shall be governed by the laws of the state of New York.

# INTRODUCTION TO CALLMASTER

[CALLMaster Overview](#)

[CALLMaster Help Documentation and Support](#)

## CALLMaster Overview

CALLMaster delivers the five most popular Computer Telephony applications in one easy to use software package for the Windows 2000, XP, 2003 SERVER platform. CALLMaster brings powerful call automation to every organization by providing full feature:

[Messaging](#)

[Attendant](#)

[Audiotex](#)

[Out Dial](#)

[Fax back](#)

[CALLBasic](#)

All the applications are totally integrated so you can answer a call in one application and transition into another then another seamlessly. CALLBasic, a powerful scripting language, provides the ability to manage other SQL databases and customize applications.

CALLMaster can utilize any third party SR - Speech Recognition engine that is Microsoft SAPI 5.x compliant to provide TTS (Text to Speech) and ASR (Automatic Speech Recognition) capabilities to callers.

1. With TTS, you can develop CALLMaster custom applications to speak text messages from a file. This feature is best suited to verify a caller's name, or say a product name.
2. CALLMaster incorporates ASR in two ways:
  1. First it allows caller to say the name of the person or extension they want instead of dialing and extension or spelling the name with telephone buttons.
  2. And second, CALLMaster features a new Recognize command in CALLBasic, the integrated scripting language, which will determine the phrase or word spoken from a designated list of words/phrases. This brings ASR functionality to IVR /Audiotex. Now any CALLMaster application like Order Entry can utilize Speech Recognition and Text to Speech to provide a more natural user interface with greater flexibility.

CALLMaster ASR feature is optional and must be purchased separately from the base product. When purchased, ASR functionality in CALLMaster is licensed by the License File. For details, see [Automatic Speech Recognition](#).

Scalable from 4 to 120 Phone Ports or 4 to 400 IP Ports, CALLMaster fits into every environment (telephone and IP network). It works with Analog and T1/PRI, Analog/Digital and IP PBXs, IP Gateways or Network and SMDI support.

For integration with your telephone network, CALLMaster requires the Dialogic hardware telephony boards for the PC platform. Scalable from 4 to 120 Phone Ports, CALLMaster works with Analog lines and T1/PRI trunks, Analog/Digital PBXs, and SMDI support.

For integration with your IP environment, CALLMaster uses the Intel Dialogic HMP software. HMP is a complete software solution that provides media processing and IP call Control. This provides the support for Voice Over IP call processing in both the SIP and H323 protocol environments and for Fax Over IP. Scalable from 4 to 120 Phone Ports or 4 to 400 IP Ports, CALLMaster integrates with IP PBXs, IP Gateways or Network.

Support for Unified Messaging and Fax In/Out provides a complete information processing solution.

CALLMaster is totally integrated with MS SQL. This allows you to expand upon the base design and build sophisticated telephony applications.

System Requirements: CALLMaster requires Windows 2000, XP Pro or 2003 Server, one or more Dialogic Telephony Communication boards. CALLMaster requires a minimum of 2 GB of Disk storage and 1GB RAM (2GB recommended).



This help covers functionality thru CALLMaster 4.0 and CALLMaster IP 5.2.4.0.

## **CALLMaster Overview - Messaging**

The Message Management capabilities of CALLMaster includes managing Voice Mail, setting up Notification records, and creating Distribution Lists with the associated Distribution List Mailboxes.

Voice Mail manages digitally recorded messages. It records, stores, retrieves, reviews, forwards messages, notifies you when new messages arrive, allows caller to be transferred to a designated phone number, receives in faxes to default or selected mailbox, and sends emails to the address(es) in the mailbox Email field when message or fax received.

To set-up your system for Messaging, you need to:

- Configure the desired Phone Ports to Voice Mail;
- Set-up the required Mailboxes;
- Turn on Notification / Message Waiting Lights, as required;
- Make private Distribution Lists if desired;
- Enable Email Notification feature (optional).

With CALLMaster you can provide multi-lingual voice mail support using the language set defined by your system administrator.

For details see [Multi-Lingual Support](#).

Now you are ready to have your users log in to their mailboxes and record their name and greeting.

[\*\*Learn more about Message Management\*\*](#)

[\*\*Continue with the CALLMaster Overview\*\*](#)

[\*\*Back\*\*](#)

## **CALLMaster Overview - Attendant**

The CALLMaster Attendant provides the capability to transfer a call from one line to another line or extension, or to connect 2 lines, routing callers to extensions or departments, voice mailboxes, alternate messages or Audiotex modules. The Attendant can process calls based on the time of day with appropriate greetings, and can intelligently perform various Failed Transfer Action depending on the results of the call (e.g. Busy, No Answer).

Calls can be transferred using Centrex, 3-way-calling, Analog Flash-hook, T1-Wink, PRI-2BCT, Digital PBXs or CALLMaster Channel Bridging functionality.

Using Attendant setup, inbound faxes can be received into the default or selected mailbox.

To set-up your system for Attendant, you need to:

- Make sure there is an Audiotex module with Action set to Attendant or use Audiotex Module ID 3 created by default with Module Action set to Attendant.
- Make sure the Attendant is the Applications set-up with the corresponding Audiotex Module ID.
- Configure the desired Phone Ports to the Attendant application;
- Set-up the relevant Attendant Records, using the time-of-day option and the Failed Transfer Action;
- Create an Attendant Directory record if desired;
- Verify Attendant parameter settings for phone system integration (e.g. Flash Hook, Msg Light codes).

If you have activated the Directory record, you will need to make the Attendant Directory entries. See how to [Set-up Attendant Directory](#).

With CALLMaster you can provide multi-lingual attendant support using the language set defined by your system administrator. For details see [Multi-Lingual Support](#).

You are now ready for calls to be answered and transferred to the appropriate extension or number.

[Learn more about Attendant](#)

[Continue with the CALLMaster Overview](#)

[Back](#)

## **CALLMaster Overview - Audiotex**

CALLMaster Audiotex provides powerful capabilities for sophisticated call processing and information delivery systems. Using Audiotex modules, you can define the sequence of messages and responses that define an automated conversation.

Callers can interact with CALLMaster to retrieve specific information/messages/faxes, to access other databases for delivery of custom data (e.g. customer's balances), or to record messages, and responses to questions, to be notified of specific information or events (e.g. appointment reminder). CALLMaster permits sophisticated relationships between Voice Mail, Attendant, Out Dial, and Fax applications.

Audiotex Modules are used as basic building blocks to create IVR applications. The conversation flow can be predetermined or be based on the caller's responses. Each module can execute a 'program script' (a set of instructions written in CALLBasic) to accomplish a desired function (e.g. access a customer record based on a Caller input key). By connecting modules using the Next Module ID, you can create various custom IVR applications.

CALLMaster can run several IVR applications concurrently. Each application needs to be assigned a name and be associated with a starting Audiotex Module ID in the IVR Application List.

To set-up your system for an IVR application, you need to:

- Create the CALLBasic scripts, if required and set-up the relevant Audiotex modules;
- Define the IVR application in the IVR Application List with the starting Audiotex module number;
- Set the desired Phone Ports to the IVR application;
- Set-up any other databases required by your application (e.g. Attendant, Mailboxes).

You are now ready to record your Audiotex messages. Temporarily set-up one of your phone ports to Voice Mail, log in as the Administrator and record the Audiotex messages. Remember to reset your phone port to your IVR application once done with the recording.

With CALLMaster you can provide multi-lingual IVR applications using the language set defined by your system administrator.

For details see [Multi-Lingual Support](#).

You are now ready activate your Audiotex application.

[\*\*Learn more about Audiotex\*\*](#)

[\*\*Continue with the CALLMaster Overview\*\*](#)

[\*\*Back\*\*](#)

## CALLMaster Overview - Out Dial

CALLMaster Out Dial application allows you to make telephone calls to a list of phone numbers that have been loaded into the system, and deliver and/or collect information or send a fax. The information delivered can be a simple message or a complex scripted conversation in Audiotex. Each phone number can be directed to a specified port or to the next available port. Each port can concurrently support a different Out Dialing program if desired via the Audiotex Start Module. CALLMaster supports Scheduled Dialing from any available port that has the Port Type enabled for Outbound.

To set up Out Dialing, make sure that the CALLMaster service is stopped. Then,

1. Customize the relevant Audiotex modules and make the desired recordings. Alternatively you can download the Out Dialing Sample Application from the [SpeechSoft Downloads](#) webpage.
2. Set the Ports to the desired Application Type when system is not Out Dialing.
3. Select the phone list template from the CALLMaster/Outdial List folder that you want to use to import to CALLMaster and populate with your phone numbers.
4. Use the CALLMaster Manager to load the phone numbers and manage numbers to be called.
5. Start CALLMaster service. Calls will be attempted on any available port regardless of application type.

You are now ready to call the phone numbers on your list.

[Learn more about Out Dial](#)

[Continue with the CALLMaster Overview](#)

[Back](#)

## CALLMaster Overview - Fax

CALLMaster supports both inbound and outbound faxing. Fax functionality requires the Dialogic Fax Boards. The Global Tone table must have entry to recognize a fax tone type.

**Inbound faxes** can be received automatically to a mailbox (default or selected by caller) via Voicemail, Attendant or Audiotex or to a specified file name via Audiotex. Faxes received into a mailbox can be forwarded as a .tif email attachment if the mailbox is set-up for email notification. During Log In, the mailbox owner can review new Faxes (header details), redirect new faxes to fax machines and save and delete faxes.

Configure the desired Phone Ports to Voicemail, Attendant or Audiotex (with appropriate Audiotex Start module);

- If Audiotex, customize the relevant Audiotex modules
- Modify system greetings to instruct caller on faxing documents
- Set-up email notification if desired

**Outbound faxes** can be sent using custom Audiotex/CALLBasic applications or via Out Dial functionality. Documents can be faxed directly to the fax machine from where the call was made supporting one-call faxing, or ask the caller to input the fax number to be called back at, supporting two-call faxing.

Configure the desired Phone Ports to Audiotex with the appropriate Audiotex Start module;

- Customize the relevant Audiotex modules;
- Create and store the fax documents;
- Set-up the Fax database.
- Update the OutcallSchedule table with the required information.

[Learn more about Fax](#)

[Continue with the CALLMaster Overview](#)

[Back](#)

## **CALLMaster Overview - CALLBasic**

CALLBasic is a scripting language that lets you write a program that is to be run every time the conversation reaches a specific Audiotex module. It gives you the ability to manage custom defined databases in addition to the CALLMaster database, execute other programs, perform computations, store information to local and global variables and control the conversation flow.

Use CALLBasic scripting with Audiotex to customize IVR applications requiring sophisticated database interaction.

[Learn more about CALLBasic](#)

[Back](#)

## **CALLMaster Documentation and Support**

There are various sources of Documentation and Support available for CALLMaster:

[CALLMaster Online Documentation](#)

[CALLMaster Website Support - \[www.speechsoft.com\]\(http://www.speechsoft.com\)](#)

[CALLMaster Installation Guide](#)

[CALLMaster Technical Support](#)



## CALLMaster Online Documentation

CALLMaster installs complete online technical reference documentation and context sensitive help when you install it. To access the online documentation select - **Start / Programs / CALLMaster / CALLMaster Documentation**.

The CALLMaster Manager has context sensitive help, which allows you to get specific help information depending on what window is open. Simply click on the arrow/question mark button in the upper right corner of the window to turn the cursor into an arrow with a question mark next to it. Then click on any section of the screen you want help with to open the Documentation window for that screen. You can then click on the desired field for details.

Using the [Online Documentation](#), the CALLMaster Tour section is a great place to start. If you are new to SpeechSoft products this will give you a comprehensive review of the product, its capabilities and its usage.

Current Help covers functionality thru CALLMaster 4.0 and CALLMaster IP 5.2.4.0.

**CALLMaster Website Support - [SpeechSoft.com](http://SpeechSoft.com).**

For additional information and to get the latest documentation check the SpeechSoft Web site [SpeechSoft.com](http://SpeechSoft.com).

## **CALLMaster Installation Guide**

This document provides a quick reference for CALLMaster Installation.

You can download the [Installation Guide](#) from our website.

## CALLMaster Technical Support

First check the [Online Documentation](#) for technical reference. Take the CALLMaster Tour and review the Walkthroughs.

Browse the Frequently Asked Questions (FAQ) in either source and benefit from the experience of other members by finding your answer in this list.

You can reach SpeechSoft Technical Support

- by phone (914-273-5560 x3);
- by e-mail ([support@speechsoft.com](mailto:support@speechsoft.com)) or
- by completing and submitting [Technical Support Form](#).

# Operating CALLMaster - Overview

## Overview

[Installation and Setup](#)

[Configuring CALLMaster](#)

[Managing CALLMaster](#)

[Running CALLMaster](#)

[Logging and Tracking Activity](#)

[Maintenance](#)

[Backups](#)

## CALLMaster Installation and Setup

There are 3 easy steps to [Install CALLMaster](#).

Once installed, you can [Set-up CALLMaster](#) by selecting your database type and configuring your telephony environment.

After set-up, you should [Validate Installation](#) before configuring CALLMaster.

If you have any problems with the installation, you may want to [Uninstall CALLMaster](#) and retry the installation again.

**NOTE: As both CALLMaster and Dialogic runs as a service, it is very important to install both using the Administrator account from a local console.**

If you are using remote terminal services, you must access the console session 0 to install. To do so, at a command prompt type `c:\>"mstsc /console"` to connect to the console session.

If you continue to have problems, see [SpeechSoft Technical Support](#)

## **CALLMaster Installation and Setup - Installing CALLMaster**

### **Download Required Software.**

Your computer must be running Windows XP/Vista/Win 7 or 2003/2008 server. Additionally, you should have downloaded the required software from [SpeechSoft Downloads](#).

This should include:

1. Prerequisite Software (required for Windows XP only)
  - Microsoft .Net Framework 3.5 SP1
2. Dialogic Software (Board Drivers or HMP Drivers)
3. CALLMaster Software which includes SQL Express 2008.

The CALLMaster installation is divided in to three (3) steps.

[Step 1: CALLMaster Prerequisite Software Installation](#)

[Step 2: Dialogic Software](#)

[Step 3: CALLMaster Software Installation](#)

NOTE: As both CALLMaster and Dialogic runs as a service, it is very important to install both using the Administrator account from a local console.

If you are using remote terminal services, you must access the console session 0 to install. To do so, at a command prompt type `c:\>"mstsc /console"` to connect to the console session.

If you are integrating with a telephone network, download and review the [CALLMaster Installation Guide](#) (pdf) or [CALLMaster Setup Instructions](#) (rtf).

For IP network integration, see [CALLMaster IP Installation Guide](#) (pdf) or [CALLMaster IP Setup Instructions](#) (pdf).

## **Installing CALLMaster - Step 1: CALLMaster Prerequisite Software Installation**

### **Download and Install Prerequisite Software. (Required for Windows XP Only)**

Your computer must be running Windows XP/Vista/Win 7 or 2003/2008 server.

#### **1.1. Verify the prerequisite programs' existence on the computer**

1. Confirmed by opening the Control Panel's "Add/Remove Programs" or "Uninstall Programs" Window , which lists all the currently installed software. Check that the following entry is listed:
  1. "Microsoft .Net Framework 3.5 SP1"
2. If this entry is not listed you will need to continue to Step 1.2 to download the install program and install the prerequisite before continuing to Step 2.  
If this entry is listed then skip to [Step 2: Dialogic Drivers & Board\(s\) Installation and Configuration](#).

#### **1.2. Download the following executable and save in My Documents folder.**

[dotnetFx35Setup.exe](#) - to install "Microsoft .Net Framework 3.5 SP1"

#### **1.3. Installing Prerequisite Software**

1. Run dotnetFx35Setup.exe by double clicking it. This program installs the .Net3.5 Framework files. This installation takes about 10-20 minutes to download files from Microsoft servers and then about 5 minutes to install.
2. When done the installed prerequisite can be confirmed by viewing the "Add/Remove Programs" or "Uninstall Programs" Window in the control panel.  
Check that the following entry is listed:
  1. "Microsoft .Net Framework 3.5 SP1"
3. If the "Add/Remove Programs" or "Uninstall Programs" Window was open already, select it and press 'F5' or close and open the window to refresh the list.  
Once the above prerequisite is confirmed as installed you can proceed with Step 2 of the installation process.

Proceed to [Step 2: Dialogic Drivers & Board\(s\) Installation and Configuration](#).



## Installing CALLMaster - Step 2: Dialogic Software

Depending on your network integration, you will need to either download and install Dialogic Board Drivers or HMP Drivers.

**For integration with your telephone network**, CALLMaster requires the Dialogic hardware telephony boards.

The Dialogic driver software must be installed on your system before installing CALLMaster in order for CALLMaster to install.

See [Step 2: Dialogic Drivers & Board\(s\) Installation and Configuration](#).

**For integration with your IP network**, CALLMaster requires HMP software.

See [Step 2: Dialogic HMP Software Installation and Configuration](#).

Once the Dialogic Service shows System Status of Running, you can proceed to [Step 3: CALLMaster Software Installation](#).

### Step 3: CALLMaster Software Installation

You can install CALLMaster from our website or use the CD-Rom provided by SpeechSoft.

**Make sure you have the Prerequisites and the Dialogic System Software installed before proceeding to install CALLMaster software.**

**You must be logged in as the local Administrator.**

If you are using remote terminal services, you must access the console session 0 to install. To do so, at a command prompt type `c:\>"mstsc /console"` to connect to the console session.

#### 3.1. If installing from the website

You must [download the CALLMaster software](#) from our website to install CALLMaster. This install includes 2 additional software components:- Microsoft SQL Express and Visual C++ Runtime Libraries.

1. Create a new folder in My Documents – e.g. CMSetup.
2. Save the downloaded file in the new CMSetup folder.
3. Extract the files from the zip file to the folder CMSetup.
4. Check that the CMSetup folder contains the following folders and files. If not then repeat the Step 3.1.3.
  - i. 2 Folders:- \SqlExpress\; \WindowsInstaller4\_5\
  - ii. 6 Files:- autorun.ini; CMInstall.msi; CMSetup.exe; Setup.exe; InstallGuide.pdf; SetupInstructions.rtf
5. Open the CALLMaster Setup folder created “CMSetup and run CMSetup.exe by double clicking it.
6. Proceed with the installation as detailed in Step 3.3.3 below.

#### 3.2. If installing from the CD-Rom

1. Insert the CD-Rom to initiate the CALLMaster installation.
2. Proceed with the installation as detailed in Step 3.3.3 below.

#### 3.3. Install CALLMaster Software

NOTE: The Dialogic software drivers must be installed before CALLMaster can operate.

If you have not done so yet this is a good time to install the Dialogic drivers (refer to Step 2).

1. You are then given a message box with a notice that:
  1. You must be logged in as the local Administrator and
  2. The “DotNet3.5SP1 Framework software” is installed as per Step #1. “Preparing your computer for CALLMaster installation.” Press ‘OK’ to continue.
2. The “Windows Installer 4.5” license will be displayed if it needs to be installed. Review and click ‘Accept’ to continue.
3. If it is not already installed on the computer, you will get the “SQL Express License Agreement”, review and click ‘Accept’ to continue.
4. The “Windows Installer 4.5” will install first. Once installed, the set-up will require a reboot of the system before proceeding. Close and save all files on the PC then choose ‘Yes’ to reboot.
5. After the system has restarted, the CALLMaster installation will automatically continue with the “SQL Server 2008 Express Edition” installation.  
The SQL Server installation will take 10-15 minutes.
6. When the SQL Server installation is complete the CALLMaster Setup Wizard will start.  
The CALLMaster License Agreement is presented, review and select “I Agree” to confirm agreement with the terms of the CALLMaster License Agreements, then press ‘Next’ to continue.
7. In the next window “Select Installation Folder”, make sure that install CALLMaster for “Everyone” is selected.  
By default “C:\Program Files\CALLMaster\” folder is selected and recommended for installation, then press ‘Next’ to start the installation.

When the “Installation Complete” window is presented, click ‘Close’ to exit.

CALLMaster is configured at installation time to run automatically as a service every time the computer is restarted. You can control this setting through the Services window on the Windows XP, Win 7, Vista or 2003/2008 Server Control Panel.

During installation, CALLMaster will install SQL Server 2008 Express with the CALLMaster database files (CALLMasterSQL.mdf and CALLMasterSQL\_Log.ldf) attached. These files are located by default in the C:\Program Files\CALLMaster\Data folder. See [Advanced Set-up for CALLMaster Database](#) for details.

The CALLMaster Manager provides a front end to this database.

Be sure to read the release notes at installation time. They will make you aware if the latest issues with this release and alert of any undocumented features and known bugs. Also check the web site [www.speechsoft.com](http://www.speechsoft.com) for the latest release information.

### **CALLMaster License File**

Install your CALLMaster License File in your CALLMaster\Data folder. CALLMaster versions older than 3.5 may still use the parallel printer port or USB port license key. If you do not have a license key or file, CALLMaster will run in demo mode.

### **IP Network Configuration and Settings**

For CALLMaster to work in a IP environment the configuration file CMVOIP.cfg must be setup to match the VOIP Network setting. See [IP Configuration](#) for details

**If you run into any problems, you can [Uninstall](#) and [Reinstall](#) CALLMaster.**

**NOTE:** Running Anti-Virus software on the CALLMaster computer could stop CALLMaster. See [Anti-Virus Conflict](#) for details.

## Installing CALLMaster - Initial Setup

### Database Support

CALLMaster is shipped with an MS SQL 2008 database. The database file, CALLMasterSQL.mdf, is located in the CALLMaster\Data folder. During installation, CALLMaster will install SQL Server 2008 Express. The CALLMaster Manager provides a front end to this database.

If you wish to continue using SQL Server 2008 Express, then you do not have to make any changes.

If you are running MS SQL Server 2008 and wish to use this server instead of the SQL Server 2008 Express, see [Database Set-up](#)

### Network Integration

If you are integrating CALLMaster with your telephone network, [Connect Phone Lines](#).  
For integration with your IP network, see [IP Configuration](#).

## **CALLMaster Database Installation Setup**

CALLMaster is shipped with an MS SQL 2008 database. The database file, CALLMasterSQL.mdf, is located in the CALLMaster\Data folder.

During installation, CALLMaster will install SQL Server 2008 Express on the local computer and will connect the database (CALLmasterSQL.mdf) to this server.

This connection string and provider name is stored in the Windows registry.

The database connection string at installation is stored with variable DBName as follows:

"Data Source=localhost\SQLEXPRESS;Initial Catalog=CALLMasterSQL;Integrated Security=SSPI;Connect Timeout=5"

The database provider at installation is stored with variable DBProvider as follows:

"Provider=SQLNCLI10;MARS Connection=True;"

## **Changing the CALLMaster Database Connection**

If you wish to use the installation setup for the CALLMaster database, then you do not have to make any changes.

If you are running another SQL Server and wish to use this server instead of the SQL Server 2008 Express, you will need to modify the connection string and provider in the Windows registry. For details see [Appendix K - Changing the CALLMaster Database Connection and Provider](#).

## **Separate Database for the OutcallSchedule tables**

If you are using the Out Dial function, the OutcallSchedule and OutcallScheduleCompleted tables can be in a database separate from the CALLMaster database. CALLMaster allows you to set-up a connection string to point to any database desired. See [Out Dial](#) for details on how to set-up these connection strings.

You are now ready to [Start CALLMaster](#).

## **Installing CALLMaster - Initial Setup - Connect Phone Lines**

If you are connecting your phone lines directly from the phone company into CALLMaster, then just plug the lines directly into the Dialogic board installed in the CALLMaster machine.

If you are using a PBX, then connect the selected extensions off your PBX into the Dialogic board installed in the CALLMaster machine. You may need to run [PBXpert](#)

If you have T1 lines, please read [T1 Setup Guide](#).

# VoIP Configuration

## VOIP – Voice Over IP

Voice over IP (VoIP) can be described as the ability to make telephone calls and send faxes over IP based data networks with a suitable Quality of Service (QoS). The voice information is sent in digital form using discrete packets rather than via dedicated connections as in the circuit-switched Public Switched Telephone Network (PSTN). There are two major international groups defining standards for VoIP:

- International Telecommunications Union, Telecommunications Standardization Sector (ITU-T), which has defined the following:
  - Recommendation H.323, covering Packet-based Multimedia Communications Systems (including VoIP)
- Internet Engineering Task Force (IETF), which has defined drafts of the several RFC (Request for Comment) documents, including the following central document:
  - RFC 3261, the Session Initiation Protocol (SIP)

The H.323 recommendation was developed in the mid 1990s and is a mature protocol.

SIP (Session Initiation Protocol) is an emerging protocol for setting up telephony, conferencing, multimedia, and other types of communication sessions on the Internet.

The Intel Dialogic HMP software provides supports both H.323 and SIP as well as FOIP – Fax Over IP.

In addition to answering inbound calls and making outbound calls these calls CALLMaster can transfer the calls to any other number either in a supervised or blind transfer fashion.

CALLMaster can register with your VOIP Gateway/GateKeeper/Registrar to be fully integrated into your VOIP environment. It also works with VOIP service providers in a remote or locally hosted setup. Providing the most flexibility in any environment.

## IP Network Configuration and Settings

CALLMaster supports both the H.323 and SIP VOIP protocols and various audio coders and DTMF formats. By default CALLMaster will run in stand alone mode and listen on port 5060 for incoming VOIP calls. Any VOIP device can call CALLMaster by dialing the IP address of the machine running the CALLMaster software. CALLMaster can also register with a Gateway or Registrar.

For CALLMaster to work in a VOIP environment the configuration file CMVOIP.cfg must be setup to match the VOIP environment. The CMVOIP.cfg file located in the CALLMaster\Data folder. This file allows the following setting to be made:

IP Protocol : The IP Protocol used for opening the IP ports, values: H323, SIP, or both

DTMFmode: Touch Tone signaling mode, possible options: OutOfBand, inband, or rfc2833

Audio format: Formats supported g711Alaw, g711Mulaw, gsm, gsmEFR, g723\_5\_3k, g723\_6\_3k, g729a, g729ab, t38UDPFax

If CALLMaster is to register with a Gateway, GateKeeper or Registrar then set the following values. If it is not registering then set the GWIP setting to 0.0.0.0.

Gateway or Registrar parameters:

GWAcctName - <username> GateWay account username for login to VOIP Gateway server

GWAcctPW - <password> GateWay account password for login to VOIP Gateway server

GWAcctRealm - <realm> of VOIP Gateway (optional)

GWIP - <IP address> Dot notation of VOIP Gateway server with optional port number eg

206.80.70.54:5060

LocalAlias - <IP address> Dot notation of CALLMaster

To learn more, see [CALLMaster VoIP Configuration](#).



## Validate CALLMaster Installation

After rebooting your computer you are ready to run CALLMaster.

First make sure that:

1. The Dialogic Software (Board Drivers or HMP Drivers) is installed and configured. You can check this by running Windows 2000, XP or 2003 SERVER Task Manager (ctrl-alt-del) and seeing if **dlgc\_srv.exe** is an available Service. If it is not then see [Dialogic Installation](#).
  1. If integrating with the telephone network, then make sure
    1. the Dialogic board is installed. If it is not then see [Dialogic Installation](#).
    2. the Phone lines are connected to the phone ports on the boards. The lines should have been tested using a regular analog telephone hand set to confirm all required features work on each line. See how to [Connect Phone Lines and Test CALLMaster](#)
  2. If integrating with the IP network, then make sure
    1. the HMP license file is in the Dialogic\Data folder. If the license file is not in the required folder, Dialogic will assume it is being run in Demo Mode (1 port license).
    2. the VoIP settings are configured and the CMVOIP.cfg file placed in the CALLMaster\Data folder. see [IP Configuration](#).
2. CALLMaster software has been installed. If installation has completed correctly, you should have two (2) icons on your desktop:

1. [CALLMaster Main Window](#)



2. [CALLMaster Manager](#)



3. The CALLMaster software license file is in the CALLMaster\Data folder. If the license file is not in the required folder, CALLMaster will assume it is being run in Demo Mode. If CALLMaster has started incorrectly in Demo Mode, stop CALLMaster via the Services panel as described below, place the license file in the CALLMaster\Data folder and Start CALLMaster again.

Second, review [Advance Set-up for CALLMaster Database](#) to determine if you need to make any changes.

Both the Manager and the Main Window can be initiated from the Start / Programs / CALLMaster group.

You are now ready to [Start CALLMaster](#).

Call the port used by CALLMaster.

As CALLMaster default setup is voicemail, you should hear the default time of day greeting 'Good morning, you've reached CALLMaster voicemail'.

- If you hear this message, your setup is successful and you can proceed to [Configuring CALLMaster](#).
- If you do not hear the default greeting, then CALLMaster has not been installed correctly. You may want to [Uninstall](#) and [Reinstall](#) CALLMaster and try the process again.

If you continue to have problems, call SpeechSoft Technical Support at 914-273-5560 x3. or email [support@speechsoft.com](mailto:support@speechsoft.com)

## Uninstall CALLMaster

Follow the steps below to uninstall CALLMaster Software, Dialogic Drivers and the Prerequisite Software.

### To Uninstall CALLMaster

1. Before you uninstall CALLMaster confirm that the CALLMaster Service is stopped. You can see the status of the CALLMaster Service by opening the Control Panel/Administrative Tools/Services window. If CALLMaster service is running then the Status column will show "Started" otherwise the Column will be blank if the Service is stopped. If the status is "Started" then select the CALLMaster entry and right Click then choose "Stop" or click the square "Stop" button on the toolbar to stop the service. You may need to Refresh or close and open the Services window to get the latest status.
2. To uninstall the CALLMaster software, open the "Add/Remove Programs" or "Uninstall Programs" window in the Control Panel. Select the CALLMaster entry and it will expand and display 'Change' and 'Remove' buttons. Click Remove and follow the uninstall wizard to completely remove the CALLMaster software.
3. If no files were created (i.e. log files or audio recordings) since the program was installed then the un-installation will remove all the folders it created \Program Files\CALLMaster\ and all its sub folders. If any files were created then those files and the folders that contain them will not be removed. Check and manually delete the main CALLMaster folder if it exists.

### To Uninstall Dialogic System Software

1. Before uninstalling the Dialogic System Software confirm that the "Dialogic System Service" is not running. You can see the status of the Dialogic System Service by opening the Control Panel/Administrative Tools/Services window. If Dialogic System Service is running then the Status column will show "Started" otherwise the Column will be blank if the Service is stopped. If the status is "Started" then select the Dialogic System Service entry and right click then choose "Stop" or click the square "Stop" button on the toolbar to stop the service. You may need to Refresh or close and open the Services window to get the latest status.
2. In the "Add/Remove Programs" or "Uninstall Programs" window select "Dialogic System Release 6.0 PCI Redistributable Edition" and click 'Remove'. If the service is not stopped, a pop up message will tell you to stop it before removing. See the above paragraph for details.
  1. You have the option of saving your current Dialogic configuration. You may choose to save these settings if you plan to reinstall or upgrade the Dialogic Software otherwise click 'No'. Then click 'OK' to remove the Dialogic software.
3. When complete you will be prompted to reboot, choose 'Yes' and reboot the computer.

### To Uninstall CALLMaster Prerequisites

After uninstalling CALLMaster you can uninstall the additional components and prerequisites if desired or you may leave them on your system if used by other programs or plan to reinstall CALLMaster. Uninstall the components in the order listed below.

1. **Uninstall SQL Server.**  
In the "Add/Remove Programs" or "Uninstall Programs" window select Microsoft SQL Server 2008 and click 'Remove'. In the 'Components Selection' window, select 'Remove SQL Server 2008 instance components', 'SQLEXPRESS Database Engine' and 'Workstation Components', then click 'Next' then 'Finish' to uninstall SQP Server Express. A "Setup Progress" window will display the removal progress and display the status of the components removed. When done they should all have a green check mark and status: Setup Finished.  
Then press 'F5' or close and reopen the "Add/Remove Programs" or "Uninstall Programs" window to refresh the displayed list of installed programs. Only the "Microsoft SQL Server Native Client" should still be listed. Select that entry and click 'Remove' to remove that component from the computer.
2. **Uninstall " Microsoft .NET Framework 3.5 SP1, 3.0 and 2.0"**  
In the "Add/Remove Programs" or "Uninstall Programs" window select "Microsoft .NET Framework 3.5 SP1" and click 'Remove' and then "uninstall" to uninstall that component. .NET Framework 3.0 and 2.0 can both be uninstalled in the same manner.

The computer now is back to the state it was before you started installing CALLMaster. If you left a Dialogic board in the machine when it reboots it will display the "Found New hardware" wizard which you can cancel and run at another time or after you reinstall the Dialogic software or remove the board to avoid this message.

## Configuring CALLMaster

[CALLMaster Manager Screen Layout](#)

[Add Records](#)

[Delete Records](#)

[Edit Records](#)

## Configuring CALLMaster - CALLMaster Manager Screen Layout

Once you make your selection, the corresponding table is loaded into the main window area.

Each screen is comprised of multiple sections as shown above.

- Header - displays key information of the selected record.
- Table Title - name of the table being accessed.
- Grid View - list of records from the table. Read only. Highlight desired record to see details in the Edit View.
- Edit View - details of record highlighted on the Grid View. Changes can be made to these fields and saved by clicking on the Save button or by selecting any other row on the grid. If you don't want to save the changes you've made, select Undo Changes button or close the window you are in and answer 'no' to the 'record(s) have changed 'Save changes?' prompt.
- Navigation and Update Tools - tools to add, delete, save records and to navigate thru the list.

For example, if you selected Mailboxes, the Mailbox table window would be displayed (see screen below).

The screenshot displays the CALLMaster Manager interface for the [Mailboxes] table. The window title is "CALLMaster Manager - [Mailboxes]". The main area is divided into several sections:

- Header:** Displays "Mailbox 1001" and "Anne DeSoto".
- Table Title:** "Mailboxes".
- Grid View:** A table listing mailboxes. The selected record is highlighted in blue.
- Edit View:** Details for mailbox 1001, including fields for Mailbox #, Last Name, First Name, Email, Settings, and Type. An "Edit View" button is present.
- Message Storage:** Fields for Allowance Max, Current Count, New, Saved, Length, and Time.
- Notification:** Fields for Enabled, Notify per Msg, Notify Delay, Message Lights, Current Event ID, and Notification Record ID.
- References:** Fields for Audio Recordings, Greeting File ID, Name File ID, Directory Listing, and Use.
- Navigation and Update Tools:** A bar at the bottom containing buttons for Add, Save, Delete, Previous, Next, Undo Changes, and Exit.

Mailbox	Last Name	First Name
0000	Patton	John
1000	Smith	Jason
1001	DeSoto	Anne
1002	Hollander	Trudi
1003	Stone	Frank
1004	Newman	Jonathan
1005	Belize	Joan

## Configuring CALLMaster - Add Records

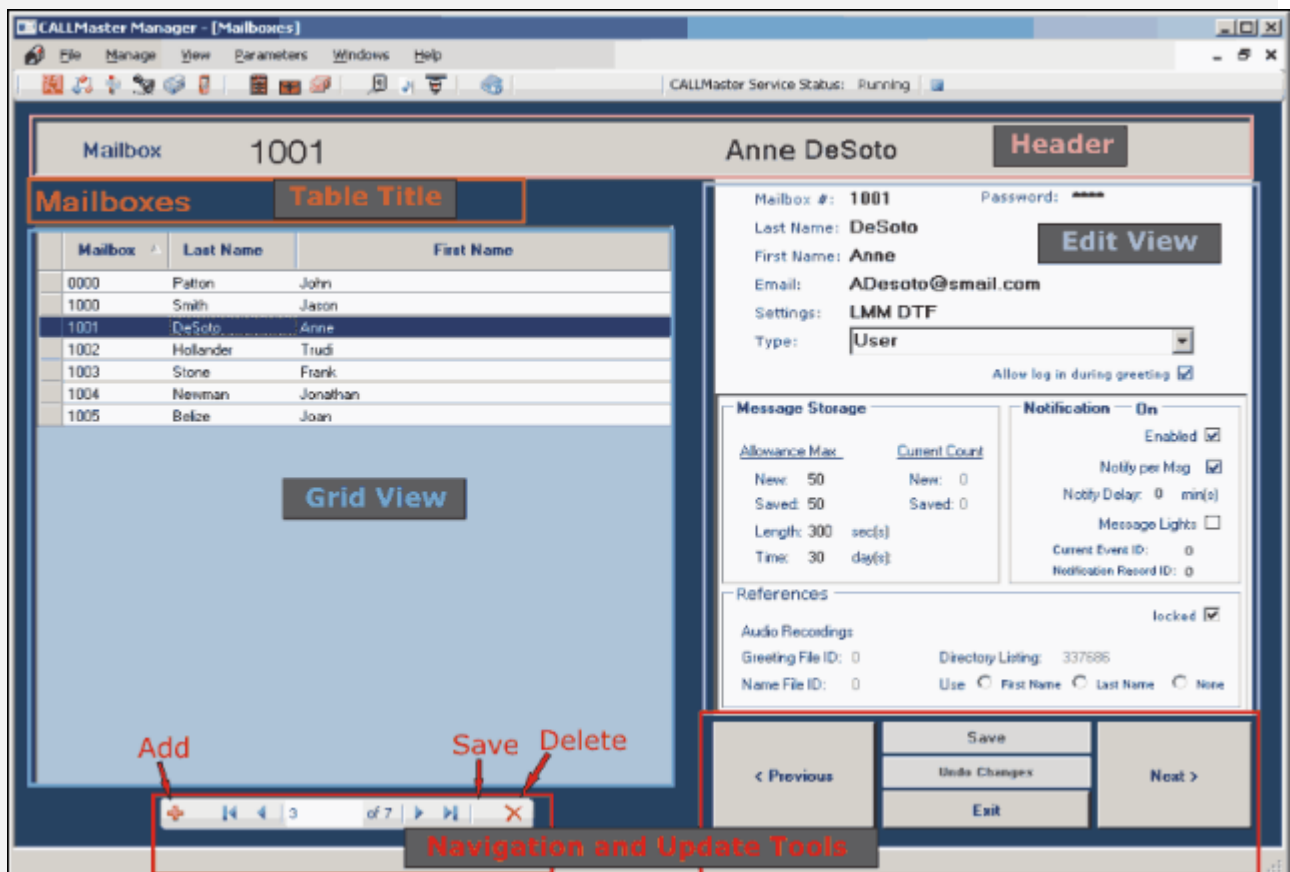
Open CALLMaster Manager and select the desired table.

Select the 'Add' icon as shown below. This will add a new row at the end of the Grid View section with field details on the right in the Edit View section.

**All information must be added in to the fields in the Edit View section of the screen.**

Click on any other section of the screen for the changes to take effect or click on the Save button.

If you do not want to add the record, click on the 'Delete' icon.

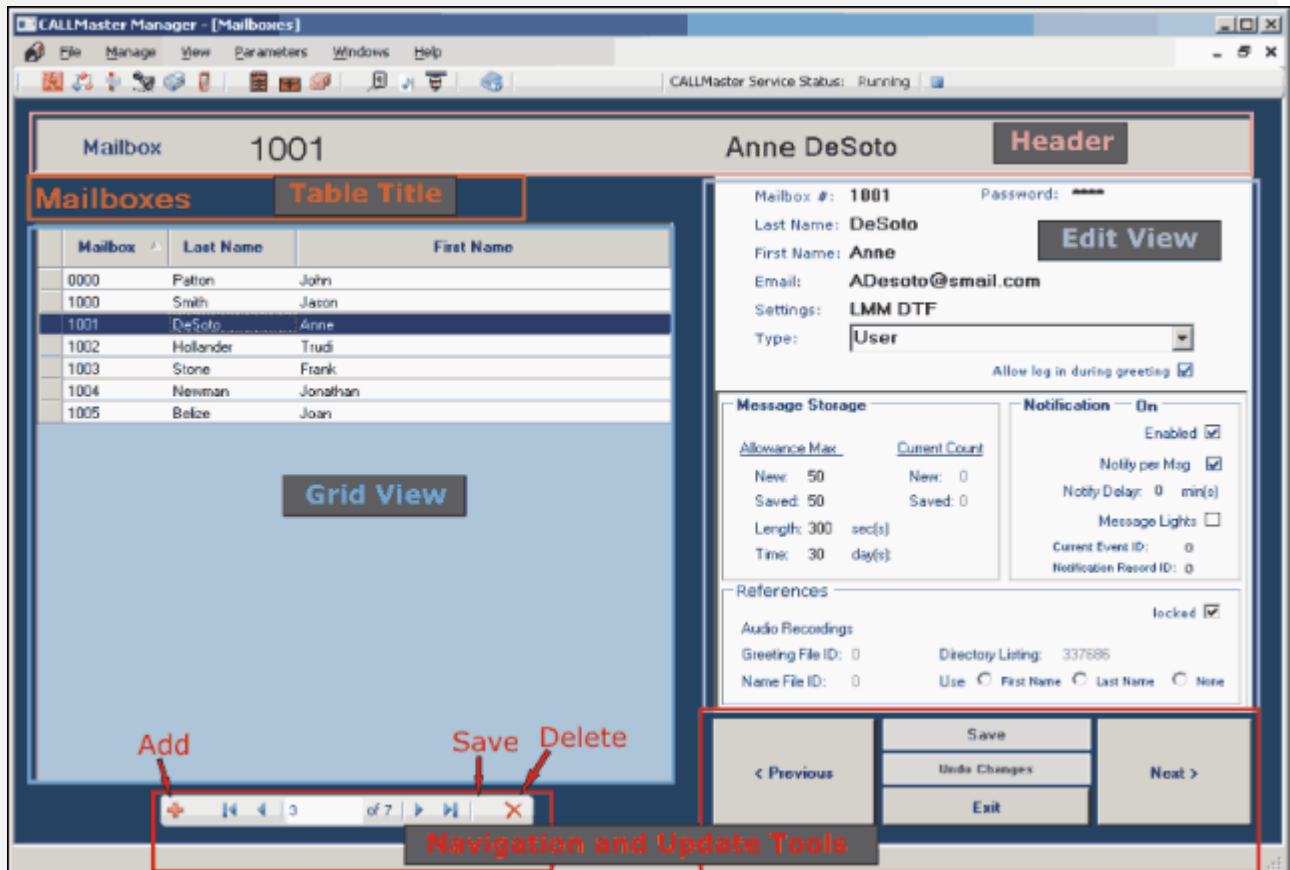


## Configuring CALLMaster - Delete Records

Open CALLMaster Manager and select the desired table.

On the Grid View section of the screen, select/highlight the record to see details. The detail fields for the selected record will be displayed on the Edit View section of the screen.

Click on the Delete button to delete the record.



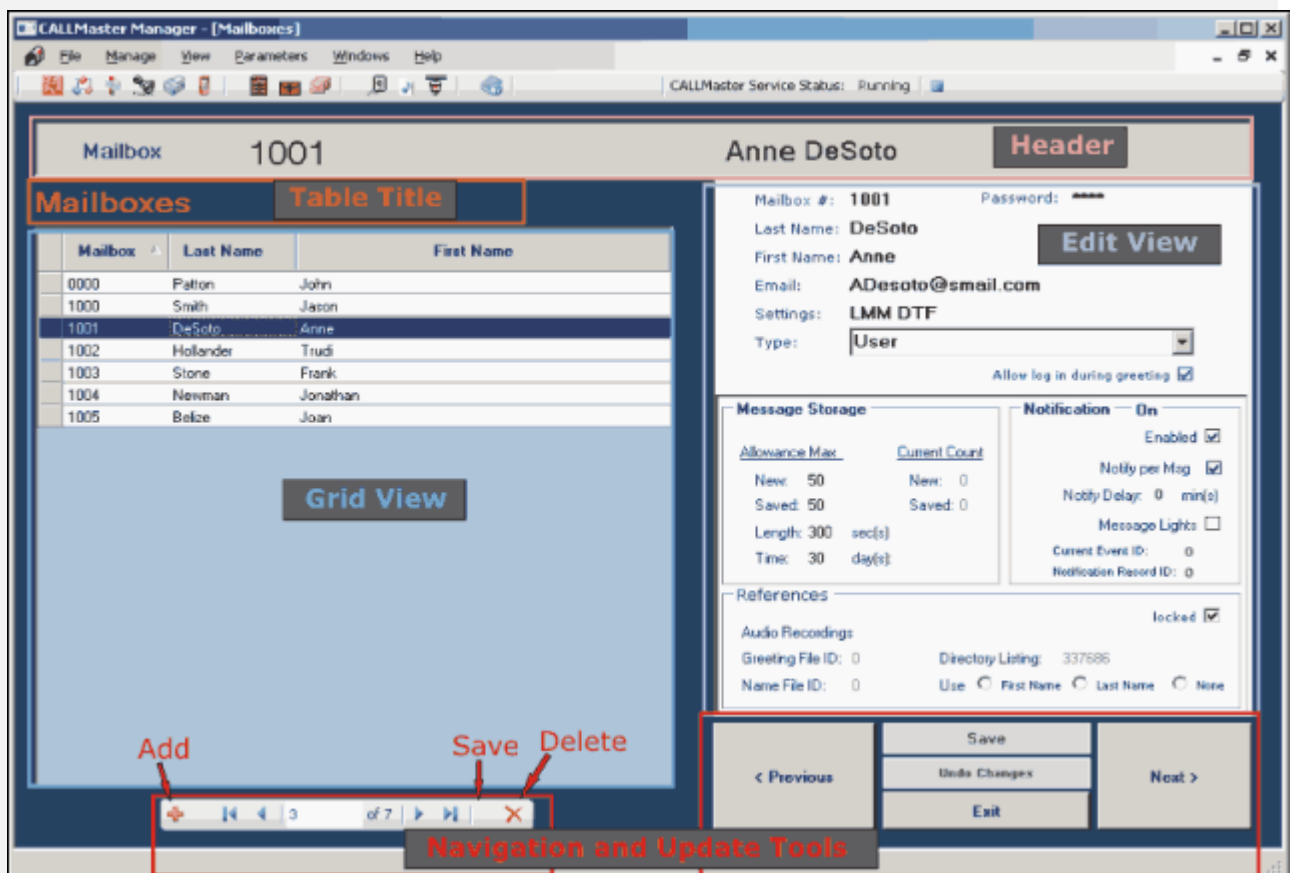
## Configuring CALLMaster - Edit Records

Open CALLMaster Manager and select the desired table.

On the Grid View section of the screen, select/highlight the record to see details. The detail fields for the selected record will be displayed on the Edit View section of the screen.

All changes must be made to the fields on the Edit View section of the screen. Change fields as required.

Click on any other section of the screen for the changes to take effect or click on the Save button. You can click on the Undo Changes button to not save the changes and keep the original data.



## **Manage CALLMaster**

[CALLMaster Manager](#)

[CALLMaster Main Window](#)

[Controlling Applications](#)

[Activating Email](#)



## Manage CALLMaster - CALLMaster Manager

The CALLMaster database provides a powerful yet simple way to define your applications. All of the action choices and speech files available to a caller are determined through the various tables of the database.

The main purpose of the **CALLMaster Manager** is to manage the data in these tables.

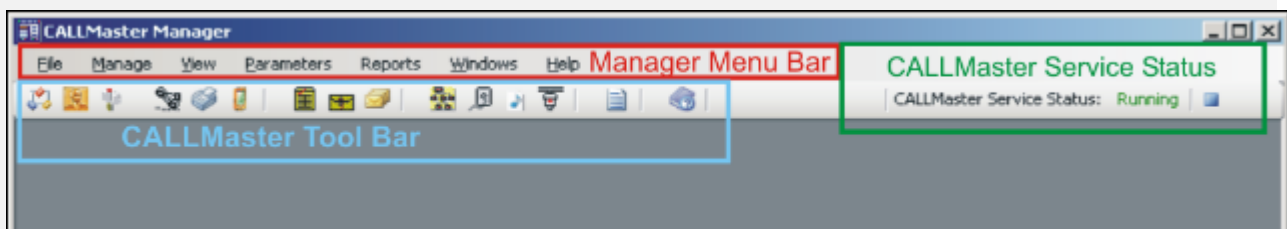
To open the **CALLMaster Manager**,

- choose **Manager** from the **Start | All Programs | CALLMaster** group; or



- click on the **CALLMaster Manager** icon on your desktop.

When CALLMaster Manager opens, it displays the **Manager Menu Bar**, the **CALLMaster Tool Bar** and the **CALLMaster Service Status**.



From the **Manager Menu Bar**, you can **Manage** the various CALLMaster tables, update the **Parameters** or run **Reports**.

The **CALLMaster Tool Bar** provides icons as a quick way to manage the tables or run reports.

The **CALLMaster Service Status** provides the status of the CALLMaster service (Running, Stopped) and allows you to control it.

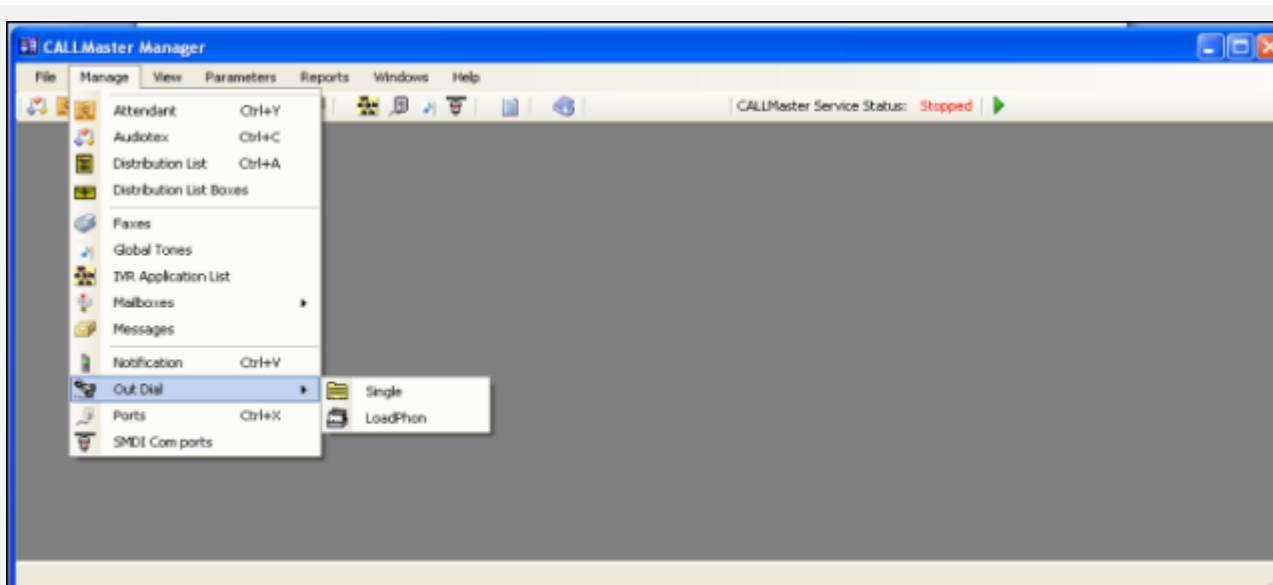
### CALLMaster Manager | Manage

You can click on the desired table Icon from the CALLMaster Tool bar to access the associated table. Alternatively, select from the **Manage** drop down list to configure individual mailboxes, attendant records, distribution lists, notification records, etc.

**NOTE: If the CALLMaster Service is running, any data changes made will take effect immediately except for Phone Ports.**

Phone Port settings changed via the CALLMaster Manager will only take effect the next time the CALLMaster service is restarted.

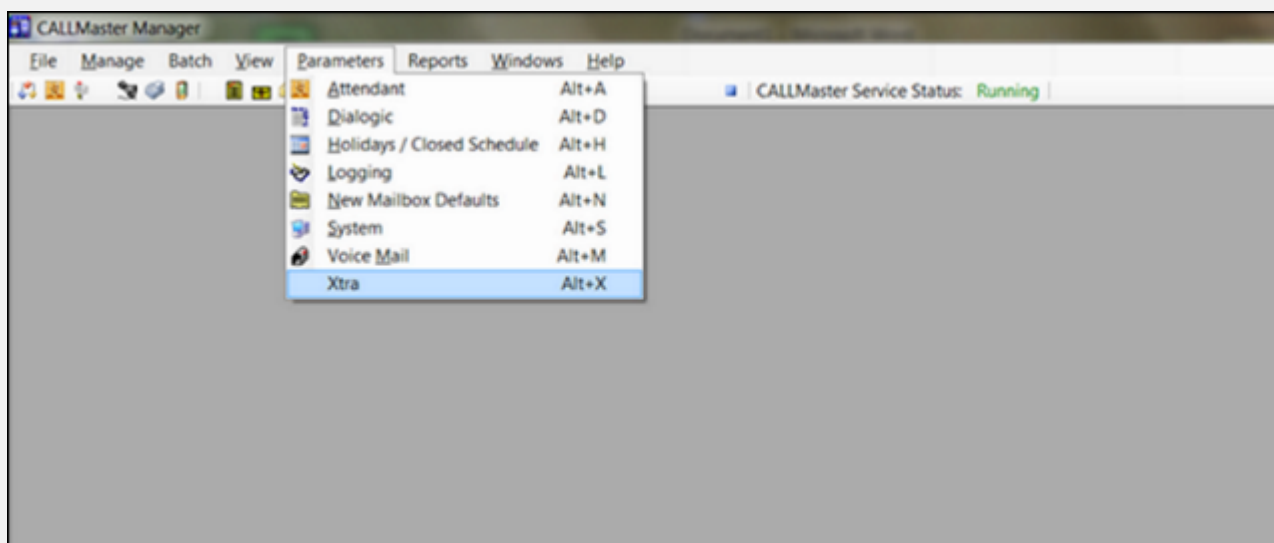
To change the Phone Port Application without restarting the CALLMaster service, see [Walkthrough: Change Application Without Restarting CALLMaster Service](#).



## CALLMaster Manager | Parameters

Select from the **Parameters** drop down list to configure CALLMaster for attributes such as telephone system requirements and user preferences, the hour after which callers hear the morning, afternoon, or evening greeting. the length of messages, or how long to wait for the line to settle after transferring, among other things.

**NOTE:** If the CALLMaster Service is running, any data changes made will take effect immediately.



Once you have selected the table, learn how to [Configure CALLMaster](#).

## CALLMaster Manager | Reports

There are several reports standard reports provided with CALLMaster.

The first time you want to run reports, you must first install the Access Runtime from the CALLMaster\BIN by double clicking on AccessRuntime.exe. This needs to be done only once after installation of CALLMaster. You are now ready to select the Reports menu.

From the Manager Menu Bar, select **Reports** to access the menu options shown below.

CALLMaster Reports Menu - CALLMaster Reports

### CALLMaster Reports Main Menu

**MailboxActivityLog Reports**

- All Activity for a Selected Date by Mailbox
- Activity for a Selected Mailbox in a Date Range
- All Activity for a Selected Date
- All Activity for a Selected Date Range

**AttendantActivityLog Reports**

- All Activity for a Selected Date by Attendant ID
- Activity for a Selected Attendant ID in a Date Range
- All Activity for a Selected Date
- All Activity for a Selected Date Range

**Out Dial Reports**

- Out Dial Detail Activity for Start Date Range
- Out Dial Summary for Start Date Range
- Out Dial Completed Detail Activity for Start Date Range
- Out Dial Completed Summary for Start Date Range

**Miscellaneous Reports**

- Fax Sent Activity for Start Date Range
- Fax Received Activity for Start Date Range
- Speech Recognition Activity for Start Date Range
- Mailboxes Pending Notification Login - Detail
- Mailboxes Pending Notification Login - Summary

Records: 1 of 1 | No Filter | Search

Form View

Powered by Microsoft Office Access

If you are using the CALLMasterSQL database, make sure your SQL Server is running and the database is attached.

## Manage CALLMaster - CALLMaster Main Window


The CALLMaster Main Window is used to start the CALLMaster service if it is not already running; to view activity; to stop and start the service, all lines or a single line; and to manage parameters and ports.

### Open the Main Window

You can open the CALLMaster Main Window via the desktop shortcut or from the 'Start/All Programs/CALLMaster' group.

- If the CALLMaster service is running, the Main Window will automatically open.
- If the CALLMaster service is not running, you will be given the option to start the service.

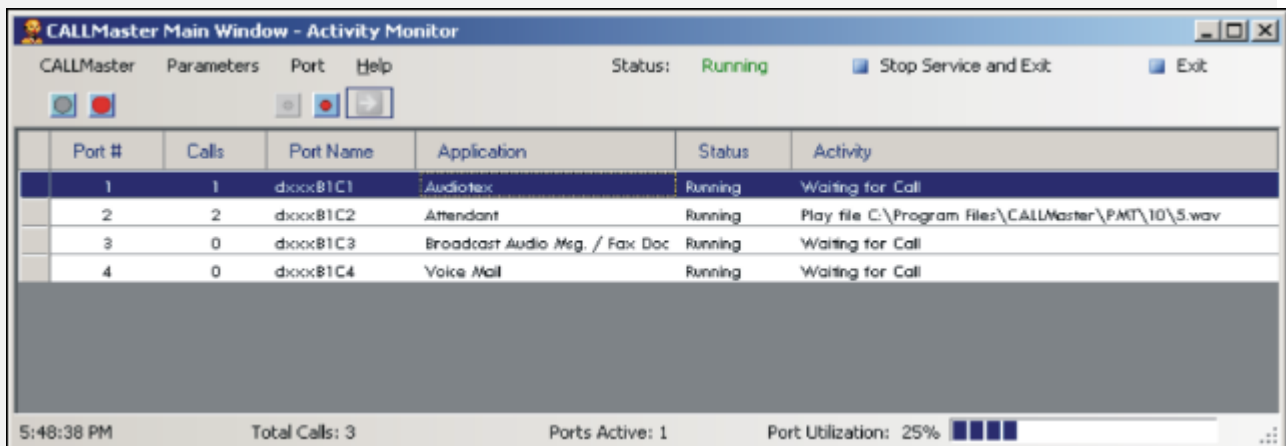
The CALLMaster service must be running to open the Main Window. However, the Main Window does not need to be open for the CALLMaster service to run.

If you minimize the Main Window, the CALLMaster Main Window icon  will appear on the bottom right of your Task Bar. By right clicking on the CALLMaster icon, you can:

- Open Activity Window to open the CALLMaster Main Window;
- Start All Ports;
- Stop All Ports selecting Immediately (terminate current activity) or Gracefully (wait for current activity to end before stopping each line);
- Exit to close the Main Window.

### View Activity

The Main Window displays the installed ports and the real time status of each port. It also displays the number of '**Calls**' per line and '**Total Calls**' inbound and outbound handled by the system since the last service start.



Port #	Calls	Port Name	Application	Status	Activity
1	1	dxocB1C1	Audiotex	Running	Waiting for Call
2	2	dxocB1C2	Attendant	Running	Play file C:\Program Files\CALLMaster\PMI\10\5.wav
3	0	dxocB1C3	Broadcast Audio Msg. / Fax Doc	Running	Waiting for Call
4	0	dxocB1C4	Voice Mail	Running	Waiting for Call

5:48:38 PM      Total Calls: 3      Ports Active: 1      Port Utilization: 25% ■■■■

Information for each port includes:

- the '**Port #**';
- the number of '**Calls**' it has received/made;
- the current '**Application**' (e.g. Voice Mail, Attendant, Audiotex, <Custom IVR>) running on the port;
- the '**Status**' (Running/Stopped); and
- the '**Activity**' (Waiting for Call, Suspended, Resumed Waiting for Call or the activity currently being executed on the line e.g. GetDTMF, prompt currently being played).

Information displayed at the bottom of the screen shows:


- the current time. It must be correct for the system to properly record the date and time your calls are received;
- '**Total Calls**' inbound and outbound handled by the system since the last service start;


- the number of '**Ports Active**';
- the '**Port Utilization**' as a percent of the total.



## Stop and Start the Service, All Lines or a Single Line

The CALLMaster service '**Status**' is displayed on the top right with the options to '**Stop Service and Exit**' the Main Window or just '**Exit**' the Main Window.

You can Start and Stop an individual line or all lines.

- Select 'CALLMaster | Stop All Ports' or press the large Red button  to stop all lines from answering calls.

- To stop a single line, highlight the line and pressing the small Red button .

- The big Green button  and the small Green button  are used to start all lines or a single line.

If you experience a line not responding or the entire program not responding you might want to wait a few minutes to confirm that it is not responding or just processing. You run the Task Manager to check for 100% CPU utilization (small bar graph in icon tray when launched), which would indicate an infinite loop. If CPU utilization is 100%, then you may Highlight that line, issue a Line Stop (small Red button) and then a Start (small Green button).

## Manage Parameters and Ports

Use the '**Parameters**' drop down list to update the required parameters. The change values will take effect as soon as the record is updated.

Use the '**Ports**' drop down list to 'Start Selected Port', 'Stop Selected Port' or 'Change Application'. The selected Port must be highlighted to make any of these changes. The application change will take effect on the next call made/received on that port.

If you are using the CALLMasterSQL database, make sure your SQL Server is running and the database is attached.

## Manage CALLMaster - Controlling Applications

Each line of CALLMaster can run a different application. This is controlled via 'Ports' in the CALLMaster Manager or the Main Window.

To change the application via the CALLMaster Manager, use the **Manage | Ports** table. Highlight the required port and change the 'Application'. If the CALLMaster service is running, you will need to stop and start the service.

See [Walkthrough: Change Application Without Restarting CALLMaster Service](#) to change the application without stopping the CALLMaster service.

Set each line to a different 'Application' if you wish to run a different application on each line.

## Manage CALLMaster - Activating Email

Starting in CALLMaster 4.0.4 and CALLMasterIP 5.2.4.4, the method of sending emails has changed from Simple MAPI or MS Exchange Server to CDOSYS.DLL.

This new method of using CDOSYS.DLL requires the IP Address or Domain Name of the SMTP server, the User Name and Password to send emails. It does not require a MAPI Client or for the CALLMaster Service to "Logon As".

### Using CDOSYS.DLL

There are 2 System Parameters that need to be set to activate email processing in CALLMaster and CALLMasterIP.

1. System Parameter #26 - 'System Settings' must contain the flag EMAIL<startalertaddress>' to activate the email process.
  - o e.g. EMAIL<Support@companyx.com>The <startalertaddress> should receive an email every time CALLMaster service starts.
2. System Parameter #30 - 'MAPI Profile' is used to configure the CDO email mode and settings. You can select from five (5) modes to send emails.

#### o **Mode 1 - Write to Pickup Folder .**

In this mode CALLMaster delivers emails to a specified folder where it is presumed an SMTP server is checking for emails to be picked up and sent.

**Format:** <Mode>;<PUFolderLocation>;<Sender>;  
- semicolon required after each parameter

<u>Parameter</u>	<u>Description</u>
------------------	--------------------

<Mode>	1
--------	---

<PUFolderLocation>	URL of pickup folder on>
--------------------	--------------------------

<Sender>	Name of Sender for Email Header
----------	---------------------------------

**Example:** 1;c:\wwwroot\emailroot\pickup\;sales@mycompany.com;

---

#### o **Mode 2 - Send to SMTP Server.**

In this mode CALLMaster uses the Windows OS to send emails directly to an SMTP server via the standard Telnet protocol.

**Format:** <Mode>;<SMTPServerAddress>;<Port#>;<UserName>;<PW>;<AuthMode>;<Sender>;  
- semicolon required after each parameter

<u>Parameter</u>	<u>Description</u>
------------------	--------------------

<Mode>	2
--------	---

<SMTPServerAd dress>	Domain Name or IP Address of SMTP Server
-------------------------	--

<Port#>	Port Number (usually 25)
---------	--------------------------

<UserName>	Username for SMTP login authentication
------------	--

<PW>	Username for SMTP login authentication
------	--

<AuthMode>	Authentication Mode 0 - No Authentication; 1 - Basic; 2 - NTLM
------------	---

<Sender>	Name of sender for email header
----------	---------------------------------

**Examples:** 2;192.1168.25.31;25;JWilliams@companyx.com;jz87\*9;0;sales@companyx.com;  
2;mail.companyx.com;25;JWilliams@companyx.com;jz87\*9;0;sales@companyx.com;

#### o **Mode 3 - Send to SMTP Server and keep copy in a Backup Folder.**

In this mode CALLMaster uses the Windows OS to send emails directly to an SMTP server via the standard Telnet protocol and also writes the email message to a backup folder. This is intended to be used for redundancy, monitoring, archiving, debugging or any other scenario where an SMTP server is used and an archive of activity is wanted.

**Format:** <Mode>;<BUFolderLocation>;<SMTPServerAddress>;<Port#>;<UserName>;<PW>;<AuthMode>;<Sender>;  
- semicolon required after each parameter

<u>Parameter</u>	<u>Description</u>
------------------	--------------------

<Mode>	3
--------	---

<BUFolderLocati on>	URL of backup folder
------------------------	----------------------



<SMTPServerAd Domain Name or IP Address of SMTP Server  
dress>

<Port#> Port Number (usually 25)

<UserName> Username for SMTP login authentication

<PW> Password for SMTP login authentication

<AuthMode> Authentication Mode  
0 - No Authentication; 1 - Basic; 2 - NTLM

<Sender> Name of sender for email header

**Examples:** 3;c:\wwwroot\emailroot\backup  
emails\;192.1168.25.31;25;JWilliams@companyx.com;jz87\*9;0;John;  
  
3;c:\wwwroot\emailroot\backup  
emails\;mail.companyx.com;25;JWilliams@companyx.com;jz87\*9;0;John;

---

o **Mode 4 - Write to a Pickup Folder and keep copy in a Backup Folder.**

In this mode CALLMaster delivers emails to a specified folder where it is presumed an SMTP server is checking for emails to be picked up and sent; and also writes the email message to another backup folder. Normal email pickup behavior deletes the original email after it is sent, leaving no copy in the pickup folder. In this mode, an additional folder can be designated to maintain a backup copy of all emails sent.

**Format:** <Mode>;<PUFolderLocation>;<DSFolderLocation>;<Sender>;  
- semicolon required after each parameter

<u>Parameter</u>	<u>Description</u>
------------------	--------------------

<Mode>	4
--------	---

<PUFolderLocati on>	URL of pickup folder
------------------------	----------------------

<BUFolderLocati on>	URL of backup folder
------------------------	----------------------

<Sender>	Name of sender for email header
----------	---------------------------------

**Examples:** 4;c:\wwwroot\emailroot\pickup\;c:\wwwroot\emailroot\backup emails\;sales@companyx.com;

4;c:\wwwroot\emailroot\pickup\;c:\wwwroot\emailroot\backup emails\;sales@companyx.com;

---

**o Mode 5 - Folder Pickup.**

In this mode CALLMaster uses the Windows OS to send emails directly to an SMTP server via the standard Telnet protocol and also writes the email message to 2 separate folders (a pickup folder and a backup folder). This is intended to be used for redundancy, monitoring, archiving, debugging or any other scenario where an SMTP server is used and archives of activity is wanted or pickup via another SMTP Server is initiated along with a backup in a designated folder.

**Format:** <Mode>;<PUFolderLocation>;<BUFolderLocation>;<SMTPServerAddress>;<Port#>;<UserName>;<PW>;<AuthMode>;<Sender>;  
- semicolon required after each parameter

<u>Parameter</u>	<u>Description</u>
<Mode>	5
<PUFolderLocation>	URL of pickup folder
<BUFolderLocation>	URL of designated folder
<SMTPServerAddress>	Domain Name or IP Address of SMTP Server
<Port#>	Port Number (usually 25)
<UserName>	Username for SMTP login authentication
<PW>	Password for SMTP login authentication
<AuthMode>	Authentication Mode 0 - No Authentication; 1 - Basic; 2 - NTLM
<Sender>	Name of sender for email header

**Examples:**

```
5;c:\wwwroot\emailroot\pickup\;c:\wwwroot\emailroot\backup
emails\;192.1168.25.31;25;JWilliams@companyx.com;jz87*9;0;John;
```

```
5;c:\wwwroot\emailroot\pickup\;c:\wwwroot\emailroot\backup
emails\;mail.companyx.com;25;JWilliams@companyx.com;jz87*9;0;JWilliams@compa
nyx.com;
```

3. Stop and start CALLMaster service. CALLMaster will send out a test email when it starts up to the email address(es) set-up in System Parameter EMAIL<startalertaddress> if one is specified or to the default email account test\_email@callmaster\_startup.com.

You have now activated email.

You are ready to Send Emails:

- via the [Email Message/Fax Notification](#); or
- via the CALLBasic [Email](#) command.

## Running CALLMaster

[Starting CALLMaster](#)

[Stopping CALLMaster](#)

**NOTE:** Running Anti-Virus software on the CALLMaster computer could stop CALLMaster. See [Anti-Virus Conflict](#) for details.

## Running CALLMaster - Starting CALLMaster

CALLMaster runs as a system Service (a program run by your computer instead of a logged in user) on your computer. Make sure your SQL Server is running and the CALLMasterSQL database is attached.

### CALLMaster Service Start-Up Options

#### Via the Control Panel | Administrative Tools | Services Window.

You can set-up the CALLMaster service to start automatically, manually or via a scheduled task.

1. **Start Automatically.** BY default, CALLMaster runs automatically every time your computer starts up. In turn, whenever CALLMaster starts up, it starts the Dialogic Driver Service, which controls the telephone board(s).
2. **Start Manually.** To have CALLMaster start manually, open the Control Panel | Administrative Tools | Services window and highlight CALLMaster. Right click and choose 'Properties' then set 'Startup Type' to 'Manual'. You will now have to manually start the CALLMaster service via the Services window by highlighting CALLMaster and selecting Action | Start.
3. **Start via a Schedule.** To setup a schedule for CALLMaster, use the Windows task scheduler to run the CMStart.cmd command from the CALLMaster/Bin folder.

#### Via the CALLMaster Main Window or the CALLMaster Manager.

At any time, you can start the CALLMaster service by opening the [CALLMaster Main Window](#) or via the [CALLMaster Manager](#) Service Status bar.

Neither the Manager nor the Main Window need to be open for the CALLMaster service to run. The current service status is reflected on both the Main Window and the Manager once the service is running.

**NOTE:** Running Anti-Virus software on the CALLMaster computer could interfere with and stop the CALLMaster service. See [Anti-Virus Conflict](#) for details.

## Running CALLMaster - Stopping CALLMaster

You can stop select CALLMaster Lines, all lines or stop the CALLMaster service:

- via [the CALLMaster Main Window](#) (Not for use with T1 Lines)

You can also stop the CALLMaster service:

- via the Services window.
  - Close all open CALLMaster screens (Main Window and Manager)
  - Open Start|Settings|Control Panel, select Administrative tools and open the Services window. Highlight CALLMaster, then click 'Action | Stop' menu choice or click on the Square black button (Stop) on the toolbar. That Stops CALLMaster service from running.
- using the CMStop.cmd command in the CALLMaster/Bin folder.

## Logging and Tracking Activity

There are several ways you can log CALLMaster activity.

[Text Log Files](#)

[Activity Log Tables](#)

## Logging and Tracking Activity - Using Text Log Files

Log files are located in the \Program Files\CALLMaster\Logs directory of the drive where you installed CALLMaster.

- **CMCalls\_<DATE>.txt** - CALLMaster text file for capturing call activity and user interactions. General information such as when the CALLMaster service started, the current CALLMaster version, Entry.bas script processing (if found), line number, application type, time call started, total calls processed in session, total lines active is logged to the file. Usage data such as maximum ports active for the day and since CALLMaster restarted, peak ports active for the day and since CALLMaster restarted, is logged at the start and end of each call. Additionally, call specific information for mailbox, attendant, notification and other activity is also logged to the file.

This file is automatically created by CALLMaster at the start of each day (12:01 am) in text format and can be opened in Notepad for review.

The "Logging Parameters" Trace Level is used to control level of details output to this file. By default this is set to level 0 so minimum trace information is written. Set to 1 or 2 for use in debugging. Make sure that for normal use, the Trace Level is set to 0 as a higher logging level will soon fill up the hard disk.

Regular maintenance to delete aged files must be performed.

- **CMTrace\_<DATE>.txt** – Not Used and is set to Null.

Each call is also logged to the [CallLog](#) table in the CALLMaster SQL database.



## Logging and Tracking Activity - Using Log Tables

In addition to the call log text file, there are 5 database tables available for logging.

**AttendantActivityLog**- data file for capturing Attendant activity;

**CALLLog**- data file for logging all call activity.

**FaxSendActivityLog** – data file for logging outbound faxes.

**MailboxActivityLog** - data file for capturing mailbox activity including notification;

**SRResultLog**– data file for logging Speech Recognition activity.

## Logging and Tracking Activity - Using Mailbox Activity Log Table

CALLMaster is shipped with the MailboxActivityLog disabled. To activate Mailbox logging, using the CALLMaster Manager, select **Parameters | System** and add the codes 'MAL' to the 'System Settings' field. To disable, just delete the code 'MAL'.

**Note:** Once activated the MailboxActivityLog will start collecting data records. It is the Administrator's responsibility to periodically archive data and remove records from the log. Otherwise the size of the database could grow to a size that inhibits proper operation of the system.

For Mailboxes, the events that are logged are:

Event	Type	Event	Type
MSG RECEIVED	100	MAILBOXOPTIONS	230
MAILBOXNOTIFIED	110	MAILBOXLOGOUT	240
MAILBOXEMAILED	120	NOTIFICATION LOGOUT	241
MAILBOXEMAILED_FAX	121	FAXREVIEW	250
MAILBOXMSGLIGHT	130	FAXPLAYED	251
MAILBOXLOGIN	200	FAXSAVED	252
MESSAGE REVIEW	210	FAXDELETED	253
MESSAGE PLAYED	211	FAXFORWARDED	254
MESSAGE SAVED	212	FAXREPLIED	255
MESSAGE DELETED	213	FAXRECEIVED	260
MESSAGE REPLIED	214	MESSAGE DELETED - AGED PAST DELETE DATE	261
MESSAGE FORWARDED	215		
MESSAGE SENT	220		

Each mailbox event is stored in the table MailboxActivityLog. Each Event record contains the following information:

EventID	Unique ID for activity event
Type	Event type code (see above)
Date and Time	Event Date and Time
Description	Event description
MailboxID	Mailbox Number
Event Data	Event data e.g. message receipt date, target mailbox number as appropriate.
Notes	User defined note field; may contain additional data or comments. E.g. result of notification, lapsed time of call

## Logging and Tracking Activity - Using Attendant Activity Log Table

CALLMaster is shipped with the AttendantActivityLog disabled. To activate Attendant logging, using the CALLMaster Manager, select **Parameters | System** and add the codes 'AAL' to the 'System Settings' field. To disable, just delete the code 'AAL'.

**Note:** Once activated, the AttendantActivityLog will start collecting data records. It is the Administrator's responsibility to periodically archive data and remove records from the log. Otherwise the size of the database could grow to a size that inhibits proper operation of the system.

For the Attendant the events that are logged are:

Event	Type	Event	Type
FIRST	100	VOICE MAIL	107
BLIND TRANSFER	101	AUDIOTEX	108
SUPERVISED TRANSFER	102	AUDIOTEX KEY	109
SCREENED TRANSFER	103	ALTERNATE MESSAGE	110
BLOCKED TRANSFER	104	LOG INTO MAILBOX	111
DIRECTORY	105	DISABLED	112
MAILBOX MESSAGE	106	SUPERVISED ENDED	113

Each Attendant event is stored in the table AttendantActivityLog. Each event record contains the following information:

Field Name	Description
EventID	Unique ID for activity event
Type	Event type code (see above)
Date and Time	Event Date and Time
Description	Event description

ExtensionID	ExtensionID selected
Event Data	Event data
Notes	User defined note field; contains result of transfer e.g. connect, alternate attendant
Port	The port number the call came in on.
DNIS	The DNIS forwarded by the phone switch/service
CallerID	The Caller ID forwarded by the phone switch/service
Number To Dial	The Numer to dial obtained from the Attendant record
Secs	The length of the call in seconds

## Logging and Tracking Activity - Using Fax Send Activity Log Table

This table is used to log outbound faxes.

CALLMaster is shipped with the FaxSendActivityLog disabled. To activate Fax Send logging, using the CALLMaster Manager, select **Parameters | System** and add the codes 'FAL' to the 'System Settings' field. To disable, just delete the code 'FAL'.

**Note:** Once activated the FaxSendActivityLog will start collecting data records. It is the Administrators responsibility to periodically archive data and remove records from the log. Otherwise the size of the database could grow to a size that inhibits proper operation of the system.

For Fax Send Activity, the events that are logged are:

Event	Type
Fax sent via CALLBasic	100
Fax sent via MAILBOXNOW	110
Fax sent via MAILBOXFORWARD	120

Each outbound fax event is stored in the table FaxSendActivityLog. Each event record contains the following information:

Field Name	Description
EventID	Unique ID for activity event
Date and Time	Event Date and Time
Type	Event type code (see above)
Seconds	The length of the call in seconds
Description	Event description
FileName	Full path name of file faxed
Mailbox	Source (mailbox #, Audiotex, ODFAX(OutcallSchedule table))
Data	

Notes

Fax status of either Faxed OK or Fax FAILED

## Logging and Tracking Activity - Using SR Result Log Table

This table is used to log Speech Recognition (SR) results.

CALLMaster is shipped with the SRResultLog disabled. To activate SR results logging, using the CALLMaster Manager, select **Parameters | System** and add the codes 'SRL' to the 'System Settings' field. To disable, just delete the code 'SRL'.

**Note:** Once activated the SRResultLog will start collecting data records. It is the Administrators responsibility to periodically archive data and remove records from the log. Otherwise the size of the database could grow to a size that inhibits proper operation of the system.

Each SR event logged is assigned a unique Recognition Event Id, , , , engine confidence score, the required confidence set for the word (1 Low, 0 Medium, 1 high), SAPI returned Confidence, the name of the file in the SRAudioArchive that has the speech, the port number on the system, and date and time of recognition.

Each SR event is stored in the table SRResultLog. Each event record contains the following information:

Field Name	Description
RecognitionEventID	Unique ID for activity event
XMLGrammar	XML Grammar file used for this recognition
PropVal	Property ID from XMLGrammar file or value returned ** Failed ** if not recognized
WordsRecognized	Text of the recognized word
EngineConfidence	Engine confidence score
RequiredConfidence	The required confidence -1 Low, 0 Medium, 1 high
SAPICConfidence	SAPI returned Confidence
AudioFileName	The name of the file in the SRAudioArchive that has the speech
Notes	Port
SRTIME	Date and time of recognition



## Maintenance

Your computer will benefit from regular maintenance. The electronic components of the computer and speech board rarely break unless subjected to electronic surges like lightning, or abuse. The hard disk, on the other hand, moves continuously and will wear out. Finally, speech applications tend to be disk-intensive; frequent recording and deleting disorganizes or 'fragments' the hard disk. For these reasons you should perform regular hard disk maintenance.

We think it is helpful to do CALLMaster maintenance at least some of the time, because you may get an early indication of hard disk failure, or you may notice an unusual number of errors that would indicate problems with your setup. For instance, if you get disk errors on a machine that never had disk errors, or you get an increasing number of disk errors, your hard drive might be getting old.

We recommend more frequent maintenance for systems that have one or more of the following characteristics: heavy voice mail call volume, heavy notification volume, frequent speech file changes, little free disk space, or numerous errors (check the log file to see what they are). The most common symptom of inadequate maintenance is choppy messages, a sign of disk fragmentation; in that case, perform maintenance more often.

If you have activated the MailboxActivityLog and AttendantActivityLog tables, make sure you delete aged records, as these files can easily grow in size and cause CALLMaster performance to slow down.

## Operating CALLMaster - Back-Ups

You should be able to quickly replace your CALLMaster system in the event of hardware failure, and be able to easily undo configuration errors. **You should have copies** of the CALLMaster databases (CALLMaster.mdf) from your system. A complete backup would be a full copy of the CALLMaster home folder by default the c:\Program Files\CALLMaster. In addition, you should be able to replace important speech files. Get a backup at regular intervals, as well as before and after making changes to configuration or databases. Have copies of these files:

- CALLMaster database in the CALLMaster\Data folder (CALLMaster.mdf )
- Important speech files (e.g. greeting and menu prompts, Audiotex speech files, etc.)
- CALLMaster Software License file in the CALLMaster\Data folder
- Scripts in the CALLMaster\Scripts folder

It is highly recommended that you backup the entire \Program Files\CALLMaster folder.

You should of course have copies of any important files you have changed.

The best scenario is to have back-ups and never need them. The worst scenario is to need a backup and not have a current one. Trust us.

# THE CALLMASTER TOUR

This section provides an overview of CALLMaster functions, tables and the underlying databases. It is highly recommended reading for all first time users and those requiring a refresher course.

We will first take [The Simple Tour](#). This will provide step by step instructions on how to:

- [Log In to CALLMaster](#)
- [Use CALLMaster Manager](#)
- [Set-up Phone Ports](#)
- [Make Attendant Records](#)
- [Create Mailboxes](#)
- [Initialize Notification](#)
- [Email Message/Fax Notification](#)

You will then [Run a Test](#) of work done so far.

We will then take [The Advanced Tour](#) of CALLMaster which will show you how to:

- [Create Audiotex Modules](#)
- [Record Audiotex Module Messages](#)
- [Record System Prompts](#)
- [Create Fax Table Entry](#)
- [Work with CALLBasic](#)

## **CALLMaster Tour - Simple**

This will provide step by step instructions on how to:

[Log in to CALLMaster](#)

[Use CALLMaster Manager](#)

[Set-up Ports](#)

[Make Attendant Records](#)

[Create Mailboxes](#)

[Initialize Notification](#)

[Email Message/Fax Notification](#)

You will then [Run a Test](#) of work done so far.

[\*\*Back\*\*](#)

## CALLMaster Tour - Log in to CALLMaster

Before proceeding, make sure you have installed CALLMaster and [validated the installation](#).

To log in to CALLMaster, call the phone line connected to CALLMaster. CALLMaster will answer, greet you, and play the voice mail main menu. Make sure you know your mailbox number and password and it has been set-up by the administrator. CALLMaster ships with Administrator mailbox 0000 and password 0000.

The pound key (#) can be used to back up CALLMaster one menu at a time, finally exiting altogether. Except for Browsing Mailboxes (BRS), only one user can log in to the mailbox at a time. Second caller attempting to log in will hear prompt 391 "Mailbox Busy. Try again later". That's it. Give it a try.

### Log In to Your Mailbox

Follow these simple steps:

1. Call CALLMaster. When you hear the main greeting 'Please enter a box number, or press two for a directory. Press star to log in, or hold on to leave a message', press star (\*) to log in to a mailbox.
2. Enter your box number and password, then press star (\*).

### First Time Log In

If this is your first time logging in, you will hear the new box holder's prompt, which is 'Welcome to voice mail. I'll prompt you (etc.).'

Follow the instructions to record your Greeting, your Name, and to spell your name with touch-tones for the Directory Listing. Remember to select 1 to approved your entries.

#### Mailbox # And Password

Your mailbox is identified by a mailbox number and protected by a password. After your first log in, the mailbox record will also have two recordings from you--your 'name' and 'greeting', as well as the spelling of your name for use in the directory. You have the ability to change everything but your box number; your system administrator can do that for you. Your box number is four digits long, but your password can be zero to four digits. You must signal the end of your password by pressing the star key (\*), even if your password has four digits. If CALLMaster does not recognize your mailbox number or password, it will ask you to try again. If you ever have trouble entering your mailbox number and password, try a slightly longer tone on the first digit of your mailbox number. Once you have logged in, CALLMaster will give you the number of new messages, saved messages, new faxes and saved faxes you have, except if your box Type is Announcement (Announcement mailboxes cannot receive or send messages). If your box Type is Announcement you will be played the Personal Options menu.

#### Mailbox Greeting and Name

Your Greeting is played for outside callers.

Your Name is used to identify you to your fellow box holders, and is played when callers enter your name in the Voice Mail directory.

Both are easily changed whenever you want; for instructions see [Voice Mail - Set Personal Options](#). The prompts will tell you when to start recording, and how to stop recording; and then how to save, replay, or redo your recording. CALLMaster uses a short recording tone to indicate when recording starts and stops.

#### Mailbox Directory Listing

During your first log in, you will also be prompted to spell your last name, for use in the box holder directory. Use the touch-tone keys to spell your name (use '7' for 'Q,' and '9' for 'Z'). Then you will be prompted to approve, hear, or redo the spelling; remember that the system will read back numbers, not letters. When a caller uses the Voice Mail directory, and spells your name, he will hear your recorded Name.

If you do not want CALLMaster to prompt you for your name, greeting and directory listing the first time you log in, use the 'NGN' flag in the Mailbox Settings field or in the 'Default Flags' field of the Voicemail Parameters

### Subsequent Log In Attempts

After you have made the recordings or on subsequent log in attempts, once you have you logged in to your mailbox, you will hear the message count and the voice mail menu 'Press one to review messages, two to send a message, or three to set personal options.'. Select the option desired and follow the options provided.

### **Administrator Options Menu**

The advanced features of CALLMaster are available by logging in to a mailbox with Administrator privileges. Log in as above then press \* for the Administrator Options when you hear the voice mail menu 'Press one to review messages, two to send a message, or three to set personal options'. **You will not be prompted for this input.**

If you were able to accomplish the above steps, then your CALLMaster installation is successful!

You may test out your mailbox, or press pound (#) to exit your mailbox.

[Continue with the CALLMaster Tour](#)

[Return to CALLMaster Tour Overview](#)

[Back](#)

## Using CALLMaster Tour - CALLMaster Manager

### Overview

The main purpose of the CALLMaster Manager is to manage the CALLMaster Database. You can also Stop and Start the CALLMaster Service via the Manager.

Use the **CALLMaster Manager | Manage** function to maintain the CALLMaster Data tables or the **CALLMaster Manager | Parameter** function to manage the CALLMaster Parameter tables. See [CALLMaster Manager](#) for details.

### Screen Configuration and Record Management Capabilities

You can view the the CALLMaster Manager [Screen Configuration](#) and then select the desired table to:

[Add Records](#)

[Delete Records](#)

[Modify/Edit Records](#)

As we will be modifying several of the CALLMaster tables, it would be best to [Stop CALLMaster Service](#) before proceeding.

The first table we'll look at is the 'Phone Ports' table.

[Continue with the CALLMaster Tour](#)

[Return to CALLMaster Tour Overview](#)

[Back](#)

## CALLMaster Tour - Set-up Ports

The Ports table lets you configure how each port or phone line answers and/or makes calls. Every call begins in the Port table. It defines how many rings to wait before answering, what to say first (greetings), and where to go next (application).

CALLMaster has already created Port records based on the number of lines supported. Let's view those records first.

Access the Ports table from the CALLMaster Manager Menu Bar by selecting **Manage | Ports**.

Remember, on the Manager's Grid View you can simply highlight the record you are interested in and see the record details in the Edit View section.

Now let's edit the ports.

- First, we want to set port number one as Attendant and verify some of the settings. To do this, highlight the existing record for port number 1 on the Grid View. On the Edit View,
  - set 'Application' to "Attendant" using the drop down list box;
  - make sure the 'Answer Ringing' is enabled and set to answer 'After' 1 ring;
  - verify the Greeting File IDs for Morning, Afternoon and Evening greetings are set to prompts 19, 20 and 21, the Closed greeting prompt is 203 and Holiday greeting prompt is 209. We will re-record these later in the tour.
- You can change all the other ports to the same settings as Port 1.
- Since we want to turn message waiting lights on/off, we must activate a port to make outbound calls. Otherwise, CALLMaster will be unable to make the call it needs to make to turn on the light. ***If none of the phone ports has Setting 'Make Outbound Calls' enabled, notification will NOT occur.***
  - As notification calls are attempted starting at the highest available port, select your highest numbered port. In a 4-line system, this is port 4. Highlight port 4 and enable 'Make Outbound Calls' in the 'Settings' section.

NOTE: Message lights are only application for the Telephone Network and does not apply for IP integration.

Save your work by clicking on the 'Save' button. If you don't want to save the record (or changes to the record), close the window and answer 'no' to the save prompt.

If you have a 4-port analog system, your port records should have information as shown below:

Port Records				
Port #	Application	Audiotex Start Module	Port Type Outbound	Port Greeting File IDs (Morning, Afternoon, Evening, Closed, Holidays)
1	Attendant	3	No	19, 20, 21, 203, 209
2	Attendant	3	Yes	19, 20, 21, 203, 209
3	Attendant	3	Yes	19, 20, 21, 203, 209
4	Attendant	3	Yes	19, 20, 21, 203, 209

Now that we've instructed the Port to use the Attendant application, the next thing we'll want to do is make Attendant records.

**Note:** At this point, Out Dialing users should skip to the section called 'Audiotex' in this chapter. Take the Audiotex portion of the tour, and then skip to [Out Dial](#).



[Continue with the CALLMaster Tour](#)

[Return to CALLMaster Tour Overview](#)

[Back](#)

## CALLMaster Tour - Make Attendant Records

When Attendant is the application, CALLMaster:

1. Plays the Port Greeting (prompts 19, 20, 21 based on the time of day), then the Attendant main menu (prompt 505)
2. Waits for touchtone input (if no input is received, it goes to Attendant ID number zero)
3. Matches the touchtone input with an Attendant ID record
4. Makes the transfer to that Attendant ID (plays 'there is no such extension' if no match is found)

For purposes of this tour, the Attendant Main Menu will offer callers the following options: "If you know your party's extension please enter it now. For Sales press 1, for Support press 2 or for the Operator press 0". We will create several Attendant records to satisfy these options. The Attendant ID for each of these records will need to match the number entered by the caller at the main menu prompt. We will add Attendant records for the Operator option (ID = 0), Sales option (ID=1), Support option (ID=2), and an individual extension option (ID=333). Since CALLMaster is shipped with sample records, some of these Attendant records may already exist. If they do then modify the existing record. We will keep the default Attendant ID length set to 4 in the Attendant Parameters (i.e. the Attendant ID can be up to 4 digits).

To access the Attendant table from the CALLMaster Manager Menu Bar, select **Manage | Attendant**.

### Let's create Attendant records for the Operator.

Since the caller is asked to press 0 for the Operator, we will create several records whose Attendant ID will be zero (0). These records will allow the caller to speak to the Operator during business hours or leave a message.

1. The first record for the Operator will be used to transfer to the Operator's extension during the day. CALLMaster will "Supervise" the transfer. If the transfer fails due to a busy the caller will be forwarded to the Operator's mailbox to leave a message; if the transfer fails due to no answer, then CALLMaster will try the Sales extension.  
If you have a record with Attendant ID = 0 then highlight to view and edit details. If not then click on the 'Add new' (plus sign) button on the bottom tool bar to add a new record and enter zero as the Attendant ID number.
  - o Select the **'Time of Day'**. The time of day determines when this record is active, and is based on the system time in the computer. Leave it as 'All Day' for now.
  - o **Type of Transfer** determines what action CALLMaster takes when executing this record. Using the drop down list select 'Supervised'. This will allow CALLMaster to 'supervise' the call to determine whether or not it was answered. How do we know where they will be transferred? That's what you're going to set next.
  - o **The Number to Dial** field tells CALLMaster what digits to send to the phone system (or phone company). If you are working with a telephone switch, put in the extension number of the desired party. If you are using Centrex, put in the telephone number to dial including any required prefix (e.g. 9) and the area code. In our example, let's assume we are working with a telephone system. We want callers to press '0' for an operator or hold on for assistance. Our 'operator' happens to be at extension 100, so we're going to enter 100 in the 'Number to Dial' field. Enter this now. **Note:** If you are doing this tutorial on a phone system and you know your operator's extension, use it instead of 100.
  - o The next field is **'Settings'**. This field is used for various indicators, including channel bridging, 3-way call, etc. For purposes of this tutorial, we will keep this field blank.
  - o **The Directory Listing** is the numeric equivalent of the Attendant ID owner's name and is used when there is an Attendant directory, where callers spell the name of the person they are trying to reach. In our example, we are making a record for the operator. It is unrealistic to think any caller would use a directory to look up the operator, but for the sake of our example notice that the telephone keys 67372867 spell 'Operator'. You may want to leave this field blank and [Set-up Attendant Directory](#) by calling into the system.
  - o **Failed Transfer Action** fields are used only for supervised and screened Type of Transfers. We're performing a supervised transfer, so we want to set these fields up properly. The first one is 'Busy'. As you might guess, this is what happens when the extension we're dialing is busy. In this case, let's say our operator is handling another call. We have a few options regarding what happens to the call when the line is busy. In our case, let's place the caller

into a voice mailbox. Use the drop down list to select the 'Mailbox' option. The 'No Answer' field is also pretty self-explanatory. In our example, if the Operator is not at their desk, we want CALLMaster to try the Sales extension so let's put calls that aren't answered to Alternate Attendant. The screened reject does not apply to us since we are not performing a 'Screened' transfer here.

