

TECHNICAL MEMORANDUM

Introduction

The current Call Ahead system supports the Motorola Call Ahead protocol that was introduced in 1987. It uses the RS-232 serial interface with different parameters to communicate with the host computer. The system receives a transaction from the host computer to dial out a phone number and deliver a prerecord message to the called party. After the call session, the Call Ahead system sends a transaction to the host computer and provides the status of the call and the response. The status may be Busy, No Answer, Answering Machine, or DTMF response.

The current system has served the industry well for the past twenty years. However the current Call Ahead system can not be easily connected to newer host computers and can not be shared by multiple host computers. The software for the current Call Ahead system was written twenty years ago and does not support the digital telephony and Voice over Internet Protocol (VoIP) connections and has no provision for a live back up system.

Since the installed base of current Call Ahead system is very large, we can not abandon the current protocol, instead the new system has been designed to support the current legacy RS-232 serial and the TCP/IP interfaces. However, all the new features and capabilities are offered through the TCP/IP interface.

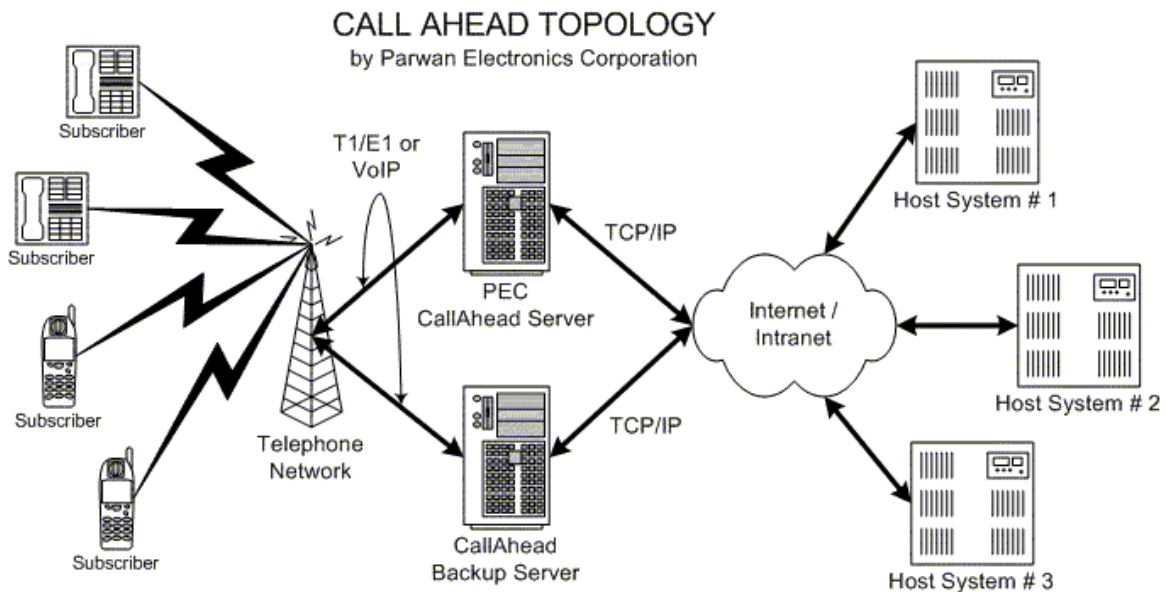
This document sets the framework for the new Call Ahead Server design. The host computer applications software has the ability control the operation of the Call Ahead Server functionality. The initial design of the software has the legacy Call Ahead features and the other features:

- Ethernet TCP/IP interface
- Serve multiple host computers
- Support Analog, digital T1, E1, VoIP telephone links
- Integration with many PBX switches
- Call transfers and digital patching
- Two B Channel Transfer (TBCT)
- Touch tone generation and detection
- Dynamic ANI delivery
- Support incoming calls
- Bulk calling and message delivery

The future release of Call Ahead Server will include:

- Speech recognition
- Text to speech
- Two way recording of the voice conversation
- Internet Client phone support
- Call conferencing

The diagram below depicts the Call Ahead Server topology that will be implemented:



Call Ahead Server XML Protocol

The Call Ahead Server Extensible Markup Language (XML) protocol allows an external applications program to communicate with Call Ahead Server through TCP/IP using the XML format. The external program may instruct Call Ahead Server to perform Administrative, call control and switching, voice processing, and other functions.

For example, the external program could connect to Call Ahead Server and instruct it to dial a phone number and play a message file to the called party.

TCP/IP Port

Call Ahead Server listens to TCP/IP messages on port 700. You may set the port to another address by setting the **eee51** parameter. The applications program must first establish a connection with Call Ahead Server and then communicate with it afterwards.

Call Ahead Server Parameters

The following Call Ahead Server parameters must be set:

Parameter	Value	Description
xxx77	1	Enable XML Messaging
xxx77	8	Enable XML Receive Call
eee51	0	Socket port address offset from 700

XML Message

Call Ahead Server supports the following four classes of XML messages:

- Administrative Messages
- Outbound Call Control and Switching Messages
- Inbound Call Control and Switching Messages
- Voice Processing Messages

Administrative Messages

The Call Ahead Server Administrative XML messages are used to initialize the link and terminate the link.

Link Initialization and Termination Messages

Initialize Link - <INIT_LINK><INFO>

The applications program sends this message to Call Ahead Server and instructs it to start accepting calls for the application.

Format:

```
<INIT_LINK>
  <INFO MSGID="Message ID" NAME="Customer Name" DNIS="Phone No"
    PASS_CODE="Pass word">
  </INFO>
</INIT_LINK >
```

Attributes:

Attribute	Min	Max	Opt	Description
MSGID	4	20	N	Message ID
NAME	3	30	Y	Name to identify the application or customer
DNIS	3	14	Y	Maximum 14 digits – report the received call with this DNIS to the application. Make sure the xxx77&8 is true.
PASS_CODE	4	4	Y	Pass code for setting the DNIS mailbox

Example 1:

```
<INIT_LINK>
  <INFO MSGID="PAN-123" NAME="PEC" DNIS="7322901900"
    PASS_CODE="2424">
  </INFO>
</INIT_LINK>
```

In the above example, Call Ahead Server receives the message from the applications program and sets the mailbox 7322901900 to receive calls for the application. The pass code 2424 is the pass code for the mailbox. The mailbox name is set to PEC. When a caller dials 7322901900, Call Ahead Server sends the <RECEIVE_CALL> message to the applications program.

In order for Call Ahead Server to process inbound calls, the xxx77 must be set to 9. If the application program needs to handle more than one DNIS number, it must send one <INIT_LINK> message per DNIS. The applications program must have the correct pass codes for the mailboxes.

Example 2:

```
<INIT_LINK>  
  <INFO MSGID="PAN-512">  
  </INFO>  
</INIT_LINK>
```

In the above example, Call Ahead Server receives the message from the applications program and only establishes the link for making outbound calls.

Response - <INIT_LINK><RESPONSE>

Call Ahead Server sends this message to the applications program in response to the <INIT_LINK><INFO> message.

Format:

```
<INIT_LINK>  
  <RESPONSE MSGID="Message ID" STATUS="SUCCESS, BUSY, FAILED,  
  IBOUND CALLS NO SUPPORTED, INVALID PASS CODE, INVALID DNIS">  
  <\RESPONSE >  
</INIT_LINK>
```

Terminate Link - <TERMINATE_LINK><INFO>

The applications program sends this message to Call Ahead Server and instructs it to disconnect all the calls and stop accepting news calls for the application.

Format:

```
<TERMINATE_LINK>  
  <INFO MSGID="Message ID">  
  </INFO>  
</TERMINATE_LINK>
```

Terminate Response - <TERMINATE_LINK><RESPONSE>

Call Ahead Server sends this message to the applications program in response to the <TERMINATE_LINK><INFO> message.

Format:

```
<TERMINATE_LINK>  
  <RESPONSE MSGID="Message ID" STATUS="SUCCESS, FAILED">  
  </RESPONSE>  
</TERMINATE_LINK>
```

Pause Link Message - <PAUSE_LINK><INFO>

The applications program send this message to Call Ahead Server to finish processing the current calls in progress, but do not allow new calls for the application.

Format:

```
<PAUSE_LINK>  
  <INFO MSGID="Message ID">  
  </INFO>  
</PAUSE_LINK>
```

Pause Response - <PAUSE_LINK><RESPONSE>

Call Ahead Server sends this message to the applications program in response to the <PAUSE_LINK><INFO> message.

Format:

```
<PAUSE_LINK>  
  <RESPONSE MSGID="Message ID" STATUS="SUCCESS, FAILED">  
  <\RESPONSE>  
</PAUSE_LINK>
```

Exit Message - <EXIT_LINK><INFO>

Call Ahead Server sends this message to the Application program after the operator enters the exit command from the Call Ahead Server menu.

Format:

```
<EXIT_LINK>  
  <INFO MSGID="Message ID" >  
  </INFO>  
</EXIT_LINK>
```

Exit Response - <EXIT_LINK><RESPONSE>

The applications program must send this message to Call Ahead Server in response to the <EXIT_LINK><INFO> message.