- **Additional Settings**
  - **Alt Attendant ID** would be used if we had selected 'Alternate Attendant' in the Failed Transfer Action area, and would point to another Attendant ID. The purpose of the field is to allow for a second Attendant ID record to be used in the event the first one failed due to a busy or no answer situation. You might use this in a 'track-me-down' scenario, when you want CALLMaster to try calling a person in several locations before going to a mailbox. As we have set-up for Alternate Attendant in case of no answer and wish to transfer the caller to Sales when the operator does not answer, let's select and enter the Sales Attendant ID in this field. In our example we selected Attendant ID 1.
  - **Mailbox** field says which mailbox to use for this Attendant ID. When there is a busy, we've told CALLMaster to send the caller to this mailbox. Let's select Mailbox 1000.
  - **Audiotex Module ID** does not concern us in this record. It is used when the Type of Transfer is 'Audiotex Module' and indicates where in Audiotex to send the call. As neither the Type of Transfer nor the Failed Transfer Action are set to Audiotex Module, this field is not used even if you have a value selected.
  - The **Retry Attempts** is used when the Failed Transfer Action is '**Retry**'. CALLMaster will retry the number of times specified after which it performs the **When Retry Fails** action.
- Click on the 'Save' button to **save the record** you have just made.
- 2. Now let's **make another Attendant record to handle operator calls in the evening**. We want to transfer to the operator during the day, but go directly to a mailbox in the evening. This will help prevent calls after hours from having to go through the entire process of waiting for the transfer to be attempted. We know that after hours, the operator has left the building, so instead of making the caller wait we're going to go right to the mailbox in the evening. Select 'Add new' (plus sign) button on the bottom tool bar to add a new record.
  - This time make the Attendant ID = 0, but in the 'Time of Day' field, select 'Evening'. This will create a record that will override the 'All Day' record in the evening (Evening is defined by the System Parameter 'Evening Starts At').
  - We want to avoid making callers wait for an operator who is not there. Make the **Type of Transfer** 'Mailbox.' In the evening when callers press '0', they'll go into the box listed in the 'Mailbox' field.
  - There is no Number to Dial, since we are going to a mailbox. Leave this field blank. Since we are not performing a transfer, there is no need for Failed Transfer Action - so skip them. We are not using the Alternate Attendant, so leave 'Alt Attendant ID' blank. In fact, the only field that is relevant to what this record is meant for is the 'Mailbox' field; make it mailbox 1000. Save the record. We now have a record for the evening that will override the 'All Day' record so that the caller will be put directly into the mailbox after hours.
- 3. The next step in creating a set of Attendant records is to provide callers with a way to go straight to the mailbox without trying to transfer to the operator regardless of the time of day. Make an Attendant record with the Attendant ID to match the mailbox number, Attendant ID = 9000. In Attendant ID record 9000 create a similar record to the evening Attendant ID you just made. The only difference will be that the record number is 9000 and the 'Time of Day' is 'All Day.' Now when a caller (or the phone system) wants to access the mailbox directly, they simply enter 9000 from the Attendant Main Menu.

This is also an important concept when using CALLMaster on a phone system that supports in-band integration. Having Attendant records for the transfer as well as one that goes directly to a mailbox is the cornerstone of in-band integration. If you will be using CALLMaster with in-band integration, be sure to read the chapter entitled 'Attendant' for more information on telephone system integration.

**Here's a tip:** If you don't want to save the changes you've made, simply close the window you are in and answer 'No' to the 'Record(s) have changed. Save changes?' prompt.

**Now you can create Attendant records for Sales, Support as well as for individual extensions similar to the records created for the Operator.** Change the Failed Transfer Action and the Additional

Settings as desired. As you see from the set-up below, calls to individual extension 333 will be 'Blind' transferred offsite to an external number.

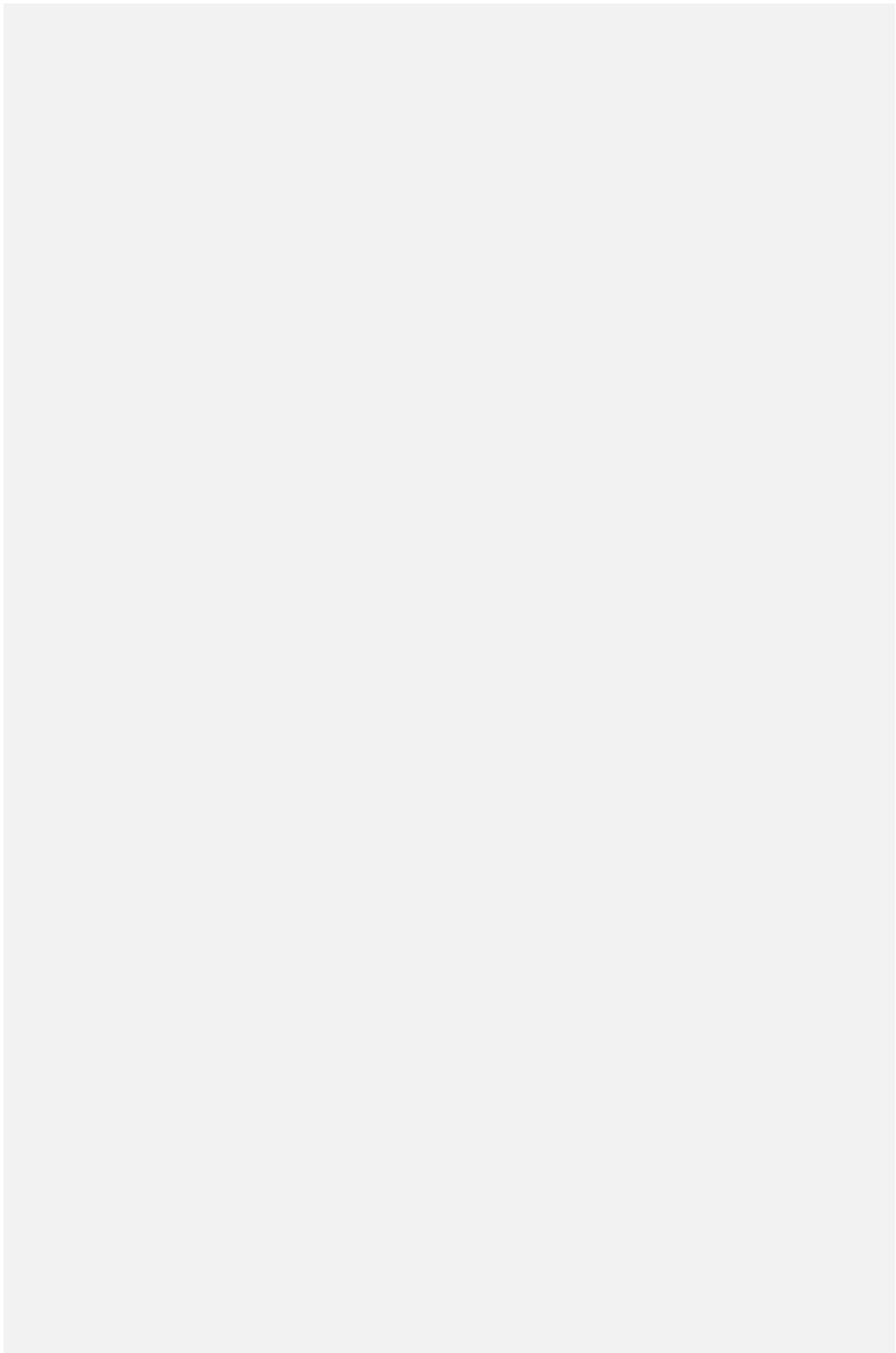
Your Attendant records should have information as shown below:

Attendant Records									
Attendant ID	Time of Day	Type of Transfer	Number to Dial	Failed Transfer Action			Additional Settings		
				Busy	No Answer	Screened Reject	Alt Attendant ID	Mailbox #	Auditex Module ID
0	All Day	Supervised	100	Mailbox	Alternate Attendant		1	1000	
0	Evening	Mailbox						1000	
9000	All Day	Log in to Mailbox						1000	
1	All Day	Supervised	111	Mailbox	Mailbox			1001	
1	Evening	Mailbox						1001	
9001	All Day	Log in to Mailbox						1001	
2	All Day	Supervised	222	Mailbox	Mailbox			1002	
2	Evening	Mailbox						1002	
9002	All Day	Log in to Mailbox						1002	
333	All Day	Supervised	333	Mailbox	Mailbox			1003	
333	Evening	Transfer	9,1,9143333333						
9333	All Day	Log in to Mailbox						1003	

[Continue with the CALLMaster Tour](#)

[Return to CALLMaster Tour Overview](#)

[Back](#)



## CALLMaster Tour - Create Mailboxes

- There is already one mailbox entered in the mailbox table, Mailbox # '0000', which is Type 'Administrator'. This is to allow the administrator to log in at the very beginning of the setting up process.
- Previously, when we created the Attendant ID records, we told CALLMaster to send failed calls and mailbox requests for the operator to mailbox 1000. The next logical step would be to set up that mailbox now.
- Select **Manage | Mailboxes** from the menu bar and click on the 'Add new' (plus sign) button on the bottom tool bar.
- The first thing it will ask you for is the mailbox number. Enter **1000** in this field. Mailboxes are usually four digits long.
- Fill in the **password** field. The password can be from zero to four digits long. Make it easy to remember. Enter 1000 here. Now when the operator wants to log into this mailbox, he will need to enter 1000 as his password.
- Enter the Last Name of the person that the mailbox belongs to. For now, let's just put 'Operator' in there. Enter this and look at the field called 'Directory Listing.' It figured out the 'numeric name' of the last name you entered, automatically. The First Name is for your reference.
- Add the mailbox owner's email address in the Email field if you want CALLMaster to automatically email messages or faxes received by the mailbox. Make sure that you have [Activated Email Notification](#) for this feature to work.
- The **Settings** field is used for special mailbox attributes (e.g. BRS-Browse Only, NMF-New Messages First, DTF-Date Time First – See [Voice Mail Settings field](#) description for details).
- The next field is the 'Type' which is essentially the activity rights of the box. Flip through the choices with the drop down list. Set the 'Type' field to 'User'.
- Let's activate the Notification indicator for message waiting lights. We will add a notification record later in the tour. Check the 'Notification Enabled' indicator and the 'Message Waiting Lights' indicator to enable message light notification record (record 0). You do not have a pager or human notification record. . The next section 'Notification' gives more details.
- The next four fields control message storage. The first governs the number of new messages the box is allowed to have. The next field deals with the number of saved messages a box is allowed to have. Maximum Message Length controls the length of messages as well as the allowable length of the mailbox's greeting. The final field here is the Maximum Message Age, which says that after 10 days a message will automatically be deleted from the system. **Save your work.**

[Continue with the CALLMaster Tour](#)

[Return to CALLMaster Tour Overview](#)

[Back](#)

## CALLMaster Tour - Create Notification Records

Notification records are used to tell CALLMaster how to signal the box holder that there is a new message in his box. Choose **Manage | Notification** from the CALLMaster Manager menu bar.

- Click on the 'Add new' (plus sign) button on the bottom tool bar.
- The first field is the 'Mailbox' which is the same as the mailbox number. Since we are adding a notification record for mailbox number 1000, enter '1000' here.
- The record number determines both the type of notification and the sequence of notifications. **Record number zero must be setup for Message Waiting light.** Anything else must be record number one (1) or higher. Record number two (2) would be used after record number one has been exhausted without success. Since we are hypothetically on a phone system here, let's make it a message light; enter record number as zero.
- The next field, **Dialing Prefix**, is where you would put a number to get an outside line if your PBX requires one (e.g. most PBXs require a '9' to dial out. The only time you would use the prefix field on a human or pager notification is if you require a dialing prefix before the area code, and do not use the System Parameter 'Local Dialing Prefix'. Since we are entering a message waiting light, we will not require an outside line and therefore not require a prefix. The **Area Code** and **Phone Number** field are self-explanatory. Put the extension that mailbox 1000 is attached to, which is 100 (or whatever you chose in the beginning of the tutorial) in the 'Phone Number.' field.
- In the '**Notification Type**' field, you are going to tell CALLMaster what kind of notification you wish to activate. View the drop down list until you get to 'Message Light'; then select it.
- Continue on to the '**When to Call**' section. Leave **Earliest Hour to Call** and **Latest Hour to Call** at 0 so that messages received at any time will trigger the notification. Keep in mind that the time is entered in military format (0 is midnight, 13 is 1:00pm, 23 is 11:00pm). One of the most common questions Technical Support gets in regards to notification is 'If I get a message during the time earliest and latest hours say NOT to notify me, can CALLMaster notify me once my notification turns back on?'. The answer is yes. If you get a message after the Latest Hour to Call, CALLMaster will notify you when it has reached the Earliest Hour to Call. Set all the days of the week to 'checked'.
- The **Call Attempt Schedule** fields are used to tell CALLMaster how many times to try and how often to try notification if there is no response to the notification. The first of these fields is for a 'No Answer', where we tell CALLMaster how many times to try the notification given a no-answer. In the 'every \_\_\_ minutes' boxes, we will tell CALLMaster how long to wait until retrying. There are controls for 'busy' and 'no-log in' also. No-log in means that the box holder hasn't logged in to check his messages. Since we are working with a message waiting light, it is unlikely that we're not going to get an answer. Set each of the next 6 fields to 1. If we were using a pager, we might set the try count to 2. CALLMaster would try 2 times before giving up. We might set the no-log in 'Every \_\_\_ minutes' box to 15, meaning the box holder has 15 minutes to log in before the system attempts another try.
- Save your work. Again, don't worry about the protected fields. These are not things you want to alter anyway.
- The next step in making notification work is to enter string data in the System Parameters 'MSG-Light-On Dial String' and 'MSG-Light-Off Dial String.'
- From the Manager Window, choose **Parameters | System Parameters** from the menu bar. Find the System Parameters 'MSG-Light-On Dial String' and 'MSG-Light-Off Dial String'. By double clicking on the parameter, a dialog box will appear where you make the appropriate selection or enter the appropriate information.
- If you are on a phone system for this tour and you know the true strings required here, go ahead and enter them. The brackets [ ] represent the extension number that you placed in the Notification record. If you don't know the strings or if you are not on a phone system, leave these fields as is.
- While we're here, we might as well set a few other parameters. Find the 'System's Area Code' and 'System's Phone Number' and fill them in. If you require it for your phone lines, fill in the 'Local' and 'Long Distance Dial Prefix'.
- Save your work.
- Now that you have enabled a port for outbound dialing, created a notification record, and updated the system parameter with the correct message waiting light string, when your operator gets a message in box 1000, CALLMaster will use the strings found in the 'MSG-Light-On Dial String' field to activate the message waiting light on the operator's phone. To turn off the light we'll use the 'MSG-Light-Off Dial String'

[Continue with the CALLMaster Tour](#)

[Return to CALLMaster Tour Overview](#)

[Back](#)



## **CALLMaster Tour - Email Message/Fax Notification**

If you want CALLMaster to send email notification on receipt of a message or fax to a mailbox, enter the recipient's email address in the Mailbox's [Email](#) field.

When a mailbox receiving the message or fax has been set-up with an address in the Email field and you have [Activated Email](#), CALLMaster will automatically send an email to that address.

[Continue with the CALLMaster Tour](#)

[Return to CALLMaster Tour Overview](#)

[Back](#)

## CALLMaster Tour - Test Set-up

Let's test what you've done so far

- As you have changed the Ports and System Parameter, make sure you have remembered to [Stop CALLMaster](#) Service from the Control Panel and then [Start CALLMaster](#) again.
- Call CALLMaster and leave a message for the operator's mailbox by entering 1000 at the Attendant Main Menu (prompt 505).
- [Open the CALLMaster Main Window](#) to see what CALLMaster does. After you leave the message, watch the highest port. You should see CALLMaster activate the port, and say 'Dial &,' then 'Dial \*3100#' (or the string you entered if you changed the default). If CALLMaster does not activate the port and attempt notification, go back and review your steps to make sure everything was done correctly. If everything happened as planned, CONGRATULATIONS! Please feel free to move on to the next section.

[Continue with the CALLMaster Tour](#)

[Return to CALLMaster Tour Overview](#)

[Back](#)

## **CALLMaster Tour - Advanced**

This section of the CALLMaster tour will show you how to:

[Create Audiotex Modules](#)

[Record Audiotex Module Messages](#)

[Record System Prompts](#)

[Create Fax Table Entry](#)

[Work with CALLBasic](#)

[Return to the CALLMaster Simple Tour](#)

[Return to CALLMaster Tour Overview](#)

[Back](#)

## CALLMaster Tour - Create Audiotex Modules

Some applications require multi-level automated conversations - if you don't need to use this feature of CALLMaster, please feel free to skip this section of the Tour. Out Dialing users should definitely take this section of the Tour.

Audiotex has many application possibilities, all of which stem from use of or a combination of two basic concepts Audiotex can be used to give callers choices of different interactive paths to travel down (we'll call this 'branching'), or to send callers down a pre-determined single path (which we'll call 'sequential path'). The first one we should look at is the 'branching.'

In branching, Audiotex modules or records are used to give callers the option to determine their next step. Each module goes through four steps, which are: 1) Play a message to the caller; 2) get a touch-tone response from the caller; 3) perform any designated CALLBasic scripts; and 4) determine and execute the next record. This is very important to keep in mind as you travel through this section of the tour. A great way to plan Audiotex routines is to draw a flowchart. Audiotex records perform four steps in the following order:

1. Say something by playing a speech file identified in the 'Message ID' field
  2. Get a touchtone response, based on the contents of the 'Response Length' field
  3. Perform a CALLBasic script (designated in the 'CALLBasic Script File')
  4. Determine the next module/record
- First, let's create our start module/record (remember that term). Select **Manage | Audiotex** from the menu bar, then click on the 'New' button on the top tool bar or click on the 'Add New'. Select a **Module ID** for the start module. You can enter up to ten digits here, so long as the number is less than 2,147,853,256. Enter the record number as '100'.
  - The second field deals with the '**Module Action**' of the record. The 'Module Action' defines what activity is to take place and helps to determine the next record. Take a moment to read the portion of the Audiotex chapter, specifically the section pertaining to the 'Module Action' field. Note that we are going to attempt a 'branching' sequence.
  - Use the drop down list to choose 'Touch Tone Table' as your Module Action for this record. This is a most effective tool for creating a 'branching' sequence.
  - The '**Next Module ID**' field contains the next module number to be executed. When the Modules Action is 'Touch Tone Table', the Next Module ID is used if the caller does not press anything in the time allowed. As we wish to keep the caller on the same module until they enter a valid selection, we will make the Next Module ID the same as the Module ID – enter 100.
  - '**Response Length**' is a very important field in branching applications. In this field, you tell CALLMaster to get a certain number of touch-tone digits from the caller. **Whenever you want to get responses or input from the caller while in a particular Audiotex record, you must have a response length.** In this first record, the caller is selecting which call path they wish to follow by pressing a one-digit response. Enter the number 1 in this field (Touch Tone Tables are always 1-digit). Now, CALLMaster will look for a single digit from the caller before continuing.
  - Use the '**Fail Module ID**' field to define which module to go to in the event the Touch Tone Table refers to an invalid module or some other error occurs in the attempt to find the next module. In the Fail Module ID record, you may want to end Audiotex and get out of the system. You also want to print an error message to the CMCalls file using the label field. We will create this record later. In the 'Fail Module ID' field, enter 9999. This means we will have to create a record 9999 to use as the fail module. We will get to that later.
  - The '**Write Label to Log File**' field is used to tell CALLMaster that the contents of the 'Label' field should be written to the ASCII file called 'CMCalls <DATE>.TXT'. Whenever you are working in Audiotex you should check this field - do so now. This allows you to debug more effectively.
  - The '**Label**' field's contents, such as comments, variable values, and touch-tone responses among other things, can be printed to the 'CMCalls' file. Enter the following in the label field: **Audiotex Start Module**
  - The '**Mailbox**', '**Attendant ID**' and '**Collect TT Input?**' fields are not used with Touch Tone Tables, so leave them blank.
  - The '**Touch-Tone Table**' is used when the Module Action is set to Touch Tone Table. Since our first record uses this Module Action, we need to fill this in. Each number in quotes ("") represents the touch-tone entered by the caller. The field to the right of each tone number indicates the module/record the caller will be sent to if that tone is pressed. If the caller presses 1, we want them to go to Audiotex module 101. Put '101' in the field to the right of the '1.' Next to '2', enter '102.' If the caller presses 2, they will go to Audiotex module 102. Numbers 3 through 0 will be used to let the

caller go to Audiotex modules 103 to 110 respectively. In fields 3-0, enter the number 103 to 110 respectively. If no response is entered the Next Module ID field is used to find the next module.

- For now, let's leave the 'CALLBasic Script File' line blank. It is reserved for CALLBasic programs, covered later in the tour.
- Save your work. Make sure your record looks like the one in the above figure. If not, you should retrace your steps and correct any mistakes.

Now we are ready to add another record. Record 9999 will be our exit record and our Fail Module ID. Let's add this record. Add record number 9999 to our routine. Go to 'Add' and start making the following 'Last Module' record as shown in Fig 4-12.

Save this record. Now we have a Last Module, which is also our fail module. You might say to yourself 'Won't Last Record end the call? We want to give callers an option to end or not.' Relax. This is eventually going to be a part of our Attendant. When you start the phone port with Attendant and enter Audiotex through some sort of Attendant ID, you'll return to Attendant when you exit Audiotex with a Last Record. Attendant will then ask you to 'Press 1 for further options.' If we had started the application in the phone port as Audiotex, we'd have to create another record for the caller and ask them if they wanted to exit the system.

Record 100, which we made before, states that if the caller presses 1, they will go to record number 101. Let's create record number 101 now. The purpose of record 101 will be to give the caller information about your company's product.

- There are several ways to play information to your caller. The most common (and by far the easiest) way this is done is by playing a speech file to the caller.
- Add record number 101 and make the Module Action 'Next Module' (See the 'Audiotex' chapter for more information on Module Key). Set the 'Write Label to Log File' to 'checked'
- In the 'Label' field enter 'play choice info to caller'
- In the 'Response Length' field, what should we put? We are going to simply say some information to the caller. There is no need to get digits from the caller, so leave the response length zero (0).
- The 'Next Module ID' field is important here. Since we have set the Module Action to 'Next Module', we must enter the next record's number in the 'Next Module ID' field. Enter '9999' here and go on to the next field.
- The 'Fail Module ID' is again 9999. The 'Mailbox' and 'Attendant ID' fields should be left blank.
- In the Touch Tone Table, we are going to leave each field at zero (0), since the action in this case is not 'Touch Tone Table'. Since we are not using the CALLBasic Script File yet, we're done with our entries. Save this record.
- Now add modules 102 thru 110 similar to module 101 to provide caller with information on various products if they select option 2, 3, 4, 5, 6, 7, 8, 9, or 0 in module 100.
- Save your work. We will soon learn how to make recordings for these modules.

[Continue with the CALLMaster Tour](#)

[Return to CALLMaster Tour Overview](#)

[Back](#)

## CALLMaster Tour - Make Recordings

### Wav Recording format for Prompts (PMT), Greetings (GRT) and Messages (VOX)

Recording format for Prompts (PMT folder), Greetings (GRT folder) and Messages (VOX folder) is wav only and must be **11 Khz linear 8bit mono format**.

Speech Recognition recordings required for Barge-in support must be **8 Khz linear 8bit mono format**.

All the prompts shipped with CALLMaster are in wav format. CALLMaster only support wav format files.

You can now:

[Record Audiotex Messages](#)

[Record System Prompts](#)

[Continue with the CALLMaster Tour](#)

[Return to CALLMaster Tour Overview](#)

[Back](#)

## CALLMaster Tour - Record Audiotex Messages

To make these recordings, we are going to have to return to the administrator mailbox. Start up CALLMaster and log in to the System Administrator Mailbox - [Log in to CALLMaster](#).

- Log in as the administrator and press star (\*). The goal is to record Audiotex records by selecting option eight (8). Just follow the prompts.
- In the first record, (Record 10) we are going to tell the callers their choices. They have to know what they can press, right? In record 10, use the recording option in the administrator menu to record the following...
- 'For information on our company's product, press 1; to receive a fax of our company's price list, press 2; or press any other key to end your call.'
- Note: After the prompt is played, and if there is some sort of response length, the system will automatically tell the caller to 'Press star to repeat' (prompt 245).
- Next, we should record information on your company's product. Record module 100 as the company's product information. Say whatever you like here, as we are just making an example.
- The next recording we should make is in record 110, where we tell the caller to 'Press one (1) to return to the main menu, or press any other key to end your call.'
- Once you have made these recordings, head back to the Audiotex menu by pressing pound (#) several times, entering the 'Databases' menu, etc.
- In record 10, we offer callers a choice of pressing 2 to receive a fax. This starts in record 200, as we indicated in the Touch Tone Table of record 10. Now we will start using some CALLBasic.

[Back](#)

## CALLMaster Tour - Record System Prompts

- Call CALLMaster just as you did in the 'Log in to CALLMaster' section at the beginning of this tour. You'll notice that the greeting is a little different. Then you will hear prompt 505, the Attendant Main Menu. The opening prompts are different because we set the phone port to 'Attendant' and changed the greeting ID from 19, 20, and 21 to 502, 503, and 504. These are system prompts that can be re-recorded from the Administrator's Mailbox. We should enter your opening greeting first. Log in to the Administrator Mailbox just as you did in the 'Log in to CALLMaster' section. Press star (\*) to log in; enter the Administrator's mailbox number and the password, and press star (\*) to complete the log in. CALLMaster will tell you how many messages you have. Now press the star (\*) key to access the administrator menu. **You will not be prompted to press star (\*) here** because we want the administrator options to be as secure as possible.
- Option seven (7) allows you to record system prompts. It will ask you for the prompt number. Enter 502 and press the star (\*) key. It should say the prompt as is to you, 'Good morning, you've reached the CALLMaster attendant,' and prompt you to record it. Change this to 'Good morning, you've reached (your company).' Don't record your menu choices here. That's for the main menu, prompt 505.
- Press star (\*) when you have finished recording. Make sure to do this quickly so there is no unnecessary silence at the end of the prompt. Once you have done so, press 1 to accept it, 2 to hear it, or 3 to try again. Press 2 and listen. If you are satisfied, press 1 and we can go on. If not, press 3 and re-record the prompt.
- Once you have accepted the recording, you will be prompted to enter another prompt number. Repeat the process and record prompts 503 and 504 for the afternoon and evening, respectively.
- Now let's record the attendant main menu, prompt number 505, which gives callers their choices. Repeat the recording process and record 'For an operator press zero, or hold on for assistance.'
- Once you have accepted (and therefore saved) this recording, exit by pressing the pound (#) key several times. **The pound (#) key is your escape key. Anytime you are somewhere you don't want to be, enter the pound (#) key to exit.**
- Call CALLMaster and listen to your greeting. If you have done everything right, you should hear your morning, afternoon, or evening greeting and then your main menu. Press '0' for an operator and watch what happens. You should hear prompt 501, which is 'Please hold while I transfer your call'. You will then see CALLMaster attempt to perform the transfer. It will perform a flash-hook (this is set in the Attendant parameter 'Call Transfer String'), plus the number to dial. On the screen you will see 'Dial &,' then 'Call 100.' Assuming that your phone system lets you dial flash-hook and 100, the transfer will take place. Try out the no answer and the busy supervisions as well. Watch the screen to see what CALLMaster does.
- While you have CALLMaster up, call the system and log in to box 1000. From the Attendant Main Menu, you log in to a box the same way you would from the Voice Mail Main Menu. Record the greeting and name for the box, which will be for unanswered operator calls. Notice you weren't prompted to enter the name with touch-tones. That's because the box already knows those digits - we entered them when we created the mailbox. Once you are finished, press the pound (#) key several times to exit CALLMaster.

[Back](#)



## CALLMaster Tour - Create Fax Table Entry

CALLMaster supports both inbound and outbound faxing with purchase of the fax license. A Dialogic board supporting fax is required.

### Inbound Faxing

- Faxes can be received in on any CALLMaster port connected to the board. Any CALLMaster application type will receive in faxes. Faxes can be received in to the default mailbox or caller can select an individual mailbox to receive fax. If the mailbox receiving the fax has an email address and Windows Messaging is enabled, then the faxed document will be emailed as a .tif attachment to the address(es) entered in the mailbox record Email field. See [Inbound Faxes](#) for details.

### Outbound / Fax-On-Demand

Outbound faxing can be done adding a record to the OutcallSchedule table or using the CALLBasic FaxSend command. This chapter will detail steps on how to use the FaxSend command.

- Select 'Faxes' from CALLMaster Manager and using the 'Add' icon [add a new record](#). The 'Fax ID' is going to be 100, so use that in the first field. Remember that we have designated the fax-on-demand to send document number 100. Setting up faxes is simple. Just follow these steps.
- Input a description of the document for reference purposes and is used to help you, the user, keep track of what is being sent in this document ID. Enter 'Product A' in this field.
- Using the Browse button, select the 'FaxFileName'. This is the document you want to fax. This field includes the file path (where the files are located) and, the file name. By default it would be the path \Program Files\CALLMaster or you may choose any other convenient folder. With the file name, be sure to include the extension, if there is one. When we're done here, we will make a file in your \Program Files\CALLMaster directory that is in ASCII format.
- Using the browse button, select the document you want to use as the 'Cover Page'. This tells CALLMaster to find a .txt file to put in the text field on the cover letter. Again, since we are not using the cover letter, leave this blank.
- The transmission of a cover page with each document is optional and controlled by setting field 'Attach Cover Page' to 'Yes' (checked) or 'No' (not checked). We don't really have one to send out, so make sure this field is not checked.
- The 'From Header Text' field, if a cover page is being used, CALLMaster will automatically create it and place this field as the 'from' party. We are not using a cover page, so leave this field blank.
- The 'Suppress Header' field allows you to turn off the fax software feature that prints a line on the top of the fax stating where it's from, then number it's from, etc. It is toggled on and off by this flag. The header information is contained in the System Parameter 'FAXFROM'.
- The File Setting fields include:
  - 'Type' tells CALLMaster what type of file we are sending. There are three types available and these are ASCII, TIF and RAW. See your fax software manual for more information on fax file types. Let's make this an ASCII file (most faxes you will send will be RAW files).
  - 'Text Size' is a field that tells what text size to use when transmitting ASCII files. Leave it in the default and move on.
  - 'Resolution' really shouldn't be tampered with until you know more about the fax software and documents you are working with. It also doesn't pertain to ASCII, so leave it as the default.
- Save this record. That's it! You've created a fax record.
- CALLMaster won't like it very much if it looks for a fax file that isn't there, so make sure you have a document as referenced in the 'Fax File Name' field.
- To send a fax, use the 'FaxSend' command in CALLBasic and specify the document number.

[Continue with the CALLMaster Tour](#)

[Return to CALLMaster Tour Overview](#)

[Back](#)

## **CALLMaster Tour - CALLBasic**

CALLBasic is a powerful scripting language that enhances your ability to interact with the caller via Audiotex. It enables you to compute information, interface with and manage databases, execute third party programs and manage call control. If you are planning on developing interactive applications, please read more about CALLBasic. For sample applications and scripts, refer to [Sample Applications & Scripts](#)

### **End of CALLMaster Tour**

Our tour of CALLMaster is now over. As you get to work with the software, you will discover the many tools you have at your disposal. You'll also learn tricks on how to make your system tighter, better, and more efficient. Remember that if you have problems, you should check troubleshooting and FAQ sections of the individual chapters associated with the different CALLMaster tables. They might answer your questions for you. If not, SpeechSoft Technical Support is there for you!

[Back](#)

# Messaging

CALLMaster Message Management capabilities include:

[Managing Voice Mail](#)

[View Messages](#)

[Email Message/Fax Notification](#)

[Set-Up Notification](#)

[Create Distribution Lists and Distribution List Boxes](#)

You can also setup the mailbox to accept [Inbound Faxes](#).

# VOICE MAIL Overview

[Getting Started With Voice Mail](#)

[Log In](#)

[Voice Mail Main Menu](#)

[Managing Mailboxes](#)

## Voice Mail - Getting Started

If the Port application is set to Voice Mail, a caller will:

1. Be forwarded to the associated Audiotex Start Module (default Module # 2). The caller will hear the time of day greeting as set on the ports table (Greeting File IDs) since the Label field for the default Audiotex Module # 2 contains flag 'PTODG';
2. Hear the main Voice Mail greeting (prompt 10) 'Please enter a box number or press 2 for a directory. Press \* to log in or hold on to leave a message'. If you have the fax/voice board and wish to receive inbound faxes, you may wish to modify this prompt to include 'If you wish to send a fax, press start on your fax machine now'.
  - a. If the caller enters a mailbox number, he/she will hear the greeting recorded by the mailbox owner. The mailbox owner's greeting can prompt the caller to:
    1. Do nothing and leave a message at the end of the greeting; or
    2. Press 2 during the greeting for beeper notification. If option 2 is selected then:
      1. CALLMaster will play prompt 321 'Enter the phone number then press star'. The System Administrator needs to record this prompt;
      2. A beeper type notification record must be created for use of this option;
    3. Press 3 during the greeting to receive fax document to mailbox. (Only available with purchase of CALLMaster Fax Software License and Dialogic Fax Board). If option 3 is selected then:
      1. Fax document is saved in the CALLMaster \Fax\Received folder using the mailbox number subfolder.
      2. The MailboxActivityLog and the FaxReceived tables will record the activity if the Activity Log tables are enabled. [Learn more.](#)
    4. Press 4 during the greeting to transfer to another number. If option 4 is selected then:
      1. CALLMaster will transfer caller to Attendant record whose ID is specified in mailbox Settings field (EXTattid – see Settings field for details).
  - b. If the caller selects 2 for a directory, he/she will be prompted to key in some or all of the person's last name for a matching entry in the voice mail directory. This directory listing is created based on the last name field or as keyed in during the set personal options menu;
  - c. The caller can press star to [log in](#); or
  - d. If the caller does not choose any of the options above, then CALLMaster will direct caller to the leave a message for or fax document to the Default Mailbox set-up in the Voice Mail Parameters.

## Email Notification of Messages or Faxes

If you have activated email in CALLMaster (see [Activating Email](#)) and the mailbox receiving the message or fax has been set-up with an address in the [Email](#) field, CALLMaster will automatically send an email to that address. The email will be sent as defined by the Mode set in the System Parameter #30 - 'MAPI Profile'.

## **Voice Mail - Main Menu**

The Mailbox Main Menu lists the basic mailbox functions: one (1) to review messages, two (2) to send a message, three (3) to set personal options or (4) to review faxes. If the mailbox Type is Announcement then you will only be played the Set Personal Options menu.

Press the number of your choice, or, press the pound key (#) to exit the Mailbox Main Menu and return to the system main menu. If you take no action, the menu will repeat as many times as specified by the system parameter 'Prompt Repeat Count' and then warn you before disconnecting.

[Review Messages](#)

[Send Messages](#)

[Set Personal Options](#)

[Review Faxes](#)

[Make Outbound Transfers](#)

[Administrator Options](#)

## Voice Mail - Main Menu - Review Messages

### Overview

'Review Messages' is the voice mail feature you will probably use most. CALLMaster gives you complete and precise control over message playback, selection, and management. You may control message playback by pausing, rewinding, or fast-forwarding. You may choose to hear the first message or the last message. There are powerful message management capabilities including: delete, save, time-stamp, and reply. As a default, messages will be played back starting with the oldest first. If you want 'new' messages to be played first, in order of receipt, then set the Mailbox Settings field to 'NMF'. You may also review 'saved' or 'deleted' messages. While listening to a message, you can press \*\* to transfer to the called party via the message's caller id if available or by entering a number. Not available for 'Announcement' Type mailboxes.

See [Features](#) for detail message review capabilities

## Voice Mail - Main Menu - Review Messages - Features

The basic Review Messages menu controls message playback, selection, and management. All of the options are available while you are listening to a message; all but the first three are available between messages. If the Mailbox is set for 'Browse Only' (Settings field has code 'BRS'), then options 1, 2, 3, 4, 5, 6, \* & # are available. You can always exit this menu by pressing the pound (#) key.

During message review, you can press 87 to slow playback, 88 to restore default and 89 to playback faster. (Currently not supported on Springware boards for wav files).

Option	Description
1	<a href="#">Rewind</a>
2	<a href="#">Pause</a>
3	<a href="#">Fast Forward</a>
4	<a href="#">Previous</a>
5	<a href="#">Replay</a>
6	<a href="#">Next</a>
7	<a href="#">Save</a>
8	<a href="#">ReplayMenu</a>
9	<a href="#">Delete</a>
0	<a href="#">Reply</a>
*	<a href="#">Additional Options</a>



## Voice Mail - Main Menu - Send Messages

### Overview

Mailbox owners can easily send messages to other mailboxes, to private and public distribution lists by pressing two (2) from the main menu. Not available for 'Announcement' Type mailboxes.

After pressing two (2) from the main menu, you are able to send a message to:

- Another mailbox (specified by the Mailbox #)
- A personal distribution list (by pressing \*)
- A public distribution list (by pressing \*\*)

CALLMaster checks for valid box or distribution list numbers. You will then be asked to record your message. Once you have finished recording the message, press the star (\*) key. You will then be given the option to press \* if you wish to send this message to another box. You can then enter the individual box number or private/public distribution list as required.

See [Features](#) for detail on send message capabilities

## **Voice Mail - Main Menu - Send Messages - Features**

You can select a single mailbox number or a private or public distribution list to send the recorded message.

[To Box Holder](#)

[To Distribution List](#)

[Recording The Message](#)

## Voice Mail - Main Menu - Set Personal Options

### Overview

Press three (3) from the mailbox main menu to set your personal options for your mailbox, such as password and greeting.

See [Features](#) for detail message review capabilities

## Voice Mail - Main Menu - Set Personal Options - Features

There are seven personal options you can set: password, name, greeting, distribution lists, directory listing, notification on/off, and notification settings.

To make only the first three options available to a mailbox, set flag 'UL3' in the mailbox's Settings field, or in the 'Default Flags' field of the Voicemail Parameters to limit options for all mailboxes. You may want to rename prompt PMT\1\6b.wav to PMT\1\6.wav for prompt to only offer caller the three options.

Options	Description
1	<a href="#">Change Password</a>
2	<a href="#">Record Greeting</a>
3	<a href="#">Record Name</a>
4	<a href="#">Maintain Distribution Lists</a>
5	<a href="#">Change Directory Listing</a>
6	<a href="#">Toggle Notification</a>
7	<a href="#">Change Notification</a>
8	<a href="#">Record Audiotex Module</a>
9	<a href="#">Edit Attendant Record</a>
#	<a href="#">Exit</a>

## Voice Mail - Main Menu - Review Faxes

### Overview

'Review Faxes' enables you review new fax headers received, fax back now by forwarding to the fax machine you are currently calling from, fax to another fax machine, save or delete the document. You may also review 'saved' or 'deleted' faxes.

See [Features](#) for detail message review capabilities

## Voice Mail - Main Menu - Review Faxes - Features

The basic Review Faxes menu enables fax management.

Option	Description
1	<a href="#">Fax Back Now</a>
2	<a href="#">Fax to Another Number</a>
4	<a href="#">Review Previous Fax Header</a>
5	<a href="#">Review Current Fax Header</a>
6	<a href="#">Review Next Fax Header</a>
7	<a href="#">Saved Fax</a>
9	<a href="#">Delete Fax</a>

## Voice Mail - Main Menu - Make Outbound Transfers

### Overview

This function is used to allow the mailbox owner to make outbound calls only if this function has been enabled by the administrator and if your phone network supports transfer/bridging capabilities.

See [Features](#) to make outbound transfer capabilities

## Voice Mail - Main Menu - Make Outbound Transfers - Features

The administrator can enable this function for all mailboxes by including "OTE", Outbound Transfers Enabled, in the System Parameters System Settings field. This function can also be enabled by individual mailboxes by including "OTE" in the Mailbox Settings field.

Once enabled, the administrator then has to set the transfer mode. By default transfers are flash hook blind transfers, but they can be 3-Way (OTT3), or Bridged (OTTB) calls based on this flag. This flag would normally be set in the System Parameters System Settings field as it would apply to all enabled mailboxes.

The transfer mode flags can be set as follows:

- OTTB<port-range> - where port range is the ports to use for the bridged call. See Bridged attendant transfers for details
- OTT3<3waystring>- where 3way string is used to create and drop a 3way call
- OTPS - Outbound Transfer Prefix Dial string
- OTSS - Outbound Transfer Suffix Dial string

The following prompts are used for this function and may need to be re-recorded:

- 398"Enter the phone number."
- 399"Hold while you are transferred."
- 400"Connection removed" (for bridged calls.)

Once transfer is terminated, caller is returned for more voicemail options.



## Voice Mail - Main Menu - Administrator Options

### Overview

The system administrator maintains the CALLMaster system, making sure speech files, databases, and configuration settings are correct, monitoring hardware and phone lines, and performing routine maintenance on the hard disk.

The administrator or administrators can perform certain functions from a touch-tone phone through one or more system administrator mailboxes. CALLMaster comes with a Mailbox record whose Mailbox # is '0000', Password is '0000' and Type is 'Administrator'.

Password security is critical in an administrator mailbox: Change the password! Those who are very security conscious should create a new administrator box and change the 0000 box to a 'user' or 'announcement' mailbox. **Do not delete box 0000**, because CALLMaster will recreate it with administrator privileges. Neither should you change its status to 'deactivated' because that deactivates the entire system. Keep in mind that box 0000 is the default voice mailbox unless you change the Voice Mail Parameter 'Default Mailbox'.

To use the [Administrator Option Features](#), you need to [Log in to CALLMaster](#).

## Voice Mail - Main Menu - Administrator Options - Features

The Administrator menu allows you various options to create, delete, and modify boxes, change default settings, check the status of your CALLMaster system, record any system prompt or Audiotex module, and record names and special messages for Attendant extensions.

Option	Description
1	<a href="#"><u>Create A Box</u></a>
2	<a href="#"><u>Delete A Box</u></a>
3	<a href="#"><u>Change Box Settings</u></a>
4	<a href="#"><u>Box Status</u></a>
5	<a href="#"><u>Change New Box Defaults</u></a>
6	<a href="#"><u>System Status</u></a>
7	<a href="#"><u>Record Prompts</u></a>
8	<a href="#"><u>Record Audiotex Messages</u></a>
9	<a href="#"><u>Record Attendant Directory</u></a>
0	<a href="#"><u>Record Barge-In Prompts</u></a>

# Managing Mailboxes

## Overview

Mailboxes may be created on-line via touch-tones or by using the CALLMaster Manager.

Use the [CALLMaster Manager | Manage | Mailboxes | Single](#) function to manage a single mailbox.

Use the [CALLMaster Manager | Manage | Mailboxes | Range](#) function to manage a range of mailboxes.

## Managing Mailboxes - Single

You can manage single Mailbox records via the **CALLMaster Manager | Manage | Mailboxes | Single** function.

Click on desired field for explanation.

The screenshot shows the CALLMaster Manager interface for managing a single mailbox. The window title is "CALLMaster Manager - [Mailboxes]". The top menu bar includes File, Manage, View, Parameters, Windows, and Help. The status bar indicates "CALLMaster Service Status: Running".

The main content area is divided into two sections. On the left, under the heading "Mailboxes", there is a table listing mailboxes. The table has columns for Mailbox, Last Name, and First Name. The row for Mailbox 1001, Anne DeSoto, is highlighted.

Mailbox	Last Name	First Name
0000	Patton	John
1000	Smith	Jason
1001	DeSoto	Anne
1002	Hollander	Trudi
1003	Stone	Frank
1004	Newman	Jonathan
1005	Belize	Joan

On the right, the configuration details for Mailbox 1001, Anne DeSoto, are displayed. The fields include:

- Mailbox #: 1001
- Password: \*\*\*\*
- Last Name: DeSoto
- First Name: Anne
- Email: ADesoto@smail.com
- Settings: LMM DTF
- Type: User

Below these fields, there are sections for Message Storage, Notification, and References.

**Message Storage**

Allowance Max	Current Count
New: 50	New: 0
Saved: 50	Saved: 0
Length: 300 sec(s)	
Time: 30 day(s)	

**Notification** - On

- Enabled ☒
- Notify per Msg ☒
- Notify Delay: 0 min(s)
- Message Lights ☐
- Current Event ID: 0
- Notification Record ID: 0

**References**

- Audio Recordings: ☒ locked
- Greeting File ID: 0
- Directory Listing: 337686
- Name File ID: 0
- Use: ☐ First Name ☐ Last Name ☐ None

At the bottom, there are navigation buttons: < Previous, Save, Undo Changes, Next >, and Exit.

If you have checked off the "Show MB Xtra" on the [Xtra Parameter screen](#), see [Mailbox screen with Xtra fields](#) for details.

## Managing Mailboxes - Range

You can manage a range of Mailbox records via the **CALLMaster Manager | Manage | Mailboxes | Range** function.

Enter the 'Add Mailbox Range' option, enter the 'Start Box No', the 'End Box No' to define the range, the 'Increment By' value and the default values you want applied to all the mailboxes added in the selected range.

Click the 'Update' button to add the mailboxes in the selected range.

The Mailbox numbers will be incremented using the 'Increment By' value. For example, if you set the 'Start Box No' = 2000, the 'End Box No' = 2020 and the 'Increment By' value = 5, you will add 5 mailboxes (2000, 2005, 2010, 2015, 2020) to the Mailbox table.

**NOTE:** Mailbox Password will be set to the Mailbox Number.

## Frequently Asked Questions - Mailboxes

### Q: Where are my messages stored?

**A:** By default, messages are stored in the C:\Program Files\CALLMaster\VOX folder. However this setting is defined in the System Parameters "Prompt folder", and can be changed. Each message a mailbox receives is logged to the CALLMaster Messages table. You can view the message activity from the **CALLMaster Manager | Manage | Messages**. You can convert the Speech File Id from the Messages table to the actual message file stored in the VOX folder by applying the algorithm specified in Appendix A.

### Q: I forgot my password. How do I get back into my mailbox?

**A:** The system administrator has password information.

### Q: Why do I get messages when the person just hangs up?

**A:** Once the beep is played so the caller can leave the message, the recording will be in the box. You can set the minimum message length to avoid very small messages. CALLMaster also supports the "Option to leave a message" feature, which asks the caller if they would like to leave a message in the box before actually doing so.

**Note:** If you have activated "allow log in during greeting" for your mailbox and the user presses the pound (#) to escape without leaving a message, a log in option will be given to the caller.

### Q: Once the person hangs up, why is there a long silence or tone signal before CALLMaster comes on again?

**A:** Most phone systems or Telephone company lines send a disconnect tone or (line) loop current drop which will terminate the call and recording. To terminate the recording based on a disconnect tone, sometimes a dial tone is used, the tone must be defined in the CALLMaster Global Tone table. You can run the Dialogic utility PBXpert to analyze and define the tones received and then use that information to define a disconnect tone, see Global Tones in Chapter 10. Contact your Telco or your phone system vendor to make sure a disconnect signal is sent.

If disconnect signaling is absent then there is a system parameter called "Record Maximum Silence" that determines the amount of time silence will be recorded before the system stops recording. The default is 15 seconds, which can be set lower. Be careful not to make it too low, as people do tend to be silent when they are thinking.

### Q: Can I manage mailboxes, system recordings and settings remotely?

**A:** Yes, if you can log in to CALLMaster as an administrator press star when you get the mailbox main-menu "Press one to Review Messages, two to Send or three for Personal Options" you will then get the Administrator's main menu see above chapter for details.

### Q: How do I set the allowable length of a mailbox greeting?

**A:** The maximum message length also applies to the maximum mailbox greeting.

### Q: Can I send one message to multiple boxes?

**A:** Yes - read up on distribution lists, in this chapter.

**Q: Can I store multiple greetings for my box and interchange them when I need to?**

**A:** The default answer to this is no. However, if you feel you're comfortable enough to use CALLBasic you can see the chapter entitled "Sample Applications" for a way to do this through Audiotex.

**Q: How does Email Notification work?**

**A:** When properly configured for email messaging, CALLMaster will send out an email message to the designated email address for the individual mailbox with an audio file of the actual message as a .WAV format attachment. By entering a dash "-" as the first character of your email address, you can choose to be only notified by email without have the actual message emailed to you as an attachment. See [Activating Email](#) and [Email Message/Fax Notification](#) for details.

**Q: How do I set-up for multi-lingual voice mail?**

**A:** CALLMaster can support applications in multiple languages as defined and set-up by the system administrator. Callers can then be given the option to select the language from the defined language set.

The language set has to be defined in the System Parameters, the multi-lingual recording need to be made and the caller's language selection needs to be processed via Audiotex CALLBasic scripts. For details see [Multi-Lingual Support](#).

## Troubleshooting - Mailboxes

**Problem:** The maximum message length in your box is 360 seconds, but when you send to a public distribution list you are cut off after 30 seconds.

**Cause:** Message length in New Account Defaults is only 30 seconds.

**Solution:** Public distribution list message lengths are determined by the '**Max Message Length**' in the New Account Defaults section of the System Parameters. Change that message length to the desired length.

**Problem:** You get hang ups in your mailbox. Shouldn't CALLMaster detect those?

**Cause:** The person hung up after the mailbox greeting finished and the record tone started.

**Solution:** You might consider telling people they can cancel by pressing the pound (#) key. If the recording tone sounds, anything after that until the system receives a disconnect tone or times out will be in your mailbox. Bear in mind that if you are using Allow Log In During Greeting, telling the person to press pound (#) key will offer them a chance to log in to your mailbox. Instead, tell them to hit pound (#) twice if you're using the Allow Log In During Greeting feature of the mailbox.

**Problem:** I received a message in my mailbox, which has notification. I didn't get notified.

**Cause:** Either you only get paged after the first new message or your notification record/phone ports are not set up correctly for notification.

**Solution:** First, if you want to be notified after every message that enters your box, make sure your Message Notification Settings are set up for 'Notify Humans/Pagers' and 'Per Msg'. You should check the validity of your Notification record, as well as your Phone Port record. Make sure you have the Notification record set-up with the correct activation days and times, phone number, area code and dial prefix if one is required. Additionally, make sure that the System Parameters for Long Distance Dialing Prefix, Local Dialing Prefix are set correctly. Lastly, check that at least one port is set for Outbound Notification. For more on this problem, see the chapter entitled *Out Dial & Notification*.



# Notification

## Overview

Message notification is a mailbox privilege assigned by the system administrator. It allows the box holder to be notified if his mailbox receives a message and/or to have message waiting light turned on/off. A box holder who has message notification may use touch-tones to turn notification on or off, change the phone number where to be notified, and change the earliest and latest time to be notified. A mailbox may have many notification records if notification type is set to Human or Pager Text; in that case, after the first record is exhausted, the next record will be used in order of record number. If a notification sequence is in progress and a new message is received, by default a new notification sequence will not be initiated. Notification will schedule a call regardless if a notification sequence is in progress only if the Mailbox Settings field contains 'NRE' (Notification Refresh Entries). If NRE is used then the current sequence is cancelled and a new one started.

If notification type is set to Human or Pager Text, CALLMaster will notify using multiple notification records based on the call schedule. For all other notification types, CALLMaster will only attempt to notify once using a single notification record.

Here is the scenario that the notification scheduler follows:

- If you only have one notification record then CALLMaster will keep retrying during the notification time permitted. E.g. if you have a record to notify M-F 9am – 5pm and the mailbox receives a message at noon Thursday, the system will keep trying to notify at the interval set even if the # of retries is set to zero until Thursday at 5pm. It will then schedule a notification for Friday at 9:00am. Notification stops once the mailbox owner logs in to his/her mailbox.

Mailbox	Record No.	Notification Type	Area Code	Phone Number	Earliest Hour to Call	Latest Hour to Call	Days to Call (enable days below)	Call Attempt Schedule
1111	1	Human	914	5551111	9	17	M, T, W, Th, F	No Answer - Retry 2 times every 20 minutes Busy - Retry 2 times every 5 minutes.

- If you have several notification records, all of which are available for notification during the same period then CALLMaster will keep rotating thru all records. E.g. If you have 3 notification records, all set for notification M-F 9 – 5 and the mailbox receives a message at noon Thursday, then CALLMaster will first use record 1 then try record 2 then record 3 and then back to trying record 1 and so on till 5pm Thursday. CALLMaster will set the next available record for notification the next day at 9am. Each record will be tried the number of times specified and based on the Call Attempt Schedule specified.

Mailbox	Record No.	Notification Type	Area Code	Phone Number	Earliest Hour to Call	Latest Hour to Call	Days to Call (enable days below)	Call Attempt Schedule
1111	1	Human	914	5551111	9	17	M, T, W, Th, F	No Answer - Retry 2 times every 20 minutes

								Busy - Retry 2 times every 5 minutes.
1111	2	Human	212	5552222	9	17	M, T, W, Th, F	
1111	3	Human	914	5553333	9	17	M, T, W, Th, F	

- If you have several notification records with a schedule then CALLMaster will follow the schedule. E.g. Record 1 scheduled for notification M-F all day (0-0) and Record 2 M-F 5pm to 9am. If the mailbox receives a message at noon Thursday then CALLMaster will try to notify using Record 1 and then schedule notification for 5pm Thursday using Record 2. Each record will be tried the number of times specified and at the interval specified before proceeding to the next record.

Mailbox	Record No.	Notification Type	Area Code	Phone Number	Earliest Hour to Call	Latest Hour to Call	Days to Call (enable days below)	Call Attempt Schedule
1111	1	Human	914	5551111	0	0	M, T, W, Th, F	No Answer - Retry 2 times every 20 minutes Busy - Retry 2 times every 5 minutes.
1111	2	Human	212	5552222	17	9	M, T, W, Th, F	

If you have a single notification record but you do not want it to call you every minute, then increase the Call Attempt Schedule interval or create a second notification record as type pager with a dummy call number. The system will try record 1 and then record 2. As record 2 is type = pager CALLMaster will only attempt notification once and then stop.

Mailbox	Record No.	Notification Type	Area Code	Phone Number	Earliest Hour to Call	Latest Hour to Call	Days to Call (enable days below)	Call Attempt Schedule
1111	1	Human	914	5551111	0	0	M, T, W, Th, F	No Answer - Retry 2 times every 20 minutes Busy - Retry 2 times every 5 minutes.
1111	2	Pager	914	1234567	17	9	M, T, W, Th, F	

[Managing Notification](#)

[How To Activate Notification](#)

[Frequently Asked Questions](#)

[Troubleshooting](#)

## Activating Notification

### Create the desired notification record(s) for each mailbox.

Set each Notification record with the proper schedule times and days activated. This is done using the CALLMaster Manager through **Manage | Notification**. For message light notification, there must be a record zero (0) with 'Notification Type' of 'Message Light' and every day of the week set to 'Yes'. For Pager/Human Notification, there must be a record one (1) with the correct 'Notification Type', with the 'Earliest Hour to Call' and the 'Latest Hour to Call' set and every day of the week set to 'Yes'.

### Set one or more phone ports in the 'Phone Ports' Database to Outbound to allow notification.

Ideally this would be your least used (highest numbered) port(s). It is advisable to enable 'Make Outbound Calls' Setting for at least one port. If none of the ports have their 'Make Outbound Calls' Setting enabled, then CALLMaster will NOT make the notification.

### Check 'Notification Enabled' for either message waiting lights or human & pager notification.

This setting is in the 'Mailbox' Database, in the 'Notification' section. Set Notification 'Enabled' and check the 'Notify per Msg' indicators for Human and Pager Notification (A box holder can toggle these notifications on or off from his mailbox). For message waiting light notification, check the 'Message Lights' option. The System Parameters called 'Msg-Light-On Per Msg' and 'Msg-Light-Off Per Login' control whether the message light is activated after every message, and whether it is extinguished after every log in.

### Set the necessary System Parameters.

For human, beeper, or pager you should look at the System Parameters 'Local Dialing Prefix', 'Long Distance Dialing Prefix', 'System's Area Code', and 'System's Phone Number'. For message light notification you should look at 'Msg-Light-On Dial String', 'Msg-Light-Off Dial String', 'Msg-Light-On Per Msg', and 'Msg-Light-Off Per Login'.

**Note:** If the System Administrator has set NTFYOFF flag in the system parameter 'System Settings' field, then notification has been disabled for the system. This flag does not control the message waiting lights.

## Managing Managing Notification

Click on desired field for explanation.

CALLMaster Manager - [Notification]

File Manage View Parameters Windows Help

CALLMaster Service Status: Running

Mailbox 1001 Record # 1 Type Human

### Notification

Mailbox	Rec #	Prefix	Area Code	Phone Number	Type
0000	1		212	1110000	Human
1001	1		212	2011001	Human
1001	2		212	2011002	Human
1001	3		914	2221212	Human
1002	1		516	5551111	Human
1002	2		914	5552222	Page Text

Mailbox: 1001  
Record #: 1  
Cal Type: Human  
Dial Prefix:  
Area Code: 212  
Phone Number: 2011001  
Send Tones:

#### Call Retry Schedule

If call failed due to:

No Answer, retry: 3 times, every: 5 min(s).  
Busy, retry: 3 times, every: 5 min(s).  
No Login, retry: 3 times, every: 5 min(s).

#### When To Call

Monday: ☒ Friday: ☒ Earliest Hour: 8  
Tuesday: ☒ Saturday: ☐ Latest Hour: 20  
Wednesday: ☒ Sunday: ☐ Always page if both = 0  
Thursday: ☒

#### Call Attempt Status

Last Time Called: 0 result: 0 tried: 0 times.  
0 - 23 hrs

< Previous Save Undo Changes Next >  
Exit

+ 2 of 6 X

## **Frequently Asked Questions - Notification**

### **Q: Why does the mailbox get a message and yet notification does not take place?**

**A:** Read the section on how 'To Activate Notification'. Retrace the steps outlined and make sure your Mailbox settings, Notification Record Settings, Ports Record and the System Parameters are set correctly for notification.

### **Q: How do I keep paging if a box holder doesn't log in to their mailbox?**

**A:** The 'Call Attempt Schedule' fields deals with this problem, specifically the 'If log in fails retry' count.

### **Q: Can I have the caller enter the pager number to be called?**

**A:** Yes, the mailbox greeting can direct the caller to press 2 to be paged. CALLMaster will then ask the caller to enter their call back number. A notification record with type = Beeper is required for this functionality. You can also do it via CALLMaster IVR scripting capabilities. You would have Audiotex set-up to get the pager number from the caller and then do a Dial command to connect to the pager.

### **Q: Can I change settings for my notification from remote?**

**A:** Yes, as long as the administrator has set-up a notification record for your mailbox. As a regular user you can change notification indicator on or off, change the phone number for notification and the earliest and latest hour to notify. You cannot, however, add a notification record if one does not already exist.

### **Q:What happens if the box gets a message during the hours when notification is off? Does it start notifying as soon as notification is on again?**

**A:** Yes, it does. CALLMaster will not make a notification attempt outside the Earliest and Latest hours to notify. However, it will store the request for later notification. If you have multiple notification records, put the record with the lowest availability last. The reason for that is if the first record is invalid, CALLMaster will not go on. For example, if record 1 is valid from 1-3 pm, notification will only be activated from 1 to 3 pm. The first record must be valid and then exhausted before moving on.

## **Troubleshooting - Notification**

**Problem: The message waiting light won't light when a new message is left in a box.**

**Cause:** Missing or incorrect notification record. Another cause is an incorrect message light on/ off code in the system parameters.

**Solution:** Correct or create the record. If you can light the light from an analog extension or phone, so can CALLMaster. If you continue to have trouble, check the phone handsets themselves, and check that the mailbox has an Active Notification Id.

**Problem: When you leave a message, notification doesn't take place.**

**Cause:** Mailbox not set-up with proper 'Message Notification Settings', or incorrectly setup notification record.

**Solution:** Make sure that the mailbox 'Message Notification Settings' are set correctly. Next, make sure that the notification record exists and that all fields are set properly. Check system parameters for the long distance and local dialing prefix. You may need a certain dial prefix or other additional items.

**Problem: You have different pager companies with different paging cadences.**

**Solution:** If the problem lies in just a few pagers, alter the notification number fields with commas to adjust for the added time. Notification calls with Type Pager, Pager Mailbox, Pager Text or Beeper, are unsupervised transfers with a # key to terminate the call.

# DISTRIBUTION LISTS AND DISTRIBUTION LIST BOXES

## Overview

Distribution Lists enable a mailbox holder to broadcast the same message to group of mailbox owners assigned to the list. A mailbox owner can have many distribution lists, with the same mailbox assigned to multiple lists.

A mailbox owner can create multiple Private Distribution Lists as well as one Auto Distribution Lists. The System Administrator can create multiple public distribution lists as well as one Global Distribution Lists.

[Managing Distribution Lists](#)

[Managing Distribution List Boxes](#)



# Managing Distribution Lists

## Overview

Voice mailbox holders can use distribution lists to send the same message to a group of box holders. Distribution lists can be private or public. To forward messages to the mailboxes in a distribution list, the sender must enter the distribution list number. The Distribution Lists database contains the lists themselves; the mailboxes on the lists are in another database file (See Distribution List Boxes).

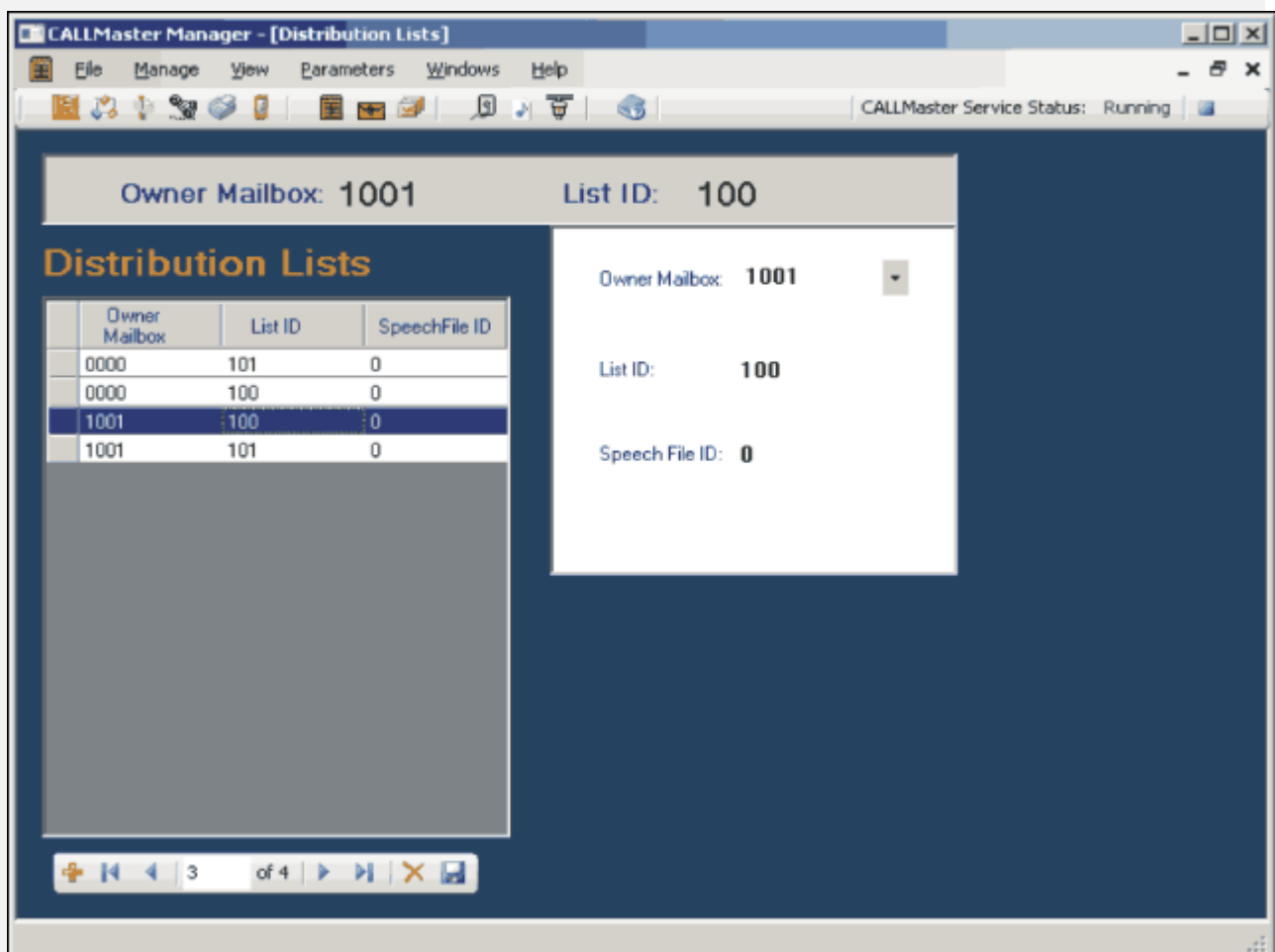
There are **2 reserved distribution lists**:

1. **List 990 (Global List)** – if selected, will forward messages to ALL mailboxes in the system, including a copy to itself. You do not need to add mailboxes to this Distribution List as it will send to ALL mailboxes in the system. **Only available as a public list created by the Administrator's Mailbox. Prompt 320 should be recorded with the name 'Global List'**
2. **List 999(Auto Send List)** – will **automatically** send messages to all boxes on this mailbox owner's distribution list. Each mailbox can have its own distribution list 999.

Public Distribution Lists must be created via the touch-tone functionality and cannot be created using the CALLMaster Manager on-line screens. Any lists made by the system administrator will be public lists to be used by any mailbox holder.

Private lists are made by a box holder and can be used by the owner alone.

Click on desired field for explanation.



# Managing Distribution List Boxes

## Overview

The Distribution List Boxes database tells which mailboxes are on which distribution lists. You do not need to assign any mailboxes to the Global List 990 as it will send messages to ALL mailboxes in the system.

Click on desired field for explanation.

The screenshot shows the CALLMaster Manager - [Distribution List Boxes] window. The window has a menu bar with File, Manage, View, Parameters, Windows, and Help. Below the menu bar is a toolbar with various icons. The status bar at the bottom right indicates CALLMaster Service Status: Running.

The main content area is divided into two sections. The top section displays the current selection: Owner Mailbox: 1001, List ID: 100, and Mailbox: 0000. Below this is a table titled "Distribution List Boxes".

Owner	ListID	Mailbox
0000	100	1001
0000	100	1002
0000	100	1000
1001	100	0000
1001	100	1000
1001	100	1002
1001	101	1003
1001	101	1004
1001	101	1005

The bottom section of the window contains a form for editing the selected entry. The form has three fields: Owner, List ID, and Mailbox. The Owner field is set to 1001, the List ID field is set to 100, and the Mailbox field is set to 0000. Each field has a dropdown arrow next to it.

# Voice Mail Messages

## Overview

Messages are a collection of Audio recordings left for a mailbox.

Messages can be saved, deleted, forwarded to another mailbox, forward to a group of mailboxes in a distribution list, emailed on receipt. See [Managing Voice Mail](#) for details.

Messages are maintained in new status until the mailbox owner saves or deletes the message.

New and Saved messages are deleted once they reach the [Allowance Age](#) set for the mailbox. When the mailbox owner 'deletes' a message during Mailbox Review, the message will only be deleted after the box owner logs off.

## Managing Messages

The Message table is used by the system to track messages in new and saved status. You may only view its contents.

Messages are maintained in new status until the mailbox owner saves or deletes the message.

New and Saved messages are deleted once they reach the [Allowance Age](#) set for the mailbox. When the mailbox owner 'deletes' a message during Mailbox Review, the message will only be deleted after the box owner logs off.

**Click on desired field for explanation.**

### Maximum Message Age

The maximum number of days a message may be kept on the system before it is removed automatically.

The message age pertains to both new and saved messages. The maximum age a message can have is 999 days, which is 2 years and 269 days.

# Attendant

## Overview

CALLMaster Attendant has the capability to transfer a call from one line to another, or to an extension. The purpose of the Attendant is to route callers to extensions or departments, voice mailboxes, alternate messages, or Audiotex. Even when telephone equipment is simple, or non-standard, the Attendant still provides a top-level menu for Voice Mail or Audiotex.

The Attendant begins by presenting various options to the caller through prompt 505, the Attendant Main Menu. Then it monitors for touch-tones and searches the Attendant database for a record with Attendant Id matching the caller's touch-tone input. CALLMaster performs according to the details of the Attendant record and then, unless the call has been transferred or terminated, gives the caller the option to 'press one for more attendant options' and return to the Attendant Main Menu. If CALLMaster does not receive touch-tones during the Attendant Main Menu, it uses attendant id zero (0), which is usually for the operator or receptionist.

Attendant transfers are dependent on the capabilities of your switching equipment and telephone lines. You should understand and test the functions and requirements of the extensions or lines to which CALLMaster is attached before adding Attendant records. To transfer a call with CALLMaster Attendant requires the use of speech processing boards that support call progress detection (e.g. D/4PCI and JCT type Dialogic boards).

## Automatic Speech Recognition (ASR) License

If you have purchased the CALLMaster ASR (Automatic Speech Recognition) capability, then you can present various verbal and touch tone options to the caller through prompt 505, the Attendant Main Menu (e.g. Enter the extension or say the name of the caller you would like to be transferred to"). CALLMaster will monitor for verbal or touch-tone input and search the Attendant database. If verbal input then CALLMaster will search for a record with Attendant Id matching the caller's recording based on the specified XML file "CMAttASRDDirectoryList.xml" in the CALLMaster\Data folder (see Appendix F – Automatic Speech Recognition XML Set-Up Guide). The CALLMaster license file will be activated for ASR if purchased. This will enable ASR capability with CALLMaster for both Attendant and CALLBasic features. If you have purchased ASR license but wish to deactivate usage with the Attendant then set the NOATTASR flag in the System Parameters System Settings field.

CALLMaster ASR supports Barge-In in conjunction with the JCT type Dialogic board. This allows the caller to Barge-In and says the desired party's name while the Attendant greeting is being played. The greeting will need to be made using Administrator option "0" for Record Barge In Prompts. See [Voice Mail – Administrator Options - Record Barge In Prompts](#).

## Fax License

If you have purchased the CALLMaster Fax Software License, you can also receive in faxes during your main greeting. Just instruct the caller in prompt 505 to 'press the start button on his/her fax machine now' and caller will receive the fax document and save in using the default mailbox. The faxed document is saved as a .tif file in the CALLMaster\Fax\Received folder and can be viewed using Windows Imaging Preview software. NOTE:- make sure that your Dialogic board or your IP Gateway/PBX provider supports fax.

For details, see:

[Understanding Call Transfers](#)

[Managing Attendant](#)

[Implementing Attendant](#)

[Attendant Switch Integration](#)

[Frequently Asked Questions](#)



## Understanding Call Transfers

Understanding the sequence of events in an Attendant transfer really helps in setting up transfers and troubleshooting problems.

During transfers, the following things happen:

See attached figure for the complete Attendant transfer sequence. They are listed in numerical order. In steps 8 and 9, we see alternate (a & b) paths that could happen. These are determined by the result of the call; whether it was answered or not.

- In Step 1, we get the digit(s) from the caller. This happens during the main menu (prompt 505).
- In Step 2, we check to see if the entry was valid. If there is a matching Attendant record for the entry, we go to step 3. If not, we tell the caller that the extension was invalid, and give them another chance to continue.
- In Step 3, we check to see if the Type of Transfer is 'Screened'. If not, we proceed to step 4 and skip step 10. If so, we ask the caller for their name (prompt 507).
- In Step 4, CALLMaster tells the caller 'Please hold while I transfer your call ' or 'Please hold while I transfer you to ...'.
- In Step 5, CALLMaster looks to the Attendant parameter 'Transfer String' for the transfer initiation string (See '*Attendant Parameters*' for more details) and activates it.
- In Step 6, CALLMaster dials the number/IP address from the field 'Number to dial', from the Attendant record used.
- In Step 7, the IP Gateway or the Board drivers perform CPA (Call Progress Analysis) on the call to see what the results are. Any problems up to this point can be attributed to faulty records, or improper programming.
- In Step 8, CALLMaster can do one of two things. If there is a busy signal or if the call goes unanswered then the Failed Transfer Action from the Attendant ID record take place (Step 9a.). If there is a connect signal, as in someone picking up the receiver, then we go to Step 9b.
- In Step 9a, we return the caller to the Attendant record's Failed Transfer Action. This could be a mailbox, another extension, or other things.
- In Step 9b, the calling and called parties are connected and the call proceeds.
- In Step 10, CALLMaster (if the transfer was 'Screened') asks the person being called if they want to take the call. If so, they go to Step 11. If not, they go to Step 9a for Screened Reject Fail Action.
- Finally, in Step 11, the call is transferred.

By now you should have tried the transfer with a regular analog telephone and the transfer should have worked. If you tried this and CALLMaster still does not transfer the call properly, you should find out where in this sequence the problem is arising. If it is in the first 5 steps, the problem may be in your CALLMaster programming. Check out the Attendant ID record carefully for proper entries.

**Helpful hint if integrating with a telephone network:** Choosing who will supervise the call is an important decision when integrating CALLMaster with a telephone system. If CALLMaster supervises the call, the Dialogic board will await a human response before sending the call through. The telephone system on the other hand knows instantly when the call is picked up, and can send the call through faster. Although failed calls may take longer depending on your phone systems setup. Having the telephone system perform the supervision can subtract a second or two from the transfer time. To do this, simply tell CALLMaster to perform a 'blind' transfer. The telephone system will now supervise the call. Program the telephone system to send busy or unanswered calls back to CALLMaster along with certain digits that are linked to voice mailboxes or Failed Transfer Action of some sort. To increase the speed of the transfer you could adjust the PAUSE and FLASH times. Other parameters that affect call analysis are in the Dialogic Parameters.

## Managing Attendant Records

You can access Attendant records via the CALLMaster Manager **Manage | Attendant** function.

Click on desired field for explanation.

The screenshot shows the CALLMaster Manager - [Attendant] window. The title bar includes 'CALLMaster Manager - [Attendant]' and standard window controls. The menu bar has 'File', 'Manage', 'View', 'Parameters', 'Windows', and 'Help'. The status bar at the top right shows 'CALLMaster Service Status: Running'.

The main content area is divided into two sections. The left section, titled 'Attendant', contains a table with the following data:

Attendant ID	Time of Day	Type of Transfer	Number To Dial
0	All Day	Supervised	0
0	Evening	Mailbox	0
1	All Day	Screened	101
1	Evening	Mailbox	0
2	All Day	Auditex ...	0

The right section contains configuration options for the selected attendant (ID 1):

- Attendant ID: 1
- Time Of Day: All Day
- Type Of Transfer: Screened
- Number To Dial: 101
- Settings: AES
- Directory Listing:

Below these are three expandable sections:

- Failed Transfer Action**
  - If Call Was:
    - Busy: Mailbox
    - No Answer: Alternate Attendant
    - Screened Reject: Mailbox
- Additional Settings**
  - Alt Attendant ID: 2
  - Mailbox: 1001
  - Auditex Module ID:
  - Retry Attempts: 0
  - When Retry Fails: Default Message
- Audio Recordings**
  - Name File ID: 0
  - Alt Message ID: 0

At the bottom, there are navigation buttons: '< Previous', 'Save', 'Undo Changes', 'Exit', and 'Next >'.



## Implementing Attendant

### Overview

Automated Attendants are most commonly used to route callers to provide information, speak to another party, or leave a message in a mailbox, depending on caller touch tone input. People using the Attendant include outside callers wanting to speak to your company, inside users checking their voice mail, and CALLMaster activating notification of some sort. Prior to setting up Attendant, you will need to know the transfer capabilities of your phone lines and determine the transfer string.

[Determine Transfer Capability of Phone, PBX and Gateway](#)

[Determine Transfer String](#) (Not applicable for IP Network)

[Determine Call Supervision](#) (Not applicable for IP Network)

[Prepare And Test CALLMaster](#)

[Backup CALLMaster Folder](#)

[Complete CALLMaster Configuration](#)

## Determine Transfer Capability of Phone Lines

CALLMaster Attendant requires transfer ability to be provided by your local phone company, your KSU/PBX or your Dialogic board. Various transfer capabilities include:

### Telephone Network

- **Centrex** – provided by your phone service provider. Centrex phone lines from the phone company (also referred to as CO lines) will be connected directly into the Dialogic board installed in the CALLMaster PC. Once the call is transferred or answered by the called party, the phone line that received the call is freed up to answer the next call.
- **3-way Call** - provided by your phone service provider. In this case, the CO lines will be connected directly into the Dialogic board installed in the CALLMaster PC. With 3-way Calling, the phone line that received the call is in use for the duration of the call.
- **KSUs/PBXs** – various switch types used in an office environment to route calls. The CO lines will be connected to the switch with CALLMaster connected to either the analog or digital extensions off your switch. CALLMaster integrates with most types of switches supporting Analog extensions requiring any analog Dialogic board. For digital extensions support, CALLMaster requires the D/42JCT-U or the D/82JCT-U boards for integrations to specific digital switches. The phone line that received the call is in use for the duration of the call. See Appendix D – Digital PBX Integration for details on switch set-up and integration.
- **Channel Bridging** – a feature provided by CALLMaster that requires special Dialogic boards with the SC Bus. With this feature, the call is answered on one line and bridged to a second line to connect the calling and called parties, requiring 2 lines to be in use for the duration of the call.
- **Extension Side T1** - provided by your T1 service provider. Similar set-up to Centrex service except supporting T1 lines.  
See [Appendix C – T1 Integration](#) for details on T1 set-up and integration.

### IP Network

- **Refers** - Refer/Transfer capability must be supported by the terminal IP device that are calling into CALLMaster. If CALLMaster is working behind a Gateway or IP PBX then that device must also support "Refer" capability for CALLMaster to affect a transfer.

## Determine and Validate Transfer String - (Not applicable for IP Network)

If you are using a digital connection, channel bridging or extension side T1, you do not need to perform the following tests. If you are using Centrex, 3-way Call or KSU/PBX analog extensions to facilitate the transfer, you will need to know the transfer string, the busy pull back string, and the no answer pull back string to make the transfer successful. It is easiest to get this information from your phone service provider or switch technical support. If you do not know these values, perform the following tests with a regular single-line phone:

### Receive A Call From Outside

Substitute a single-line telephone for CALLMaster. Plug the CO line directly if using Centrex or 3-way Call or an extension from the CALLMaster set of extensions if using a KSU/PBX into a single-line analog telephone. You should be able to pick up an outside line and call your single-line phone. If it doesn't ring, your CALLMaster will not answer. If you are using a switch, make sure it is programmed to allow incoming calls to be automatically routed to the single-line analog extensions so that an incoming call from the outside makes the single-line telephone ring.

### Transfer

You should be able to perform a flash-hook transfer from the single-line phone to another phone. To perform a hook-flash transfer, simply tell the caller to hold, and then press the flash button (or the on-hook switch) briefly. If you are using 3-way Calling then you may require a special code in addition to the flash hook. You should now get a dial tone from the telephone system to proceed to the next step. Dial another number or switch extension. When that number/extension picks up, if you have Centrex or a switch, then hang up the phone; else if you have 3-way Calling, then enter another flash hook to connect the 3 parties. If the party that called in is now speaking to the number you dialed, then you have correctly tested for hook-flash transfer. Make sure you initially called in on an outside trunk line; otherwise the test will not be valid.

### No Answer

If you are performing in-band integration (where the switch performs the supervision and sends the call back to CALLMaster along with a special code that tells CALLMaster what to do with the call, such as send caller to voice mailbox or try an alternate attendant record), you do not need to test this phase of the transfer - please move on to *message light on and off*. Once you have made a flash-hook transfer happen, try the exact same process to an unanswered phone. Pull the call back, making note of what process is required to make that happen. For example, your switch may require you to press the flash-hook key once to transfer, then again to bring the call back. This information is very important, because it will be entered into the 'Attendant Parameters' so that CALLMaster can use it for pulling calls back.

### Busy

Perform the same test as the unanswered call, but try to transfer to a busy extension and make notes of what is required to pull the call back.

### Message Light On And Off

Applicable only if you have a switch. From the single-line phone, activate a message waiting light on one of your phones. Make note of exactly what tones and signals are required. This information can probably be found in your switch user's guide. If not, contact your switch vendor. Make sure you look up the code for turning on the message waiting light from a *single line telephone*. Some telephone systems have different codes for lighting a message waiting light from a digital station set versus a single line analog telephone set.

### Disconnect

Have someone calling from an outside line hang up on you and listen for a disconnect signal of some kind. The telephone system may have certain codes it sends, such as an 'A' tone or a combination of touch-tones such as #9. If your switch does not send a disconnect signal, your voice mail messages may have long silences at the end after the caller has hung up. Additionally, your lines won't disconnect as fast, which will slow down turnaround time on incoming calls. Make sure you have run [PBXpert](#).

## Validate Transfer Strings

The two most frequent causes of incorrect Attendant transfers are (1) incorrect transfer pullback strings, or timings, and (2) misinterpreted ringing and busy signals. The following test should help determine the problem. Remove CALLMaster from its extension and plug in a standard telephone instead. Receive a call, and transfer it to another extension. Then repeat the procedure, but call a busy extension and pull the call back. Repeat those steps again with an unanswered extension and pull the call back. The procedures you just used should be the same as the transfer and pullback strings specified in the Attendant configuration (with CALLMaster running, go to 'Parameter' then 'Attendant'). The default values below mean that to transfer a call, CALLMaster dials a flash-hook, pause, and the number to dial. If the call is not answered, CALLMaster dials a flash-hook and a pause to pull it back. If the extension is busy, it dials a flash-hook and a pause.

```
Transfer String    '&,'  
Drop 'No Answer' Call    '&,'  
Drop 'Busy' Call    '&,'
```

If you have three-way calling, your pullback strings might look like this:

```
Drop 'No Answer' Call    '&,&,'  
Drop 'Busy' Call    '&,&,'
```

Because the first flash-hook makes a three-way call, and the second drops the busy or unanswered extension. FLASH\_TIME (&) and PAUSE\_TIME (,) are Dialogic parameters, which you can adjust. When the FLASH\_TIME is too long the switch treats it as a hang-up; when too short the switch doesn't detect it. When the PAUSE\_TIME is too long the switch aborts the transfer; when too short the switch misses the following digit. Flash and pause time should be almost as low as practicable, but slightly above the threshold, because call volume affects switch performance. If the pullback strings are correct but the pullback still fails, the problem may be that CALLMaster analyzed the call result incorrectly. To accurately detect a no answer or busy, you should tune CALLMaster to your phone system's ring and busy signals. CALLMaster uses Dialogic parameters to measure ring and busy signals.

## **KSU/PBX Users Determine Call Supervision - (Not applicable for IP Network)**

### **Switch Supervision using In-band Integration.**

A telephone 'switch' (a.k.a. KSU, PBX, phone system) has the ability to send integration information along the line that tells the phones where the call came from and why. CALLMaster can use this information to route callers to Failed Transfer Actions or to voice mailboxes.

The switch has much more knowledge about the phone call than CALLMaster does. It knows if and when the call was answered, or if the line is busy or unanswered. Having the switch perform supervision allows your users to keep using their phones the way they've been using them. The only condition of this tight integration is that the switch must send the call back to CALLMaster along with a special code that tells CALLMaster what to do with the call, such as go to a specific Mailbox or Attendant.

For example: Bob is at extension 45, and has a mailbox 1045. A call comes in to CALLMaster and the caller dials 45. CALLMaster does a hook-flash, then dials 45 and hangs up. The switch now supervises the call progress. Bob is already on the line, so the switch sends the call back to CALLMaster. When CALLMaster picks up, the switch must enter/send Bob's Mailbox #. To do this, it presses the touch-tones 1045, and the Mailbox # is entered. The system administrator must make 2 Attendant records - Attendant ID 45 with Type of Transfer set to Transfer so CALLMaster can try Bob's extension and Attendant 1045 with Type of Transfer set to Mailbox and Mailbox Additional Settings to 1045 to send caller directly to Bob's mailbox. In other words, on what is typically called a 'Busy/No Answer forward' the telephone system returns the call to the CALLMaster extension. When CALLMaster answers the call, the telephone system dials a '10' PLUS the extension number (in this case, '45'). Thus, we know that if the Attendant Main Menu receives a '1045' CALLMaster will select Attendant ID 1045 which sends the call directly to Bob's Mailbox.

In an ideal situation using 'in-band integration,' each employee has two Attendant records...

1. An Attendant record whose Attendant ID is the extension number itself (e.g. 10), with a Type of Transfer of 'Transfer' and the extension number entered in the 'Number to Dial' field.
2. A second Attendant record whose Attendant ID is set to the string sent by the PBX, OTHER THAN JUST THE EXTENSION NUMBER. For example, if the switch sends the extension number plus a 2 (like in many Comdial switches), we'll have a second Attendant record whose Attendant ID is set to 102 (the extension 10 with a 2). The Type of Transfer is 'Mailbox' and the mailbox number in the Additional Settings 'Mailbox' field.

CALLMaster System Parameter 'Phone System Manufacturer' and 'System Settings' would need to be set based on the type of switch.

### **CALLMaster Supervision**

If CALLMaster supervises the call, it must listen to the call progress and determine what to do with the call. The CALLMaster does not know as much as the switch about the call progress, so it must listen for a human response. If a connection is made and human voice is verified, THEN CALLMaster releases the call. This often causes a delay between the user picking up the handset and the call actually going through. It is for this reason that switch supervision (in-band integration) is more desirable. On the other hand, having CALLMaster perform the call supervision is easier to implement, and allows for Supervised and Screened transfers.

## Prepare and Test CALLMaster

### Make Records and Update CALLMaster Database

Making the following records will help you in creating your Attendant. We recommend that you create them in the following sequence. Details on how to create these databases are available in the Tour. Please read the entire chapter and familiarize yourself with these databases before you proceed.

#### Mailbox

Create a mailbox for testing your system out. Make sure that if you have a notification record you set the 'Message Notification Properties' field properly.

#### Notification

Notification is optional. Be sure that if you are working with message waiting lights, you designate them as record number zero (0).

#### Attendant All Day

Create access to an extension with an 'All Day' designator on it. Set the Number to Dial and the Failed Transfer Actions appropriately to send the caller to the mailbox on a failed transfer attempt (unless you are using in-band integration, which you would just make it a Transfer with an number to dial.

For telephone network integration:

- If you are using 3-way Calling, put the 3-way Call code and dial string in the Settings field.
- If you are using Channel Bridging, put the bridging indicator and port number in the Settings field.
- If you are performing in-band integration, you also need to create an access to the mailbox from the Attendant main menu.

For example, if your extension is 12 and the switch sends 55 plus the extension to indicate a mailbox call,

create an Attendant record 5512 with a type of transfer of Mailbox.

#### Ports

The port has to be altered to answer with 'Attendant'. Make sure to change the greetings to the Attendant greetings, which are 502-504. If you want the CALLMaster to be able to dial out on the port, set the type to 'Inbound' and 'Outbound'.

If you have T1 lines, please read [T1 Set-up and Integration](#).

#### System Parameters

Once you are set up in the databases, you should change certain system parameters, such as message waiting light/transfer strings if applicable, system area code and phone number, default paths for faxes, etc. If you are integrating to a digital PBX using the JCT-U board, set the System Settings parameter to 'PBXCID'.

#### Attendant Parameters

Use the notes you made when determining transfer string and update the Transfer String, Drop 'Busy' Call, Drop 'No Answer' Call parameters in the 'Attendant Parameters' section if integrating with a telephone network. CALLMaster will use these parameters to perform transfers and pull back failed attempts.

### Test Transfers

Test your work. See if you can make a transfer happen to your extension. Make sure that you are always calling in from an outside line. No one would be transferring from within the office, so make sure your tests are logical.

### Completed Transfer

Make a call, select a valid attendant id and make sure call is transferred successfully.

#### No Answer

**Did you go to mailbox?**

After the unanswered call was pulled back, did you go to the mailbox you designated in your Attendant record? If not, check the Attendant ID, then check that the mailbox was created. If you are using in-band integration, make sure the switch is sending the correct codes.

**Did the light go on?**

If you chose to include a message waiting light notification, your message light should have lit once you left a message. If it did not, check to see if CALLMaster made an attempt to light it. This can be found in the CMCalls.TXT file or by looking at the running screen (the highest port with the designation of Notification checked should have increased by 1). If you chose to activate a pager, did the pager go off? If not, check the CMCalls file for an attempt. A common mistake here is an incorrect placement of the notification phone number. If you had a problem, you will need to log into the mailbox so that the notification can be halted.

**Check mailbox log in did light go off?**

If your notification was successful, log into your box. Make sure to log out by pressing the pound (#) key several times. This helps the line clear faster. If the light did not go out, check to see that the code is correct in the configuration screen.

**Busy****Did you go to mailbox?**

If an unanswered call worked correctly and the busy did not, it could mean that the sequence of events for busy signals is different, or you have made a mistake in the 'Attendant Parameters' configuration screen. Check both of these possibilities.

**Did the light go on?****Check mailbox log in, did the light go off?****Test Disconnect**

Call into the system, watching the CALLMaster screen. Leave yourself a message and hang up instead of pressing star (\*) to end your message. See how quickly CALLMaster disconnects. If it is a long delay, you either have problems detecting the disconnect tone or your switch is not sending one. To verify that CALLMaster is disconnecting when a disconnect tone is received, leave a message then during the message unplug the line from CALLMaster. Did it hang up immediately? If so, the problem is that the telephone system or phone company is not sending a strong disconnect signal.

## **Backup The CALLMaster Folder**

Once you have successfully completed the test, make a full BACK UP of the system now. It is imperative to backup your system often.

It is recommended that you periodically (daily, weekly, monthly) backup the entire Program Files\CALLMaster folder. By backing up the entire folder, you will not only back up the CALLMaster database but also any prompts, messages, faxes, scripts, etc.



## Complete CALLMaster Configuration

### Make Additional Records.

Add additional Attendant, Mailbox and Notification records using the test records created above.

### Make Recordings

#### System

In your system, you have over 300 pre-recorded prompts. You can either keep these, or record over them. If you choose to record over them you should make a backup of the \pmt folder or you will need to reinstall to retrieve the original files.

#### Attendant

Your Attendant main menu is prompt 505, which is the prompt where you tell callers what they can enter. You also can record alternate directory names and extension IDs if you are using an Attendant directory. See how to [Set-up Attendant Directory](#).

#### Mailboxes

At this point, invite your users to log into their boxes and record personal greetings, names, etc.

Test again using different attendant ids and test scenarios. Once you are satisfied that CALLMaster is performing as desired, make another backup of the CALLMaster folder.

## **Attendant Switch Integration**

[Using PBXpert To Analyze Your Phone System Signaling](#)

[Digital PBX Integration](#)

[T1 Integration](#)

[SMDI Integration](#)

## **PBXpert – Determine and Enable Your Phone System Signaling**

The Dialogic PBXpert utility program located in the Dialogic program group or Start / Programs / Dialogic may be required to determine and register the signal from your PBX.

Make sure that CALLMaster service is not running when you run PBXpert. You must have two lines from the CALLMaster connected to two extensions on your phone system. PBXpert will then run a wizard, which will help you set up the analysis. Basically you tell it what lines are hooked up to which extensions. It will take a few minutes to run and will graphically display its activity. When it is done it will generate a Tone file, which is what it learned. You then can select to enable this tone file so it will recognize these signals correctly. You can also disable the tone file if needed.

In addition to the PBXpert program CALLMaster supports Global Tone Detection controlled by the Global Tone table. In that table you can define additional Tone Types for busy, ringing, reorder, disconnect and fax tones. You will need to know the timing and frequency of the signals you want to define. It is possible to also run PBXpert to analyze these additional tones and then you would translate those signal values into the CALLMaster table.

## Digital PBX Integration

CALLMaster supports in-band integration to various digital PBXs using the Intel/Dialogic PBX Integration boards (D/42JCT-U and D/82JCT-U boards).

The Dialogic drivers emulate the functionality of the specific handset models to provide tight integration for call progress and disconnect supervision.

See [Appendix D – Digital PBX Integration Set Up Guide](#) for PBX support and integration.

## **T1 Integration**

CALLMaster supports various T1 and ISDN-PRI protocols.

See [T1 Set Up Guide](#).

## SMDI INTEGRATION

Simplified Message Desk Interface (SMDI) is an integration protocol defined by Bellcore and supported by the telephone company and certain PBXs (such as some by NEC) and integration devices. SMDI provides information about who is calling whom and why. It is an **out of band** signal that comes into a serial port on the PC. In Centrex SMDI, there is a modem connected to the CO from the computer. In a PBX environment, the serial link may come directly from the PBX to the PC, or it may pass through a hardware integration device. The information is then passed to the PC from the telephone switch. SMDI provides three important pieces of information:

- The called party
- The calling party
- The reason for the call (forwarded, no answer, busy, or direct)

The SMDI feature of CALLMaster can process that information and use it to perform a tight integration, route to mailboxes, etc. SMDI also provides for message waiting indication; in the PBX environment, that usually means message waiting lights; in the Centrex environment it means setting a stutter dial tone. Notification record for SMDI Message Waiting Light ON/Off is not required. The mailbox is used as default station ID. If a different station id is required, create a corresponding notification record with the desired station id in the Dialing Prefix field. The notification record will then be used.

For SMDI to work:

1. The COM ports table must be configured correctly. The com port # and the line settings are usually set to 1200 baud, E, 7, 1;
2. The system variable 'Phone Manufacturer' is set to '5';
3. A cable must be connected from the computer's serial port defined in step 1 above to the SMDI modem or PBX;
4. The SMDI modem or PBX must be set to transmit and receive the SMDI packets.
5. The SMDI MDT tags on each Phone Port must be set to match the incoming SMDI packet.

## Frequently Asked Questions - Attendant

### **Q: How do I log into my mailbox from Attendant?**

A: There are two ways to accomplish this. The easiest way (by far) to log in from the main Attendant menu is to simply press the star (\*) key. The second way is to create a password access to the voice mail main menu. To do this, you must create an Attendant ID that opens some sort of access to the voice mail main menu. You should set the Attendant ID 'Type of Transfer' to "Voice mail", "Log in to Mailbox" or "Mailbox" if abbreviated log in is set. For example, to make the caller enter "999" before logging in, you must make the Attendant ID equal 999 with Type of Transfer set to "Voicemail", "Log in to Mailbox" or "Mailbox".

### **Q: How do I change the opening greeting?**

A: The system administrator must log into the administrator mailbox and press star (\*). This will make the administrator options menu available. From this menu, pressing seven (7) will allow the user to change system prompts. We recommend recording prompts 502-504 to say the time of day greeting only. Prompt 505 should give callers their options (see next ques.).

### **Q: How do I change the options presented to the Caller?**

A: The system administrator must log into the administrator mailbox and press star (\*). This will make the administrator options menu available. From this menu, pressing seven (7) will allow the user to change system prompts. The opening Attendant main menu is prompt 505.

### **Q: How do I perform a transfer?**

A: If you have flash-hook transfer capability, simply enter the number to dial in the "Number to Dial" field. You should know whether CALLMaster is supposed to supervise the call or if the telephone switch is supposed to do it. Consult your phone system dealer or your SpeechSoft sales representative if you're not sure.

### **Q: What is a flash-hook transfer?**

A: A flash-hook transfer is a little like a 3-way call. The only difference is that when the person that initiated the 3-way call hangs up, the other two parties are still connected. For example, if you have a WE2500 handset (a home phone) and party A calls you, then you tell party A to hold on and you hit the flash-hook button. The flash-hook button is commonly used to answer a call coming in while you're on the phone. Most phone companies call this call waiting. Now that you have put party A on hold with the flash-hook, you get a special dial tone. Most times there will be a stutter, then a regular tone. If you are on a telephone switch, you should get a dial tone from your switch. Call the number you want to transfer the caller to and wait for someone to pick up. Party B picks up and you say you're going to send party A to them. You hit the flash-hook button or hang up the phone, and party A will now be transferred to party B. This is how a flash-hook transfer works.

### **Q: Do I have to fill in all of the fields?**

A: Definitely not. If you are not sending to a mailbox, don't fill in the mailbox field. If the field has nothing to do with what the Attendant ID record is trying to accomplish, leave the unimportant fields blank. If you are not performing "Retry" as an "Failed Transfer Action", the "Retry Attempts" field is unimportant and can be left blank.

### **Q: How do I access the protected fields?**

A: We do not recommend altering these fields, thus the protection. If you need to do this, see the section on "Protected Fields" in Chapter 4 for more information.

**Q:What goes in the Directory Listing field?**

A: This field is a protected field, and be managed by the administrator. It should contain the first or last name of the person or party associated with this extension, in numeric letters. For example, the name "Smith" would be entered as 76484. Unlike the Voice mail directory, the Attendant directory is not generated automatically. You need to have control over which Attendant IDs have directory listing because Attendant IDs that do not lead to extensions should not have directory listing. See how to [Set-up Attendant Directory](#) for details.

**Q:If I am not performing a transfer, do I need to alter the Failed Transfer Actions?**

A: No, they will not be used. These fields are only for screened and supervised transfer use.

**Q: What happens if the caller doesn't enter anything at the main menu?**

A: If the caller enters nothing, the default Attendant ID of '0' (zero) will be used. If you do not create an Attendant ID of zero (0), the caller will hear "There is no such extension." We recommend that you make a zero (0) Attendant ID since there are rotary phone callers that will be unable to enter any sort of touchtone.

**Q: My CALLMaster does not seem to recognize the ring signal my telephone switch gives. How do I teach CM the signals?**

A: Try running "PBXpert", which can be found by clicking on the Start/Program Files/Dialogic group.

**Q: When I send the caller to Audiotex, it doesn't go to the module I want. It tells me that there is no such module and hangs up on me.**

A: The "Audiotex" Type of Transfer sends the caller to the module defined in the phone ports database for this port. If you have not set one, it tries to go to module zero (0), which is not allowed to exist. It fails, and then hangs up. If you have defined a module in the phone ports database, it goes there. If you want a specific starting point, set the Type of Transfer to "Audiotex Module", choose the specific module number in the Audiotex Module ID field and it should fix the problem.

**Q:What Attendant IDs does CALLMaster use on a holiday or a closed day?**

A: The "Evening" IDs, if available. Otherwise, it will use the "All Day" ID.

**Q: Why is there a pause after I enter the digits before the transfer sequence starts?**

A: If you have the Attendant Parameter called "Max Extension Length" set for a number that is higher than the extension you entered, then CALLMaster is waiting for more digits. For example, if your extensions are all 3 digits, set the Attendant parameter listed above to 3. In this case, a caller entering 1 for information might have to wait a second or two before accessing that area, since CALLMaster is waiting for 3 digits before transferring the call.

**Q: How do I return callers to the Attendant Main Menu (prompt 505)?**

A: Assuming you started your phone port with Attendant, give them a digit choice to do this and make an Attendant ID record that goes to a single Audiotex module. That module should have an Module Action of "Last Module."

**Q:What happens if the caller enters an extension that is not in the Attendant table?**

A: There is an Attendant parameter "Allow Any Transfer" which defaults to yes, in which case the system will



transfer the call to the extension dialed. There is another Attendant parameter “Call Analysis on Default transfers” which determines if these transfers are supervised for busy / no answer or not, the defaults is no. If this is set to yes then if a default transfer fails the call goes to the mailbox defined in the Voicemail parameter “default account”.

## Troubleshooting - Attendant

**Problem: CALLMaster tells me there is no such extension.**

**Cause:** There is no Attendant ID associated with the number entered.

**Solution:** Create an Attendant ID for the number the caller is to dial.

**Problem: CALLMaster does not transfer the call.**

**Cause:** There are several reasons this could happen. Outlined below are several causes and solutions...

**Cause:** The number to dial is incorrect

**Solution:** Enter the phone number or extension to be dialed as if from a single-line analog extension.

**Cause:** The phone system does not support single-line analog extensions

**Solution:** Obtain single-line analog extensions from your phone dealer. This can be accomplished with an OPX (Off-Premise Extension) card, or by simply adding the functionality to the switch. This capability should have been tested before purchasing the CALLMaster product.

**Cause:** CALLMaster hangs up and the phone doesn't ring.

**Solution:** CALLMaster somehow thinks the transfer has taken place. Make the transfer yourself and listen to what CALLMaster is hearing. If the board hears a signal that it reads as a connect, it will hang up as if it were letting the call transfer. You may need to run PBXpert. The switch may be sending a tone that makes CALLMaster disconnect; in that case, you may need to use Global Tone Detection

**Cause:** After the phone is picked up, CALLMaster doesn't send the call through immediately.

**Solution:** Make sure that a definite sound is made, such as a sharp 'hello'. The Dialogic board is looking for some sort of voice response so that it knows an answer has taken place. This is normal on supervised transfers. Another solution is to have the telephone system perform call supervision.

**Cause:** The person to be called is there and the call goes somewhere else, such as a mailbox.

**Solution:** The system time in the computer may have indicated to switch to another Attendant ID that sends the caller elsewhere. Check the Attendant IDs and the system time.

**Problem: After hours, transfers are still happening. No one is here to take the calls and it takes too long to go to voice mail.**

**Cause:** There are no evening Attendant IDs.

**Solution:** The best thing to do to avoid this problem is to create Attendant IDs that are set for evening. Each extension should have one if you don't want calls to go through. The Type of Transfer for the evening Attendant IDs would be Mailbox, with the associated mailbox for that Attendant in the Additional Settings - Mailbox field.

**Problem: When you transfer a call and it returns to the voice mailbox, there is still ringing in the background of the message.**

**Cause:** Incorrect pullback string

**Solution:** You may need to insert another flash-hook into your Attendant pullback string (Attendant parameters - 'Drop 'No Answer' Call'). This is most commonly found to be true in POTS and Centrex (Essex, Plexar, etc.) lines.

**Problem: The call does not pull back when there is no answer.**

**Cause:** There are two reasons this could happen. The first and easiest to deal with is that the Type of Transfer may be set to 'Transfer'. If this is true, simply change it to 'Supervised'. If the type is in fact 'Supervised', then it could be that the Dialogic board doesn't understand the ring that is heard.

**Solution:** First, run the PBXpert program described in this chapter. Next, check the switch manuals in the section on single-line analog extensions and see if there is a prefix or suffix required with the extension to

make the extension ring. On many switches, dialing only the extension will make an 'intercom transfer' which rings once then opens the circuit. CALLMaster needs to hear ringing or busy signals when it is supervising a transfer. You should also test the ring with your own ears by affecting a flash-hook transfer from a regular phone and listening to the ring yourself. If it is normal, look at the Dialogic parameter 'NBRDNA' to see how many rings it takes before CALLMaster determines a no answer.

**Problem: When callers use the Attendant directory, they hear numbers instead of names.**

**Cause:** No recordings for the Attendant extension name have been made.

**Solution:** The system administrator has access to the Attendant ID name recordings. See how to [Set-up Attendant Directory](#).

**Problem: Pulled back calls go to the default mailbox instead of the user's mailbox.**

**Cause:** Incorrect or no number in the Attendant ID Mailbox field

**Solution:** Make sure the Mailbox field contains the correct mailbox number in the Attendant record and your Failed Transfer Action is set to 'Mailbox'.

**Problem: The protected fields are not accessible.**

**Cause:** There is a reason for that.

**Solution:** You should not attempt to alter these fields. If you feel you must, see Chapter 4 - *Protected Fields*.

# AUDIOTEX

## OVERVIEW

CALLMaster Audiotex provides a powerful capability for developing sophisticated IVR systems. Each module/record is defined by a 'Module ID' and has four basic capabilities, which are performed in this order:

1. Say something to the caller by playing a recorded message;
2. Get a touch-tone response from the caller and/or record a message;
3. Execute an action based on the Module Action and run a CALLBasic script if defined
4. Determine the next module/record.

Audiotex Modules are used as basic building blocks to create IVR applications. The conversation flow can be predetermined or be based on the caller's responses. Each module can execute a 'program script' (a set of instructions written in CALLBasic) to accomplish a desired function (e.g. access a customer record based on a Caller input key). By connecting modules using the Next Module ID, you can create various custom IVR applications. Combined with the ability to run CALLBasic script for any module, the possibilities are endless. Custom applications range from systems that organize and deliver information, to ones identifying the closest store location and giving directions, to providing services (voice mail, dating, fax).

### Reserved Audiotex Modules

Audiotex Module IDs 0 thru 4 are reserved and should not be used by your custom application. These modules should not be changed and will be recreated by CALLMaster at each service start-up as follows.

Audiotex	
Module ID	Module Action
0	Last Module
2	Voice Mail
3	Attendant
4	Next Module

For details, see:

[Managing Audiotex](#)

[Frequently Asked Questions](#)

[Troubleshooting](#)

[CALLBasic](#)

You can also view and download an [Example of Custom IVR Development \(txt\)](#).

## Managing Audiotex

As you will see from the following description of the database table fields, the possibilities are endless. They range from systems that organize and deliver volumes of information, to ones identifying the closest store location and giving directions, to dating services and trivia games.

Click on desired field for explanation.

The screenshot shows the CALLMaster Manager interface for the Audiotex module. The window title is "CALLMaster Manager - [Audiotex]". The status bar indicates "CALLMaster Service Status: Running".

The main configuration area is for Module ID 100, with the label "Default Key" and the action "Start call and initialize values".

**Audiotex Table:**

Module ID	Action	Label	Next Module ID
100	Default Key	Start call and initialize values	110
110	Default Key	Perform call analysis to determine if it is a human, an...	9999
200	Default Key	Human Connect Detected. Play Audiotex Module r...	9999
300	Default Key	Answering Machine Connect Detected. Play Audiot...	9999
400	Default Key	Fax Connect Detected. Use FaxSend to fax docum...	9999
500	Default Key	Other Connect	9999
9900	Default Key	Call Failure Detected	9999
9910	Default Key	Caller Hang-up Detected	9999
9920	Default Key	Maximum tries attempted	9999
9999	Last Module	Last Module - End call	0

**Additional Module Data:**

Module ID: 100  
Module Action: Default Key  
Next Module ID: 110  
Fail Module ID: 9900  
Label: Start call and initialize values  
Write Label to Log File: ☒

Response Length: 0  
Message ID: 0  
Mailbox: 0000  
Attendant ID: 0  
Collect TT Input:

Script File to Run:  
C:\Program Files\CALLMaster\Scripts\SampleApp-ODMsgFax\InitializeVal.bas

TT Table: "Touch Tones" <Module ID>

1	0	2	0	3	0
4	0	5	0	6	0
7	0	8	0	9	0
*	- Repeat	0		#	- Exit

Navigation buttons: < Previous, Save, Undo Changes, Exit, Next >

If you have checked off the "Show Audiotex Xtra" on the [Xtra Parameter screen](#), see [Audiotex screen with Xtra fields](#) for details.

## Frequently Asked Questions – Audiotex

### **Q: How do I access the Audiotex table?**

**A:** By selecting "Audiotex" from the Manage option in the CALLMaster Manager.

### **Q: How do I make a path with Audiotex?**

**A:** The first thing you should do is write-up a plan and a flowchart. The next thing you should do is to take the tour of CALLMaster. If, after taking the tour, you still don't know how to create a call path, go step by step and plan how you will use Audiotex to implement each part of your flowchart.

### **Q: What does 'Concatenate' mean?**

**A:** This is a rarely used Module Action that gathers the responses of the caller and joins it to the current Audiotex Module ID. For example, if you are in record 30, and the caller enters 45, concatenating the digits would give you record number 3045 as the next destination.

### **Q: What should I put into the 'Label' field?**

**A:** Anything you want. This field lets you print whatever you type here into the CMCalls\_<DATE>.TXT file for debugging and reporting. By putting an exclamation point (!) in front of an entry, it will print a numeric value to the file. This is used to test variables and capture response information. For example, you want to count the number of times a certain record has been hit, you would use a CALLBasic command in the script that adds one to a variable (e.g. `var = var + 1`). In the "Text for File Output" field, you would then type `!var` to have the number in the variable called 'var' printed to the CMCalls.TXT file. Very useful for debugging! Try to stay away from punctuation marks of any kind in your label field.

This field can also contain the mailbox number(s) that could make the audio recording for this Audiotex module. If all mailboxes are to be given permission to make the recording for this Audiotex module then this field should contain RECORDALL.

### **Q: How do I use Audiotex to interact with the databases?**

**A:** CALLBasic database commands used CALLBasic scripts perform this function. For more information on the commands themselves, refer to the CALLBasic chapter.

### **Q: What do I put in the Touch Tone Table?**

**A:** Unless your Module Action is 'Touch Tone Table', don't put anything there. If your Module Action is 'Touch Tone Table' and you don't know what to put here, you should refer to the 'Touch Tone Table' portion of this chapter to learn about its functions. Taking the CALLMaster tour will also help you use Touch Tone Table.

### **Q: When should I use a response length?**

**A:** The response length is used when the Module Action is set to Touch Tone Table and you want to get some digit(s) from the caller. For example, if you are getting a one-digit response to a question and placing it into a variable, make the response length one (1).

### **Q: Why doesn't my Audiotex routine work?**

**A:** Well, because you did something wrong. The way to deal with this is to debug the routine. This is accomplished by enabling the 'Write Label to Log File?' indicator and entering details of the record's function in the Label field. Run the routine, and then look at the CMCalls<DATE>.TXT file. In this file you will see each record and where things went wrong. Watch the system carefully to see where this happens by observing the

port's activities. For more information about the label field, read the description of the field in the beginning of the 'Audiotex' section. The CMCalls.TXT file is defined by the Logging Parameter 'Call Log File', and can be viewed while CALLMaster is running. It would be helpful to turn off the status screen first so you don't get a jumble of data on the screen. Another way to debug is to make a recording for each module. Each Audiotex module's message will be played. The last message you hear before the system freezes is the one you need to check for the problem.

**Q: How can I repeat the same prompt for several records without having to record them individually?**

**A:** Update the Message ID number for the records you want to repeat the message in.

## Troubleshooting - Audiotex

**Problem: The CALLBasic command you entered is not doing anything.**

**Cause:** The command was probably misspelled or misused.

**Solution:** Check the syntax and spacing of the command you wrote carefully against the example in the CALLBasic chapter. Remember - CALLBasic is case sensitive.

Another common mistake is typing the following:

```
If var = 1000
    Return 30
Else
    Return 40
End If
```

The 'Then' part of the clause is missing.

- The first line should read 'If var = 1000 Then'.
- EndIf is a single word and should be EndIf.

**Problem: Every time you pass through a record, it fails. You've checked it over thoroughly, but nothing seems to be wrong.**

**Cause:** Check the CMCalls <DATE>.TXT file for 'parse error'. This will direct you to the line that is incorrect. The syntax for CALLBasic is very case sensitive.

**Problem: A quick couple of touch-tones are heard instead of the recording.**

**Cause:** The recording pointed to by the Message ID does not exist.

**Solution:** Check to see that the Message ID points to the correct file ID for your recording.

**Problem: You want to go on to the next record, but CALLMaster just keeps saying 'Press star to repeat'.**

**Cause:** Response length used incorrectly.

**Solution:** Go through your call path to the trouble spot and look in the record. If you have a response length and are not looking for a digit there, make the Response Length zero (0). You also may not have recorded a choice for the caller to make in the record where you are looking for digits.

**Problem: Your Audiotex application keeps looping to the same record.**

**Cause:** Touch Tone Table 'zero' field used improperly.

**Solution:** Change the record listed in the 'zero' field of your table to something other than the same record, **or** set the 'Next Module ID' to X, where X is the record to send non-responses. The 'Next Module ID' is used when the caller does not enter a response. If your phone system or phone company is not sending a strong enough disconnect signal, your application might loop if your 'zero' field goes backwards in your application.

Audiotex is a powerful tool that can help you create thousands of application possibilities. Some sample applications are listed in Chapter 13 - *Sample Applications*. If you still need assistance with Audiotex, contact SpeechSoft Technical Support.



# Out Dial

## Overview

CALLMaster Out Dial application allows you to make telephone calls to a list of phone numbers that have been loaded into the system, and deliver and/or collect information or send a fax. The information delivered can be a simple message or a complex scripted conversation in Audiotex. Each phone number can be directed to a specified port or to the next available port. Each port can concurrently run a different Out Dialing program if desired by setting it to a different Application and setting the numbers to be called to a specific port. Alternatively, a number can use any available port and run a specific application based on the data in its 'Audiotex Module' field.

Calls are initiated on any available port regardless of the application type as long as the Outbound indicator is enabled for that port. Out Dialing uses phone numbers scheduled to be called on the CALLMaster OutcallSchedule table. The OutcallSchedule table is used to enter and track numbers for CALLMaster to call.

## Separate Database for the OutcallSchedule tables

The OutcallSchedule and OutcallScheduleCompleted tables can be in a database separate from the CALLMaster database if required.

CALLMaster allows you to set-up a connection string to point to any database desired. These connection strings must be set-up in the registry where the CALLMaster DB connection settings is, and called OutcallScheduleConnectionString and OutcallScheduleCompletedConnectionString.

To set-up these strings, use the regedit command, select My Computer / HKEY\_LOCAL\_MACHINE / Software /CALLMaster, then right click on CALLMaster and select New | Key and enter in the connection string.

## Activating Out Dial

To activate Out Dialing, you need to set a flag "ODS" in the System Parameter 'System Setting' field. This flag initiates the OutcallSchedule Dispatcher. The OutcallSchedule Dispatcher is a program running continuously in the background, that initiates outbound calls and manages the records to be called in the OutcallSchedule table.

To activate Out Dialing, open the CALLMaster Manager, select Parameters then System and add code "ODS" in the 'System Settings' field. To disable Out Dialing, remove the flag from the field.

**NOTE:-**CALLMaster service will need to be stopped and started to activate Out Dial.

## Initiating a Call.

There are several System Parameters and OutcallSchedule table fields that the Dispatcher uses to initiate calls.

The dispatcher first checks the System Parameter 'Start Calling at' and 'Stop Calling at' to schedule outbound calls. If the current time falls between these 2 parameters then the call will be attempted. The dispatcher will also check the date and time set-up in the StartDateTime, Expires, EarliestHour and LatestHour fields for each record on the OutcallSchedule table to make sure the call can be initiated.

When the call is connected, CALLMaster goes to the Audiotex Start Module specified for the phone port or Audiotex Module field in the OutcallSchedule record for the individual phone number in use.

If the call does not connect (receives busy or no answer) then CALLMaster will retry the call based values in the retry fields.

## Managing Out Dial

There are several ways that you can load/import numbers and manage the data. See [Managing Out Dial](#) for

details.

## Managing Out Dial

There are several ways that you can manage the Out Dial data:

1. [Via the OutcallSchedule Table](#) in the CALLMaster database using your own macros/executables. As the OutcallSchedule table is an SQL database table, you can create your own macros/executables to import and manage the data as required for your application.
2. [Via the CALLMaster Manager - Out Dial function.](#)
3. [Via the CALLMaster Manager - LoadPhone function.](#)

## Managing Out Dial - Via the OutcallSchedule Table

The OutcallSchedule table in the CALLMaster database should contain the list of phone numbers that you want to dial. The information in each field determines how each phone number called should be handled:

**Click on column title for field explanation.**

ID	Dia IPr efix	Are aC ode	Ph on eN um ber	Sta tus	Lin e	Re sul t	Co nn ect	Rin gs	Sta rtD ate Ti me	Ex pir es	La stC allT	Ear liest Ho	Lat est Ho	Da ys To Cal	Bu sy Ret ry	Bu sy Ret ry	Bu sy Ret ry	No An sw er M ax	No An sw er M ax	No An sw er M ax	No Co nfir m M ax	No Co nfir m M ax	No Co nfir m M ax	Au dio tex t Mo dul e	Ma ilbo nd	Se nd To ne s	Re cei veT es	Re cor dM sgl Ds	Ms glD s	AM Ms glD s	Fa xiD s	Na me	Me mo	
191718551212				2	0	6	0	0	5/17/2000	5/18/2000		0:00	0:00	5/14/1900	3	0	3	3	0	3	0	0	0	0	20580000	600								
29,12,212		2151212	55	2	0	6	0	0	5/17/2000	5/18/2000		9:00	5:00		3	0	3	3	0	3	0	0	0	20660000	700									
3		914	55	2	0	6	0	0	5/17/2000	5/18/2000		0:00	0:00		3	0	3	3	0	3	0	0	0	20660000	0									

## Managing Out Dial - Via the CALLMaster Manager - Manage Out Dial - Overview

You can manage OutcallSchedule records via the **CALLMaster Manager | Manage | Out Dial** function.

There are three grouping of fields available for use in the Out Dial screen:

- 1) Status
- 2) [Schedule](#)
- 3) [User Data](#)

Click on desired field for explanation.

### Status Fields

The screenshot shows the CALLMaster Manager - [Out Dial] window. At the top, there is a search bar with "Number to Dial: 914 2741234" and "Status: 4". Below this is a table titled "Out Dial" with columns: ID, DiaPrefix, AreaCode, PhoneNumber, Status, Line, Result, Connect, Rings, StartDateTime, Expires, LastCallTime, and EarliestHour. The table contains 20 rows of data. Below the table, there is a detailed view for ID 2595. This view includes tabs for "Status", "Schedule", and "User Data". The "Status" tab is selected, showing fields for "Last Called" (10/27/2009 12:17 PM), "Result" (10 Connect), "Connect" (4), "Disconnect Reason" (Finished OK), and "Secs" (78). A "Notes" field contains the text: "2: Call not allowed at this time 10/26/09 20:32:10. Rescheduled. 2: 10/27/09 12:16:51 Calling. 2: 10/27/09 12:17:27 connect." At the bottom, there is a pagination bar showing "11 of 18" records.

ID	DiaPrefix	AreaCode	PhoneNumber	Status	Line	Result	Connect	Rings	StartDateTime	Expires	LastCallTime	EarliestHour
2595	914	1111111	1	0	6	0	0	0	11/4/2009 11:56 AM	11/5/2009 11:56 AM		00:00:00
2596	914	2731111	6	0	0	0	0	0	10/27/2009 10:00 AM	10/27/2009 8:00 PM	10/25/2009	10:00:00
2597	914	2732222	6	0	0	0	0	0	10/27/2009 10:00 AM	10/27/2009 8:00 PM	10/25/2009	10:00:00
2598	914	2733333	6	0	0	0	0	0	10/27/2009 10:00 AM	10/27/2009 8:00 PM	10/25/2009	10:00:00
2599	914	2734444	4	0	10	4	0	0	10/27/2009 10:00 AM	10/27/2009 8:00 PM	10/27/2009 10:07 AM	10:00:00
2590	914	2735555	4	0	10	3	0	0	10/27/2009 10:00 AM	10/27/2009 8:00 PM	10/27/2009 10:06 AM	10:00:00
2591	914	2736666	4	0	10	4	0	0	10/27/2009 10:00 AM	10/27/2009 8:00 PM	10/27/2009 10:58 AM	10:00:00
2592	914	2737777	4	0	10	3	0	0	10/27/2009 10:00 AM	10/27/2009 8:00 PM	10/27/2009 10:59 AM	10:00:00
2593	914	2738888	4	0	10	4	0	0	10/27/2009 10:00 AM	10/27/2009 8:00 PM	10/27/2009 11:39 AM	10:00:00
2594	914	2739999	5	0	7	0	0	0	10/27/2009 10:00 AM	10/27/2009 8:00 PM	10/27/2009 11:40 AM	10:00:00
2595	914	2741234	4	0	10	4	0	0	10/27/2009 10:00 AM	10/27/2009 8:00 PM	10/27/2009 12:17 PM	10:00:00
2596	914	2742222	4	0	10	4	0	0	10/27/2009 10:00 AM	10/27/2009 8:00 PM	10/27/2009 12:30 PM	10:00:00
2597	914	2743333	4	0	10	0	0	0	10/27/2009 10:00 AM	10/27/2009 8:00 PM	10/27/2009 1:10 PM	10:00:00
2598	914	2744444	4	0	10	3	0	0	10/27/2009 10:00 AM	10/27/2009 8:00 PM	10/27/2009 1:10 PM	10:00:00
2599	914	2745555	4	0	10	4	0	0	10/27/2009 10:00 AM	10/27/2009 8:00 PM	10/27/2009 1:47 PM	10:00:00
2600	914	2746666	4	0	10	4	0	0	10/27/2009 10:00 AM	10/27/2009 8:00 PM	10/27/2009 1:48 PM	10:00:00
2601	914	2747777	5	0	8	0	0	0	10/27/2009 10:00 AM	10/27/2009 8:00 PM	10/27/2009 2:37 PM	10:00:00
2602	914	2748888	5	0	7	0	0	0	10/27/2009 10:00 AM	10/27/2009 8:00 PM	10/27/2009 2:38 PM	10:00:00

## Managing Out Dial - Via CALLMaster Manager - Manage LoadPhone

### LoadPhone Menu

The **CALLMaster Manager | Manage | LoadPhone** function provides several options to load/import numbers, as well to reset the numbers, recall all the numbers and set numbers to Don't Call status.

See [Out Dial LoadPhone Menu](#) for options.

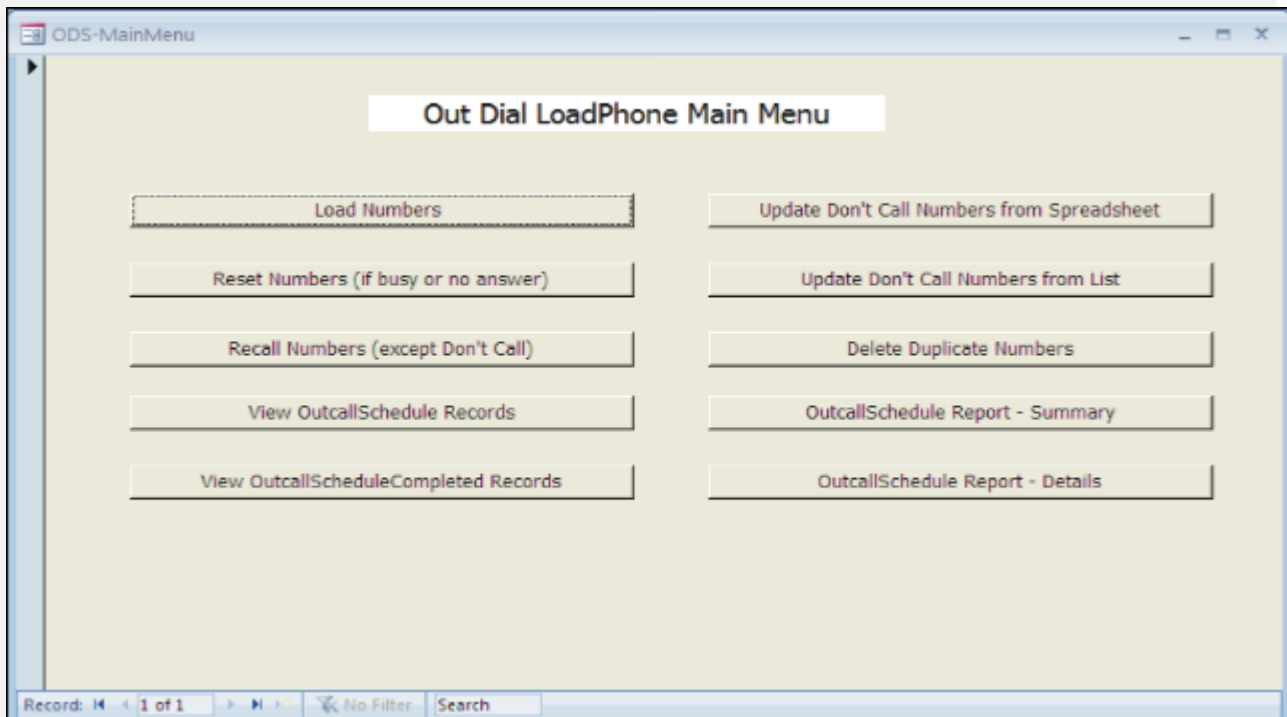
### Import Templates

There are several templates provided to assist in loading numbers to the OutcallSchedule table. See [Out Dial Input Templates](#) for details.

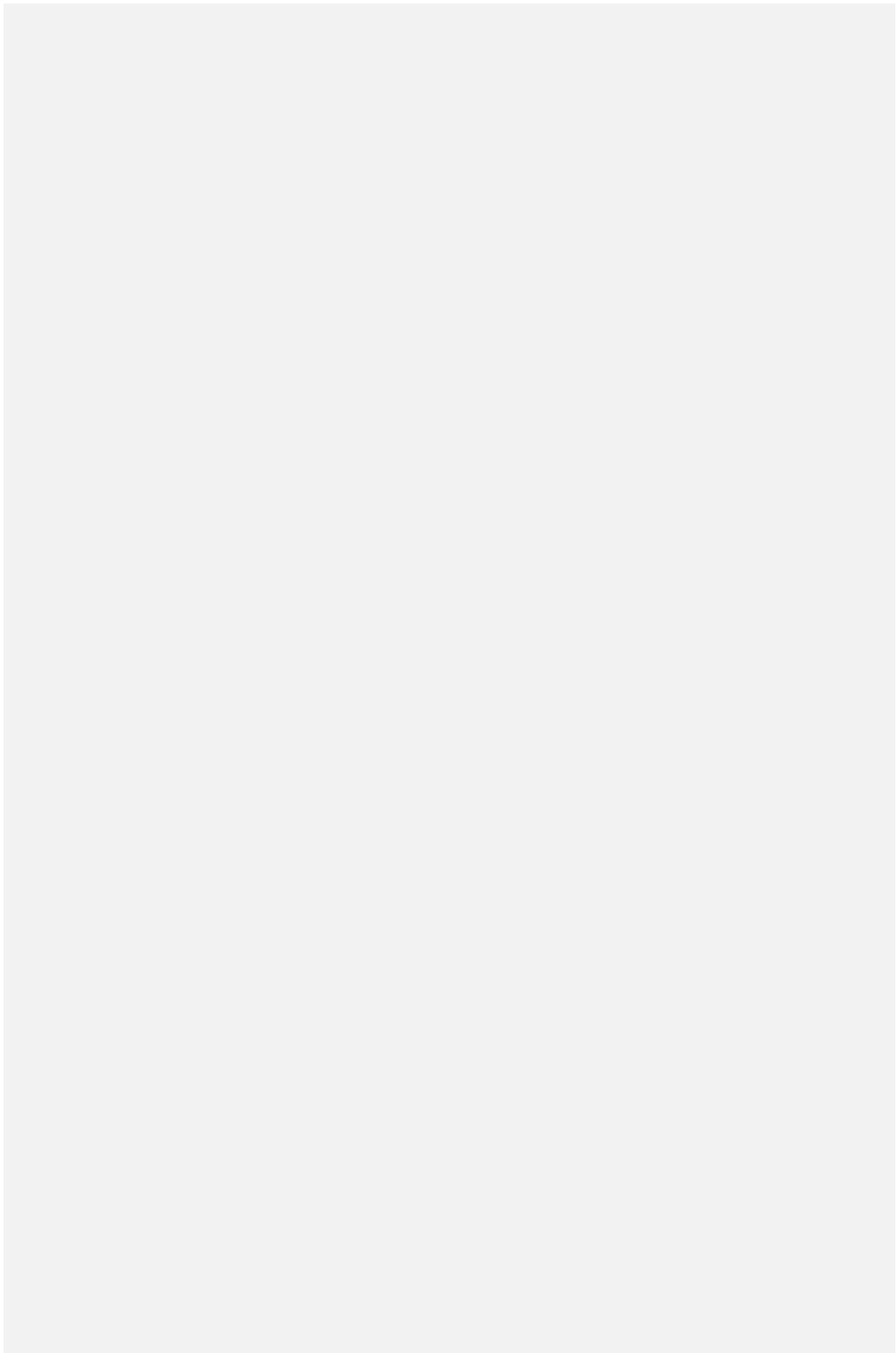
The name, format and location of these lists templates **MUST** remain the same and cannot be changed if you want to use the [LoadPhone Load Numbers](#) option to import to the OutcallSchedule table.

## Managing Out Dial - Via CALLMaster Manager - Manage LoadPhone Main Menu

There are several options provided on this menu to load/import numbers, as well to reset the numbers, recall all the numbers and set numbers to Don't Call status.



Option	Description
<b>Load Numbers</b>	Enables you to load phone numbers from various text and spreadsheet templates.
<b>Reset Numbers</b>	Resets the status of all numbers to 'Call' except those with status of 'Don't Call' or 'Called'.
<b>Recall Numbers</b>	Resets the status of all numbers to 'Call' except those with status of 'Don't Call'.
<b>Update Don't Call Numbers From Spreadsheet</b>	Allows you to match numbers loaded in the OutcallSchedule table to external list of numbers that should not be called. The matching numbers in the OutcallSchedule table are set to Don't Call status.
<b>Update Don't Call Numbers From List</b>	The Don't Call numbers list must first be loaded into either the ODS-DontCallList.txt or the ODS-DontCallSpreadsheet.xls templates. Select the function corresponding to the template used. The numbers in the OutcallSchedule table that match the ones in the selected template are updated with Status of 'Don't Call'. This is a way of not calling specified phone numbers from the OutcallSchedule.
<b>Delete Duplicate Numbers</b>	Delete any duplicate numbers in the OutcallSchedule table.





## Managing Out Dial - Via CALLMaster Manager - Manage LoadPhone - Overview

There are 5 Steps involved in loading numbers:

[Step 1 -Click 'Load Numbers' button on the Out Dial LoadPhone menu](#)

[Step 2 -Determine if you want to delete numbers existing on the OutcallSchedule table](#)

[Step 3 -Select the input template for loading numbers](#)

[Step 4 -Update fields with required fixed settings](#)

[Step 5 -Determine if you wish to delete duplicate numbers](#)

## Out Dial - Input Templates

Out Dial List templates are available in the CALLMaster\Outdial Lists folder. Select from these templates to load your phone numbers or the Don't Call numbers.

[Don't Call Number List](#)

[Input Templates for Phone Numbers To Be Called](#)

The name, format and location of these lists templates **MUST** remain the same and cannot be changed if you want to use the [LoadPhone Load Numbers](#) option to import to the OutcallSchedule table.

## Out Dial - Human, Answering Machine, Fax Detection

CALLMaster receives information from the Dialogic board to enable it to determine the connect type. Various parameters need to be adjusted depending on your Out Dialing program.

1. **If you want the system to call a number and leave a message for an answering machine only:**
  - a. Set the Dialogic Parameters
    - i. INTFLG to '1'
    - ii. NBRDNA to '8'
  - b. Stop CALLMaster via the Control Panel\Service.
  - c. Start CALLMaster via the Start\Programs\CALLMaster\CM Main Windows
2. **If you want the system to call a number and respond to only a human:**
  - a. Set the Dialogic Parameter
    - i. INTFLG to '8'
  - b. Stop CALLMaster via the Control Panel\Service.
  - c. Start CALLMaster via the Start\Programs\CALLMaster\CM Main Windows
3. **If you want the system to call a number, differentiate between a human and answering machine response:**
  - a. Set the Dialogic Parameters
    - i. NBRDNA to '8';
    - ii. HEDGE to 'Detect on TRAILING Edge';
    - iii. INTFLG to '7'
    - iv. PAMDFAILTIME to '10'
    - v. Close the window and SAVE the changes.
  - b. Set System Parameters
    - i. Record Max Silence to '3';
    - ii. Close the window and SAVE the changes.

NOTE: The 'Record Max Silence' parameter monitors the amount of silence required to detect end of answering machine greeting i.e. CALLMaster will require 3 seconds of silence after the answering machine greeting to play your message. If you wish that to be less, say 2 seconds, set the value to '2').

It is highly recommended that you review the [Sample Application - Out Dial Human/Answering Machine/Fax Detection](#) for a quick startup.

Out Dialing Sample Applications are available for download and installation from our website [Sample Applications](#) or install from the CALLMaster CD-ROM.

## Frequently Asked Questions – Out Dial

### **Q: Where do I start? The tour says nothing about Out Dialing.**

**A:** Follow the steps below to set-up for out dialing:

Customize the relevant Audiotex modules and make the desired recordings;  
Populate the selected input template from the CALLMaster/Outdial List folder with the phone numbers and desired settings;  
Use the [CALLMaster Manager Out Dial LoadPhone](#) option to load phone numbers;  
Calls will be attempted on any available port regardless of application type that has Outbound indicator enabled.

### **Q: How can I enter a large list of numbers into the Out Dialing database?**

**A:** There are several ways to load numbers:

- Load numbers directly into OutcallSchedule table in CALLMaster database. You can customize queries/stored procedures to import the numbers directly into the table.
- Use the [CALLMaster Manager Out Dial LoadPhone](#) option to upload to the OutcallSchedule table. You will need to select and populate one of the three input templates with phone numbers and desired settings.

### **Q: How do I reset the calls that have been marked as 'Failed' so I can call them again?**

**A:** When importing/loading the numbers, you can get the set each record to retry multiple times using the retry fields in the OutcallSchedule record. Alternatively, you can manually reset the numbers using the [CALLMaster Manager Out Dial LoadPhone](#) options.

### **Q: Can Out Dialing call a randomized range of numbers automatically?**

**A:** No, Out Dialing can only call the specific numbers in its database. How you get them there and how you generate them, is up to you. A spreadsheet can usually generate a text file of numbers, in any order you create them. Many different programs have been used for this purpose.

### **Q: How do I halt the Out Dial in the middle of a series of calls?**

**A:** Open the CALLMaster Main Window, select the BIG RED button to stop all lines, or highlight the line you wish to stop and select the SMALL RED button. CALLMaster will stop making calls and will finish with the calls currently being performed.

### **Q: Can CALLMaster differentiate a response from an answering machine, a fax and a human?**

**A:** There are 2 system variables in CALLMaster (**\$Contype**, **\$CR**), which indicate if the response was by a human, fax or answering machine. You can then branch out accordingly in CALLBasic.  
\$CR=10 (Connect either by human or answering machine);  
\$CR=18 indicates fax machine;  
\$Contype = 3 indicates human;  
\$Contype=4 indicates answering machine.  
Check \$CR first and then check \$Contype.

**Q: Once I have finished with my numbers, how do I clear them from CALLMaster?**

**A:** Select the option to delete existing numbers before loading new numbers when running the Load Numbers menu option. This will append the existing numbers to the OutcallScheduleCompleted table before loading the new numbers.

**Q: The Out Dialing log has all kinds of numeric codes for Events. Where can I find what those codes mean?**

**A:** In the Appendix entitled *Dialogic Connection Type Events and Call Results*. The Connection Type Events section will list these events for you.

**Q: How do I import numbers from a CD-ROM or diskette into CALLMaster?**

**A:** You are responsible for creating a text file or an excel file as described in the [Out Dial Input Templates](#). We recommend contacting the manufacturer of your CD-ROM or the list house from which you purchased your call list and consult with them on how to create such a text file or spreadsheet. Once you have that file, use the [CALLMaster Manager Out Dial LoadPhone](#) option to import and manage the numbers in CALLMaster.

**Q: Can CALLMaster just dial a range of numbers?**

**A:** No. Since in many areas this is an illegal practice, CALLMaster has never included this functionality.

## Troubleshooting – Out Dial

**Problem:** You've got numbers in the database but CALLMaster won't call them when you start up.

**Cause:** There are a variety of reasons this could be...

**Cause:** The Out Dialing database does not contain a valid phone port designation.

**Solution:** Check the Port table. Make sure one or more ports have the Outbound indicator enabled and the numbers are loaded in the OutcallSchedule table. If you are calling out on more than one port, select port zero (0), which will make a call on any open port available for Out Dialing.

**Cause:** Phone lines may require a dial prefix.

**Solution:** If you are using a PBX, you may need to dial 9 before calling out. You also may need to dial a 1 on long distance calls. Put these prefixes in either the Out Dialing database records or in the system parameters. Putting these numbers in the system parameters will change the numbers globally.

**Problem:** When you Out dial, CALLMaster tells the caller 'There is no such module' and hangs up.

**Cause:** Caller is being routed to an Audiotex module that does not exist. Check the solutions below to make sure that the Audiotex module exists.

**Solution:** Your Phone Ports database for that port does not have the correct Audiotex start module. When CALLMaster makes an Out Dial, it looks to the phone port database record for that port, and uses the Audiotex start module to determine where the call should go. If you do not fill this field, it will look for record zero.

You may have an invalid Audiotex Start Module number in the AudiotexModule field in the OutcallSchedule table.

Check your custom Out Dial application call flow. You may not be sending the caller to a valid Audiotex module number.

# Faxes

## OVERVIEW

To enable fax capability using CALLMaster software, you need to purchase:

- 1) the CALLMaster Fax Software license, and
- 2) the Dialogic Fax type board.

Contact your SpeechSoft Sales Representative for more information.

### Fax Tone Detection - Define Global Tone for Fax

For the Dialogic board to detect a fax tone, we need to define the tone settings in the Global Tones table.

Open the **CALLMaster Manager**. Click on **Manage**, then select **Global Tones**. If there is an record for a Fax tone, then just verify the setting. If not then Add a record to this table with values as below.

Global Tones													
Tone Type	Characteristic	Detection Mode	Translation		Frequencies (Hz)				Cadence (10ms)				
			Translate to Char	Character	Tone 1 Freq	Tone 1 Dev	Tone 2 Freq	Tone 2 Dev	On Time	On Time Dev	Off Time	Off Time Dev	Minimum Repetition Count
Fax	Single Tone	Leading Edge	Yes	F	1100	50	0	0	0	0	0	0	1

[Inbound Faxes](#) can be received automatically to a mailbox (default or selected by caller) via Voice Mail, Attendant or Audiotex or to a specified file name via Audiotex. All inbound faxes are logged to the [FaxReceived table](#).

[Outbound Faxes](#) can be sent via custom Audiotex/CALLBasic applications or via the Out Dial functionality. Documents can be faxed directly to the fax machine from where the call was made supporting one-call faxing or to faxed a new number entered by the caller supporting two-call faxing. Outbound fax activity can be logged to the [FaxSendActivityLog table](#).

[Managing Faxes](#) details information required to maintain outbound fax documents.

## Faxes - Inbound

A fax document can be received into a mailbox if CALLMaster is set to Voice Mail, Attendant or Audiotex mode or to a document via Audiotex.

Inbound faxes are logged to the CALLMaster \Fax\Received folder as .tif files. Faxes received into a mailbox can be emailed as a .tif attachment if the mailbox is set-up for email notification. During log in, the mailbox owner can review new Faxes (header details), redirect new faxes to fax machines and save and delete faxes.

Learn more about:

[Voice Mail](#)

[Inbound Fax via Voice Mail & Attendant Mode](#)

[Inbound Fax via Audiotex Mode](#)



### **Faxes - Inbound - FaxReceived Table.**

This table is used to log all incoming faxes. The fax document file is stored in the CALLMaster\Fax\Received folder in the receiving Mailbox's sub-folder.

<b>Field Name</b>	<b>Description</b>
FaxID	Unique ID for activity event
Type	Fax type (1=ASCII, 2=RAW, 3=Tif)
Resolution	1=Fine (200 x 200); 2=Regular (200 x 100)
Pages	Number of pages received
Status	1=Received OK, -1= Receive Failed, 2=Saved
Seconds	The length of the call in seconds
DateTimeReceived	Date and Time fax was received
DateTimeDeleted	Date and Time fax was deleted (auto set to 2 months unless saved)
FileName	Full path name of file faxed
From	Name of the sending fax along with any caller id
Account	Account number containing the receiving mailbox # or 'FILE' if fax received to file
Comments	Field not currently used
Flags	Field not currently used

## Faxes - Outbound

Faxes can be sent using custom Audiotex/ CALLBasic applications or via OutcallSchedule functionality. Documents can be faxed directly to the fax machine from where the call was made supporting one-call faxing, or ask the caller to input the fax number to be called back at, supporting two-call faxing.

### Fax Document Type

Documents to be faxed must be in TIF, ASCII or RAW file format. You can create documents in the required format by:

- [Using the Faxc.exe Program to Convert Documents to Fax Format](#).
- [Using Manual Fax Control to Capture Fax Documents](#).

You can download the [Fax Test Document](#) as a sample document.

### Fax Out Capabilities

Documents can be faxed out by using either:

1. [The FaxSend CALLBasic Command](#), or
2. [Out Dial Faxing](#).

### Same-Call Vs. Two-Call Faxing

There are two ways you can send a fax to a caller:

1. Same-call fax. The caller calls from a fax machine and presses the start key to initiate the fax signal. The fax goes out on the current connection.
2. Two-call fax. Via an Audiotex script, the caller enters the desired fax number. This then triggers a separate call made by the system to fax out the desired document.

CALLMaster faxing currently supports Same-call fax using the Dialogic Fax board. You would develop an Audiotex module to support Two-call fax via the Out Dialing capability.

## Managing Faxes - Outbound

The Fax table identifies, locates and describes a file that is to be faxed using the **CALLBasic FaxSend** command. Important parameters are graphic format, document number, and file name. Create a single record in the database for each document to be faxed. The image to be faxed was either, received as a fax previously and saved as a DOS file, or created by scanning or desktop publishing. Files to be faxed must be ASCII, TIF or PCX files.

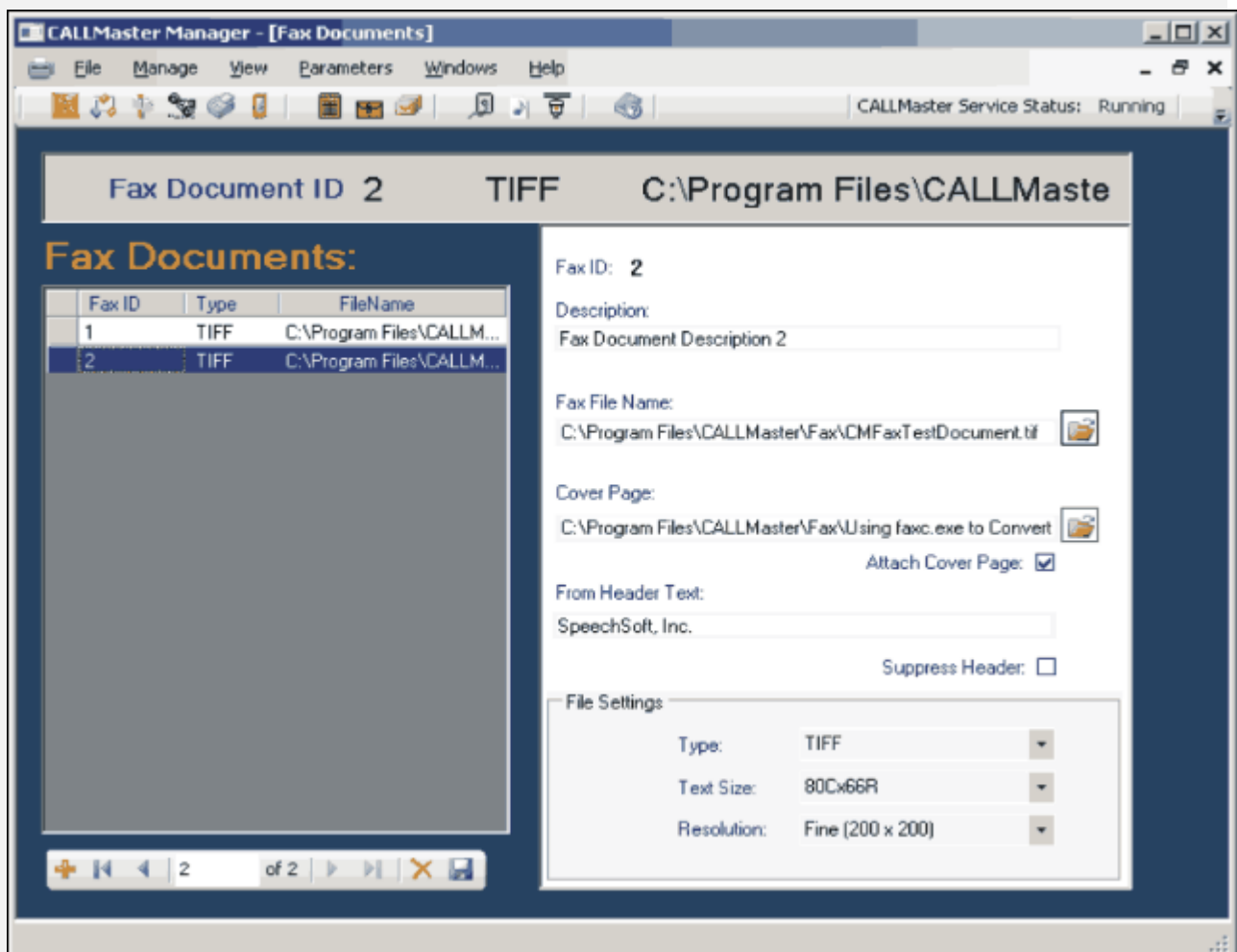
TIF files can be made in two ways: (download Fax Support from [SpeechSoft Download Page](#).)

1. By faxing a document into the fax board residing in the PC and saving the file as a .RAW or .TIF.
2. By saving a document as a .RAW file. Many graphics and word processing programs can save files in either TIF or PCX format. By creating the documents on a PC that has a fax board in it, you can usually *print* to the fax driver as your printer. Preview the fax (with no cover sheet) and then save the previewed file as a .RAW file.

You can configure the Fax table using the CALLMaster Manager

Each record contains information about the document or graphic: its type, resolution, and other pertinent header and cover page information. Fax support requires the fax license of CALLMaster plus a Dialogic board that supports fax.

Click on desired field for explanation.



## CALLBasic Command - FaxSend

**Command:** **FaxSend**

**Description:** Schedule a document to be faxed to the number called. For this command to work, you must have the necessary CALLMaster license with fax support, plus an optional fax board properly installed, configured, and tested. CALLMaster returns the fax send status in system variable \$faxsent.

**Format:** **FaxSend** *docID phonenum car*

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>docID</i>	Integer	The key of a Record in the fax database that describes which document to send.
--------------	---------	--

<i>phonenum</i>	String	The phone number is used only for reference purposes and not to make a call. This field can be null if faxing to the same number that the call is connected to as part of the Out Dial.
-----------------	--------	--

### Output:

<i>car</i>		Returns the Dialogic call result (see Appendix B).
------------	--	--

Program

**FaxSend** doc callernumber car

EndProgram

### Example:

```
Program
  phonenum = "919145551212"
  docID = 1
  Print "sending Fax 1"
```

**FaxSend** docID phonenum CAR

Print "CAR - "

Print CAR

'This will fax the document from the Fax table whose Fax ID is referenced in the variable docID.

' The phonenum variable in this command is for reference purposes only and not to make the call. The document will be faxed to the phone number called.

'Upon determining the results of the call it will place the Dialogic call result in the variable CAR.

EndProgram

For details on faxing see [CALLMaster Fax Support](#)

## **Faxes - Outbound - Using Out Dial**

A document can be faxed by creating record(s) in the OutcallSchedule table in the CALLMaster database and enabling Out Dial. The record can be created by a CALLBasic application, via the CALLMaster Manager or by any other third party application directly updating the CALLMasterSQL database.

The document to be faxed can either be defined in the CALLMaster FAX table and selected using the fax id or can be identified by the full path name. The record has to be assigned a port that is connected to the fax board.

When Out Dial is enabled, CALLMaster continuously scans the OutcallSchedule table regardless of the port application type and will attempt to schedule the call when a line is available. (see [Out Dial](#)).

## Logging and Tracking Activity - Using Fax Send Activity Log Table

This table is used to log outbound faxes.

CALLMaster is shipped with the FaxSendActivityLog disabled. To activate Fax Send logging, using the CALLMaster Manager, select **Parameters | System** and add the codes 'FAL' to the 'System Settings' field. To disable, just delete the code 'FAL'.

**Note:** Once activated the FaxSendActivityLog will start collecting data records. It is the Administrators responsibility to periodically archive data and remove records from the log. Otherwise the size of the database could grow to a size that inhibits proper operation of the system.

For Fax Send Activity, the events that are logged are:

Event	Type
Fax sent via CALLBasic	100
Fax sent via MAILBOXNOW	110
Fax sent via MAILBOXFORWARD	120

Each outbound fax event is stored in the table FaxSendActivityLog. Each event record contains the following information:

Field Name	Description
EventID	Unique ID for activity event
Date and Time	Event Date and Time
Type	Event type code (see above)
Seconds	The length of the call in seconds
Description	Event description
FileName	Full path name of file faxed
Mailbox	Source (mailbox #, Audiotex, ODFAX(OutcallSchedule table))
Data	

Notes

Fax status of either Faxed OK or Fax FAILED



## **Frequently Asked Questions – Faxes**

### **Q: What kind of files can I send/receive?**

A: .TIF, .RAW, and ASCII files are allowed. To make these files, you can fax the document into a fax board (saves as a .RAW file) or convert a file into .TIF with typical graphics programs. There are other ways to save files as .TIF or .RAW, but that depends on the software you are using to create the document. Consult the software manual that came with your graphics program(s) for 'Save As' options. You can save the document as a .RAW file by printing to the Microsoft Fax program and mailing it to yourself. The resultant item being sent will be a .RAW file, ready for faxing.

### **Q: How do I know what to do with the resolution?**

A: If you are sending an ASCII file, you should make the resolution 200x100. Otherwise, don't tamper with the 200x200 default.

### **Q: How do I send multiple documents in a single request?**

A: You need to use multiple FaxSend commands.

### **Q: I have my system set to voicemail and I also have a Dialogic fax board but I am unable to receive any faxes?**

A: Make sure that you have purchased the CALLMaster Fax software license. You should also make sure that the Fax tone has been defined in the Global Tone table.

## Troubleshooting - Faxes

**Problem:** The fax is not going through.

**Cause:** There are a couple of reasons this could be happening.

**Solution(s):** First, the command may not be correct. Make sure that the syntax of the FaxSend command is correct. Next, check to see there is a fax database record matching the document number in the **FaxSend** command. If there are indications in CALLMaster CMCall.TXT files that the request was made correctly, then the problem could be with the line itself, or the fax machine it was sent to. Be sure to test logically to determine where the error is occurring. Try sending to another machine.

**Problem:** The file you want to send exists, but CALLMaster is not sending it.

**Cause:** This probably means you have the incorrect path or filename.

**Solution:** Check that the path is correct in the 'Path' field. Also, check that the filename is correct and that the file is one of the three acceptable file types. If the file is something other than ASCII, TIF-F, or RAW the fax will be sent incorrectly.

# Global Tone Detection

## Overview

Global Tone Detection (GTD) allows the Dialogic board to detect specified tones at all times during a call. When it detects the specified tone, the Dialogic driver reports an event to CALLMaster. GTD is frequently used to detect disconnects on PBXs and key telephone systems that do not support loop current disconnect; when the driver detects a tone that has been defined as a disconnect tone, it reports a loop current drop to CALLMaster.

You will need to know the characteristics of the tone or tones you are trying to detect. Your telephone vendor can supply you with the necessary information. You must add one or more Global Tone records to your Global Tone database.

## Example of Global Tone for Fax

For the Dialogic board to detect a fax tone, we need to define the tone settings in the Global Tones table.

Open the **CALLMaster Manager**. Click on **Manage**, then select **Global Tones**. If there is an record for a Fax tone, then just verify the setting. If not then Add a record to this table with values as below.

Global Tones													
Tone Type	Characteristic	Detection Mode	Translation		Frequencies (Hz)				Cadence (10ms)				
			Translate to Char	Character	Tone 1 Freq	Tone 1 Dev	Tone 2 Freq	Tone 2 Dev	On Time	On Time Dev	Off Time	Off Time Dev	Minimum Repetition Count
Fax	Single Tone	Leading Edge	Yes	F	1100	50	0	0	0	0	0	0	1

## Managing Managing Global Tone Detection

Click on desired field for explanation.

The screenshot shows the CALLMaster Manager - [Global Tone Definitions] window. The title bar includes the application name and standard window controls. The menu bar contains File, Manage, View, Parameters, Windows, and Help. The toolbar includes icons for file operations, settings, and status. The status bar at the bottom right indicates 'CALLMaster Service Status: Running'.

The main content area is divided into two sections. The left section, titled 'Global Tones', contains a table with the following data:

ID	Type	Characteristic
1	Fax	Single Tone
2	Disconnect	Single Tone Cadence

The right section displays the configuration for the selected Global Tone ID 1. The configuration includes the following fields:

- Tone ID: 1
- Tone Type: Fax
- Characteristic: SINGLETONE
- Detection Mode: Leading Checked, Trailing Unchecked
- Translation: Translate To Char: ☒ Char: F
- Frequencies (Hz): Frequency1: 1100, Frequency2: 0, Deviation1: 50, Deviation2: 0
- Cadence (10ms): On Time: 0, Off Time: 0, On Time Deviation: 0, Off Time Deviation: 0, Min Repetition Count: 0

The bottom of the window features a navigation bar with a plus icon, a list of IDs (1 of 2), and a minus icon.

## Global Tone Detection - Example For Detecting Continuous Tones

To detect the continuous tone of a predetermined duration, create a global tone record with the 'Tone Characteristic' of 'Single Tone Cadence' or 'Dual Tone Cadence.'

Determine the 'Frequency' and 'Deviation' of the tone or tones. Set the 'Minimum Repetition Count' to zero.

Determine the minimum time that the continuous tone must be present before it is detected (this should be an even number). The minimum time the signal is present equals the 'Cadence On Time' minus the 'Cadence On Time Deviation.'

At the same time, the maximum time the signal is present must have no limit; in other words, it must equal zero. The calculation for the maximum time is 'Cadence On Time' plus 'Cadence On Time Deviation.'

In the diagram below, the signal must be present for four seconds before it is detected. Here is the math:

minimum time    4 secs. = 2 secs. - (-2 secs.)

maximum time    0 secs. = 2 secs. + (-2 secs.) - zero (0) indicates no maximum

## DTMF Frequencies Generated by the Telephone Push-Button Keypad

## **Frequently Asked Questions - Global Tones**

### **Q: How do I determine the frequencies that the record is asking for?**

**A:** Your switch's manual may have this information. Another option is to ask your phone vendor. You can also use PBXpert in the Dialogic Program Group

### **Q:When should I use Global Tone Detection?**

**A:** If you are experiencing problems in detecting a disconnect tone or other tones, there are other steps that should be implemented before resorting to GTD. First of all, if you are not getting a loop current drop (LCD) from the phone company (CO), or if the LCD is not long enough for the Dialogic board to detect, talk to the phone company to make the LCD longer. If you are working with a telephone switch or KSU, you should see if there is a programmable loop current drop signal. The best signal is a pound (#) key. Other than that, a D key is good because it is an invalid touch-tone according to CALLMaster. Only if these tools are not working for you should you create a global tone detection record.

## Troubleshooting - Global Tone Detection

**Problem: Panasonic 1232 switch does not recognize disconnect tone.**

**Cause:** The tone generated by the 1232 is not consistent.

**Solution:** Set the ports on the Panasonic to Voice Mail so that it sends a reorder tone as a disconnect tone. If that is compatible with your application, make a global tone record as it appears in the Global Tone Table above. If your Panasonic phone ports are not set to Voice Mail, you'll want to create a record that detects a #9 instead.

# IVR Application List

## Overview

CALLMaster can run several different IVR applications on a single system. The IVR Application List is used to identify an IVR application with the Starting Audiotex Module ID. This list is used to associate a port to the IVR Application it will be running.

## Reserved Application List Entries

The following entries in the IVR Application List are reserved to correlate to the reserved Audiotex Modules. These entries should not be changed and will be recreated by CALLMaster at each service start-up.

IVR Application List	
Audiotex Start Module	Application Name
1	Disabled
2	Voice Mail
3	Attendant
4	Audiotex

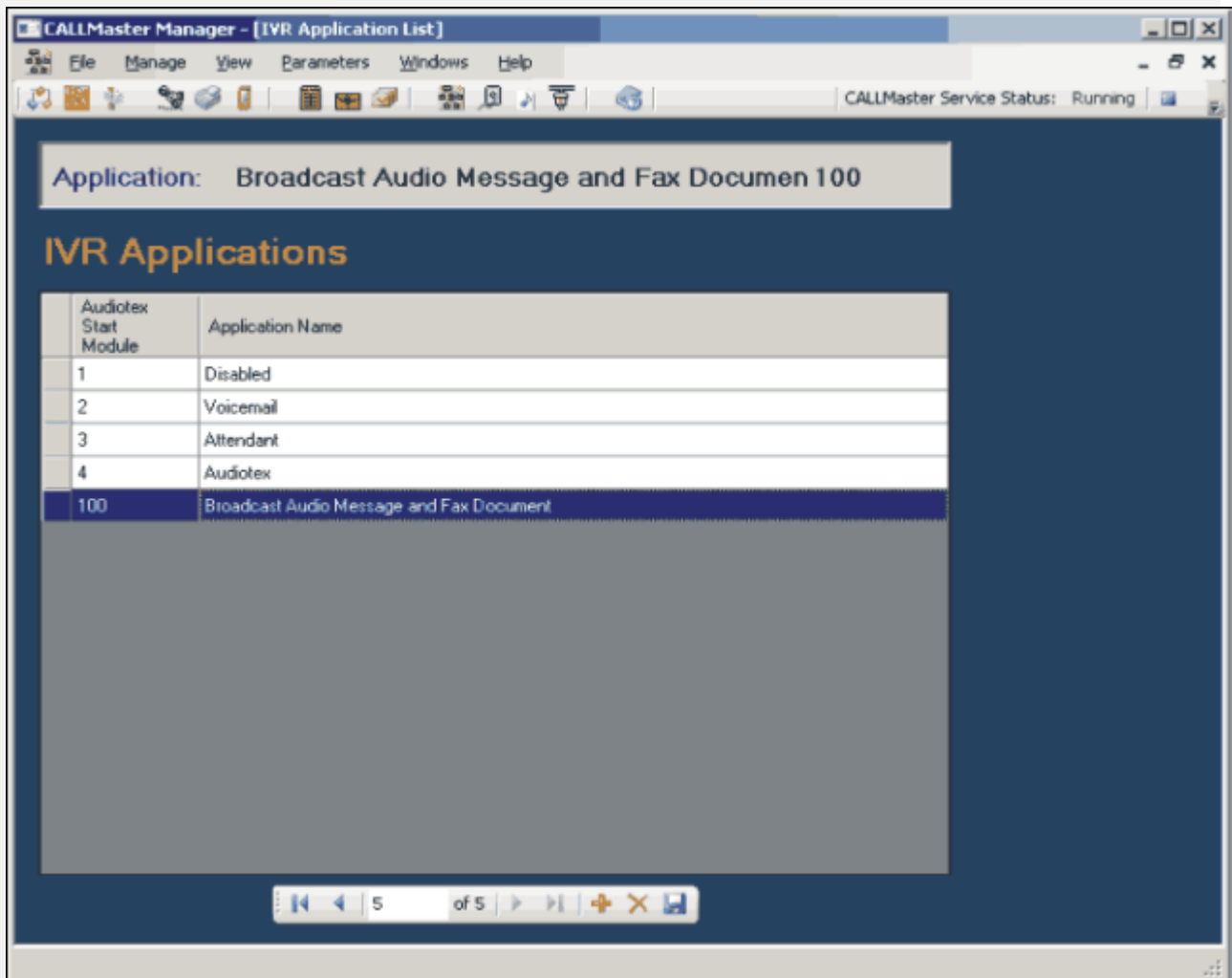
See [Managing IVR Application List](#) for details.



## Managing IVR Application List

Use the **CALLMaster Manager | Manage | IVR Application List** function to define each application with an Application Name and the corresponding Audiotex Starting Module ID. This list will be used to associate a port to the IVR Application it will be running.

Click on desired field for explanation.



The reserved entries in the IVR Application List should not be changed and will be recreated by CALLMaster at each service start-up.

IVR Application List	
Audiotex Start Module	Application Name
1	Disabled

2	Voice Mail
3	Attendant
4	Audiotex

# PORTS Overview

## Overview

Each call that CALLMaster processes starts in Ports. The Ports table determines whether to answer a call or make a call. It sets the number of rings in which to answer, determines the first speech, and controls the starting application.

Each Port can run a different application. The IVR Application List is used to define the starting application for that port.

[Manage Phone Ports](#) for telephone network integration.

[Manage IP Ports](#) for IP network integration.

## Managing Managing Phone Ports

Click on desired field for explanation.

CALLMaster Manager - [Phone Ports Loading]

File Manage View Parameters Windows Help

CALLMaster Service Status: Running

Port# 3 dxxxB1 Broadcast Audio Msg. / Fax Doc 100

### Phone Ports

Port #	Name	Application	Auditex Starting Module
1	dxxxB1C1	Voice Mail	2
2	dxxxB1C2	Attendant	3
3	dxxxB1C3	Broadcast Audio Msg. / Fax Doc	100
4	dxxxB1C4	Banking Demo	10099

Port #: 3 Name: dxxxB1C3

Application: **Broadcast Audio Msg. / Fax Doc**

Auditex Start Module: 100

**Settings**

Answer ringing ☒ after 1 Ring(s).

Makes Outbound Calls ☒

Line current loss drops call ☒

**Digital Network Interface**

T1 ☐ E1 ☐

ISDN PRI ☐

clear

**Greetings Files IDs**

Morning: 0 Closed: 0

Afternoon: 0 Holiday: 0

Evening: 0

Startup Tone File:

SMDI MDT: 0003

< Previous Save Undo Changes Next >

Exit

3 of 4

Managing IP Ports

Click on desired field for explanation.

CALLMaster Manager - [IP Ports]

FileManageViewParametersReportsWindowsHelp

CALLMaster Service Status: Running

Port# 1dxxxB1C1Voice Mail2

IP Ports

Port #	Name	Application	Auditex Starting Module
1	dxxxB1C1	Voice Mail	2
2	dxxxB1C2	Voice Mail	2
3	dxxxB1C3	Voice Mail	2
4	dxxxB1C4	Voice Mail	2

Port #: 1Name: dxxxB1C1

Application: Voice Mail

Auditex Start Module: 2

Greetings Files IDs

Morning: 19Closed: 203

Afternoon: 20Holiday: 209

Evening: 21

Startup Tone File: |

< Previous

Save

Undo Changes

Exit

Next >

1 of 4

## **Frequently Asked Questions - Ports**

### **Q: Can each port perform a different function?**

**A:** Yes. You can be performing Attendant in one port, Audiotex in another, and Voice mail in yet another. CALLMaster's port settings are completely customizable. In addition, you can perform different Audiotex routines for each port based on the Application set-up for that port. The Application must correlate to the IVR Application List.

### **Q: How do I know what greeting should be in the greeting fields?**

**A:** That depends on your application. If you are performing an Attendant application, you might want to use prompts 502-504. You can use any prompt you want here, including ones you have created outside of the originally provided prompts. Simply record the prompt desired and put the prompt number in the proper field. Remember, these greeting files will only be played for any Audiotex Module whose "Label" field includes the flag PTODG.

### **Q: How do I get CALLMaster to allow for outbound dialing for pagers and message lights?**

**A:** Edit the port number and activate the "Make Outbound Calls" field (set check mark).

## Troubleshooting - Phone Ports

### **Problem: The Out Dialing application doesn't call out.**

**Cause:** Your phone ports are not set correctly.

**Solution:** Make sure the field 'Make Outbound Calls' is turned on. If the problem persists, make sure your 'Local Dialing Prefix' and 'System's Area Code' parameters are correct. Make sure that the System Parameter 'System Settings' includes flag ODS. See [Out Dial](#) for details.

### **Problem: The CALLMaster doesn't answer on one of the ports.**

**Cause:** Could be a bad port. This sometimes happens when lightning strikes or power surges through the line. Rarely, it is the board as it is shipped since all CALLMaster-sent boards are tested before shipment.

**Solution:** If you think you may have a bad board or port, use the Dialogic Diagnostic utility. Check to see that the board driver loaded correctly, by looking for a line that says the board(s) were successfully installed. Next, select Dialogic Universal Diagnostics from the Dialogic program group. If the board passes, return to the phone ports and make sure that the 'Rings' is not set to zero (0).

### **Problem: Can't get caller ID information**

**Cause:** Incorrect Ring Count

**Solution:** Increase the ring count in the phone port to at least 2. The phone company doesn't send caller ID information on the line until it's between the first and second ring.

### **Problem: The ports answer out of order.**

**Cause:** Incorrect rollover.

**Solution:** Check your rollover with either the phone company or your telephone switch.

# CALLMASTER PARAMETERS

## Overview

CALLMaster can be configured for many different telephone system requirements and user preferences. You control the hour after which callers hear the morning, afternoon, or evening greeting. You can require messages to be brief, or change how long to wait for the line to settle after transferring, among other things. The parameters, described in detail in the following section, can be changed through the menus.

You can maintain CALLMaster Parameters via the CALLMaster Manager. Just click on the start button and choose Start | All Program | CALLMaster | Manager, then select from the Parameters drop down list.

The Parameter menu is divided into sub-menus, which address different groups of parameters:

<a href="#">System Parameters</a>	Defines the general functions such as the hours for using different greetings and how many repeat prompts.
<a href="#">Dialogic Parameters</a>	Controls the interaction with the Dialogic board, including call progress analysis and call transfer.
<a href="#">Voice Mail Parameters</a>	Defines the general characteristics of new mailboxes including the default mailbox, the number of messages a box can hold, and the number of messages a box can hold.
<a href="#">New Account Parameters</a>	Controls the default values of several mailbox fields such as message length, the number of messages a box can hold, and the number of messages a box can hold.
<a href="#">Attendant Parameters</a>	Controls how the Attendant features integrate to your phone system.
<a href="#">Directory Parameters</a>	Manages the Phone Directory parameters.
<a href="#">Logging Parameters</a>	Identifies the diagnostic tools and call logging data.
<a href="#">Holiday Schedules</a>	Defines holidays and closed days for custom greetings.
<a href="#">Xtra</a>	Groups Mailboxes and Audiotex modules by Account and Type categories.



## Managing Parameters - Attendant

### Overview

The Attendant Parameters enables you to customize the Attendant settings to your specifications.

To view/change Attendant Parameters, select **CALLMaster Manager | Parameters | Attendant**.

ID	Parameter	Value	Unit	Description
1	Allow Any Transfer	<input type="checkbox"/>	Yes/No	Allow transfers to extensions not listed in the Attendant
2	Complete Transfer Delay	1	seconds	Delay before dropping the line to complete the transfer of a call
3	Drop 'Busy' Call	8	string	The dial sequence to drop a Busy signal
4	Drop 'No Answer' Call	8	string	The dial sequence to drop an unanswered call
5	Extension Selection Timeout	8	seconds	Time allowed for making an Attendant selection before a transfer to the default extension is made
6	Fax Extension	109	seconds	Attendant ID designating the extension to transfer incoming fax calls to
7	Max Extension Length	4	TT count	Maximum count of Touch Tone digits for Attendant Extensions
8	Screened Name Maximum	5	seconds	Maximum number of seconds for callers to record their name for a Screened Transfer
9	Transfer String	8,	string	The DTMF sequence to initiate the transfer of a call, usually a flash hook (&#92)
10	Call Analysis on Default Transfer	<input checked="" type="checkbox"/>	Yes/No	Monitor default transfer for completion, busy or no answer conditions

Save & Exit Cancel

**Parameter:** Allow Any Transfer

**Description:** Transfer to extension not found in Attendant database?

**Default:** No

**Range:** No, Yes

**Usage:** This parameter controls whether the Attendant system checks a database of valid extensions before performing a transfer. When the value is 'yes,' the system will transfer without checking the extension's existence. This parameter set to 'no' will limit the extensions to which the system may transfer. When a caller requests a transfer, the system will check the Attendant database and transfer only if it finds the extension; otherwise, it will play prompt 506 (Sorry, there is no such extension). Messages will be directed to Default Mailbox if Attendant Parameter 'Allow Any Transfer' = Yes and 'Call Analysis on Default Transfer' = Yes if call analysis determines busy or no answer.

### Complete Transfer Delay

**Parameter:**

**Description:** Seconds to wait after dialing before hang-up

**Default:** 1

**Range:** 0-9 seconds

**Usage:** The system waits this long after performing a transfer before it exits the connection. If this value is too small, your PBX may determine that the transfer request was aborted.

#### **Drop 'Busy' Call**

**Parameter:**

**Description:** Transfer pullback sequence, busy

**Default:** & ,

**Range:** 0-9, \*, #, A-D (touch-tones), & (flash-hook), , (pause) and [ext]

**Usage:** Not applicable for IP Network.  
The system uses this string to retrieve a supervised or screened transfer if the extension is busy. The default value (& ,) means send a flash-hook and pause, which works for phone company lines and most switches; if the line has three-way calling, use two flash-hooks (&,& ,); see your phone system manual. **[ext]** provides for dialing the current extension's code for switches that require it as part of the pullback string, see above example.

#### **Drop 'No Answer' Call**

**Parameter:**

**Description:** Transfer pullback sequence, no answer

**Default:** & ,

**Range:** 0-9, \*, #, A-D (touch-tones), & (flash-hook), , (pause) and [ext]

**Usage:** Not applicable for IP Network.  
The system uses this string to retrieve a supervised or screened transfer if the extension is not answered or the call is rejected (the Dialogic parameter, NBRDNA, sets the number of rings to define no-answer). The default value (& ,) means send a flash-hook and pause, which works for phone company lines and most switches; if the line has three-way calling, use two flash-hooks (&,& ,); see your phone system manual. **[ext]** provides for dialing the current extension's code for switches that require it as part of the pullback string, see above example.

#### **Extension Selection Timeout**

**Parameter:**

**Description:** Time-out in main Attendant menu before transfer to operator

**Default:** 3

**Range:** 0-99 seconds

**Usage:** An Attendant system, after playing the main greeting, waits this long for a touch-tone response and then performs a default action, usually transferring to the operator. A value of zero will turn off the default transfer and require a touch-tone response.

### **Fax Extension**

**Parameter:**

**Description:** Attendant ID to transfer to for fax receive

**Default:** valid extensions

**Range:** valid extensions

**Usage:** Attendant ID on the PBX designated as the extension to transfer faxes to. If CALLMaster is not enabled for fax or specifically set-up to not accept faxes (Flag ATTNOFAX in the System Parameter 'System Settings' field) then CALLMaster will use this extension to transfer inbound fax received.

### **Max Extension Length**

**Parameter:**

**Description:** Length for valid extensions

**Default:** 4

**Range:** 1-4

**Usage:** This parameter takes effect only when 'Allow Any Transfer' is 'yes.' It tells the system how many digits to expect from callers trying to transfer to an extension, and controls how many digits the system sends during a transfer.

### **Screened Name Maximum**

Parameter:	
Description:	Time callers have to record their names
Default:	5
Range:	0-99 seconds
Usage:	Systems that have 'call-screening' ask the caller to record his name; then play that recording to the called party for acceptance. The caller has the amount of time listed in this field to record his name.

### **Transfer String**

Parameter:	
Description:	Transfer initiation sequence
Default:	& ,
Range:	0-9, *, #, A-D (touch-tones), & (flash-hook) and , (pause)
Usage:	Not applicable for IP Network. The system uses this string to initiate a transfer, and then enters the desired extension. The default value (& ,) means send a flash-hook and pause, which works for phone company lines and most switches. Refer to your phone system manual.

### **Call Analysis on Default Transfers**

Parameter:	
Description:	Monitor default transfers
Default:	no
Range:	yes/no
Usage:	Not applicable for IP Network. Monitor default transfer for transfer completion. Messages will be directed to Default Mailbox if

Attendant Parameter 'Allow Any Transfer'= Yes  
and 'Call Analysis on Default Transfer' = Yes if  
call analysis determines busy or no answer.

## Managing Parameters - Dialogic

### Overview

The Dialogic Parameters are provided to customize the Dialogic Board's response to your telephony environment and requirements. There are 93 parameters that can be configured. It is recommended that you keep the default values for most of these parameters and make changes only as instructed.

To view/change Dialogic Parameters, select **CALLMaster Manager | Parameters | Dialogic**.

- Dialogic Parameters Overview
- [Dialogic Parameters 1 thru 19](#)
- [Dialogic Parameters 20 thru 38](#)
- [Dialogic Parameters 39 thru 57](#)
- [Dialogic Parameters 58 thru 76](#)
- [Dialogic Parameters 77 thru 93](#)

**NOTE:** Dialogic Parameters are not applicable for IP Network.

## Managing Parameters - Directory

**Parameter:**     **Inter Digit Timeout**

**Description:**     Maximum seconds between tones for directory lookup

**Default:**             4

**Range:**                0-9 seconds

**Usage:**                The directory of box holders looks up box numbers based on a touch-tone spelling of the box holder's name. The touch-tones may be up to this many seconds apart. If more time passes, the system will assume the entry complete, and give the first matching name. Letters on the touch-tone keypad are harder to see than numbers, so this parameter should be longer than the related 'Prompt Inter DTMF Timeout', which applies to all other touch-tone entries.

# Managing Parameters - Holiday and Closed Day Calendar

## Overview

CALLMaster provides special call handling for weekends and holidays. It can answer with a special greeting and handle Attendant transfers differently. CALLMaster permits you to use special all day greetings on holidays or days that you are closed during the week.

The greeting prompts are specified in the 'Phone Ports' table.

- The Voice Mail Closed-Day greeting is prompt 203; the Voice Mail Holiday greeting is prompt 209.
- You may also create custom prompts numbered between 600 and 900 and reference in the Phone Ports record.

On days that are both closed-days and holidays, the holiday greeting will be played. **On closed days and holidays, the Attendant uses the 'evening' records if available.**

To instruct CALLMaster which holidays and days of the week you are closed, select **Parameters | Holidays** to bring up this screen.

The screenshot shows the 'CALLMaster Manager - [Closed Days and Holiday Schedule]' window. The window has a menu bar with 'File', 'Manage', 'View', 'Parameters', 'Reports', 'Windows', and 'Help'. The status bar at the bottom right indicates 'CALLMaster Service Status: Running'. The main content area is titled 'Closed Days and Holiday Schedule' and contains two sections: 'Days Closed' and 'Holidays Closed'.

**Days Closed:**

Sunday:	<input checked="" type="checkbox"/>
Monday:	<input type="checkbox"/>
Tuesday:	<input type="checkbox"/>
Wednesday:	<input type="checkbox"/>
Thursday:	<input type="checkbox"/>
Friday:	<input type="checkbox"/>
Saturday:	<input checked="" type="checkbox"/>

**Holidays Closed:**

12/25/2009  
1/1/2010

**Calendar:**

December, 2009

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Today: 12/6/2009

**Buttons:**

Add >>  
Remove <<  
Done

## Weekdays Closed

Check the appropriate boxes for days you wish to be considered as closed every week. On these days prompt 203 (Voice Mail) or prompt 515 (Attendant) is played.

## Holidays Closed



By choosing the appropriate date on the calendar, then clicking the corresponding date appears in the 'Holidays Closed' list box and is considered by CALLMaster to be a closed holiday. On these days prompt 209 (Voice Mail) or 516 (Attendant) is played.

## Managing Parameters - Logging

### Overview

The Logging Parameters control the log level and log file name.

To view/change System Parameters, select **CALLMaster Manager | Parameters | Logging**.

The screenshot shows the 'CALLMaster Manager - [Logging Parameters]' window. It has a menu bar with 'File', 'Manage', 'View', 'Parameters', 'Reports', 'Windows', and 'Help'. A toolbar contains various icons. The status bar at the bottom right shows 'CALLMaster Service Status: Running'. The main area is titled 'Logging Parameters' and contains a table with the following data:

#	Parameter	Value	Unit	Description
1	Log File:	C:\Program Files\CALLMaster\Logs\CMCalls	path	File name
2	Log Level:	0	path	Level of detail
3	Flags:		flags	Feature flags

Below the table are two buttons: 'Save & Exit' (green) and 'Cancel' (red).

**Parameter:** Call Log File

**Description:** File for log information

**Default:** CMCalls\_<DATE>.TXT

**Range:** any valid filename

**Usage:** Application activity is written to this file. Level of detail information written to this file is controlled by the Trace Level parameter defined below. It includes call start and stop times, as well as mailbox and Audiotex activity. To automatically discard logging information, set the filename to **null**. Deleting this information on a periodic basis is important if your system has a lot of activity. Every day the system will create a new file with the date in the file name.

**Parameter:** Trace File

**Description:** File for log information

**Default:** NULL

Range: any valid filename

Usage: Not used.

Parameter: **Trace Level**

Description: Level of log output

Default: 0

Range: 0-3

Usage: This parameter controls diagnostic detail printed in the CMCalls\_<Date>.txt log file. Zero results in the minimum logging and should be used unless a specific problem is encountered. In that case, the logging should be set to one (and if necessary two) and the problem duplicated and then solved. Delete the log file and reset this parameter to zero when finished so as not to consume disk space. **Level three will soon result in a full hard drive.**

## Managing Parameters - New Mailbox Default

### Overview

The New Mailbox Parameters enables you to set the default values for new Mailboxes.

To view/change New Mailbox Default Parameters, select **CALLMaster Manager | Parameters | New Mailbox Default**.

#	Parameter	Value	Unit	Description
1	Max Days Msgs are stored:	30	days	The number of days msgs are saved before being automatically deleted
2	Notification Flags:	0	bit mask	Notification / Login settings
3	Max Msgs Length:	300	secs	Maximum length in seconds of messages
4	Max New Msgs:	50	count	Maximum New messages stored before full
5	Max Saved Msgs:	50	count	Maximum New messages stored before full

**Parameter:** Days Message Stored

**Description:** Default maximum age for messages

**Default:** 30

**Range:** 0-999 days

**Usage:** New voice mailboxes are created with this message age limit. Both this value and the message age limit for an individual box may be changed over the phone.

**Parameter:** Default Properties

**Description:** Default flags setting

**Default:** 00

Range: 00 - 0F

Usage: New mailboxes are created with this flag. Refer to the Mailbox database Message Notification Properties field.

Parameter: **Max Message Length**

Description: Default maximum length of messages

Default: 300

Range: 0-9999 seconds

Usage: New voice mailboxes are created with this message length limit. Set this field to zero (0) for unlimited record time. Both this value and the message length limit for an individual box may be changed over the phone. **This field sets the value for the allowable length of public distribution list messages.**

Parameter: **Max New Messages**

Description: Default maximum new messages

Default: 50

Range: 0-99 messages

Usage: New voice mailboxes are created with this new message limit. Both this value and the new message limit for an individual box may be changed over the phone.

Parameter: **Max Saved Messages**

Description: Default maximum saved messages

Default: 50

Range: 0-99 messages

Usage: New voice mailboxes are created with this saved message limit. Both this value and the

saved message limit for an individual box may be changed over the phone.

## Managing Parameters - System

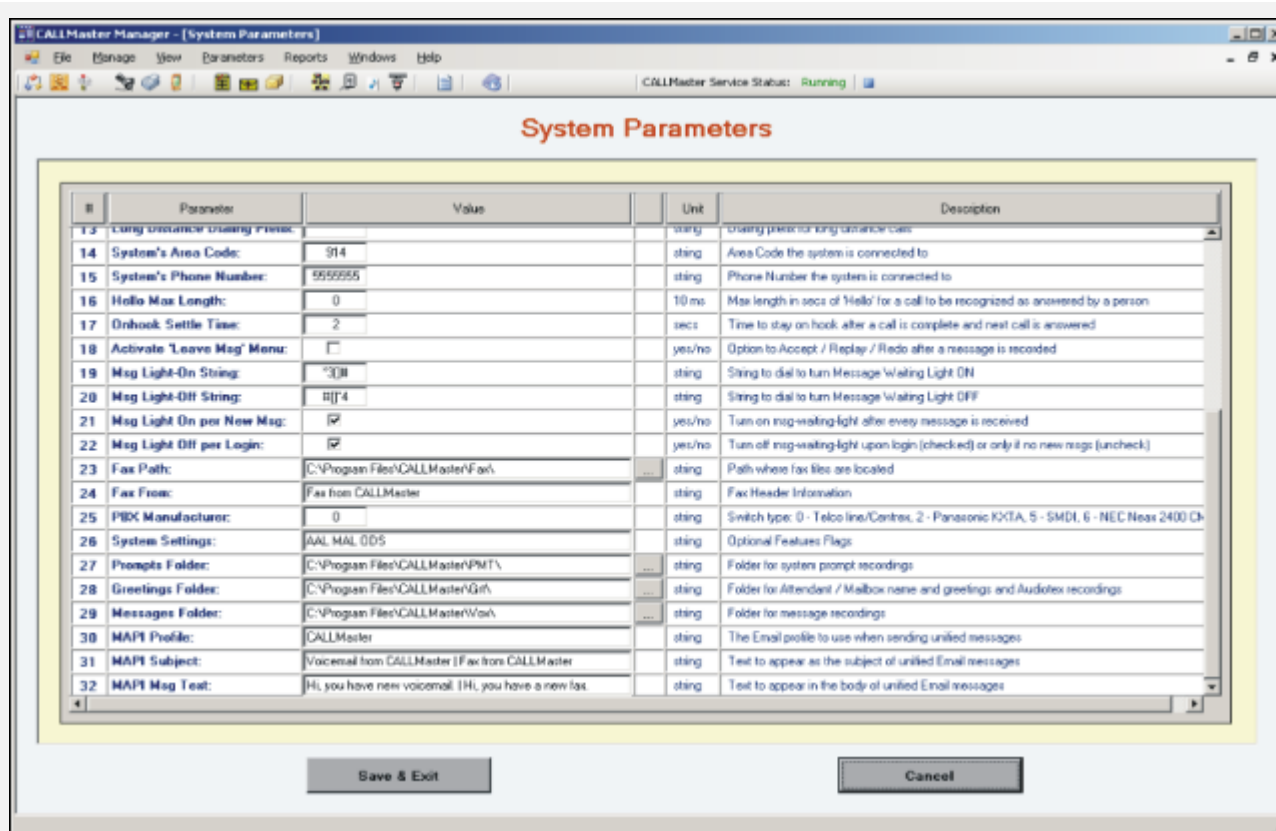
### Overview

The System Parameters enables you to customize the CALLMaster system settings to your specifications.

To view/change System Parameters, select **CALLMaster Manager | Parameters | System**.

ID	Parameter	Value	Unit	Description
1	Morning starts at:	9:00 AM	time	Time of day to start playing the morning greeting
2	Afternoon starts at:	12:00 PM	time	Time of day to start playing the afternoon greeting
3	Evening starts at:	5:00 PM	time	Time of day to start playing the evening greeting
4	Prompt Inter DTMF Timeout:	4	sec(s)	Maximum time between Touch-Tone digit entries
5	Prompt Repeat Count:	3	count	Maximum New messages stored before full
6	Prompt Response Timeout:	10	secs	How many seconds a user has to respond to an input request
7	Playback Rewind/FF rate:	5	secs	Number of seconds to fast forward or rewind a message during playback
8	Record Max Silence:	15	secs	Maximum continuous silence to cause recording to stop
9	Record Format:	WAV	string	Audio Recording Format: wav (default) or vox
10	Start Calling at:	12:00 AM	time	Time of the Day at which to start making out dialing calls
11	Stop Calling at:	12:00 AM	time	Time of the Day at which to stop making out dialing calls
12	Local Dialing Prefix:		string	Dialing prefix for local calls
13	Long Distance Dialing Prefix:		string	Dialing prefix for long distance calls
14	System's Area Code:	914	string	Area Code the system is connected to
15	System's Phone Number:	5555555	string	Phone Number the system is connected to
16	Hello Max Length:	0	10 ms	Max length in secs of 'Hello' for a call to be recognized as answered by a person
17	Onhook Settle Time:	2	secs	Time to stay on hook after a call is complete and next call is answered
18	Activate 'Leave Msg' Menu:	<input type="checkbox"/>	yes/no	Option to Accept / Replay / Redo after a message is recorded
19	Msg Light-On String:	*30#	string	String to dial to turn Message Waiting Light ON
20	Men Light-On String:	#1#	string	String to dial to turn Message Waiting Light OFF

Save & Exit Cancel



**Parameter:** 1 - Morning Starts At

**Description:** Hour after which morning greeting should be played

**Default:** 6:00:00

**Range:** 0:00:00 to 23:59:00 military time

**Usage:** After this time, the system plays the morning greeting. Time is expressed in a twenty-four hour format. The Voice Mail morning greeting is prompt 19. The Attendant morning greeting is prompt 502.

**Parameter:** 2 - Afternoon Starts At

**Description:** Time after which afternoon greeting should be played

**Default:** 12:00:00

**Range:** 0:00:00 to 23:59:99 military time

**Usage:** After this time, the system plays the afternoon greeting. Time is expressed in a twenty-four



hour format: twelve is noon, thirteen is one p.m. and zero is midnight. The Voice Mail afternoon greeting is prompt 20. The Attendant afternoon greeting is prompt 503. In order to keep things straight, we recommend using the proper prompt greeting for your application(s).

**Parameter: 3 - Evening Starts At**

**Description:** Time after which evening greeting should be played

**Default:** 18:00:00

**Range:** 0:00:00 to 23:59:00 military time

**Usage:** After this time, the system plays the evening greeting. Time is expressed in a twenty-four hour format. The Voice Mail evening greeting is prompt 21. The Attendant evening greeting is prompt 504.

**Parameter: 4 - Prompt Inter DTMF Timeout**

**Description:** Maximum time between digit entries

**Default:** 2

**Range:** 1-99 seconds

**Usage:** This parameter specifies the time allowed between touch-tones of a multi-tone response. When that time elapses, the system acts on the touch-tones received. Inter Digit Timeout, a Directory parameter, controls the timing in the directory of box holders separately.

**Parameter: 5 - Prompt Repeat Count**

**Description:** Number of times a prompt is repeated before logout

**Default:** 3

**Range:** 0 to 9

**Usage:** Some system prompts permit a touch-tone response from the caller. When a caller fails to respond, or responds in an invalid way, the prompt is repeated this many times. The system will then give a warning and hang up.

**Parameter: 6 - Prompt Response Timeout**

**Description:** Time-out for prompts, in seconds

**Default:** 10

**Range:** 0-999 seconds

**Usage:** The system will wait this long after a prompt for a caller's touch-tone response. It will then repeat the prompt as many times as specified in 'Prompt Repeat Count'. A value of zero will result in an infinite wait. The 'Prompt Response Timeout' parameter does not control the time-out after the main greeting, which is controlled by either 'Log in Timeout' or 'Extension Selection Timeout', depending on port configuration.