Format:

```
<EXIT_LINK>  
  <RESPONSE MSGID="Message ID" STATUS="SUCCESS, FAILED">  
  <\RESPONSE>  
</EXIT_LINK>
```


Outbound Call Control and Switching Messages

Dial-out Messages

The applications program sends this data to Call Ahead Server to initiate an outbound call activity. The first response message Call Ahead Server sends is the immediate message and the second response message is the result of the call progress after the call has been made.

Required Attributes for All Elements and Responses within <DIALOUT>

Attribute	Description
MSGID	Minimum length 4, maximum length 50, THIS IS AN ARBITRARY TRANSACTION ID USED FOR IDENTIFICATION

Outbound Dial-out - <DIALOUT><DIAL>

The applications program sends this message to Call Ahead Server to initiate an outbound call and performs other activities after the called party answers the call.

Format:

```
<DIALOUT>
  <DIAL MSGID="Message ID" CAMPID="Campaign ID" DNIS="Number to
  Dial" ANI="Caller ID to Display" LIVEMESSAGE="Voice File Name"
  ANSMESSAGE="Voice File Name" MODE="BLIND, SUPERVISED">
  </DIAL>
</DIALOUT>
```

Attributes:

Attribute	Min	Max	Opt	Description
CAMPID	E	20	Y	Optional - minimum length 4, Maximum length 20 – Campaign ID
DNIS	3	14	N	Maximum 14 digits, number to dial.
ANI	3	14	Y	Optional - Calling Party ID, Maximum 10 digits.
LIVEMESSAGE	4	60	Y	Optional - voice message to play upon detecting live hello.
ANSMESSAGE	4	60	Y	Optional – voice message to play upon detecting answering machine.
MODE	5	10	N	BLIND or SUPERVISED

Example:

```
<DIALOUT>
  <DIAL MSGID="781234A5" CAMPID="OSHKOSH" DNIS="9204567812"
  ANI="8009871212", ANSMESSAGE="HELLO.VOX", MODE="SUPERVISED">
  </DIAL>
</DIALOUT>
```

Example:

```
<DIALOUT>
  <DIAL MSGID="781234A6" CAMPID="FLUSHING" DNIS="7184568812"
  ANI="8005551212" LIVEMESSAGE="MEETING.VOX",
  ANSMESSAGE="CALL.VOX", MODE="SUPERVISED">
  </DIAL>
</DIALOUT>
```

Immediate Response - <DIALOUT> <RESPONSE>

Call Ahead Server checks the data received and sends the message immediately to the applications program.

Attributes:

Attribute	Description
STATUS	SUCCESS, NUMBER_TOO LONG, FAILED, BUSY

Example:

```
<DIALOUT>
  <RESPONSE MSGID="781234A5" STATUS="SUCCESS">
  </RESPONSE>
</DIALOUT>
```

Call Progress Status - <DIALOUT><TERMINATION>

After dialing the number, Call Ahead Server returns the status of the call progress to the applications program.

Format:

```
<DIALOUT>
  <TERMINATION MSGID="Message ID" DNIS="Phone Number Dialed"
  MESSAGE="Voice File Played" TRUNK_PORT="Sever Telephone Port
  Number" EVENT="Call Progress Event" DTMF="Digit Pressed">
  </TERMINATION>
</DIALOUT>
```

Attributes:

Attribute	Description
DNIS	The phone number that was dialed
MESSAGE	File that was played
EVENT	CONNECT, BUSY, NOANS, INTERCEPT, ANSMACHINE, FAX, DTMF, or NODIALTONE
TRUNK_PORT	Port number Call Ahead Server used to dial out the number.
DTMF	The DTMF digit that was entered

Example 1:

```
<DIALOUT>
  <TERMINATION MSGID="672312" DNIS="7185231234"
  MESSAGE="METER_READ.VOX" EVENT="CONNECT" TRUNK_PORT="14"
  DTMF="3">
  </TERMINATION>
</DIALOUT>
```

In the example above the phone number 718-523-1234 was dialed and after connect, it played the file METER_READ.VOX. The called party listened to the message and pressed the DTMF key.

In general the message file script must have instructions for the called party to press a DTMF key for approving the message.

Example 2:

```
<DIALOUT>
  <TERMINATION MSGID="672314" DNIS="9205231236" MESSAGE=""
  EVENT="BUSY" TRUNK_PORT="14" DTMF="">
  </TERMINATION>
</DIALOUT>
```

In the example above the phone number 920-523-1236 was dialed and it detected a BUSY.