**Parameter: 7 - Play FF/REW Rate**

**Description:** Amount of message skipped by rewind and forward

**Default:** 5

**Range:** 1-99 seconds

**Usage:** During Review messages, the box holder may skip backward or forward in the message by pressing one (1) or three (3), respectively. This parameter controls how much of the message is skipped. The default value, five seconds, is about how long it takes to say a phone number.

**Parameter: 8 - Record Max Silence**

**Description:** Silence time-out for recording, in seconds

**Default:** 15

**Range:** 0-999 seconds

**Usage:** The system will stop recording a voice message if it detects silence for this period of time. If messages are being cut off when the speaker pauses, increase the value. If messages have too much dead silence or background noise at the end, decrease the value. If you have problems with hang-ups during the greeting and it doesn't hang up before a message is left, make sure you are getting a disconnect signal (loop current drop from the phone company or

touch-tone or global tone disconnect from your phone system).

**Parameter:**    **9 - Record Format**

**Description:**   Recording file format

**Default:**        wav

**Range:**           wav

**Usage:**           The recording format for messages, prompts, greetings is wav - windows native 11 Khz linear 8bit mono format.

**Parameter:**    **10 - Stop Calling At**

**Description:**   Auto shutdown hour

**Default:**        0:00:00

**Range:**           0:00:00 to 23:59:00 military time

**Usage:**           Time after which the system will not make out bound calls. The time is based on the system time of the computer you are using. If 'Stop Calling at' and 'Start Calling at' are both equal to zero (0), the system will continuously make calls.

**Parameter:**    **11 - Start Calling At**

**Description:**   Auto start hour

**Default:**        0:00:00

**Range:**           0:00:00 to 23:59:00 military time

**Usage:**           Time at which the system starts making out bound calls. The time is based on the system time of the computer you are using. If 'Stop Calling at' and 'Start Calling at' are both equal to zero (0), the system will continuously make calls.

**Parameter: 12 - Local Dialing Prefix**

**Description:** Dial prefix for local calls

**Default:** “

**Range:** “, 0-9, \*, #, A-D (touch-tones), & (flash-hook) and, (pause)

**Usage:** The system uses this prefix for local calls. A message notification call or an Out Dial is considered ‘local’ if its area code matches the ‘System’s Area Code’. The prefix and the number are dialed; the area code is not dialed.

**Parameter: 13 - Long Distance Dialing Prefix**

**Description:** Dial prefix for long distance calls

**Default:** 1

**Range:** “, 0-9, \*, #, A-D (touch-tones), & (flash-hook) and, (pause)

**Usage:** The system uses this prefix for long distance calls. A message notification call or an Out Dial is considered ‘long distance’ if its area code does not match the ‘System’s Area Code’. The prefix, the area code, and the number are dialed.

**Parameter: 14 - System’s Area Code**

**Description:** System dialing area code

**Default:** 914

**Range:** 0-999

**Usage:** The area code where the system is located. During message Notification (except message lights) and Out Dialing, the phone number to call is compared with this parameter to determine whether the local or long distance dial prefix should be included.

**Parameter: 15 - System’s Phone Number**

**Description:** Callback number for pagers

**Default:** 9145555555#

**Range:** 0-9, \*, #, A-D (touch-tones), and & (flash-hook) and, (pause)

**Usage:** If a message Notification call is made to a pager, this number will be sent to the pager. Many pagers require a pound (#) to terminate.

**Parameter: 16 - 'Hello' Max length**

**Description:** Maximum Out Dial hello length in 1/100 second; 0 for no hang-up

**Default:** 0

**Range:** 0-99,999 1/100 seconds

**Usage:** In an Out Dialing application, after the call has been answered, the system compares the length of talking or other noise to this parameter; if the parameter is exceeded, the system disconnects. The default value, zero (0), disables the disconnect feature. This parameter can be used to help detect an answering machine.

**Parameter: 17 - Onhook Settle Time**

**Description:** Time to stay on hook after finishing a call

**Default:** 2

**Range:** 0-99 seconds

**Usage:** This parameter controls the number of seconds to remain on hook after finishing a call the system initiated. It gives the telephone company and switch time to disconnect and reset the line.

**Parameter: 18 - Activate 'Leave Msg' Menu**

**Description:** Accept/replay/redo (append) menu after leaving message

Default:	No
Range:	No, Yes
Usage:	This parameter controls whether someone leaving a message will be given the opportunity to accept, replay, redo (or append to) their message (prompt 3). Messages left will be saved either way, unless the caller (or the phone system) presses pound (#).
Parameter:	<b>19 - Msg-Lite-On Dial String</b>
Description:	String to dial to turn message waiting light on
Default:	*3[]#
Range:	0-9, *, #, A-D (touch-tones), & (flash-hook) and, (pause)
Usage:	The system uses this string to turn on message waiting lights. It substitutes the 'Dial Prefix' field from Notification Record 0 in place of the brackets. For example, the 'Msg-Lit-On Dial String' *3[]# means dial *3 then the extension number, followed by # to turn on a message light. If this parameter is blank (null) the message waiting lights will not be turned on. <b>The code in this field should be the same code that a single-line phone user would use to light the message waiting light.</b>
Parameter:	<b>20 - Msg-Light-Off Dial String</b>
Description:	String to dial to turn off message waiting light
Default:	#[]*4
Range:	0-9, *, #, A-D (touch-tones), & (flash-hook) and , (pause)
Usage:	The system uses this string to turn off message waiting lights. It inserts the Dial Prefix field from Notification Record 0 in place of the brackets. For example the 'Msg-Lit-Off Dial_String' #[]*4 means dial # then the extension number, followed by *4 to turn off a message light. If this parameter is blank (null) the message waiting lights will not be turned off. <b>The code in this field should be the same code that a single-line phone user would use to deactivate the message waiting light.</b>
Parameter:	<b>21 - Msg-Light-On Per Msg</b>

**Description:** Turn on message waiting light after every message

**Default:** No

**Range:** Yes, No

**Usage:** This system parameter determines whether the message light is activated after every message, or only after the first message received following a log in.  
Yes - Turn on after every message received.  
No - Turn on only at the first message since the light was turned off.

**Parameter:** **22 - Msg-Light-Off Per Log In**

**Description:** Turn off message waiting light upon log in

**Default:** Yes

**Range:** Yes, No

**Usage:** This parameter determines whether the 'Msg-Light-Off Dial' String is sent upon a log in. Setting the parameter to 'No' means the message light will be extinguished only after the new message count becomes zero.  
Yes - Light out when box holder logs into their mailbox.  
No - Light out when new messages = 0 (zero)

**Parameter:** **23 - Fax Path**

**Description:** Folder where fax files are located

**Default:** None

**Range:** Any valid folder

**Usage:** This parameter specifies where the fax documents are located, including the path itself. For example, \Program Files\CALLMaster\faxes\price.RAW  
If nothing is entered in this field, the fax software's default path will be used. This field is not related to the path field in the fax database.

**Parameter:** **24 - Fax From**

**Description:** Header information (32 characters maximum)

**Default:** None

**Range:** Up to 32 characters

**Usage:** This is header information that will appear on faxes. This field is superseded by the fax database 'From' field.

**Parameter:** **25 - PBX Manufacturer**

**Description:** Switch integration

**Default:** 0

**Range:** 0-99

**Usage:** The parameter controls special switch integration procedures. 0=Centrex; 2=Panasonic; 5=SMDI, 6=NEC NEAX 2400 MCI serial link integration

**Parameter:** **26 - System Settings**

**Description:** Switch initialization settings

**Default:** Null

**Range:** field can contain multiple values as specified below:

**Usage:** Multi-purpose. Set field to various flags to control system behavior. Each flag should be separated by a space.  
e.g. ABS ADT115 ASRMXR100

Value	Description
-------	-------------

<b>AAL</b>	Use this flag to enable logging to the Attendant Activity Log (AttendantActivityLog).
------------	---

<b>ABS</b>	Announce Bridged Call Start. Use this flag to play prompt 564 to announce start of bridged call.
------------	--



<b>ADT&lt;ext#&gt;</b>	Alternate Default Transfer where <ext#> is the Attendant Id to be used instead of Attendant Id 0 when caller does select an attendant input.
<b>ASRMXR&lt;x&gt;</b>	ASR Maximum Recording where x is the max length of the ASR recording in MS units. Default is 100. Recording greater than x will be cut-off.
<b>ASRMXS&lt;y&gt;</b>	ASR Maximum Silence where y is timeout for response in 10 MS units. Default is 10.
<b>ASRTONE</b>	ASR tone (prompt# 566) will be played before the recognition recording starts. Otherwise no tone is played.
<b>ATTNOFAX</b>	Attendant No Fax will prevent default fax reception on fax calls and instead will just transfer the call to the "fax extension" defined in the Attendant parameters.
<b>AUAC</b>	Always Use Area Code when Out Dialing regardless of the system area code.
<b>BRG&lt;Bridge Line #&gt;</b>	Used to indicate a bridged call. The Line# can identify a single line, a range of lines or any available line used to bridge the call. e.g. BRG2 would bridge call using line 2; BRG4-8 would use any lines from 4 to 8 to bridge call; BRG would use any available line.
<b>CLO</b>	Use this flag to disable logging to the Call Log table (CALLLog).
<b>DCAT</b>	Don't Confirm Alternate Transfers. If this flag is set then CALLMaster will not play prompt "Extension is busy. Please hold while we transfer your call" a second time after the first transfer fails and Failed Transfer Module Action is initiated.
<b>DCT</b>	Don't Confirm Transfers. CALLMaster will not play prompt 521 'Please hold while I transfer you to ...' when transferring caller.
<b>DML</b>	Display Module Label. CALLMaster Main Window Activity column will display the literal from the 'Audiotex Label' field to monitor the call flow.
<b>EMAIL&lt;address&gt;</b>	Flag to enable emails. The <address> is optional to specify a specific address to email start-up notification. If the optional email address is not specified then startup emails are sent to <a href="mailto:test_email@callmaster_startup.com">test_email@callmaster_startup.com</a> . Multiple addresses can be set-up separated by a semi-colon (e.g. EMAILsales@speechsoft.com;support@speechsoft.com)
<b>EMDP&lt;x&gt;</b>	Email Mailbox Digits Printed where the last <x> digits of the mailbox number are printed for mailbox ID in email notification.
<b>FAL</b>	Use this flag to enable logging to the FaxSendActivityLog table.
<b>FCSA</b>	Transfer control to the Audiotex module even if OutcallSchedule fails to connect call.

<b>GC2BCT</b>	Used for T1 PRI call transfer in conjunction with BRG code. Works similar to Centrex transfer for analog, lines. The call starts just like a bridged call but then invokes a transfer ISDN packet to instruct the CO to perform the connection. Once connected the two channels used are freed up to handle new calls.
<b>GCANIMBOX</b>	Use the ANI to set message waiting lights on/off without requiring notification records. The mailbox number in these cases is set to the ANI received by the system.
<b>GC1@ATIME</b>	Stagger start of successive lines only on completion of previous line.
<b>GCDELAYxx</b>	Wait time where xx is the time in milliseconds to wait between starting successive lines.
<b>GCDMSOUT</b>	Required for Nortel DMS switch to support T1 Global Call Outbound.
<b>GCERRORLOG</b>	Send additional error log information to the CMCalls log file.
<b>GCIDIS</b>	Global Call Ignore Disconnect in Idle State. If set, this will not do a line reset if a disconnect is received in the idle state.
<b>GCSCSQUIT</b>	Resets/terminates CALLMaster service if a GCEV_SETCHANSTATE event is received.
<b>GCT1-&lt;T1 protocol&gt;</b>	Defines the T1 protocol to be used. <T1 protocol> specifies the T1 Protocol (for US T1 protocol this value is US_MF_IO).
<b>GCTRACEXX</b>	ISDN packet trace called GCTRACEXXLOG where XX is the board number 1-99. Only one board can be traced at a time. All ISDN packet sent and received for all channels on that board are captured. The log file is binary and needs to be converted to ASCII with the Dialogic utility ISDNTRACE.EXE.
<b>GCWAITSTART</b>	Wait for an unblocked event at start-up. This is only for GLOBALCALL for the FXS/FXO protocol.
<b>GLOBALCALL</b>	Enable Dialogic Global Call Libraries (for T1& E1 Analog & ISDN-PRI service).
<b>LNGDEF&lt;#&gt;</b>	Identifies the default language the call starts in. # is an integer from 1 to 9. Used in conjunction with the LNG<#xxx> flag.
<b>LNG&lt;#xxx&gt;</b>	Flag to define the multi-lingual language set. # is the language id and xxx contains the language folder e.g. LNG1SPN for language 1 as Spanish, LNG2CHN for language 2 as Chinese, LNG0 is always English.
<b>MAL</b>	Use this flag to enable logging to the MailboxActivityLog table.

<b>NDT</b>	No Default Transfer where system will not use any attendant record when caller does select an attendant input. System will just return to the calling point if this indicator is used.
<b>NMR</b>	No Mailbox Request. If mailbox login failure, suppress mailbox request for login retry.
<b>NOATTASR</b>	No Attendant ASR. If ASR feature purchased, this flag will disable attendant ASR transfer capability only enabling ASR for CALLBasic commands.
<b>NTFYOFF</b>	Turn off notification threads from running NotifyHandler and OutcallHandler when not needed (i.e. Mailbox Notification functionality not being used).
<b>NTR</b>	No Touch Tone Repeat. Used with Module Action 'Touch Tone Table'. This will not prompt caller to "Press * to repeat" if caller does not make Touch Tone Table selection.
<b>OCHLOG, OCHLOG2</b>	Enable OutCall Handler Logging to log detailed Notification activity to debug stopping.
<b>OCNV</b>	Outcall No Voicemail. All ports to remain in Out Dial mode at the end of processing phone list. Enables Out Dial applications to continuously check the OutcallSchedule table for numbers to call.
<b>OSRCR</b>	Automate move of completed and expired entries from the OutcallSchedule to the OutcallScheduleCompleted table.
<b>OSICT&lt;x&gt;</b>	OS Inter Call sleep Time where <x> is the time to sleep in milli seconds between dispatching calls. The default is 10ms if no flag is present. Simultaneous GCMakecall events on multiple lines caused some PBXs to hang requiring a delay.
<b>OSST&lt;x&gt;</b>	OutcallSchedule Sleep Time where <x> is the time to override the default 30 sec sleep time when checking if OutcallSchedule records are available to be called by the dispatcher.
<b>OTE</b>	Enable Outbound Transfers for the individual mailbox.
<b>OTTB&lt;port-range&gt;</b>	Outbound Transfers where port range is the ports to use for the bridged call. See Bridged attendant transfers for details.
<b>OTT3&lt;3waystring&gt;</b>	Outbound Transfers where 3way string is used to create and drop a 3way call.
<b>OTPS&lt;prefix string&gt;</b>	Outbound Transfer Prefix Dial string. Used with Outbound Transfers.
<b>OTSS&lt;suffix string&gt;</b>	Outbound Transfer Suffix Dial string. Used with Outbound Transfers.

<b>PBXCID</b>	For integration to D/82JCT-U PBX Integration Board.
<b>PBXM1</b>	Used with Digital PBX cards and the Nortel Meridian 1 to send flash hook at end of transfer complete and for blind transfers.
<b>PBXME1&lt;secs&gt; &lt;s&gt;</b>	Will wait <secs> seconds for valid caller ID data trying every 1/2 sec to get some ID data. Used in conjunction with digital PBX cards to tell the port to check the line indicators for a pending call (ringing) after a call is terminated and before going to wait for a ring. A race condition created a scenario where a line was ringing but the system did not pickup. A delay will also tell the port to wait for caller ID a specified amount of seconds before answering if it sees a ringing condition. If no ringing condition is detected the line will go into the normal Wait for Ringing state.
<b>PBXMITEL</b>	Instructs the system not to sent <esc>O code just the default light string 41# on and 42# off.
<b>PBXMSGGLITE OFFHK</b>	Required to enable message lights for Nortel Norstar.
<b>PFM</b>	PRI Facility Message. This flag will cause channel to wait for a Facility Message event in order to retrieve Caller ID name info for T1/PRI circuits. This is performed right after the call is in connect state but before application starts.
<b>PRN</b>	PRI Redirect Number support for PRI IE for transfer reason for Cisco Call Manager.
<b>PTM&lt;x&gt;</b>	Port Max where x is the maximum # of lines to start up. Only applicable if software license used.
<b>SAR</b>	Enables Strict Automatic/Speech Recognition. This will require a match with the XML Grammar file. If this flag is excluded, in Attendant mode using Speech Recognition, if the caller's request cannot be matched, the call will be transferred to Attendant ID 0.
<b>SHAREDFA[X] L#-L#]</b>	Enables lines to share fax lines using Channel Bridging. [L#-L#] is a range of fax lines. E.g. SHAREDFA[X]5-8 to share fax lines 5 thru 8.
<b>SOGCF</b>	Stop line on GetConnect failure exception. If not set then line will be set out of service.
<b>SOUCE</b>	Stop On Unknown Channel Exception. Determines if an Analog channel is taken out of service if there is an unknown exception.
<b>SRL</b>	Use this flag to enable logging to the Speech Recognition Activity Log (SRResultsLog).
<b>SSM</b>	Start/Stop MapiAgent. Not required from CALLMaster version 3.0.4.0. since CMMApiAgent no longer used for email processing.
<b>STODG</b>	Suppress Time Of Day Greeting on Audiotex ports. Application greeting can play Attendant or Voice Mail greeting for PBX integration.

**TTR** Touch Tone Repeat. Used with Module Action 'Touch Tone Table'.

**TBC#** Timed Bridge Call where # is number of seconds. This flag specifies the time limit for bridged call to be # seconds. A 2 second warning is played (prompt 563) to notify parties.

**UDPIP<IPaddress>:<Port#>** Activates UDP communications in support of CALLBasic SendUDP command.

**Parameter:** **27 - Prompts Folder**

**Description:** System Prompt folder

**Default:** c:\Program Files\CALLMaster\pmt

**Range:** valid folder name

**Usage:** This is the location of the folder where the CALLMaster prerecorded prompt files are located.

**Parameter:** **28 - Greetings Folder**

**Description:** System Greetings folder

**Default:** c:\Program Files\CALLMaster\grt

**Range:** valid folder name

**Usage:** This is the location of the folder where the Audiotex messages, greetings and name files are located.

**Parameter:** **29 - Messages Folder**

**Description:** System Message folder

**Default:** c:\Program Files\CALLMaster\vox

**Range:** valid folder name

Usage:	This is the location of the folder where the voicemail message files are located.
Parameter:	<b>30 - MAPI Profile</b>
Description:	Used to configure the CDO email mode and settings. You can select from five (5) modes to send emails.
Default:	CALLMaster
Range:	See <a href="#">Activating Email</a> for the various modes and setting for each mode.
Usage:	The Mode and associated setting determines how you want CALLMaster to process emails. e.g. whether you want to utilize your SMTP Servers via the domain name or IP address, if you want to use the Exchange server and identify the pickup folder and/or if you want to utilize a back-up folder to store all emails that were attempted to be sent up CALLMaster. See <a href="#">Activating Email</a>
Parameter:	<b>31 - MAPI Subject</b>
Description:	MAPI Subject
Default:	'Voice Mail from CALLMaster   Fax from CALLMaster'
Range:	text field
Usage:	Allows separate subject text for emails sent as a result of voice mail and fax receipt. The text to the left of the vertical bar ( ) will appear in the subject field of emails CALLMaster sends as a result of voice mail. The text to the right of the vertical bar ( ) will appear in the subject field of emails CALLMaster sends as a result of fax receipt.
Parameter:	<b>32 - MAPI Message Text</b>
Description:	MAPI message text
Default:	'Hi, you have a new voicemail   Hi, you have received a new fax'
Range:	text field

**Usage:**

Allows separate text for body of emails sent as a result of voice mail and fax receipt.

The text to the left of the vertical bar (|) will appear in the body of emails CALLMaster sends as a result of voice mail.

The text to the right of the vertical bar (|) will appear in the body of emails CALLMaster sends as a result of fax receipt.

## Managing Parameters - Voice Mail

### Overview

The Voice Mail Parameters enables you to customize Voice Mail settings to your specifications.

To view/change Voice Mail Parameters, select **CALLMaster Manager | Parameters | Voice Mail**.

#	Parameter	Value	Unit	Description
1	Default Mailbox:	0000	digits	Default mailbox for messages
2	Login Timeout:	30	secs	Time in seconds for mailbox log in
3	Mailbox Length:	4	count	Maximum length in Touch Tone digits of a Mailbox
4	Default Flags:		flags	APM - AutoPlay Msg, DTF - DateTime First, ASM - AutoSave Msg, NSD - No Seconds in
5	Message Recording Delay	0	secs	Delay after mailbox greeting is played and before recording starts
6	Minimum Message Length:	0	secs	Minimum length of a message in seconds for it to be considered valid for storage
7	Password Length:	4	count	Maximum length of a password in Touch Tone digits
8	Inband Tones Timeout:	0	secs	Time to wait before Voice Mail menu to receive tones from PBX
9	'Send to Another' Timeout:	2	secs	Timeout for sending a message to another mailbox

Save & Exit Cancel

**Parameter:** Default Mailbox

**Description:** Account ID for messages left by outside callers.

**Default:** 0000

**Range:** Any valid mailbox number

**Usage:** CALLMaster uses this mailbox by default unless directed to another box by touch-tones from the caller or the PBX. In systems where the caller cannot reach the voice mail main menu, this parameter is meaningless. Messages will be directed to Default Mailbox if Attendant Parameter 'Allow Any Transfer' = Yes and 'Call Analysis on Default Transfer' = Yes and the call result determines busy or no answer.

### Login Timeout

**Parameter:**

**Description:** Time-out for main prompt.



**Default:** 5

**Range:** 0-999 seconds

**Usage:** After the main voice mail greeting, the system waits this long for a touch-tone response and then performs a default action, usually recording a message. The value should be increased if callers need more time to consider their response or if their touch-tones are so weak or short that they cannot cut through while the greeting is being played. A value of zero will result in an infinite wait: no messages will be taken. (See also 'Prompt Response Timeout' and 'Extension Selection Timeout')

### Mailbox Length

**Parameter:**

**Description:** The number of digits required for the mailbox number.

**Default:** 4

**Range:** 1-10

**Usage:** All mailbox numbers must have the same number of digits. Four is the most commonly used but if for some reason you need a longer or shorter number, you can change that setting here.

**Parameter:** **Default Flags**

**Description:** This field contains indicators used to set default messaging attributes (e.g. 'BRS'=Browse, 'NMF'=New Messages First; 'DTF'=Date & Time First).

**Default:** <null>

**Range:** A single code or combination of 2 or more codes can be placed in the field as a string (e.g. NMF DTF).

- ASM** Auto Saves Messages during message review when caller enters 6 to review next message.
- APM** Auto Play Message assumes the caller pressed 1 to start message review process.
- DTF** plays the Date, Time and Sender of a message first before the message is played back.
- NEML** No Exclusive Mailbox Log In. The default is yes.
- NGN** No Greeting and Name recorded during initial mailbox log in.
- NMF** plays the Newest Message First, instead of the oldest first. NMF is the default.
- NRE** will reset notification schedule, starting notification from notification record number 1. Only applicable if you have multiple notification records per mailbox.
- NSD** No Seconds for message length played during message date and time. Only applicable when using DTF code.
- OMF** plays the Oldest Message First, instead of the newest first. NMF is the default.

**TSP** Toll Saver Prompt used for DID trunks. CALLMaster will play prompt 401 to denote no messages in the mailbox.

**Usage:** Default messaging attributes

#### **Message Recording Delay**

**Parameter:**

**Description:** Seconds before tone for message recording

**Default:** 0

**Range:** 0-99 seconds

**Usage:** This parameter controls how long the system waits between the prompt 'record a message after the tone' and the record tone. Zero, the default value means the tone is played immediately after the prompt. If callers have trouble distinguishing the record tone, increase Message Recording Delay to one or two seconds.

#### **Min Message Length**

**Parameter:**

**Description:** Minimum message length

**Default:** 0

**Range:** 0-99 seconds

**Usage:** Minimum length of a message for it to be considered valid for storage. A setting of 0 allows for any message length.

#### **Password Length**

**Parameter:**

**Description:** The maximum length a password can be.

**Default:** 4

**Range:** 0-10

**Usage:** Since this entry is terminated by pressing the \* key, passwords may be of variable length. However, you may want to set the maximum length that the password may be.

### **Inband Signal Wait**

**Parameter:**

**Description:** Time to wait for tones from PBX in main greeting

**Default:** 0

**Range:** 0-9 seconds

**Usage:** Systems with full PBX integration may require a brief pause after the line connects and before playing the greeting. Where the PBX's tones are very short or very fast, the first tone may not be heard clearly by the system. A pause of one or two seconds will allow the system to clearly hear the first tone; then it will not play the main greeting, but will listen for additional tones.

### **'Send Another' Timeout**

**Parameter:**

**Description:** Time-out for sending to another user

**Default:** 2

**Range:** 1-9 seconds

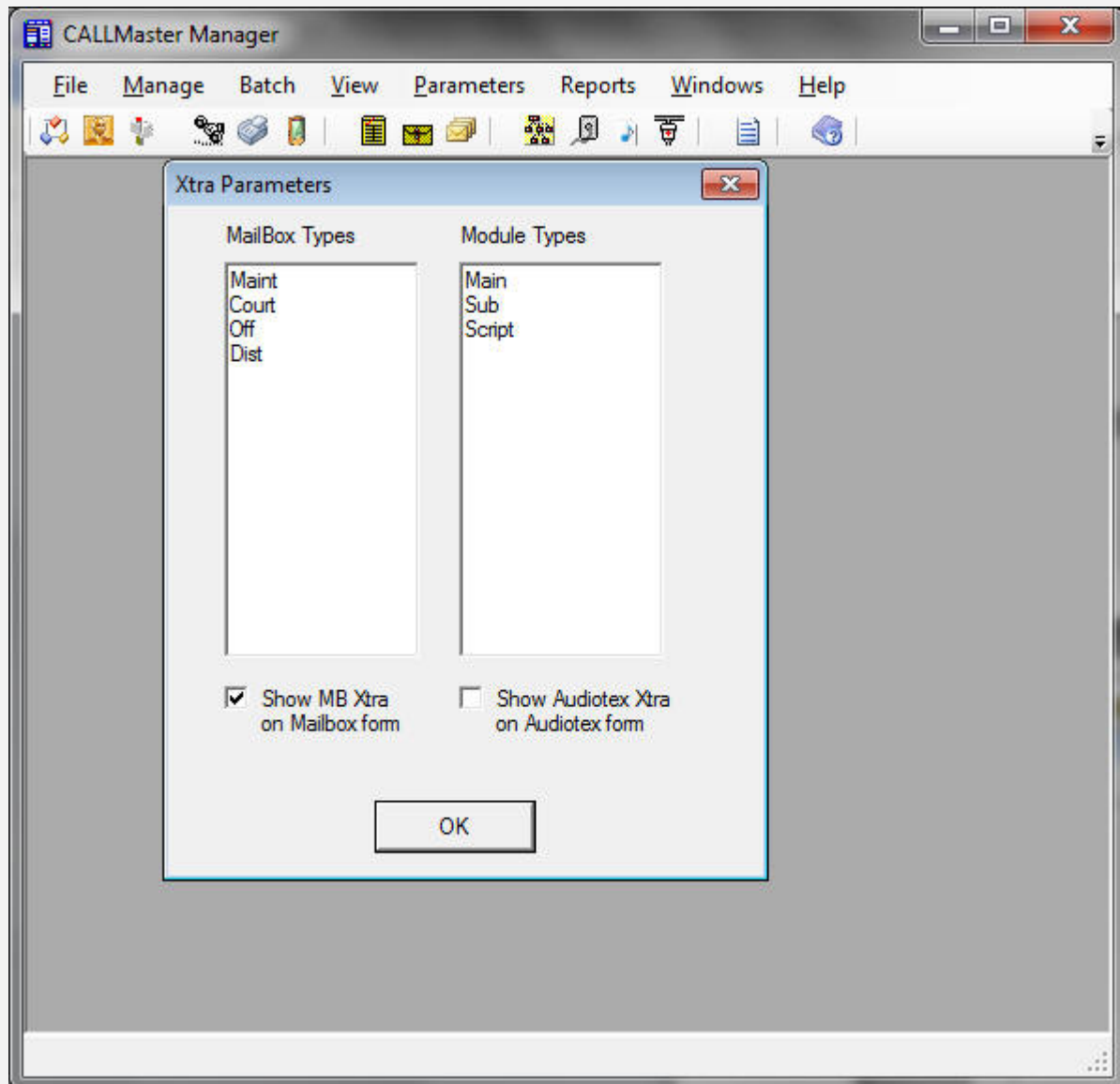
**Usage:** After sending a message, a box holder may also send that message to additional boxes. After approving his message, he will hear a prompt to 'press star to send to another box' and has this much time to do so before returning to the main menu. A value of zero will cause an infinite wait.

# Managing Parameters - Xtra

## Overview

CALLMaster provides extra fields to group Mailboxes and Audiotex modules into associated account. This grouping is optional and controlled by the Xtra parameter.

To use this grouping, select **Parameters | Xtra** to bring up this screen.



Check the appropriate box if you wish to use the Xtra fields for the Mailboxes and/or Audiotex records.

Just add, delete or modify the type list on this screen. You will get the values from this list as a drop down when updating Mailboxes and/or Audiotex Records.

Check "Show MB Xtra" or "Show Audiotex Xtra" box if you wish to view/update the Xtra fields for the underlying mailbox or Audiotex record.

# CALLBasic

## Overview

CALLBasic is a powerful scripting language that enhances the ability of CALLMaster to interact with the caller via Audiotex. It enables you to compute information, interface with and manage databases, execute third party programs and manage call control.

[Getting Started](#)

[Scripts Overview](#)

[Local and Global Variables](#)

[Commands Summary](#)

[Database Access](#)

## **Getting Started with CALLBasic**

CALLMaster offers a great deal of flexibility in dealing with your call management needs. Using CALLBasic can greatly enhanced the versatility and control of your call flow.

CALLBasic is the programming language proprietary to CALLMaster. Its many functions afford you the opportunity to have complete control over how CALLMaster interacts with your caller.

After placing CALLBasic commands into a text file referred to as a script, you call the script via the 'CALLBasic Script File' field in any Audiotex module.\_

## CALLBasic Scripts

A script is merely a text file with a *.bas* extension and consists of processing logic between **Program** and **EndProgram** commands.

A script must start with a Program command and end with the EndProgram command.

The processing logic or program code makes up the body of the script and is placed between the Program and EndProgram commands.

Program

    'Program code goes here.

EndProgram

The program code consists of Local and Global Variables, CALLBasic Commands, and Database Commands.

Code lines which start with a single apostrophe ( ' ) are comment lines and are not processed.

CALLBasic is very particular as to the syntax, capitalization, carriage return at the end of each line (no spaces), and must have a carriage return after the EndProgram line.

Script files can be created and edited using Notepad. They must be saved as *.bas* extension.

Sample scripts are included with CALLMaster and located in the CALLMaster\Scripts\Sample Scripts folder.

### Entry.bas and Exit.bas Scripts

These scripts are executed during CALLMaster system service start-up and shutdown, respectively. These files must named Entry.bas and Exit.bas and placed in the CALLMaster\Scripts folder.

#### Entry.bas

Entry.bas is executed when CALLMaster service starts, before any calls are processed. It is intended to be the Audiotex initialization script.

Use this script to declare global variables and assign starting values that can be used by all lines or execute database commands at service start-up. For example, open all the databases needed and declare initial values for global variables to be used as counters before any calls get handled.

Note: In the Entry.bas script, Local and Global Variables can only be assigned a value and cannot be reassigned. You will need to remove the variable before reassigning it to a new value .

For example: if you set Var1=10 and then reassign Var1=20 in the Entry.bas script, the value of Var1 will remain as 10.

You will need to assign Var1=10, then RemoveVar Var1, before reassigning Var1=20

#### Exit.bas

The Exit.bas file performs the same function as the Entry.bas file but does so at system service shutdown time, after all calls have been processed.

By placing printing and database commands in the Exit.bas file, an application can easily print closing summary information or close all open databases.



## **CALLBasic Local and Global Variables**

### **Overview**

[Notational Conventions](#)

[Commands Syntax](#)

[Variable Types](#)

[System Variables](#)

## CALLBasic Notational Conventions

The CALLBasic command reference uses the following notational conventions:

Convention	Description
<i>variable</i>	Characters in italics represent variables. A variable is a name you assign for storing numbers or characters in memory. <b>Do not begin variable names with a numeral.</b>
	A vertical bar represents the word 'or.' You may enter either what precedes or follows the vertical bar, but not both.
[ ]	Items enclosed in square brackets are optional.
Example:	<i>NumberOfRecords</i> = '1254'
command	All commands are printed in bold face letters.
Example:	Print <i>NumberOfRecords</i>
constants	Constants are a string of characters representing word constants, names, or positive or negative numbers. Constants must be enclosed in quotation marks ('constant'). Embedded blank spaces are not allowed.
Example:	<i>FirstName</i> = 'William'

## CALLBasic Command Syntax

Syntax means the spelling, capitalization, spacing, and order of CALLBasic commands and any required parameters. The most seemingly popular mistakes are *Case and parameter* matching.

Some variables (and constants) are information you must supply. Other variables are placeholders for information that the system will return. Multiple commands can be placed in one script file, but only one command may be placed on a line.

**CALLBasic is very particular as to the syntax & capitalization required.** Make sure there is a carriage return at the end of each line (no spaces), you must have a carriage return after the EndProgram line, commands must be spelled correctly, have the right letters capitalized with the required parameters. Use sample scripts that are located in the CALLMaster Scripts folder.

## CALLBasic Variables

All variables display data in a standard ASCII format even for number values.

### Local, Global & Disk Variables

Assign a value to a local, global, or disk variable. The command creates the variable if it doesn't already exist. Variable names can be any length and can contain any valid printable ASCII character with no embedded spaces.

- **Local** variables are cleared at the beginning of each call and whose values are private to each port.
- **Global** variables are prefixed with the '@' character, are cleared at service shut down, are accessible by all lines and whose value is shared by all lines.
- **Disk** variables - *Not currently implemented*. Have an '@' character in the second position of their name (e.g. @@answer2) and are saved to the hard disk file 'disk.var'. These variables are restored to their values the next time CALLMaster starts up. For local disk variables, a convenient naming convention is to use the '\_' character as the first character, e.g., '\_@varname'. Variables saved in the 'disk.var' file are prefixed by their port number (e.g., 1 \_@varname = 3). Global variables are prefixed by port number zero (e.g., 0 @@global = 7). Disk.var is an ASCII file. Only numeric values can be assigned by this command. Setting a variable to a string is done with the string functions.

### Constants

Any numeric value in the range of -2,000,000,000 to +2,000,000,000. Constants should be enclosed in double quotation marks ("").

### System Variables

A system variable returns system information when used in an assignment or expression. System variables may not be used on the left side of an equal sign (=). Below is the list of globally defined system variables and the values they return.

### Comments

Comments can be added to a script for maintenance and debugging. Comment lines MUST START with a single quote (').

## CALLBasic System Variables

System Variable	Type	Description / Value Returned
<b>\$areacode</b>	String	The number being dialed on that port during an Out Dial application., from the 'Area Code' field in the CALLMaster Manager Out Dial Screen or OutcallSchedule Table.
<b>\$calls</b>	Integer	The total number of calls in the system since last start-up. This should match the Call Count from the CALLMaster - Main Window.
<b>\$channel</b>	Integer	The current channel or line number of the call.
<b>\$col</b>	Integer	The total number of calls on this port since last start-up.
<b>\$contype</b>	Integer	The type of connection type event made. i.e. 3=Voice, 4=answering machine.
<b>\$cr</b>	Integer	Call Results of an Out Diall attempt. i.e. 10=connect, 18=connect fax machine.
<b>\$CSN</b>	String	Used by SMDI to return the caller's phone number. Returns the caller ID input from the external unit.
<b>\$day</b>	Integer	The day of the month, a value between 1 and 31.
<b>\$DBError</b>	Integer	Database action result. Set to 1 if occurrence of error after a database action (e.g. open, read). Must be initialized to 0 before issuance of each database command.
<b>\$dialprefix</b>	String	The dialing prefix of the number being dialed on that port during an Out Dial application., from the 'Dialing Prefix' field in the OutcallSchedule Table.
<b>\$dow</b>	Integer	The day of the week, a value between 1 and 7. (1=Sunday; 7=Saturday)
<b>\$doy</b>	Integer	The day of the year, a value between 0 and 365.
<b>\$DQuote</b>	String	Represents Double Quote (') character.
<b>\$error</b>	Integer	The last error encountered on that port, or a zero.

<b>\$faxsent</b>	String	The fax transmission result - 'OK' (successful) or 'NOK' (not successful).
<b>\$faxresult</b>	String	The fax transmission result. A numeric code returned by the Dialogic board.
<b>\$FF</b>	String	Form Feed inserts a new page
<b>\$FR</b>	String	Used in SMDI to return the forwarding reason.
<b>\$FSN</b>	String	Used in SMDI to return the forwarding station number, & in caller ID to return the caller's name (if provided).
<b>\$hour</b>	Integer	The current hour, a value between 0 and 23.
<b>\$lcddisplay</b>	String	The LCD display string used with the Digital Integration Boards (D/42JCT-U, D/82JCT-U).
<b>\$minute</b>	Integer	The current minute, a value between 0 and 59.
<b>\$modulenum</b>	Integer	The value of current Audiotex module key.
<b>\$month</b>	Integer	The year, a value between 1 and 12. (1=January; 12=December)
<b>\$NL</b>		Use this command to issue a carriage return.
<b>\$odAMmsgIDs</b>	String	The value in the AMMsgIDs field in the OutcallSchedule table.
<b>\$odbusyretrycount</b>	Integer	The value in the BusyRetryCount field in the OutcallSchedule table.
<b>\$odbusyretrymax</b>	Integer	The value of the BusyRetryMax field in the OutcallSchedule table.
<b>\$odbusyretrytime</b>	Integer	The value of the BusyRetryTime field in the OutcallSchedule table.
<b>\$oddata</b>	String	The value of the Data field in the OutcallSchedule table.
<b>\$oddayscale</b>	String	The value of the DaysToCall field in the OutcallSchedule

table.

<b>\$oddisconnectreason</b>	String	The value of the DisconnectReason field in the OutcallSchedule table.
<b>\$odearliesthour</b>	DateTime	The value of the EarliestHour field in the OutcallSchedule table.
<b>\$odexpires</b>	DateTime	The value of the Expires field in the OutcallSchedule table.
<b>\$odfaxIDs</b>	String	The value of the FaxIDs field in the OutcallSchedule table.
<b>\$odflags</b>	String	The value of the Flags field in the OutcallSchedule table.
<b>\$odlastcalltime</b>	DateTime	The value of the LastCallTime field in the OutcallSchedule table.
<b>\$odlatesthour</b>	DateTime	The value of the LatestHour field in the OutcallSchedule table.
<b>\$odmailbox</b>	String	The value of the Mailbox field in the OutcallSchedule table.
<b>\$odmemo</b>	String	The value of the Memo field in the OutcallSchedule table.
<b>\$odmsgIDs</b>	String	The value of the MsgIDs field in the OutcallSchedule table.
<b>\$odname</b>	String	The value of the Name field in the OutcallSchedule table.
<b>\$odnoanswerretrycount</b>	Integer	The value of the NoAnswerRetryCount field in the OutcallSchedule table.
<b>\$odnoanswerretrymax</b>	Integer	The value of the NoAnswerRetryMax field in the OutcallSchedule table.
<b>\$odnoanswerretrytime</b>	Integer	The value of the NoAnswerRetryTime field in the OutcallSchedule table.
<b>\$odnoconfirmretrycount</b>	Integer	The value of the NoConfirmRetryCount field in the OutcallSchedule table.
<b>\$odnoconfirmretrymax</b>	Integer	The value of the NoConfirmRetryMax field in the OutcallSchedule table.

<b>\$odnoconfirmretrytime</b>	Integer	The value of the NoConfirmRetryTime field in the OutcallSchedule table.
<b>\$odnotes</b>	String	The value of the Notes field in the OutcallSchedule table.
<b>\$odreceivetones</b>	String	The value of the ReceiveTones field in the OutcallSchedule table.
<b>\$odrecordedmsgIDs</b>	String	The value of the RecordedMsgIDs field in the OutcallSchedule table.
<b>\$odrings</b>	Integer	The value of the Rings field in the OutcallSchedule table.
<b>\$odsecs</b>	Integer	The value of the Secs field in the OutcallSchedule table.
<b>\$odsendtones</b>	String	The value of the SendTones field in the OutcallSchedule table.
<b>\$odstartdatetime</b>	Integer	The value of the StartDateTime field in the OutcallSchedule table.
<b>\$odstartmodule</b>	Integer	The value of the AudiotexModule field in the OutcallSchedule table.
<b>\$odstatus</b>	String	The value of the Status field in the OutcallSchedule table.
<b>\$oduserdatetime</b>	Integer	The value of the UserDateTime field in the OutcallSchedule table.
<b>\$outdialID</b>	Integer	The value of the ID field in the OutcallSchedule table.
<b>\$phonenum</b>	String	The phone number being dialed on that port during an Out Dial application., from the 'Phone No' field in the CALLMaster Manager Out Dialing Screen or OutcallSchedule Table.
<b>\$rand</b>	Integer	A random number between 0 and 32,767.
<b>\$sec</b>	Integer	The current seconds, a value between 0 and 59.
<b>\$today</b>	DateTime	Format: MM/DD/YY HH:MM:SS military time; For use in command 1) SayDate for month and day (e.g. January twenty fifth), and



2) SayTime for hour, minute and am/pm (e.g. Seven forty-five pm).

<b>\$tt</b>	String	The caller's touch-tone response to the current module.
-------------	--------	---

<b>\$year</b>	Integer	Format: YYYY; Returns the current 4 digit year.
---------------	---------	---

**CALLBasic Error Handling (use in CALLBasic script to point to Audiotex module for error handling)**

<b>exit</b>	Integer	Format: exit = xxxx, where xxxx is the module number to handle # key.
-------------	---------	---

<b>lctrp</b>	Integer	Format: lctrp = xxxx, where xxxx is the module number to handle a hang-up.
--------------	---------	--

**Caller ID (Services supported – CLASS, ACLIP, CLIP , JCLIP Japanese CLIP)**

Dialogic Board must support Caller-id. If the call is answered before Caller ID information has been received, Caller ID information will not be available. 2 rings required in the CLASS and ACLIP services (USA); 1 ring required for CLIP and JCLIP (Europe& Japan).

<b>\$absence1</b>	String	Reason for absence of Caller ID (only available if caller name is absent): O = out of area, P = private
-------------------	--------	---

<b>\$absence2</b>	String	Reason for absence of name (only available if caller name is absent): O = out of area, P = private
-------------------	--------	--

<b>\$callerid</b>	String	Caller Id string sent by the phone company. Could be upto 256 characters and contain the phone number. CALLMaster must receive atleast 2 rings to get the Caller-id.
-------------------	--------	--

<b>\$calltype</b>	String	1= voice call, 2 = ring back when free call, 129 = message waiting call - (CLIP only)
-------------------	--------	---

<b>\$dn</b>	String	Directory Number (caller's phone number)
-------------	--------	--

<b>\$ddn</b>	String	Dialed Directory Number (callee's phone number) or DNIS (Dialed Number Identification Service)
<b>\$name</b>	String	Caller's subscriber name. Contains name of caller as returned in the caller-id.
<b>\$netmsg</b>	String	Network Message System status: number of messages waiting - (CLIP only)
<b>\$qualifier</b>	String	L = long distance call
<b>\$redirect</b>	String	Call forward: 0 = universal; 1 = busy; 2 = unanswered
<b>ASR System Variables</b>		
<b>\$ASR</b>	Integer	Flag to check if ASR enabled. Return 1 if ASR enabled or 0 if not enabled.
<b>\$ASRConfidence</b>	Floating Point	The confidence score from the Speech Recognition engine.
<b>\$ASRREQConfidence</b>	Integer	Required confidence if set in the TexttoRecognize file. -1=low, 0=med, 1=high.
<b>\$ASRSAPICConfidence</b>	Integer	The confidence score from the SAPI engine. -1=low, 0=med, 1=high.

## CALLBasic Commands

Command	Description
<a href="#"><u>&amp;</u></a>	Concatenate
<a href="#"><u>AddDate</u></a>	Add days, hours, minutes, second increments to date/time variable creating new date/time variable.
<a href="#"><u>AddMessage</u></a>	Add a recorded message to a mailbox. This will initiate notification if enabled.
<a href="#"><u>AppendFile</u></a>	Appends a copy of the source file to the end of the destination file with the option to rewind some bytes before appending
<a href="#"><u>Ask</u></a>	Play file and record response
<a href="#"><u>Branching</u></a>	Branch based on a conditional execution commands.
<a href="#"><u>Bridge</u></a>	Bridge connects 2 ports to talk to each other or share resource (fax)
<a href="#"><u>Call</u></a>	Dial a number (with call analysis)
<a href="#"><u>ClearDTMF</u></a>	Clear touch-tone inputs for a given port
<a href="#"><u>CopyFile</u></a>	Copy a file
<a href="#"><u>Dial</u></a>	Dial a number
<a href="#"><u>Email</u></a>	Send email with or without attachment, with custom text directly to email address(es)
<a href="#"><u>Enslave</u></a>	One channel can issue dialogic commands to another channel that it enslaves.
<a href="#"><u>Exec</u></a>	Execute a third party program from within a script file
<a href="#"><u>ExecBas</u></a>	Run another .bas script from within the current .bas script.
<a href="#"><u>Exit</u></a>	Exit a While...WEnd or For...Next loop
<a href="#"><u>ExpandAbrv</u></a>	Expands abbreviated fields from a string variable. Requires a connection string to access the database that has the CMAbbreviations table.

<b><u>ExtractFields</u></b>	Extracts fields from a string variable that are separated by a delimiter ' '. Returns up to 5 variables with the string values of each field.
<b><u>FaxReceive</u></b>	Receive fax document as a file or to a mailbox
<b><u>FaxSend</u></b>	Schedule a fax to be sent
<b><u>FileDelete</u></b>	Delete a file
<b><u>FileIDToName</u></b>	Convert a File Id number to a qualified path name
<b><u>FileRead</u></b>	Read text from an ASCII file
<b><u>FileRename</u></b>	Rename a filename using full path names
<b><u>FileStatus</u></b>	Check existence of a file
<b><u>FileWrite</u></b>	Creates a new ASCII file or appends to an existing ASCII file
<b><u>FindDups</u></b>	Find duplicate fields in a string variable with a delimiter ' '.
<b><u>FindString</u></b>	Finds starting position of token string in source variable starting from position 0
<b><u>FloatToString</u></b>	Returns the string value of an floating point
<b><u>For...To...Step...Next</u></b>	Repeat a block of statements until the Step bound is reached
<b><u>GetDate</u></b>	Get variable in date data type format
<b><u>GetDayofWeek</u></b>	Returns integer variable for day of week based on date variable provided
<b><u>GetDTMF</u></b>	Gets caller's Touch-Tone input
<b><u>GetField</u></b>	Gets string variable in the field number specified of source variable starting from field 1
<b><u>GetFloat</u></b>	Converts a string to a floating point number
<b><u>GetInteger</u></b>	Converts a string to an integer

<b><u>If...Then...Else...EndIf</u></b>	Branching statement
<b><u>IntegerToString</u></b>	Returns the string value of an integer
<b><u>MakeDate</u></b>	Make date variable in date data type format
<b><u>Max</u></b>	Return the larger of two integer values
<b><u>MemMove</u></b>	Copy characters from one string variable into another
<b><u>Min</u></b>	Return the smaller of two integer values
<b><u>Play</u></b>	Play a digital audio file referenced by fileID
<b><u>PlayFileName</u></b>	Play a digital audio file referenced by a valid file path
<b><u>Print</u></b>	Print the value designated to the CMCalls <DATE>.TXT file
<b><u>PrintNL</u></b>	Same as 'Print' but this adds a newline character to the end of the print statement
<b><u>PutEvent</u></b>	Places an event onto the current event queue
<b><u>Recognize</u></b>	Recognize a spoken word or phrase from a list/file and returns the word position
<b><u>Record</u></b>	Record a digital audio file
<b><u>RecordAppend</u></b>	Record at the end of a previously recorded file
<b><u>RemoveVar</u></b>	Remove a variable from memory
<b><u>RemoveVars</u></b>	Clear all the local memory variables for this port
<b><u>Return</u></b>	Use a variable to determine which mailbox or Attendant ID to access, or which module to go to next
<b><u>SayNumber</u></b>	Say the contents of a variable in number format
<b><u>SayNumeral</u></b>	Say the contents of a variable in numeral format

<b><u>SayOrdinal</u></b>	Say the contents of a variable in ordinal format
<b><u>SayTime</u></b>	Say a time
<b><u>SayDate</u></b>	Say a date
<b><u>SendUDP</u></b>	Sends a datagram packet with the UDP protocol
<b><u>SetHook</u></b>	Enables a line to go on hook or off hook
<b><u>Sleep</u></b>	Sets a delay in the processing
<b><u>Speak</u></b>	Using SAPI Text-to-Speech, say the written word
<b><u>StringConcat</u></b>	Concatenate one variable string to the end of another
<b><u>StringCopy</u></b>	Copy one variable string to another
<b><u>StringLength</u></b>	Return the string length
<b><u>StringNConcat</u></b>	Concatenate up to n characters from the beginning of one variable to the end of another variable
<b><u>Transcribe</u></b>	Transcribes the spoken word or phrase into a string variable or ASCII file.
<b><u>While...WEnd</u></b>	Repeat a block of statements while a condition is true

trk

## CALLBasic Command - &

**Command:** &

**Description:** Concatenate/Make one string from two separate strings by copying one to the end of the other.

**Format:** variable = *variable1* & *variable2*

Parameter	Type	Description
<b>Input:</b>		
<i>variable1</i>	String	<i>variable</i> , "constant", or system variable
<i>variable2</i>	String	<i>variable</i> , "constant", or system variable
<b>Output:</b>		
None		

### Example 1:

```
Program
  var1 = "1234" & "5678"
  Print var1
  'This should print 12345678
EndProgram
```

### Example 2:

```
Program
  var3 = "abc"
  var4 = "def"
  var5 = "ghi"
  var1 = var3 & var4
  var2 = var3 & var4 & var5
  Print var1
  'This should print abcdef
  Print var2
  'This should print abcdefghi
EndProgram
```

## CALLBasic Command - AddDate

**Command:**            **AddDate**

**Description:**        Add days, hours, minutes, seconds increments to date/time variable creating new date/time variable. Use negative values to subtract from required date/time variable.

**Format:**            **AddDate** *NewDateTime Day-Inc HH-Inc Min-Inc Sec-Inc*

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>NewDateTime</i>	DateTime	starting date and time. New date time will be returned in this parameter.
--------------------	----------	---

<i>Day-Inc</i>	Integer	variable containing the incremental days to add to the starting date and time.
----------------	---------	--

<i>HH-Inc</i>	Integer	variable containing the incremental hours to add to the starting date and time.
---------------	---------	---

<i>Min-Inc</i>	Integer	variable containing the incremental minutes to add to the starting date and time.
----------------	---------	---

<i>Sec-Inc</i>	Integer	variable containing the incremental seconds to add to the starting date and time.
----------------	---------	---

### Output:

None

### Example:

```
Program
  StartDateTime = $today
  Print "Current time is "
```



PrintNL StartDateTime

'Initialize values to increase minutes by 5 from current date and time

Day-Inc = 0

HH-Inc = 0

Min-Inc = 5

Sec-Inc = 0

NewDateTime = StartDateTime

**AddDate** NewDateTime Day-Inc HH-Inc Min-Inc Sec-Inc

Print "Increased current time by 5 minutes to "

PrintNL NewDateTime

'Initialize values to decrease days by 10 from current date and time

Day-Decr = 0 - 10

HH-Inc = 0

Min-Inc = 0

Sec-Inc = 0

NewDateTime = StartDateTime

**AddDate** NewDateTime Day-Decr HH-Inc Min-Inc Sec-Inc

Print "Decreased current date by 10 days to "

PrintNL NewDateTime

EndProgram

## CALLBasic Command - AddMessage

**Command:**        **AddMessage**

**Description:**        Add a recorded message file to a specified mailbox. This will send out email notification(s) if the mailbox is set-up to do so.

**Format:**        **AddMessage** *ToMB MsgID FromMB*

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>ToMB</i>	String	mailbox number that the message is being added to.
<i>MsgID</i>	Integer	CALLMaster speech file id.
<i>FromMB</i>	String	mailbox number that the message is being sent from.

### Output:

None

### Example:

```
Program
  ToMB = "1102"
  MsgID = 100011
  FromMB = "0000"
  AddMessage ToMB MsgID FromMB
EndProgram
```

## CALLBasic Command - AppendFile

**Command:**           **AppendFile**

**Description:**       Appends a copy of the source file to the end of the destination file with the option to rewind some bytes before appending.  
This function can be used to concatenate and append wav files.

**Format:**           **AppendFile** *source dest rewindbytes*

Parameter	Type	Description
-----------	------	-------------

---

### Input:

<i>source</i>	String	full path name of source file to append
---------------	--------	---

<i>dest</i>	String	full path name of destination file to append to
-------------	--------	---

<i>rewindbytes</i>	Integer	the bytes to rewind on the destination file before appending source file
--------------------	---------	--

### Output:

None

### Example:

Program

```
source = "C:\\Program Files\\CALLMaster\\PMT\\10\\660tone.wav"  
dest = "C:\\Program Files\\CALLMaster\\PMT\\10\\2.wav "  
rewindbytes = 3000
```

**AppendFile** *source dest rewindbytes*

EndProgram

CALLBASIC COMMAND - Ask

Command: Ask

Description: Play file and record response. Places the recorded file PlayFileName and places the record response in the variables RecordFileName and RecordFileID. This command is used in conjunction with the Recognize command for ASR. See sample script for ASK. If the file used in PlayFileName is recorded in 8Khz 8Bit Mono Linear PCM wave file format, the caller will be able to Barge-In and respond (i.e. the caller will not have to wait for the end of the message to respond.)

Format: Ask PlayFileName AudioFileIn RecordFileID

Parameter	Type	Description
Input:		
PlayFileN ame	String	name of the file with full path that has the audio to play first
RecordFilen ame	String	name of the file with full path of the Audio Recording
RecordFile/ D	Integer	the file ID of the RecordFilename above
Output:		
None		

Example:  
Program  
RecordFileID = 0  
AudioFileIn = ""  
PlayFileName = "C:\Program Files\CALLMaster\Grf\33\50.wav"

Ask PlayFileName AudioFileIn RecordFileID

EndProgram

## CALLBasic Command - Branching

### Branching and Looping

Branching refers to conditionally executing a group of statements based on the value of an expression. This is done in CALLBasic using one of the commands below.

- **If...Then...Else...EndIf** - Repeats a set of statements until a certain condition expires.
- **For...Next** - Repeats a block of statements until the upper bound of the Step clause is reached.
- **While ... Wend** - Repeats a block of statements while a condition is True.

## CALLBasic Command - Bridge

**Command:**        **Bridge**

**Description:**        Bridge connects 2 Dialogic hardware devices. This is normally used to bridge 2 telephone line devices (analog/LSI, digital/DTI) for a conversation, line monitoring and/or conferencing. It can also be used to connect a fax device that is shared by other analog or digital voice devices. Device types include LSI=analog, VOX=voice, DTI=digital T1, FAX=fax, MSI=PBX

Prompt 560 will be played while bridge is attempted. See also Prompts 561 and 562.

**Format:**        **Bridge** *port1 port1Type port2 port2Type duplex*

Parameter	Type	Description
-----------	------	-------------

---

### Input:

<i>port1</i>	Integer	Port Number of first of the 2 devices to be connected.
<i>port1Type</i>	String	Device type of the first of the 2 devices to be connected.
<i>port2</i>	Integer	Port Number of the second of the 2 devices to be connected.
<i>port2Type</i>	String	Device type of the first of the 2 devices to be connected.
<i>duplex</i>	String	Two way or one way connect or disconnect. Values include FULLDUPLEX HALFDUPLEX FULLDISCONNECT HALFDISCONNECT

### Output:

None

### Example 1:

#### Program

Print "Bridge Test Script **connecting** analog devices of port 2 to port 4 for a conference call "

port1 = 2

port1Type = "LSI"

port2 = 4

port2Type = "LSI"

duplex = "FULLDUPLEX"

**Bridge** port1 port1Type port2 port2Type duplex

Print "Bridge Test Script **disconnecting** analog devices of port 2 to port 4 "

port1 = 2

port1Type = "LSI"

port2 = 4

port2Type = "LSI"

duplex = "FULLDISCONNECT"

**Bridge** port1 port1Type port2 port2Type duplex

EndProgram

#### Example 2:

##### Program

Print "Bridge Test Script **connecting** digital T1 device of port 2 to fax device of port 26 for sharing fax functionality "

port1 = 2

port1Type = "LSI"

port2 = 26

port2Type = "FAX"

duplex = "FULLDUPLEX"

**Bridge** port1 port1Type port2 port2Type duplex

Print "Bridge Test Script **disconnecting** digital T1 device of port 2 to fax device of port 26 "

port1 = 2

port1Type = "LSI"

port2 = 26

port2Type = "FAX"

duplex = "FULLDISCONNECT"

**Bridge** port1 port1Type port2 port2Type duplex

EndProgram



## CALLBasic Command - Call

**Command:**        **Call**

**Description:**        Make a call using Dialogic call analysis. The command returns the Dialogic Call Result (or CAR), Connection Type Event and Call Result. Subsequent modules typically test the event data to determine the call result.

**Format:**        **Call** *number car dlg\_event dlg\_result*

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>number</i>	String	The string to dial.
---------------	--------	---------------------

### Output:

<i>car</i>	Integer	Returns the Dialogic call analysis result block (see Appendix B).
------------	---------	---

<i>dlg_event</i>	Integer	Returns the Dialogic Connection Type Event (see Appendix B).
------------------	---------	--

<i>dlg_result</i>	Integer	Returns the Dialogic Call Result where 7 is busy, 8 is no answer, 9 is no ring back, 10 is connect, 11 is intercept and 18 is fax connect (see Appendix B).
-------------------	---------	---

### Example:

Program

**Call** phonenum car event result

    'CALLMaster will dial the phone number in the variable phonenum.

    'Upon determining the results of the call it will place the Dialogic call analysis result block in the variable car,

    'the Connection Type Event in the variable event, and the Call Result in the variable result.

EndProgram

### Example:

Program

**Call** "&,567" c e r

'CALLMaster will flash-hook (&); pause (,); and dial "567".

'The CAR will be returned to c and can be read using **GetInteger**, e should equal "18" (call analysis complete) and,

'if the transfer is successful, r will equal "10."

EndProgram

**CALLBasic Command - ClearDTMF**

**Command:** ClearDTMF

**Description:** Clear any remaining touch-tone characters received after the timeout.

**Format:** ClearDTMF

Parameter	Type	Description
-----------	------	-------------

**Input:**

None

**Output:**

None

**Example:**

```
Program
DIGITS = ""
TMASK = "#"
NDIGITS = 7
TIMEOUT = 5
GetDTMF DIGITS TMASK NDIGITS TIMEOUT
PrintNL DIGITS
ClearDTMF
msg = "After clearing digits "
Print msg
Print DIGIT
EndProgram
```

## CALLBasic Command - CopyFile

**Command:** **CopyFile**

**Description:** Copies a file to a target file with the option to overwrite if the target file exists.  
NOTE: File names with paths require double backslashes.  
The target folder used in the full file name will be created if it does not exist.  
e.g. if target file c:\CALLMaster\targetfiles\0.wav then target folder c:\CALLMaster\targetfiles will be created if it does not exist.

**Format:** **CopyFile** *source target overwrite*

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>source</i>	String	Name of source file to be copied. File names with paths require double backslashes.
<i>target</i>	String	Name of copied file. File names with paths require double backslashes.
<i>overwrite</i>	String	Set to "YES" to copy file even if target already exists otherwise set to "NO" to skip the copy.

### Output:

None

### Example:

Program

```
source = "C:\\Program Files\\CALLMaster\\PMT\\0\\0.wav"
```

```
target = "C:\\00.wav"
```

```
overwrite = "YES"
```

```
CopyFile source target overwrite
```

```
'This example will make a copy of the source file "C:\\Program Files\\CALLMaster\\PMT\\0\\0.wav" and  
'save as target file "C:\\00.wav".
```

```
EndProgram
```

**CALLBasic Command - Dial**

**Command:** **Dial**

**Description:** Dial the specified Hayes compatible strings supporting flash, pause, and tones. Similar to the "Call" command, but does not perform call analysis. The **Dial** command should return a 0 (no result) because it does not perform call analysis.

**Format:** **Dial** *number dlg\_event dlg\_result*

Parameter	Type	Description
<b>Input:</b>		
<i>number</i>		Touch-tone string to be dialed.
<i>dlg_event</i>		Returns the Dialogic "Connection Type Event" (see Appendix B).
<i>dlg_result</i>		Returns the Dialogic "Call Result" (see Appendix B).
<b>Output:</b>		
None		

**Example:**  
Program  
  phoneNum = "5551212"  
  **Dial** phonenum e r  
  'The command dials the string in phonenum, then returns the Dialogic Connection Type Event in the e variable  
  'and the Dialogic Call Result in the r variable. See Appendix B for details.  
EndProgram

## CALLBasic Command - Email

**Command:**                      **Email**

**Description:**                      Use this command to send an email.  
The email will be sent as defined by the Mode set in the System Parameter #30 - 'MAPI Profile'. See [Activating Email](#) for details.

**Format:**                              **Email** *MapiProfile FromAddress ToAddress CCAddress BCCAddress Subject  
BodyText FileAttach*

Parameter	Type	Description
<b>Input:</b>		
<i>MapiProfile</i>	String	email account name will be used if matching name exists
<i>FromAddress</i>	String	email address of sender
<i>ToAddress</i>	String	email address of recipient. Can contain multiple addresses separated by a semi-colon.
<i>CCAddress</i>	String	email address of recipient to receive copy. Can contain multiple addresses separated by a semi-colon.
<i>BCCAddress</i>	String	email address of recipient to receive blind copy. Can contain multiple addresses separated by a semi-colon.
<i>Subject</i>	String	email subject
<i>BodyText</i>	String	content of email
<i>FileAttach</i>	String	full path of file to be used as email attachment

**Output:**

None

NOTE: space required for any blank parameters

**Example:**

Program

MapiProfile = "CallMaster"

From = "callmaster@speechsoft.com"

To = "sales@speechsoft.com;service@speechsoft.com"

CC = "support@speechsoft.com "

BCC = " "

Subject = "CALLMaster Survey"

BodyText = "The attached message is from CALLMaster."

FileAttach = "C:\Program Files\CallMASTER\Doc\Readme.txt"

**Email** MapiProfile From To CC BCC Subject BodyText FileAttach  
EndProgram

## CALLBasic Command - Enslave

**Command:** **Enslave**

**Description:** Use this command to control dialogic functions of another line. One channel can control board functions of the channel that it enslaves. Once a line is enslaved, these dialogic commands (Sethook, Dial/Call, FAXSend/Receive, ClearDTMF, Play, PlayFileName, Speak, SayNumeral, SayOrdinal, SayNumber, SayTime, SayDate, GetDTMF, Record) will control the enslaved line.

**Format:** **Enslave** *slaveport slavehandle*

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>slaveport</i>	String	the port # to be enslaved; number can be "0" to try any free port, a specific port# or a range e.g. "2-9"
<i>slavehandle</i>	Integer	should be set to zero to enslave port. A non-zero handle will release enslaved port.

### Output:

<i>slavehandle</i>	CALLMaster returns a valid non zero handle to be used for future telephony calls
--------------------	--

### Example:

Program

```
PrintNL "SlaveTest test Script"
```

```
' slaveport returns with the enslaved port. Port 0 means any free port, a range
```

```
slaveport = "3"
```

```
'The name "slavehandle" is key to controlling a slave channel.
```

```
'If that variable exists then the value it contains is used to control the slave channel.
```

```
'The variable "slavehandle" must be zero to request to enslave a port.
```

```
'CALLMaster returns a valid non zero handle to be used for future telephony calls or to test if the slave  
command was successful
```

```
slavehandle = 0
```

```
Enslave slaveport slavehandle
```



' Enslave called with valid port# and matching Handle from a prior Enslave call will Release enslaved port  
**Enslave** slaveport slavehandle  
EndProgram

## CALLBasic Command - Exec

**Command:** **Exec**

**Description:** Use this command to execute a third party program from within the current .bas script.

**Format:** **Exec** *prog mode*

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>prog</i>	String	name of program to be executed and its command line parameters. Can only be .exe files. If the executable has parameters, they will need to be preset to string variables ENV<x> where x is the variable sequence stands for a number from 0 to 9 for a maximum of 10 variables. All variables must be string and must be defined in order. The executable file must use the GetEnvironment fuction to retrieve the values and SetEnvironment function to change a value. The returned values are put back in to the original variables after the program terminates.
<i>mode</i>	String	variable set to "WAIT" for CALLMaster to wait before continuing with the next CALLBasic command

### Output:

None

### Example 1:

```
Program
  Prog = "vox2wav.exe -w e:\cm\1.vox e:\cm\1.wav -adpcm 6000 -8"
  Mode = "WAIT"
  Exec Prog Mode
EndProgram
```

### Example 2:

Program

ENV0 = "Hello there"

ENV1 = "128"

Prog = "<YourApplication.exe>"

Mode = "WAIT"

**Exec** Prog Mode

EndProgram

## CALLBasic Command - ExecBas

**Command:** ExecBas

**Description:** Use this command to run another .bas script from within the current .bas script. When the second script completes, control will return back to the current script. Note: the called script cannot be used as a subroutine. Variables cannot be exchanged between the calling and called scripts.

**Format:** ExecBas *filepath\filename*

Parameter	Type	Description
-----------	------	-------------

---

### Input:

<i>filepath\filename</i>	String	The full path, filename and extension of the CALLBasic script to be executed.
--------------------------	--------	---

### Output:

None

### Example:

```
Program
```

```
    ExecBas "c:\Program Files\CallMaster\Scripts\GetPinNumber.bas"
```

```
EndProgram
```

## CALLBasic Command - Exit

**Command:** Exit

**Description:** Use this command to exit a While...Wend loop or a For...Next loop.

**Format:** Exit

Parameter	Type	Description
-----------	------	-------------

---

**Input:**

None

**Output:**

None

### Example:

```
Program
  For n = 1 To 10 Step 1
    If n > 5 Then
      Exit
    Else
      n = n
    EndIf
  Next
  'In this example, the For loop exits before n reaches 10.
EndProgram
```

## CALLBasic Command - FaxReceive

**Command:** **FaxReceive**

**Description:** Receive fax document as a file or to a mailbox.

**Format:** **FaxReceive** *faxfile faxtype faxid faxpages*

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>faxfile</i>	String	Name of file to receive in fax document; or file name returned of actual saved document.
<i>faxtype</i>	String	Contains "FILE" if fax received as file; or mailbox number if fax received to mailbox.

### Output:

<i>faxtype</i>	String	Returns the status of the fax "-1"=Receiving Site Failure, "1" or "3"=OK, "2" or "5"=Sending Site Failure, "4"=Operator Intervention Requested
<i>faxid</i>	Integer	Returned faxid used to identify this fax in the FaxReceived table in the CALLMaster database
<i>faxpages</i>	Integer	Number of fax pages received

### Example:

Program

'Example of FaxReceive to File

faxfile = "H:\Program Files\CALLMaster\fax\received\myfax3.tif"

faxtype = "FILE"

faxid = 0

faxpages = 0

**FaxReceive** faxfile faxtype faxid faxpages

```
'Example of FaxReceive to mailbox # 1102
faxfile = ""
faxtype = "1102"
faxid = 0
faxpages = 0
FaxReceive faxfile faxtype faxid faxpages
EndProgram
```

## CALLBasic Command - FaxSend

**Command:** **FaxSend**

**Description:** Schedule a document to be faxed to the number called. For this command to work, you must have the necessary CALLMaster license with fax support, plus an optional fax board properly installed, configured, and tested. CALLMaster returns the fax send status in system variable \$faxsent.

**Format:** **FaxSend** *docID phonenum car*

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>docID</i>	Integer	The key of a Record in the fax database that describes which document to send.
--------------	---------	--

<i>phonenum</i>	String	The phone number is used only for reference purposes and not to make a call. This field can be null if faxing to the same number that the call is connected to as part of the Out Dial.
-----------------	--------	--

### Output:

<i>car</i>		Returns the Dialogic call result (see Appendix B).
------------	--	--

Program

**FaxSend** doc callernumber car

EndProgram

### Example:

```
Program
  phonenum = "919145551212"
  docID = 1
  Print "sending Fax 1"
```



**FaxSend** docID phonenum CAR

Print "CAR - "

Print CAR

'This will fax the document from the Fax table whose Fax ID is referenced in the variable docID.

' The phonenum variable in this command is for reference purposes only and not to make the call. The document will be faxed to the phone number called.

'Upon determining the results of the call it will place the Dialogic call result in the variable CAR.

EndProgram

For details on faxing see [CALLMaster Fax Support](#)

## CALLBasic Command - FileDelete

**Command:** **FileDelete**

**Description:** Delete a file by specifying its name with the folder path.

**Format:** **FileDelete** *name*

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>name</i>	String	The file name with the folder path.
-------------	--------	-------------------------------------

### Output:

None

### Example 1 :

```
Program
  FileDelete c:\temp\tempfile
  'Note: name string cannot contain spaces if used directly
  'This will delete a file named tempfile in the c:\temp folder.
EndProgram
```

### Example 2:

```
Program
  filename = "c:\Program Files\CALLMaster\Vox\833\49.wav"
  'Note: filename string can contain spaces if used as a variable
  FileDelete filename
  'This will delete a file named 49.wav in the c:\Program Files\CALLMaster\Vox\833 folder.
EndProgram
```

## CALLBasic Command - FileIDToName

**Command:** FileIDToName

**Description:** Convert a CALLMaster "file ID" value into a corresponding FAT file system name including full path. This is a convenient way to determine the name of a speech file just recorded, usually for the purpose of renaming it. CALLMaster will automatically assign "file IDs" when it makes a new recording and returns it in the "file ID" integer variable of the **Record** command. CALLMaster performs integer division by 120 and modulo 120 on the "file ID" value to generate the subdirectory and file name, respectively.

**Format:** FileIDToName *destname file\_id*

Parameter	Type	Description
-----------	------	-------------

---

### Input:

<i>destname</i>	String	Returns a FAT file name with folder path.
<i>file_id</i>	Integer	A speech "file ID" number (integer).

### Output:

None

### Example:

Program

**FileIDToName** nextadmsg recfileid

'This will place in the variable *nextadmsg* the FAT file system name of the speech file whose "file ID" is in the variable *recfileid*.

'If *recfileid* had a value of 102621 then it would be converted to "c:\Program Files\CALLMaster\VOX\855\21.wav" and placed in variable nextadmsg.

'A **FileRename** command would normally follow to move the file to a subdirectory of your choice for future retrieval.

' Setup full name for current recording number

' Make sure that folders used in selected name exist

' Create new file name with full path

```
RecFilePath = "C:\Program Files\CallMASTER\PMPT\"
RecFileName = "RenamedFile.wav"
RenamedFile = RecFilePath & RecFileName
' Renames file to user selected name
FileRename newname RenamedFile

'After renaming the file, you may want to then issue a FileDelete command to delete nextadmsg file.
FileDelete newname
EndProgram
```

## CALLBasic Command - FileRead

**Command:** **FileRead**

**Description:** Read text from an ascii file.

**Format:** **FileRead** *FileName ByteOffset ByteRequestCount BytesRead ByteBuffer*

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>FileName</i>	String	The name of the file to be read. The default directory is the CALLMaster directory or you may specify a path.
<i>ByteOffset</i>	Integer	Byte count offset from the beginning of the file to start reading.
<i>ByteRequestCount</i>	Integer	Number of bytes to be read from the file. Used only if variable ignoreNL > 0 is declared.
<i>BytesRead</i>	Integer	The actual number of bytes returned from the file.
<i>ByteBuffer</i>	String	A variable that will contain the bytes read from the file.

### Output:

None

If the variable named "**ignoreNL**" (integer) is defined and set > 0 , the command is to read past new line (CRLF) chars and the ByteRequestCount is used to determine how many bytes to read. Other wise the read will break on NL chars.

### Example:

Program

```
FileName = "readtest.txt"  
ByteOffset = 8
```

```
ByteRequestCount = 125
ignoreNL = 1
FileRead FileName ByteOffset ByteRequestCount BytesReadCount ByteBuffer
EndProgram
```

## CALLBasic Command - FileRename

**Command:**            **FileRename**

**Description:**        Rename a file. The command supports full path names.  
Note: This function will not replace the existing file (newfile) if one exists with the same path and name. Make sure that the file does not exist with the same name. Use the FileStatus command to check if file exists. If it does then use the FileDelete command before issuing the FileRename command.

**Format:**            **FileRename** *oldfile newfile*

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>oldfile</i>	String	The name of the file to be Renamed. The default directory is the CALLMaster directory you may specify a path.
<i>newfile</i>	String	The name that the file will be given, a name which cannot currently exist. The default directory is \CALLMaster.

### Output:

None

Example: **FileRename** temprec01 "greeting.vox"

This renames the file whose name is in variable *temprec01* to a file named "greeting.vox."

### Example:

```
Program
  ' this example renames the nextadmsg file created from the FileIDToName function to the newfile name
  oldfile = nextadmsg
  newfile = "c:\CALLMaster\RenamedFiles\File1.wav"
  FileRename oldfile newfile
EndProgram
```

## CALLBasic Command - FileStatus

**Command:** **FileStatus**

**Description:** Check the existence of a specified file.

**Format:** **FileStatus** *FileName Status Size Mdate*

Parameter	Type	Description
-----------	------	-------------

---

### Input:

<i>FileName</i>	String	Full path name of file to check
-----------------	--------	---------------------------------

### Output:

<i>Status</i>	Integer	Zero (0) returned if file does not exist
---------------	---------	--

<i>Size</i>	Integer	Size of file in bytes if file exists
-------------	---------	--------------------------------------

<i>Mdate</i>	Integer	Not in use
--------------	---------	------------

### Example:

Program

    FileName = "c:\Program Files\CALLMaster\Doc\Release Notes.txt"

    Status = 0

    Size = 0

    Mdate = 0

    FileStatus FileName Status Size Mdate

EndProgram



## CALLBasic Command - FileWrite

**Command:** FileWrite

**Description:** The **FileWrite** command is used to create a new ASCII file or append to an existing one. If Mode = "APPEND" and file does not exist then file will be created. If Mode = "CREATE" and file does exist then original file will be over written.

**Format:** FileWrite *FileName Mode Text*

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>FileName</i>	String	Name of ASCII file to be created or appended to
<i>Mode</i>	String	Must be set to either "CREATE" or "APPEND"
<i>Text</i>	String	Variable containing text to include in file.

### Output:

None

### Example 1:

Program

```
FileName = "writetest.txt"
```

```
' Create a new file
```

```
Mode = "CREATE"
```

```
Text = "Hello There"
```

```
Text = $NL & Text
```

```
FileWrite FileName Mode Text
```

```
'File "writetest.txt" will contain 2 lines as follows:
```

```
'
```

```
'Hello There
```

```
EndProgram
```

### Example 2:

Program

FileName = "writetest.txt"

' Write at the end of the file from Example 1

Mode = "APPEND"

Text = "This is a test message written by CALLMaster"

Text = \$NL & Text

**FileWrite** FileName Mode Text

'File "writetest.txt" will contain 4 lines as follows:

'

'Hello There

'

'This is a test message written by CALLMaster

EndProgram

## CALLBasic Command - FindString

**Command:** FindString

**Description:** The **FindString** command finds starting position of token string in source variable starting from position 0.

**Format:** FindString *source token pos*

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>source</i>	String	Variable to be searched
<i>token</i>	String	Variable containing the string to be searched

### Output:

<i>pos</i>	Integer	Starting position of the token in the source, starting at 0 If token not found then <i>pos</i> value returned is -1
------------	---------	--

### Example:

```
Program
' Find starting position of token string "SSM".
Source = "NMF SSM APS43 DMV"
Token = "SSM"
FindString Source Token Pos
Print "Token starting position is "
Print Pos
'Pos should print value of 4.
EndProgram
```

## CALLBasic Command - FloatToString

**Command:** **FloatToString**

**Description:** **FloatToString** returns the string value of a floating point variable. Sometimes you may need to retrieve a field with a type of 'Float' or 'Decimal' or 'Money' and use it in a function that requires a string. FloatToString allows you to do this.

**Format:** **FloatToString** *dest source*

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>dest</i>	String	A string variable that receives the converted value.
<i>source</i>	Floating Point	The floating point variable from which to get the value.

### Output:

None

This converts a floating point value from the variable *source*, and stores it in a string variable *dest*. This command is usually used when working with databases. Database records often use floating point values and to store decimal numbers for money values etc. For example, if the account balance is stored as a decimal value field (*fltBal*), then FloatToString allows you to convert the decimal value of *fltBal* to a string value to say balance to the caller.

### Example:

Program

```
'fltbal filed retrieved from a database field (e.g. fltbal = 3128.46)
```

```
fltbal = BLSTNDPvarBal.Value
```

```
'convert floating point value to string value
```

```
FloatToString strbal fltbal
```

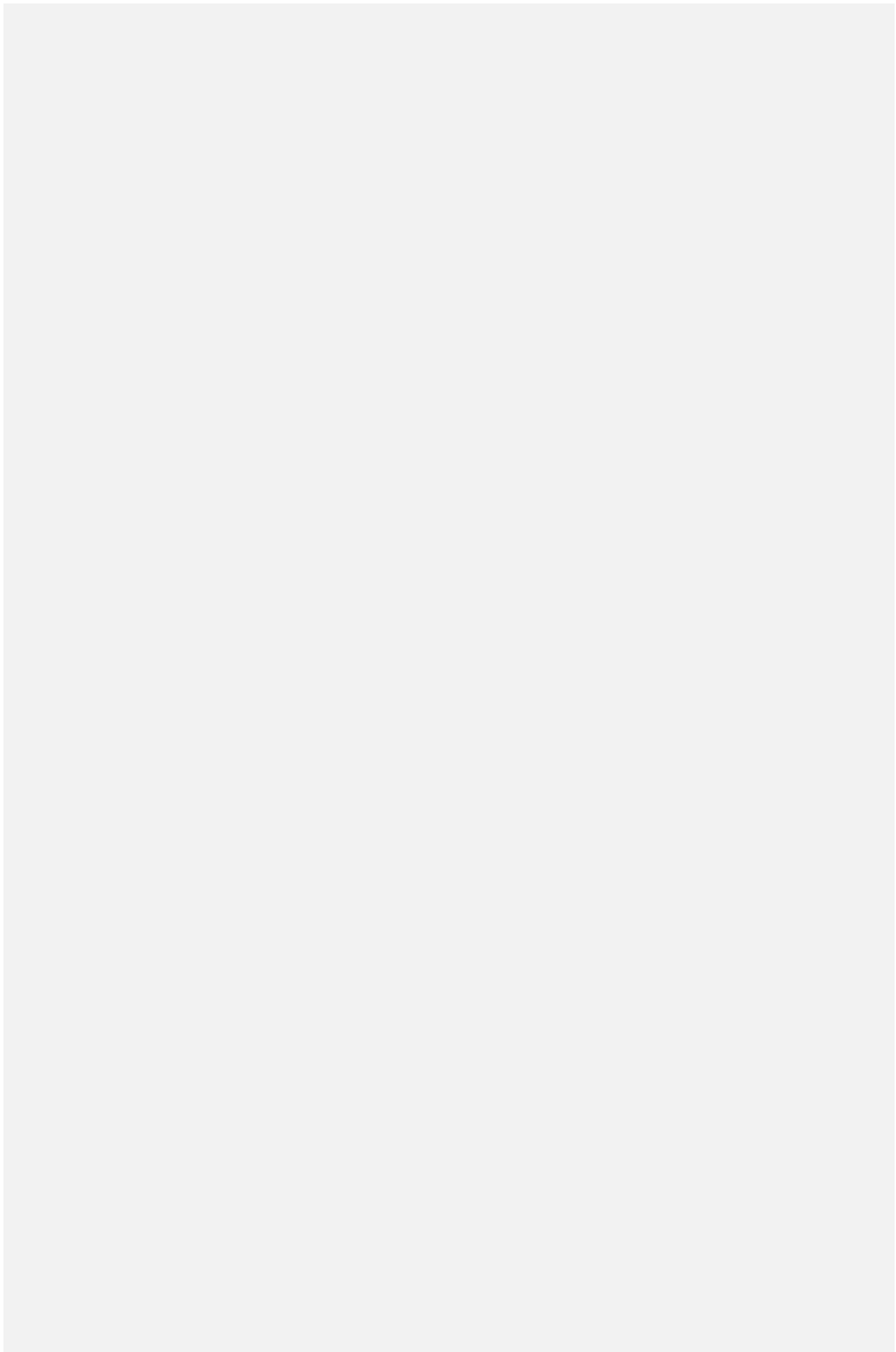
```
' Set variable saymoney = 1 to say strbal as money "Three thousand one hundred twenty three dollars  
and forty six cents"
```

```
saymoney = 1
```

```
SayNumber strbal
```

```
saymoney = 0
```

```
EndProgram
```



## CALLBasic Command - For...To...Step...Next

**Command:** **For...Next**

**Description:** The **For...Next** clause repeats a block of statements until the upper bound of the **Step** part of the clause is reached.

**Format:** **For** *variable* = *beg* **To** *end* **Step** *num* *command* **Next**

Parameter	Type	Description
<b>Input:</b>		
<i>variable</i>	Variable	Variable to be incremented
<i>beg</i>	Integer	The lower end of the scope to be iterated
<i>end</i>	Integer	The upper end of the scope to be iterated
<i>num</i>	Integer	The amount to increment
<i>command</i>	Commands	Any block of CallBASIC commands

**Output:**

None

**Example:**

Program

```
For count = 1 To 10 Step 1
```

```
    Print count
```

```
Next
```

'This will print this sequence: 1, 2, 3, 4, 5, 6, 7,8, 9 10 because count is being incremented by 1 after each pass of **Next**.

```
For count = 1 To 10 Step 2
```

Print count

Next

This will print this sequence: 1, 3, 5, 7, 9 because count is being incremented by 2 after each pass of

**Next.**

EndProgram

## CALLBasic Command - GetDate

**Command:** **GetDate**

**Description:** Returns Date components from a date field variable.

**Format:** **GetDate** *date mymonth myday myyear myhour myminute mysecond*

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>date</i>	DateTime	Variable that contains the date and time value.
-------------	----------	---

### Output:

<i>mymonth</i>	Integer	Month Value from <i>date</i> variable
----------------	---------	---------------------------------------

<i>myday</i>	Integer	Day Value from <i>date</i> variable
--------------	---------	-------------------------------------

<i>myyear</i>	Integer	Year Value from <i>date</i> variable
---------------	---------	--------------------------------------

<i>myhour</i>	Integer	Hour Value from <i>date</i> variable
---------------	---------	--------------------------------------

<i>myminute</i>	Integer	Minute Value from <i>date</i> variable
-----------------	---------	--

<i>mysecond</i>	Integer	Seconds Value from <i>date</i> variable
-----------------	---------	---

### Example:

Program

    'Returns date components based on \$today.

    date = \$today

**GetDate** date mon2 day2 year2 hour minute sec

EndProgram



**Example:**

Program

'Returns date components from a DateTime data type field from a database.

date = DateField

**GetDate** date mon2 day2 year2 hour minute sec

EndProgram

## CALLBasic Command - GetDayofWeek

**Command:** **GetDayofWeek**

**Description:** Returns integer value for day of week for a given date field

**Format:** **GetDayofWeek** *date dow*

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>date</i>	DateTime	A date type variable with the date value
-------------	----------	--

### Output:

<i>dow</i>	Integer	Day of week. (1 for Sunday to 7 for Saturday) Returns -1 if date submitted is invalid date (e.g. February 30 <sup>th</sup> ).
------------	---------	--

### Example:

```
Program
'Returns day of week using system date variable $today.
date = $today
GetDayofWeek date dow
EndProgram
```

## CALLBasic Command - GetDTMF

**Command:** **GetDTMF**

**Description:** Use this command to retrieve caller touch-tone input.

**Format:** **GetDTMF** *digits tmask ndigits timeout*

Parameter	Type	Description
<b>Input:</b>		
<i>digits</i>	String	Variable into which the caller input is placed
<i>tmask</i>	String	Variable defining termination character
<i>ndigits</i>	Integer	Number of digits to receive before automatically terminating input
<i>timeout</i>	Integer	Amount of time to wait for caller input

### Output:

None

### Example:

```
Program
  digits = ""
  tmask = "#"
  ndigits = 7
  timeout = 5
  GetDTMF digits tmask ndigits timeout
EndProgram
```

In the above example, if the caller presses 8 numbers, the first seven will be placed into the variable digits, if the caller presses "5678" then "#", the string "5678#" will be placed into the variable digits. If the caller pauses longer than 5 seconds without pressing anything, the command terminates and the next module is played.

Prior to executing the GetDTMF command, if you define an integer variable "iddelay", it will override the System Parameter "Prompt Inter DTMF Timeout" during the GetDTMF command. Once defined, you will need to execute a RemoveVar iddelay command to use default System Parameter "Prompt Inter DTMF Timeout" value.

## CALLBasic Command - GetField

**Command:** **GetField**

**Description:** The **GetField** command gets string variable in the field number specified of source variable starting from field 1.

**Format:** **GetField** *source target delimiter fieldnumber*

Parameter	Type	Description
<b>Input:</b>		
<i>source</i>	String	Variable to be searched
<i>target</i>	String	Variable containing the string in specified field number
<i>delimiter</i>		Delimiter used to separate source string in to multiple fields
<i>fieldnumber</i>		Field number in source starting at 1. Returns 0 if field number does not exist in source.

### **Output:**

None

### **Example:**

Program

```
' Initialize variables
'Source variable has comma delimited fields in this example
Source = "abc,def,ghi,jkl"
Delimiter = ","
FieldNumber = 3
Target = ""
GetField Source Target Delimiter FieldNumber
Print "Field# "
Print FieldNumber
```

```
Print " value is "  
PrintNL Target  
' Target variable should have "ghi"  
EndProgram
```

## CALLBasic Command - GetFloat

**Command:** **GetFloat**

**Description:** Converts a string source variable starting at a specified position to a floating point number and places it in the destination variable.

**Format:** **GetFloat** *dest source position*

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>dest</i>	Floating Point	A floating point variable that receives the converted value.
<i>source</i>	String	The string variable from which to get the floating point variable.
<i>position</i>	Integer	The starting position (counting from zero) in the source variable.

### Output:

None

Example:

Program

```
var1 = "123456789.95"
```

```
GetFloat dest var1 4
```

```
Print dest
```

```
' This program should print "56789.95"
```

```
EndProgram
```

## CALLBasic Command - GetInteger

**Command:** **GetInteger**

**Description:** Convert an integer from the source variable starting at a specified position to ASCII format and place it into the destination variable.

**Format:** **GetInteger** *dest source position*

Parameter	Type	Description
-----------	------	-------------

---

### Input:

<i>dest</i>	Integer	A integer variable that receives the converted value.
<i>source</i>	String	The string variable from which to get the integer.
<i>position</i>	Integer	The starting position (counting from zero) of the integer in the source variable.

### Output:

None

### Example:

```
Program
  var1 = "123456789"
  GetInteger dest var1 4
  Print dest
  ' This program should print '56789'
EndProgram
```



## CALLBasic Command - If...Then...Else...EndIf

**Command:** `If...Then...Else...EndIf`

**Description:** Test values for a specific condition to be true before executing a defined action. The condition is true if a math test has a value other than zero (0) or a logical expression has a Boolean true value.  
**Note:** All components of the If...Then...Else...EndIf command are required.

**Format:** `If condition Then command Else command EndIf`

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>condition</i>		Is a logical or mathematical test. The following are the valid mathematical symbols: >(greater than); >=(greater than or equal to); =(equal to); <(less than); <=(less than or equal to); <>(not equal to)
------------------	--	---

<i>command</i>		Any valid set of CALLBasic commands
----------------	--	-------------------------------------

### Output:

None

### Example 1:

```
Program
x = 2000
y = 500
If x<> 1000 Then
  If y <= 500 Then
    PrintNL "x not equal to 1000 and y less than or equal to 500"
  Else
    PrintNL "x not equal to 1000 and y greater than 500"
  EndIf
Else
  PrintNL "x equal to 1000"
EndIf
```

```
'This example should print 'x not equal to 1000 and y less than or equal to 500'  
EndProgram
```

#### Example 2:

```
Program  
  j = 100  
  If j < 100 Then  
    PrintNL "j less than 100"  
  Else  
    If j = 100 Then  
      PrintNL "j equal to 100"  
    Else  
      PrintNL "j greater than 100"  
    EndIf  
  EndIf  
  'This example should print 'j equal to 100'  
EndProgram
```

#### Example 3:

```
Program  
  j = 200  
  If j < 100 Then  
    PrintNL "j less than 100"  
  Else  
  EndIf  
  'This example should not have any statement printed.  
  'This example should only print statement when j is less than 100.  
EndProgram
```

## CALLBasic Command - IntegerToString

**Command:** `IntegerToString`

**Description:** IntegerToString returns the string value of an integer. Sometimes you may need to retrieve a field with a type of 'integer' and use it in a function that requires a string. IntegerToString allows you to do this.

**Format:** `IntegerToString dest source`

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>source</i>	Integer	The integer variable from which to get the integer.
---------------	---------	---

### Output:

<i>dest</i>	String	A string variable that receives the converted value.
-------------	--------	--

### Example 1:

```
Program
  var1 = 12345
  IntegerToString dest var1
  SayNumeral dest
  'This example should say 'one, two, three, four, five'
EndProgram
```

### Example 2:

```
Program
  IntegerToString dest $Calls
  SayNumeral dest
  'This example should say the number of calls in the system since last start-up
EndProgram
```

### Example 3:

```
Program
```

```
IntegerToString dest $year
SayNumber dest
  'This example should say the current year 'two thousand'
EndProgram
```

## CALLBasic Command - MakeDate

**Command:** **MakeDate**

**Description:** Returns a Date field from assigned integers. Can be used in the SayDate and SayTime command.

**Format:** **MakeDate** *mydate mymonth myday myyear myhour myminute mysecond*

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>mymonth</i>	Integer	The integer from which to get the month value.
<i>myday</i>	Integer	The integer from which to get the day value.
<i>myyear</i>	Integer	The integer from which to get the year value.
<i>myhour</i>	Integer	The integer from which to get the hour value.
<i>myminute</i>	Integer	The integer from which to get the minute value.
<i>mysecond</i>	Integer	The integer from which to get the second value.

### Output:

<i>mydate</i>	DateTime	Variable that receives the date value.
---------------	----------	--

### Example 1:

```
Program
  mymonth = 06
  myday = 22
  myyear = 2000
  myhour = 00
```

```
myminute = 00
mysecond = 00
MakeDate mydate mymonth myday myyear myhour myminute mysecond
Print mydate
'Returns mydate variable as a date data type in MM/DD/YY HH:MM:SS (military time format).
EndProgram
```

#### Example 2:

```
Program
dd = $today
GetDate dd mon2 day2 year2 hour minute sec
minute = minute + 5
MakeDate ddE mon2 day2 year2 hour minute sec
'Integer ddE will be 5 more minutes than the current time.
EndProgram
```

## CALLBasic Command - Max

**Command:** **Max**

**Description:** Returns the larger of two integer values compared.

**Format:** *var* = *var1* **Max** *var2*

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>var1</i>	Integer	Variable 1 to compare
-------------	---------	-----------------------

<i>var2</i>	Integer	Variable 2 to compare
-------------	---------	-----------------------

### Output:

<i>var</i>	Integer	Value of the larger of the 2 compared variables
------------	---------	---

### Example:

```
Program
  var1 = 80
  var2 = 1000
  var5 = var1 Max var2
  Print var5
  ' This program should print '1000'
EndProgram
```

**CALLBasic Command - MemMove**

**Command:** **MemMove**

**Description:** Move (copy) the characters of one string variable into another. This command dynamically allocates and reallocates space for the variable. Moving bytes (characters) into a destination that does not exist, or moving bytes that are not part of the source causes unpredictable behavior. **MemMove** automatically null terminates the variable.

**Format:** **MemMove** *dest source pos movecount*

Parameter	Type	Description
<b>Input:</b>		
<i>dest</i>	String	The target string variable where the memory will be moved.
<i>source</i>	String	The string variable from which the memory will be moved.
<i>pos</i>	Integer	The character starting position in <i>source</i> , counting from 0 as the first position. Must be previously defined variable.
<i>movecount</i>		The number of characters to be copied to <i>dest</i> . Must be previously defined variable.

<b>Output:</b>		
<i>dest</i>	String	The target string variable where the memory will be moved.

**Example 1:**  
Program  
Var2 = "123456"  
Var3 = "abcd"  
Pos = 0  
Count = 6  
**MemMove** Var2 Var3 Pos Count



```
PrintNL Var2
'The print results would show value of Var2 as 'abcd'
EndProgram
```

#### Example 2:

```
Program
Var2 = "123456"
Var3 = "abcd"
Pos = 2
Count = 4
MemMove Var2 Var3 Pos Count
PrintNL Var2
'The print results would show value of Var2 as '12abcd'
EndProgram
```

## CALLBasic Command - Min

**Command:** **Min**

**Description:** Return the smaller of two values from comparing integer variables *var1* and *var2*.

**Format:** *var* = *var1* **Min** *var2*

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>var1</i>	Integer	Variable 1 to compare
-------------	---------	-----------------------

<i>var2</i>	Integer	Variable 2 to compare
-------------	---------	-----------------------

### Output:

<i>var</i>	Integer	Value of the smaller of the 2 compared variables
------------	---------	--

### Example:

```
Program
  var1 = 80
  var2 = 1000
  var5 = var1 Min var2
  Print var5
  ' This program should print '80'
EndProgram
```

## CALLBasic Command - Play

**Command:** **Play**

**Description:** Play a digital audio file pointed to by a CALLMaster "file ID" number. CALLMaster controls and maintains all speech files via their file ID numbers. Prompts can be played by providing their prompt ID numbers given in Appendix A. Any system speech file can be played via its file ID number found in a "protected field" of the database record. If you do not know the file ID, use the **PlayFileName** command.

The Play command used in conjunction with the RecordAppend command can provide dictation functionality (fast forward, pause, rewind).

See [Walkthrough: Dictation Functionality with Playback Navigation](#)

**Format:** **Play** *file\_id dlg\_event dlg\_result*

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>file_id</i>	Integer	An integer variable with value of a CALLMaster speech "file ID" number or one returned by a previous <b>Record</b> command.
----------------	---------	---

### Output:

<i>dlg_event</i>		Returns the Dialogic Connection Type Event (see Appendix B).
------------------	--	--

<i>dlg_event</i>		Returns the Dialogic Connection Type Event (see Appendix B).
------------------	--	--

In addition to playing a speech file, the Play command provides navigation/playback options - 1 for fast forward; 2 for pause/continuation; 3 for rewind; any other key to terminate play back. There are 3 system variables that are required for this capability:

- **playbackposition** - an integer variable that:
  - contains the position in bytes to start the playback. The variable "playbackposition", returned at the end of this command, contains an integer value of the recording pointer position. If you have previously issued a Play, PlayFileName, Record or RecordAppend command, then you should reset variable playbackposition to 0 if you want to play selected file from its start.
  - enables playback options (i.e. 1 for fast forward; 2 for pause/continuation; 3 for rewind; any other key to terminate play back). Use the RemoveVar command to remove

“playbackposition” variable and disable associated playback options.

- termdigit - and can be used for play back analysis (i.e. 1 for fast forward; 2 for pause/continuation; 3 for rewind; any other key will terminate play back). If “playbackposition” variable is defined, then the single entry tone entered by caller during play back will be returned in variable “termdigit”. The variable “termdigit” can be queried in subsequent scripts to determine tone entered by caller for further processing. This variable will be set to “T” if no response is received during playback pause state.
- ttcutthrough - an integer variable that enables/disables the caller's ability to “cut through” the play back by entering the desired key to stop hearing the message played. Set variable “ttcutthrough” to one (1) in any script to enable cut through. The variable “ttcutthrough” must be reset to zero after the Play command if you do not want cut-through for subsequent Play commands.

If either “playbackposition” or “ttcutthrough” variable is defined, then play back will be terminated with entry of any single tone. The value of variable “playbackposition” will be used to determine starting position of next Play/PlayFileName command.

The Play command used in conjunction with the RecordAppend command can provide dictation capabilities.

#### Example 1:

```
Program
  PromptId = 202
  Play PromptId e r
  'Plays prompt 202, the word "and," the returns the Dialogic event and result into e and r, respectively.
EndProgram
```

#### Example 2:

```
Program
  Prompt = 177
  RecTime = 60
  t = "0"
  a = ""
  ttcutthrough = 1
  Record recfileid Prompt RecTime a t e r
  playbackposition = 0
  Play recfileid c r
  ttcutthrough = 0
  'Plays Prompt 177 “Record prompt”, records caller’s message, then plays back the recorded message.
EndProgram
```

**CALLBasic Command - PlayFileName**

**Command:** **PlayFileName**

**Description:** Play a digital audio file pointed to by a FAT File System name, with full drive and path support.  
Play a digital audio file pointed to by a FAT File System name, with full drive and path support. Functionality provided by variables “playbackposition”, “tcutthrough” and “termdigit” as described in the Play command also apply to this command.

**Format:** **PlayFileName** *filename dlg\_event dlg\_result*

Parameter	Type	Description
<b>Input:</b>		
<i>filename</i>	String	Variable with a valid FAT File System name with path;
<b>Output:</b>		
<i>dlg_event</i>		Returns the Dialogic Connection Type Event (see Appendix B).
<i>dlg_result</i>		Returns the Dialogic Call Result (see Appendix B).

**Example:**  
Program  
    'Note: fileld string variable can contain spaces  
    fileld = "C:\Program Files\CALLMaster\vox\833\51.wav"  
    **PlayFileName** fileld e r  
    'This will play file named 51.wav in the c:\Program Files\CALLMaster\Vox\833 folder.  
EndProgram

## CALLBasic Command - Print

**Command:** **Print**

**Description:** Print the value designated to the CMCalls <DATE>.TXT file. Separate multiple variables by spaces. The more common way to Print variables is via the 'Text for data file output' field of the Audiotex module; put *!variable* in the 'Text for data file output' field. The result will be printed into the ' CMCalls <DATE>.TXT ' file.

**Format:** **Print** *variable*

Parameter	Type	Description
-----------	------	-------------

---

### Input:

<i>variable</i>	String	The variable to Print to the ' CMCalls <DATE>.TXT ' file.
-----------------	--------	---

### Output:

None

### Example:

Program

**Print** answer3

'This will Print the value in *answer3* to the CMCalls <DATE>.TXT file.

EndProgram

## CALLBasic Command - PrintNL

**Command:** **PrintNL**

**Description:** This is the same as the 'Print' command except that it adds a newline character to the end of the string.

**Format:** **PrintNL** *variable*

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>variable</i>	String	The variable to Print to the 'CMCalls <DATE>.TXT' file.
-----------------	--------	---

### Output:

None

### Example:

Program

var1 = "This is line one "

var2 = "and this is line two"

**PrintNL** var1

Print var2

'This example produces: This is line one

'and this is line two

'If 'Print' had been used instead of 'PrintNL', the result would be a single print line: This is line one and this is line two.

EndProgram

CALLBasic Command - PutEvent

Command: PutEvent

Description: Places an event onto the current event queue. By selecting a channel in the command line of PutEvent, you can activate that port to perform the Audiotex routine designated by the Audiotex start module for that port. You can create an Audiotex routine to make a call out, or make use of any CALLBasic command on the secondary port.

Format: PutEvent channel event\_num dlg\_result

Parameter	Type	Description
<b>Input:</b>		
channel	Integer	A phone port number
event_num		The Event Number (See Dialogic Documentation)
<b>Output:</b>		
dlg_result	Integer	The Dialogic Call Result returned (See <a href="#">Appendix B</a> )

Example:

```
Program
  channel = 2
  event_num = 134
  dlg_result = 10
  PutEvent channel event_num dlg_result
  'This command will put a ring event onto port 2. The event number 134 means rings received. The result
  (dlg_result) is the returned result from the event.
  'This command could trigger port 2 to go off hook. Port 2 would then proceed to perform its predefined
  Phone Port action.
EndProgram
```



## CALLBasic Command - Recognize

**Command:** Recognize

**Description:** Recognizes a spoken word or phrase from a list/file and returns the word position. This function is part of the Automatic Speech Recognition (ASR). ASR features need to be purchased in addition to CALLMaster software and required license activation.

**Format:** Recognize *AudioFile TexttoRecognize Type RecognizedWords WordCount wPos wValStr ASROK ResultTree*

Parameter	Type	Description
<b>Input:</b>		
<i>AudioFile</i>	String	Name of file with path that has the recording to be recognized
<i>TexttoRecognize</i>	String	Text to be recognized. This can be a list of words separated by commas or the name of a file containing the words separated by commas or XML Grammar.
<i>Type</i>	String	Variable containing either TEXT   FILE   XMLFILE. Used to determine what the <TexttoRecognize> parameter contains. If TEXT then <TexttoRecognize> parameter contains a string of the words that can be recognized. If FILE then <TexttoRecognize> parameter contains the name of a ASCII file that has the words to be recognized. If XMLFILE then <TexttoRecognize> parameter contains a file with xml grammar.
<b>Output:</b>		
<i>RecognizedWords</i>	String	Word(s) that were recognized
<i>WordCount</i>	Integer	Number of words recognized

<i>wPos</i>	Integer	The recognized word/phrase position in the list or the XML PropertyID of the word recognized
<i>wValStr</i>	String	The XML VALSTR attribute for the recognized words/Phrases.
<i>ASROK</i>	Integer	The status of the recognize process. 1 - Pass or 0 - Fail. Determines whether the returned values are valid.
<i>ResultTree</i>	String	The semantic tree of the spoken sentence. Each word recognized is preceded with a tree node address and then the recognized word. This can then be parsed to capture the Rule from the XML grammar recognized and the spoken response.

#### Example:

Program

TexttoRecognize = "C:\Program Files\Callmaster\Data\OrderCommands.txt"

wValue = 0

Type = "FILE"

ASROK = 0

ResultTree = ""

**Recognize** AudioFileName TexttoRecognize Type RecognizedWords WordCount wPos wValStr

ASROK ResultTree

'Recognize the words in the AudioFile recording from the words in the TexttoRecognize file

'Use system variables \$ASRConfidence, \$ASRReqConfidence, \$ASRSAPIconfidence for debugging/testing (See [System Variables](#) for details).

'See Sample scripts in the CALLMaster\Scripts\Sample Scripts\SpeechRecognition folder for details.

EndProgram

## CALLBasic Command - Record

**Command:** **Record**

**Description:** Record a digital audio file. When complete the command returns a speech "file ID" number used for file identification. CALLMaster controls and maintains all speech files via their "file ID" numbers. CALLMaster creates and increments sequential "file ID" numbers when making a new recording. Since all recordings are in the ' \CALLMaster\vox' directory it is up to the application to rename the file, thereby moving the recordings to their custom speech directory for future retrieval using this command. Applications that maintain speech files not indexed by any CALLMaster database file must store the files in a folder not used by CALLMaster. The command requires variable that must be previously defined. CALLMaster will issue a 'beep tone' at the start of the recording. If you do not want to receive this tone, define a variable 'NORECTONE = 1' before issuing the record command.

**Format:** **Record** *recfileid promptid Maxsec abort\_dtmf term\_dtmf dlg\_event dlg\_result*

Parameter	Type	Description
<b>Input:</b>		
<i>recfileid</i>	Integer	Contains the recorded speech "file ID" number used for future access. If this is zero before recording starts CALLMaster assigns a unique number which you should save.
<i>promptid</i>	Integer	The file to play before recording starts (usually recording instructions). Variable must be previously defined.
<i>Maxsec</i>	Integer	Maximum amount of time in seconds for recording. Variable must be previously defined.
<i>abort_dtmf</i>	String	Reserved for future use. Required as a placeholder.
<i>term_dtmf</i>	String	Returns the touch-tone that terminated the recording.

**Output:**

<i>recfileid</i>	Integer	Contains the recorded speech "file ID" number used for future access. If this is zero before recording starts CALLMaster assigns a unique number which you should save.
<i>dlg_event</i>	Integer	Returns the Dialogic Connection Type Event (see <a href="#">Appendix B</a> ).
<i>dlg_result</i>	Integer	Returns the Dialogic Call Result (see <a href="#">Appendix B</a> ).

#### Example:

Program

PromptID = 100

Maxsec = 30

t = "\*"

a = ""

recfileid = 0

**Record** recfileid PromptID RecTime a t e r

'CALLMaster first plays PromptID 100 "Press star to stop recording" (see [Appendix A](#)), and records a file for up to 30 seconds (Maxsec).

'When done, the "file ID" for this file is in the recfileid variable, and the variable termtdt has the touch-tone "\*" that stopped the recording.

'The e and r variables return the Dialogic Connection Type Event and Call Result. The remaining variable a is required.

EndProgram

## CALLBasic Command - RecordAppend

**Command:** **RecordAppend**

**Description:** Record at the end of a previously recorded file. A variable "playbackposition", returned at the end of this command, contains an integer value of the recording pointer position.

The RecordAppend command used in conjunction with the Play command can provide dictation functionality (fast forward, pause, rewind).

See [Walkthrough: Dictation Functionality with Playback Navigation](#)

**Format:** **RecordAppend** *recfileid promptid Maxsec abort\_dtmf term\_dtmf dlg\_event dlg\_result*

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>recfileid</i>	Integer	The recorded speech "file ID" number. If set to zero, then a new recfileid is created. Before executing initial RecordAppend command or to have caller re-record, initialize recfileid variable to zero to create new file. Subsequent RecordAppend commands should keep the recfileid from previous commands so new recording can be appended to existing recfileid.
<i>promptid</i>	Integer	The file to play before recording starts (usually recording instructions). Variable must be previously defined.
<i>Maxsec</i>	Integer	Maximum amount of time in seconds for recording. Variable must be previously defined.
<i>abort_dtmf</i>	String	Reserved for future use. Required as a placeholder.
<i>term_dtmf</i>	String	Initialize variable

### Output:

<i>recfileid</i>	Integer	Contains the recorded speech "file ID" number used for future access. If this is zero before recording starts CALLMaster assigns a unique number which you should save.
<i>term_dtmf</i>	String	Returns the touch-tone that terminated the recording.
<i>dlg_event</i>	Integer	Returns the Dialogic Connection Type Event (see <a href="#">Appendix B</a> ).
<i>dlg_result</i>	Integer	Returns the Dialogic Call Result (see <a href="#">Appendix B</a> ).

#### Example:

```

Program
  PromptID = 100
  Maxsec = 30
  t = "*"
  a = ""
  recfileid = 100011
  RecordAppend recfileid PromptID Maxsec a t e r
  'Append recording to existing file. CALLMaster plays prompt 100 that says, "Press star to stop
  recording" (see Appendix A),
  'records a file at the end of the current file recfileid for up to 30 seconds. If the caller pressed star (*) to
  stop recording, the variable t equals "*";
  'The e and r variables return the Dialogic Connection Type Event and Call Result. The remaining variable
  a is required.
  EndProgram

```

#### Example:

```

Program
  'Prompt 92 says "At the tone, say your name"
  Pmt92 = 92
  'Prompt 93 says "At the tone, record your personal greeting"
  Pmt92 = 93
  'Prompt 94 says "At the tone, record your reply"
  Pmt92 = 94
  Prompt = 100
  RecTime = 30
  t = "*"
  a = ""
  recfileid = 0
  Play Pmt92 c r
  RecordAppend recfileid Prompt RecTime a t e r
  Play Pmt93 c r
  RecordAppend recfileid Prompt RecTime a t e r
  Play Pmt94 c r
  RecordAppend recfileid Prompt RecTime a t e r
  Play recfileid c r
  'Create a new recording file and append subsequent answers to newly created recording.
  'CALLMaster asks caller several questions, starts a new recording file and then appends recorded
  answers in one single recording.
  EndProgram

```

**CALLBasic Command - RemoveVar**

**Command:** RemoveVar

**Description:** Remove a variable from local memory.

**Format:** RemoveVar *targetvar*

Parameter	Type	Description
<b>Input:</b>		
<i>targetvar</i>		The variable to be removed.
<b>Output:</b>		
None		

**Example:**  
Program  
var1 = "This is a variable"  
PrintNL var1  
RemoveVar var1  
PrintNL var1  
'This example prints 'This is a variable' on the first PrintNL but generates an execution error on the second since var1 has been removed  
EndProgram

**CALLBasic Command - RemoveVars**

**Command:** RemoveVars

**Description:** Clear all the local variables for this port.

**Format:** RemoveVars

Parameter	Type	Description
<b>Input:</b>		
None		
<b>Output:</b>		
None		

**Example:**  
Program  
  RemoveVars  
  ' All of the local variables for this port are removed.  
EndProgram



## CALLBasic Command - Return

**Command:** **Return**

**Description:** Use a string or integer variable to determine which Mailbox or Attendant ID to access; or which Audiotex module to proceed to based on the Audiotex Module Action. The **Return** command is used only with Module Action of 'Mailbox', 'Attendant ID', or 'Log In', or with an action of 'Next Module'. **Return** overrides any value placed in the 'Next Module ID' field.

**Format:** **Return** *variable*

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>variable</i>	Integer or String	Value of the module to return control
-----------------	-------------------	---------------------------------------

### Output:

None

### Example 1:

Program

CallerInput = \$tt

**Return** CallerInput

'Depending on the Audiotex 'Module Action' field, this would route to an Audiotex, Mailbox or Attendant record based on the input by the caller.

EndProgram

### Example 2:

Program

**Return** 5000

'The call control would route to Audiotex module 5000 if the 'Module Action' field of the Audiotex module running the Return script is set to 'Next Module'.

'The call would route to mailbox 5000 if the 'Module Action' field was set to 'Mailbox' or 'Log In'

'The call would route to Attendant ID 5000 if the 'Module Action' field was set to 'Attendant ID'.

EndProgram

## CALLBasic Command - SayNumber | SayNumeral | SayOrdinal

**Command:** SayNumber | Numeral | Ordinal

**Description:** Say the contents of a string variable in number, numeral, or ordinal format.  
By initializing a variable "saymoney" to one (1), this command will say the numbers in dollars and cents format (e.g. fifty two dollars and twenty one cents). To use a different currency, re-record prompts 381, 382, 383, 384.  
By initializing a variable "sayreal" to one (1), this command will say the numbers in decimal format (e.g. fifty two point twenty one).  
Set variable "speakprecision" to value you want for the decimal precision using the sayreal command.  
The variables "saymoney" and "sayreal" must be reset to zero to normalize Say command.

**Format:** SayNumber | Numeral | Ordinal *variable*

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>variable</i>	String	Contains the number to say. The SayNumber Maximum value is 999,999,999. The SayNumeral has an unlimited value. The SayOrdinal Maximum value is 32,999.
-----------------	--------	---

### Output:

None

### Example 1:

Program

```
strNum = "123"
```

```
'CALLMaster says "one hundred twenty-three"
```

```
SayNumber strNum
```

```
'CALLMaster says "one,two,three"
```

```
SayNumeral strNum
```

```
'CALLMaster says "one hundred twenty-third"
```

### **SayOrdinal** strNum

'CALLMaster says "one hundred twenty-three dollars and forty six cents" (round to 2 decimal places)

saymoney = 1

strMoney = "123.456"

**SayNumber** strMoney

'CALLMaster says "one hundred twenty-three dollars and forty five cents" (round to 2 decimal places)

strMoney = "123.454"

**SayNumber** strMoney

'CALLMaster says "one hundred twenty-three point four five six seven" (precision set to 4 decimal digits, used to say negative numbers)

saymoney = 0

sayreal = 1

speakprecision = 4

strReal = "-123.4567"

**SayNumber** strReal

sayreal = 0

EndProgram

### Example 2:

Program

IntegerToString strCalls \$Calls

**SayNumber** strCalls

'The above command converts the system variable *\$calls* from integer format to string and says the number.

EndProgram

### Example 3:

Program

SayNumeral \$tt

'The command will speak the value of the caller input string \$tt in numeral format.

EndProgram

## CALLBasic Command - SayDate

**Command:** **SayDate**

**Description:** Say date(month and day) from system variable \$today or from a database Date/Time data type field.

**Format:** **SayDate** *date*

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>date</i>	DateTime	A variable with the time value from <b>\$today</b> system variable or a database Date/Time data type field.
-------------	----------	---

### Output:

None

### Example:

Program

**SayDate** \$today

Print \$today

'The SayDate command will play the current month and day (e.g. January 20<sup>th</sup> ), which comes from the system variable \$today.

'The Print statement will print the date as '01/20/00 14:31:27'

EndProgram

**CALLBasic Command - SayTime**

**Command:** SayTime

**Description:** Say time(hour, sec with AM or PM) from the system variable \$today or from a database Date/Time data type field.

**Format:** SayTime *time*

Parameter	Type	Description
<b>Input:</b>		
<i>time</i>	DateTime	A variable with the time value from <b>\$today</b> system variable or a database Date/Time data type field.

**Output:**

None

**Example:**  
Program  
    **SayTime** \$today  
    'This command will say the current time. (e.g. five, forty seven AM)  
EndProgram

**CALLBasic Command - SendUDP**

**Command:** SendUDP

**Description:** Sends a datagram packet with the UDP protocol to the host defined in the UDPIP system flag address. This requires the UDPIP<address>:<Port#> flag set in the System Parameter 'System Settings'.

**Format:** SendUDP *Datagram*

Parameter	Type	Description
-----------	------	-------------

**Input:**

<i>Datagram</i>	String	Data Packet with the UDP protocol.
-----------------	--------	------------------------------------

**Output:**

None

**Example:**

```
Program
  Datagram = "!0101ALTINQ 1234503"
  UDPRecvd = "[Data not available]"
  SendUDP Datagram
  'The returned packet has the port# in the first 2 bytes of the packet and the whole packet received is
  returned in UDPRecvd variable
EndProgram
```

**CALLBasic Command - SetHook**

**Command:** **SetHook**

**Description:** Enables a line to go on hook or off hook.

**Format:** **SetHook** *hook state*

Parameter	Type	Description
<b>Input:</b>		
<i>hook state</i>	String	Hookstate to set line. Values include 'ONHOOK' or 'OFFHOOK'

**Output:**

None

**Example:**  
Program  
hookstate = "OFFHOOK"  
**SetHook** hookstate  
'This command will set line to offhook state to enable processing.  
EndProgram

## CALLBasic Command - Sleep

**Command:** **Sleep**

**Description:** Sets a delay in the processing based on the *sleeptime* variable (must be a variable and cannot be a literal).

**Format:** **Sleep** *sleeptime*

Parameter	Type	Description
-----------	------	-------------

---

### Input:

<i>sleeptime</i>	Integer	sleep time is the milliseconds.
------------------	---------	---------------------------------

### Output:

None

### Example:

```
Program
  sleeptime = 100
  Sleep sleeptime
EndProgram
```



## CALLBasic Command - Speak

**Command:** **Speak**

**Description:** Convert text variable to speech using Microsoft SAPI engine. To use this function, you must have Microsoft's SAPI engine installed. For details and download, see [CALLMaster Text-to-Speech](#).

**Format:** **Speak** *text type*

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>text</i>	String	Variable to be spoken or a full path\filename
<i>type</i>	String	Containing either the string "TEXT" or "FILE" indicating the value of what the first parameter represents

### Output:

None

### Example 1:

```
Program
saythis = "Hello from Speech Soft CALLMaster. You are hearing a sample of text to speech with the
Microsoft SAPI speech engine using the Speak Text option."
type = "TEXT"
PrintNL saythis
Speak saythis type
EndProgram
```

### Example 2:

```
Program
'The next set of lines are used to select a specific voice from the SAPI voice folder.
'You must have purchased and installed these additional voices to use.
TexttoSpeak = AnyTextFieldFromDatabase
type = "TEXT"
saythis = "<voice required=" & $DQuote
```

```
saythis = saythis & "Name=Microsoft Sam"  
saythis = saythis & $DQuote  
saythis = saythis & "> "  
saythis = saythis & TexttoSpeak  
PrintNL saythis  
  Speak saythis type  
EndProgram
```

### Example 3:

```
Program  
  sayfile = "c:\program files\CALLMaster\speechfiles\filename"  
  type = "FILE"  
  PrintNL sayfile  
  Speak sayfile type  
EndProgram
```

## CALLBasic Command - StringConcat

**Command:** **StringConcat**

**Description:** Concatenate one variable string to the end of another.

**Format:** **StringConcat** *destvar sourcevar*

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>destvar</i>	String	The variable that will receive characters at the end of its current string of characters, if any.
<i>sourcevar</i>	String	The variable that has the characters to copy from.

### Output:

None

### Example 1:

```
Program
  lastname = "Smith "
  firstname = "Joe"
  StringConcat lastname firstname
  Print lastname
  'This will copy to the end of any characters in lastname all the characters in firstname.
  'This will print 'Smith Joe'
EndProgram
```

### Example 2:

```
Program
  var1 = "To this "
  var2 = "add this"
  StringConcat var1 var2
  PrintNL var1
  This would print 'To this add this'
```

EndProgram

**CALLBasic Command - StringCopy**

**Command:**           **StringCopy**

**Description:**       Pronounced 'string copy.' Copy one variable string to another, overwriting any characters in the receiving variable. StringCopy dynamically reallocated the length of the variable and will terminate it. Treats variables as character strings, as opposed to integers.

**Format:**           **StringCopy** *dest source*

Parameter	Type	Description
<b>Input:</b>		
<i>dest</i>	String	Variable which should get the new value
<i>source</i>	String	Variable from which to copy the value
<b>Output:</b>		
<i>dest</i>	String	Variable returned with the new value

**Example 1:**  
Program  
  Var1 = "anystring"  
  **StringCopy** Var1 "abc"  
  'The variable Var1 gets set to the letters 'abc.'  
EndProgram

**Example 2:**  
Program  
  StringCopy x \$tt  
  'The variable x becomes equal to the touch-tones entered by the caller in this module. If the caller presses '001,' x will equal '001'.  
EndProgram

## CALLBasic Command - StringLength

**Command:** **StringLength**

**Description:** Return the string length. The value equals the number of characters in the string; the null terminator is not included in the count.  
Note: source must be a defined variable. This command will not accept a direct string.

**Format:** **StringLength** *length* *source*

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>source</i>	String	Variable whose number of characters will be counted including spaces.
---------------	--------	---

### Output:

<i>length</i>	Integer	Returns an integer value equal to the number of characters of the string in the variable called <i>source</i> .
---------------	---------	---

### Example 1:

Program

zipcode = "12345"

**StringLength** ziplength zipcode

'The variable *ziplength* will contain the value of the number of characters in *zipcode*. If *zipcode* was 12345, then *ziplength* will be 5.'

EndProgram

### Example 2:

Program

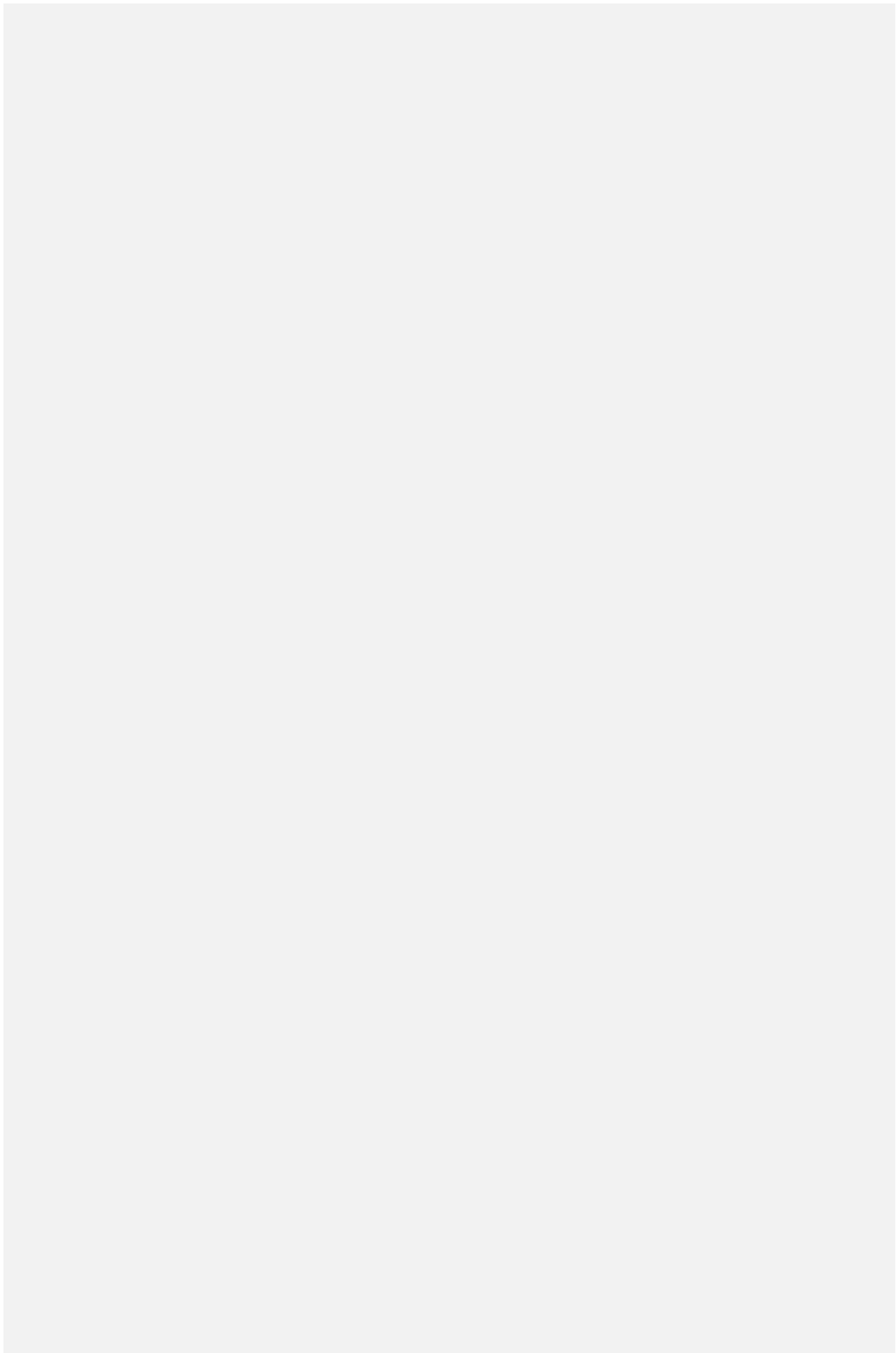
var1 = "tricadecaphobia"

**StringLength** length var1

PrintNL length

'This code should print '15'

EndProgram



**CALLBasic Command - StringNConcat**

**Command:** StringNConcat

**Description:** Concatenate up to n characters from the beginning of one variable to the end of another variable. (If n, the number of characters to copy, is greater than the number of characters in the source variable results will be unpredictable. If you do not know the length of the source variable, we recommend using **StringLength** to find out.)  
Note: dest and source must be defined variables. This command will not accept direct strings.

**Format:** StringNConcat *dest source n*

Parameter	Type	Description
<b>Input:</b>		
<i>dest</i>	String	Variable that should receive characters at the end of its current string.
<i>source</i>	String	Contains the string of characters to be copied.
<i>n</i>	Integer	Number of characters to concatenate.
<b>Output:</b>		
<i>dest</i>	String	Variable returned with received characters at the end of its current string.

**Example:**  
Program  
  var1 = "Strauss"  
  var2 = "Ronald"  
  **StringNConcat** var1 var2 3  
  PrintNL var1  
  'This program should print 'StraussRon'  
EndProgram



## CALLBasic Command - Transcribe

**Command:** **Transcribe**

**Description:** Transcribes the spoken word or phrase into a string variable or ASCII file. This function is part of the Automatic Speech Recognition (ASR). ASR features need to be purchased in addition to CALLMaster software and requires license activation. Note: The Microsoft speech engine's transcription capability is not accurate over the telephone.

**Format:** **Transcribe** *AudioFile Text ASRType*

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>AudioFile</i>	String	Name of voice file with path that has the recording to be transcribed
------------------	--------	---

<i>Type</i>		TEXT or FILE where TEXT means the prior variable will contain the words transcribed, FILE points to the file name assigned in the TEXT parameter to receive the transcribed words
-------------	--	---

### Output:

<i>Text</i>		Transcribed text word - should be set to nulls to get back transcribed text as variable if the Type parameter is set to TEXT - should be set to file name to get back transcribed text in the file if the Type parameter is set to FILE
-------------	--	---

### Example:

```
Program
FileIDToName AudioFileName recfileid
Text = ""
ASRType = "TEXT"
'Recognize the words in the Text field that were spoken
Transcribe AudioFileName Text ASRType
ptext = "Transcribed words are: " & Text
Print ptext
```

' This will print the transcribed words.  
EndProgram

**CALLBasic Command - var**

Command:                **var =**

Description:            Assign an integer or string value to a local, global, or disk variable. The command creates the variable if it doesn't already exist. The variable length is dynamically reallocated, and the variable is automatically null terminated. Variable names can be any length (although long names are easy to read, they take up memory) and can contain any valid printable ASCII character with no embedded spaces. All variables display data in a standard ASCII format even for number values.

Format:                 *variable* = *variable* | constant | system variable

Parameter	Type	Description
<hr/>		
Input:		
<i>variable</i>	String or Integer	A local or global temporary memory storage area.
Output:		
None		

Example:

```
Program
  ansr1 = "0"
  'This will make ansr1 a string variable with the value zero.
  lastcolor = 100
  @color = lastcolor
  'This will make the global variable @color an integer variable with the value 100.
  input = $tt
  'This will make input a string variable equal to the caller's last touch-tone response.
EndProgram
```

## CALLBasic Command - While...WEnd

**Command:** **While...WEnd**

**Description:** The While...WEnd loop repeats a block of statements while a condition is True.

**Format:** **While** *condition* ...**WEnd**

Parameter	Type	Description
-----------	------	-------------

### Input:

<i>condition</i>		Is a logical or mathematical test. The following are the valid mathematical tests: >(greater than); >=(greater than or equal to); =(equal to); <(less than); <=(less than or equal to); <>(not equal to)
------------------	--	---

### Output:

None

### Example 1:

```
Program
  num = 1
  While num < 10
    Print num
    num = num + 1
  WEnd
'This example will print 1 2 3 4 5 6 7 8 9
EndProgram
```

### Example 2:

```
Program
  num = 1
  endnum = 10
  While num < endnum
    Print num
    num = num + 2
  WEnd
```

```
'This example will print 1 3 5 7 9  
EndProgram
```

## Database Access

[Overview](#)

[Database Commands](#)

[Example of Using Database Commands](#)

## Database Access Overview

The CALLMasterSQL.mdf is a Microsoft SQL Database, which includes tables representing all of the items designated in the CALLMaster Manager under the 'Manage' and 'Parameters' menu items. The CALLMaster Manager is a Graphical User Interface used to make changes to the various CALLMaster tables. When changes need to be made or information needs to be extracted from the tables interactively with the caller, this is done in CALLBasic using *Recordset Objects* or *Command Objects*.

There are four database objects used in CALLBasic, Recordset objects, Fields objects, Field objects and Connection objects. All objects are created by using the **Set** command, i.e. **Set rst = New Recordset**. The **New** keyword allocates enough memory for a variable of the size of 'Recordset'. All objects have **Methods** and **Properties**. Methods are functions associated with an object to perform a particular action on that object. Properties contain the values of the attributes of an object. For example, the name of a particular field object is accessed through the 'Name' property, i.e. **strFieldName = object.Name**.

### Recordset Objects

Recordset Objects are variables representing various tables of the database. Once memory is allocated for the object using the New command, a particular table can be assigned to the object using the Open command. CALLBasic knows which table to open by the two parameters of the .Open method. The first is an SQL statement, which determines which table is to be used, and the second is a connection string, which indicates the database that contains the table. Once we establish a recordset object, we can use the methods (functions) and properties of that recordset.

### Fields Objects

Fields objects are member objects of the 'Recordset' object. Fields objects are collections of all the fields within a certain recordset.

### Field Objects

Field objects are member objects of the 'Fields' object. Each field is accessed via a parameter, which represents its ordinal position in the 'Fields' collection, i.e. *object.Field 2* represents the 'BoxNumber' field of the **Mailboxes** recordset.

### Command Objects

Command objects are another way of accessing data. Command objects are used when performing advanced SQL commands or when you need to use multiple connections.

Before objects can be created, resources must be allocated using the **Set** command along with the **New** keyword, and must be de-allocated using the **Set** command with the **Nothing** keyword, i.e. **Set rst = Nothing**

In CALLBasic scripting you can check if a database error was encountered using the system variable \$DBError. If CALLMaster encounters a database error that cannot be trapped, the call flow is directed to the Fail Module ID during Audiotex.

## Database Commands

Command	Description
<a href="#"><u>ActiveConnection</u></a>	Establishes a connection with a database
<a href="#"><u>AddNew</u></a>	Add a new record to a recordset
<a href="#"><u>BOF / EOF</u></a>	Contains a zero (False) or non-zero (True) value determining the Beginning Of File or the End Of File
<a href="#"><u>Close</u></a>	Close the recordset
<a href="#"><u>Count</u></a>	Contains the number of fields in a recordset
<a href="#"><u>Delete</u></a>	Delete the current record from the recordset
<a href="#"><u>Execute</u></a>	Executes an SQL statement
<a href="#"><u>Fields / Field</u></a>	Fields is a collection object of fields in a recordset where Field represents a particular field of a table.
<a href="#"><u>MoveFirst</u></a>	Move to the first record of a recordset
<a href="#"><u>MoveLast</u></a>	Move to the last record of a recordset
<a href="#"><u>MoveNext</u></a>	Move forward one record at a time through a recordset
<a href="#"><u>MovePrevious</u></a>	Move backward one record at a time through a recordset
<a href="#"><u>Name</u></a>	The name of the field
<a href="#"><u>Open</u></a>	Open a recordset for use
<a href="#"><u>SetValue</u></a>	Changes the value of the current records fields
<a href="#"><u>Type</u></a>	The type of field
<a href="#"><u>Update</u></a>	Update changes made to a record's fields



**Value**

The value contained within the field

## Database Command - ActiveConnection

**Command:**            **ActiveConnection**

**Description:**        ActiveConnection is used to establish a connection to a data source using either an ODBC setting or connection string.

**Example 1:** Using ODBC setting - must create an ODBC entry

```
Program
  strConnect = "ODBC;DATABASE=CALLMaster;UID=;PWD=;DSN=CALLMaster"
  Set varCommand = New Command
  varCommand.ActiveConnection strConnect
EndProgram
```

**Example 2:** Using a connection string - change your server and database name.

```
Program
  strConnect =
"Provider=SQLOLEDB;Server=<server>;DATABASE=CALLMasterSQL;UID=;PWD=;Integrated
Security=SSPI;"
  Set varCommand = New Command
  varCommand.ActiveConnection strConnect
EndProgram
```

## Database Command - AddNew

**Command:** AddNew

**Description:** Adds a new record to the recordset.  
**Note:** after the AddNew command, you must set the record fields and then execute an Update command to set field values.

### Example:

```
Program
    ' Change your server and database name in your connection string
    strConnect =
"Provider=SQLOLEDB;Server=<server>;DATABASE=CALLMasterSQL;UID=;PWD=;Integrated
Security=SSPI;"
    Set rst = New Recordset
    strSQL = "SELECT * FROM mailboxes"
    rst.Open strSQL strConnect
    Set varFields = rst.Fields
    Set varF3 = varFields.Field 3
    Set varF7 = varFields.Field 7
    ' this next block of codes adds a new record to the 'mailboxes' table and assigns it to Charles Jones
    rst.AddNew
    varLName = "Jones"
    varF3.SetValue varLName
    varFName = "Charles"
    varF7.SetValue varFName
    rst.Update
    ' you must use the update method after adding a record or the values are not placed in their respective
fields
    rst.close
    Set rst = Nothing
EndProgram
```

## Database Command - BOF | EOF

**Command:** BOF | EOF

**Description:** The BOF property of the Recordset object is a boolean value which is used to determine whether you are at the beginning of the recordset. EOF determines if you are at the end of the recordset. If both BOF and EOF are 0 then the recordset is NOTempty. If both BOF and EOF are 'True' (value not 0) then the recordset is empty.

### Example:

```
Program
    ' Change your server and database name in your connection string
    strConnect =
"Provider=SQLOLEDB;Server=<server>;DATABASE=CALLMasterSQL;UID=;PWD=;Integrated
Security=SSPI;"
    Set rst = New Recordset
    strSQL = "SELECT * FROM mailboxes"
    rst.Open strSQL strConnect
    count = 0
    While rst.EOF = 0
        ' once the last record is reached, the value of .EOF is non-zero and we exit the While...WEnd loop
        count = count + 1
        rst.MoveNext
    WEnd
    rst.close
    Print count
    rst.Close
    Set rst = Nothing
    ' This program would print the number of records to CMCalls <DATE>.TXT
EndProgram
```

## Database Command - Close

**Command:**        **Close**

**Description:**        Use this command when the recordset object is no longer needed. This insures that the table is properly saved.

### Example:

```
Program
' Change your server and database name in your connection string
strConnect =
"Provider=SQLOLEDB;Server=<server>;DATABASE=CALLMasterSQL;UID=;PWD=;Integrated
Security=SSPI;"
Set rst = New Recordset
strSQL = "SELECT * FROM mailboxes"
rst.Open strSQL strConnect
rst.Close
Set rst = Nothing
EndProgram
```

## Database Command - Count

Command:        **Count**

**Description:**        Count is a property of the 'Fields' object that designates how many fields there are in a given recordset. This property can be used in a For...Loop to access each field in a record. This example below would print out all of the field names in the recordset.

### Example:

Program

    ' Change your server and database name in your connection string

    strConnect =

    "Provider=SQLOLEDB;Server=<server>;DATABASE=CALLMasterSQL;UID=;PWD=;Integrated  
Security=SSPI;"

    Set rst = New Recordset

    strSQL = "SELECT \* FROM mailboxes"

    rst.Open strSQL strConnect

    Set varFields =rst.Fields

    ' iMax will now contain the number of fields (32) in the 'Fields' collection of the 'mailboxes' table

    iMax = varFields.**Count**

    For num = 1 To iMax Step 1

        set varField = varFields.Field num

        strFieldName = varField.Name

        PrintNL strFieldName

        Set varField = Nothing

    Next

    rst.close

    Set rst = Nothing

EndProgram

## Database Command - Delete

**Command:**            **Delete**

**Description:**        Use the following set of commands to execute commands such as Delete.

### Example:

Program

```
Set varCommand = New Command
strSQL = "DELETE * FROM RecordingInfo WHERE RIPresNo ="
strSQL = strSQL & PresNo
strSQL = strSQL & " AND RStatus ="
strSQL = strSQL & TStatus
strSQL = strSQL & "'"
Print " "
PrintNL strSQL
' Setup database connection and execute Delete Command
' Change your server and database name in your connection string
strConnect =
"Provider=SQLOLEDB;Server=<server>;DATABASE=CALLMasterSQL;UID=;PWD=;Integrated
Security=SSPI;"
varCommand.ActiveConnection strConnect
varCommand.Execute strSQL
PrintNL " Deleted"
' Clear the command
varCommand.CloseConnection
Set varCommand = Nothing
EndProgram
```

## Database Command - Execute

**Command:**        **Execute**

**Description:**        This is a member of the *Command* object. It is used when you want to execute an [SQL statement](#).

### Example 1:

```
Program
' Setup database connection and execute Select Command
' Change your server and database name in your connection string
strConnect =
"Provider=SQLOLEDB;Server=<server>;DATABASE=CALLMasterSQL;UID=;PWD=;Integrated
Security=SSPI;"
varCommand.ActiveConnection strConnect

strSQL = "select * from MailBoxes WHERE BoxNumber = '0000'"
varCommand.Execute strSQL
' clear the command
Set varCommand = Nothing
EndProgram
```

### Example 2:

```
Program
' Setup database connection and execute Delete Command
' Change your server and database name in your connection string
strConnect =
"Provider=SQLOLEDB;Server=<server>;DATABASE=CALLMasterSQL;UID=;PWD=;Integrated
Security=SSPI;"
varCommand.ActiveConnection strConnect

strSQL = "Delete * from MailBoxes WHERE BoxNumber = '0000'"
varCommand.Execute strSQL
' clear the command
Set varCommand = Nothing
EndProgram
```



## Database Command - Fields | Field

**Command:** Fields and Field

**Description:** The Fields property of a recordset is a collection object, which contains all of the fields in the recordset. Each field is accessed via a field object (a sub-object of 'Fields', which is a sub-object of 'Recordset').  
Field is a property of the 'Fields' collection object. When passed a parameter, it designates which field is being accessed.

### Example:

Program

```
Set rst = New Recordset  
strSQL = "SELECT * FROM mailboxes"
```

```
' Setup database connection. Change your server and database name in your connection string  
strConnect =
```

```
"Provider=SQLOLEDB;Server=<server>;DATABASE=CALLMasterSQL;UID=;PWD=;Integrated  
Security=SSPI;"
```

```
rst.Open strSQL strConnect
```

```
Set varFields = rst.Fields
```

```
' varF3 now becomes the 3rd member of the Fields collection
```

```
Set varF3 = varFields.Field 3
```

```
varName = varF3.Name
```

```
rst.close
```

```
Set rst = Nothing
```

```
EndProgram
```

## Database Command - MoveFirst

**Command:**        **MoveFirst**

**Description:**        MoveFirst is a member function of the Recordset object. It is used to navigate to the first Record of a recordset object. Using the MovePrevious function after MoveFirst causes the *recordsetObject*.BOF property to change from 0 to -1.

### Example:

Program

```
' Setup database connection. Change your server and database name in your connection string
strConnect =
"Provider=SQLOLEDB;Server=<server>;DATABASE=CALLMasterSQL;UID=;PWD=;Integrated
Security=SSPI;"
```

```
Set rst = New Recordset
```

```
strSQL = "SELECT * FROM mailboxes"
```

```
rst.Open strSQL strConnect
```

```
rst.MoveNext
```

```
rst.MoveNext
```

```
' after moving to the third record we can go back to the beginning of the table
```

```
rst.MoveFirst
```

```
rst.close
```

```
Set rst = Nothing
```

```
EndProgram
```

## Database Command - MoveLast

**Command:**            **MoveLast**

**Description:**        MoveLast is a member function of the Recordset object. It is used to navigate to the last Record of a recordset object. Using the MoveNext function after MoveLast causes the *recordsetObject.EOF* property to change from 0 to -1.

### Example:

```
Program
    ' Setup database connection. Change your server and database name in your connection string
    strConnect =
"Provider=SQLOLEDB;Server=<server>;DATABASE=CALLMasterSQL;UID=;PWD=;Integrated
Security=SSPI;"

    Set rst = New Recordset
    strSQL = "SELECT * FROM mailboxes"
    rst.Open strSQL strConnect
    ' we move to the end of the recordset
    rst.MoveLast
    rst.close
    Set rst = Nothing
EndProgram
```

## Database Command - MoveNext

**Command:**            **MoveNext**

**Description:**        MoveNext is a member function of the Recordset object. It is used to forward navigate through the database one Record at a time. When you're on the last record, this command causes the *recordsetObject.EOF* property to change from 0 to -1.

### Example:

Program

    ' Setup database connection. Change your server and database name in your connection string

    strConnect =

    "Provider=SQLOLEDB;Server=<server>;DATABASE=CALLMasterSQL;UID=;PWD=;Integrated  
Security=SSPI;"

    Set rst = New Recordset

    strSQL = "SELECT \* FROM mailboxes"

    rst.Open strSQL strConnect

    rst.MoveNext

    rst.MoveNext

    ' at this point we are at the third record

    rst.close

    Set rst = Nothing

EndProgram

## Database Command - MovePrevious

**Command:** MovePrevious

**Description:** MovePrevious is a member function of the Recordset object. It is used to backward navigate through the database one Record at a time. When you're on the first record, this command causes the *recordsetObject*.BOF property to change from 0 to -1.

### Example:

Program

' Setup database connection. Change your server and database name in your connection string

strConnect =

"Provider=SQLOLEDB;Server=<server>;DATABASE=CALLMasterSQL;UID=;PWD=;Integrated  
Security=SSPI;"

Set rst = New Recordset

strSQL = "SELECT \* FROM mailboxes"

rst.Open strSQL strConnect

rst.MoveNext

rst.MoveNext

' after moving to the third record, we can move back to the second record

rst.MovePrevious

rst.close

Set rst = Nothing

EndProgram

## Database Command - Name

**Command:**        **Name**

**Description:**        Name is a property of the 'Field' object that returns the name of the field.

### Example:

Program

```
' Setup database connection. Change your server and database name in your connection string
strConnect =
"Provider=SQLOLEDB;Server=<server>;DATABASE=CALLMasterSQL;UID=;PWD=;Integrated
Security=SSPI;"
```

```
Set rst = New Recordset
strSQL = "SELECT * FROM mailboxes"
rst.Open strSQL strConnect
Set varFields =rst.Fields
Set varF8 = varFields.Field 8
' places the name of the field ('status') into the variable varName
varName = varF1.Name
rst.close
Set rst = Nothing
EndProgram
```

## Database Command - Connection String

**Command:**            **Connection String**

**Description:**        The Connection String is a string passed to the .Open or the .ActiveConnection methods of the recordset object (rst). This string is required to communicate to CALLBasic which database to search for the various tables.  
The connection can be via an ODBC entry or Provider.

You can also define a connection string to your database for use with the database commands.

“ODBC;DATABASE=CALLMaster;UID=;PWD=;DSN=CALLMaster”

' Setup database connection. Change your server and database name in your connection string  
strConnect = “Provider=SQLOLEDB;Server=<server>;DATABASE=CALLMasterSQL;UID=;PWD=;Integrated  
Security=SSPI;”

See [Open](#) and [ActiveConnection](#) database commands for details

## Database Command - Open

Command:      **Open**

**Description:**      This method of the *rst* recordset opens the recordset for use in our program. It is passed two parameters, which can be either string or variable to represent an [SQL String](#) and an [ODBC or Connection](#) String. Using the example below we have now opened a recordset referenced by the variable *rst*, for use in our program.

### Example 1:- using Connection string

Program

    ' Setup database connection. Change your server and database name in your connection string

    strConnect =

    "Provider=SQLOLEDB;Server=<server>;DATABASE=CALLMasterSQL;UID=;PWD=;Integrated  
    Security=SSPI;"

    Set rst = New Recordset

    strSQL = "SELECT \* FROM Mailboxes"

    rst.Open strSQL strConnection

    rst.close

    Set rst = Nothing

EndProgram

### Example 2:- using ODBC string

Program

    ' Setup database connection. Change your server and database name in your connection string

    strConnection = "ODBC;DATABASE=CALLMaster;PWD=;DSN=CALLMaster"

    Set rst = New Recordset

    strSQL = "SELECT \* FROM Mailboxes"

    rst.Open strSQL strConnection

    rst.close

    Set rst = Nothing

EndProgram



## Database Command - Set

Command:       **Set**

**Description:**       This command is used to create an object. In the case of a recordset object, it is used in conjunction with a connection string used to reference a database. The New keyword allocates memory for the Recordset object, and the memory is returned to the heap with the Nothing keyword. (i.e. – Set *varObject* = Nothing)

### Example:

```
Program
    Set rst = New Recordset
    strSQL = "SELECT * FROM mailboxes"
    rst.Open strSQL
    "Provider=SQLOLEDB;Server=<server>;DATABASE=CALLMasterSQL;UID=;PWD=;Integrated
Security=SSPI;"
    Set varFields =rst.Fields
    Set varF8 = varFields.Field 8
    varType = varF8.Type
    rst.close
    Set rst = Nothing
EndProgram
```

## Database Command - SetValue

Command:       **SetValue**

**Description:**       This is a function of the .Fields object, which is a member of the *recordset* object. It is used to change the value of the field referenced by the *set* command (i.e. - set varField = varFields.Field 6). The field will be changed in whatever Record is pointed to by the database cursor. SetValue is a method of the 'Field' object. It is used to enter a new value into the field. Note: the SetValue is not effective until we use the .Update function of the *rst* recordset object.

### Example:

Program

```
' Setup database connection. Change your server and database name in your connection string
strConnect =
"Provider=SQLOLEDB;Server=<server>;DATABASE=CALLMasterSQL;UID=;PWD=;Integrated
Security=SSPI;"
```

```
Set rst = New Recordset
strSQL = "SELECT * FROM mailboxes"
rst.Open strSQL strConnect
Set varFields = rst.Fields
Set varF8 = varFields.Field 8
' sets the 'status' field of the 'mailboxes' table to 'user' (4)
varF8.SetValue 4
rst.Update
rst.Close
Set rst = Nothing
EndProgram
```

## Database Command - SQL String with SELECT

**Command:** SQL String with SELECT

**Description:** The SQL String is a string passed to the .Open method of the recordset object (rst). This string defines the recordset, i.e. - which table of the database we will be using, which fields we will be using from the table, etc. SELECT designates which fields we want included in the recordset. FROM designates which table we will use. We can further define the recordset by using the WHERE keyword.

SELECT <\* for all fields or select fields separated by commas> FROM TableA WHERE KeyField <expression value1>

### Example1:

Program

'This example will select records from the MailBoxes table whose BoxNumber is equal to the caller's input.

'Single Quotes are required around Code in the Select statement as BoxNumber is defined as a text data type in the Mailboxes table.

' Setup database connection. Change your server and database name in your connection string

strConnect =

"Provider=SQLOLEDB;Server=<server>;DATABASE=CALLMasterSQL;UID=;PWD=;Integrated Security=SSPI;"

Code = \$tt

Set CMvarSet = New Recordset

strSQL = "**SELECT** \* FROM MailBoxes WHERE BoxNumber ="

strSQL = strSQL & Code

strSQL = strSQL & ""

' The SELECT is normally followed by an Open command to open the database and find record

CMvarSet.Open strSQL strConnect

'Add other processing statements to get the data then close the recordset and database.

EndProgram

### Example 2:

Program

'This example will select records from the MailBoxes table whose BoxStatus is equal to the caller's input.

'Single Quotes are NOT required around BoxStat in the Select statement as BoxStatus is defined as a number data type in the Mailboxes table.

' Setup database connection. Change your server and database name in your connection string

strConnect =

"Provider=SQLOLEDB;Server=<server>;DATABASE=CALLMasterSQL;UID=;PWD=;Integrated Security=SSPI;"

BoxStat = \$tt

Set CMvarSet = New Recordset

strSQL = "**SELECT** \* FROM MailBoxes WHERE BoxStatus ="

strSQL = strSQL & BoxStat

strSQL = strSQL & ""

' The SELECT is normally followed by an Open command to open the database and find record  
CMvarSet.Open strSQL strConnect  
'Add other processing statements to get the data then close the recordset and database.

EndProgram

## Database Command - Type

**Command:**            **Type**

**Description:**        Type is a property of the 'Field' object that returns the field type.

### Example:

Program

```
' Setup database connection. Change your server and database name in your connection string
strConnect =
"Provider=SQLOLEDB;Server=<server>;DATABASE=CALLMasterSQL;UID=;PWD=;Integrated
Security=SSPI;"
```

```
Set rst = New Recordset
```

```
strSQL = "SELECT * FROM mailboxes"
rst.Open strSQL strConnect
```

```
Set varFields =rst.Fields
```

```
Set varF8 = varFields.Field 8
```

```
' places the type of the field ('integer') into the variable varType
```

```
varType = varF8.Type
```

```
rst.close
```

```
Set rst = Nothing
```

```
EndProgram
```

## Database Command - Update

**Command:**            **Update**

**Description:**        When using the .SetValue or the .AddNew member function, changes aren't implemented until the .Update function is used.

### Example:

Program

    ' Setup database connection. Change your server and database name in your connection string

    strConnect =

    "Provider=SQLOLEDB;Server=<server>;DATABASE=CALLMasterSQL;UID=;PWD=;Integrated  
    Security=SSPI;"

    Set rst = New Recordset

    strSQL = "SELECT \* FROM mailboxes"

    rst.Open strSQL strConnect

    Set varFields =rst.Fields

    Set varF3 = varFields.Field 3

    Set varF7 = varFields.Field 7

    rst.AddNew

    varLName = "Jones"

    varF3.SetValue varLName

    varFName = "Charles"

    varF7.SetValue varFName

    rst.**Update**

    'you must use the update method after adding a record or changing the values of a field or the values  
are not placed in their respective fields

    rst.close

    Set rst = Nothing

EndProgram

## Database Command - Value

**Command:**            **Value**

**Description:**        Value is a property of the 'Field' object that returns the field value.

### Example:

Program

```
' Setup database connection. Change your server and database name in your connection string
strConnect =
"Provider=SQLOLEDB;Server=<server>;DATABASE=CALLMasterSQL;UID=;PWD=;Integrated
Security=SSPI;"
```

```
Set rst = New Recordset
strSQL = "SELECT * FROM mailboxes"
rst.Open strSQL strConnect
```

```
Set varFields =rst.Fields
Set varF3 = varFields.Field 3
' places the value of the'LastName' field (i.e. 'Jones') into the variable varLastName
varLastName = varF3.Value
rst.close
Set rst = Nothing
EndProgram
```

## Example Of Using Database Commands

### Example Program

Below is an example which changes the 'Status' field of all mailboxes in the 'Mailboxes' table to 'User'.

#### Example:

Program

' Setup database connection. Change your server and database name in your connection string  
strConnect =

"Provider=SQLOLEDB;Server=<server>;DATABASE=CALLMasterSQL;UID=;PWD=;Integrated  
Security=SSPI;"

' First we declare a recordset object.

' Once this is done, all of the methods and properties of the object are available to us.

Set rst = New Recordset

' Next we open the object for use

strSQL = "SELECT \* FROM Mailboxes"

rst.Open, strSQL strConnect

' Now we can traverse through the recordset and make whatever changes that are needed

Set varFields = rst.Fields

rst.MoveFirst

While rst.EOF = 0

Set varField = varFields.Field 8

intValue = 4

varField.SetValue = strValue

rst.Update

Set varField = Nothing

rst.MoveNext

WEnd

' Before closing the recordset, we need to release the resources that rst and varFields have been using

Set varFields = Nothing

Set rst = Nothing

rst.Close

EndProgram



# SAMPLE APPLICATIONS AND SCRIPTS

Included with CALLMaster software are sample applications and scripts. These sample applications and scripts are designed only as a template to assist you in customizing CALLMaster to meet your business needs. Use them to guide you in your application development, adding, modifying and deleting parts as dictated by your custom application requirements.

[Sample Applications](#)

[Sample Scripts](#)

## Sample Applications & Scripts - Sample Applications

Sample applications are designed and provided only as a template to help you in customizing CALLMaster to meet your business needs. Use them to guide you in your application development, add, modify and delete any component of the template as required by your custom application.

Learn how to develop the application by referring to the [Walkthrough with Sample Applications](#) for online instructions and the ability to download the application if desired.

You can [download Sample Applications](#) from the SpeechSoft website.

## Sample Applications & Scripts - Sample Scripts

Scripts written in CALLBasic are used in conjunction with Audiotex modules to customize real time information processing telephony applications.

Sample scripts are provided for each of the CALLBasic commands. These sample scripts are designed only as a template to assist you in customizing CALLMaster to meet your business needs. Use them to guide you in your application development, adding, modifying and deleting parts as dictated by your custom application requirements.

Sample scripts are available in the \Program Files\CALLMaster\Scripts\Samples folder once you have installed CALLMaster.

[Sample Scripts](#) can also be downloaded from the SpeechSoft website.

See [Walkthroughs](#) for details on how to integrate CALLBasic scripts in developing your application.

## Walkthroughs

Walkthroughs are provided only as a template to facilitate your understanding of CALLMaster and assist in your custom application development.

Use the walkthroughs as a learning guide to the various CALLMaster features. Add your own data, modify any of the steps as desired to customize the application to your needs.

Make sure that you start the walkthroughs using a clean install of CALLMaster. Make sure that CALLMaster has installed correctly and the installation has been validated. See [Validating Installation](#) for details.

- [Walkthrough: Change Application Without Restarting CALLMaster Service](#)
- [Walkthrough: Change Parameter and Apply Change Without Restarting CALLMaster Service](#)
- [Walkthrough: Get Fixed Length Input Using GetDTMF and Read Corresponding Record](#)
- [Walkthrough: Get Variable Length Input Using GetDTMF and Play Back Input Entered](#)
- [Walkthrough: Managing Date/Time Data Type](#)
- [Walkthrough: Play Audiotex Message Based on a System Variable and Get Caller Input Using Touch Tone Table](#)
- [Walkthrough: Send Caller to a Mailbox Based on the Value of a System Variable](#)
- [Walkthrough: Set-up Attendant with Transfer, Fax Receive and Audiotex Processing](#)
- [Walkthrough: Use Global Variable with Entry.bas Script to Play Rotating Advertisement](#)
- [Walkthrough: Get Single Digit Input Using Touch Tone Index Table and Play Corresponding Message](#)

## Walkthrough: Change Application Without Restarting CALLMaster Service

CALLMaster application (e.g. Voice Mail, Audiotex) can be update via either the CALLMaster Manager or the Main Window (non T1 systems only). If you want the change to take effect without having to stop and restart the CALLMaster service, then use the CALLMaster Main Window to update the application.





During this walkthrough, you will learn to change the Application using the Main Window. The Application change can be applied to an individual line(s) or to all lines.

### Prerequisites and Assumptions

- CALLMaster software installed and running with the Dialogic board installed and configured. For more information, see [Installation and Setup](#).
- A working knowledge of [CALLMaster Main Window](#).

### Update Line Application via the CALLMaster Main Window.

Open the **CALLMaster Main Window**.

1. First stop the line(s) that need to be changed.
  - If you want to change the application for all lines then click on the large red button  to stop all lines.
  - If you want to change the application for a line, then select the line and click on the small red button  to stop only the selected line.
2. Next highlight the port/line that you want to change then select the Port tab on the top and from the drop down list, select Change Application.
  - **Change Application** - select application as desired, click **OK**.
3. Restart the line(s).
  - If you stopped all lines then click on the large green button  to start all lines.
  - If you stopped select lines, then highlight each stopped line and click on the small green button  button to start the line.

Each line will need to be changed individually.

The change will be applied to the next call made to the line.

TIP: Use this method to set a line to Voice Mail to make recordings without have to stop the entire CALLMaster service.

## Walkthrough: Change Parameter and Apply Change Without Restarting CALLMaster Service

CALLMaster parameters can be update via either the Manager or the Main Window. If you want the change to take effect without having to stop and restart the CALLMaster service, then use the CALLMaster Main Window to update the Parameters.

During this walkthrough, you will learn to change the log level in the Logging Parameter using the Main Window.

### Prerequisites and Assumptions

- CALLMaster software installed and running with the Dialogic board installed and configured. For more information, see [Installation and Setup](#).
- A working knowledge of [CALLMaster Main Window](#).

### Update Logging Parameter via the CALLMaster Main Window.

Open the **CALLMaster Main Window**. Click on **Parameters**, then select **Logging**. Double click on the parameter below to change their values.

- **Trace Level** - Change the value to 1, click **OK**.

Close the Logging Parameters window making sure to save the changes.

The change will be applied to the next call.

### Test that the Parameter change has been applied

Depending on the parameter, you may be able to view the change in the Log file. In this example, you should be able to see additional call details printed at the end of the log file.

## Walkthrough: Get Fixed Length Input Using GetDTMF and Read Corresponding Record

In this walkthrough we will get a 4 digit fixed length input from the caller, e.g. Mailbox number. We will then read the CALLMaster Mailbox table for a record with a matching box number. If we find the record, the caller will be played back the new message count. If we do not find the record, the caller will be played an error message and given the option to enter another number.

### Walkthrough Overview

In this walkthrough, you will:

1. Create Audiotex Records/Modules and Associated Scripts;
2. Define entry in the IVR Application List for the new custom IVR;
3. Set Ports.

During this walkthrough, you will learn to do the following:

- Configure CALLMaster to answer in Audiotex mode;
- Get input using the GetDTMF command;
- Use CALLBasic commands StringLength;
- Use CALLBasic database commands to Read a record;
- Send caller to another Audiotex Module using the Next Module Audiotex Module Action;
- Make Audiotex recordings;
- Handle caller hang up and call failure.

### Prerequisites and Assumptions

- CALLMaster software installed and running with the Dialogic board installed and configured. For more information, see [Installation and Setup](#).
- A working knowledge of [CALLMaster Manager](#). Specifically how to add, delete and edit records.
- Phone lines enabled and connected to CALLMaster.

### Create Audiotex Records/Modules and Associated Scripts

As all calls will be starting in Audiotex mode in this walkthrough, the Audiotex Modules will determine the call flow.

In the Audiotex set-up, the caller will be requested to enter a 4 digit mailbox number. If a matching Mailbox record is found then the number of new messages will be played back to the caller. If a matching mailbox record is not found, the caller will be given the option to enter another mailbox number.

We will need to create several Audiotex records as shown below. The Label field provides a brief description of each module.

Audiotex Records																	
Mod ule ID	Label	Wri te Lab el to Log File	Mod ule Acti on	Next Mod ule ID	Fail Mod ule ID	Respo nse Length	Script File to Run (Must include full path name)	Touch Tone Table									
								1	2	3	4	5	6	7	8	9	0
1000	Start call and Initialize values	Yes	Next Modu le	1100	9900	0000	InitializeVal.bas										

1100	Get fixed length input from caller	Yes	Next Module	1200	9900	0	GetDTMF-FixLe n.bas											
1190	Invalid input.	Yes	Next Module	1191	9900	0	CheckTryCount. bas											
1191	Try again.	Yes	Next Module	1100	9900	0												
1200	Read Mailbox table	Yes	Next Module	9999	9900	0	ReadMailboxTa ble.bas											
1210	Say new message count if mailbox record found	Yes	Next Module	9999	9900	0	SayNewMsgCo unt.bas											
1220	Record not found. Press 1 to try again or hang-up to exit.	Yes	Touch Tone Table	9999	9900	1		1100	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999
9900	Call Failure Detected	Yes	Next Module	9999	9999	0	SetCallFailure.b as											
9910	Caller Hang-up Detected	Yes	Next Module	9999	9999	0	SetCallerHangu p.bas											
9920	Maximum tries attempted	Yes	Next Module	9999	9999	0	SetMaxAttempt s.bas											
9999	Last Module - End call	Yes	Last Module															



## Audiotex Module Description

- Module 1000 is the starting module. We will use this module to initialize variables using script InitializeVal.bas. In particular we want to fix the allowed Input Length (intMBInputLen) to be the same as the Mailbox length. We also want to define a variable (strCMConnection) for the connection string required to read the CALLMasterSQL database. You will need to change the connection string for your environment settings. We will set the system variable lctrap to an Audiotex module to send caller if they hang-up. See sample code below.
- Module 1100 is used to get the caller's input using the GetDTMF command. This script uses the StringLength command to make sure the caller has entered the number of digits defined by variable intMBInputLen. If they have not then the call will be forwarded to module 1190 for error processing.
- Module 1190 is to notify the caller that they have not entered the required number of digits. The associated script, CheckTryCount.bas, gives the caller 3 attempts for a valid input. If the caller has not entered a valid input after 3 attempts, processing will be forwarded to module 9920.
- Module 1200 reads the CALLMaster Mailbox table for record whose BoxNumber is equal to the number entered (strMBNoInput). If a matching record is found then we save the value of the NewMessageCount field and forward caller to module 1210; if not found then caller is forwarded to module 1220.
- Module 1210 will use the SayNumeral command to play the value in this field.
- Module 1220 is used to tell caller that a matching mailbox record was not found. It will also give the caller the option to enter another number.
- Module 9920 is used to notify caller that they have reached the maximum attempts allowed for a valid input. The script associated with this module can be used to set flags and route processing as required.
- Modules 9900 and 9910 are required to handle call failure and capture caller hang-up respectively. Both these modules can be used to set flags and route processing as required.

Based on the Audiotex modules shown above, we will need 6 scripts.

Let's create the scripts first. Open Notepad and create scripts following code examples below. Remember to save them as .bas type files in the Program Files\CALLMaster\Scripts folder.

### CALLBasic Script - InitializeVal.bas

[Copy to Clipboard](#)

#### Program

```
*****
' Script: InitializeVal.bas
' Function: This script is used to initialize values
*****

strYes = "Yes"
strNo = "No"
intYes = 1
intNo = 0
strCallFailure = strNo
strCallerHangup = strNo
strMaxAttempts = strNo
intOne = 1

' Set retry counter and maximum attempts allowed
intTryCount = 1
intMaxAttemptsAllowed = 3

' Set intMaxInputLen to include maximum number allowed plus the termination digit
intMBInputLen = 4

' Mailbox record values
MBMatchExists = intNo
MBDBOpen = intNo
```

MB-intNewMsgCount = 0

' Set connection string to read CALLMasterSQL database

strCMConnection = "Provider=SQLOLEDB;Server=sscmsql;DATABASE=CALLMasterSQL;UID=;PWD=;Integrated Security=SSPI;"

' Set the module to return to if caller hangs up

lctrp = 9910

' Convert system variable \$channel from integer to string to be used in concatenated print statement.

IntegerToString strchannel \$channel

pline = "\*\*\*\*\* For channel=" & strchannel

pline = pline & " Started initialization process "

PrintNL pline

EndProgram

CAL  
LBa  
sic  
Scri  
pt -  
Get  
DTM  
F-Fi  
xLe  
n.ba  
s

[Copy  
to  
Clipb  
oard](#)

Program

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

\*

'

Script:  
GetDTMF-  
FixLen.ba  
s  
'

Function:  
This script  
gets a  
fixed  
length  
input from  
the  
caller.

```

*****
*****
*****
*****
*****
*****
*****
*****
*****
*
    ' Set
up
parameter
s for
GetDTMF
command

strNoInput
= ""

strMBNoIn
put = ""
    tmask
= ""
    ndigits
=
intMBInput
Len
    timeout
= 15

GetDTMF
strNoInput
tmask
ndigits
timeout

    ' Get
length of
input

StringLeng
th intlen
strNoInput

    pline =
***** For
channel="
&
strchannel
    pline =
pline & "
strNoInput
entered is
"
    pline =
pline &

```

```

strNoInput
PrintNL
pline

' Check
caller
input. The
caller
must enter
the
number of
digits
defined by
variable
intMBInput
Len.

' If valid
input then
save
number
entered

If intlen
<>
intMBInput
Len Then
pline
= "*****"
For
channel="
&
strchannel
pline
= pline & "
Invalid
strNoInput
entered.
Return to
1190 for
error and
retry"

PrintNL
pline

Return
1190
Else

intTryCoun
t = 0

strMBNoIn
put =
strNoInput
EndIf

EndProgra
m

```

CAL LBa sic Scri pt - Che ckTr yCo unt. bas	<a href="#">Copy to Clipb oard</a>
--	--

```
Program
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
****
'
Script:
CheckTry
Count.bas
'
Function:
This script
checks
the
number of
invalid
input
attempts.
*****
*****
*****
*****
*****
*****
*****
*****
*****
****
' If the
caller has
reached
the
maximum
attempts
allowed
then the
call is
routed to
module
```

```

9920.
    ' If not
then they
are
returned
back to
module
1000 to
retry input
based on
Next
Module ID.
    If
intTryCoun
t >
intMaxAtte
mptsAllow
ed Then

Return
9920
    Else

intTryCoun
t =
intTryCoun
t + 1
    EndIf
    ,

Convert
system
variable
intTryCoun
t from
integer to
string to
be used in
concatena
ted print
statement.

IntegerToS
tring
strTryCoun
t
intTryCoun
t

    pline =
"***** For
channel="
&
strchannel
    pline =
pline & "
TryCount
= "
    pline =

```

pline &  
strTryCoun  
t  
PrintNL  
pline  
  
EndProgra  
m

CAL LBa sic Scri pt - Rea dMa ilbo xTa ble. bas	<a href="#">Copy to Clipb oard</a>
---	--

Program  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
'  
Script:  
ReadMailb  
oxTable.ba  
s  
'  
Function:  
This script  
reads the  
CALLMast  
er Mailbox  
table.  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
'  
Create a

```

new
recordset
    Set
    MBvarSet
    = New
    Recordset

    strSQL
    =
    "SELECT
    NewMess
    ageCount
    FROM
    Mailboxes
    WHERE
    BoxNumb
    er ="
    strSQL
    = strSQL
    &
    strMBNoIn
    put
    strSQL
    = strSQL
    & ""

    pline =
    "For
    channel "
    &
    strchannel
    pline =
    pline & "
    strSQL = "
    & strSQL
    PrintNL
    pline

    ' Open
    MBDB
    database
    and find
    record

    MBvarSet.
    Open
    strSQL
    strCMCon
    nection

    Set
    MBvarFiel
    ds =
    MBvarSet.
    Fields

    MBDBOpe
    n = intYes

```



```

' If the
record
exists
then move
selected
field values
for
processing
If
MBvarSet.
EOF = 0
Then
    Set
    MBvarF1 =
    MBvarFiel
ds.Field 1

    MB-intNew
    MsgCount
    =
    MBvarF1.
    Value

    IntegerToS
    tring
    strNewMs
    gCount
    MB-intNew
    MsgCount

' Free
field
variables
    Set
    MBvarF1 =
    Nothing

' Set
Switches

MBMatch
Exists =
intYes
Else

MBMatch
Exists =
intNo
EndIf

' Free
MBDB
varSet,
varFields,
and close
database
'
Destroy

```

```

Fields
Object
Set
MBvarFiel
ds =
Nothing

' Close
recordset

MBvarSet.
Close

' Free
memory
Set
MBvarSet
= Nothing

,

MBDB
database
closed

MBDBOpe
n = intNo

' If
record
found then
return
1210 else
return
1220
If
MBMatch
Exists =
intYes
Then
'Print
information
to log file
for
reference
and
debugging
pline
= "*****
For
channel="
&
strchannel
pline
= pline & "
Mailbox
record
found for
Mailbox =
" &

```

```

strMBNoIn
put
    pline
= pline & "
with new
message
count = "
&
strNewMs
gCount

PrintNL
pline

Return
1210
    Else
        pline
= "*****
For
channel="
&
strchannel
    pline
= pline & "
Mailbox
record
NOT found
for Mailbox
= " &
strMBNoIn
put

PrintNL
pline

Return
1220
    EndIf

EndProgra
m

```

<b>CAL LBa sic Scri pt - Say New Msg Cou nt.b as</b>	<a href="#">Copy to Clipb oard</a>
--	--

Program  
 \*\*\*\*\*

```

*****
*****
*****
*****
*****
*****
*****
*****
****
'
Script:
SayNumb
erInput.ba
s
'
Function:
This script
says the
Number
Input.
!*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
****

SayNumer
al
strNewMs
gCount

'Print
information
to log file
for
reference
and
debugging
pline =
"***** For
channel="
&
strchannel
pline =
pline &
"***** New
message
count
played
back to
caller is
*****"

pline =

```

pline &  
strNewMs  
gCount  
PrintNL  
pline

EndProgra  
m

CAL  
LBa  
sic  
Scri  
pt -  
SetC  
allF  
ailur  
e.ba  
s

[Copy  
to  
Clipb  
oard](#)

Program

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

'  
Script:  
SetCallFail  
ure.bas  
'

Function:  
This script  
sets the  
strCallFail  
ure  
variable to  
yes and  
prints  
information  
in the call  
log file.

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

```

strCallFailure = strYes

'Print information to log file for reference and debugging
pline = "***** For channel=" & strchannel
pline = pline & "***** Call Failure.
*****"

PrintNL pline

EndProgram

```

<b>CAL LBa sic Scri pt - SetC aller Han gup. bas</b>	<a href="#">Copy to Clipboard</a>
--	-----------------------------------

```

Program
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
'
Script:
SetCallerHangup.bas
'
Function:

```

This script  
sets the  
strCallFail  
ure  
variable to  
yes and  
prints  
information  
in the call  
log file.

```
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****
```

```
strCallerH  
angup =  
strYes
```

```
'Print  
information  
to log file  
for  
reference  
and  
debugging  
pline =  
"***** For  
channel=" &  
strchannel  
pline =  
pline &  
"*****
```

```
Caller  
Hangup.  
*****"
```

```
PrintNL  
pline
```

```
EndProgra  
m
```

**CAL  
LBa  
sic  
Scri  
pt -  
Set  
Max**

[Copy  
to  
Clipb  
oard](#)

[illegible]



```

strchannel
pline =
pline &
*****

Maximum
Attempts
reached to
enter valid
input
*****

PrintNL
pline

EndProgra
m

```

Now you can create the Audiotex modules and reference the scripts above. Open the **CALLMaster Manager**, click on **Manage**, then select **Audiotex**. Let's add the Audiotex records shown above.

### Define IVR Application List Entry

We need to define the custom IVR in the IVR Application List and set it to Audiotex Start Module 1000. Open CALLMaster Manager and select Manage | IVR Application List. Let's add a new record as follows:

IVR Application List	
Audiotex Start Module	Application Name
1000	Fixed Length Input Walkthrough

### Set Ports

As a last step, we need set the port(s) to the new IVR Application entry created above. Make sure the Audiotex Start Module field (read only) correctly points to module 1000.

Set the Port Greeting File IDs to the main greeting for all calls if you have recorded one. If you do not want a main greeting then set the Greeting File IDs to zero. Leave all other fields to the default values.

If you are working with a demo license or have only one line connected, then make sure the connected port is set as specified below. All other ports should be set to Application 'Disabled'.

Port Records			
Port #	Application	Audiotex Start Module	Port Greeting File IDs (Morning, Afternoon, Evening, Closed, Holidays)
1	Fixed Length Input Walkthrough	1000	0, 0, 0, 0, 0
2	Fixed Length Input Walkthrough	1000	0, 0, 0, 0, 0

3	Fixed Length Input Walkthrough	1000	0, 0, 0, 0, 0
4	Fixed Length Input Walkthrough	1000	0, 0, 0, 0, 0

### Test the custom application

Make sure you have set the CALLMasterSQL database connection string variable strCMConnection in the InitializeVal.bas script for your environment.

Save all your work.

Stop CALLMaster service.

Make Administrator Audiotex Recordings:

- Set Port 1 to Application Voice Mail.
- Start CALLMaster service.
- Make a call using Port 1.
- [Log in](#) as the Administrator using the Voice Mail Main Menu and make recordings for the Audiotex modules below.

Audiotex Module ID	Proposed Recording
1100	Please enter you 4 digit mailbox number.
1190	Invalid option.
1191	Please try again
1210	The new message count for your mailbox is...
1220	Mailbox not found. Press 1 to enter a new number or hang-up to end the call.
9900	We are unable to process your call. Please contact the system administrator for further assistance.
9920	You have reached the maximum attempts allowed for a valid input. Please contact the system administrator for further assistance.

Reset Port 1 to Application Audiotex. Set the Port Greeting File IDs to zeros.

Stop and Start CALLMaster service.

Make a phone call to test the application.

Tip: See [Walkthrough: Change Application Without Restarting CALLMaster Service](#) to set port to Voice Mail to make recording without stopping and restarting the CALLMaster service.

## Walkthrough: Get Variable Length Input Using GetDTMF and Play Back Input Entered

In this walkthrough we will get a variable length input from the caller and play back input entered.

### Walkthrough Overview

In this walkthrough, you will:

1. Create Audiotex Records/Modules and Associated Scripts;
2. Define entry in the IVR Application List for the new custom IVR;
3. Set Ports.

During this walkthrough, you will learn to do the following:

- Configure CALLMaster to answer in Audiotex mode;
- Get input using the GetDTMF command;
- Use CALLBasic commands StringLength, MemMove, SayNumeral;
- Send caller to another Audiotex Module using the Next Module Audiotex Module Action;
- Make Audiotex recordings;
- Handle caller hang up and call failure.

### Prerequisites and Assumptions

- CALLMaster software installed and running with the Dialogic board installed and configured. For more information, see [Installation and Setup](#).
- A working knowledge of [CALLMaster Manager](#). Specifically how to add, delete and edit records.
- Phone lines enabled and connected to CALLMaster.

### Create Audiotex Records/Modules and Associated Scripts

As all calls will be starting in Audiotex mode in this walkthrough, the Audiotex Modules will determine the call flow.

In the Audiotex set-up, the caller will be asked to input a variable length number, which will be played back to them.

We will need to create several Audiotex records as shown below. The Label field provides a brief description of each module.

Audiotex Records							
Module ID	Label	Write Label to Log File	Module Action	Next Module ID	Fail Module ID	Response Length	Script File to Run (Must include full path name)
1000	Start call and Initialize values	Yes	Next Module	1100	9900	0000	InitializeVal.bas
1100	Get variable length input from caller	Yes	Next Module	1200	9900	0	GetDTMF-VarLen.bas
1190	Invalid input.	Yes	Next Module	1191	9900	0	CheckTryCount.bas
1191	Try again.	Yes	Next Module	1100	9900	0	

1200	Say input entered	Yes	Next Module	9999	9900	0	SayNumberInput.bas
9900	Call Failure Detected	Yes	Next Module	9999	9999	0	SetCallFailure.bas
9910	Caller Hang-up Detected	Yes	Next Module	9999	9999	0	SetCallerHangup.bas
9920	Maximum tries attempted	Yes	Next Module	9999	9999	0	SetMaxAttempts.bas
9999	Last Module - End call	Yes	Last Module				

#### Audiotex Module Description

- Module 1000 is the starting module. We will use this module to initialize variables using script InitializeVal.bas. In particular we want to set the Maximum Input Length (intMaxInputLen). As we are allowing a variable length input, we will need the caller to enter a terminating digit such as \*. The intMaxInputLen must include the terminating digit. e.g. if we want caller to enter a number up to 5 digits then intMaxInputLen must be set to 6. See sample code below.
- Module 1100 is used to get the caller's input using the GetDTMF command. As the input is variable length with a maximum length defined by intMaxInputLen, we need to make sure to define a terminating digit, an asterisk (\*). This script uses the StringLength command to make sure the caller has entered a number. If they have not then the call will be forwarded to module 1190 for error processing.
- Module 1190 is to notify the caller that they have not entered any numbers. The associated script, CheckTryCount.bas, gives the caller 3 attempts to make an input. If the caller has not entered any keys after 3 attempts, processing will be forwarded to module 9920.
- Before we say the number entered back to the caller, we need to remove the last digit. The script associated with module 1200 will make a new string variable without the last digit entered using the MemMove command. This new variable will be used to say the number back to the caller using the SayNumeral command.
- Module 9920 is used to notify caller that they have reached the maximum attempts allowed for a valid input. The script associated with this module can be used to set flags and route processing as required.
- Modules 9900 and 9910 are required to handle call failure and capture caller hang-up respectively. Both these modules can be used to set flags and route processing as required.

Based on the Audiotex modules shown above, we will need several scripts.

Let's create the scripts first. Open Notepad and create scripts following code examples below. Remember to save them as .bas type files in the Program Files\CALLMaster\Scripts folder.

<b>CALLBasic Script - InitializeVal.bas</b>	<a href="#">Copy to Clipboard</a>
<b>Program</b> <pre>***** ' Script: InitializeVal.bas ' Function: This script is used to initialize values *****  strYes = "Yes" strNo = "No"</pre>	

```

intYes = 1
intNo = 0
strCallFailure = strNo
strCallerHangup = strNo
strMaxAttempts = strNo
intOne = 1

' Set retry counter and maximum attempts allowed
intTryCount = 1
intMaxAttemptsAllowed = 3

' Set intMaxInputLen to include maximum number allowed plus the termination digit
intMaxInputLen = 6
strTermDigit = "*"

' Set the module to return to if caller hangs up
lctrp = 9910

' Convert system variable $channel from integer to string to be used in concatenated print statement.
IntegerToString strchannel $channel

pline = "***** For channel=" & strchannel
pline = pline & " Started initialization process "
PrintNL pline

```

EndProgram

CAL  
LBa  
sic  
Scri  
pt -  
Get  
DTM  
F-Va  
rLen  
.bas

[Copy  
to  
Clipb  
oard](#)

Program

```

*****
*****
*****
*****
*****
*****
*****
*****
*****
***
'

```

Script:  
GetDTMF-  
VarLen.ba  
s  
,

Function:

This script  
gets a  
variable  
length  
input from  
the  
caller.

```
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
***
```

```
' Set  
up  
parameter  
s for  
GetDTMF  
command
```

```
strNoInput  
= ""
```

```
strNoInput-  
wo-termdig  
it = ""
```

```
    tmask  
=  
strTermDig  
it  
    ndigits  
=  
intMaxInpu  
tLen  
    timeout  
= 15
```

```
GetDTMF  
strNoInput  
tmask  
ndigits  
timeout
```

```
' Get  
length of  
input
```

```
StringLeng  
th intlen  
strNoInput
```

```
    pline =  
"***** For
```

```

channel="
&
strchannel
  pline =
pline & "
strNoInput
entered is
"

  pline =
pline &
strNoInput
  PrintNL
pline

  ' Check
caller
input. The
caller
must enter
more than
1 digit
  ' Save
selected
option

  If intlen
< 2 Then
    pline
= "*****
For
channel="
&
strchannel
  pline
= pline & "
Invalid
strNoInput
entered.
Return to
1190 for
error and
retry"

  PrintNL
pline

  Return
1190
  Else

intTryCoun
t = 0
  EndIf

  ' Strip
off
terminatio
n mask
digit from

```



```

input
length

intlenminus
sone =
intlen - 1

'

Initialize
count start
position
    intstart
= 0

'

Move/Cop
y digits
input to
new
variable

MemMove
strNoInput-
wo-termdig
it
strNoInput
intstart
intlenminus
sone

EndProgra
m

```

<b>CAL LBa sic Scri pt - Che ckTr yCo unt. bas</b>	<a href="#">Copy to Clipb oard</a>
--	--

```

Program
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
Script:

```

CheckTry  
Count.bas  
,

Function:  
This script  
checks  
the  
number of  
invalid  
input  
attempts.

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

' If the  
caller has  
reached  
the  
maximum  
attempts  
allowed  
then the  
call is  
routed to  
module  
9920.

' If not  
then they  
are  
returned  
back to  
module  
1000 to  
retry input  
based on  
Next  
Module ID.

If  
intTryCoun  
t >  
intMaxAtte  
mptsAllow  
ed Then

Return  
9920

Else

intTryCoun  
t =  
intTryCoun

```

t + 1
EndIf

'
Convert
system
variable
intTryCoun
t from
integer to
string to
be used in
concatena
ted print
statement.

IntegerToS
tring
strTryCoun
t
intTryCoun
t

pline =
"***** For
channel="
&
strchannel
pline =
pline & "
TryCount
= "
pline =
pline &
strTryCoun
t
PrintNL
pline

EndProgra
m

```

<b>CAL LBa sic Scri pt - Say Num berl nput .bas</b>	<a href="#">Copy to Clipb oard</a>
---	--

```

Program
*****
*****
*****

```

```

*****
*****
*****
*****
*****
*****
*****
'
Script:
SayNumb
erInput.ba
s
'
Function:
This script
says the
Number
Input.
!*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****

SayNumer
al
strNoInput-
wo-termdig
it

'Print
information
to log file
for
reference
and
debugging
pline =
"***** For
channel="
&
strchannel
pline =
pline &
"*****
Number
played
back to
caller is
*****"

pline =
pline &
strNoInput-

```

wo-termdig  
it  
PrintNL  
pline  
  
EndProgra  
m

CAL LBa sic Scri pt - SetC allF ailur e.ba s	<a href="#">Copy to Clipb oard</a>
---	--

Program  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
'  
Script:  
SetCallFail  
ure.bas  
'  
Function:  
This script  
sets the  
strCallFail  
ure  
variable to  
yes and  
prints  
information  
in the call  
log file.  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

```
strCallFailure = strYes
```

```
    'Print information to log file for reference and debugging  
    pline = "***** For channel=" & strchannel  
    pline = pline & "***** Call Failure.  
*****"
```

```
    PrintNL pline
```

```
EndProgram
```

**CAL  
LBa  
sic  
Scri  
pt -  
SetC  
aller  
Hang  
up.  
bas**

[Copy to Clipboard](#)

Program

```
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****
```

```
'  
Script:  
SetCallerHangup.bas  
,
```

```
Function:  
This script
```

sets the  
strCallFail  
ure  
variable to  
yes and  
prints  
information  
in the call  
log file.

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

strCallerH  
angup =  
strYes

'Print  
information  
to log file  
for  
reference  
and  
debugging  
pline =

"\*\*\*\*\* For  
channel=" &  
strchannel  
pline =  
pline &  
\*\*\*\*\*

Caller  
Hangup.  
\*\*\*\*\*"

PrintNL  
pline

EndProgra  
m

CAL  
LBa  
sic  
Scri  
pt -  
Set  
Max  
Atte

[Copy  
to  
Clipb  
oard](#)

mpts  
.bas

Program

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

'  
Script:  
SetMaxAtt  
empts.bas  
'

Function:  
This script  
sets the  
strMaxAtt  
empts  
variable to  
yes and  
prints  
information  
in the call  
log file.

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

strMaxAtt  
empts =  
strYes

'Print  
information  
to log file  
for  
reference  
and  
debugging  
pline =  
"\*\*\*\*\* For  
channel=" &  
strchannel



```

pline =
pline &
"*****"

Maximum
Attempts
reached to
enter valid
input "*****"
PrintNL
pline

EndProgra
m

```

Now you can create the Audiotex modules and reference the scripts above. Open the **CALLMaster Manager**, click on **Manage**, then select **Audiotex** to add the Audiotex records shown above.

### Define IVR Application List Entry

We need to define the custom IVR in the IVR Application List and set it to Audiotex Start Module 1000. Open CALLMaster Manager and select Manage | IVR Application List. Let's add a new record as follows:

IVR Application List	
Audiotex Start Module	Application Name
1000	Variable Length Input Walkthrough

### Set Ports

As a last step, we need set the port(s) to the new IVR Application entry created above. Make sure the Audiotex Start Module field (read only) correctly points to module 1000.

Set the Port Greeting File IDs to the main greeting for all calls if you have recorded one. If you do not want a main greeting then set the Greeting File IDs to zero. Leave all other fields to the default values.

If you are working with a demo license or have only one line connected, then make sure the connected port is set as specified below. All other ports should be set to Application 'Disabled'.

Port Records			
Port #	Application	Audiotex Start Module	Port Greeting File IDs (Morning, Afternoon, Evening, Closed, Holidays)
1	Variable Length Input Walkthrough	1000	0, 0, 0, 0, 0
2	Variable Length Input Walkthrough	1000	0, 0, 0, 0, 0
3	Variable Length Input Walkthrough	1000	0, 0, 0, 0, 0

4	Variable Length Input Walkthrough	1000	0, 0, 0, 0, 0
---	-----------------------------------	------	---------------

### Test the custom application

Save all your work.

Stop CALLMaster service.

Make Administrator Audiotex Recordings:

- Set Port 1 to Application Voice Mail.
- Start CALLMaster service.
- Make a call using Port 1.
- [Log in](#) as the Administrator using the Voice Mail Main Menu and make recordings for the Audiotex modules below.

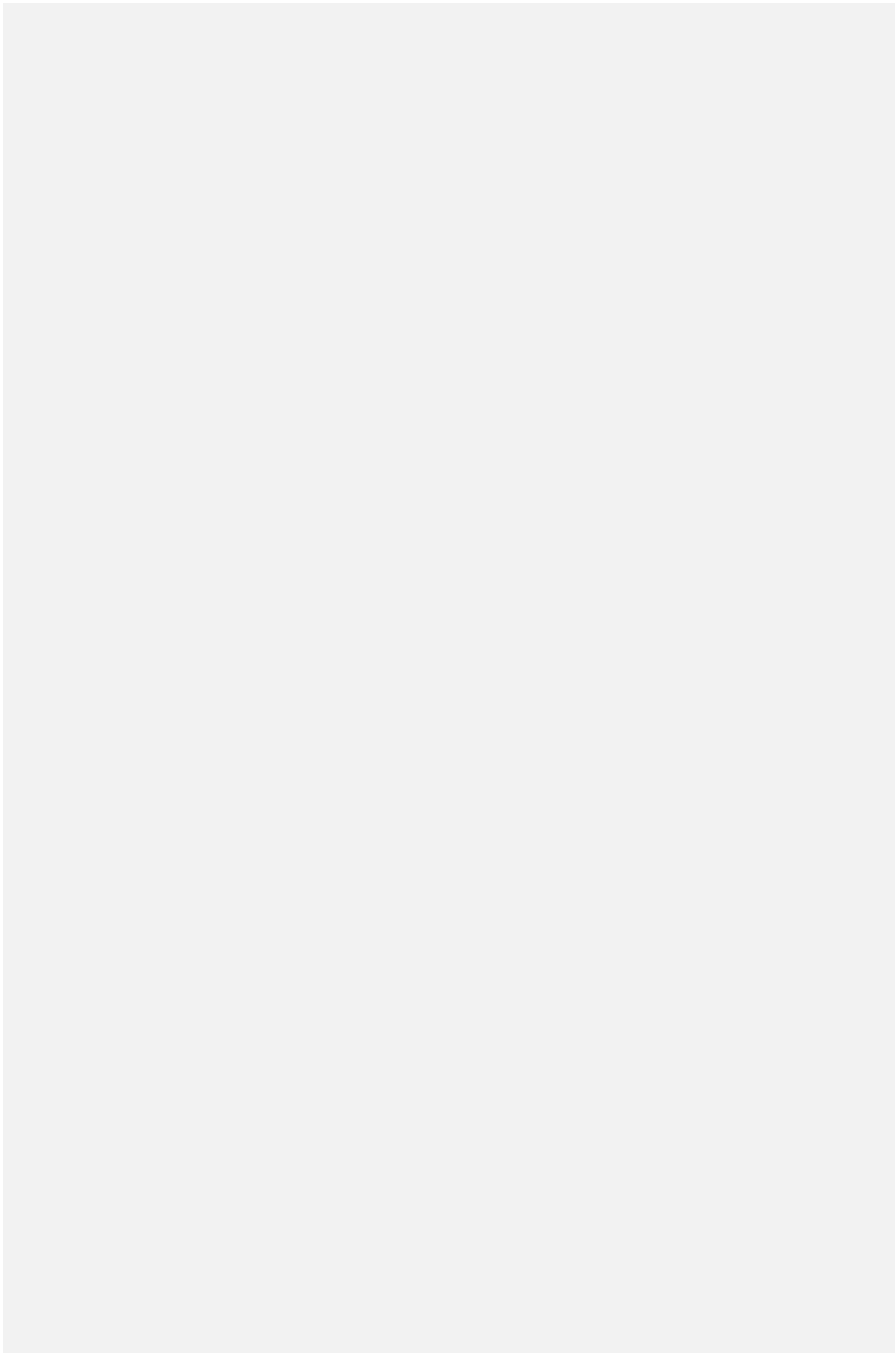
Audiotex Module ID	Proposed Recording
1100	Please enter you ID number followed by the * key. Remember, your ID cannot be greater than 5 digits.
1190	Invalid option.
1191	Please try again
1200	You entered ...
9900	We are unable to process your call. Please contact the system administrator for further assistance.
9920	You have reached the maximum attempts allowed for a valid input. Please contact the system administrator for further assistance.

Reset Port 1 to Application Audiotex. Set the Port Greeting File IDs to zeros.

Stop and Start CALLMaster service.

Make a phone call to test the application.

Tip: See [Walkthrough: Change Application Without Restarting CALLMaster Service](#) to set port to Voice Mail to make recording without stopping and restarting the CALLMaster service.



## Walkthrough: Managing Date/Time Data Type

There are 5 CALLBasic commands that require a date/time SQL data type variable:

- GetDate
- MakeDate
- AddDate
- SayDate
- SayTime

The script below shows examples of using these CALLBasic commands.

### CALLBasic Script - GetDate-AddDate-MakeDate.bas

[Copy to Clipboard](#)

#### Program

```
*****
```

```
' Script: GetDate-AddDate-MakeDate.bas
```

```
' Function: Manipulate Date Time variable
```

```
*****
```

#### ' Example 1

```
' Make 2 Date/Time data type fields
```

```
' dteStartDate will be set to current date and time
```

```
' dteEndDate will be 1 day and 2 hours later than the dteStartDate
```

```
dteStartDate = $today
```

```
dteEndDate = dteStartDate
```

```
intDayInc = 1
```

```
intHourInc = 2
```

```
intMinInc = 0
```

```
intSecInc = 0
```

```
AddDate dteEndDate intDayInc intHourInc intMinInc intSecInc
```

```
Print "Start Date is "
```

```
PrintNL dteStartDate
```

```
Print "End Date is "
```

```
PrintNL dteEndDate
```

```
'If today is 09/30/07 10:30:00 then
```

```
'dteStartDate = 09/30/07 10:30:00
```

```
'dteEndDate = 10/1/07 12:30:00
```

#### ' Example 2

```
' Get date from a date/time data type system variable, local variable or database field
```

```
' (e.g. OutcallSchedule date/time field UserDateTime, system variable $oduserdatetime)
```

```
' Make new date by replacing time component with string system variable
```

```
' (e.g. OutcallSchedule string field MailBox, system variable $odmailbox)
```

```
' Get components of $oduserdatetime field or replace with local date variable from Example 1 (dteEndDate)
```

```
dteApptDate = $oduserdatetime
```

```
' If you do not want to use $oduserdatetime field or replace with local date variable from Example 1. Uncomment statement
```

```
'dteApptDate = dteEndDate
```

```
GetDate dteApptDate intApptMM intApptDay intApptYear intHour intMin intSec
```

```

' Get time from Mailbox field. Check # of digits to if variable length string provided.
' If 3 digits then use digit 1 for hour, digits 2,3 for min else digits 1,2 for hour and digits 3,4 for min
ApptTime = $odmailbox

' If you do not want to use $odmailbox, you can replace with a literal as shown below
'ApptTime = "945"

StringLength len ApptTime
If len = 3 Then
    MemMove strHour ApptTime 0 1
    MemMove strMin ApptTime 1 2
Else
    MemMove strHour ApptTime 0 2
    MemMove strMin ApptTime 2 2
EndIf

' Convert from string to integer to use with the MakeDate command
GetInteger intApptHour strHour 0
GetInteger intApptMin strMin 0
intApptSec = 0

' Make date
MakeDate dteApptDateTime intApptMM intApptDay intApptYear intApptHour intApptMin intApptSec

'If $oduserdatetime is 09/30/07 10:30:40 and $odmailbox is "945" then
'dteApptDateTime = 09/30/07 9:45:00

' Use the SayDate command to say the date component
SayDate dteApptDateTime

' Use the SayTime command to say the time component
SayTime dteApptDateTime

EndProgram

```

## Walkthrough: Play Audiotex Message Based on a System Variable and Get Caller Input Using Touch Tone Table

A System Variable returns information about the status of a call or the environment relating to the call. A system variable can be the DNIS (Number Dialed by the caller), the line number answering the call or some other information. See [System Variables](#).

In this walkthrough we will send the caller to an Audiotex module based on the DNIS. The caller will hear a custom greeting and will be given the ability to select from several options. Though each DNIS can have a custom greeting, the options provided will be the same system wide. Options provided will allow caller to press 1 to hear the custom greeting again, 2 to leave a message for the DNIS admin or 3 to transfer.

DNIS is available only for T1 systems and returned by your T1 provider. The number of DNIS digits returned is based on your T1 provider and set-up in your T1 configuration file. See [T1 Setup Guide](#).

### Walkthrough Overview

In this walkthrough, you will:

1. Set-up Voice Mail Parameters to set Mailbox Length to the same number of digits as DNIS received;
2. Create Mailbox Records for each DNIS with email and phone notification;
3. Set System Parameter required for notification to values required by your phone system and phone lines;
4. Create Notification Records required by the Mailbox owner;
5. Verify Attendant Parameters;
6. Set-up Attendant Record for caller transfer option;
7. Create Audiotex Modules and Associated Scripts to handle the call flow;
8. Define entry in the IVR Application List for the new custom IVR;
9. Set Ports to start in Audiotex mode.

During this walkthrough, you will learn to do the following:

- Configure CALLMaster to answer in Audiotex mode;
- Get value of system variable and assign to local variable;
- Use the CALLBasic Return command with a local variable;
- Send caller to another Audiotex Module, Mailbox or Attendant ID based on the Audiotex Module Action;
- Get caller's input using Touch Tone Table and process selected option;
- Notify mailbox owner via email and phone when caller has left a message in mailbox;
- Transfer caller;
- Allow mailbox owner to log in during greeting;
- Enable mailbox owner to make Audiotex recordings;
- Handle caller hang up and call failure.

### Prerequisites and Assumptions

- CALLMaster software installed and running with the Dialogic board installed and configured. For more information, see [Installation and Setup](#).
- A working knowledge of [CALLMaster Manager](#). Specifically how to add, delete and edit records.
- CALLMaster setup for T1 service. T1 operational and tested. You need to know the number of digits your T1 provider will transmit for the DNIS. For more information, see [T1 Setup Guide](#). For this walkthrough we will assume that the number of DNIS digits transmitted is 10.
- The DNIS enabled for your service. We will assume that there are 3 DNIS enabled for the following numbers: 8005551111, 8005552222, 8005553333.
- Email activated and tested. See [Activating Email](#).

### Setup the Voice Mail Parameters.

As we need to make the mailbox number the same as the DNIS returned, we need to make sure that the Mailbox Length is set to the same number of digits received as the DNIS (assumed to be 10 digits for this walkthrough).

Open the **CALLMaster Manager**. Click on **Parameters**, then select **Voice Mail**. Double click on each of

the parameters below to change their values.

- **Mailbox Length** - Change the value to 10, click **OK**.
- **Password Length** - Change to the desired maximum length. For this walkthrough, set it to 4, click **OK**.

Save and close the Voice Mail Parameters window.

## Create Mailbox Records

We need to create a mailbox record for each DNIS enabled. For each record, the **Mailbox #** will be equal to a DNIS, the **Type** set to User, **Allow Login During Greeting** enabled. Add any other optional information as desired.

Note: Keep Mailbox record 0000 with the default settings as CALLMaster requires it. You should change the password for this record to maintain security.

If the mailbox owner wishes to receive email notification of messages or faxes received in their mailbox, enter their email address in the **Email** field.

Open the **CALLMaster Manager**. Click on **Manage**, then select **Mailboxes**. Let's add 3 records in this walkthrough as follows:

Mailbox Records							
Mailbox #	Password	Last Name	Email	Box Status	Allow Log in During Greeting	Notification Enabled	Notify per Msg
8005551111	1111	DNIS1111	8005551111@txt.att.net	User	Yes	Yes	Yes
8005552222	2222	DNIS2222	DNIS2222@support.com	User	Yes	Yes	
8005553333	3333	DNIS3333	DNIS3333@service.com; 8005553333@vtext.com	User	Yes		

By setting all records to [Allow Login During Greeting](#), the mailbox owner will be able to log in while the greeting is played to [Set Personal Options](#) for the mailbox. The changes made will take effect as soon as the owner logs out.

## Verify System Parameters

As we are using Notification, we need to make sure that the System Parameters - Local Dialing Prefix, Long Distance Dialing Prefix, System's Area Code and System's Phone Number have the correct values for your environment.

Open the **CALLMaster Manager**. Click on **Parameters**, then select **System**. Double click on each of the parameters to change their values as needed. Remember, CALLMaster will use this information with the Notification data input below to make notification calls. See [System Parameters](#) for details.

For this walkthrough we will assume the following values:

- **Local Dialing Prefix** - set to 9.
- **Long Distance Dialing Prefix** - set to 9,1,.
- **System's Area Code** - set to 914.
- **System's Phone Number** - set to 2735560

## Create Notification Records

As we have enabled the Notification On indicator for Mailboxes 8005551111 and 8005552222, we need to create Notification records for them.

Open the **CALLMaster Manager**. Click on **Manage**, then select **Notification** to add the records specified below. For each record, set the Earliest Hour to Call, Latest Hour to Call, Days to Call and the Call Attempt Schedule as desired. See [Notification](#) for details.

Notification Records				
Mailbox #	Record #	Call Type	Area Code	Phone Number
8005551111	1	Human	914	1111111
8005552222	1	Human	914	2222222

### Verify Attendant Parameters

To transfer calls, you must know the requirements of your phone lines or PBX. In particular, you need to know the transfer string to initiate a transfer, and the pull back strings to drop a call in case of busy or no answer. Since the caller can enter a variable length extension ID, you need to set the Maximum Extension Length parameter.

Open the **CALLMaster Manager**. Click on **Parameters**, then select **Attendant**. Confirm values for all the parameters on the list. Double click on the parameter to change the value if required. For this walkthrough, we will assume the values below. Save and close the Attendant Parameters window after making any changes.

Attendant Parameters				
Allow Any Transfer	Drop 'Busy' Call	Drop 'No Answer' Call	Max Extension Length	Transfer String
No	&,	&,	4	&,

### Set-up Attendant Records

As one of the options offered to the caller is to be transferred to an agent, we need to setup Attendant records. We will create an Attendant record for each DNIS enabled. For each record, set the **Attendant ID** to a DNIS, the **Time Of Day** to All Day, and the **Type of Transfer** to Transfer.

Since the **Type of Transfer** is set to Transfer, the **Failed Transfer Action** fields do not apply. If you set the **Type of Transfer** to Supervised, requiring CALLMaster to monitor call transfer status, then you must set the **Failed Transfer Action** fields as desired.

We will also setup an evening record for one of the Attendants to enable caller to go directly to the associated mailbox during the evening hours (between the [System Parameters](#) Evening Starts At and Morning Starts At).

See [Managing Attendant](#) for details.

Open the **CALLMaster Manager**. Click on **Manage**, then select **Attendant**. Let's add 4 records in this walkthrough as follows:



Attendant Records				
Attendant ID	Time of Day	Type of Transfer	Number to Dial	Mailbox
8005551111	All Day	Transfer	9,1,2125551111	
8005552222	All Day	Transfer	9,5552222	
8005552222	Evening	Mailbox		8005552222
8005553333	All Day	Transfer	9,1,2125553333	

### Create Audiotex Records/Modules and Associated Scripts

As all calls will be starting in Audiotex mode in this walkthrough, the Audiotex Modules will determine the call flow.

In the Audiotex set-up, the caller will first hear the greeting/message recorded for the DNIS called. They will then be given the option to "Press 1 to hear DNIS greeting again, 2 to leave message or 3 to Transfer".

We will need to create several Audiotex records as shown below. The Label field provides a brief description of each module.

Audiotex Records																	
Module ID	Label	Write Label to Log File	Module Action	Next Module ID	Fail Module ID	Response Length	Script File to Run (Must include full path name)	Touch Tone Table									
								1	2	3	4	5	6	7	8	9	0
100	Start call and Initialize values	Yes	Next Module	110	9900	0	InitializeVal.bas										
110	Send caller to associated	Yes	Next Module	9999	9900	0	ReturnVar.bas										

	Audi otex modu le																
120	Send caller to asso ciate d Mailb ox to leave mess age	Yes	Mailb ox	9999	9900	0	Retur nVar. bas										
130	Send caller to asso ciate d Atten dant ID to be transf erred	Yes	Atten dant ID	9999	9900	0	Retur nVar. bas										
190	Invali d input.	Yes	Next Modu le	191	9900	0	Chec kTry Coun t.bas										
191	Try again .	Yes	Next Modu le	1000	9900	0											
1000	Provi de optio ns "1 to hear DNIS greeti ng again , 2 to leave mess age or 3	Yes	Touc h Tone Table	9920	9900	1		110	120	130	190	190	190	190	190	190	190

	to Transfer" and get caller response																
9900	Call Failure Detected	Yes	Next Module	9999	9999	0	SetCallFailure.bas										
9910	Caller Hang-up Detected	Yes	Next Module	9999	9999	0	SetCallerHangup.bas										
9920	Maximum tries attempted	Yes	Next Module	9999	9999	0	SetMaxAttempts.bas										
9999	Last Module - End call	Yes	Last Module														
8005551111	8005551111 Play custom recording for DNIS and return to module 120	Yes	Next Module	1000	9900	0											

8005 5522 22	8005 5522 22 Play custo m recor ding for DNIS  and retur n to modu le 120	Yes	Next Modu le	1000	9900	0											
8005 5533 33	8005 5533 33 Play custo m recor ding for DNIS  and retur n to modu le 120	Yes	Next Modu le	1000	9900	0											

#### Audiotex Module Description

- Module 100 is the starting module. We will use this module to initialize variables using script InitializeVal.bas. In particular we want to set a local variable strRetVar, to the value of the system variable \$ddn. We also want to set the system variable lctrap to an Audiotex module to send caller if they hang-up. See sample code below.
- The CALLBasic script ReturnVar.bas is used in 3 different modules (110, 120, 130), each module performing a different function. This is possible as the Return command in the script just returns a value based on a variable, in this case the variable strRetVar. The Module Action determines the return action type.
- Module 1000 is a Touch Tone Table action module requiring a 1 digit input (response length set to 1). This type of module is used to offer caller options and get a corresponding touch tone input. This module offers the caller 3 options requiring a valid touch tone input of either 1, 2 or 3. The caller will be directed to module 110 if they enter 1, to module 120 if they enter 2 or module 130 if they enter 3. If they enter any other touch tone key, they will be directed to module 190. As a Touch Tone Table action module, at the end of the module recording, CALLMaster will play prompt 245 "press \* to repeat", giving the caller 3 attempts to enter any touch tone key (does not have to be a valid input). If you want to give the caller 3 attempts to press any key but do not want to play this prompt, then you can replace the associated wav file with a null recording (see [Replacing Recordings](#)). If you do not want to give the caller 3 attempts to enter any key, use the [NTR](#) code in the Audiotex record Label field.
- Module 190 and 191 is to notify the caller that they have entered an invalid option. The associated script, CheckTryCount.bas, gives the caller 3 attempts to enter a valid option. If the caller has not

entered a valid option after 3 attempts, processing will be forwarded to module 9920.

- Module 9920 is used to notify caller that they have reached the maximum attempts allowed for a valid input. The script associated with this module can be used to set flags and route processing as required.
- Modules 9900 and 9910 are required to handle call failure and capture caller hang-up respectively. Both these modules can be used to set flags and route processing as required.
- As a default, only the System Administrator can make recordings for Audiotex modules. As we want to allow the mailbox owner to make recordings for their own Audiotex module, we need to enter the Mailbox number as the first value in the [Label](#) field.  
Therefore the first value in the Label field for Audiotex modules 8005551111, 8005552222 and 8005553333 is their associated Mailbox #.

Based on the Audiotex modules shown above, we will need several scripts.

Let's create the scripts first. Open Notepad and create scripts following code examples below. Remember to save them as .bas type files in the Program Files\CALLMaster\Scripts folder.

#### CALLBasic Script - InitializeVal.bas

[Copy to Clipboard](#)

##### Program

```
*****
' Script: InitializeVal.bas
' Function: This script is used to initialize values
*****

strYes = "Yes"
strNo = "No"
intYes = 1
intNo = 0
strCallFailure = strNo
strCallerHangup = strNo
strMaxAttempts = strNo

' Set retry counter and maximum attempts allowed
intTryCount = 1
intMaxAttemptsAllowed = 3

' Assign the DNIS value from system variable $ddn to strRetVar
strRetVar = $ddn

' If you want to determine the Audiotex module based on the Port #,
' then comment line above (strRetVar = $ddn)
' and uncomment If..Else..EndIf statement below
'If $channel <= 2 Then
' strRetVar = "8005551111"
'Else
' strRetVar = "8005552222"
'EndIf

' Set the module to return to if caller hangs up
lctrp = 9910

'Print information to log file for reference and debugging
pline = "Call received for DNIS " & strRetVar
PrintNL pline
```

EndProgram

[Copy to Clipboard](#)

## CALLBasic Script - ReturnVar.bas

### Program

```
*****
' Script: ReturnVar.bas
' Function: This script uses the Return command with a variable.
*****

' When used with an Audiotex module whose Module Action is 'Next Module',
' the caller will be forwarded to the Audiotex module whose number is equal to variable strRetVal.
Return strRetVal

EndProgram
```

[Copy to Clipboard](#)

## CALLBasic Script - CheckTryCount.bas

### Program

```
*****
' Script: CheckTryCount.bas
' Function: This script checks the number of invalid input attempts.
*****

' If the caller has reached the maximum attempts allowed then the call is routed to module 9920.
' If not then they are returned back to module 1000 to retry input based on Next Module ID.
If intTryCount > intMaxAttemptsAllowed Then
    Return 9920
Else
    intTryCount = intTryCount + 1
EndIf

' Convert system variable intTryCount from integer to string to be used in concatenated print statement.
IntegerToString strTryCount intTryCount

pline = "***** For channel=" & strchannel
pline = pline & " TryCount = "
pline = pline & strTryCount
PrintNL pline

EndProgram
```

[Copy to Clipboard](#)

## CALLBasic Script - SetCallFailure.bas

### Program

```
*****
' Script: SetCallFailure.bas
' Function: This script sets the strCallFailure variable to yes and prints information in the call log file.
*****

strCallFailure = strYes

'Print information to log file for reference and debugging
pline = "***** Call Failure. *****"
PrintNL pline
```

EndProgram

#### CALLBasic Script - SetCallerHangup.bas

[Copy to Clipboard](#)

Program

```
*****  
' Script: SetCallerHangup.bas  
' Function: This script sets the strCallFailure variable to yes and prints information in the call log file.  
*****  
  
strCallerHangup = strYes  
  
'Print information to log file for reference and debugging  
pline = "***** Caller Hangup. *****"  
PrintNL pline
```

EndProgram

#### CALLBasic Script - SetMaxAttempts.bas

[Copy to Clipboard](#)

Program

```
*****  
' Script: SetMaxAttempts.bas  
' Function: This script sets the strMaxAttempts variable to yes and prints information in the call log file.  
*****  
  
strMaxAttempts = strYes  
  
'Print information to log file for reference and debugging  
pline = "***** Maximum Attempts reached to enter valid input *****"  
PrintNL pline
```

EndProgram

Now you can create the Audiotex modules and reference the scripts above. Open the **CALLMaster Manager**, click on **Manage**, then select **Audiotex** to add the Audiotex records shown above.

### Define IVR Application List

We need to define the custom IVR in the IVR Application List and set it to Audiotex Start Module 100. Open CALLMaster Manager and select Manage | IVR Application List. Let's add a new record as follows:

IVR Application List	
Audiotex Start Module	Application Name
100	DNIS Based Processing Walkthrough

### Set Ports

As a last step, we need set the port(s) to the new IVR Application entry created above. Make sure the Audiotex Start Module field (read only) correctly points to module 100.

As this walkthrough uses Notification, make sure at least one port is enabled for Type Outbound; all the ports should be enabled for Inbound. Set the Port Greeting File IDs to the main greeting for all calls if you have recorded one. If you do not want a main greeting then set the Greeting File IDs to zero. Leave all other fields to the default values.

If you are working with a demo license or have only one line connected, then make sure the connected port is set as specified below. All other ports should be set to Application 'Disabled'.

Port Records				
Port #	Application	Audiotex Start Module	Port Type Outbound	Port Greeting File IDs (Morning, Afternoon, Evening, Closed, Holidays)
1	DNIS Based Processing Walkthrough	100	No	0, 0, 0, 0, 0
2	DNIS Based Processing Walkthrough	100	No	0, 0, 0, 0, 0
3	DNIS Based Processing Walkthrough	100	Yes	0, 0, 0, 0, 0
4	DNIS Based Processing Walkthrough	100	Yes	0, 0, 0, 0, 0

### Test the custom application

Save all your work.

Stop CALLMaster service.

Make Administrator Audiotex Recordings:

- Set Port 1 to Application Voice Mail.
- Start CALLMaster service.
- Make a call using Port 1.
- [Log in](#) as the Administrator using the Voice Mail Main Menu and make recordings for the Audiotex modules below.

Tip:

- The Administrator should record a general message for one of the DNIS Audiotex modules and note the Message ID from this module. This Message ID should be used when creating new DNIS Audiotex Modules. e.g. make a recording for module 8005551111 and use the Message ID to update modules 8005552222 and 8005553333.
- Similarly, the Administrator should make a general greeting for a Mailbox and use the Greeting File ID as the initial recordings for new mailbox records.



The general Audiotex recording and Mailbox greeting will provide a reference for the DNIS owner to make their custom recordings.

<b>Audiotex Module ID</b>	<b>Proposed Recording</b>
190	Invalid option.
191	Please try again
1000	Press 1 to hear DNIS greeting again, 2 to leave message or 3 to Transfer
9900	We are unable to process your call. Please contact the system administrator for further assistance.
9920	You have reached the maximum attempts allowed for a valid input. Please contact the system administrator for further assistance.
8005551111	General recording for DNIS Audiotex module

<b>Mailbox #</b>	<b>Proposed Greeting</b>
8005551111	Hi, at the tone please record a message for this mailbox. Hang up when finished.

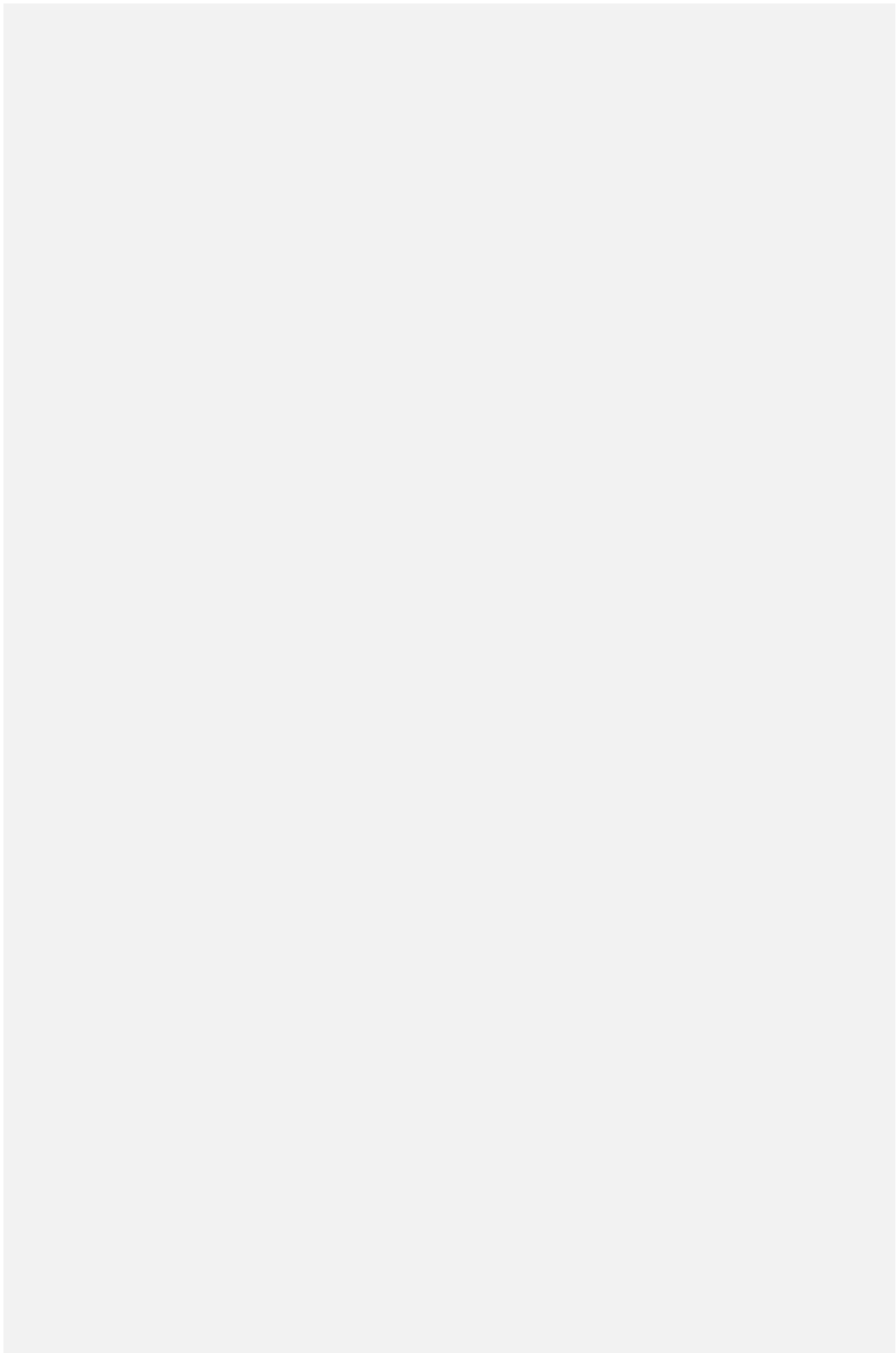
Reset Port 1 to Application Audiotex. Set the Port Greeting File IDs to zeros.

Stop and Start CALLMaster service.

Make a phone call to test the application.

Make the custom Mailbox greetings and corresponding Audiotex Recordings

- Have the Mailbox owner call the system using their DNIS;
- Select option 2 to leave a message. As the mailbox records are enabled to [Allow Login During Greeting](#), the mailbox owner will be able to log in while the greeting is played to [Set Personal Options](#) for the mailbox.



## Walkthrough: Send Caller to a Mailbox Based on the Value of a System Variable

A System Variable returns information about the status of a call or the environment relating to the call. A system variable can be the DNIS (Number Dialed by the caller), the line number answering the call or some other information. See [System Variables](#).

In this walkthrough we will send the caller to the mailbox based on the DNIS called. The caller will hear the mailbox greeting and be able to leave a message. The mailbox owner will be notified by email and phone when a caller has left a message.

DNIS is available only for T1 systems and returned by your T1 provider. The number of DNIS digits returned is based on your T1 provider and set-up in your T1 configuration file. See [T1 Setup Guide](#).

### Walkthrough Overview

In this walkthrough, you will:

1. Set-up Voice Mail Parameters;
2. Create Mailbox Records;
3. Verify System Parameters;
4. Create Notification Records;
5. Create Audiotex Records/Modules and Associated Scripts;
6. Define entry in the IVR Application List for the new custom IVR;
7. Set Ports.

During this walkthrough, you will learn to do the following:

- Configure CALLMaster to answer in Audiotex mode;
- Get value of a system variable and assign it to a local variable;
- Send caller to the mailbox associated with the local variable using the CALLBasic Return command;
- Setup mailbox to allow mailbox owner to log in during greeting;
- Notify mailbox owner via email and phone when caller has left a message in mailbox;
- Handle caller hang up and call failure.

### Prerequisites and Assumptions

- CALLMaster software installed and running with the Dialogic board installed and configured. For more information, see [Installation and Setup](#).
- A working knowledge of [CALLMaster Manager](#). Specifically how to add, delete and edit records.
- CALLMaster setup for T1 service. T1 operational and tested. You need to know the number of digits your T1 provider will transmit for the DNIS. For more information, see [T1 Setup Guide](#). For this walkthrough we will assume that the number of DNIS digits transmitted is 10.
- The DNIS enabled for your service. We will assume that there are 3 DNIS enabled for the following numbers: 8005551111, 8005552222, 8005553333.
- Email activated and tested. See [Activating Email](#).

### Set-up the Voice Mail Parameters.

As the Mailbox number is the same as the DNIS returned, we need to make sure that the Mailbox Length is set to the same number of digits received as the DNIS (assumed to be 10 digits for this walkthrough).

Open the **CALLMaster Manager**. Click on **Parameters**, then select **Voice Mail**. Double click on each of the parameters below to change their values.

- **Mailbox Length** - Change the value to 10, click **OK**.
- **Password Length** - Change to the desired maximum length. For this walkthrough, set it to 4, click **OK**.

Save and close the Voice Mail Parameters window.

### Create Mailbox Records

We need to create a mailbox record for each DNIS enabled. For each record, the **Mailbox #** will be equal to a DNIS, the **Type** set to User, **Allow Login During Greeting** enabled. Add any other optional information

as desired.

Note: Keep Mailbox record 0000 with the default settings as CALLMaster requires it. You should change the password for this record to maintain security.

If the mailbox owner wishes to receive email notification of messages or faxes received in their mailbox, enter their email address in the **Email** field.

Open the **CALLMaster Manager**. Click on **Manage**, then select **Mailboxes**. Let's add 3 records in this walkthrough as follows:

Mailbox Records							
Mailbox #	Password	Last Name	Email	Box Status	Allow Log in During Greeting	Notification Enabled	Notify per Msg
8005551111	1111	DNIS1111	8005551111@txt.att.net	User	Yes	Yes	Yes
8005552222	2222	DNIS2222	DNIS2222@support.com	User	Yes	Yes	
8005553333	3333	DNIS3333	DNIS3333@service.com; 8005553333@vtext.com	User	Yes		

By setting all records to [Allow Login During Greeting](#), the mailbox owner will be able to log in while the greeting is played to [Set Personal Options](#) for the mailbox. The changes made will take effect as soon as the owner logs out.

## Verify System Parameters

As we are using Notification, we need to make sure that the System Parameters - Local Dialing Prefix, Long Distance Dialing Prefix, System's Area Code and System's Phone Number have the correct values for your environment.

Open the **CALLMaster Manager**. Click on **Parameters**, then select **System**. Double click on each of the parameters mentioned above to change their values as needed. Remember, CALLMaster will use this information with the Notification data input below to make notification calls. See [System Parameters](#) for details.

For this walkthrough we will assume the following values:

- **Local Dialing Prefix** - set to 9.
- **Long Distance Dialing Prefix** - set to 9,1,.
- **System's Area Code** - set to 914.
- **System's Phone Number** - set to 2735560

## Create Notification Records

As we have enabled the Notification On indicator for Mailboxes 8005551111 and 8005552222, we need to create Notification records for them.

Open the **CALLMaster Manager**. Click on **Manage**, then select **Notification** to add the records specified below. For each record, set the Earliest Hour to Call, Latest Hour to Call, Days to Call and the Call Attempt Schedule as desired. See [Notification](#) for details.

Notification Records
----------------------

Mailbox #	Record #	Call Type	Area Code	Phone Number
8005551111	1	Human	914	1111111
8005552222	1	Human	914	2222222

### Create Audiotex Records/Modules and Associated Scripts

As all calls will be starting in Audiotex mode in this walkthrough, the Audiotex Modules will determine the call flow.

In this Audiotex set-up, the caller is forwarded to a mailbox based on the DNIS called.

We will need to create several Audiotex records as shown below. The Label field provides a brief description of each module.

Audiotex Records							
Module ID	Label	Write Label to Log File	Module Action	Next Module ID	Fail Module ID	Mailbox	Script File to Run (Must include full path name)
1000	Start call and Initialize values	Yes	Next Module	1100	9900	0000	InitializeVal.bas
1100	Check DNIS Send caller to associated Mailbox	Yes	Mailbox	9999	9900	0000	ReturnMailbox.bas
9900	Call Failure Detected	Yes	Next Module	9999	9999	0000	SetCallFailure.bas
9910	Caller Hang-up Detected	Yes	Next Module	9999	9999	0000	SetCallerHangup.bas
9999	Last Module - End call	Yes	Last Module				

#### Audiotex Module Description

- Module 1000 is the starting module. We will use this module to initialize variables using script InitializeVal.bas. In particular we want to set a local variable strMB, to the value of the system variable \$ddn. We also want to set the system variable lctrp to an Audiotex module to send caller if they hang-up. See sample code below.
- Module 1100 uses the Return Command to send caller to Mailbox (Module Action) based on the local variable strMB.

- Modules 9900 and 9910 are required to handle call failure and capture caller hang-up respectively. Both these modules can be used to set flags and route processing as required.

Based on the Audiotex modules shown above, we will need 4 scripts.

Let's create the scripts first. Open Notepad and create scripts following code examples below. Remember to save them as .bas type files in the Program Files\CALLMaster\Scripts folder.

#### CALLBasic Script - InitializeVal.bas

[Copy to Clipboard](#)

##### Program

```
*****
' Script: InitializeVal.bas
' Function: This script is used to initialize values
*****

' This script is used to initialize values
strYes = "Yes"
strNo = "No"
intYes = 1
intNo = 0
strCallFailure = strNo
strCallerHangup = strNo

' Create variable strMB
strMB = ""

' Assign the DNIS value from system variable $ddn to strMB
strMB = $ddn

' If you want to determine mailbox based on the Port #,
' then comment line above (strMB = $ddn)
' and uncomment If..Else..EndIf statement below
'If $channel <= 2 Then
' strMB = "8005551111
'Else
' strMB = "8005552222
'EndIf

' Set the module to return to if caller hangs up
lctrp = 9910

'Print information to log file for reference and debugging
pline = "Call received for DNIS " & strMB
PrintNL pline

EndProgram
```

#### CALLBasic Script - ReturnMailbox.bas

[Copy to Clipboard](#)

##### Program

```
*****
' Script: ReturnMailbox.bas
' Function: This script uses the Return command with a variable.
'           When used with Audiotex Module Action 'Mailbox',
'           the caller will be forwarded to the mailbox whose value is in variable strMB.
*****
```

Return strMB

EndProgram

#### CALLBasic Script - SetCallFailure.bas

[Copy to Clipboard](#)

Program

```
*****  
' Script: SetCallFailure.bas  
' Function: This script sets the strCallFailure variable to yes and prints information in the call log file.  
*****  
  
strCallFailure = strYes  
  
'Print information to log file for reference and debugging  
pline = "***** Call Failure. *****"  
PrintNL pline
```

EndProgram

#### CALLBasic Script - SetCallerHangup.bas

[Copy to Clipboard](#)

Program

```
*****  
' Script: SetCallerHangup.bas  
' Function: This script sets the strCallFailure variable to yes and prints information in the call log file.  
*****  
  
strCallerHangup = strYes  
  
'Print information to log file for reference and debugging  
pline = "***** Caller Hangup. *****"  
PrintNL pline
```

EndProgram

Now you can create the Audiotex modules and reference the scripts above. Open the **CALLMaster Manager**, click on **Manage**, then select **Audiotex** to add the Audiotex records shown above.

### Define IVR Application List Entry

We need to define the custom IVR in the IVR Application List and set it to Audiotex Start Module 1000. Open CALLMaster Manager and select Manage | IVR Application List. Let's add a new record as follows:

IVR Application List	
Audiotex Start Module	Application Name
1000	DNIS Based MB Walkthrough

### Set Ports

As a last step, we need set the port(s) to the new IVR Application entry created above. Make sure the Audiotex Start Module field (read only) correctly points to module 1000.

As this walkthrough uses Notification, make sure at least one port is enabled for Type Outbound; by default, all the ports are enabled for Inbound. Set the Port Greeting File IDs to the main greeting for all calls if you have recorded one. If you do not want a main greeting then set the Greeting File IDs to zero. Leave all other fields to the default values.

If you are working with a demo license or have only one line connected, then make sure the connected port is set as specified below. All other ports should be set to Application 'Disabled'.

Port Records				
Port #	Application	Audiotex Start Module	Port Type Outbound	Port Greeting File IDs (Morning, Afternoon, Evening, Closed, Holidays)
1	DNIS Based MB Walkthrough	1000	No	0, 0, 0, 0, 0
2	DNIS Based MB Walkthrough	1000	No	0, 0, 0, 0, 0
3	DNIS Based MB Walkthrough	1000	Yes	0, 0, 0, 0, 0
4	DNIS Based MB Walkthrough	1000	Yes	0, 0, 0, 0, 0

### Test the custom application

Save all your work.

Stop and Start CALLMaster service.

Make a phone call to test the application.

Make the custom Mailbox greetings.

- Have the Mailbox owner call the system using their DNIS;
- The mailbox owner will be forwarded to the corresponding mailbox. As the mailbox records are enabled to [Allow Login During Greeting](#), the mailbox owner will be able to log in while the greeting is played to [Set Personal Options](#) for the mailbox.



## Walkthrough: Set-up Attendant with Transfer, Fax Receive and Audiotex Processing

The purpose of the Attendant is to route callers to extensions or departments, voice mailboxes, alternate messages, or Audiotex. The Attendant begins by presenting various options to the caller through prompt 505, the Attendant Main Menu. Then it monitors for touch-tones and searches the Attendant database for a record with Attendant ID matching the caller's touch-tone input.

When Attendant is the application, CALLMaster:

1. Plays the Port Greeting (prompts 19, 20, 21 based on the time of day), then the Attendant main menu (prompt 505)
2. Waits for touchtone input (if no input is received, it goes to Attendant ID 0)
3. Matches the touchtone input with an Attendant ID record
4. Takes action based on the 'Type of Transfer' for the selected Attendant ID (plays 'there is no such extension' if no match is found)

For this walkthrough, we will be starting all calls in Attendant mode with prompt 505 offering options as follows:

"If you know your party's extension please enter it now. For sales, press 1. For product information, press 2. For the firm directory, press 3."

Since CALLMaster is shipped with default records, some of these records may already exist. If they do then modify the existing record.

## Walkthrough Overview

In this walkthrough, you will learn to:

1. Configure CALLMaster to answer in Attendant mode;
2. Give caller custom options, get their Touch Tone input and process selected option;
3. Verify Attendant Parameters required to transfer a call, or drop call in case of a busy or no answer;
4. Define the Global Tone for faxing;
5. Set-up Mailboxes;
6. Confirm Voice Mail Parameters and Assign Default Mailbox;
7. Notify mailbox owner via email when they receive a message or fax in their mailbox;
8. Notify mailbox owner via phone when they receive a message in their mailbox;
9. Create Attendant Records to route caller to:- be transferred, to leave a message, to log in to a mailbox, to hear the firm directory, or to an Audiotex module to get information;;
10. Automatically receive a fax document which will be saved in the default mailbox; Create Audiotex Records/Modules and Associated Scripts;
11. Set Ports.

## Prerequisites and Assumptions

- CALLMaster software installed and running with the Dialogic board installed and configured. For more information, see [Installation and Setup](#).
- A working knowledge of [CALLMaster Main Window](#).
- For fax capabilities, you must have purchased the CALLMaster Fax License, and have a board that supports fax.
- Phone lines or PBX extensions connected to CALLMaster and tested, including transfers with connect, no answer and busy capabilities.
- Email activated and tested. See [Activating Email](#).

## Verify Attendant Parameters.

To transfer calls, you must know the requirements of your phone lines or PBX. In particular, you need to know the transfer string to initiate a transfer, and the pull back strings to drop a call in case of busy or no answer. Since the caller can enter a variable length extension ID, you need to set the Maximum Extension Length parameter.

Open the **CALLMaster Manager**. Click on **Parameters**, then select **Attendant**. Confirm values for all the parameters on the list. Double click on the parameter to change the value if required. For this walkthrough, we will assume the values below. Save and close the Attendant Parameters window after making any

changes.

Attendant Parameters				
Allow Any Transfer	Drop 'Busy' Call	Drop 'No Answer' Call	Max Extension Length	Transfer String
No	&,	&,	4	&,

### Define Global Tone for Fax

For the Dialogic board to detect a fax tone, we need to define the tone settings in the Global Tones table.

Open the **CALLMaster Manager**. Click on **Manage**, then select **Global Tones**. If there is an record for a Fax tone, then just verify the setting. If not then Add a record to this table with values as below.

Global Tones													
Tone Type	Characteristic	Detection Mode	Translation		Frequencies (Hz)				Cadence (10ms)				
			Translate to Char	Character	Tone 1 Freq	Tone 1 Dev	Tone 2 Freq	Tone 2 Dev	On Time	On Time Dev	Off Time	Off Time Dev	Minimum Repetition Count
Fax	Single Tone	Leading Edge	Yes	F	1100	50	0	0	0	0	0	0	1

### Confirm Voice Mail Parameters and Assign Default Mailbox

As we are setting up this application to receive faxes to the default mailbox, we need to assign a default mailbox in the Voice Mail Parameter. We need to also confirm the default values set for the other Voice Mail parameters.

Open the **CALLMaster Manager**. Click on **Parameters**, then select **Voice Mail**. Double click on each of the parameters to change their values as needed. We will assign Mailbox 1111 as the default mailbox.

- **Default Mailbox** - Change the value to 1111, click **OK to save**.

### Set-up Mailboxes

In this walkthrough, we will set-up 3 mailbox records as shown below. We will keep the Voice Mail Parameters Mailbox Length and Password length to 4. See section below (Confirm Voice Mail Parameters and Assign Default Mailbox) if you need to change these parameters.

If the mailbox owner wishes to receive email notification of messages or faxes received in their mailbox, enter their email address in the **Email** field.

Note: Keep Mailbox record 0000 with the default settings as CALLMaster requires it. You should change the password for this record to maintain security.

Open the **CALLMaster Manager**. Click on **Manage**, then select **Mailboxes**. Let's add Mailbox records in this walkthrough as follows:

Mailbox Records
-----------------

Mailbox #	Password	Last Name	Email	Box Status	Allow Log in During Greeting	Notification Enabled	Notify per Msg
1000	1000	Operator	Oper@sscorp.com	Administrator	Yes		
1111	1111	Sales	MB1111@txt.att.net	User	Yes	Yes	Yes
2222	2222	MB2222	MB2222@support.com	User	No	Yes	
3333	3333	MB3333	MB3333@service.com; MB3333@vtext.com	User	Yes		

## Create Notification Records

As we have Enabled Notification for Mailboxes 1111 and 2222, we need to create Notification records for them.

For each record, set the Earliest Hour to Call, Latest Hour to Call, Days to Call and the Call Attempt Schedule as desired.

Let's create notification records for Mailbox 1111.

Open the **CALLMaster Manager**. Click on **Manage**, then select **Notification** to add the records specified below.

See [Notification](#) for details.

Notification Records				
Mailbox #	Record #	Call Type	Area Code	Phone Number
1111	1	Human	914	1111111
2222	1	Human	914	2222222

## Create Attendant Records

The Attendant processing is described below.

- Most callers who encounter an auto Attendant will press 0 to speak to an operator. If you wish to allow this functionality then create the records for Attendant ID 0.
  - Attendant ID 0, All Day:- CALLMaster will attempt a supervised transfer to extension 100 if a callers presses 0 during the day. If extension 100 is busy or there is no answer, the caller will be sent to Mailbox 1000 based on the Failed Transfer Action and Additional Settings set-up for the record.
  - Attendant ID 0, Evening:- CALLMaster will send caller directly to Mailbox 1000 during the evening hours (between the [System Parameters](#) Evening Starts At and Morning Starts At).
- The similar structure is set-up for Attendant IDs 1, 222, and 333 with minor differences.
  - Attendant ID 9333, Evening:- transfers the call to an external phone number as input in the Number to Dial field. The Number to Dial field must be set-up with the complete dial string to make the call as required by your phone lines or PBX. In this example, we assume that a 9

and 1 is required along with the area code and phone number. As the Type of Transfer for this record is Transfer (i.e. blind transfer ), CALLMaster will just transfer the call and hang up and will not supervise the call connection.

- For all Mailboxes, an Attendant record with 'Type of Transfer' set to "Mailbox" should be created to allow caller to leave a message for the mailbox regardless of the time of day. For example, Attendant ID 8000 allows the caller to enter 9000 from the Attendant Main Menu to leave a message for mailbox 1000. Use Attendant 8001 to leave a message for mailbox 1111; Attendant 8222 to leave a message for mailbox 2222 and Attendant 8333 to leave a message for mailbox 3333.
- For all the Mailbox owners, an Attendant record with 'Type of Transfer' set to "Log in to Mailbox" should be created for direct Mailbox log in. For example, Attendant ID 9000 enables the owner of Mailbox 1000 to enter 9000 from the Attendant Main Menu to log in to mailbox 1000. Use Attendant 9001 to log in to mailbox 1111; Attendant 9222 to log in to mailbox 2222 and Attendant 9333 to log in to mailbox 3333.
- Attendant ID 2 will send caller to Audiotex Module 100. See section below - Set-up Audiotex Modules and Associated Scripts.
- Attendant ID 3 will allow the caller to select an extension from the [Attendant Directory](#) as set-up by the System Administrator.

Open the **CALLMaster Manager**. Click on **Manage**, then select **Attendant** to add the Attendant records required for this walkthrough.

Attendant Records									
Attendant ID	Time of Day	Type of Transfer	Number to Dial	Failed Transfer Action			Additional Settings		
				Busy	No Answer	Screened Reject	Alt Attendant ID	Mailbox #	Audiotex Module ID
0	All Day	Supervised	100	Mailbox	Mailbox			1000	
0	Evening	Mailbox						1000	
8000	All Day	Mailbox						1000	
9000	All Day	Log in to Mailbox						1000	
1	All Day	Supervised	111	Mailbox	Mailbox			1111	
1	Evening	Mailbox						1111	
8001	All Day	Mailbox						1111	
9001	All Day	Log in to Mailbox						1111	
2	All Day	Audiotex							100

		Module							
3	All Day	Directory							
222	All Day	Supervised	222	Mailbox	Mailbox			2222	
222	Evening	Mailbox						2222	
8222	All Day	Mailbox						2222	
9222	All Day	Log in to Mailbox						2222	
333	All Day	Supervised	333	Mailbox	Mailbox			3333	
333	Evening	Transfer	9,1,9143333333						
8333	All Day	Mailbox						3333	
9333	All Day	Log in to Mailbox						3333	

### Create Audiotex Records/Modules and Associated Scripts

As we want to direct the caller to the CALLMaster Attendant, we will use the default Audiotex module 3 to send caller to Attendant. If the caller selects option 2 (for product information) in the Attendant, they will be redirected back to Audiotex Module 100 per the Attendant set-up to give caller with product information.

Let's create several Audiotex records as shown below to handle this request. The Label field provides a brief description of each module.

Audiotex Records																	
Module ID	Label	Write Label to Log File	Module Action	Next Module ID	Fail Module ID	Response Length	Script File to Run (Must include full path name)	Touch Tone Table									
								1	2	3	4	5	6	7	8	9	0
3	Atten	Yes	Atten	0	0	0											

	dant Main Menu		dant														
100	Start call and Initial ize value s	Yes	Next Modu le	105	9900	0	Initial izeV al.ba s										
105	Provi de optio ns "1 for infor matio n about produ ct A, or 2 for infor matio n about produ ct B" and get caller respo nse	Yes	Touc h Tone Table	9920	9900	1		110	120	190	190	190	190	190	190	190	190
110	Infor matio n about produ ct A	Yes	Next Modu le	9999	9900	0											
120	Infor matio n about produ ct B	Yes	Next Modu le	9999	9900	0											
190	Invali d input.	Yes	Next Modu le	191	9900	0	Chec kTry Coun										

							t.bas										
191	Try again	Yes	Next Module	105	9900	0											
9900	Call Failure Detected	Yes	Next Module	9999	9999	0	SetCallFailure.bas										
9910	Caller Hang-up Detected	Yes	Next Module	9999	9999	0	SetCallerHangup.bas										
9920	Maximum tries attempted	Yes	Next Module	9999	9999	0	SetMaxAttempts.bas										
9999	Last Module - End call	Yes	Last Module														

#### Auditex Module Description.

- Module 100 is the starting module. We will use this module to initialize variables using script InitializeVal.bas. In particular we want to set the system variable lctrp to an Auditex module to send caller if they hang-up. See sample code below.
- Module 105 has Module Action 'Touch Tone Table' requiring a 1 digit input (response length set to 1). This type of module is used to offer caller options and get a corresponding touch tone input. This module offers the caller 3 options requiring a valid touch tone input of either 1, 2 or 3. The caller will be directed to module 110 if they enter 1, to module 120 if they enter 2. If they enter any other touch tone key, they will be directed to module 190. As a 'Touch Tone Table' Module Action, at the end of the module recording, CALLMaster will play prompt 245 "press \* to repeat, giving the caller 3 attempts to enter any touch tone key (does not have to be a valid input). If you want to give the caller 3 attempts to press any key but do not want to play this prompt, then you can replace the associated wav file with a null recording (see [Replacing Recordings](#)). If you do not want to give the caller 3 attempts to enter any key, use the [NTR](#) code in the Auditex record Label field.
- Module 190 and 191 is to notify the caller that they have entered an invalid option. The associated script, CheckTryCount.bas, gives the caller 3 attempts to enter a valid option. If the caller has not entered a valid option after 3 attempts, processing will be forwarded to module 9920.
- Module 9920 is used to notify caller that they have reached the maximum attempts allowed for a valid input. The script associated with this module can be used to set flags and route processing as required.
- Modules 9900 and 9910 are required to handle call failure and capture caller hang-up respectively. Both these modules can be used to set flags and route processing as required.

- Module 9999 is the Last Module. Since the call originated in Attendant mode, the caller will be given the option to "press 1 for further Attendant option" (prompt 519).

#### CALLBasic Script - InitializeVal.bas

[Copy to Clipboard](#)

##### Program

```
*****
' Script: InitializeVal.bas
' Function: This script is used to initialize values
*****

strYes = "Yes"
strNo = "No"
intYes = 1
intNo = 0
strCallFailure = strNo
strCallerHangup = strNo
strMaxAttempts = strNo

' Set retry counter and maximum attempts allowed
intTryCount = 1
intMaxAttemptsAllowed = 3

' Set the module to return to if caller hangs up
lctrp = 9910

'Print information to log file for reference and debugging
pline = "Call received for DNIS " & strRetVar
PrintNL pline
```

EndProgram

#### CALLBasic Script - CheckTryCount.bas

[Copy to Clipboard](#)

##### Program

```
*****
' Script: CheckTryCount.bas
' Function: This script checks the number of invalid input attempts.
*****

' If the caller has reached the maximum attempts allowed then the call is routed to module 9920.
' If not then they are returned back to module 105 to retry input based on Next Module ID.
If intTryCount > intMaxAttemptsAllowed Then
    Return 9920
Else
    intTryCount = intTryCount + 1
EndIf

' Convert system variable intTryCount from integer to string to be used in concatenated print statement.
IntegerToString strTryCount intTryCount

pline = "***** For channel=" & strchannel
pline = pline & " TryCount = "
pline = pline & strTryCount
PrintNL pline
```



EndProgram

### CALLBasic Script - SetCallFailure.bas

[Copy to Clipboard](#)

Program

```
*****
' Script: SetCallFailure.bas
' Function: This script sets the strCallFailure variable to yes and prints information in the call log file.
*****

strCallFailure = strYes

'Print information to log file for reference and debugging
pline = "***** Call Failure. *****"
PrintNL pline
```

EndProgram

### CALLBasic Script - SetCallerHangup.bas

[Copy to Clipboard](#)

Program

```
*****
' Script: SetCallerHangup.bas
' Function: This script sets the strCallFailure variable to yes and prints information in the call log file.
*****

strCallerHangup = strYes

'Print information to log file for reference and debugging
pline = "***** Caller Hangup. *****"
PrintNL pline
```

EndProgram

### CALLBasic Script - SetMaxAttempts.bas

[Copy to Clipboard](#)

Program

```
*****
' Script: SetMaxAttempts.bas
' Function: This script sets the strMaxAttempts variable to yes and prints information in the call log file.
*****

strMaxAttempts = strYes

'Print information to log file for reference and debugging
pline = "***** Maximum Attempts reached to enter valid input *****"
PrintNL pline
```

EndProgram

Now you can create the Audiotex modules and reference the scripts above. Open the **CALLMaster Manager**, click on **Manage**, then select **Audiotex** to add the Audiotex records shown above.

## Verify IVR Application List

Open CALLMaster Manager and select Manage | IVR Application List. Make sure there is the default entry for the Attendant as shown below:

IVR Application List	
Audiotex Start Module	Application Name
3	Attendant

## Set Ports

As a last step, we need set the port(s) to application Attendant. As this walkthrough uses Notification, make sure at least one port is enabled for Type Outbound; all the ports should be enabled for Inbound. Set the Port Greeting File IDs to the main greeting for all calls if you have recorded one or you can utilize the default Greeting File IDs. Leave all other fields to the default values. Make sure the Audiotex Start Module field (read only) correctly points to module 3.

If you are working with a demo license or have only one line connected, then make sure the connected port is set as specified below. All other ports should be set to Application 'Disabled'.

Port Records				
Port #	Application	Audiotex Start Module	Port Type Outbound	Port Greeting File IDs (Morning, Afternoon, Evening, Closed, Holidays)
1	Attendant	3	No	19, 20, 21, 203, 209
2	Attendant	3	No	19, 20, 21, 203, 209
3	Attendant	3	Yes	19, 20, 21, 203, 209
4	Attendant	3	Yes	19, 20, 21, 203, 209

## Test the custom application

Save all your work.

Stop and restart CALLMaster service.

Make Recordings:

- o Call the system and enter extension 90 (Attendant ID set-up to log in to Administrator Mailbox 1000).
- o [Log in](#) as the Administrator and access the Administrator Options Menu.
  - To re-record Prompts, select Administrator Option 7

Prompt #	Proposed Recording
505	If you know your party's extension please enter it now. For sales, press 1. For product information, press 2. For the firm directory, press 3.
19	Good morning. Thank you for calling My Company
20	Good afternoon. Thank you for calling My Company
21	Good evening. Thank you for calling My Company
203	Thank you for calling My Company. We are closed today.
209	Thank you for calling My Company. We are closed for the holiday.
- To make recordings for the Audiotex modules below, select Administrator Option 8	
Audiotex Module ID	Proposed Recording
105	Press 1 for information about product A, or 2 for information about product B
110	Information about product A
120	Information about product B
190	Invalid option.
191	Please try again
9900	We are unable to process your call. Please contact the system administrator for further assistance.
9920	You have reached the maximum attempts allowed for a valid input. Please contact the system

administrator for further assistance.

- Make Attendant Directory Entries select Administrator Option 9 (See how to [Set-up Attendant Directory](#)).

Record Mailbox greetings and name.

- Have the Mailbox owner call the system and enter the Attendant ID to log in to their mailbox. They can use Voice Mail Menu Option 3 to [Set Personal Options](#) for their mailbox.

Make test calls to verify desired functionality.

## Walkthrough: Use Global Variable with Entry.bas Script to Play Rotating Advertisement

In this walkthrough we will use a [Global Variable](#) defined in an [Entry.bas script](#) to set the starting Advertisement Number. The Advertisement Number will be incremented sequentially to determine the current Advertisement Number to use. Each Advertisement will be set-up as an Announcement Type Mailbox whose Mailbox # will be equal to the Advertisement Number and the Mailbox Greeting will be the Advertisement message. Each Advertisement message will be played sequentially at the start of a call across all lines.

### Walkthrough Overview

In this walkthrough, you will:

1. Create Mailbox Records;
2. Create Audiotex Records/Modules and Associated Scripts.
3. Define entry in the IVR Application List for the new custom IVR;
4. Set Ports.

During this walkthrough, you will learn to do the following:

- Use a Global variable;
- Create and run an Entry.bas script;
- Set-up an Announcement Status Mailbox with Settings UL3 (limit owner options to 3) and NGN (do not require mailbox name and greeting for first time login);
- Get value of a global variable and assign it to a local variable;
- Send caller to the mailbox associated with the local variable using the CALLBasic Return command;
- Setup mailbox to allow mailbox owner to log in during greeting;
- Handle caller hang up and call failure.

### Prerequisites and Assumptions

- CALLMaster software installed and running with the Dialogic board installed and configured. For more information, see [Installation and Setup](#).
- A working knowledge of [CALLMaster Manager](#). Specifically how to add, delete and edit records.
- Phone lines connected to CALLMaster and tested.

### Create Mailbox Records

For this walkthrough we will use 9900 as the first Advertisement Number and 9905 as the last Advertisement Number. We will reserve mailbox numbers 9900 thru 9999 for Advertisement. There must be a sequentially numbered mailbox records set-up and used.

For each record, the **Mailbox #** will be equal to the Advertisement Number, the **Type** set to Announcement, and **Allow Login During Greeting** enabled. Each mailbox will have **Settings** UL3 (limit owner options to 3) and NGN (do not require mailbox name and greeting for first time login). Add any other optional information as desired.

Note: Keep Mailbox record 0000 with the default settings as CALLMaster requires it. You should change the password for this record to maintain security.

Open the **CALLMaster Manager**. Click on **Manage**, then select **Mailboxes**. Let's add the records in this walkthrough as follows:

Mailbox Records				
Mailbox	Password	Last Name	Settings	Type
9900	9900	Advertisement 9900	UL3 NGN	Announcement
9901	9901	Advertisement 9901	UL3 NGN	Announcement

9902	9902	Advertisement 9902	UL3 NGN	Announcement
9903	9903	Advertisement 9903	UL3 NGN	Announcement
9904	9904	Advertisement 9904	UL3 NGN	Announcement
9905	9905	Advertisement 9905	UL3 NGN	Announcement

As the mailbox has Setting UL3, only the first 3 [Set Personal Option Features](#) will be available. The changes made will take effect as soon as the owner logs out.

## Create Audiotex Records/Modules and Associated Scripts

As all calls will be starting in Audiotex mode in this walkthrough, the Audiotex Modules will determine the call flow.

In the Audiotex set-up, the caller will hear aa Advertisement message. There will be a sequential rotation thru the list of Advertisement, with each Advertiser having the ability to record their message at any time.

We will need to create several Audiotex records as shown below. The Label field provides a brief description of each module.

Audiotex Records							
Module ID	Label	Write Label to Log File	Module Action	Next Module ID	Fail Module ID	Response Length	Script File to Run (Must include full path name)
10	Start call and Initialize values	Yes	Next Module	100	9900	0	InitializeVal.bas
100	Send caller to associated Advertisement Mailbox	Yes	Mailbox	9999	9900	0	GetAdvertisementNo.bas
9900	Call Failure Detected	Yes	Next Module	9999	9999	0	SetCallFailure.bas
9910	Caller Hang-up Detected	Yes	Next Module	9999	9999	0	SetCallerHangup.bas
9999	Last Module - End call	Yes	Last Module				

### Audiotex Module Description

- We start with an Entry.bas script. This script will initialize the Global variable @intAdvertisementNo to 9900 (the first Advertisement Number). This script is placed in the Scripts folder like all the other scripts, however it is not assigned to any Audiotex module. The Entry.bas script will run only when

the CALLMaster service starts. We also want to set the system variable lctrap to an Audiotex module to send caller if they hang-up. See sample code below.

- Module 10 is the starting module. We will use this module to initialize variables using script InitializeVal.bas. In particular, we need to initialize the variable for the first Advertisement Number (intFirstAdvertisementNo) and last Advertisement Number (intLastAdvertisementNo). See sample code below.
- Module 100 selects the next Advertisement message to play. The associated script, GetAdvertisementNo.bas, checks the value of global variable @intAdvertisementNo to see if it is equal to the last Advertisement Number. If it is then @intAdvertisementNo is reset to the first Advertisement Number (intFirstAdvertisementNo); if not then it is incremented by 1. The local variable intCurrAdvertisementMBNo is set to the global variable @intAdvertisementNo as the current Advertisement Number. This value is used to send caller to the Advertisement's mailbox via the Return command.
- Modules 9900 and 9910 are required to handle call failure and capture caller hang-up respectively. Both these modules can be used to set flags and route processing as required.

Based on the Audiotex modules shown above, we will need 4 scripts as well as the Entry.bas script.

Let's create the scripts first. Open Notepad and create scripts following code examples below. Remember to save them as .bas type files in the Program Files\CALLMaster\Scripts folder.

#### CALLBasic Script - Entry.bas

[Copy to Clipboard](#)

##### Program

```
*****
' Script: Entry.bas
' Function: Runs at CALLMaster Service start-up. Used to initialize global variables.
*****

Print "CM Entry Script started on "
PrintNL $today

' @intAdvertisementNo set to the first Advertisement Number
@intAdvertisementNo = 9900

EndProgram
```

#### CALLBasic Script - InitializeVal.bas

[Copy to Clipboard](#)

##### Program

```
*****
' Script: InitializeVal.bas
' Function: This script is used to initialize values
*****

strYes = "Yes"
strNo = "No"
intYes = 1
intNo = 0
strCallFailure = strNo
strCallerHangup = strNo
strMaxAttempts = strNo

' Set retry counter and maximum attempts allowed counter
intTryCount = 1
intMaxAttemptsAllowed = 3
```

```

' Assign the starting and ending Advertisement numbers
intFirstAdvertisementNo = 9900
intLastAdvertisementNo = 9905
intCurrAdvertisementNo = 0

' Set the module to return to if caller hangs up
lctrp = 9910

' Convert system variable $channel from integer to string to be used in concatenated print statement.
IntegerToString strchannel $channel

pline = "***** For channel=" & strchannel
pline = pline & " Started initialization process "
PrintNL pline

```

EndProgram

#### CALLBasic Script - GetAdvertisementNo.bas

[Copy to Clipboard](#)

Program

```

*****
' Script: GetAdvertisementNo.bas
' Function: This script checks the value of global variable @intAdvertisementNo to create the current Advertisement ID.
*****

' If the global variable @intAdvertisementNo is greater than the last Advertisement Number (intLastAdvertisementNo), res
@intAdvertisementNo to the first Advertisement No (intFirstAdvertisementNo); if not then it is incremented by 1.
If @intAdvertisementNo = intLastAdvertisementNo Then
    @intAdvertisementNo = intFirstAdvertisementNo
Else
    @intAdvertisementNo = @intAdvertisementNo + 1
EndIf

' The local variable intCurrAdvertisementMBNo is set as the current Advertisement Number.
intCurrAdvertisementMBNo = @intAdvertisementNo

' Convert local variable intCurrAdvertisementMBNo from integer to string to be used in concatenated print statement.
IntegerToString strCurrAdvertisementMBNo intCurrAdvertisementMBNo
pline = "***** For channel=" & strchannel
pline = pline & " Caller sent to current Advertisement mailbox "
pline = pline & strCurrAdvertisementMBNo
PrintNL pline

' The Return command is used to send caller to the Advertisement's mailbox.
Return intCurrAdvertisementMBNo

```

EndProgram

#### CALLBasic Script - SetCallFailure.bas

[Copy to Clipboard](#)

Program

```

*****
' Script: SetCallFailure.bas
' Function: This script sets the strCallFailure variable to yes and prints information in the call log file.

```



\*\*\*\*\*

```
strCallFailure = strYes
```

```
'Print information to log file for reference and debugging
```

```
pline = "***** Call Failure. *****"
```

```
PrintNL pline
```

```
EndProgram
```

### CALLBasic Script - SetCallerHangup.bas

[Copy to Clipboard](#)

Program

\*\*\*\*\*

```
' Script: SetCallerHangup.bas
```

```
' Function: This script sets the strCallFailure variable to yes and prints information in the call log file.
```

\*\*\*\*\*

```
strCallerHangup = strYes
```

```
'Print information to log file for reference and debugging
```

```
pline = "***** Caller Hangup. *****"
```

```
PrintNL pline
```

```
EndProgram
```

Now you can create the Audiotex modules and reference the scripts above. Open the **CALLMaster Manager**, click on **Manage**, then select **Audiotex** to add the Audiotex records shown above.

### Define IVR Application List Entry

We need to define the custom IVR in the IVR Application List and set it to Audiotex Start Module 10. Open CALLMaster Manager and select Manage | IVR Application List. Let's add a new record as follows:

IVR Application List	
Audiotex Start Module	Application Name
10	Rotating Message Walkthrough

### Set Ports

As a last step, we need set the port(s) to the new IVR Application entry created above. Make sure the Audiotex Start Module field (read only) correctly points to module 10.

As this walkthrough uses Notification, make sure at least one port is enabled for Type Outbound; all the ports should be enabled for Inbound. Set the Port Greeting File IDs to the main greeting for all calls if you have recorded one. If you do not want a main greeting then set the Greeting File IDs to zero. Leave all other fields to the default values.

If you are working with a demo license or have only one line connected, then make sure the connected port is set as specified below. All other ports should be set to Application 'Disabled'.

### Port Records

Port #	Application	Audiotex Start Module	Port Type Outbound	Port Greeting File IDs (Morning, Afternoon, Evening, Closed, Holidays)
1	Rotating Message Walkthrough	10	No	0, 0, 0, 0, 0
2	Rotating Message Walkthrough	10	No	0, 0, 0, 0, 0
3	Rotating Message Walkthrough	10	Yes	0, 0, 0, 0, 0
4	Rotating Message Walkthrough	10	Yes	0, 0, 0, 0, 0

### Test the custom application

Save all your work.

Stop CALLMaster service.

Make Mailbox Recordings:

- Set Port 1 to Application Voice Mail.
- Start CALLMaster service.
- Make a call using Port 1.
- [Log in](#) to each Advertisement Mailbox using the Voice Mail Main Menu and record mailbox name, greeting and directory listing.

Reset Port 1 to Application Audiotex. Set the Port Greeting File IDs to zeros.

Stop and Start CALLMaster service.

Make a phone call to test the application.

Tip: See [Walkthrough: Change Application Without Restarting CALLMaster Service](#) to set port to Voice Mail for ongoing recording changes.

## Walkthroughs with Sample Applications

Walkthroughs should only be used as a template to facilitate your understanding of CALLMaster and assist in your custom application development.

Use the walkthroughs as a learning guide to the various CALLMaster features. Add your own data, modify any of the steps as desired to customize the application to your needs.

Make sure that you start the walkthroughs using a clean install of CALLMaster and the installation has been validated. See [Validating Installation](#) for details.

Click on the desired walkthrough for detailed instructions.

- [Walkthrough: Account Information Retrieval](#)
  - Get Account Balances and Information Based on Fixed Length Input
- [Walkthrough: Appointment Reminder with Reschedule Option](#)
  - Make Outbound Calls for an Appointment Reminder with Option to Reschedule
- [Walkthrough: Broadcast Audio Message and Fax Document](#)
  - Make Outbound Calls to Broadcast Message and Fax Document
- [Walkthrough: Dictation Functionality with Playback Navigation](#)
  - Make New Dictation Recordings and Review Existing Recordings
- [Walkthrough: Out Dial with 'Remove from List' Option](#)
  - Make Outbound Calls with 'Remove from List' Option
- [Walkthrough: Survey Questionnaire](#)
  - Take Survey / Ask Questions with Concatenated Response Logging
- [Walkthrough: Team Bulletin Board with Rotating Sponsor Messages](#)
  - Set-up Team Bulletin Board with Rotating Sponsor Message

The Walkthroughs provided in this section can also be downloaded as [Sample Applications](#).

## Walkthrough: Get Account Balances and Information Based on Fixed Length Input

This walkthrough shows you how to retrieve information from a database based on a caller's input and provide related information back to the caller.

In this walkthrough the caller will be given the option to "Press 1 for Account Information, 2 for general information or 3 to leave a message".

- Option 1: Get balance for the selected Account.  
The caller will be asked to enter a 9 digit account number and a 6 digit PIN to login and hear their balances. CALLMaster will read the AccountInformation table from the SampleApps SQL database to get this information.
- Option 2: Caller will hear pre-recorded message on related information.
- Option 3: Caller will be directed to a mailbox to leave a message

This Sample Application is available for download. Use link at the end of this walkthrough to download and install sample application.

### Walkthrough Overview

In this walkthrough, you will:

1. Set-Up a mailbox;
2. Create Audiotex Records/Modules and Associated Scripts;
3. Define entry in the IVR Application List for the new custom IVR;
4. Set Ports.

You will learn to do the following:

- Configure CALLMaster to answer in Audiotex mode;
- Get input using the GetDTMF command;
- Use CALLBasic commands SayNumber as money, FloatToString to convert floating number to string, GetFloat to convert string to floating number;
- Use CALLBasic database commands to Read a record;
- Send caller to another Audiotex Module using the Next Module Audiotex Module Action;
- Make Audiotex recordings;
- Handle caller hang up and call failure.

### Prerequisites and Assumptions

- CALLMaster software installed and running with the Dialogic board installed and configured. For more information, see [Installation and Setup](#).
- A working knowledge of [CALLMaster Manager](#). Specifically how to add, delete and edit records.
- Phone lines enabled and connected to CALLMaster.
- Email activated and tested. See [Activating Email](#).
- Attach the SampleApps.mdf file from the CALLMaster\Data folder to your SQL Server. This database has 3 records:

RecID (int)	AccountNumber (nvarchar(10))	AccountPIN (nvarchar(10))	BalanceAm t (money)	NextPayAm t (money)	NextPayDat e (datetime)
1	123456789	123456	1245.75	125.00	10/10/2009
2	222222222	222222	20222.47	2000.63	1/15/2010
3	333333333	333333	3573.48	0.00	10/1/2009

### Set-up Mailbox Record

We need to create a mailbox record for the caller to leave a message. Set the **Status** to User, and enable **Allow Login During Greeting** . Add any other optional information as desired.

By setting the record to [Allow Login During Greeting](#), the mailbox owner will be able to log in while the greeting is played for [message management](#).

If the mailbox owner wishes to receive email notification of messages or faxes received in their mailbox, enter their email address in the **Email** field.

Note: Keep Mailbox record 0000 with the default settings as CALLMaster requires it. You should change the password for this record to maintain security.

Open the **CALLMaster Manager**. Click on **Manage**, then select **Mailboxes**. Let's add the record below:

Mailbox Records					
Mailbox #	Password	Last Name	Email	Box Status	Allow Log in During Greeting
1111	1111	MB 1111	1111@txt.att.net	User	Yes

## Create Audiotex Records/Modules and Associated Scripts

As all calls will be starting in Audiotex mode in this walkthrough, the Audiotex Modules will determine the call flow.

In the Audiotex set-up, the caller will be given the option to hear their account information by entering a 9 digit account number and a 6 digit PIN. We will read the AccountInformation table for this information. The caller will also have the option to hear general information, or to leave a message.

We will need to create several Audiotex records as shown below. The Label field provides a brief description of each module.

Audiotex Records																		
Mod ule ID	Label	Wri te La bel  to Lo g Fil e	Mod ule Acti on	Next Mod ule ID	Fail Mod ule ID	Respo nse Lengt h	Mail box	Script File to Run (Must include full path name) (c:\Program Files\CALLMaster\Script s\SampleApp-GetAc ctInfo\)	Touch Tone Table									
									1	2	3	4	5	6	7	8	9	0
100	Start call and Initialize values	Yes	Next Mod ule	1000	9900	0000		InitializeVal.bas										
1000	Give caller options (Press 1 for Account	Yes	Touc h Tone Tabl	9999	9900	1			11 00	12 00	13 00	11 90	11 90	11 90	11 90	11 90	11 90	11 90

	Information, 2 for general information or 3 to leave a message)		e																
1090	Invalid input.	Yes	Next Module	1091	9900	0		CheckTryCount.bas											
1091	Try again.	Yes	Next Module	1000	9900	0													
1100	Get 9 digit account number	Yes	Next Module	1110	9900	0		GetDTMF-Acct.bas											
1110	Get 6 digit pin	Yes	Next Module	1120	9900	0		GetDTMF-PIN.bas											
1120	Read AccountInformation table for matching account and PIN. If found return 1130 else return 1194	Yes	Next Module	9900	9900	0		ReadAcctInfo.bas											
1130	Play message "Your balance as of ..." and say today's date	Yes	Next Module	1135	9900	0		SayDateToday.bas											
1135	Play message "is.." and say balance amount as money	Yes	Next Module	1140	9900	0		SayAcctBal.bas											
1140	Play	Yes	Next	1145	9900	0		SayNextPayAmt.b											

	message "Your next payment amount of .." and say NextPayAmt as money	s	Module					as										
1145	Play message "is due on..." and say NextPayDate	Yes	Next Module	1150	9900	0		SayNextPayDate.bas										
1150	Give option to hear balance again "Press 1 to hear balance again, 2 to return to the main menu or any other key to end the call"	Yes	Touch Tone Table	9999	9900	1			1130	1000	9999	9999	9999	9999	9999	9999	9999	9999
1190	Play error message "Account number must be 9 digits."	Yes	Next Module	1191	9900	0		CheckTryCount.bas										
1191	Please try again	Yes	Next Module	1100	9900	0												
1192	Play error message "PIN must be 6 digits."	Yes	Next Module	1193	9900	0		CheckTryCount.bas										
1193	Please try again	Yes	Next Module	1110	9900	0												
1194	Record not found.	Yes	Next Mod	1195	9900	0		CheckTryCount.bas										

	Account or Pin mismatch.		ule																
1195	Record not found. Give option to try again "Press 1 to try again or hang-up to exit".	Yes	Touch Tone Table	9999	9900	1			1100	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999
1200	Play general information	Yes	Next Module	1210	9900	0													
1210	Give option to hear balance again. Press 1 to hear again, 2 to return to the main menu or any other key to end the call.	Yes	Touch Tone Table	9999	9900	1			1200	1000	9999	9999	9999	9999	9999	9999	9999	9999	9999
1300	Send caller to Mailbox	Yes	Mail box	1310	9900	0	1111												
1310	Press 1 to return to the main menu or any other key to end the call.	Yes	Touch Tone Table	9999	9900	1			1000	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999
9900	Call Failure Detected	Yes	Next Module	9999	9999	0		SetCallFailure.bas											
9910	Caller Hang-up Detected	Yes	Next Module	9999	9999	0		SetCallerHangup.bas											
9920	Maximum tries attempted	Yes	Next Module	9999	9999	0		SetMaxAttempts.bas											



9999	Last Module - End call	Yes	Last Mod ule																	
------	------------------------------	-----	--------------------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

#### Audiotex Module Description

- Module 100 is the starting module. We will use this module to initialize variables using script InitializeVal.bas. In particular we want to fix the allowed Input Length (intAcctNoInputLen, intPINInputLen) for the Account number and the PIN. We also want to define a variable (strSAConnection) for the connection string required to read the SampleApps SQL database. You will need to change the connection string for your environment settings. We will set the system variable lctrap to an Audiotex module to send caller if they hang-up. See sample code below.
- Module 1000 gives caller options via a Touch Tone Table requiring a 1 digit response. Module 1090 and 1091 will give the caller 3 attempts to input the correct options.
- Modules 1100 and 1110 are used to get the account number and PIN inputs using the GetDTMF command. This scripts use the StringLength command to make sure the caller has entered the required number of digits. If they have not then the call will be forwarded to module 1190 or 1192 for error processing.
- Module 1120 reads the AccountInformation table for matching AccountNumber and AccountPIN. If matching record is not found then Module 1190 is used to notify the caller that their input is not valid and they are given the option to press 1 to try again or exit.
- Modules 1130, 1135, 1140 and 1145 will play account information to the caller.
- Module 1150 gives caller the option to hear the account information again, or return to the main menu.
- Module 1200 plays the Audiotex module recording as the general information message.
- Module 1210 gives caller the option to hear the information again, or return to the main menu.
- Module 1300 sends caller to Mailbox 1111 to leave a message. On the return from leaving a message, module 1310 gives caller the option to return to the main menu.
- Module 9920 is used to notify caller that they have reached the maximum attempts allowed for a valid input. The script associated with this module can be used to set flags and route processing as required.
- Modules 9900 and 9910 are required to handle call failure and capture caller hang-up respectively. Both these modules can be used to set flags and route processing as required.

Based on the Audiotex modules shown above, we will need several scripts. Let's make a Scripts sub-folder (Program Files\CALLMaster\Scripts\SampleApp-GetAcctInfo folder) to save the custom scripts.

Create the scripts first. Open Notepad and create scripts following code examples below. Remember to save them as .bas type files in the Program Files\CALLMaster\Scripts\SampleApp-GetAcctInfo folder.

#### CALLBasic Script - InitializeVal.bas

[Copy to Clipboard](#)

#### Program

\*\*\*\*\*

' Script: InitializeVal.bas

' Function: This script is used to initialize values

\*\*\*\*\*

strYes = "Yes"

strNo = "No"

strZero = "0"

intYes = 1

intNo = 0

intOne = 1

intZero = 0

strCallFailure = strNo

strCallerHangup = strNo

```

strMaxAttempts = strNo

' Set retry counter and maximum attempts allowed
intTryCount = 1
intMaxAttemptsAllowed = 3

' Set input lengths for use by the GetDTMF command
intAcctNoInputLen = 9
intPINInputLen = 6

'SampleApps db
SADBOpen = intNo

' AccountInformation table values
AlMatchExists = intNo
Al-strAccountNumber = ""
Al-strAccountPIN = ""
GetFloat Al-fltBalanceAmt strZero 0
GetFloat Al-fltNextPayAmt strZero 0
Al-dteNextPayDate = $today

' Set connection string to read SampleApps SQL database
strSAConnection = "Provider=SQLOLEDB;Server=<servername>;DATABASE=SampleApps;UID=;PWD=;Integrated Se

' Set the module to return to if caller hangs up
lctrap = 9910

' Convert system variable $channel from integer to string to be used in concatenated print statement.
IntegerToString strchannel $channel

pline = "***** For channel=" & strchannel
pline = pline & " Started initialization process "
PrintNL pline

```

EndProgram

[Copy to Clipboard](#)

## CALLBasic Script - CheckTryCount.bas

### Program

```

*****
' Script: CheckTryCount.bas
' Function: This script checks the number of invalid input attempts.
*****

' If the caller has reached the maximum attempts allowed then the call is routed to module 9920.
' If not then they are returned back to module 1000 to retry input based on Next Module ID.
If intTryCount > intMaxAttemptsAllowed Then
    Return 9920
Else
    intTryCount = intTryCount + 1
EndIf

' Convert system variable intTryCount from integer to string to be used in concatenated print statement.
IntegerToString strTryCount intTryCount

```

```

pline = "***** For channel=" & strchannel
pline = pline & " TryCount = "
pline = pline & strTryCount
PrintNL pline

```

EndProgram

[Copy to Clipboard](#)

## CALLBasic Script - GetDTMF-Acct.bas

Program

```

*****
' Script: GetDTMF-Acct.bas
' Function: This script gets the account number from the caller.
*****

' Set up parameters for GetDTMF command
strNoInput = ""
strAcctNoInput = ""
tmask = ""
ndigits = intAcctNoInputLen
timeout = 15

GetDTMF strNoInput tmask ndigits timeout

' Get length of input
StringLength intlen strNoInput

pline = "***** For channel=" & strchannel
pline = pline & " Account No entered is "
pline = pline & strNoInput
PrintNL pline

' Check caller input. The caller must enter the number of digits defined by variable intMBInputLen.
' If valid input then save number entered

If intlen <> intAcctNoInputLen Then
    pline = "***** For channel=" & strchannel
    pline = pline & " Invalid Account No entered. Return to 1190 for error and retry"
    PrintNL pline
    Return 1190
Else
    intTryCount = 0
    strAcctNoInput = strNoInput
EndIf

```

EndProgram

[Copy to Clipboard](#)

## CALLBasic Script - GetDTMF-PIN.bas

Program

```

*****
' Script: GetDTMF-PIN.bas
' Function: This script gets the account number from the caller.
*****

```

```

' Set up parameters for GetDTMF command
strNoInput = ""
strPINInput = ""
tmask = ""
ndigits = intPINInputLen
timeout = 15

GetDTMF strNoInput tmask ndigits timeout

' Get length of input
StringLength intlen strNoInput

pline = "***** For channel=" & strchannel
pline = pline & " PIN entered is "
pline = pline & strNoInput
PrintNL pline

' Check caller input. The caller must enter the number of digits defined by variable intMBInputLen.
' If valid input then save number entered

If intlen <> intPINInputLen Then
    pline = "***** For channel=" & strchannel
    pline = pline & " Invalid PIN entered. Return to 1192 for error and retry"
    PrintNL pline
    Return 1192
Else
    intTryCount = 0
    strPINInput = strNoInput
EndIf

EndProgram

```

[Copy to Clipboard](#)

#### CALLBasic Script - ReadAcctInfo.bas

##### Program

```

*****
' Script: ReadAcctInfo.bas
' Function: This script reads the AccountInformation table.
*****

' Set new command and open connection to SampleApps database.
Set AlvarCommand = New Command

AlvarCommand.ActiveConnection strSAConnection

If $DBError <> intZero Then
    AlvarSet.Close
    Set AlvarSet = Nothing
    pline = "***** For channel=" & strchannel
    pline = pline & " Error Connecting db. "
    PrintNL pline
    Return 9900
Else
EndIf

' Create a new recordset

```

```
Set AlvarSet = New Recordset
```

```
strSQL = "SELECT BalanceAmt, NextPayAmt, NextPayDate FROM AccountInformation WHERE AccountNumber = " & strAccountNumber  
strSQL = strSQL & " and AccountPIN = " & strPINInput  
strSQL = strSQL & ""
```

```
pline = "For channel " & strchannel  
pline = pline & strSQL  
PrintNL pline
```

```
' Open SADB database and find record  
Set AlvarSet = AlvarCommand.Execute strSQL
```

```
If $DBError <> intZero Then  
    AlvarSet.Close  
    Set AlvarSet = Nothing  
    pline = "***** For channel=" & strchannel  
    pline = pline & " Error reading AI db. "  
    PrintNL pline  
    Return 9900  
Else  
EndIf
```

```
Set AlvarFields = AlvarSet.Fields  
SADBOpen = intYes
```

```
If AlvarSet.EOF = 0 Then  
    Set AlvarF1 = AlvarFields.Field 1  
    Set AlvarF2 = AlvarFields.Field 2  
    Set AlvarF3 = AlvarFields.Field 3
```

```
AI-strAccountNumber = strAcctNoInput  
AI-strAccountPIN = strPINInput  
AI-fltBalanceAmt = AlvarF1.Value  
AI-fltNextPayAmt = AlvarF2.Value  
AI-dteNextPayDate = AlvarF3.Value
```

```
' Free field variables  
Set AlvarF1 = Nothing  
Set AlvarF2 = Nothing  
Set AlvarF3 = Nothing
```

```
' Set Switches  
AIMatchExists = intYes
```

```
Else  
    AIMatchExists = intNo  
EndIf
```

```
' Free AIDB varSet, varFields, and close database  
' Destroy Fields Object  
Set AlvarFields = Nothing
```

```
' Close recordset  
AlvarSet.Close
```

```
' Free memory  
Set AlvarSet = Nothing
```

```

' clear command
AlvarCommand.CloseConnection
Set AlvarCommand = Nothing

' AIDB database closed
AIDBOpen = intNo

' If found return 1130 else return 1194
If AIMatchExists = intYes Then
    'Print information to log file for reference and debugging
    pline = "***** For channel=" & strchannel
    pline = pline & " AccountInformation record found for AccountNumber = " & AI-strAccountNumber
    PrintNL pline
    Return 1130
Else
    pline = "***** For channel=" & strchannel
    pline = pline & " AccountInformation record NOT found for AccountNumber = " & strAcctNoInput
    PrintNL pline
    Return 1194
EndIf

```

EndProgram

[Copy to Clipboard](#)

#### CALLBasic Script - SayDateToday.bas

Program

\*\*\*\*\*

```

' Script: SayDateToday.bas
' Function: This script says today's date.

```

\*\*\*\*\*

SayDate \$today

```

'Print information to log file for reference and debugging
pline = "***** For channel=" & strchannel
pline = pline & "***** Say date today *****"
PrintNL pline

```

EndProgram

[Copy to Clipboard](#)

#### CALLBasic Script - SayAcctBal.bas

Program

\*\*\*\*\*

```

' Script: SayAcctBal.bas
' Function: This script says the account balance.

```

\*\*\*\*\*

```

' Convert balance from float to string format
FloatToString strBalanceAmt AI-fltBalanceAmt

```

```

' To say the balance as money, set the variable saymoney to 1, use SayNumber then reset saymoney to 0
saymoney = 1
SayNumber strBalanceAmt

```

```
saymoney = 0
```

```
'Print information to log file for reference and debugging  
pline = "***** For channel=" & strchannel  
pline = pline & "***** Said balance amount as money *****"  
PrintNL pline
```

```
EndProgram
```

#### CALLBasic Script - SayNextPayAmt.bas

[Copy to Clipboard](#)

```
Program
```

```
!*****
```

```
' Script: SayNextPayAmt.bas  
' Function: This script says the next payment amount.
```

```
!*****
```

```
' Convert balance from float to string format  
FloatToString strNextPayAmt AI-fltNextPayAmt
```

```
' To say the next pay amount as money, set the variable saymoney to 1, use SayNumber then reset saymoney to 0  
saymoney = 1  
SayNumber strNextPayAmt  
saymoney = 0
```

```
'Print information to log file for reference and debugging  
pline = "***** For channel=" & strchannel  
pline = pline & "***** Said next payment amount as money *****"  
PrintNL pline
```

```
EndProgram
```

#### CALLBasic Script - SayNextPayDate.bas

[Copy to Clipboard](#)

```
Program
```

```
!*****
```

```
' Script: SayNextPayDate.bas  
' Function: This script says the next payment date.
```

```
!*****
```

```
' Use the SayDate command as variable AI-dteNextPayDate is a date/time data type  
SayDate AI-dteNextPayDate
```

```
'Print information to log file for reference and debugging  
pline = "***** For channel=" & strchannel  
pline = pline & "***** Said next payment date *****"  
PrintNL pline
```

```
EndProgram
```

[Copy to Clipboard](#)

### CALLBasic Script - SetCallFailure.bas

#### Program

```
*****
' Script: SetCallFailure.bas
' Function: This script sets the strCallFailure variable to yes and prints information in the call log file.
*****
```

```
strCallFailure = strYes
```

```
'Print information to log file for reference and debugging
pline = "***** For channel=" & strchannel
pline = pline & "***** Call Failure. *****"
PrintNL pline
```

```
EndProgram
```

[Copy to Clipboard](#)

### CALLBasic Script - SetCallerHangup.bas

#### Program

```
*****
' Script: SetCallerHangup.bas
' Function: This script sets the strCallFailure variable to yes and prints information in the call log file.
*****
```

```
strCallerHangup = strYes
```

```
'Print information to log file for reference and debugging
pline = "***** For channel=" & strchannel
pline = pline & "***** Caller Hangup. *****"
PrintNL pline
```

```
EndProgram
```

[Copy to Clipboard](#)

### CALLBasic Script - SetMaxAttempts.bas

#### Program

```
*****
' Script: SetMaxAttempts.bas
' Function: This script sets the strMaxAttempts variable to yes and prints information in the call log file.
*****
```

```
strMaxAttempts = strYes
```

```
'Print information to log file for reference and debugging
pline = "***** For channel=" & strchannel
pline = pline & "***** Maximum Attempts reached to enter valid input *****"
PrintNL pline
```

```
EndProgram
```

Now you can create the Audiotex modules and reference the scripts above. Open the **CALLMaster Manager**, click on **Manage**, then select **Audiotex**. Let's add the Audiotex records shown above.



## Define IVR Application List Entry

We need to define the custom IVR in the IVR Application List and set it to Audiotex Start Module 100. Open CALLMaster Manager and select Manage | IVR Application List. Let's add a new record as follows:

IVR Application List	
Audiotex Start Module	Application Name
100	Account Information Retrieval Sample

## Set Ports

As a last step, we need set the port(s) to the new IVR Application entry created above. Make sure the Audiotex Start Module field (read only) correctly points to module 100.

Set the Port Greeting File IDs to the main greeting for all calls if you have recorded one. If you do not want a main greeting then set the Greeting File IDs to zero. Leave all other fields to the default values.

If you are working with a demo license or have only one line connected, then make sure the connected port is set as specified below. All other ports should be set to Application 'Disabled'.

Port Records			
Port #	Application	Audiotex Start Module	Port Greeting File IDs (Morning, Afternoon, Evening, Closed, Holidays)
1	Account Information Retrieval Sample	100	0, 0, 0, 0, 0
2	Account Information Retrieval Sample	100	0, 0, 0, 0, 0
3	Account Information Retrieval Sample	100	0, 0, 0, 0, 0
4	Account Information Retrieval Sample	100	0, 0, 0, 0, 0

## Test the custom application

Make sure you have set the SampleApps database connection string variable strSACConnection in the InitializeVal.bas script for your environment.

Save all your work.

Stop CALLMaster service.

Make Audiotex Recordings:

- Set Port 1 to Application Voice Mail.

- Start CALLMaster service.
- Make a call using Port 1.
- [Log in](#) as the Administrator using the [Voice Mail Main Menu](#) and make recordings for the Audiotex modules below.

<b>Audiotex Module ID</b>	<b>Proposed Recording</b>
1000	Press 1 for Account Information, 2 for general information or 3 to leave a message.
1090	Invalid option.
1091	Please try again
1100	Please enter your 9 digit account number.
1110	Please enter your 6 digit PIN
1130	Your balance as of ...
1135	is...
1140	Your next payment amount of ..
1145	is due on...
1150	Press 1 to hear balance again, 2 to return to the main menu or any other key to end the call
1190	Account number must be 9 digits.
1191	Please try again
1192	PIN must be 6 digits.
1193	Please try again
1194	Account number or PIN not valid.

1195	Press 1 to try again or hang-up to exit
1200	General information message
1210	Press 1 to hear the information again, 2 to return to the main menu or any other key to end the call
1310	Press 1 to return to the main menu or any other key to end the call.
9900	We are unable to process your call. Please contact the system administrator for further assistance.
9920	You have reached the maximum attempts allowed for a valid input. Please contact the system administrator for further assistance.

Reset Port 1 to Application Audiotex. Set the Port Greeting File IDs to zeros.

Stop and Start CALLMaster service.

Make a phone call to test the application.

Tip: See [Walkthrough: Change Application Without Restarting CALLMaster Service](#) to set port to Voice Mail to make recording without stopping and restarting the CALLMaster service.

The owner of Mailbox 1111 can use option 3 (leave message), then use the [Allow Login During Greeting](#) functionality to log in while the greeting is played for [message management](#) (check messages left by callers or to make mailbox recordings).

To download and install, see [Sample Applications](#).

## Walkthrough: Make Outbound Calls for an Appointment Reminder with Option to Reschedule

In this walkthrough we will make outbound calls using CALLMaster Out Dial functionality to play a reminder message with the date and time of the appointment as imported for each number and give caller the options to hear the message again or to transfer to agent to reschedule.

- If the call connect type is human, the called party will be played a message with the date and time of the appointment and be given the option to transfer to agent to reschedule.
- If the call connect type is answering machine, the called party will be played a message with the date and time of the appointment.
- If the call connect type is fax, the call will terminate.

This Sample Application is available for download. Use link at the end of this walkthrough to download and install sample application.

### Walkthrough Overview

In this walkthrough, you will:

1. Set System Parameters;
2. Set Dialogic Parameters;
3. Verify Attendant Parameters required to transfer a call, or drop call in case of a busy or no answer;
4. Set-up Attendant Record for caller reschedule option;
5. Create Audiotex Records and Associated Scripts;
6. Define entry in the IVR Application List for the new custom IVR;
7. Set Ports.

During this walkthrough, you will learn to do the following:

- Configure CALLMaster to answer in Audiotex mode;
- Get input using the Audiotex Module Action 'Touch Tone Table';
- Get value of system variables from the OutcallSchedule record;
- Distinguish between a human, answering machine and fax connection;
- Use CALLBasic commands including SayDate and SayTime;
- Make Audiotex recordings;
- Handle caller hang up and call failure.

### Prerequisites and Assumptions

- CALLMaster software installed and running with the Dialogic board installed and configured. For more information, see [Installation and Setup](#);
- A working knowledge of [CALLMaster Manager](#). Specifically how to add, delete and edit records;
- Phone lines enabled and connected to CALLMaster;
- Activate Out Dial process by adding flag "ODS" to the System Parameters 'System Setting' field.
- Numbers to be called imported to the CALLMaster OutcallSchedule table, with required fields:- Dialing Prefix, Area Code and Phone Number and UserDateTime (Appointment Date and Time). All other fields could have the default values or you can upload custom values as part of the import process. See [OutcallSchedule Record](#) for details.

### Set System Parameters

As we are using Out Dial, we need to make sure that the System Parameters - Local Dialing Prefix, Long Distance Dialing Prefix, System's Area Code and System's Phone Number have the correct values for your environment. We also need to set the time that CALLMaster can 'Start Calling at' and needs to 'Stop Calling at'.

As we want CALLMaster to continuously attempt outbound calls and reduce the sleep default time from 30 seconds to 3 seconds, we will add flags 'OCNV' and 'OSST3' in the System Flags field.

As we want to leave a message if call receives an answering machine connect, we need to adjust the Parameter Record Max Silence (see [Out Dial - Human/Answering Machine/Fax Detection](#) for details).

Open the **CALLMaster Manager**. Click on **Parameters**, then select **System**. Double click on each of the parameters to change their values as needed. See [System Parameters](#) for details.

For this walkthrough we will assume the following values:

- **Local Dialing Prefix** - set to 9.
- **Long Distance Dialing Prefix** - set to 9,1,.
- **System's Area Code** - set to 914.
- **System's Phone Number** - set to 5555555.
- **Start Calling At** - set to 09:00:00
- **Stop Calling At** - set to 17:00:00
- **Record Max Silence** - set to 3.
- **System Flags** - set to OCNV OSST3 ODS

## Set Dialogic Parameters

As we are using Out Dial and want to detect Human, Answering Machine and Fax connect, we need to adjust the several Dialogic Parameters (see [Out Dial - Human/Answering Machine/Fax Detection](#) for details).

Open the **CALLMaster Manager**. Click on **Parameters**, then select **Dialogic**. Double click on each of the parameters to change their values as needed.

For this walkthrough we will assume the following values:

- **NBRDNA** - set to '8';
- **HEDGE** - set to 'Detect on TRAILING Edge';
- **INTFLG** - set to '7'
- **PAMDFAILTIME** - set to '10'

## Verify Attendant Parameters.

To transfer calls for appointment reschedule option, you must know the requirements of your phone lines or PBX. In particular, you need to know the transfer string to initiate a transfer, and the pull back strings to drop a call in case of busy or no answer. Since the caller can enter a variable length extension ID, you need to set the Maximum Extension Length parameter.

Open the **CALLMaster Manager**. Click on **Parameters**, then select **Attendant**. Confirm values for all the parameters on the list. Double click on the parameter to change the value if required. For this walkthrough, we will assume the values below. Save and close the Attendant Parameters window after making any changes.

Attendant Parameters				
Allow Any Transfer	Drop 'Busy' Call	Drop 'No Answer' Call	Max Extension Length	Transfer String
No	&	&	4	&

## Set-up Attendant Records

As one of the options offered to the caller is to reschedule appointment, we need to setup an Attendant record to transfer to an agent. We will create an Attendant record set the **Time Of Day** to All Day, and the **Type of Transfer** to Transfer.

Since the **Type of Transfer** is set to Transfer, the **Failed Transfer Action** fields do not apply. If you set the **Type of Transfer** to Supervised, requiring CALLMaster to monitor call transfer status, then you must set the **Failed Transfer Action** fields as desired.

See [Managing Attendant](#) for details.

Open the **CALLMaster Manager**. Click on **Manage**, then select **Attendant**. Let's add 4 records in this walkthrough as follows:

Attendant Records
-------------------

Attendant ID	Time of Day	Type of Transfer	Number to Dial	Mailbox
1	All Day	Transfer	9,1,2125551111	

### Create Audiotex Records and Associated Scripts

As all calls will be starting in Audiotex mode in this walkthrough, the Audiotex Modules will determine the call flow.

In the Audiotex set-up, the called party will be played a message and given the option to "Press 1 to hear message again or 2 to reschedule".

NOTE: the Audiotex application will only get processing control if the call results in a connect (i.e. \$CR = 10). If the call does not result in a connect, then the call is reset based on the busy/no answer/no confirm retry values from the OutcallSchedule record.

We will need to create several Audiotex records as shown below. The Label field provides a brief description of each module.

Audiotex Records																	
Module ID	Label	Write Label to Log File	Module Action	Next Module ID	Fail Module ID	Response Length	Script File to Run (Must include full path name) (c:\Program Files\CALLMASTER\Scripts\SampleApp-AppletRem\)	Touch Tone Table									
								1	2	3	4	5	6	7	8	9	0
100	Start call and Initialize values	Yes	Next Module	110	9900	0	InitializeVal.bas										
110	Perform call analysis to deter	Yes	Next Module	9999	9900	0	OD-CallAnalysis.bas										

	mine if it is a human, answering machine, or fax response.																
200	Human Connect Detected.  Play Audiotext Module recording.	Yes	Next Module	210	9900	0											
210	Say Appointment Date & Time	Yes	Next Module	220	9900	0	SayAppptDateTime.bas										
220	Provide options "1 to hear message again or 2 to reschedule" and get caller response	Yes	Touch Tone Table	9920	9900	1		200	230	290	290	290	290	290	290	290	290

230	Transfer to Attendant ID 1	Yes	Attendant ID	9999	9900		ReturnAttID.bas										
290	Invalid input.	Yes	Next Module	291	9900	0	CheckTryCount.bas										
291	Try again.	Yes	Next Module	220	9900	0											
300	Answering Machine Connect Detected.  Play Audiotex Module recording.	Yes	Next Module	9999	9999												
310	Say Appointment Date & Time	Yes	Next Module	9999	9900	0	SayAppointmentTime.bas										
400	Fax/Other Connect Detected. End Call	Yes	Next Module	9999	9999												
9900	Call Failure Detected	Yes	Next Module	9999	9999	0	SetCallFailure.bas										



	cted																
9910	Caller Hang-up Detected	Yes	Next Module	9999	9999	0	SetCallerHangup.bas										
9920	Maximum tries attempted	Yes	Next Module	9999	9999	0	SetMaxAttempts.bas										
9999	Last Module - End call	Yes	Last Module														

#### Auditex Module Description

- Module 100 is the starting module. We will use this module to initialize variables using script InitializeVal.bas. In particular we want to set local variables to the Out Dial system variables returned by CALLMaster. We will set the system variable lctrap to an Auditex module ID to send caller if they hang-up.
- Module 110 will perform call analysis to determine if it is a human, answering machine, or fax response.
  - If human connect detected, call control will return to Module 200 to play a message;
  - If answering machine connect detected, then a blank recording is made for up to 60 seconds or it detects continuous silence for duration of Record Max Silence System Parameter. In our walkthrough, CALLMaster will require 3 seconds of silence after the answering machine greeting before returning call control to module 300 to play the message;
  - If fax connect detected, call control will return to Module 400 to fax document;
  - If none of the above, call control will return to 500.
- Module 200 will just play the Auditex message and go to module 210 (next module ID).
- Module 210 uses the SayDate and SayTime CALLBasic command to say the appointment date and time from the UserDateTime field in the OutcallSchedule record.
- Module 220 has 'Touch Tone Table' Module Action requiring a 1 digit input (response length set to 1). This type of module is used to offer caller options and get a corresponding touch tone input. This module offers the caller 2 options. The caller will be directed to module 200 if they press 1 to hear the message again or to module 230 if they enter 2 to reschedule. If they enter any other touch tone key, they will be directed to module 290. As a 'Touch Tone Table' Module Action, at the end of the module recording, CALLMaster will play prompt 245 "press \* to repeat", giving the caller 3 attempts to enter any touch tone key (does not have to be a valid input). If you want to give the caller 3 attempts to press any key but do not want to play this prompt, then you can replace the associated wav file with a null recording (see [Replacing Recordings](#)). If you do not want to give the caller 3 attempts to enter any key, use the [NTR](#) code in the Auditex record Label field.
- Module 230 sends caller to the Attendant ID using the Return CALLBasic command.
- Module 290 and 291 is to notify the caller that they have entered an invalid option. The associated script, CheckTryCount.bas, gives the caller 3 attempts to enter a valid option. If the caller has not entered a valid option after 3 attempts, processing will be forwarded to module 9920.
- Module 300 will play the Auditex greeting recorded for Answering Machine connection.
- Module 400 will terminate call.
- Module 9920 is used to notify caller that they have reached the maximum attempts allowed for a valid

input. The script associated with this module can be used to set flags and route processing as required.

- Modules 9900 and 9910 are required to handle call failure and capture caller hang-up respectively. Both these modules can be used to set flags and route processing as required.

Based on the Audiotex modules shown above, we will need several scripts. Let's make a Scripts sub-folder (Program Files\CALLMaster\Scripts\SampleApp-ApptRem folder) to save the custom scripts.

Create the scripts first. Open Notepad and create scripts following code examples below. Remember to save them as .bas type files in the Program Files\CALLMaster\Scripts\SampleApp-ApptRem folder.

#### CALLBasic Script - InitializeVal.bas

[Copy to Clipboard](#)

##### Program

\*\*\*\*\*

' Script: InitializeVal.bas

' Function: This script is used to initialize values

\*\*\*\*\*

strYes = "Yes"

strNo = "No"

intYes = 1

intNo = 0

strCallFailure = strNo

strCallerHangup = strNo

strMaxAttempts = strNo

intOne = 1

' Set retry counter and maximum attempts allowed

intTryCount = 1

intMaxAttemptsAllowed = 3

strRescheduleAttID = "1"

' Get value of OutcallSchedule system variables and assign to local variables.

' If variable is an integer then convert to string local variable

OD-intOutDialID = \$outdialID

IntegerToString OD-strOutDialID \$outdialID

OD-intResult = \$scr

IntegerToString OD-strResult \$scr

OD-intConnection = \$contype

IntegerToString OD-strConnection \$outdialID

OD-strDialPrefix = \$dialprefix

OD-strAreaCode = \$areacode

OD-strPhoneNum = \$phonenum

OD-dteApptDateTime = \$duserdatetime

' Set the module to return to if caller hangs up

lctrp = 9910

' Convert system variable \$channel from integer to string to be used in concatenated print statement.

IntegerToString strchannel \$channel

pline = "\*\*\*\*\* For channel=" & strchannel

pline = pline & " Started Out Dial for ID=" & OD-strOutDialID

```

pline = pline & " with OD-strDialPrefix=" & OD-strDialPrefix
pline = pline & "; OD-strAreaCode=" & OD-strAreaCode
pline = pline & "; OD-strPhoneNum=" & OD-strPhoneNum
PrintNL pline

```

EndProgram

## CALLBasic Script - OD-CallAnalysis.bas

[Copy to Clipboard](#)

Program

```

*****
' Script: OD-CallAnalysis.bas
' Function: This script checks the call connection type and returns based on the system variable $contype.
*****

' Set variables for recording if Answering Machine Connection
intPmt = 0
intRecTime = 60
strT = "0"
strA = ""

' If Call Result Connected
If OD-intResult = 10 Then
' Human Voice Connection
If OD-intConnection < 4 Then
Return 200
Else
'Answering machine
Record recfileid intPmt intRecTime strA strT e d
FileIDToName destname recfileid
FileDelete destname
Return 300
EndIf
Else
' Call Result Fax or other
Return 400
EndIf

```

EndProgram

## CALLBasic Script - SayApptDateTime.bas

[Copy to Clipboard](#)

Program

```

*****
' Script: SayApptDateTime.bas
' Function: This script says the Date and Time.
*****

SayDate OD-dteApptDateTime
SayTime OD-dteApptDateTime

```

EndProgram

### CALLBasic Script - CheckTryCount.bas

[Copy to Clipboard](#)

Program

```
*****
' Script: CheckTryCount.bas
' Function: This script checks the number of invalid input attempts.
*****

' If the caller has reached the maximum attempts allowed then the call is routed to module 9920.
' If not then they are returned back to Next Module ID.
If intTryCount > intMaxAttemptsAllowed Then
    Return 9920
Else
    intTryCount = intTryCount + 1
EndIf

' Convert system variable intTryCount from integer to string to be used in concatenated print statement.
IntegerToString strTryCount intTryCount

pline = "***** For channel=" & strchannel
pline = pline & " TryCount = "
pline = pline & strTryCount
PrintNL pline
```

EndProgram

### CALLBasic Script - ReturnAttID.bas

[Copy to Clipboard](#)

Program

```
*****
' Script: ReturnAttID.bas
' Function: This script uses the Return command with a variable.
'           When used with an Audiotex Module Action 'Attendant ID',
'           the caller will be forwarded to the Attendant whose value is in variable strRescheduleAttID.
*****
```

```
Return strRescheduleAttID
```

EndProgram

### CALLBasic Script - SetCallFailure.bas

[Copy to Clipboard](#)

Program

```
*****
' Script: SetCallFailure.bas
' Function: This script sets the strCallFailure variable to yes and prints information in the call log file.
*****
```

```
strCallFailure = strYes
```

```
'Print information to log file for reference and debugging
pline = "***** For channel=" & strchannel
pline = pline & "***** Call Failure. *****"
PrintNL pline
```

EndProgram

#### CALLBasic Script - SetCallerHangup.bas

[Copy to Clipboard](#)

Program

```
!*****
' Script: SetCallerHangup.bas
' Function: This script sets the strCallFailure variable to yes and prints information in the call log file.
!*****
```

```
strCallerHangup = strYes
```

```
'Print information to log file for reference and debugging
pline = "***** For channel=" & strchannel
pline = pline & "***** Caller Hangup. *****"
PrintNL pline
```

EndProgram

#### CALLBasic Script - SetMaxAttempts.bas

[Copy to Clipboard](#)

Program

```
!*****
' Script: SetMaxAttempts.bas
' Function: This script sets the strMaxAttempts variable to yes and prints information in the call log file.
!*****
```

```
strMaxAttempts = strYes
```

```
'Print information to log file for reference and debugging
pline = "***** For channel=" & strchannel
pline = pline & "***** Maximum Attempts reached to enter valid input *****"
PrintNL pline
```

EndProgram

Now you can create the Audiotex modules and reference the scripts above. Open the **CALLMaster Manager**, click on **Manage**, then select **Audiotex**. Let's add the Audiotex records shown above.

### Define IVR Application List Entry

We need to define the custom IVR in the IVR Application List and set it to Audiotex Start Module 100. Open CALLMaster Manager and select Manage | IVR Application List. Let's add a new record as follows:

IVR Application List	
Audiotex Start Module	Application Name

100	Appointment Reminder w/ Reschedule Option Sample
-----	--

## Set Ports

As a last step, we need set the port(s) to the new IVR Application entry created above. Make sure the Audiotex Start Module field (read only) correctly points to module 100.

Set the Port Greeting File IDs to the main greeting for all calls if you have recorded one. If you do not want a main greeting then set the Greeting File IDs to zero. Leave all other fields to the default values.

If you are working with a demo license or have only one line connected, then make sure the connected port is set as specified below. All other ports should be set to Application 'Disabled'.

Port Records				
Port #	Application	Audiotex Start Module	Port Type Outbound	Port Greeting File IDs (Morning, Afternoon, Evening, Closed, Holidays)
1	Appointment Reminder w/ Reschedule Option Sample	100	Yes	0, 0, 0, 0, 0
2	Appointment Reminder w/ Reschedule Option Sample	100	Yes	0, 0, 0, 0, 0
3	Appointment Reminder w/ Reschedule Option Sample	100	Yes	0, 0, 0, 0, 0
4	Appointment Reminder w/ Reschedule Option Sample	100	Yes	0, 0, 0, 0, 0

## Test the custom application

Save all your work.

Stop CALLMaster service.

Make Administrator Audiotex Recordings:

- Set Port 1 to Application Voice Mail.
- Start CALLMaster service.
- Make a call using Port 1.

- [Log in](#) as the Administrator using the Voice Mail Main Menu and make recordings for the Audiotex modules below.

Audiotex Module ID	Proposed Recording
200	Recording for Human Connection - e.g. Hi this is Dr. Joe's office calling to remind you of your appointment on...
220	Press 1 if you want to hear the message again or 2 to reschedule
290	Invalid option.
291	Please try again
300	Recording for Answering Machine Connection - e.g. Hi this is Dr. Joe's office calling to remind you of your appointment on...
9900	We are unable to process your call. Please contact the system administrator for further assistance.
9920	You have reached the maximum attempts allowed for a valid input. Please contact the system administrator for further assistance.

Reset Port 1 to Application Audiotex. Set the Port Greeting File IDs to zeros.

#### [Stop and Start CALLMaster service.](#)

Make a phone call to test the application.

Tip: See [Walkthrough: Change Application Without Restarting CALLMaster Service](#) to set port to Voice Mail to make recording without stopping and restarting the CALLMaster service.

To download and install, see [Sample Applications](#).

## Walkthrough: Make Outbound Calls to Broadcast Message and Fax Document

In this walkthrough we will make outbound calls using CALLMaster Out Dial functionality to play a message or fax a document depending on the call connect type.

- If the call connect type is human or answering machine, the pre-recorded message will be played.
- If the call connect type is fax, a pre-defined document will be faxed to the called number.

This Sample Application is available for download. Use link at the end of this walkthrough to download and install sample application.

### Walkthrough Overview

In this walkthrough, you will:

1. Set System Parameters;
2. Set Dialogic Parameters;
3. Create Fax Record;
4. Create Audiotex Records and Associated Scripts;
5. Define entry in the IVR Application List for the new custom IVR;
6. Set Ports.

During this walkthrough, you will learn to do the following:

- Configure CALLMaster to answer in Audiotex mode;
- Distinguish between a human, answering machine and fax connection;
- Use CALLBasic commands including FaxSend;
- Make Audiotex recordings;
- Handle caller hang up and call failure.

### Prerequisites and Assumptions

- CALLMaster software installed and running with the Dialogic board installed and configured. For more information, see [Installation and Setup](#);
- A working knowledge of [CALLMaster Manager](#). Specifically how to add, delete and edit records;
- Phone lines enabled and connected to CALLMaster;
- Activate Out Dial process by adding flag "ODS" to the System Parameters 'System Setting' field.
- Numbers to be called imported to the CALLMaster OutcallSchedule table, with required fields - Dialing Prefix, Area Code and Phone Number. All other fields could have the default values or you can upload custom values as part of the import process. See [OutcallSchedule Record](#) for details.

### Set System Parameters

As we are using Out Dial, we need to make sure that the System Parameters - Local Dialing Prefix, Long Distance Dialing Prefix, System's Area Code and System's Phone Number have the correct values for your environment. We also need to set the time that CALLMaster can 'Start Calling at' and needs to 'Stop Calling at'.

As we want CALLMaster to continuously attempt outbound calls and reduce the sleep default time from 30 seconds to 3 seconds, we will add flags 'OCNV' and 'OSST3' in the System Flags field.

As we want to leave a message if call receives an answering machine connect, we need to adjust the Parameter Record Max Silence (see [Out Dial - Human/Answering Machine/Fax Detection](#) for details).

Open the **CALLMaster Manager**. Click on **Parameters**, then select **System**. Double click on each of the parameters to change their values as needed. See [System Parameters](#) for details.

For this walkthrough we will assume the following values:

- **Local Dialing Prefix** - set to 9.
- **Long Distance Dialing Prefix** - set to 9,1,.
- **System's Area Code** - set to 914.
- **System's Phone Number** - set to 5555555.
- **Start Calling At** - set to 09:00:00
- **Stop Calling At** - set to 17:00:00
- **Record Max Silence** - set to 3.
- **System Flags** - set to OCNV OSST3 ODS



## Set Dialogic Parameters

As we are using Out Dial and want to detect Human, Answering Machine and Fax connect, we need to adjust the several Dialogic Parameters (see [Out Dial - Human/Answering Machine/Fax Detection](#) for details).

Open the **CALLMaster Manager**. Click on **Parameters**, then select **Dialogic**. Double click on each of the parameters to change their values as needed.

For this walkthrough we will assume the following values:

- **NBRDNA** - set to '8';
- **HEDGE** - set to 'Detect on TRAILING Edge';
- **INTFLG** - set to '7'
- **PAMDFAILTIME** - set to '10'

## Define Global Tone for Fax

For the Dialogic board to detect a fax tone, we need to define the tone settings in the Global Tones table.

Open the **CALLMaster Manager**. Click on **Manage**, then select **Global Tones**. If there is an record for a Fax tone, then just verify the setting. If not then Add a record to this table with values as below.

Global Tones													
Tone Type	Characteristic	Detection Mode	Translation		Frequencies (Hz)				Cadence (10ms)				
			Translate to Char	Character	Tone 1 Freq	Tone 1 Dev	Tone 2 Freq	Tone 2 Dev	On Time	On Time Dev	Off Time	Off Time Dev	Minimum Repetition Count
Fax	Single Tone	Leading Edge	Yes	F	1100	50	0	0	0	0	0	0	1

## Create Fax Record

To fax a document, we need to create the document in the format required by the Dialogic board, and define the document in the Fax table.

The fax document needs to be in a Tiff-F format. See [Fax Overview](#) for details.

For this walkthrough, download the [Fax Test Document](#) as a sample document and save to a folder.

To define the document created above, open the **CALLMaster Manager**. Click on **Manage**, then select **Faxes**. Add record using the Fax Test Document saved as the File to Send.

Note the Fax ID of the record added for use with the FaxSend command.

If you wish to fax various fax document, you may want to consider either:

- using a field in the OutcallSchedule record as an indicator to the fax document number and then use this indicator in the FaxSend command;
- use the Flags and FaxIDs fields in the OutcallSchedule record to specify the fax document and enable CALLMaster to fax out without using the FaxSend command.

See [OutcallSchedule Record](#) for details.

## Create Audiotex Records and Associated Scripts

As all calls will be starting in Audiotex mode in this walkthrough, the Audiotex Modules will determine the call flow.

the Audiotex set-up,

If call result is a Connect and it receives a human or answering machine connect.

- the called party will be played a message

- If connect type is human then the called party will be given the option to "Press 1 to be removed from the List". If the called party enters a 1 then the OutcallSchedule record Status field will be updated to 1 - Don't call and the number called will be added to OutcallSchedule-DontCall table.

If call result is Fax then a document defined in the Fax table will be faxed to the called party.

NOTE: the Audiotex application will only get processing control if the call results in a connect (i.e. \$CR = 10). If the call does not result in a connect, then the call is reset based on the busy/no answer/no confirm retry values from the OutcallSchedule record.

We will need to create several Audiotex records as shown below. The Label field provides a brief description of each module.

Audiotex Records																	
Module ID	Label	Write Label to Log File	Module Action	Next Module ID	Fail Module ID	Response Length	Script File to Run (Must include full path name) (c:\Program Files\CALLMASTER\Scripts\SampleApp-ODMsgFax\)	Touch Tone Table									
								1	2	3	4	5	6	7	8	9	0
100	Start call and Initialize values	Yes	Next Module	110	9900	0	InitializeV al.bas										
110	Perform call analysis to	Yes	Next Module	9999	9900	0	OD-C allAn alysis.bas										

	determine if it is a human, answering machine, or fax response.																
200	Human Connect Detected.  Play Audiotext Module recording.	Yes	Next Module	9999	9900	0											
300	Answering Machine Connect Detected.  Play Audiotext Module recording.	Yes	Next Module	9999	9999												
400	Fax Connect Detected. Use FaxSend to fax	Yes	Next Module	9999	9999		FaxSend.bas										

	document																
500	Other Connect	Yes	Next Module	9999	9999												
9900	Call Failure Detected	Yes	Next Module	9999	9999	0	SetCallFailure.bas										
9910	Caller Hang-up Detected	Yes	Next Module	9999	9999	0	SetCallerHangup.bas										
9920	Maximum tries attempted	Yes	Next Module	9999	9999	0	SetMaxAttempts.bas										
9999	Last Module - End call	Yes	Last Module														

#### Auditex Module Description

- Module 100 is the starting module. We will use this module to initialize variables using script InitializeVal.bas. In particular we want to set local variables to the Out Dial system variables returned by CALLMaster. We will set the system variable lctrp to an Auditex module ID to send caller if they hang-up. We also want to define a variable (strCMConnection) for the connection string required to read the CALLMasterSQL database. You will need to change the connection string for your environment settings.
- Module 110 will perform call analysis to determine if it is a human, answering machine, or fax response.
  - If human connect detected, call control will return to Module 200 to play a message;
  - If answering machine connect detected, then a blank recording is made for up to 60 seconds or it detects continuous silence for duration of Record Max Silence System Parameter. In our walkthrough, CALLMaster will require 3 seconds of silence after the answering machine greeting before returning call control to module 300 to play the message;
  - If fax connect detected, call control will return to Module 400 to fax document;
  - If none of the above, call control will return to 500.
- Module 200 will just play the Auditex message recorded for human connect and go to module 9999 (next module ID).
- Module 300 will play the Auditex message recorded for Answering Machine connection.
- If you want to play the same message regardless of human or answering machine connect, then you

can use either module 200 or 300. Remember to change the Return command accordingly in the OD-CallAnalysis.bas script.

- Module 400 will fax document using the FaxSend CALLBasic command. The phone number is used only for reference purposes and not to make a call. We will set it to the called area code and phone number.
- Module 9920 is used to notify caller that they have reached the maximum attempts allowed for a valid input. The script associated with this module can be used to set flags and route processing as required.
- Modules 9900 and 9910 are required to handle call failure and capture caller hang-up respectively. Both these modules can be used to set flags and route processing as required.

Based on the Audiotex modules shown above, we will need 6 scripts.

Let's create the scripts first. Open Notepad and create scripts following code examples below. Remember to save them as .bas type files in the Program Files\CALLMaster\Scripts folder.

#### CALLBasic Script - InitializeVal.bas

[Copy to Clipboard](#)

##### Program

\*\*\*\*\*

```
' Script: InitializeVal.bas
' Function: This script is used to initialize values
```

\*\*\*\*\*

```
strYes = "Yes"
strNo = "No"
intYes = 1
intNo = 0
strCallFailure = strNo
strCallerHangup = strNo
strMaxAttempts = strNo
intOne = 1
intZero = 0
```

```
' Set retry counter and maximum attempts allowed
intTryCount = 1
intMaxAttemptsAllowed = 3
```

```
' OutcallSchedule record values
ODMatchExists = intNo
ODDBOpen = intNo
OD-strStatus-DontCall = "1"
```

```
' Get value of OutcallSchedule system variables and assign to local variables.
' If variable is an integer then convert to string local variable
OD-intOutDialID = $outdialID
IntegerToString OD-strOutDialID $outdialID
```

```
OD-intResult = $scr
IntegerToString OD-strResult $scr
```

```
OD-intConnection = $contype
IntegerToString OD-strConnection $outdialID
```

```
OD-strDialPrefix = $dialprefix
OD-strAreaCode = $areacode
OD-strPhoneNum = $phonenum
```

```
' Set connection string to read CALLMasterSQL database
```

```
strCMConnection = "Provider=SQLOLEDB;Server=sscmsql;DATABASE=CALLMasterSQL;UID=;PWD=;Integrated Security=SSPI;"
```

```
' Set the module to return to if caller hangs up  
lctrp = 9910
```

```
' Convert system variable $channel from integer to string to be used in concatenated print statement.  
IntegerToString strchannel $channel
```

```
pline = "***** For channel=" & strchannel  
pline = pline & " Started Out Dial for ID=" & OD-strOutDialID  
pline = pline & " with OD-strDialPrefix=" & OD-strDialPrefix  
pline = pline & "; OD-strAreaCode=" & OD-strAreaCode  
pline = pline & "; OD-strPhoneNum=" & OD-strPhoneNum  
PrintNL pline
```

EndProgram

#### CALLBasic Script - OD-CallAnalysis.bas

[Copy to Clipboard](#)

#### Program

```
*****
```

```
' Script: OD-CallAnalysis.bas
```

```
' Function: This script checks the call connection type and returns based on the system variable $contype.
```

```
*****
```

```
' Set variables for recording if Answering Machine Connection
```

```
intPmt = 0
```

```
intRecTime = 60
```

```
strT = "0"
```

```
strA = ""
```

```
' If Call Result Connected
```

```
If OD-intResult = 10 Then
```

```
  ' Human Voice Connection
```

```
  If OD-intConnection < 4 Then
```

```
    Return 200
```

```
  Else
```

```
    'Answering machine
```

```
    Record recfileid intPmt intRecTime strA strT e d
```

```
    FileIDToName destname recfileid
```

```
    FileDelete destname
```

```
    Return 300
```

```
  EndIf
```

```
Else
```

```
  ' If Call Result Fax
```

```
  If OD-intResult = 18 Then
```

```
    Return 400
```

```
  Else
```

```
    Return 500
```

```
  EndIf
```

```
EndIf
```

EndProgram

[Copy to Clipboard](#)

## CALLBasic Script - FaxSend.bas

### Program

```
*****
' Script: FaxSend.bas
' Function: This script faxes the document from the Fax Table.
*****

' Set the Docuement ID from the Fax table of the document you want to fax.
intFaxDocID = 1

' The phone number is used only for reference purposes and not to make a call.
' This field can be null if faxing to the same number that the call is connected to as part of the Out Dial.
strFaxPhoneNum = $areacode & $phonenumber

FaxSend intFaxDocID strFaxPhoneNum car

' CALLMaster returns the fax send status in system variable $faxsent.

pline = "***** For channel=" & strchannel
pline = pline & " Faxed Document ID = " & strFaxPhoneNum
pline = pline & " with transmission status returned " & $faxsent
PrintNL pline
```

EndProgram

[Copy to Clipboard](#)

## CALLBasic Script - SetCallFailure.bas

### Program

```
*****
' Script: SetCallFailure.bas
' Function: This script sets the strCallFailure variable to yes and prints information in the call log file.
*****

strCallFailure = strYes

'Print information to log file for reference and debugging
pline = "***** For channel=" & strchannel
pline = pline & "***** Call Failure. *****"
PrintNL pline
```

EndProgram

[Copy to Clipboard](#)

## CALLBasic Script - SetCallerHangup.bas

### Program

```
*****
' Script: SetCallerHangup.bas
' Function: This script sets the strCallFailure variable to yes and prints information in the call log file.
*****

strCallerHangup = strYes

'Print information to log file for reference and debugging
```

```

pline = "***** For channel=" & strchannel
pline = pline & "***** Caller Hangup. *****"
PrintNL pline

```

EndProgram

[Copy to Clipboard](#)

#### CALLBasic Script - SetMaxAttempts.bas

Program

```

!*****

```

```

' Script: SetMaxAttempts.bas

```

```

' Function: This script sets the strMaxAttempts variable to yes and prints information in the call log file.

```

```

!*****

```

```

strMaxAttempts = strYes

```

```

'Print information to log file for reference and debugging

```

```

pline = "***** For channel=" & strchannel

```

```

pline = pline & "***** Maximum Attempts reached to enter valid input *****"

```

```

PrintNL pline

```

EndProgram

Now you can create the Audiotex modules and reference the scripts above. Open the **CALLMaster Manager**, click on **Manage**, then select **Audiotex**. Let's add the Audiotex records shown above.

#### Define IVR Application List Entry

We need to define the custom IVR in the IVR Application List and set it to Audiotex Start Module 100. Open CALLMaster Manager and select **Manage | IVR Application List**. Let's add a new record as follows:

IVR Application List	
Audiotex Start Module	Application Name
100	Broadcast Audio Msg & Fax Doc Sample

#### Set Ports

As a last step, we need set the port(s) to the new IVR Application entry created above. Make sure the Audiotex Start Module field (read only) correctly points to module 100.

Set the Port Greeting File IDs to the main greeting for all calls if you have recorded one. If you do not want a main greeting then set the Greeting File IDs to zero. Leave all other fields to the default values.

If you are working with a demo license or have only one line connected, then make sure the connected port is set as specified below. All other ports should be set to Application 'Disabled'.

Port Records				
Port #	Application	Audiotex Start Module	Port Type Outbound	Port Greeting File IDs (Morning, Afternoon,



				Evening, Closed, Holidays)
1	Broadcast Audio Msg & Fax Doc Sample	100	Yes	0, 0, 0, 0, 0
2	Broadcast Audio Msg & Fax Doc Sample	100	Yes	0, 0, 0, 0, 0
3	Broadcast Audio Msg & Fax Doc Sample	100	Yes	0, 0, 0, 0, 0
4	Broadcast Audio Msg & Fax Doc Sample	100	Yes	0, 0, 0, 0, 0

### Test the custom application

Make sure you have set the CALLMasterSQL database connection string variable strCMConnection in the InitializeVal.bas script for your environment.

Save all your work.

Stop CALLMaster service.

Make Administrator Audiotex Recordings:

- Set Port 1 to Application Voice Mail.
- Start CALLMaster service.
- Make a call using Port 1.
- [Log in](#) as the Administrator using the Voice Mail Main Menu and make recordings for the Audiotex modules below.

#### Audiotex Module ID

#### Proposed Recording

200	Recording for Human Connection
300	Recording for Answering Machine Connection
9900	We are unable to process your call. Please contact the system administrator for further assistance.
9920	You have reached the maximum

attempts allowed for a valid input.  
Please contact the system  
administrator for further assistance.

Reset Port 1 to Application Audiotex. Set the Port Greeting File IDs to zeros.

Stop and Start CALLMaster service.

Make a phone call to test the application.

Tip: See [Walkthrough: Change Application Without Restarting CALLMaster Service](#) to set port to Voice Mail to make recording without stopping and restarting the CALLMaster service.

To download and install, see [Sample Applications](#).

## Walkthrough: Make Outbound Calls with 'Remove from List' Option

In this walkthrough we will make outbound calls using CALLMaster Out Dial functionality to play a message or fax a document depending on the call connect type.

- If the call connect type is human, the called party will be played a message and be given the option to be removed from the list.
  - If remove from the list option is selected, the number will be inserted in the OutcallSchedule-DontCall table and the OutcallSchedule record's Status field will be updated to 1 (Don't Call).
- If the call connect type is answering machine, the message will be played.
- If the call connect type is fax, a pre-defined document will be faxed to the called number.

This Sample Application is available for download. Use link at the end of this walkthrough to download and install sample application.

## Walkthrough Overview

In this walkthrough, you will:

1. Set System Parameters;
2. Set Dialogic Parameters;
3. Create Fax Record;
4. Create Audiotex Records and Associated Scripts;
5. Define entry in the IVR Application List for the new custom IVR;
6. Set Ports.

During this walkthrough, you will learn to do the following:

- Configure CALLMaster to answer in Audiotex mode;
- Get input using the Audiotex Module Action 'Touch Tone Table';
- Get value of system variables from the OutcallSchedule record;
- Use CALLBasic database commands to Insert and Update a record;
- Distinguish between a human, answering machine and fax connection;
- Use CALLBasic commands including FaxSend;
- Make Audiotex recordings;
- Handle caller hang up and call failure.

## Prerequisites and Assumptions

- CALLMaster software installed and running with the Dialogic board installed and configured. For more information, see [Installation and Setup](#);
- A working knowledge of [CALLMaster Manager](#). Specifically how to add, delete and edit records;
- Phone lines enabled and connected to CALLMaster;
- Activate Out Dial process by adding flag "ODS" to the System Parameters 'System Setting' field.
- Numbers to be called imported to the CALLMaster OutcallSchedule table, with required fields - Dialing Prefix, Area Code and Phone Number. All other fields could have the default values or you can upload custom values as part of the import process. See [OutcallSchedule Record](#) for details.
- Incorporate records from the OutcallSchedule-DontCall table to mark records with Status 1 (Don't Call) during the import process.

## Set System Parameters

As we are using Out Dial, we need to make sure that the System Parameters - Local Dialing Prefix, Long Distance Dialing Prefix, System's Area Code and System's Phone Number have the correct values for your environment. We also need to set the time that CALLMaster can 'Start Calling at' and needs to 'Stop Calling at'.

As we want CALLMaster to continuously attempt outbound calls and reduce the sleep default time from 30 seconds to 3 seconds, we will add flags 'OCNV' and 'OSST3' in the System Flags field.

As we want to leave a message if call receives an answering machine connect, we need to adjust the Parameter Record Max Silence (see [Out Dial - Human/Answering Machine/Fax Detection](#) for details).

Open the **CALLMaster Manager**. Click on **Parameters**, then select **System**. Double click on each of the parameters to change their values as needed. See [System Parameters](#) for details.

For this walkthrough we will assume the following values:

- **Local Dialing Prefix** - set to 9.
- **Long Distance Dialing Prefix** - set to 9,1,.
- **System's Area Code** - set to 914.
- **System's Phone Number** - set to 5555555.
- **Start Calling At** - set to 09:00:00
- **Stop Calling At** - set to 17:00:00
- **Record Max Silence** - set to 3.
- **System Flags** - set to OCNV OSST3 ODS

## Set Dialogic Parameters

As we are using Out Dial and want to detect Human, Answering Machine and Fax connect, we need to adjust the several Dialogic Parameters (see [Out Dial - Human/Answering Machine/Fax Detection](#) for details).

Open the **CALLMaster Manager**. Click on **Parameters**, then select **Dialogic**. Double click on each of the parameters to change their values as needed.

For this walkthrough we will assume the following values:

- **NBRDNA** - set to '8';
- **HEDGE** - set to 'Detect on TRAILING Edge';
- **INTFLG** - set to '7'
- **PAMDFAILTIME** - set to '10'

## Define Global Tone for Fax

For the Dialogic board to detect a fax tone, we need to define the tone settings in the Global Tones table.

Open the **CALLMaster Manager**. Click on **Manage**, then select **Global Tones**. If there is an record for a Fax tone, then just verify the setting. If not then Add a record to this table with values as below.

Global Tones													
Tone Type	Characteristic	Detection Mode	Translation		Frequencies (Hz)				Cadence (10ms)				
			Translate to Char	Character	Tone 1 Freq	Tone 1 Dev	Tone 2 Freq	Tone 2 Dev	On Time	On Time Dev	Off Time	Off Time Dev	Minimum Repetition Count
Fax	Single Tone	Leading Edge	Yes	F	1100	50	0	0	0	0	0	0	1

## Create Fax Record

To fax a document, we need to create the document in the format required by the Dialogic board, and define the document in the Fax table.

The fax document needs to be in a Tiff-F format. See [Fax Overview](#) for details.

For this walkthrough, download the [Fax Test Document](#) as a sample document and save to a folder.

To define the document created above, open the **CALLMaster Manager**. Click on **Manage**, then select **Faxes**. Add record using the Fax Test Document saved as the File to Send.

Note the Fax ID of the record added for use with the FaxSend command.

If you wish to fax various fax document, you may want to consider either:

- using a field in the OutcallSchedule record as an indicator to the fax document number and then use

this indicator in the FaxSend command;

- use the Flags and FaxIDs fields in the OutcallSchedule record to specify the fax document and enable CALLMaster to fax out without using the FaxSend command.

See [OutcallSchedule Record](#) for details.

## Create Audiotex Records and Associated Scripts

As all calls will be starting in Audiotex mode in this walkthrough, the Audiotex Modules will determine the call flow.

In the Audiotex set-up,

If call result is a Connect and it receives a human or answering machine connect.

- the called party will be played a message

- If connect type is human then the called party will be given the option to "Press 1 to be removed from the List". If the called party enters a 1 then the OutcallSchedule record Status field will be updated to 1 - Don't call and the number called will be added to OutcallSchedule-DontCall table.

If call result is Fax then a document defined in the Fax table will be faxed to the called party.

NOTE: the Audiotex application will only get processing control if the call results in a connect (i.e. \$CR = 10). If the call does not result in a connect, then the call is reset based on the busy/no answer/no confirm retry values from the OutcallSchedule record.

We will need to create several Audiotex records as shown below. The Label field provides a brief description of each module.

Audiotex Records																	
Module ID	Label	Write Label to Log File	Module Action	Next Module ID	Fail Module ID	Response Length	Script File to Run (Must include full path name) (c:\Program Files\CALLMaster\Scripts\SampleApp-ODR-removeOpt\)	Touch Tone Table									
								1	2	3	4	5	6	7	8	9	0
100	Start call and Initial	Yes	Next Module	110	9900	0	InitializeVal.bas										

	ize value s																
110	Perfo rm call analy sis to deter mine if it is a huma n, answ ering mach ine, or fax respo nse.	Yes	Next Modu le	9999	9900	0	OD-C allAn alysi s.bas										
200	Hum an Conn ect Dete cted.  Play Audi otex Modu le recor ding.	Yes	Next Modu le	210	9900	0											
210	Provi de optio ns "1 to be remo ved from the list" and get caller respo nse	Yes	Touc h Tone Table	9920	9900	1		220	290	290	290	290	290	290	290	290	290

220	Update Outcall Schedule record and set Status to Don't Call (1). Insert number dialed to the Outcall Schedule-Don't Call table.	Yes	Next Module	230	9900		UpdateOD Status-InsertODDon'tCall.bas										
230	Let caller know they have been removed from the list	Yes	Next Module	9999	9999												
290	Invalid input.	Yes	Next Module	291	9900	0	CheckTry Count.bas										
291	Try again.	Yes	Next Module	210	9900	0											
300	Answering Machine Connect	Yes	Next Module	9999	9999												

	Dete cted.  Play Audi otex Modu le recor ding.																
400	Fax Conn ect Dete cted. Use FaxS end to fax docu ment .	Yes	Next Modu le	9999	9999		FaxS end.b as										
500	Other Conn ect	Yes	Next Modu le	9999	9999												
9900	Call Failur e Dete cted	Yes	Next Modu le	9999	9999	0	SetC allFai lure.b as										
9910	Calle r Hang -up Dete cted	Yes	Next Modu le	9999	9999	0	SetC aller Hang up.ba s										
9920	Maxi mum tries atte mpte d	Yes	Next Modu le	9999	9999	0	SetM axAtt empt s.bas										
9999	Last Modu le - End call	Yes	Last Modu le														



## Audiotex Module Description

- Module 100 is the starting module. We will use this module to initialize variables using script InitializeVal.bas. In particular we want to set local variables to the Out Dial system variables returned by CALLMaster. We will set the system variable lctrp to an Audiotex module ID to send caller if they hang-up. We also want to define a variable (strCMConnection) for the connection string required to read the CALLMasterSQL database. You will need to change the connection string for your environment settings.
- Module 110 will perform call analysis to determine if it is a human, answering machine, or fax response.
  - If human connect detected, call control will return to Module 200 to play a message;
  - If answering machine connect detected, then a blank recording is made for up to 60 seconds or it detects continuous silence for duration of Record Max Silence System Parameter. In our walkthrough, CALLMaster will require 3 seconds of silence after the answering machine greeting before returning call control to module 300 to play the message;
  - If fax connect detected, call control will return to Module 400 to fax document;
  - If none of the above, call control will return to 500.
- Module 200 will just play the Audiotex message and go to module 210 (next module ID).
- Module 210 has Module Action 'Touch Tone Table' requiring a 1 digit input (response length set to 1). This type of module is used to offer caller options and get a corresponding touch tone input. This module offers the caller 1 option requiring a valid touch tone input of 1. The caller will be directed to module 220 if they enter 1. If they enter any other touch tone key, they will be directed to module 290. As a Touch Tone Table' Module Action, at the end of the module recording, CALLMaster will play prompt 245 "press \* to repeat", giving the caller 3 attempts to enter any touch tone key (does not have to be a valid input). If you want to give the caller 3 attempts to press any key but do not want to play this prompt, then you can replace the associated wav file with a null recording (see [Replacing Recordings](#)). If you do not want to give the caller 3 attempts to enter any key, use the [NTR](#) code in the Audiotex record Label field.
- Module 220 will update the OutcallSchedule record with Status set to Don't Call (1) and will insert a record to the OutcallSchedule-Don'tCall table if remove from the list option selected.
- Module 290 and 291 is to notify the caller that they have entered an invalid option. The associated script, CheckTryCount.bas, gives the caller 3 attempts to enter a valid option. If the caller has not entered a valid option after 3 attempts, processing will be forwarded to module 9920.
- Module 300 will play the Audiotex message recorded for Answering Machine connection.
- Module 400 will fax document using the FaxSend CALLBasic command. The phone number is used only for reference purposes and not to make a call. We will set it to the called area code and phone number.
- Module 9920 is used to notify caller that they have reached the maximum attempts allowed for a valid input. The script associated with this module can be used to set flags and route processing as required.
- Modules 9900 and 9910 are required to handle call failure and capture caller hang-up respectively. Both these modules can be used to set flags and route processing as required.

Based on the Audiotex modules shown above, we will need 6 scripts.

Let's create the scripts first. Open Notepad and create scripts following code examples below. Remember to save them as .bas type files in the Program Files\CALLMaster\Scripts folder.

### CALLBasic Script - InitializeVal.bas

[Copy to Clipboard](#)

#### Program

```
*****
' Script: InitializeVal.bas
' Function: This script is used to initialize values
*****

strYes = "Yes"
strNo = "No"
intYes = 1
```

```

intNo = 0
strCallFailure = strNo
strCallerHangup = strNo
strMaxAttempts = strNo
intOne = 1
intZero = 0

' Set retry counter and maximum attempts allowed
intTryCount = 1
intMaxAttemptsAllowed = 3

' OutcallSchedule record values
ODMatchExists = intNo
ODDBOpen = intNo
OD-strStatus-DontCall = "1"

' Get value of OutcallSchedule system variables and assign to local variables.
' If variable is an integer then convert to string local variable
OD-intOutDialID = $outdialID
IntegerToString OD-strOutDialID $outdialID

OD-intResult = $scr
IntegerToString OD-strResult $scr

OD-intConnection = $contype
IntegerToString OD-strConnection $outdialID

OD-strDialPrefix = $dialprefix
OD-strAreaCode = $areacode
OD-strPhoneNum = $phonenum

' Set connection string to read CALLMasterSQL database
strCMConnection = "Provider=SQLOLEDB;Server=<server>;DATABASE=CALLMasterSQL;UID=;PWD=;Integrated Security=SSPI;"

' Set the module to return to if caller hangs up
lctrapp = 9910

' Convert system variable $channel from integer to string to be used in concatenated print statement.
IntegerToString strchannel $channel

pline = "***** For channel=" & strchannel
pline = pline & " Started Out Dial for ID=" & OD-strOutDialID
pline = pline & " with OD-strDialPrefix=" & OD-strDialPrefix
pline = pline & "; OD-strAreaCode=" & OD-strAreaCode
pline = pline & "; OD-strPhoneNum=" & OD-strPhoneNum
PrintNL pline

```

EndProgram

[Copy to Clipboard](#)

**CALLBasic Script - OD-CallAnalysis.bas**

Program

\*\*\*\*\*

' Script: OD-CallAnalysis.bas

' Function: This script checks the call connection type and returns based on the system variable \$contype.

\*\*\*\*\*

```

' Set variables for recording if Answering Machine Connection
intPmt = 0
intRecTime = 60
strT = "0"
strA = ""

' If Call Result Connected
If OD-intResult = 10 Then
  ' Human Voice Connection
  If OD-intConnection < 4 Then
    Return 200
  Else
    'Answering machine
    Record recfileid intPmt intRecTime strA strT e d
    FileIDToName destname recfileid
    FileDelete destname
    Return 300
  EndIf
Else
  ' If Call Result Fax
  If OD-intResult = 18 Then
    Return 400
  Else
    Return 500
  EndIf
EndIf

```

EndProgram

[Copy to Clipboard](#)

#### CALLBasic Script - CheckTryCount.bas

##### Program

```

*****
' Script: CheckTryCount.bas
' Function: This script checks the number of invalid input attempts.
*****

' If the caller has reached the maximum attempts allowed then the call is routed to module 9920.
' If not then they are returned back to Next Module ID.
If intTryCount > intMaxAttemptsAllowed Then
  Return 9920
Else
  intTryCount = intTryCount + 1
EndIf

' Convert system variable intTryCount from integer to string to be used in concatenated print statement.
IntegerToString strTryCount intTryCount

pline = "***** For channel=" & strchannel
pline = pline & " TryCount = "
pline = pline & strTryCount
PrintNL pline

```

EndProgram

**CALLBasic Script - UpdateODStatus-InsertODDontCall.bas****Program**

```
*****
' Script: UpdateODStatus-InsertODDontCall.bas
' Function: This script updates the Status to 1 for Don't Call
*****

' Create a new command and open a connection to the CALLMaster database
Set ODvarCommand = New Command
ODvarCommand.ActiveConnection strCMConnection

ODDBOpen = intYes

If $DBError <> intZero Then
    Set ODvarCommand = Nothing
    ODDBOpen = intNo
    pline = "***** For channel=" & strchannel
    pline = pline & " Error Opening db. "
    PrintNL pline
    Return 9900
Else
EndIf

' Set Query to update record
strSQL = "Update [OutcallSchedule] Set [Status] ="
strSQL = strSQL & OD-strStatus-DontCall
strSQL = strSQL & " WHERE [ID] ="
strSQL = strSQL & OD-strOutDialID
strSQL = strSQL & ""

pline = "For channel " & strchannel
pline = pline & " strSQL = " & strSQL
PrintNL pline

' Open CMDM database and find record
ODvarCommand.Execute strSQL

If $DBError <> intZero Then
    ODvarCommand.CloseConnection
    Set ODvarCommand = Nothing
    ODDBOpen = intNo
    pline = "***** For channel=" & strchannel
    pline = pline & " Error Updating Record. "
    PrintNL pline
    Return 9900
Else
EndIf

' Set Query to insert record to the OutcallSchedule-DontCall table
strSQL = "INSERT INTO [OutcallSchedule-DontCall] (AreaCode, PhoneNumber) VALUES ("
strSQL = strSQL & OD-strAreaCode
strSQL = strSQL & ", " & OD-strPhoneNum
strSQL = strSQL & OD-strPhoneNum
strSQL = strSQL & ")"
pline = "For channel " & strchannel
pline = pline & strSQL
```

```

PrintNL pline

' Execute Insert command
ODvarCommand.Execute strSQL

If $DBError <> intZero Then
    ODvarCommand.CloseConnection
    Set ODvarCommand = Nothing
    ODDBOpen = intNo
    pline = "***** For channel=" & strchannel
    pline = pline & " Error inserting. "
    PrintNL pline
    Return 9900
Else
EndIf

' Close DB Connection and clear command
ODvarCommand.CloseConnection
Set ODvarCommand = Nothing

' ODDDB database closed
ODDBOpen = intNo

'Print information to log file for reference and debugging
pline = "***** For channel=" & strchannel
pline = pline & " Updated OutcallSchedule to Don't Call for ID=" & OD-strOutDialID
pline = pline & " with OD-strDialPrefix=" & OD-strDialPrefix
pline = pline & "; OD-strAreaCode=" & OD-strAreaCode
pline = pline & "; OD-strPhoneNum=" & OD-strPhoneNum
pline = pline & ". Inserted record to the OutcallSchedule-DontCall table"
PrintNL pline

EndProgram

```

[Copy to Clipboard](#)

## CALLBasic Script - FaxSend.bas

### Program

```

*****
' Script: FaxSend.bas
' Function: This script faxes the document from the Fax Table.
*****

' Set the Document ID from the Fax table of the document you want to fax.
intFaxDocID = 1

' The phone number is used only for reference purposes and not to make a call.
' This field can be null if faxing to the same number that the call is connected to as part of the Out Dial.
strFaxPhoneNum = $areacode & $phonenumber

FaxSend intFaxDocID strFaxPhoneNum car

' CALLMaster returns the fax send status in system variable $faxsent.

pline = "***** For channel=" & strchannel
pline = pline & " Faxed Document ID = " & strFaxPhoneNum
pline = pline & " with transmission status returned " & $faxsent

```

PrintNL pline

EndProgram

### CALLBasic Script - SetCallFailure.bas

[Copy to Clipboard](#)

Program

```
*****
' Script: SetCallFailure.bas
' Function: This script sets the strCallFailure variable to yes and prints information in the call log file.
*****

strCallFailure = strYes

'Print information to log file for reference and debugging
pline = "***** For channel=" & strchannel
pline = pline & "***** Call Failure. *****"
PrintNL pline
```

EndProgram

### CALLBasic Script - SetCallerHangup.bas

[Copy to Clipboard](#)

Program

```
*****
' Script: SetCallerHangup.bas
' Function: This script sets the strCallFailure variable to yes and prints information in the call log file.
*****

strCallerHangup = strYes

'Print information to log file for reference and debugging
pline = "***** For channel=" & strchannel
pline = pline & "***** Caller Hangup. *****"
PrintNL pline
```

EndProgram

### CALLBasic Script - SetMaxAttempts.bas

[Copy to Clipboard](#)

Program

```
*****
' Script: SetMaxAttempts.bas
' Function: This script sets the strMaxAttempts variable to yes and prints information in the call log file.
*****

strMaxAttempts = strYes

'Print information to log file for reference and debugging
pline = "***** For channel=" & strchannel
pline = pline & "***** Maximum Attempts reached to enter valid input *****"
PrintNL pline
```

EndProgram

Now you can create the Audiotex modules and reference the scripts above. Open the **CALLMaster Manager**, click on **Manage**, then select **Audiotex**. Let's add the Audiotex records shown above.

### Define IVR Application List Entry

We need to define the custom IVR in the IVR Application List and set it to Audiotex Start Module 100. Open CALLMaster Manager and select **Manage | IVR Application List**. Let's add a new record as follows:

IVR Application List	
Audiotex Start Module	Application Name
100	Broadcast with Remove Option Sample

### Set Ports

As a last step, we need set the port(s) to the new IVR Application entry created above. Make sure the Audiotex Start Module field (read only) correctly points to module 100.

Set the Port Greeting File IDs to the main greeting for all calls if you have recorded one. If you do not want a main greeting then set the Greeting File IDs to zero. Leave all other fields to the default values.

If you are working with a demo license or have only one line connected, then make sure the connected port is set as specified below. All other ports should be set to Application 'Disabled'.

Port Records				
Port #	Application	Audiotex Start Module	Port Type Outbound	Port Greeting File IDs (Morning, Afternoon, Evening, Closed, Holidays)
1	Broadcast with Remove Option Sample	100	Yes	0, 0, 0, 0, 0
2	Broadcast with Remove Option Sample	100	Yes	0, 0, 0, 0, 0
3	Broadcast with Remove Option Sample	100	Yes	0, 0, 0, 0, 0
4	Broadcast with Remove Option Sample	100	Yes	0, 0, 0, 0, 0

### Test the custom application

Make sure you have set the CALLMasterSQL database connection string variable strCMConnection in the InitializeVal.bas script for your environment.

Save all your work.

Stop CALLMaster service.

Make Administrator Audiotex Recordings:

- Set Port 1 to Application Voice Mail.
- Start CALLMaster service.
- Make a call using Port 1.
- [Log in](#) as the Administrator using the Voice Mail Main Menu and make recordings for the Audiotex modules below.

Audiotex Module ID	Proposed Recording
200	Recording for Human Connection
210	Press 1 if you want to be removed from the list
230	You have been removed from the list
290	Invalid option.
291	Please try again
300	Recording for Answering Machine Connection
9900	We are unable to process your call. Please contact the system administrator for further assistance.
9920	You have reached the maximum attempts allowed for a valid input. Please contact the system administrator for further assistance.

Reset Port 1 to Application Audiotex. Set the Port Greeting File IDs to zeros.

Stop and Start CALLMaster service.

Make a phone call to test the application.

Tip: See [Walkthrough: Change Application Without Restarting CALLMaster Service](#) to set port to Voice Mail to make recording without stopping and restarting the CALLMaster service.



To download and install, see [Sample Applications](#).

## Walkthrough: Dictation Functionality with Playback Navigation

Dictation functionality can be customized in CALLMaster using the RecordAppend command in conjunction with the Play command's playback navigation capabilities.

In this walkthrough, authorized callers will be allowed to make new dictation recordings, review their existing recordings, get recording instructions or set their personal options.

A mailbox record will need to be set-up for each authorized caller. The caller will need to enter their mailbox number and password to access the system. Each mailbox could be set-up with an email address if the owner requires email delivery of their dictation recording.

If the caller selects option 1 to make a new recording, they will hear recording instructions the first time they log in to record. They will have the option to press 1 to bypass the instructions or 9 to hear the instructions again.

During recording they will have the ability to press 2 to pause/unpause or any key terminate the recording and start playback.

During recording playback, the caller will have the option to press 1 to rewind, 2 to pause/unpause, 3 to fast forward or any other key to terminate playback.

After the caller has heard the recording or terminated playback, he/she will be given the option to press 1 to save the recording, 2 to replay the recording again, 3 to redo the entire recording, or 4 to continue recording from where they stopped playback. By pressing 4, the recording will continue from the place where playback was last stopped.

This Sample Application is available for download. Use link at the end of this walkthrough to download and install sample application.

## Walkthrough Overview

In this walkthrough, you will:

1. Create Mailbox Records;
2. Create Audiotex Records/Modules and Associated Scripts;
3. Define entry in the IVR Application List for the new custom IVR;
4. Set Ports.

During this walkthrough, you will learn to do the following:

- Configure CALLMaster to answer in Audiotex mode;
- Get value of a system variable and assign it to a local variable;
- Send caller to the mailbox to specifically review messages or to set personal options using RVM and POP flags;
- Get input using the Audiotex Module Action 'Touch Tone Table';
- Use CALLBasic database commands to Read a record;
- Use the RecordAppend command to concatenate recording;
- Use the Play command with system variables 'playbackposition', 'termdigit' and 'tcutthrough' to navigate playback;
- Use the Email command to email mailbox owner dictation file as wav attachment.
- Use the backslash character ('\') as the first character of the email address in the mailbox record field. This will enable CALLMaster to maintain the email address but not automatically send out the email;
- Handle caller hang up and call failure.

## Prerequisites and Assumptions

- CALLMaster software installed and running with the Dialogic board installed and configured. For more information, see [Installation and Setup](#).
- A working knowledge of [CALLMaster Manager](#). Specifically how to add, delete and edit records.
- Email activated and tested. See [Activating Email](#).

## Create Mailbox Records

We need to create a mailbox record for each Authorized caller where the **Type = User**.

We do not want CALLMaster to automatically send email notification but we do want to maintain the email address for a custom email message. To accomplish this, we will use the backslash character (\) as the first character of the email address in the mailbox **Email** field. This will enable CALLMaster to maintain the email address but not automatically send out the email notifications. Leave the **Email** field blank if you want the recording to be forwarded to a central administrative email (see InitValDictation.bas script below for details).

Note: Keep Mailbox record 0000 with the default settings as CALLMaster requires it. You should change the password for this record to maintain security.

Open the **CALLMaster Manager**. Click on **Manage**, then select **Mailboxes**. Let's add 4 records in this walkthrough as follows:

Mailbox Records				
Mailbox #	Password	Last Name	Email	Box Status
1000	1000	Problem Reporting	\User1000@txt.att.net	User
1001	1001	User 1001	\User1001@support.com	User
1002	1002	User 1002	\User1002@service.com	User
1003	1003	User 1003		User

### Create Audiotex Records/Modules and Associated Scripts

As all calls will be starting in Audiotex mode in this walkthrough, the Audiotex Modules will determine the call flow.

We will need to create several Audiotex records as shown below. The Label field provides a brief description of each module.

Audiotex Records																	
Mod ule ID	Label	Wri te La bel  to Lo g Fil e	Modu le Actio n	Next Mod ule ID	Fail Mod ule ID	Respo nse Lengt h	Script File to Run (Must include full path name) (c:\Program Files\CALLMaster\Script s\SampleApp-Dictation\ )	Touch Tone Table									
								1	2	3	4	5	6	7	8	9	0
10	Start call and Initialize values. Set	Yes	Next Modul e	20	9990	0	InitializeVal.bas										

	lctrp = 9999.																	
20	Play main menu greeting (1 for dictation or 2 to report a problem) and get caller response.	Yes	Touch Tone Table	9998	9990	1		1000	2000	9998	9998	9998	9998	9998	9998	9998	9000	9998
100	Get 4-digit User Id	Yes	Next Module	9998	9990	0	GetDTMF-MBNo.bas											
110	Get 4-digit PIN	Yes	Next Module	9998	9990	0	GetDTMF-PIN.bas											
120	Read MB table for record with matching User ID (MB#) and PIN (Pswd).	Yes	Next Module	9998	9990	0	ReadCMMB.bas											
130	Play Dictation menu to caller (To start a new dictation press 1 , to review your dictations press 2 , to hear instructions on how to use dictation this dictation press 3, to change personal options press 4 or any keyto	Yes	Touch Tone Table	9998	9990	1		1000	2000	3000	4000	9998	9998	9998	9998	9998	9000	9998

	end this call).																		
190	User Id not 4 digits	Yes	Next Module	191	9990	0	CheckTryCount.bas												
191	Try again	Yes	Next Module	100	9990	0													
192	PIN not 4 digits	Yes	Next Module	193	9990	0	CheckTryCount.bas												
193	Try again	Yes	Next Module	110	9990	0													
194	User ID and PIN not authorized (not found on Mailboxes table)	Yes	Next Module	195	9990	0	CheckTryCount.bas												
195	Try again	Yes	Next Module	100	9990	0													
200	Send caller to MB 1000 to report a problem	Yes	Next Module	9999	9990	0													
1000	Initialize values used for RecordAppend. Set lctrp and exit to 1110.	Yes	Next Module	1001	9990	0	InitValDictation.bas												
1001	Play recording instructions . Give caller	Yes	Touch Tone Table	1003	9990	1		10 03	10 03	10 03	10 03	10 03	10 03	10 03	10 03	10 03	10 01	10 03	

	option to press 1 to skip hearing the instruction or 9 to repeat the instructions .																		
1003	Play message "Record"	Yes	Next Module	1005	9990	0													
1004	Play message "Continue"	Yes	Next Module	1005	9990	0													
1005	Start RecordAppend function.	Yes	Next Module	1006	9990	0	RecordAppend.bas												
1006	Check recording termination digit	Yes	Next Module	9910	9910	0	CheckRecTermination.bas												
1008	Play message "Paused" and wait 5 minutes or until caller enters 2 to continue.	Yes	Next Module	9910	9910	0	GetDTMF-RecContinue.bas												
1012	Play back recording.	Yes	Next Module	1020	9910	0	PlayRecFileId.bas												
1020	Give caller recording options "Press 1 to approve and save recording, 2 to replay the recording again, 3 to redo the	Yes	Touch Tone Table	9999	9910	1		11 00	12 00	13 00	10 04	10 20	10 20	10 20	10 20	10 20	10 20	10 20	

	entire recording, or 4 to continue recording from last termination point."																		
1100	Select message priority '1 for Stat, 2 for Urgent, * for regular'	Yes	Next Module	1110	9910	0	GetDTMF-MsgPriority.bas												
1110	Check is recording deleted	Yes	Next Module	9999	9910	0	CheckRecordingDeleted.bas												
1120	Save recording as message to mailbox	Yes	Next Module	1130	9910	0	AddMessage.bas												
1130	Email Message	Yes	Next Module	1150	9910	0	Email.bas												
1150	Confirm recording has been saved	Yes	Next Module	130	9910	0	ResetLCtrapExit.bas												
1200	Reset playback position to replay recording	Yes	Next Module	1012	9910	0	ResetPlayBackPosition.bas												
1300	Delete recording file to redo	Yes	Next Module	1000	9990	0	DeleteRecording.bas												
2000	RVM Send caller to review dictation - Note RVM flag at the	Yes	Login	130	9990	0	Return-BasedonMB.bas												

	start of this field).																		
3000	Play recording instructions	Yes	Next Module	130	9990	0													
4000	POP Send caller to manage personal options - Note POP flag at the start of this field)	Yes	Login	130	9990	0	Return-BasedonMB.bas												
9000	Cut-thru to send caller to voice mail - used by administrator to make recordings, etc.	Yes	Voice Mail	9999	9999	0													
9910	Caller Hang-up Detected - delete recording before exiting.	Yes	Next Module	9999	9999	0	DeleteRecording.bas												
9990	System error encountered - notify caller	Yes	Next Module	9999	9999	0													
9998	Play prompt 218 'Thank you for calling - goodbye'	Yes	Next Module	9999	9999	0													
9999	Last Module - End call	Yes	Last Module	0	0	0													

#### Audiotex Module Description

- Module 10 is the starting module. We will use this module to initialize variables using script



InitializeVal.bas. In particular we want to initialize local variable intFirstTimeInstructionsPlayed to No so recording instructions are played before the first recording. We will set the system variable lctrp to an Audiotex module ID to send caller if they hang-up. We also want to define a variable (strCMConnection) for the connection string required to read the CALLMasterSQL database. You will need to change the connection string for your environment settings.

- Module 20 is a Touch Tone Table type module giving caller the option to press 1 to make a recording or 2 to report a problem. Option 9 is a non-announced option to cut-thru to voice mail, primarily used by the system administrator to make/maintain system recordings.
- Module 100 gets the caller's 4-digit user id (mailbox number) and module 110 gets their 4-digit PIN (password). If caller enters less than 4 digits for either the User ID or PIN, Modules 190 and 192 notifies caller of the error.
- Module 120 reads the Mailboxes table for matching record. If found then processing is returned to Module 130 for the dictation menu, if not then Module 194 notifies caller of the error.
- Module 130 is a Touch Tone Table module that plays the Dictation Menu - 'To start a new dictation press 1, to review your dictations press 2, to hear instructions on how to use dictation this dictation press 3, to change personal options press 4 or any key to end this call', and directs processing based on caller's selection.
- Module 200 directs caller to Mailbox 1000 to report a problem.
- Module 1000 runs script InitValDictation.bas. This script initializes variable used to make a new dictation. It directs caller Module 1001 to hear the recording instructions if flag intFirstTimeInstructionsPlayed is set to No.
- Module 1001 plays the recording instructions and gives caller the option to press 1 to skip the instructions or 9 to repeat the instructions.
- Module 1003 instructs caller to start recording and Module 1004 to continue recording.
- Module 1005 starts recording using the RecordAppend CALLBasic command. This command uses the system variable 'playbackposition' to determine the position to start recording or playback.
- Module 1006 checks the termination digit returned by RecordAppend command. If the caller paused recording by entering 2 then processing is directed to Module 1008. If the caller entered any other key to stop recording then processing is forwarded to Module 1200.
- Module 1008 tells caller that the recording has been paused. Using the GetDTMF command, processing is on hold for the duration of the value set for variable 'intRecordingPauseSecs' (default is 300 seconds) in the InitializeVal.bas script.
- Module 1012 plays back recording.
- Module 1020 gives caller the option to press 1 to save, 2 to replay, 3 to redo or 4 to continue recording.
- Module 1100 gets caller to select message priority "Press 1 for Stat, 2 for urgent, or \* for regular".
- Module 1110 checks if recording saved.
- If recording saved then call is directed to Module 1120 to add recording as a message to the owner's mailbox. Module 1130 emails message to address in the mailbox record if available else to the variable strToEmailBackup assigned in the InitValDictation.bas script. Module 1150 notifies caller that their recording has been saved.
- Module 1200 resets system variable 'playbackposition' to zero for replaying recording from the start.
- Module 1300 deletes recording if caller has elected to redo the recording.
- Module 2000 logs in caller to review their mailbox recordings using the RVM flag in the Audiotex Label field.
- Module 3000 plays system instructions to the caller.
- Module 4000 logs in caller to set personal options using the POP flag in the Audiotex Label field.
- Module 9000 forwards caller to Voice Mail, primarily for use by the administrator to make/maintain system recordings.
- Modules 9910 and 9990 are required to handle caller hang-up and call failure respectively.

Based on the Audiotex modules shown above, we need several scripts.

Let's create the scripts first. Open Notepad and create scripts following code examples below. Remember to save them as .bas type files in the Program Files\CALLMaster\Scripts folder.

<b>CALLBasic Script - InitializeVal.bas</b>	<a href="#">Copy to Clipboard</a>
Program	

```

! *****
' File Type : CALLMaster script file by SpeechSoft, Inc. 06/09/00
' File Name : InitValues.bas
' Description : Initialize Values.
! *****

' Initialize Variables

'Accept = "1"
strAccept = "1"
strReject = "2"
strHearAgain = "3"

intYes = 1
intNo = 0

intZero = 0
intOne = 1

intTryCount = 0

ttcutthrough = 1

intMBMatchExists = intNo
intCMDB-Open = intNo

intFirstTimeInstructionsPlayed = intNo

' Set connection string to read CALLMasterSQL database
strCMConnection = "Provider=SQLOLEDB;Server=<server>;DATABASE=CALLMasterSQL;UID=;PWD=;Integrated Security=SSPI;"

lctrp = 9999
exit = 9999

EndProgram

```

[Copy to Clipboard](#)

#### CALLBasic Script - GetDTMFMBNo.bas

#### Program

```

! *****
' File Type : CALLMaster script file by SpeechSoft, Inc. 12/216/99
' File Name : GetDTMF-MBNo.bas
' Description : Get Userid(Mailbox) from caller using the touchtone keypad
'
! *****

' Initialize variables

strMBNoInp = ""
strTMask = ""
intNDigits = 4
intTimeout = 9

' Get input from caller
GetDTMF strMBNoInp strTMask intNDigits intTimeout

' Get length of input

```

```
StringLength intMBNoInpLen strMBNoInp
```

```
If intMBNoInpLen = intNDigits Then  
    Return 110  
Else  
    Return 190  
EndIf
```

```
Print "strMBNoInp = "  
PrintNL strMBNoInp
```

```
EndProgram
```

[Copy to Clipboard](#)

#### CALLBasic Script - GetDTMF PIN.bas

Program

```
! *****
```

```
' File Type : CALLMaster script file by SpeechSoft, Inc. 12/216/99
```

```
' File Name : GetDTMF-PIN.bas
```

```
' Description : Get PIN from caller using the touchtone keypad
```

```
'
```

```
! *****
```

```
' Initialize variables
```

```
strPINInp = ""
```

```
strTMask = ""
```

```
intNDigits = 4
```

```
intTimeout = 9
```

```
' Get input from caller
```

```
GetDTMF strPINInp strTMask intNDigits intTimeout
```

```
' Get length of input
```

```
StringLength intPINInpLen strPINInp
```

```
If intPINInpLen = intNDigits Then
```

```
    Return 120
```

```
Else
```

```
    Return 192
```

```
EndIf
```

```
Print "strPINInp = "
```

```
PrintNL strPINInp
```

```
EndProgram
```

[Copy to Clipboard](#)

#### CALLBasic Script - ReadCMDB.bas

Program

```
! *****
```

```
' File Type : CALLMaster script file by SpeechSoft, Inc. 10/16/00
```

```
' File Name : RetrieveMailbox.bas
```

```
' Description : Retrieve Mailbox from CMDB.
```

```
'
```

```
! *****
```

```

' Initialize variables
strMBNo = strMBNoInp
strPSW = strPINInp

' Select & Open CMDB database and see if Mailbox exists
Set CMvarSet = New Recordset

' Set Query for the Last used Mailbox number
strSQL = "SELECT * FROM MailBoxes WHERE BoxNumber ="
strSQL = strSQL & strMBNo
strSQL = strSQL & " And Password = "
strSQL = strSQL & strPSW
strSQL = strSQL & ""
Print " "
PrintNL strSQL

' Open CMDB database and find record
CMvarSet.Open strSQL strCMConnection

Set CMvarFields = CMvarSet.Fields

intCMDB-Open = intYes
PrintNL "CMDB Open."

If CMvarSet.EOF = 0 Then
    Set CMvarF30 = CMvarFields.Field 30
    strMBEmail = CMvarF30.Value
    intMBMatchExists = intYes
    Set CMvarF30 = Nothing
Else
    intMBMatchExists = intNo
    PrintNL "CMDB Match does not exist"
EndIf

' Free CMDB varSet, varFields, and close database

' Destroy Fields Object
Set CMvarFields = Nothing

' Close recordset
CMvarSet.Close

' Free memory
Set CMvarSet = Nothing

' CMDB database closed
intCMDB-Open = intNo
PrintNL "CMDB Closed."

If intMBMatchExists = intYes Then
    Return 130
Else
    Return 194
EndIf

```

EndProgram

**CALLBasic Script - CheckTryCount.bas**[Copy to Clipboard](#)**Program**

```
*****
' File Type : CALLMaster script file by SpeechSoft, Inc. 12/21/99
' File Name : CheckTryCount.bas
' Description : Give caller 3 attempts to enter correct Contract No
'
```

```
*****

intTryCount = intTryCount + 1
```

```

If intTryCount > 2 Then
    Return 9990
Else
EndIf
```

```
EndProgram
```

**CALLBasic Script - InitValDictation.bas**[Copy to Clipboard](#)**Program**

```
! *****
' File Type : CALLMaster script file by SpeechSoft, Inc. 07/10/00
' File Name : InitVal.bas
' Description : Initialize Values.
! *****
```

```

' Initialize variables
```

```
recfileid = 0
playbackposition = 0
```

```
strMsgPriority = ""
intRecordingDeleted = intNo
intRecordingApproved = intNo
```

```
WavName = ""
```

```
intRecordingPauseSecs = 300
intPauseCount = 0
```

```
strFromEmail = "callmaster@speechsoft.com"
strToEmailBackup = "\sales@speechsoft.com"
```

```
lctrp = 1110
exit = 1110
```

```

If intFirstTimeInstructionsPlayed = intNo Then
    intFirstTimeInstructionsPlayed = intYes
    Return 1001
Else
    Return 1003
EndIf
```

EndProgram

[Copy to Clipboard](#)

### CALLBasic Script - RecordAppend.bas

Program

```
! *****
' File Type : CALLMaster script file by SpeechSoft, Inc. 07/10/00
' File Name : RecordAppend.bas
' Description : Examples of RecordAppend command.
! *****

' Initialize variables
intPmt = 0
intRecTime = 3600
strTDigit = "*"
a = ""

RecordAppend recfileid intPmt intRecTime a strTDigit e d
intSavedplaybackposition = playbackposition

' Renames speechfile id from integer to full name with folder path.
FileIDToName WavName recfileid

Print "Saved wav recording as "
PrintNL WavName

Print recfileid

Print "PBPosition after RecordAppend = "
PrintNL playbackposition

Print "RecordAppend term digit "
PrintNL strTDigit
```

EndProgram

[Copy to Clipboard](#)

### CALLBasic Script - CheckRecTermination.bas

Program

```
! *****
' File Type : CALLMaster script file by SpeechSoft, Inc. 01/15/01
' File Name : CheckRecTermination.bas
' Description :
! *****

If strTDigit = "2" Then
    Return 1008
Else
    Return 1200
EndIf
```

EndProgram

**CALLBasic Script - GetDTMFRecContinue.bas**[Copy to Clipboard](#)**Program**

```
! *****
' File Type : CALLMaster script file by SpeechSoft, Inc. 12/216/99
' File Name : GetDTMF-RecContinue.bas
' Description : Get PIN Number from caller using the touchtone keypad
'
! *****

' Initialize variables
intPauseCount = intPauseCount + 1

strContInp = ""
strTMask = ""
intNDigits = 1
intTimeout = intRecordingPauseSecs

' Get input from caller
GetDTMF strContInp strTMask intNDigits intTimeout
Print "strContInp = "
PrintNL strContInp

' Get length of input
StringLength intContInpLen strContInp

If strContInp = "2" Then
    Return 1004
Else
    If intPauseCount > 2 Then
        Return 1020
    Else
        Return 1008
    EndIf
EndIf

EndProgram
```

**CALLBasic Script - PlayRecFileId.bas**[Copy to Clipboard](#)**Program**

```
! *****
' File Type : CALLMaster script file by SpeechSoft, Inc. 07/10/00
' File Name : PlayFile.bas
' Description : Play back recorded file with dictation capabilities.
! *****

' System variables termdigit and playbackposition are returned after the Play command
termdigit = ""

' Plays back recorded file using speechfile id value.
Play recfileid c d

Print "PBPosition after play command = "
PrintNL playbackposition
```

```
Print "termdigit = "  
PrintNL termdigit
```

EndProgram

#### CALLBasic Script - CheckRecordingApproval.bas

[Copy to Clipboard](#)

Program

! \*\*\*\*\*

' File Type : CALLMaster script file by SpeechSoft, Inc. 12/216/99

' File Name : Check CheckRecordingApproval.bas

' Description : Check if recording approved and set indicator

'

! \*\*\*\*\*

' Initialize variables

If \$tt = "1" Then

intRecordingApproved = intYes

Else

intRecordingApproved = intNo

EndIf

Print "RecordingApproved = "

PrintNL intRecordingApproved

EndProgram

#### CALLBasic Script - GetDTMFMsgPriority.bas

[Copy to Clipboard](#)

Program

! \*\*\*\*\*

' File Type : CALLMaster script file by SpeechSoft, Inc. 12/216/99

' File Name : GetDTMF-MsgPriority.bas

' Description : Get Message Priority from caller using the touchtone keypad

'

! \*\*\*\*\*

' Initialize variables

strMsgPriorityInp = ""

strTMask = ""

intNDigits = 1

intTimeout = 10

' Get input from caller

GetDTMF strMsgPriorityInp strTMask intNDigits intTimeout

Print "strMsgPriorityInp = "

PrintNL strMsgPriorityInp

EndProgram

[Copy to Clipboard](#)



## CALLBasic Script - CheckRecordingDeleted.bas

### Program

```
' *****  
' File Type : CALLMaster script file by SpeechSoft, Inc. 01/15/01  
' File Name : CheckRecordingDeleted.bas  
' Description :  
' *****  
  
' Check file status.  
  
Print "intRecordingDeleted = "  
PrintNL intRecordingDeleted  
  
Print "intRecordingApproved = "  
PrintNL intRecordingApproved  
  
If intRecordingDeleted = intYes Then  
    Return 9999  
Else  
    If intRecordingApproved = intYes Then  
        Return 1120  
    Else  
        Return 9910  
    EndIf  
EndIf  
  
EndProgram
```

[Copy to Clipboard](#)

## CALLBasic Script - AddMessage.bas

### Program

```
' *****  
' File Type : CALLMaster script file by SpeechSoft, Inc. 07/10/00  
' File Name : AddMessage.bas  
' Description : adds a recorded message file into a mailbox.  
' *****  
  
Print "CM Add message script started on "  
PrintNL $today  
  
strToMB = strMBNo  
intMsgID = recfileid  
strFromMB = "0000"  
  
AddMessage strToMB intMsgID strFromMB  
  
Print "Message added strToMB = "  
Print strToMB  
Print " intMsgID = "  
Print intMsgID  
Print " strFromMB = "  
PrintNL strFromMB  
  
EndProgram
```

**CALLBasic Script - Email.bas****Program**

```
! *****
' File Type : CALLMaster script file by SpeechSoft, Inc. 07/10/00
' File Name : Email.bas
' Description : Send email.
! *****

' Setup email subject based on priority
If strMsgPriorityInp = "1" Then
    strMsgPriority = "Stat"
Else
    If strMsgPriorityInp = "2" Then
        strMsgPriority = "Urgent"
    Else
        strMsgPriority = "Regular"
    EndIf
EndIf

strSubject = strMBNo & " - "
strSubject = strSubject & strMsgPriority

' Place default email address if mailbox does not have email address
StringLength intMBEmailLen strMBEmail

If intMBEmailLen = 0 Then
    strToEmail = strToEmailBackup
Else
    MemMove strToEmail strMBEmail intOne intMBEmailLen
EndIf

' SendEmail <mapiprofile> <from> <address> <CCaddress> <BCCaddress> <subject> <Bodytext> <fileattachments>
' NOTE: a space is needed for blank fields

strMapiProfile = "CALLMaster"
strFromAddress = strFromEmail
strToAddress = strToEmail
strCCAddress = " "
strBCCAddress = " "

strBodyText = "The attached message is from mailbox " & strMBNo
strFileAttach = WavName

' send email
Email strMapiProfile strFromAddress strToAddress strCCAddress strBCCAddress strSubject strBodyText strFileAttach

PLine = "Message emailed to " & strToAddress
PLine = PLine & " Recording file "
PLine = PLine & WavName
PLine = PLine & " for "
PLine = PLine & strSubject
PrintNL PLine

EndProgram
```

**CALLBasic Script - ResetLCTrapExit.bas**[Copy to Clipboard](#)**Program**

```
! *****
' File Type : CALLMaster script file by SpeechSoft, Inc. 10/24/00
' File Name : ResetLCTrapExit.bas
' Description : On hangup, reset LCTrap and # (Exit) and go to last module
! *****

    lctrp = 9999
    exit = 9999
```

**EndProgram****CALLBasic Script - ResetPlayBackPosition.bas**[Copy to Clipboard](#)**Program**

```
! *****
' File Type : CALLMaster script file by SpeechSoft, Inc.
' File Name : ResetPlayBackPosition.bas
' Description : Reset system variable playbackposition.
! *****

    playbackposition = 0
```

**EndProgram****CALLBasic Script - DeleteRecording.bas**[Copy to Clipboard](#)**Program**

```
! *****
' File Type : CALLMaster script file by SpeechSoft, Inc. 07/10/00
' File Name : DeleteRecording.bas
' Description : Delete recorded file.
! *****

' Deletes file using full path name.

' Check if recording file exists
intFStatus = 0
intFSize = 0
intFMdate = 0
FileStatus WavName intFStatus intFSize intFMdate

If intFStatus <> 0 Then
    ' Recording exists
    If intRecordingDeleted = intNo Then
        FileDelete WavName
        intRecordingDeleted = intYes
        Print "Deleted wav recording "
        PrintNL WavName
    Else
        PrintNL "Recording already deleted"
```

```

    EndIf
Else
    If recfileid = 0 Then
        PrintNL "Recording does not exist"
    Else
        ' This is to check if the hangup event occurred after recording file was created
        ' but before it was named as the wav file name
        ' i.e. recfileid created but FileIDToName command not executed.
        FileIDToName WavName recfileid
        FileDelete WavName
        intRecordingDeleted = intYes
        Print "Created and Deleted wav recording "
        PrintNL WavName
    EndIf
EndIf

lctrp = 9999
exit = 9999

EndProgram

```

[Copy to Clipboard](#)

#### CALLBasic Script - ReturnBasedOnMB.bas

```

Program
! *****
' File Type : CALLMaster script file by SpeechSoft, Inc.
' File Name : Return-BasedOnMB.bas
' Description : Return to module depending on mailbox.
! *****

' Return command based on input strMBNo. Caller will be sent to module number = strMBNo
Return strMBNo

EndProgram

```

Now you can create the Audiotex modules and reference the scripts above. Open the **CALLMaster Manager**, click on **Manage**, then select **Audiotex** to add the Audiotex records shown above.

#### Define IVR Application List

We need to define the custom IVR in the IVR Application List and set it to Audiotex Start Module 10. Open CALLMaster Manager and select Manage | IVR Application List. Let's add a new record as follows:

IVR Application List	
Audiotex Start Module	Application Name
100	Dictation

#### Set Ports

As a last step, we need set the port(s) to the new IVR Application entry created above. Make sure the Audiotex Start Module field (read only) correctly points to module 10.

As this walkthrough uses Notification, make sure at least one port is enabled for Type Outbound; by default,

all the ports are enabled for Inbound. Set the Port Greeting File IDs to the main greeting for all calls if you have recorded one. If you do not want a main greeting then set the Greeting File IDs to zero. Leave all other fields to the default values.

If you are working with a demo license or have only one line connected, then make sure the connected port is set as specified below. All other ports should be set to Application 'Disabled'.

Port Records				
Port #	Application	Audiotex Start Module	Port Type Outbound	Port Greeting File IDs (Morning, Afternoon, Evening, Closed, Holidays)
1	Dictation	10	No	0, 0, 0, 0, 0
2	Dictation	10	No	0, 0, 0, 0, 0
3	Dictation	10	Yes	0, 0, 0, 0, 0
4	Dictation	10	Yes	0, 0, 0, 0, 0

### Test the custom application

Make sure you have set the CALLMasterSQL database connection string variable strCMConnection in the InitializeVal.bas script for your environment.

Save all your work.

Stop CALLMaster service.

Make Administrator Audiotex Recordings:

- Call the system and select non-announced option 9 when you hear main menu 'press 1 to make a recording or 2 to report a problem'. Option 9 is provided to direct the system administrator to Voice Mail make/maintain system recordings.

- [Log in](#) as the Administrator using the [Voice Mail Main Menu](#) and make recordings for the Audiotex modules below.

Audiotex Module ID	Proposed Recording
20	To login for dictation press 1, to report a problem press 2
100	Please enter your 4-digit user-id
110	Please enter your 4-digit PIN

130	To start a new dictation press 1 , to review your dictations press 2 , to hear instructions on how to use dictation this dictation press 3, to change personal options press 4 or any key to end this call.
190	User Id must be 4 digits.
191	Please try again
192	PIN must be 4 digits
193	Please try again
194	UserId or PIN invalid
195	Please try again
1001	During recording press 2 to pause and un-pause and any key to stop recording and start playback. During playback press 1 to rewind, 2 to pause and un-pause, 3 to FF. Any other key will stop playback at which point you can press 1 to save, 2 to replay, or 3 to redo the entire recording. To continue recording from where you stopped playback press 4. In the future, to skip these instructions press 1. To repeat these instructions press 9.
1003	Record
1004	Continue
1008	Paused
1020	1 to save, 2 to replay, 3 to redo or 4 to continue.
1100	Please select message priority. Press 1 for Stat, 2 for urgent, or * for regular.

1150	Your recording has been saved
3000	This system allows you to make recordings, review and delete or save your recordings or to set your password and greeting.
9990	Sorry you are having trouble with the system. Please call the administrator for more information

Make a phone call to test the application.

The Mailbox owners can use option 4 (set personal options) at the Main Menu to record their mailbox name and greeting.

To download and install, see [Sample Applications](#).

## Walkthrough: Survey/Questionnaire with Concatenated Response Logging

In this walkthrough we will ask several questions, get a verbal answer to each question, concatenate the answers and log the concatenate the answer for future reference.

This Sample Application is available for download. Use link at the end of this walkthrough to download and install sample application.

### Walkthrough Overview

In this walkthrough, you will:

1. Create Audiotex Records/Modules and Associated Scripts;
2. Define entry in the IVR Application List for the new custom IVR;
3. Set Ports.

During this walkthrough, you will learn to do the following:

- Configure CALLMaster to answer in Audiotex mode;
- Record caller's answer and concatenate using the Record, AppendFile, FileIDToName and FileDelete commands;
- Use CALLBasic database commands to Read and Insert a record using a date/time data type field;
- Send caller to another Audiotex Module using the 'Next Module' Audiotex Module Action;
- Email recording using the Email CALLBasic command;
- Make Audiotex and Custom Prompt recordings;
- Handle caller hang up and call failure.

### Prerequisites and Assumptions

- CALLMaster software installed and running with the Dialogic board installed and configured. For more information, see [Installation and Setup](#).
- A working knowledge of [CALLMaster Manager](#). Specifically how to add, delete and edit records.
- Phone lines enabled and connected to CALLMaster.
- Email activated and tested. See [Activating Email](#).
- As the recordings will be saved to a new SurveyAnswerRecording folder, using the explorer, create folder C:\Program Files\CALLMaster\SurveyAnswerRecording.
- SampleApps database (SampleApps.mdf) in the CALLMaster\Bin folder attached to your local SQL server. In this walkthrough, CALLMaster will ask the pre-recorded questions from the SurveyQuestions table and logging the responses to SurveyAnswersLog tables from this database .

SurveyQuestions table has 6 questions. The QuestionPrompt field contains a 3 digit prompt number. This number represents a custom prompt and will need to be recorded via the Administrator log in options.

RecID (int)	QuestionNumber (int)	QuestionPrompt (int)	Memo (nvarchar(150))
1	1	701	Please state your name
2	2	702	Please say your address
3	3	703	How many family members in your household
4	4	704	How many credit cards do you have?



5	5	705	What is the total amount owed on your credit card?
6	6	706	Would you like us to call you to consolidate your credit?

## Create Audiotex Records/Modules and Associated Scripts

As all calls will be starting in Audiotex mode in this walkthrough, the Audiotex Modules will determine the call flow.

In the Audiotex set-up, the caller will be played a welcome greeting followed by instructions about the survey process. We will read the SurveyQuestions table for the first question, play the recording matching the Question Prompt, get answer. We will then ask the next question, concatenate the answer to the end of the previous answers. This question/answer process will continue until we process the last question. The concatenated answer will be written to the SurveyAnswersLog table with the date and time of the call.

We will need to create several Audiotex records as shown below. The Label field provides a brief description of each module.

Audiotex Records						
Module ID	Label	Write Label to Log File	Module Action	Next Module ID	Fail Module ID	Script File to Run (Must include full path name) (c:\Program Files\CALLMaster\Scripts\SampleApp-Survey\)
1000	Start call and Initialize values	Yes	Next Module	1010	9900	InitializeVal.bas
1010	Play welcome greeting	Yes	Next Module	1020	9900	
1020	Play instructions	Yes	Next Module	1100	9900	
1100	Read first question from the SurveyQuestions table.	Yes	Next Module	1110	9900	ReadFirst-SurveyQuestion.bas
1110	Play current question prompt	Yes	Next Module	1200	9900	PlayCurrentQuestion.bas
1200	Record single answer	Yes	Next Module	1210	9900	RecordAnswer.bas
1210	Concatenate Answer	Yes	Next Module	1500	9900	CopyAppendAnswer.bas

1500	Read next question	Yes	Next Module	1510	9900	ReadNext-SurveyQuestion.bas
1510	Delete single answer recording	Yes	Next Module	1520	9900	DeleteAnswerFile.bas
1520	Email concatenated answer recording	Yes	Next Module	9500	9900	EmailRecording.bas
9110	Questionnaire record not found on SurveyQuestions table	Yes	Next Module	9500	9900	
9500	Add SurveyAnswersLog record	Yes	Next Module	9510	9900	Insert-SurveyAnswersLog.bas
9510	Thank you for calling. Goodbye	Yes	Next Module	9999	9900	
9900	Call Failure Detected	Yes	Next Module	9999	9999	SetCallFailure.bas
9910	Caller Hang-up Detected	Yes	Next Module	9999	9999	SetCallerHangup.bas
9999	Last Module - End call	Yes	Last Module			

#### Auditex Module Description

- Module 1000 is the starting module. We will use this module to initialize variables using script InitializeVal.bas. In particular we want to convert the current date and time to string use as part of the recording name. We also want to define a variable (strSAConnection) for the connection string required to read the SampleApps database. You will need to change the connection string for your environment settings in this script. We also want to set the system variable lctrp to an Auditex module to send caller if they hang-up. See sample code below.
- Module 1010 plays a welcome greeting and Module 1020 provides instruction about the survey process.
- Module 1100 reads the SurveyQuestions table for the first question.
- Module 1110 Plays the question prompt, module 1200 records answer, module 1210 concatenates answer.
- Module 1500 reads the SurveyQuestions table for the next question. If found then processing continues to module 1110.
- Module 1510 and 1520 ends recording processing and emails recording. As the recordings will be saved to a new SurveyAnswerRecording folder, using the explorer, create folder C:\Program Files\CALLMaster\SurveyAnswerRecording
- Module 9110 is used to tell caller that a matching mailbox record was not found. It will also give the caller the option to enter another number.
- Modules 9900 and 9910 are required to handle call failure and capture caller hang-up respectively. Both these modules can be used to set flags and route processing as required.

Based on the Auditex modules shown above, we will need several scripts.

Create a SampleApp-Survey subfolder in the Program Files\CALLMaster\Scripts folder - we will save the scripts for this sample application in this folder.  
 Let's create the scripts first. Open Notepad and create scripts with the following code examples below.  
 Remember to save them as .bas type files and save them in the Program Files\CALLMaster\Scripts\SampleApp-Survey folder.

#### CALLBasic Script - InitializeVal.bas

[Copy to Clipboard](#)

#### Program

\*\*\*\*\*

```
' Script: InitializeVal.bas
' Function: This script is used to initialize values
```

\*\*\*\*\*

```
strYes = "Yes"
strNo = "No"
intYes = 1
intNo = 0
strCallFailure = strNo
strCallerHangup = strNo
strMaxAttempts = strNo
intOne = 1
intZero = 0
intNegOne = 0 - 1
strZero = "0"
GetFloat fltZero strZero 0
```

```
intDecMM = 12
intJanMM = 1
```

```
strStatus-OK = "OK"
strStatus-NOK = "NOK"
```

```
' Set retry counter and maximum attempts allowed
intTryCount = 1
intMaxAttemptsAllowed = 3
```

```
ttcutthrough = 1
intInvalidSelection = intNo
intCallerHangUp = intNo
```

```
intAnswerTime = 120
intAnsweredAllQuestions = intNo
```

```
' Get Date and Time string component for Answer Recording Name and SurveyAnswerLog
dteCallDateTime = $today
GetDate dteCallDateTime intcurrmm intcurrday intcurryear intcurrrhour intcurrmin intcurrsec
```

```
intcurrmm = $month
IntegerToString strcurrmm intcurrmm
If intcurrmm < 10 Then
    strcurrmm = "0" & strcurrmm
Else
EndIf
```

```
intcurrday = $day
IntegerToString strcurrday intcurrday
```

```

If intcurrday < 10 Then
    strcurrday = "0" & strcurrday
Else
EndIf

intcurryear = $year
IntegerToString strcurryear intcurryear

intcurrhour = $hour
IntegerToString strcurrhour intcurrhour

If intcurrhour < 10 Then
    strcurrhour = "0" & strcurrhour
Else
EndIf

intcurrmin = $minute
IntegerToString strcurrmin intcurrmin
If intcurrmin < 10 Then
    strcurrmin = "0" & strcurrmin
Else
EndIf

intcurrsec = $sec
IntegerToString strcurrsec intcurrsec
If intcurrsec < 10 Then
    strcurrsec = "0" & strcurrsec
Else
EndIf

' Create Date and Time string variable for adding to SurveyAnswers record DateTime Data Type in mm/dd/yyyy hh:mm:ss format
' Date format = MMDDYYYY
strCallDate = ""
strCallDate = strcurrmm & "/"
strCallDate = strCallDate & strcurrday
strCallDate = strCallDate & "/"
strCallDate = strCallDate & strcurryear

' Time format HHMM Military Time
strCallTime = ""
strCallTime = strCallTime & strcurrhour
strCallTime = strCallTime & ":"
strCallTime = strCallTime & strcurrmin & ":"
strCallTime = strCallTime & strcurrsec

strCallDateTime = strCallDate & " " & strCallTime

' Create Date and Time string variable for Survey Answer recording name in mmddyyyy-hhmmss format
' Date format = MMDDYYYY
strCallDate = ""
strCallDate = strcurrmm
strCallDate = strCallDate & strcurrday
strCallDate = strCallDate & strcurryear

' Time format HHMM Military Time
strCallTime = ""
strCallTime = strCallTime & strcurrhour

```

```

strCallTime = strCallTime & strcurrmin
strCallTime = strCallTime & strcurrsec

strRecDateTimeFormat = strCallDate & "-" & strCallTime

' Set up path and name for SurveyAnswer recording
strAnswerFilePath = "C:\\Program Files\\CALLMaster\\SurveyAnswerRecording\\" & "\\"
strAnswerFileName = strRecDateTimeFormat & ".wav"
strAnswerFilePath-n-Name = strAnswerFilePath & strAnswerFileName

' Email variables
strMapiProfile = "CALLMaster"
strToAddress = "sales@speechsoft.com"
strFromAddress = "support@speechsoft.com"
strSubject = "CALLMaster Sample Application - Survey"
strFileAttach = strAnswerFilePath-n-Name
strBCCAddress = " "
strCCAddress = " "
strBodyText = "Test recording from CALLMaster Sample Application - Survey and Questionnaire"

'SampleApps db
SADBOpen = intNo

' SurveyQuestions values
SQMatchExists = intNo
SQ-intQuestionNumber = 0
SQ-intQuestionPrompt = 0

' SurveyAnswer values
SAMatchExists = intNo
SA-strCallDateTime = strCallDateTime
SA-strStatus = strStatus-OK
SA-strSurveyAnswerRecordingName = ""
SA-strLastQuestionAnswered = "0"
intSurveyAnswerLogAdded = intNo

' Set connection string to read SampleApps database.
strSAConnection = "Provider=SQLOLEDB;Server=<servername>;DATABASE=SampleApps;UID=;PWD=;Integrated Se

' Set the module to return to if caller hangs up
lctrp = 9910

' Convert system variable $channel from integer to string to be used in concatenated print statement.
IntegerToString strchannel $channel

pline = "***** For channel=" & strchannel
pline = pline & " Started initialization process "
PrintNL pline

```

EndProgram

**CALLBasic Script - ReadFirst-SurveyQuestion.bas**

[Copy to Clipboard](#)

Program

!\*\*\*\*\*

```

' Script: ReadFirst-SurveyQuestion.bas
' Function: This script reads the SurveyQuestions table for the first question.
*****

' Set new command and open connection to SampleApps database.
Set SQvarCommand = New Command

SQvarCommand.ActiveConnection strSAConnection

If $DBError <> intZero Then
    SQvarSet.Close
    Set SQvarSet = Nothing
    SA-strStatus = strStatus-NOK
    pline = "***** For channel=" & strchannel
    pline = pline & " Error Connecting db. "
    PrintNL pline
    Return 9900
Else
EndIf

' Create a new recordset
Set SQvarSet = New Recordset

strSQL = "SELECT QuestionNumber, QuestionPrompt FROM SurveyQuestions WHERE QuestionNumber >= 0 ORDER
QuestionNumber"

pline = "For channel " & strchannel
pline = pline & " Read First for strSQL = " & strSQL
PrintNL pline

' Open SADB database and find record
Set SQvarSet = SQvarCommand.Execute strSQL

If $DBError <> intZero Then
    SQvarSet.Close
    Set SQvarSet = Nothing
    SA-strStatus = strStatus-NOK
    pline = "***** For channel=" & strchannel
    pline = pline & " Error Opening db. "
    PrintNL pline
    Return 9900
Else
EndIf

Set SQvarFields = SQvarSet.Fields
SADBOpen = intYes

' If the record exists then move selected field values for processing
SQvarSet.MoveFirst

If SQvarSet.EOF = 0 Then
    Set SQvarF1 = SQvarFields.Field 1
    Set SQvarF2 = SQvarFields.Field 2
    SQ-intQuestionNumber = SQvarF1.Value
    SQ-intQuestionPrompt = SQvarF2.Value

' Free field variables
Set SQvarF1 = Nothing

```

```

Set SQvarF2 = Nothing

' Set Switches
SQMatchExists = intYes

Else
    SQMatchExists = intNo

' Free SQDB varSet, varFields, and close database
' Destroy Fields Object
Set SQvarFields = Nothing

' Close recordset
SQvarSet.Close

' Free memory
Set SQvarSet = Nothing

' clear command
SQvarCommand.CloseConnection
Set SQvarCommand = Nothing

' SQDB database closed
SQDBOpen = intNo
EndIf

' If record found then return 1110 else return 9110
If SQMatchExists = intYes Then
    IntegerToString strCurrentQuestionNumber SQ-intQuestionNumber
    IntegerToString strCurrentQuestionPrompt SQ-intQuestionPrompt
    'Print information to log file for reference and debugging
    pline = "***** For channel=" & strchannel
    pline = pline & " SurveyQuestions record found for QuestionNumber = " & strCurrentQuestionNumber
    pline = pline & " with QuestionPrompt = " & strCurrentQuestionPrompt
    PrintNL pline
    Return 1110
Else
    pline = "***** For channel=" & strchannel
    pline = pline & " SurveyQuestions record NOT found for QuestionNumber >= 0"
    PrintNL pline
    Return 9110
EndIf

EndProgram

```

[Copy to Clipboard](#)

#### CALLBasic Script - PlayCurrentQuestion.bas

#### Program

```

*****
' Script: PlayCurrentQuestion.bas
' Function: This script plays the current question.
*****

playbackposition = 0
Play SQ-intQuestionPrompt c d
playbackposition = 0

```

EndProgram

[Copy to Clipboard](#)

## CALLBasic Script - RecordAnswer.bas

Program

```
*****
' Script: RecordAnswer.bas
' Function: This script records answer to current question.
*****

' Initialize variables
Prompt = 100
RecTime = intAnswerTime
t = "*"
a = ""
recfileid = 0
fatfilename = ""

strAnswerFile = "C:\Program Files\CALLMaster\Answer.wav"

playbackposition = 0

' Plays prompt 100 (Press * to stop recording) to instruct caller, Then.
' Records file and places speechfile id in integer variable recfileid.

Record recfileid Prompt RecTime a t e d

playbackposition = 0

' Renames speechfile id from integer to full name with folder path.
' This is required if you wish to rename or delete speechfiles

FileIDToName fatfilename recfileid

' The file to be renamed cannot exist. Use the FileStatus function to check and delete if it exists.
intStatus = 0
intSize = 0
intMdate = 0
FileStatus strAnswerFile intStatus intSize intMdate
If intStatus > 0 Then
    FileDelete strAnswerFile
Else
EndIf

' Renames file to user selected name.
FileRename fatfilename strAnswerFile

playbackposition = 0

' Deletes file using full path name.
FileDelete fatfilename
```

EndProgram



**CALLBasic Script - CopyAppendAnswer.bas**[Copy to Clipboard](#)**Program**

```
*****
' Script: CopyAppendAnswer.bas
' Function: This script concatenates the current answer to previous answers
*****

' If SQ-intQuestionNumber = 1 then copy answer to saved name else append answer

If SQ-intQuestionNumber = 1 Then
    strOverwrite = "Yes"
    CopyFile strAnswerFile strAnswerFilePath-n-Name strOverwrite
Else
    rewindbytes = 0
    AppendFile strAnswerFile strAnswerFilePath-n-Name rewindbytes
EndIf

IntegerToString SA-strLastQuestionAnswered SQ-intQuestionNumber
SA-strSurveyAnswerRecordingName = strAnswerFileName

EndProgram
```

**CALLBasic Script - ReadNext-SurveyQuestion.bas**[Copy to Clipboard](#)**Program**

```
*****
' Script: ReadNext-SurveyQuestion.bas
' Function: This script reads the SurveyQuestions table for the next question.
*****

SQMatchExists = intNo

pline = "For channel " & strchannel
pline = pline & " Read Next Record "
PrintNL pline

SQvarSet.MoveNext

' If the record exists then move selected field values for processing
If SQvarSet.EOF = 0 Then
    Set SQvarF1 = SQvarFields.Field 1
    Set SQvarF2 = SQvarFields.Field 2

    ' Assign field values to variables
    SQ-intQuestionNumber = SQvarF1.Value
    SQ-intQuestionPrompt = SQvarF2.Value

    ' Free field variables
    Set SQvarF1 = Nothing
    Set SQvarF2 = Nothing

    ' Set Switches
    SQMatchExists = intYes
```

```

Else
    SQMatchExists = intNo

    ' Free SQDB varSet, varFields, and close database
    ' Destroy Fields Object
    Set SQvarFields = Nothing

    ' Close recordset
    SQvarSet.Close

    ' Free memory
    Set SQvarSet = Nothing

    ' clear command
    SQvarCommand.CloseConnection
    Set SQvarCommand = Nothing

    ' SQDB database closed
    SQDBOpen = intNo
EndIf

' If record found then return 1110 else return 1510
If SQMatchExists = intYes Then
    IntegerToString strCurrentQuestionNumber SQ-intQuestionNumber
    IntegerToString strCurrentQuestionPrompt SQ-intQuestionPrompt
    'Print information to log file for reference and debugging
    pline = "***** For channel=" & strchannel
    pline = pline & " SurveyQuestions record found for QuestionNumber = " & strCurrentQuestionNumber
    pline = pline & " with QuestionPrompt = " & strCurrentQuestionPrompt
    PrintNL pline
    Return 1110
Else
    intAnsweredAllQuestions = intYes
    pline = "***** For channel=" & strchannel
    pline = pline & " SurveyQuestions record NOT found for QuestionNumber > " & strCurrentQuestionNumber
    PrintNL pline
    Return 1510
EndIf

EndProgram

```

[Copy to Clipboard](#)

#### CALLBasic Script - DeleteAnswerFile.bas

#### Program

```

*****
' Script: DeleteAnswerFile.bas
' Function: This script deletes the single answer file.
*****

' Deletes file using full path name.
FileDelete strAnswerFile

'Print information to log file for reference and debugging
pline = "***** For channel=" & strchannel
pline = pline & "***** Deleted Answer File = " & strAnswerFile
PrintNL pline

```

EndProgram

### CALLBasic Script - EmailRecording.bas

[Copy to Clipboard](#)

Program

\*\*\*\*\*

```
' Script: EmailRecording.bas
' Function: This script emails the concatenated recording.
```

\*\*\*\*\*

```
' send email
Email strMapiProfile strFromAddress strToAddress strCCAddress strBCCAddress strSubject strBodyText strFileAttach

'Print information to log file for reference and debugging
pline = "***** For channel=" & strchannel
pline = pline & "***** Recording Emailed "
PrintNL pline
```

EndProgram

### CALLBasic Script - Insert-SurveyAnswersLog.bas

[Copy to Clipboard](#)

Program

\*\*\*\*\*

```
' Script: Insert-SurveyAnswersLog.bas
' Function: This script inserts the SurveyAnswersLog table for the first question.
```

\*\*\*\*\*

```
' Set new command and open connection to SampleApps database.
Set SAvarCommand = New Command
```

```
SAvarCommand.ActiveConnection strSAConnection
```

```
If $DBError <> intZero Then
    SA-strStatus = strStatus-NOK
    pline = "***** For channel=" & strchannel
    pline = pline & " Error Connecting to SAdb. "
    PrintNL pline
    Return 9900
Else
EndIf
```

```
strSQL = "INSERT INTO [SurveyAnswersLog] (CallDateTime, CallStatus, RecordingName,LastQuestionAnswered) VAL
strSQL = strSQL & SA-strCallDateTime
strSQL = strSQL & ", " & SA-strStatus
strSQL = strSQL & ", " & SA-strSurveyAnswerRecordingName
strSQL = strSQL & ", " & SA-strLastQuestionAnswered
strSQL = strSQL & ")"
pline = "For channel " & strchannel
pline = pline & strSQL
PrintNL pline
```

```
' Execute Insert command
SAvarCommand.Execute strSQL
```

```

If $DBError <> intZero Then
    SA-strStatus = strStatus-NOK
    pline = "***** For channel=" & strchannel
    pline = pline & " Error inserting. "
    PrintNL pline
    Return 9900
Else
EndIf

' Close connection and clear command
SAvarCommand.CloseConnection
Set SAvarCommand = Nothing

' SADB database closed
SADBOpen = intNo

intSurveyAnswerLogAdded = intYes
pline = "For channel " & strchannel
pline = pline & " SurveyAnswersLog added for recording="
pline = pline & SA-strSurveyAnswerRecordingName
PrintNL pline

EndProgram

```

#### CALLBasic Script - SetCallFailure.bas

[Copy to Clipboard](#)

##### Program

```

*****
' Script: SetCallFailure.bas
' Function: This script sets the strCallFailure variable to yes and prints information in the call log file.
*****

strCallFailure = strYes

'Print information to log file for reference and debugging
pline = "***** For channel=" & strchannel
pline = pline & "***** Call Failure. *****"
PrintNL pline

EndProgram

```

#### CALLBasic Script - SetCallerHangup.bas

[Copy to Clipboard](#)

##### Program

```

*****
' Script: SetCallerHangup.bas
' Function: This script sets the strCallFailure variable to yes and prints information in the call log file.
*****

strCallerHangup = strYes

```

```
'Print information to log file for reference and debugging
pline = "***** For channel=" & strchannel
pline = pline & "***** Caller Hangup. *****"
PrintNL pline
```

```
EndProgram
```

Now you can create the Audiotex modules and reference the scripts above. Open the **CALLMaster Manager**, click on **Manage**, then select **Audiotex**. Let's add the Audiotex records shown above.

### Define IVR Application List Entry

We need to define the custom IVR in the IVR Application List and set it to Audiotex Start Module 1000. Open CALLMaster Manager and select **Manage | IVR Application List**. Let's add a new record as follows:

IVR Application List	
Audiotex Start Module	Application Name
1000	Survey Sample

### Set Ports

As a last step, we need set the port(s) to the new IVR Application entry created above. Make sure the Audiotex Start Module field (read only) correctly points to module 1000.

Set the Port Greeting File IDs to the main greeting for all calls if you have recorded one. If you do not want a main greeting then set the Greeting File IDs to zero. Leave all other fields to the default values.

If you are working with a demo license or have only one line connected, then make sure the connected port is set as specified below. All other ports should be set to Application 'Disabled'.

Port Records			
Port #	Application	Audiotex Start Module	Port Greeting File IDs (Morning, Afternoon, Evening, Closed, Holidays)
1	Survey Sample	1000	0, 0, 0, 0, 0
2	Survey Sample	1000	0, 0, 0, 0, 0
3	Survey Sample	1000	0, 0, 0, 0, 0
4	Survey Sample	1000	0, 0, 0, 0, 0

### Test the custom application

Make sure you have set the SampleApps database connection string variable strSAConnection in the InitializeVal.bas script for your environment.

Save all your work.

Stop CALLMaster service.

Make Administrator Audiotex Recordings:

- Set Port 1 to Application Voice Mail.
- Start CALLMaster service.
- Make a call using Port 1.
- [Log in](#) as the Administrator using the [Voice Mail Main Menu](#) and make recordings for the Audiotex modules below.

Audiotex Module ID	Proposed Recording
1010	Welcome Greeting - Welcome to the CALLMaster Sample Survey Questionnaire
1020	Survey Instruction - You will be asked a series of questions. Please speak your answers clearly.
9510	Thank you for participating in the CALLMaster Sample Survey Questionnaire
9900	We are unable to process your call. Please contact the system administrator for further assistance.

Prompt #	Proposed Recording
701	Please state your name
702	Please say your address
703	How many family members in your household?
704	How many credit cards do you have?
705	What is the total amount owed on your credit card?
706	Would you like us to call you to

consolidate your credit?

Reset Port 1 to Application Audiotex. Set the Port Greeting File IDs to zeros.

Stop and Start CALLMaster service.

Make a phone call to test the application.

Tip: See [Walkthrough: Change Application Without Restarting CALLMaster Service](#) to set port to Voice Mail to make recording without stopping and restarting the CALLMaster service.

To download and install, see [Sample Applications](#).

## Walkthrough: Set-up Team Bulletin Board with Rotating Sponsor Message

In this walkthrough the caller will hear a sponsor message, input a fixed length number identifying their team, listen to their team leader's message with the option to leave a message. The team leader will have the option to log in to change the team message at any time and will be notified when a team member has left a message.

The Sponsor list will be incremented sequentially to determine the current Sponsor Number to use. Each Sponsor will be set-up with a Mailbox record whose Mailbox # is set to the Sponsor Number, Type is Announcement and Greeting is the Sponsor's message. Each Sponsor message will be played sequentially at the start of a call across all lines.

This Sample Application is available for download. Use link at the end of this walkthrough to download and install sample application.

### Walkthrough Overview

In this walkthrough, you will:

1. Create Mailbox Records;
2. Verify System Parameters;
3. Create Notification Records;
4. Create Audiotex Records/Modules and Associated Scripts.
5. Define entry in the IVR Application List for the new custom IVR;
6. Set Ports.

During this walkthrough, you will learn to do the following:

- Use a Global variable;
- Create and run an Entry.bas script;
- Set-up an mailboxes with:
  - [Type](#) Announcement and Option To Leave Message
  - [Settings](#) UL3 (limit owner options to 3);
- Get value of a global variable and assign it to a local variable;
- Get input using the GetDTMF command;
- Send caller to the mailbox associated with the local variable using the CALLBasic Return command;
- Notify mailbox owner via email and phone when caller has left a message in mailbox;
- Handle caller hang up and call failure.

### Prerequisites and Assumptions

- CALLMaster software installed and running with the Dialogic board installed and configured. For more information, see [Installation and Setup](#).
- A working knowledge of [CALLMaster Manager](#). Specifically how to add, delete and edit records.
- Phone lines connected to CALLMaster and tested.
- Voice Mail Parameters Mailbox Length and Password Length set to 4.
- Email activated and tested. See [Activating Email](#).

### Create Mailbox Records

**Sponsor Mailboxes** - For this walkthrough we will reserve Mailboxes 9900 thru 9999 for Sponsors. The first Sponsor Number will be 9900 and 9905 as the last Sponsor Number. There must be a sequentially numbered mailbox records set-up and used.

For each Mailbox record, the **Mailbox #** will be equal to the Sponsor Number, the **Type** set to Announcement. Each mailbox will have **Settings** UL3 (limit owner options to 3) and NGN (do not require mailbox name and greeting for first time login). Add any other optional information as desired.

**Team Mailboxes** - You can use any other Mailbox range for the Team mailboxes. We will set-up 3 Team mailbox records with **Mailbox #** 1000, 1001, 1002. As we want to give the Team member the option to leave a message, we will set the **Type** to Option to Leave Msg.

**Administrator Mailbox** - Keep Mailbox record 0000 with the default settings as CALLMaster requires it. You should change the password for this record to maintain security.



If the mailbox owner wishes to receive email notification of messages or faxes received in their mailbox, enter their email address in the **Email** field.

Open the **CALLMaster Manager**. Click on **Manage**, then select **Mailboxes**. Let's add the records in this walkthrough as follows:

Mailbox Records							
Mailbox #	Password	Last Name	Settings	Email	Type	Notification Enabled	Notify per Msg
1000	1000	Team 1000	UL3	Team1000@abc.com	Option to Leave Msg	Yes	Yes
1001	1001	Team 1001	UL3	Team1001@cbc.com	Option to Leave Msg		
1002	1002	Team 1002	UL3	Team1000@dcx.com	Option to Leave Msg		
9900	9900	Sponsor 9900	UL3		Announcement		
9901	9901	Sponsor 9901	UL3		Announcement		
9902	9902	Sponsor 9902	UL3		Announcement		
9903	9903	Sponsor 9903	UL3		Announcement		
9904	9904	Sponsor 9904	UL3		Announcement		
9905	9905	Sponsor 9905	UL3		Announcement		

As the mailbox has Setting UL3, only the first 3 [Set Personal Option Features](#) will be available. The changes made will take effect as soon as the owner logs out.

## Verify System Parameters

As we are using Notification, we need to make sure that the System Parameters - Local Dialing Prefix, Long Distance Dialing Prefix, System's Area Code and System's Phone Number have the correct values for your environment.

Open the **CALLMaster Manager**. Click on **Parameters**, then select **System**. Double click on each of the parameters mentioned above to change their values as needed. Remember, CALLMaster will use this information with the Notification data input below to make notification calls. See [System Parameters](#) for

details.

For this walkthrough we will assume the following values:

- **Local Dialing Prefix** - set to 9.
- **Long Distance Dialing Prefix** - set to 9,1,.
- **System's Area Code** - set to 914.
- **System's Phone Number** - set to 5555555

### Create Notification Records

As we have enabled the Notification On indicator for Mailboxes 1000, we need to create Notification record(s) for the mailbox.

Open the **CALLMaster Manager**. Click on **Manage**, then select **Notification** to add the records specified below. For each record, set the Earliest Hour to Call, Latest Hour to Call, Days to Call and the Call Attempt Schedule as desired. See [Notification](#) for details.

Notification Records				
Mailbox #	Record #	Call Type	Area Code	Phone Number
1000	1	Human	914	1111111
1000	2	Human	914	2222222

### Create Audiotex Records/Modules and Associated Scripts

As all calls will be starting in Audiotex mode in this walkthrough, the Audiotex Modules will determine the call flow.

In the Audiotex set-up, the caller will hear a Sponsor message. There will be a sequential rotation thru the list of Sponsors, with each Sponsor having the ability to record their message at any time.

We will need to create several Audiotex records as shown below. The Label field provides a brief description of each module.

Audiotex Records																	
Module ID	Label	Write Label to Log File	Module Action	Next Module ID	Fail Module ID	Response Length	Script File to Run (Must include full path name) (c:\Program	Touch Tone Table									
								1	2	3	4	5	6	7	8	9	0

							Files\C ALLM aster\ Script s\Sam pleAp p-Tea mBulle tinBoa rd\)										
10	Start call and Initial ize value s	Yes	Next Modu le	100	9900	0	Initial izeV al.ba s										
100	Send caller to asso ciate d Spon sor Mailb ox	Yes	Mailb ox	1000	9900	0	GetS pons orNo. bas										
1000	Get fixed lengt h input (Mail box #) from caller	Yes	Next Modu le	1100	9900	0	GetD TMF- FixLe n.bas										
1100	Pres s 1 to hear team infor matio n or 2 if you are a team leade r	Yes	Touc h Tone Table	9999	9900	1		2000	2100	1192	1192	1192	1192	1192	1192	1192	1192

1190	Number must be 4 digits	Yes	Next Module	1191	9900	0	CheckTry Count.bas										
1191	Try again.	Yes	Next Module	1000	9900	0											
1192	Invalid input.	Yes	Next Module	1193	9900	0	CheckTry Count.bas										
1193	Try again.	Yes	Next Module	1100	9900	0											
2000	Send team member to associated Mailbox to hear message	Yes	Mailbox	9998	9900	0	ReturnMailbox.bas										
2100	Send team leader to associated Mailbox to log in	Yes	Login	9998	9900	0	ReturnMailbox.bas										
9900	Call Failure Detected	Yes	Next Module	9999	9999	0	SetCallFailure.bas										
9910	Caller	Yes	Next Module	9999	9999	0	SetCaller										

	Hang-up Detected		le				Hang up.bas										
9920	Maximum tries attempted	Yes	Next Module	9999	9999	0	SetMaxAttempts.bas										
9998	Thank you for using the CALL Master Team Bulletin Board	Yes	Next Module	9999	9999	0											
9999	Last Module - End call	Yes	Last Module														

#### Auditex Module Description

- We start with an Entry.bas script. This script will initialize the Global variable @intSponsorNo to 9900 (the first Sponsor Number). This script is placed in the Scripts folder like all the other scripts, however it is not assigned to any Auditex module. The Entry.bas script will run only when the CALLMaster service starts. We also want to set the system variable lctrp to an Auditex module to send caller if they hang-up. See sample code below.
- Module 10 is the starting module. We will use this module to initialize variables using script InitializeVal.bas. In particular, we need to initialize the variable for the first Sponsor Number (intFirstSponsorNo) and last Sponsor Number (intLastSponsorNo). We will fix the Input Length allowed (intMBInputLen) to be the same as the Mailbox length and initialize a variable strMB to contain the 4 digit number entered by the caller. See sample code below.
- Module 100 selects the next sponsor message to play. The associated script, GetSponsorNo.bas, checks the value of global variable @intSponsorNo to see if it is equal to the last Sponsor Number. If it is then @intSponsorNo is reset to the first Sponsor Number (intFirstSponsorNo); if not then it is incremented by 1. The local variable intCurrSponsorMBNo is set to the global variable @intSponsorNo as the current Sponsor Number. This value is used to send caller to the Sponsor's mailbox via the Return command.
- Module 1000 is used to get the caller's fixed length input using the GetDTMF command. This script uses the StringLength command to make sure the caller has entered the number of digits defined by variable intMBInputLen. If they have not then the call will be forwarded to module 1190 for error processing.
- Module 1100 gives the caller the option to 'press 1 to hear the team message or 2 if you are a team leader'. Using Touch Tone Table, the caller is sent to either module 2000 or 2100. If any other key entered the caller will be forwarded to 1192 for error message and retry.

- Module 1190 is to notify the caller that they have not entered the required number of digits. The associated script, CheckTryCount.bas, gives the caller 3 attempts to make an input. If the caller has not entered a valid input after 3 attempts, processing will be forwarded to module 9920.
- Module 1192 is to notify the caller that they have not entered the required number of digits. The associated script, CheckTryCount.bas, gives the caller 3 attempts to make an input. If the caller has not entered a valid input after 3 attempts, processing will be forwarded to module 9920.
- Module 2000 uses the Return Command to send caller to Mailbox (Module Action) based on the local variable strMB.
- Modules 9900 and 9910 are required to handle call failure and capture caller hang-up respectively. Both these modules can be used to set flags and route processing as required.
- Module 9920 is used to notify caller that they have reached the maximum attempts allowed for a valid input. The script associated with this module can be used to set flags and route processing as required.

Based on the Audiotex modules shown above, we will need 4 scripts as well as the Entry.bas script. Let's make a Scripts sub-folder (Program Files\CALLMaster\Scripts\SampleApp-TeamBulletinBoard folder) to save the custom scripts.

Create the scripts first. Open Notepad and create scripts following code examples below. Remember to save them as .bas type files in the Program Files\CALLMaster\Scripts\SampleApp-TeamBulletinBoard folder.

The Entry.bas script must be save in the root Program Files\CALLMaster\Scripts folder.

#### CALLBasic Script - Entry.bas

[Copy to Clipboard](#)

##### Program

```
*****
' Script: Entry.bas
' Function: Runs at CALLMaster Service start-up. Used to initialize global variables.
*****

Print "CM Entry Script started on "
PrintNL $today

' @intSponsorNo set to the first Sponsor Number
@intSponsorNo = 9900

EndProgram
```

#### CALLBasic Script - InitializeVal.bas

[Copy to Clipboard](#)

##### Program

```
*****
' Script: InitializeVal.bas
' Function: This script is used to initialize values
*****

strYes = "Yes"
strNo = "No"
intYes = 1
intNo = 0
strCallFailure = strNo
strCallerHangup = strNo
strMaxAttempts = strNo
intOne = 1
```

```

' Set retry counter and maximum attempts allowed
intTryCount = 1
intMaxAttemptsAllowed = 3

' Assign the starting and ending sponsor numbers
intFirstSponsorNo = 9900
intLastSponsorNo = 9905
intCurrSponsorNo = 0

' Set intMaxInputLen to include maximum number allowed plus the termination digit
intMBInputLen = 4

' Mailbox record values
MBMatchExists = intNo
MBDBOpen = intNo
MB-intNewMsgCount = 0

' Set the module to return to if caller hangs up
lctrp = 9910

' Convert system variable $channel from integer to string to be used in concatenated print statement.
IntegerToString strchannel $channel

pline = "***** For channel=" & strchannel
pline = pline & " Started initialization process "
PrintNL pline

```

EndProgram

[Copy to Clipboard](#)

#### CALLBasic Script - GetSponsorNo.bas

##### Program

```

*****
' Script: GetSponsorNo.bas
' Function: This script checks the value of global variable @intSponsorNo to create the current Sponsor ID.
*****

' If the global variable @intSponsorNo is greater than the last Sponsor Number (intLastSponsorNo), reset the global variable
the first Sponsor No (intFirstSponsorNo); if not then it is incremented by 1.
If @intSponsorNo = intLastSponsorNo Then
    @intSponsorNo = intFirstSponsorNo
Else
    @intSponsorNo = @intSponsorNo + 1
EndIf

' The local variable intCurrSponsorMBNo is set as the current Sponsor Number.
intCurrSponsorMBNo = @intSponsorNo

' Convert local variable intCurrSponsorMBNo from integer to string to be used in concatenated print statement.
IntegerToString strCurrSponsorMBNo intCurrSponsorMBNo
pline = "***** For channel=" & strchannel
pline = pline & " Caller sent to current Sponsor mailbox "
pline = pline & strCurrSponsorMBNo
PrintNL pline

```

' The Return command is used to send caller to the Sponsor's mailbox.  
Return intCurrSponsorMBNo

EndProgram

#### CALLBasic Script - GetDTMF-FixLen.bas

[Copy to Clipboard](#)

Program

```
*****
' Script: GetDTMF-FixLen.bas
' Function: This script gets a fixed length input from the caller.
*****

' Set up parameters for GetDTMF command
strNoInput = ""
strMBNoInput = ""
tmask = ""
ndigits = intMBInputLen
timeout = 15

GetDTMF strNoInput tmask ndigits timeout

' Get length of input
StringLength intlen strNoInput

pline = "***** For channel=" & strchannel
pline = pline & " strNoInput entered is "
pline = pline & strNoInput
PrintNL pline

' Check caller input. The caller must enter the number of digits defined by variable intMBInputLen.
' If valid input then save number entered

If intlen <> intMBInputLen Then
    pline = "***** For channel=" & strchannel
    pline = pline & " Invalid strNoInput entered. Return to 1190 for error and retry"
    PrintNL pline
    Return 1190
Else
    intTryCount = 0
    strMBNo = strNoInput
EndIf

EndProgram
```

#### CALLBasic Script - CheckTryCount.bas

[Copy to Clipboard](#)

Program

```
*****
' Script: CheckTryCount.bas
' Function: This script checks the number of invalid input attempts.
*****

' If the caller has reached the maximum attempts allowed then the call is routed to module 9920.
' If not then they are returned back to module 1000 to retry input based on Next Module ID.
```



```

If intTryCount > intMaxAttemptsAllowed Then
    Return 9920
Else
    intTryCount = intTryCount + 1
EndIf

' Convert system variable intTryCount from integer to string to be used in concatenated print
statement.
IntegerToString strTryCount intTryCount

pline = "***** For channel=" & strchannel
pline = pline & " TryCount = "
pline = pline & strTryCount
PrintNL pline

EndProgram

```

#### CALLBasic Script - ReturnMailbox.bas

[Copy to Clipboard](#)

```

Program
*****
' Script: ReturnMailbox.bas
' Function: This script uses the Return command with a variable.
'           When used with an Audiotex module whose Module Action is Mailbox,
'           the caller will be forwarded to the mailbox whose value is in variable strMBNo.
*****

Return strMBNo

EndProgram

```

#### CALLBasic Script - SetCallFailure.bas

[Copy to Clipboard](#)

```

Program
*****
' Script: SetCallFailure.bas
' Function: This script sets the strCallFailure variable to yes and prints information in the call log file.
*****

strCallFailure = strYes

'Print information to log file for reference and debugging
pline = "***** Call Failure. *****"
PrintNL pline

EndProgram

```

#### CALLBasic Script - SetCallerHangup.bas

[Copy to Clipboard](#)

```

Program
*****
' Script: SetCallerHangup.bas
' Function: This script sets the strCallFailure variable to yes and prints information in the call log file.

```

\*\*\*\*\*

```
strCallerHangup = strYes
```

```
'Print information to log file for reference and debugging
```

```
pline = "***** Caller Hangup. *****"
```

```
PrintNL pline
```

```
EndProgram
```

**CALLBasic Script - SetMaxAttempts.bas**

[Copy to Clipboard](#)

Program

\*\*\*\*\*

```
' Script: SetMaxAttempts.bas
```

```
' Function: This script sets the strMaxAttempts variable to yes and prints information in the call log file.
```

\*\*\*\*\*

```
strMaxAttempts = strYes
```

```
'Print information to log file for reference and debugging
```

```
pline = "***** For channel=" & strchannel
```

```
pline = pline & "***** Maximum Attempts reached to enter valid input *****"
```

```
PrintNL pline
```

```
EndProgram
```

Now you can create the Audiotex modules and reference the scripts above. Open the **CALLMaster Manager**, click on **Manage**, then select **Audiotex** to add the Audiotex records shown above.

### Define IVR Application List Entry

We need to define the custom IVR in the IVR Application List and set it to Audiotex Start Module 10. Open CALLMaster Manager and select **Manage | IVR Application List**. Let's add a new record as follows:

IVR Application List	
Audiotex Start Module	Application Name
10	Team Bulletin Board Sample

### Set Ports

As a last step, we need set the port(s) to the new IVR Application entry created above. Make sure the Audiotex Start Module field (read only) correctly points to module 10.

As this walkthrough uses Notification, make sure at least one port is enabled for Type Outbound; by default, all the ports are enabled for Inbound. Set the Port Greeting File IDs to the main greeting for all calls if you have recorded one. If you do not want a main greeting then set the Greeting File IDs to zero. Leave all other fields to the default values.

If you are working with a demo license or have only one line connected, then make sure the connected port is set as specified below. All other ports should be set to Application 'Disabled'.

Port Records				
Port #	Application	Audiotex Start Module	Port Type Outbound	Port Greeting File IDs (Morning, Afternoon, Evening, Closed, Holidays)
1	Team Bulletin Board Sample	10	No	0, 0, 0, 0, 0
2	Team Bulletin Board Sample	10	No	0, 0, 0, 0, 0
3	Team Bulletin Board Sample	10	Yes	0, 0, 0, 0, 0
4	Team Bulletin Board Sample	10	Yes	0, 0, 0, 0, 0

### Test the custom application

Save all your work.

Stop CALLMaster service.

Make Audiotex Recordings:

- Set Port 1 to Application Voice Mail.
- Start CALLMaster service.
- Make a call using Port 1.
- [Log in](#) as the Administrator using the Voice Mail Main Menu and make recordings for the Audiotex modules below.

Audiotex Module ID	Proposed Recording
1000	Please enter your 4 digit team id.
1100	Press 1 to hear team information or 2 if you are a team leader
1190	Team ID must be 4 digits
1191	Please try again
1192	Invalid input

1193	Please try again
9900	We are unable to process your call. Please contact the system administrator for further assistance.
9920	You have reached the maximum attempts allowed for a valid input. Please contact the system administrator for further assistance.
9998	Thank you for using the CALLMaster Team Bulletin Board

Tip: See [Walkthrough: Change Application Without Restarting CALLMaster Service](#) to set port to Voice Mail for ongoing recording changes.

Reset Port 1 to Application Audiotex. Set the Port Greeting File IDs to zeros.

Stop and Start CALLMaster service.

Make a phone call to test the application.

Make Mailbox Greetings.

- The Team leader and Sponsor can call in to the system and use option 2 to record team messages and retrieve messages.

To download and install, see [Sample Applications](#).

# System Prompts

## Overview

The system prompts are shipped with the CALLMaster software. They contain over 500 professionally recorded prompts and are the default voice of CALLMaster. This section of the manual lists each prompt with its Prompt Number and Text Spoken.

Prompts are stored as individual files in separate subdirectories to provide fast disk access and user flexibility. Up to fifty prompts are stored under a subdirectory. One can determine the path and filename of a prompt from its prompt number. Integer division of the prompt number by 50 yields the subdirectory name, and modulo 50 of the prompt number gives the filename. For instance, prompt 2 is \CALLMaster\pmt\0\2 because integer division of 2 by 50 equals 0, with a remainder of 2. Prompt 53 is \CALLMaster\pmt\1\3; 53 divided by 50 is 1, with a remainder of 3. An ellipsis ( . . . ) indicates that the prompt will be spoken in combination with other prompts; the position of the ellipsis before or after the prompt shows where the other prompt will be. For instance, ' . . . hundred . . . ' could be combined with '...one...' to make 'one hundred' or even 'one hundred twenty-one.' It is important to have a neutral inflection so that prompts flow together well. It is also important to have as little 'dead air' as possible so prompts flow together quickly.

System prompts can be replaced directly by recording over the telephone or by manually replacing the audio files with custom prerecorded files.

[Replacing System Prompts and Other Audio Recordings](#)

[Prompts in Numerical Order](#)

[Prompts in Alphabetical Order](#)

## **Replacing System Prompts and Other Audio Recordings**

[Direct Recording Over the Telephone](#)

[Manually Replacing CALLMaster Audio Files with Custom Recordings](#)

## Replacing CALLMaster Audio Files with Telephone Recordings

CALLMaster audio files can be replaced by directly re-recording over the existing recording via the telephone.

### To change System prompts and Audiotex Messages:

1. Call the system and log-in as the Administrator;
2. At the main mailbox menu, 'Press 1 to retrieve messages, 2 to send a message or 3 to set personal options', Press '\*' for Administrator Options. You will not be prompted for this option;
3. Select Option 7 to record system prompts, Option 8 to record Audiotex message or Option 0 to record ASR Barge-In messages;
4. If you selected Option 7 or 0, the system will ask you to enter the prompt number You can enter a standard prompt number (see listing at the end of this section for standard Prompt Numbers and text) or a custom prompt # between 700 and 999. If you have selected Option 8, the system will ask you to enter the Audiotex module number.
5. Enter the number of the prompt or the Audiotex module number, followed by '\*'. (e.g. if you wish to replace Prompt #6, enter '6\*'; if you wish to record a message for Audiotex module 1000, enter '1000\*');
6. The system will play the current prompt/Audiotex message, if one exists, and then ask you to re-record.
7. Say your new recording and press star (\*) to terminate the recording.
8. You will be given the option to 'press 1 to accept, 2 to replay or 3 to redo'. You must Press 1 to accept the newly recorded message.

### To change Mailbox names and greetings:

1. Call the system and log into your mailbox;
2. At the main mailbox menu, 'Press 1 to retrieve messages, 2 to send a message or 3 to set personal options', select '3' to set personal options.
3. Select Option 2 to record your greeting or Option 3 to record your name;

## Replacing CALLMaster Audio Files with Custom Recording Files

You can overlay any and all audio recordings in CALLMaster with custom recordings from any other source. In general wave format files recorded in the '11.025KHz Linear 8bit mono' format can be simply copied over to the appropriate folders and they will replace the existing recording, if your hardware supports wave and the system parameter 'Recording Format' is set to 'wav'.

First backup your system before proceeding.

Stop CALLMaster via 'Control Panel' or 'Administrative Tools' in Win 2000 \ 'Services Window' (or Task Manager if CALLMaster is not configured to run as a service).

Backup the current CALLMaster folder.

Select 'CALLMaster' folder in the 'Program Files' folder right click and choose 'Copy' then Select the 'Program Files' folder right click and choose 'Paste'. A folder called 'Copy of CALLMaster' will be created and contain all the files needed for CALLMaster to be backed up. You can save that folder off-line for permanent backup or archive.

Recordings are kept in the following folders:

- [System Prompts](#) are in the 'CALLMaster/pmt' folder
- [Mailbox Greetings, Names, and Audiotex Message Files](#) are stored in the 'CALLMaster/grt' folder
- [Voicemail message recordings and scripted record files](#) are in the 'CALLMaster/vox' folder.



## System Prompts

These files are in the 'CALLMaster/Pmt' folder.

They are subdivided into groups of 50 files in subfolders named '0' through '11' (up to 19 is allowed).

The system prompts are shipped with the CALLMaster software. They contain over 500 professionally recorded prompts and are the default voice of CALLMaster. This section of the CALLMaster manual lists each prompt with its Prompt Number and Text Spoken. The highest allowed prompt number allowed is 999 (folder 19).

### Prompt Number to File Name translation

To translate between a Prompt number in the manual and an actual file on your computer use the following algorithm:

- o The subfolder number is the whole number part of the Prompt number divided by 50.
  - Formula:  $\text{Subfolder\#} = \text{WH}$  where WH is the whole number part of  $\text{Prompt\#} / 50$
  - Example: Prompt# 175 is in subfolder '3' of folder 'pmt' ('pmt/3') because 3 is the whole number part of  $175/50 = 3.5$
- o The file number for a prompt is the Prompt# minus the subfolder number multiplied by 50.
  - Formula:  $\text{File\#} = \text{Prompt\#} - (\text{subfolder\#} * 50)$
  - Example: Prompt# 175 will be a file named '25' (25.wav if wave is enabled) in the pmt\3 folder ( $175 - (3*50) = 25$ )

So prompt number 175 is file 'C:/Program Files/CALLMaster/pmt/3/25' (25.wav if extension is used)

If you have enabled CALLMaster for multi-lingual support then you should create sub-folder for each language identifier.

See [Appendix E – Multi-Lingual Support](#) for details.

## Mailbox Greetings, Names and Audiotex Module Recording Files

All Mailbox name and greeting recordings, all Attendant name and alternate message recordings, and all Audiotex module message recordings are stored in the 'grt' folder.

These recordings are identified by their IDs, a number from 1000 through 99,999. They are subdivided into groups of 120 files in subfolders named 8 and higher as needed.

- Mailboxes have Greeting File IDs and Name File IDs in each mailbox record. Open CALLMaster Manager and select **Manage | Mailboxes** to see the value for any mailbox.
- Attendant records have Name File IDs (the extension name e.g. Sales), and Alternate Message IDs used when the attendant action is 'Play Alternate Message' instead of transferring.
- Audiotex records have Message IDs. This audio file will be played when that Audiotex module is reached.

### Audio Message ID to File Name translation

To translate between an audio message ID number and an actual file on your computer use the following algorithm:

- o The subfolder number is the whole number part of the MSGID# divided by 120.
  - Formula: Subfolder# = WH where WH is the whole number part of MSGID# divided by 120.
  - Example: MSGID 1017 is in subfolder '8' of folder 'grt' ('grt/8') because 8 is the whole number part of  $1017/120 = 8.475$
- o The file number for a MSGID is the MSGID# minus the subfolder# multiplied by 120.
  - Formula: File# = MSGID# minus (subfolder# multiplied by 120)
  - Example: MSG ID# 1017 will be in a file named '57' (57.wav if wave is enabled) in the grt\8 folder ( $1017 - (8 \times 120) = 57$ )

So an Audiotex Message ID number 1017 is file 'C:/Program Files/CALLMaster/grt/8/57' (57.wav if extension is used)

If you have enabled CALLMaster for multi-lingual support then you should create sub-folder for each language identifier.

See [Appendix E – Multi-Lingual Support](#) for details.

## Voice Mail Messages And CALLBasic Script Recordings

These files are in the 'CALLMaster/Vox' folder.

They are subdivided into groups of 120 files in subfolders named 833 and higher as needed.

All mailbox messages have a MSG ID and the recordings are stored in the 'Vox' folder. The values for messages are 100,000 and higher.

To translate between a voice message MSG ID number and an actual file on your computer use the following algorithm:

- o **The subfolder number is the whole number part of the MSGID# divided by 120.**
  - Formula: Subfolder# = WH where WH is the whole number part of MSGID# divided by 120
  - Example: MSGID 100,029 is in subfolder '833' of folder 'Vox' because 833 is the whole number part of  $100029/120 = 833.455$
- o **The file number for a MSGID is the MSGID# minus the subfolder# multiplied by 120.**
  - Formula: File# = MSGID# - (subfolder# \* 120)
  - Example: MSG ID# 100029 will be in file '69' (69.wav if wave enabled) in the Vox\833 folder **(100029 – (833 x120)) = 69**

So a voice mail message MSG ID number 100,029 is file 'C:/Program Files/CALLMaster/Vox/833/69' (69.wav if is used)

## System Prompts in Numerical Order

Number	Text
--------	------

0	(null)
2	At the tone, record your message for . . .
3	Press one to accept, two to replay, or three to redo. (Note: four to append.)
4	Press four to play the previous message, five to play this message again, six to play the next message, seven to save, nine to delete, zero to reply, or star for more options.
5	NU - Not Used
6	Box number, please.
7	Enter your password, then press star.
8	Press one to review messages, two to send a message, or three to set personal options.
9	To which box?
10	Please enter a box number, or press two for a directory. Press star to log in, or hold on to leave a message.
11	I'm sorry you're having trouble. Please try from another phone, or call back later.
12	Press any key if you'd like to continue your session.
13	I can not hear your response; Thank you, good-bye.
14	Paused.
15	Your box number or password was incorrect, please re-enter your box number.
16	Saved.
17	Deleted.
18	Press seven to review saved messages, eight to review new messages, or nine to review deleted messages.
19	Good morning, you've reached SpeechSoft voice mail.
20	Good afternoon, you've reached SpeechSoft voice mail.

21	Good evening, you've reached SpeechSoft voice mail.
22	NU - Not used
23	One to lower volume; Two for message count; three to raise volume; four for the first message; five for time, date and sender; six for the last message; seven for saved messages; eight for new messages; nine for deleted messages; or zero to copy to another box.
24	zero . . .
25	one . . .
26	two . . .
27	three . . .
28	four . . .
29	five . . .
30	six . . .
31	seven . . .
32	eight . . .
33	nine . . .
34	ten
35	eleven . . .
36	twelve . . .
37	thirteen . . .
38	fourteen . . .
39	fifteen . . .
40	sixteen . . .
41	seventeen . . .

42	eighteen . . .
43	nineteen . . .
44	twenty . . .
45	thirty . . .
46	forty . . .
47	fifty . . .
48	sixty . . .
49	seventy . . .
50	eighty . . .
51	ninety . . .
52	hundred . . .
53	Your password is . . .
54	Your new password is . . .
55	I'm sorry, there is no such box.
56	Press one to change your password, two to record your greeting, three to record your name, four to maintain distribution lists, five to change your directory listing, six to turn notification on or off, or seven to change notification settings. Eight to record audiotex messages, or Nine to edit attendant records
57	Your greeting is . . .
58	Your name is . . .
59	One to create a box, two to delete, three to change a box's settings, four for a box status report, five to change default box settings, six for a system status report, seven to change prompts, eight to record audiotex modules, or nine for attendant directory.
60	I'm sorry, that box already exists.
61	The new box number is . . .
62	a.m.

63	p.m.
64	January . . .
65	February . . .
66	March . . .
67	April . . .
68	May . . .
69	June . . .
70	July . . .
71	August . . .
72	September . . .
73	October . . .
74	November . . .
75	December . . .
76	I'm sorry, that box is full and cannot accept any more messages.
77	You have reached your saved message limit, so you cannot save this message.
78	Press one to confirm deletion.
79	You have . . .
80	Press star to leave a message.
81	NU - Not Used
82	One to change new message capacity, two to change saved message capacity, three to change maximum message length, four to change maximum message age five for a box status, or six to change notification options.
83	The new-message capacity is . . .

84	The saved-message capacity is . . .
85	The maximum message length is . . .
86	The maximum message age is . . .
87	Enter the new value, then press star.
88	You entered . . .
89	Welcome to voice mail. if you know which key to press, you may press it at any time if you make an error or need to back up, press the pound key. Now, record your personal greeting.
90	Thousand . . .
91	Million
92	At the tone, say your name.
93	At the tone, record your personal greeting.
94	At the tone, record your reply to . . .
95	Box created.
96	Message sent.
97	Reply sent.
98	Copy sent.
99	At the tone record an introduction for the message you are forwarding to . . .
100	Press star to stop recording.
101	Begin recording at the tone.
102	I'm sorry, the password you entered is incorrect, please re-enter your password then press star.
103	Copy to which box?
104	NU - Not Used
105	You cannot send yourself a copy.



106 Press star to send this message to another box.

107 NU - Not Used

108 Sending this message to . . .

109 Box . . .

110 belongs to . . .

111 and has . . .

112 The message capacity is . . .

113 The maximum message age is . . .

114 days.

115 The maximum message length is . . .

116 seconds.

117 your system is . . .

118 percent free.

119 lines are active.

120 calls have been processed.

121 errors have occurred.

122 first . . .

123 second . . .

124 third . . .

125 fourth . . .

126 fifth . . .

127 sixth . . .

128	seventh . . .
129	eighth . . .
130	ninth . . .
131	tenth . . .
132	eleventh . . .
133	twelfth . . .
134	thirteenth . . .
135	fourteenth . . .
136	fifteenth . . .
137	sixteenth . . .
138	seventeenth . . .
139	eighteenth . . .
140	nineteenth . . .
141	twentieth . . .
142	thirtieth . . .
143	fortieth . . .
144	fiftieth . . .
145	sixtieth . . .
146	seventieth . . .
147	eightieth . . .
148	ninetieth . . .
149	hundredth . . .

150	Enter the public list number.
151	Enter the private list number, or press star for a public list.
152	I'm sorry, there is no such list.
153	This list is empty.
154	At the tone, record a name for this list.
155	One to create a list, two to edit, three to delete, or four to review lists.
156	Enter a three-digit number to identify the list.
157	NU - List created.
158	Enter the list number to delete.
159	in this list.
160	does not exist.
161	One to add boxes, two to edit boxes, or three to delete boxes.
162	Enter the box number to add.
163	Enter the box number to delete.
164	Added.
165	Enter the list number to edit.
166	is not in this list.
167	Four for the previous box, five for the current box, six for the next, seven to add, or nine to delete the current box.
168	This is the last box in the list.
169	Review of . . .
170	One to review your private lists, or star to review public lists.
171	private lists.

172 public lists.

173 Four for the previous list, five for the current list, or six for the next list.

174 No public lists.

175 No private lists.

176 Enter the prompt number, then press star.

177 Record prompt . . .

178 You may not . . .

179 create a public distribution list.

180 modify a public distribution list.

181 delete a public distribution list.

182 That list exists already.

183 is already in the list.

184 One to log in as another user, two to shut down the system, or three to deactivate.

185 The system is full! No messages can be delivered!

186 NU - Not Used

187 I'm sorry, that message is private, and may not be copied.

188 Message was not sent. Press 1 to try again.

189 oh . . .

190 no . . .

191 This is . . .

192 That was . . .

193 your first . . .

194	your last . . .
195	new message.
196	new messages.
197	saved message.
198	saved messages.
199	deleted message.
200	deleted messages.
201	messages.
202	and . . .
203	Hello, you've reached SpeechSoft Voice mail; we're closed today.
204	NU - Not Used
205	Spell your last name using the telephone keys, then press the star key; for "q" press seven, for "z" press nine.
206	Spell your first name using the telephone keys, then press star; for "q" press seven, for "z" press nine.
207	from . . .
208	NU - Not used
209	Hello, you've reached SpeechSoft voice mail; we're closed for the holiday.
210	NU - One to create an attendant record, two to delete an attendant record, three to change an attendant record, four to change a phone port, five to change the system date, six to change the system time, seven to change the system parameters, eight to change global tone detection, or nine to shut down the system.
211	NU - You have reached a non-working extension.
212	NU - You selected . . .
213	you entered
214	NU - Password changed.

215	NU - Greeting changed.
216	NU - Name changed.
217	NU - to continue recording, press four.
218	Thank you for calling, good-bye.
219	NU - Box number to create?
220	Box number to delete?
221	NU - Box number to change?
222	NU - Box number to report on?
223	NU - The new, new-message capacity is . . .
224	NU - The new saved-message capacity is . . .
225	NU - The new maximum message length is
226	NU - The new maximum message age is . . .
227	NU - line is active.
228	NU - call has been processed.
229	error has occurred.
230	NU - full.
231	Use the telephone keys to spell some or all of the last name of the person you are looking for, then press star, for "q" press seven, for "z" press nine.
232	Press star to use this listing, one to hear the number for this listing, four for the previous listing, five to hear this listing again, or six for the next listing.
233	matching directory entry.
234	Your directory listing is...
235	NU - Not Used

236	NU - Not Used
237	NU - Not Used
238	NU - Not Used
239	Message too short
240	NU - Not Used
241	Enter the module key, then press star.
242	There is no such module.
243	NU - Not Used
244	Record module . . .
245	Press star to repeat.
246	Notification is . . .
247	on.
248	off.
249	Enter the notification record number.
250	Enter the phone number, or press star to leave as is.
251	Enter the area code, or star to leave as is.
252	Enter the latest hour to call, from zero zero for twelve midnight, to twenty-three for eleven p.m., or star to leave as is.
253	Enter the earliest hour to call, from zero zero for twelve midnight, to twenty-three for eleven p.m., or star to leave as is.
254	NU - The number is . . .
255	Invalid notification record.
256	NU - Not used

257	Press one to change.
258	Hello, this is SpeechSoft voice mail, with a message for . . .
259	o'clock . . .
260	The earliest hour to call is . . .
261	The latest hour to call is . . .
262	That option is not available.
263	Press one for further voice mail options.
264	NU - Not used
265	NU - Not used
266	The alternate message is . . .
267	At the tone, record the alternate message.
268	The directory listing is . . .
269	At the tone, record the directory listing.
270	Enter the extension ID, then press star.
271	Enter the part of day: zero for all day, one for morning, two for afternoon, or three for evening.
272	No such extension record.
273	NU - Not used
274	NU - Not used
275	NU - Not used
276	NU - Not used
277	NU - Not used
278	NU - Not used



279	NU - Not used
280	NU - Not used - This is your first new message
281	NU - Not used - This is your first saved message
282	NU - Not used - This is your first deleted message
283	NU - Not used - That was your first new message
284	NU - Not used - That was your first saved message
285	NU - Not used - That was your first deleted message
286	NU - Not used - This is your last new message
287	NU - Not used - This is your last saved message
288	NU - Not used - This is your last deleted message
289	NU - Not used - That was your last new message
290	NU - Not used - That was your last saved message
291	NU - Not used - That was your last deleted message
292	NU - Not used
293	NU - Not used
294	The current box status is ...
295	deleted.
296	deactivated.
297	announcement.
298	user.
299	menu announcement.
300	superuser.

301	maintenance.
310	During playback the following options are available: one to rewind, two to pause, three to skip forward, four for the previous recording, five to repeat, six for the next recording, zero to reply or pound (#) to exit.
311	four for the previous, five to repeat, six for the next recording zero to reply or pound (#) to exit.
320	Global List
321	Please dial your call back number then press star.
325	Fax
326	Faxes
327	Press one to start fax transmission now, two to send this fax to another number, four to play previous fax header, five to play the current fax, six to play the next fax, seven to save or nine to delete fax.
328	Press start on your fax machine now.
329	Your Fax could not be sent.
330	Your Fax has been scheduled.
331	This hardware port does not support Faxing, check with your system administrator.
332	Failed
333	The status of this fax is failed and it cannot be forwarded.
334	Enter the full number to send this fax to including any access codes for pause press star. Press pound when done.
335	NU - Not used
336	No Faxes
337	Status
338	Received OK
339	Receive Failed
340	New Faxes

341	Saved Faxes
342	Deleted Faxes
343	New Fax
344	Saved Fax
345	Deleted Fax
346	Press seven to review saved faxes, eight to review new faxes or nine to review deleted faxes
350	Press one to change the type of transfer, two for the Extension to dial, three for three-way code, four for Alternate-Busy Action, five for Alternate-No Answer Action, six for Alternate-Screened Reject Action, seven for Alternate-Attendant ID, eight for Mailbox number, nine for Audiotex Key or star for Directory and Record Options.
351	Press one to record the Alternate Message, two to record the directory listing or three to change the directory entry
352	Enter the transfer type then pres star. Press one for transfer, two for Supervised, three for Screened, four for Blocked, five for Directory, six for Mailbox, seven for Voicemail, eight for Audiotex, nine for Audiotex Key, ten for Alternate Message, eleven for Log in to mailbox, or twelve for Disabled
353	Transfer
354	Supervised
355	Screened
356	Blocked
357	Directory
358	Mailbox
359	Audiotex
360	Audiotex Key
361	Alternate Message
362	Log in to mailbox
363	Disabled

364	Setting To ..
365	You do not have permission to change this attendant record. Speak to your administrator.
366	Press one for Default Message, two for Alternate Attendant ID, three for Mailbox, four for Mailbox Menu, five for Alternate Message, six for Retry, seven for Audiotex, and eight for Audiotex key.
367	Default Message
368	Alternate Attendant ID
369	Mailbox
370	Mailbox Menu
371	Alternate Message
372	Retry
373	Audiotex
374	Audiotex key
375	Enter the extension to dial. For pause press Star. Press Pound when done.
376	Enter the three-way code. For pause press Star. Press Pound when done.
377	Enter the Alternate Attendant ID Then Press Star
378	Enter the mailbox number
379	Enter the Audiotex Key
380	Press one to accept, or two to reject a call from CALLMaster.
381	Dollar
382	Dollars
383	Cent
384	Cents
385	and

386	point
387	Please enter a box number or press two for a directory, press star to log in or hold on to leave a message. If you want to fax a document, press start on your fax machine now.
388	Press one to rewind, two to Pause and Resume, three to Fast Forward, or any other key to terminate playback.
389	Press one to review messages, two to send a message, three to set personal options, or four to review faxes.
390	Negative
391	Mailbox busy. Please try again later.
392	Press one to change the LanguageID
393	Enter the language ID
394	The language ID is
395	You must record the English message for this module before any other language
396	That language is not defined
501	Please hold while I transfer your call
502	Good morning, you've reached the SpeechSoft attendant.
503	Good afternoon, you've reached the SpeechSoft attendant.
504	Good evening, you've reached the SpeechSoft attendant.
505	You may enter an extension, or hold on for assistance. (This should be where you offer choices to your callers)
506	Sorry, there is no such extension.
507	May I ask who is calling?
508	Hello, would you like to take a call from . . .
509	Press one to accept, two to reject, or three to hear again.
510	Extension . . .

511 can not be reached.

512 Press any key if you'd like to continue your session.

513 I can not hear your response; Thank you, good-bye.

514 Sorry, I can not get through. Please hold for further assistance.

515 Hello, you've reached the SpeechSoft attendant; we're closed today.

516 Hello, you've reached the SpeechSoft attendant; we're closed for the holiday.

518 That option is not available.

519 Press one for further attendant options.

520 Press star to leave a message, or enter another extension.

521 Please hold while I transfer you to ...

524 zero . . .

525 one . . .

526 two . . .

527 three . . .

528 four . . .

529 five . . .

530 six . . .

531 seven . . .

532 eight . . .

533 nine . . .

534 ten

535 eleven . . .

536	twelve . . .
537	thirteen . . .
538	fourteen . . .
539	fifteen . . .
540	sixteen . . .
541	seventeen . . .
542	eighteen . . .
543	nineteen . . .
544	twenty . . .
545	thirty . . .
546	forty . . .
547	fifty . . .
548	sixty . . .
549	seventy . . .
550	eighty . . .
551	ninety . . .
552	hundred . . .
560	Bridge Call - Prompt to be played while call is being bridged. This prompt needs to be recorded. It can contain music or promotion to be played or a simple on-hold message "Please hold while you are connected".
561	Your connection could not be made (for Bridge Call)
562	Press one for more options (played after Bridge Call Removed)
563	Two minutes remaining for this call (for Bridge Call)
564	You may now start your call. (for Bridge Call)

565        How may I direct your call?

566        < Optional Tone to indicate start speaking > (to be recorded if desired)

567        Would you like further assistance? Press 1 to continue.

568        No match found. Would you like to try again?

569        Did you mean ....



## System Prompts in Alphabetical Order

Number	Text
--------	------

0	(null)
---	--------

566	< Optional Tone to indicate caller should start speaking > (to be recorded if desired)
-----	--

62	a.m.
----	------

164	Added.
-----	--------

368	Alternate Attendant ID
-----	------------------------

361	Alternate Message
-----	-------------------

371	Alternate Message
-----	-------------------

385	and
-----	-----

202	and . . .
-----	-----------

111	and has . . .
-----	---------------

297	announcement.
-----	---------------

67	April . . .
----	-------------

99	At the tone record an introduction for the message you are forwarding to . . .
----	--

154	At the tone, record a name for this list.
-----	---

267	At the tone, record the alternate message.
-----	--

269	At the tone, record the directory listing.
-----	--

2	At the tone, record your message for . . .
---	--

93	At the tone, record your personal greeting.
----	---

94	At the tone, record your reply to . . .
----	---

92	At the tone, say your name.
----	-----------------------------

359	Audiotex
373	Audiotex
360	Audiotex Key
374	Audiotex key
71	August . . .
101	Begin recording at the tone.
110	belongs to . . .
356	Blocked
109	Box . . .
95	Box created.
220	Box number to delete?
6	Box number, please.
560	Bridge Call - Prompt to be played while call is being bridged. This prompt needs to be recorded. It can contain music or promotion to be played or a simple on-hold message "Please hold while you are connected".
120	calls have been processed.
511	can not be reached.
383	Cent
384	Cents
98	Copy sent.
103	Copy to which box?
179	create a public distribution list.
114	days.
296	deactivated.

75	December . . .
367	Default Message
181	delete a public distribution list.
345	Deleted Fax
342	Deleted Faxes
199	deleted message.
200	deleted messages.
17	Deleted.
295	deleted.
569	Did you mean ....
357	Directory
363	Disabled
160	does not exist.
381	Dollar
382	Dollars
310	During playback the following options are available: one to rewind, two to pause, three to skip forward, four for the previous recording, five to repeat, six for the next recording, zero to reply or pound (#) to exit.
32	eight . . .
532	eight . . .
42	eighteen . . .
542	eighteen . . .
139	eighteenth . . .
129	eighth . . .

147	eightieth . . .
50	eighty . . .
550	eighty . . .
35	eleven . . .
535	eleven . . .
132	eleventh . . .
156	Enter a three-digit number to identify the list.
377	Enter the Alternate Attendant ID Then Press Star
251	Enter the area code, or star to leave as is.
379	Enter the Audiotex Key
162	Enter the box number to add.
163	Enter the box number to delete.
253	Enter the earliest hour to call, from zero zero for twelve midnight, to twenty-three for eleven p.m., or star to leave as is.
270	Enter the extension ID, then press star.
375	Enter the extension to dial. For pause press Star. Press Pound when done.
334	Enter the full number to send this fax to including any access codes for pause press star. Press pound when done.
393	Enter the language ID
252	Enter the latest hour to call, from zero zero for twelve midnight, to twenty-three for eleven p.m., or star to leave as is.
158	Enter the list number to delete.
165	Enter the list number to edit.
378	Enter the mailbox number

241	Enter the module key, then press star.
87	Enter the new value, then press star.
249	Enter the notification record number.
271	Enter the part of day: zero for all day, one for morning, two for afternoon, or three for evening.
250	Enter the phone number, or press star to leave as is.
151	Enter the private list number, or press star for a public list.
176	Enter the prompt number, then press star.
150	Enter the public list number.
376	Enter the three-way code. For pause press Star. Press Pound when done.
352	Enter the transfer type then pres star. Press one for transfer, two for Supervised, three for Screened, four for Blocked, five for Directory, six for Mailbox, seven for Voicemail, eight for Audiotex, nine for Audiotex Key, ten for Alternate Message, eleven for Log in to mailbox, or twelve for Disabled
7	Enter your password, then press star.
229	error has occurred.
121	errors have occurred.
510	Extension . . .
332	Failed
325	Fax
326	Faxes
65	February . . .
39	fifteen . . .
539	fifteen . . .
136	fifteenth . . .

126	fifth . . .
144	fiftieth . . .
47	fifty . . .
547	fifty . . .
122	first . . .
29	five . . .
529	five . . .
143	fortieth . . .
46	forty . . .
546	forty . . .
28	four . . .
528	four . . .
167	Four for the previous box, five for the current box, six for the next, seven to add, or nine to delete the current box.
173	Four for the previous list, five for the current list, or six for the next list.
311	four for the previous, five to repeat, six for the next recording zero to reply or pound (#) to exit.
38	fourteen . . .
538	fourteen . . .
135	fourteenth . . .
125	fourth . . .
207	from . . .
320	Global List
20	Good afternoon, you've reached SpeechSoft voice mail.

503	Good afternoon, you've reached the SpeechSoft attendant.
21	Good evening, you've reached SpeechSoft voice mail.
504	Good evening, you've reached the SpeechSoft attendant.
19	Good morning, you've reached SpeechSoft voice mail.
502	Good morning, you've reached the SpeechSoft attendant.
258	Hello, this is SpeechSoft voice mail, with a message for . . .
508	Hello, would you like to take a call from . . .
209	Hello, you've reached SpeechSoft voice mail; we're closed for the holiday.
203	Hello, you've reached SpeechSoft Voice mail; we're closed today.
516	Hello, you've reached the SpeechSoft attendant; we're closed for the holiday.
515	Hello, you've reached the SpeechSoft attendant; we're closed today.
565	How may I direct your call?
52	hundred . . .
552	hundred . . .
149	hundredth . . .
13	I can not hear your response; Thank you, good-bye.
513	I can not hear your response; Thank you, good-bye.
11	I'm sorry you're having trouble. Please try from another phone, or call back later.
60	I'm sorry, that box already exists.
76	I'm sorry, that box is full and cannot accept any more messages.
187	I'm sorry, that message is private, and may not be copied.
102	I'm sorry, the password you entered is incorrect, please re-enter your password then press star.

55	I'm sorry, there is no such box.
152	I'm sorry, there is no such list.
159	in this list.
255	Invalid notification record.
183	is already in the list.
166	is not in this list.
64	January . . .
70	July . . .
69	June . . .
119	lines are active.
362	Log in to mailbox
358	Mailbox
369	Mailbox
391	Mailbox busy. Please try again later.
370	Mailbox Menu
301	maintenance.
66	March . . .
233	matching directory entry.
68	May . . .
507	May I ask who is calling?
299	menu announcement.
96	Message sent.



239	Message too short
188	Message was not sent. Press 1 to try again.
201	messages.
91	Million
180	modify a public distribution list.
390	Negative
343	New Fax
340	New Faxes
195	new message.
196	new messages.
33	nine . . .
533	nine . . .
43	nineteen . . .
543	nineteen . . .
140	nineteenth . . .
148	ninetieth . . .
51	ninety . . .
551	ninety . . .
130	ninth . . .
190	no . . .
336	No Faxes
568	No match found. Would you like to try again?

175	No private lists.
174	No public lists.
272	No such extension record.
246	Notification is . . .
74	November . . .
221	NU - Box number to change?
219	NU - Box number to create?
222	NU - Box number to report on?
228	NU - call has been processed.
230	NU - full.
215	NU - Greeting changed.
227	NU - line is active.
157	NU - List created.
216	NU - Name changed.
5	NU - Not Used
22	NU - Not used
81	NU - Not Used
104	NU - Not Used
107	NU - Not Used
186	NU - Not Used
204	NU - Not Used
208	NU - Not used

235	NU - Not Used
236	NU - Not Used
237	NU - Not Used
238	NU - Not Used
240	NU - Not Used
243	NU - Not Used
256	NU - Not used
264	NU - Not used
265	NU - Not used
273	NU - Not used
274	NU - Not used
275	NU - Not used
276	NU - Not used
277	NU - Not used
278	NU - Not used
279	NU - Not used
292	NU - Not used
293	NU - Not used
335	NU - Not used
291	NU - Not used - That was your last deleted message
289	NU - Not used - That was your last new message
290	NU - Not used - That was your last saved message

285	NU - Not used - That was your first deleted message
283	NU - Not used - That was your first new message
284	NU - Not used - That was your first saved message
282	NU - Not used - This is your first deleted message
280	NU - Not used - This is your first new message
281	NU - Not used - This is your first saved message
288	NU - Not used - This is your last deleted message
286	NU - Not used - This is your last new message
287	NU - Not used - This is your last saved message
210	NU - One to create an attendant record, two to delete an attendant record, three to change an attendant record, four to change a phone port, five to change the system date, six to change the system time, seven to change the system parameters, eight to change global tone detection, or nine to shut down the system.
214	NU - Password changed.
226	NU - The new maximum message age is . . .
225	NU - The new maximum message length is
224	NU - The new saved-message capacity is . . .
223	NU - The new, new-message capacity is . . .
254	NU - The number is . . .
217	NU - to continue recording, press four.
211	NU - You have reached a non-working extension.
212	NU - You selected . . .
259	o'clock . . .
73	October . . .

248	off.
189	oh . . .
247	on.
25	one . . .
525	one . . .
161	One to add boxes, two to edit boxes, or three to delete boxes.
82	One to change new message capacity, two to change saved message capacity, three to change maximum message length, four to change maximum message age five for a box status, or six to change notification options.
59	One to create a box, two to delete, three to change a box's settings, four for a box status report, five to change default box settings, six for a system status report, seven to change prompts, eight to record audiotex modules, or nine for attendant directory.
155	One to create a list, two to edit, three to delete, or four to review lists.
184	One to log in as another user, two to shut down the system, or three to deactivate.
23	One to lower volume; Two for message count; three to raise volume; four for the first message; five for time, date and sender; six for the last message; seven for saved messages; eight for new messages; nine for deleted messages; or zero to copy to another box.
170	One to review your private lists, or star to review public lists.
63	p.m.
14	Paused.
118	percent free.
321	Please dial your call back number then press star.
387	Please enter a box number or press two for a directory, press star to log in or hold on to leave a message. If you want to fax a document, press start on your fax machine now.
10	Please enter a box number, or press two for a directory. Press star to log in, or hold on to leave a message.
521	Please hold while I transfer you to ...

501	Please hold while I transfer your call
386	point
12	Press any key if you'd like to continue your session.
512	Press any key if you'd like to continue your session.
4	Press four to play the previous message, five to play this message again, six to play the next message, seven to save, nine to delete, zero to reply, or star for more options.
366	Press one for Default Message, two for Alternate Attendant ID, three for Mailbox, four for Mailbox Menu, five for Alternate Message, six for Retry, seven for Audiotex, and eight for Audiotex key.
519	Press one for further attendant options.
263	Press one for further voice mail options.
562	Press one for more options (played after Bridge Call Removed)
380	Press one to accept, or two to reject a call from CALLMaster.
509	Press one to accept, two to reject, or three to hear again.
3	Press one to accept, two to replay, or three to redo. (Note: four to append.)
392	Press one to change the LanguageID
350	Press one to change the type of transfer, two for the Extension to dial, three for three-way code, four for Alternate-Busy Action, five for Alternate-No Answer Action, six for Alternate-Screened Reject Action, seven for Alternate-Attendant ID, eight for Mailbox number, nine for Audiotex Key or star for Directory and Record Options.
56	Press one to change your password, two to record your greeting, three to record your name, four to maintain distribution lists, five to change your directory listing, six to turn notification on or off, or seven to change notification settings. Eight to record audiotex messages, or Nine to edit attendant records
257	Press one to change.
78	Press one to confirm deletion.
351	Press one to record the Alternate Message, two to record the directory listing or three to change the directory entry
8	Press one to review messages, two to send a message, or three to set personal options.

389	Press one to review messages, two to send a message, three to set personal options, or four to review faxes.
388	Press one to rewind, two to Pause and Resume, three to Fast Forward, or any other key to terminate playback.
327	Press one to start fax transmission now, two to send this fax to another number, four to play previous fax header, five to play the current fax, six to play the next fax, seven to save or nine to delete fax.
346	Press seven to review saved faxes, eight to review new faxes or nine to review deleted faxes
18	Press seven to review saved messages, eight to review new messages, or nine to review deleted messages.
520	Press star to leave a message, or enter another extension.
80	Press star to leave a message.
245	Press star to repeat.
106	Press star to send this message to another box.
100	Press star to stop recording.
232	Press star to use this listing, one to hear the number for this listing, four for the previous listing, five to hear this listing again, or six for the next listing.
328	Press start on your fax machine now.
171	private lists.
172	public lists.
339	Receive Failed
338	Received OK
244	Record module . . .
177	Record prompt . . .
97	Reply sent.
372	Retry
169	Review of . . .

344	Saved Fax
341	Saved Faxes
197	saved message.
198	saved messages.
16	Saved.
355	Screened
123	second . . .
116	seconds.
108	Sending this message to . . .
72	September . . .
364	Setting To ..
31	seven . . .
531	seven . . .
41	seventeen . . .
541	seventeen . . .
138	seventeenth . . .
128	seventh . . .
146	seventieth . . .
49	seventy . . .
549	seventy . . .
30	six . . .
530	six . . .



40	sixteen . . .
540	sixteen . . .
137	sixteenth . . .
127	sixth . . .
145	sixtieth . . .
48	sixty . . .
548	sixty . . .
514	Sorry, I can not get through. Please hold for further assistance.
506	Sorry, there is no such extension.
206	Spell your first name using the telephone keys, then press star; for "q" press seven, for "z" press nine.
205	Spell your last name using the telephone keys, then press the star key; for "q" press seven, for "z" press nine.
337	Status
300	superuser.
354	Supervised
34	ten
534	ten
131	tenth . . .
218	Thank you for calling, good-bye.
396	That language is not defined
182	That list exists already.
262	That option is not available.
518	That option is not available.

192	That was . . .
266	The alternate message is . . .
294	The current box status is ...
268	The directory listing is . . .
260	The earliest hour to call is . . .
394	The language ID is
261	The latest hour to call is . . .
86	The maximum message age is . . .
113	The maximum message age is . . .
85	The maximum message length is . . .
115	The maximum message length is . . .
112	The message capacity is . . .
61	The new box number is . . .
83	The new-message capacity is . . .
84	The saved-message capacity is . . .
333	The status of this fax is failed and it cannot be forwarded.
185	The system is full! No messages can be delivered!
242	There is no such module.
124	third . . .
37	thirteen . . .
537	thirteen . . .
134	thirteenth . . .

142	thirtieth . . .
45	thirty . . .
545	thirty . . .
331	This hardware port does not support Faxing, check with your system administrator.
191	This is . . .
168	This is the last box in the list.
153	This list is empty.
90	Thousand . . .
27	three . . .
527	three . . .
9	To which box?
353	Transfer
133	twelfth . . .
36	twelve . . .
536	twelve . . .
141	twentieth . . .
44	twenty . . .
544	twenty . . .
26	two . . .
526	two . . .
563	Two minutes remaining for this call (for Bridge Call)
231	Use the telephone keys to spell some or all of the last name of the person you are looking for, then press star; for "q" press seven, for "z" press nine.

298	user.
89	Welcome to voice mail. if you know which key to press, you may press it at any time if you make an error or need to back up, press the pound key. Now, record your personal greeting.
567	Would you like further assistance? Press 1 to continue.
105	You cannot send yourself a copy.
365	You do not have permission to change this attendant record. Speak to your administrator.
213	you entered
88	You entered . . .
79	You have . . .
77	You have reached your saved message limit, so you cannot save this message.
505	You may enter an extension, or hold on for assistance. (This should be where you offer choices to your callers)
178	You may not . . .
564	You may now start your call. (for Bridge Call)
395	You must record the English message for this module before any other language
15	Your box number or password was incorrect, please re-enter your box number.
561	Your connection could not be made (for Bridge Call)
234	Your directory listing is...
329	Your Fax could not be sent.
330	Your Fax has been scheduled.
193	your first . . .
57	Your greeting is . . .
194	your last . . .
58	Your name is . . .

54 Your new password is . . .

53 Your password is . . .

117 your system is . . .

24 zero . . .

524 zero . . .

# Dialogic Connection Type Events and Call Results

## Connection Type Event

- |   |  |
|---|--|
| 1 | Cadence Break (any connection)             |
| 2 | Loop Current Break (any connection)        |
| 3 | Positive Voice Detection (Human Response)  |
| 4 | Positive Answering or Voice Mail Detection |

After the Out Dialing application makes a call, the Connection Type Event is returned in the system variable \$contype.

When using the "Call" function in Audiotex/CALLBasic, the Connection Type Event is returned in the *event* (e) parameter.

## Call Results (CAR)

- |    |               |   |
|----|---------------|---|
| 0  | (no result)   |   |
| 7  | CA_BUSY       | Called line is busy.                        |
| 8  | CA_NOAN       | Called line did not answer.                 |
| 9  | CA_NORNG      | Called line did not ring.                   |
| 10 | CA_CONN       | Called line connected.                      |
| 11 | CA_OPINT      | Called line received an operator intercept. |
| 17 | CA_NODIALTONE | Called line received no dial tone.          |
| 18 | CA_FAXTONE    | Called line received a fax tone.            |
| 20 |               | Not In Service                              |
| 21 |               | Normal Clearing                             |

After the Out Dialing application makes a call, the CAR value is returned in the system variable \$cr.

When using the "Call" function in Audiotex/CALLBasic, the CAR value is returned in the *data* (d) parameter.

# T1 Setup Guide

CALLMaster interfaces with T1, PRI-ISDN lines supporting various protocols.

Using the Intel/Dialogic T1 board, CALLMaster requires a T1 software license to support T1 trunks. With Dialogic Global Call Drivers, CALLMaster supports the most popular protocols such as NI2, 5ESS, 4ES, DMS.

To learn more, see [CALLMaster T1/ISDN/PRI Support](#).

## Digital PBX Setup Guide

CALLMaster supports in-band integration to various digital PBXs using the Intel/Dialogic PBX Integration boards (D/42JCT-U and D/82JCT-U boards). These Dialogic boards and drivers simulate functionality of specific handsets to provide the integration to the PBX.

For a list of the PBXs supported and to learn more, see [CALLMaster Digital PBX Integration](#).



## MULTI-LINGUAL SUPPORT

CALLMaster can support applications in multiple languages as defined and set-up by the system administrator. Callers can then be given the option to select the language from the defined language set.

You will need to define a language set required by your customers, make the multi-lingual recordings. Your Audiotex application flow remains the same regardless of language selected. You just need to set the language id in a script based on the selection by your caller for CALLMaster to say the prompts in the desired language.

To learn more, see [CALLMaster Multi-lingual Support](#).

# Automatic Speech Recognition (ASR)

CALLMaster now supports ASR (Automatic Speech Recognition) capability via the Attendant and CALLBasic Scripting features. If you have purchased the CALLMaster ASR license, the CALLMaster license file will be activated for ASR.

CALLMaster can utilize any third party SR - Speech Recognition engine that is Microsoft SAPI 5.x compliant to provide ASR Automatic Speech Recognition capabilities to callers. CALLMaster incorporates ASR in two ways.

1. First it allows caller to say the name of the person or extension they want instead of dialing and extension or spelling the name with telephone buttons. Use flag "SAR" in the [System Parameter](#) 'System Settings' field for strict automatic/speech recognition.
2. And second, CALLMaster features a new Recognize command in CALLBasic, the integrated scripting language, which will determine the phrase or word spoken from a designated list of words/phrases. This brings ASR functionality to IVR /Audiotex. Now any CALLMaster application like Order Entry can utilize Speech Recognition to provide a more natural user interface with greater flexibility.

## Barge In Support

CALLMaster ASR supports Barge-In in conjunction with the JCT type Dialogic board. Barge-In allows the caller to say their desired selection while the greeting is being played. Barge-In requires that the greeting be recorded in 8Khz 8Bit Mono Linear PCM wave file format.

See [Voice Mail – Administrator Options - Record Barge In Prompts](#).

For more information, see [CALLMaster Speech Recognition](#).

## Text-to-Speech (TTS)

CALLMaster now supports TTS (Text-to-Speech) via the CALLBasic Scripting features. TTS functionality is included in the CALLMaster software and does not require any additional licensing.

CALLMaster can utilize any third party TTS - Text-to-Speech engine that is Microsoft SAPI 5.x compliant to provide TTS capabilities to callers. Purchase SAPI compliant voices for greater selection of voice types.

CALLMaster provides text-to-speech as part of its powerful CALLBasic Scripting commands. CALLBasic features a Speak command which can translate the specified text into words.

CALLBasic script can be customized to use selected voice, giving you flexibility of gender and accent. Additional parameters such as the ability to spell vs. speak the text, delivery speed, etc. can also be customized using the CALLBasic Speak command. The text to be spoken can be a literal, a field in a database table or a variable based on processing. See [CALLBasic Commands](#) for details.

For details and download, see [CALLMaster Text-to-Speech](#)

## Anti-Virus Software Conflict

Most anti-virus software lock the files while they are checking for viruses. Conflict could arise when both the anti-virus software and CALLMaster require exclusive control of the CALLMaster database or any of the files in the CALLMaster folder. This will probably cause the CALLMaster service to crash.

If you need to run anti-virus software on the CALLMaster PC, you have two alternatives:

1. Exclude the CALLMaster folder from the set of files scanned by the anti-virus software.
2. Schedule the anti-virus software to run while CALLMaster service has been stopped. The anti-virus can be scheduled to run off hours, stopping the CALLMaster service before running the anti-virus and restarting the CALLMaster service on completion of the anti-virus check. You can use the CMStop.cmd and CMStrat.cmd in the CALLMaster/Bin folder to set-up this process.

If you have a problem or need further assistance contact SpeechSoft Tech Support at

- [support@speechsoft.com](mailto:support@speechsoft.com)
- (914) 273-5560 option '3'.

# Set-up CALLMaster Attendant Directory

Use the following steps to set-up the CALLMaster Attendant Directory.

1. Backup the CALLMaster folder
2. Record Alternate Message and Attendant Name; Enter Directory Listing Entry:
  - A. Log in as the Administrator.  
Call the system and log into an Administrator mailbox;
  - B. Access the Administrator Menu Options.  
At the main mailbox menu, 'Press 1 to retrieve messages, 2 to send a message or 3 to set personal options', Press '\*' for Administrator Options. You will not be prompted for this option;
  - C. Select Option 9 to record Attendant Directory;
  - D. Enter the Attendant Id for which you want to make the directory entry followed by the \* key when prompted;
  - E. Enter the time of day (0 for all day, one for morning, two for afternoon, or three for evening);
  - F. Record Alternate Message.  
CALLMaster will prompt you to record an alternate message for this Attendant record. Press \* when finished recording. Select 1 to approve, 2 to replay or 3 to hear again. The recording ID will be placed in the Attendant record 'Alt Message ID' field. This recording will be used if the Type of Transfer set for this attendant record is Alternate Message;
  - G. Record Attendant Name.  
CALLMaster will prompt you to record the Name for this Attendant record. Press \* when finished recording. Select 1 to approve, 2 to replay or 3 to hear again. The recording ID will be placed in the Attendant record 'Name File ID' field. This recording will be used when transferring caller. If the Name is not recorded, the extension number will be played to the caller instead of the name of the person or department the caller is trying to reach;
  - H. Enter Directory Listing Entry.  
CALLMaster will prompt you to use the touch tone keys to spell the last name for this Attendant record. Press \* when finished. Select 1 to approve, 2 to replay or 3 to hear again. The recording ID will be placed in the Attendant record 'Directory Listing' field. This entry will be used to lookup and find match against caller selection;
  - I. Repeat steps D to H for each Attendant record with the associated time of day you wish to have included in your Attendant Directory.
3. Create Attendant Directory Record:
  - A. Using the CALLMaster Manager, select Manage | Attendant;
  - B. Add an Attendant record setting the 'Attendant ID' to the number you wish the caller to enter for directory listing option. Set 'Time of Day' to All Day; set the 'Type of Transfer' to Directory.
  - C. Save records and close the Manager.
4. Record Attendant Menu Prompt# 505:
  - A. Call the system and log in to the Administrator Main Menu per Steps 2A and 2B;
  - B. From the Administrator's Menu, select option 7 to Record System Prompts;
  - C. Select Prompt 505 and re-record to provide the caller with options.  
For example, if in Step 3B you created a Directory record whose Attendant ID = 4 , then your prompt 505 could be as follows "If you know your party's extension please enter it now or press 1 for information, 2 for sales, 3 for technical support or 4 for a company directory".

You have now set-up the Attendant Directory and are ready to use it.

# VoIP Configuration

## VOIP – Voice Over IP

Voice over IP (VoIP) can be described as the ability to make telephone calls and send faxes over IP based data networks with a suitable Quality of Service (QoS). The voice information is sent in digital form using discrete packets rather than via dedicated connections as in the circuit-switched Public Switched Telephone Network (PSTN). There are two major international groups defining standards for VoIP:

- International Telecommunications Union, Telecommunications Standardization Sector (ITU-T), which has defined the following:
  - Recommendation H.323, covering Packet-based Multimedia Communications Systems (including VoIP)
- Internet Engineering Task Force (IETF), which has defined drafts of the several RFC (Request for Comment) documents, including the following central document:
  - RFC 3261, the Session Initiation Protocol (SIP)

The H.323 recommendation was developed in the mid 1990s and is a mature protocol.

SIP (Session Initiation Protocol) is an emerging protocol for setting up telephony, conferencing, multimedia, and other types of communication sessions on the Internet.

The Intel Dialogic HMP software provides supports both H.323 and SIP as well as FOIP – Fax Over IP.

In addition to answering inbound calls and making outbound calls these calls CALLMaster can transfer the calls to any other number either in a supervised or blind transfer fashion.

CALLMaster can register with your VOIP Gateway/GateKeeper/Registrar to be fully integrated into your VOIP environment. It also works with VOIP service providers in a remote or locally hosted setup. Providing the most flexibility in any environment.

## IP Network Configuration and Settings

CALLMaster supports both the H.323 and SIP VOIP protocols and various audio coders and DTMF formats. By default CALLMaster will run in stand alone mode and listen on port 5060 for incoming VOIP calls. Any VOIP device can call CALLMaster by dialing the IP address of the machine running the CALLMaster software. CALLMaster can also register with a Gateway or Registrar.

For CALLMaster to work in a VOIP environment the configuration file CMVOIP.cfg must be setup to match the VOIP environment. The CMVOIP.cfg file located in the CALLMaster\Data folder. This file allows the following setting to be made:

IP Protocol : The IP Protocol used for opening the IP ports, values: H323, SIP, or both

DTMFmode: Touch Tone signaling mode, possible options: OutOfBand, inband, or rfc2833

Audio format: Formats supported g711Alaw, g711Mulaw, gsm, gsmEFR, g723\_5\_3k, g723\_6\_3k, g729a, g729ab, t38UDPFax

If CALLMaster is to register with a Gateway, GateKeeper or Registrar then set the following values. If it is not registering then set the GWIP setting to 0.0.0.0.

Gateway or Registrar parameters:

GWAcctName - <username> GateWay account username for login to VOIP Gateway server

GWAcctPW - <password> GateWay account password for login to VOIP Gateway server

GWAcctRealm - <realm> of VOIP Gateway (optional)

GWIP - <IP address> Dot notation of VOIP Gateway server with optional port number eg

206.80.70.54:5060

LocalAlias - <IP address> Dot notation of CALLMaster

To learn more, see [CALLMaster VoIP Configuration](#).

## CALLMaster Database Connection and Provider

CALLMaster is shipped with an MS SQL 2008 database. The database file, CALLMasterSQL.mdf, is located in the CALLMaster\Data folder.

During installation, CALLMaster will install SQL Server 2008 Express on the local computer and will connect the database (CALLmasterSQL.mdf) to this server.

This connection string and provider name is stored in the Windows registry.

The database connection string at installation is stored with variable DBName as follows:

```
"Data Source=localhost\SQLEXPRESS;Initial Catalog=CALLMasterSQL;Integrated Security=SSPI;Connect Timeout=5"
```

The database provider at installation is stored with variable DBProvider as follows:

```
"Provider=SQLNCLI10;MARS Connection=True;"
```

## Changing the CALLMaster Database Connection and Provider

If you wish to use the installation setup for the CALLMaster database, then you do not have to make any changes.

If you are running another SQL Server and wish to use this server instead of the SQL Server 2008 Express, you may need to change the connection string and/or provider. Follow the steps outlined below to change:

1. Stop the CALLMaster service from the control panel;
2. Detach the CALLMaster.mdf file from the SQL Server 2008 Express, then attach the CALLMasterSQL.mdf and the CALLMasterSQL.ldf files from the CALLMaster\Data folder to your SQL Server;
3. Make sure that you have the correct provider and connection string for your SQL server.  
TIP: to get the connection string details, try the steps below:
  - i) Log in with the same user id as the one used to install CALLMaster;
  - ii) On the desktop right click and choose "New" - "Text Document"
  - iii) Then type "test.udl" and hit enter to rename the document. You will get a "Rename" pop up message that says "If you change a file name extension, the file may become unusable. Are you sure you want to change it?". Click "Yes" to rename the document to test.udl.
  - iv) Then double click the test.udl icon to run a "Data Link Properties" wizard. Follow the wizard to connect to your database as needed:
    - a) From the provider tab, select 'Microsoft OLE DB Provider for SQL Server'; click 'Next';
    - b) Select or enter the server name. You may need to enter "<servername>\localhost";
    - c) Select the mode to log on to the server;
    - d) Dropdown and select the CALLMasterSQL database. If this entry does not show then you will need to retry connecting to a different server name;
    - e) Click "Test Connection"
  - v) Once you have the "Test Connection", open the test.udl with notepad to see the connection string.
4. Update Windows Registry Setting:-
  - i) Option 1 - Using files provided in the CALLMaster\Bin\ folder:  
To change Provider:
    - From the CALLMaster\Bin folder, open the CMDBProvider.reg file in Notepad and edit it for your provider setting  
(use the provider from the test.udl file as described above). Close and save;
    - Double click to run the CMDBProvider.reg file to set-up the CALLMaster registry to use the Provider;  
To change Connection string:
    - From the CALLMaster\Bin folder, open the CMDBName.reg file in Notepad and edit it for your connection string (e.g. SQL Server name, security information)  
(use the connection string from the test.udl file as described above). Close and save;
    - Double click to run the CMDBName.reg file to set-up the CALLMaster registry to use the



connection string;

ii) Option 2 - Using the Command prompt:

- Open a Command window then type regedit and press enter;
- Expand tab HKEY\_LOCAL\_MACHINE, SOFTWARE, SpeechSoft, CALLMaster. Double click either the DBName or the DBProvider parameter and enter desired string in the pop-up window (use the information from the test.udl file as described above).
- Close to save new info.

5. Start the CALLMaster Service.

### **Separate Database for the OutcallSchedule tables**

If you are using the Out Dial function, the OutcallSchedule and OutcallScheduleCompleted tables can be in a database separate from the CALLMaster database. CALLMaster allows you to set-up a connection string to point to any database desired. See [Out Dial](#) for details on how to set-up these connection strings.

You are now ready to [Start CALLMaster](#).

## Hunt Group Restrictions

Hunt Group Restrictions provides a way to control how many simultaneous calls can be active for a particular Hunt Group, which is composed of one or more DNIS – Dialed Number. Once that limit is reached, CALLMaster will reject further calls with a SIP – “BUSY HERE” message or a disconnect until the number of calls falls below the limit set for the hunt group.

This feature is managed via 2 new tables and enabled via a flag in the System Parameter.

### HuntGroup and HuntGroupDNIS tables

- 1) HuntGroup table keeps a list of Hunt groups and the maximum ports that can be active for that group. Fields include:

GroupID	- Group Identifier
Description	- Optional
Max Active Calls	- Number of simultaneous active calls for the group

- 2) HuntGroupDNIS table identifies the Hunt Group for each DNIS. Fields include:

DNIS#	- Number dialed
DNISOwner	- Optional
HuntGroupID	- Group Identifier from the HuntGroup table

### Enabling Hunt Group Restrictions

The Hunt group restriction is enabled by adding one of the flags below to the System Parameter 'System Settings' (field #26):

- HGR - Hunt Group Restriction activated - Unregisterd DNIS accepted;
- HGR0 - Hunt Group Restriction activated - Unregisterd DNIS rejected;