Disconnect Messages

Call Ahead Server handles the network generated and applications generate disconnects, in each case the signaling must be handled appropriately.

Application Disconnect - <APP_TRUNK_DISCONNECT><INFO>

The applications program sends this message to Call Ahead Server to disconnect and a port. After Call Ahead Server receives this message, it sends the disconnect signal to the telephone network.

Format:

```
<APP_TRUNK_DISCONNECT>
  <INFO MSGID="Message Identification ID" TRUNK_PORT="Telephony
  Line Number">
  </INFO>
</APP_TRUNK_DISCONNECT>
```

Attributes:

Attribute	Description
TRUNK_PORT	Port number Call Ahead Server used to dial out the number.

Example:

```
<APP_TRUNK_DISCONNECT>
  <INFO MSGID="786512M3" TRUNK_PORT="14">
  </INFO>
</APP_TRUNK_DISCONNECT>
```

Response - <APP_TRUNK_DISCONNECT><RESPONSE>

Call Ahead Server sends this message to the applications program after sending the disconnect signal to the network. This message is send in response to the <APP_TRUNK_DISCONNECT><INFO> message.

Format:

```
<APP_TRUNK_DISCONNECT>
  <RESPONSE MSGID="ID String" TRUNK_PORT="Line" STATUS="code">
  </RESPONSE>
</APP_TRUNK_DISCONNECT>
```

Attributes:

Attribute	Description
STATUS	SUCCESS, INVALID_TRUNKPORT

Example:

```
<APP_TRUNK_DISCONNECT>  
  <RESPONSE MSGID="786512M3" TRUNK_PORT="Port Number"  
    STATUS="SUCCESS">  
  </RESPONSE>  
</APP_TRUNK_DISCONNECT>
```

Network Initiated Disconnect - <NETWORK_TRUNK_DISCONNECT><INFO>

Call Ahead Server informs the application program when the Network sends a disconnect signal.

Format:

```
<NETWORK_TRUNK_DISCONNECT>
  <INFO MSGID="Message ID" TRUNK_PORT="Server Telephony Port #">
  </INFO>
</NETWORK_TRUNK_DISCONNECT>
```

Attributes:

Attribute	Description
TRUNK_PORT	The port number from where the disconnect signal was received.

Example:

```
<NETWORK_TRUNK_DISCONNECT>
  <INFO MSGID="765121N3" TRUNK_PORT="21">
  </INFO>
</NETWORK_TRUNK_DISCONNECT>
```

Application Disconnect Response - <NETWORK_TRUNK_DISCONNECT><RESPONSE>

Example:

```
<NETWORK_TRUNK_DISCONNECT>
  <RESPONSE MSGID="Message ID" TRUNK_PORT="Server Telephony Port
  Number" STATUS="code">
  </RESPONSE>
</NETWORK_TRUNK_DISCONNECT>
```

Attributes:

Attribute	Description
STATUS	SUCCESS, INVALID_TRUNKPORT

Example:

```
<NETWORK_TRUNK_DISCONNECT>
  <RESPONSE MSGID="765121N3" TRUNK_PORT="Port Number"
  STATUS="SUCCESS">
  </RESPONSE>
</NETWORK_TRUNK_DISCONNECT>
```

Initiate Digital Transfer and Patching

Dial and Patch - <DIAL_PATCH><INFO>

The applications program sends this message to Call Ahead Server and instructs it to grab a free port and dial a phone number. After the called party answers, then patch the call to the trunk port specified in the message.

Format:

```
<DIAL_PATCH>  
  <INFO MSGID="Message ID" TRUNK_PORT1="Physical Port No to Patch  
  to" DNIS="Phone Number to Call" ANI="Number to Show">  
  </INFO>  
</DIAL_PATCH>
```

Attributes:

Attribute	Min	Max	Opt	Description
MSGID	4	20	N	Message ID
TRUNK_PORT1	2	4	N	Trunk port to patched the called party to
DNIS	3	14	N	Maximum 14 digits, number to dial.
ANI	3	14	Y	Number to display on the called party's telephone set

Example:

```
<DIAL_PATCH>  
  <INFO MSGID="786123" TRUNK_PORT1="05" DNIS="17322901900"  
  ANI="2126793400">  
  </INFO>  
</DIAL_PATCH>
```

In the above example, Call Ahead Server receives the message from the applications program and finds an available trunk port to dial out on. It sets the ANI to 2126793400 and dials the called party's number 17322901900. After the called party answers the call on the available trunk port, Call Ahead Server patches the trunk port 5 to the called party's trunk port.

Dial and Patch Response - <DIAL_PATCH><RESPONSE>

Call Ahead Server sends this message to the applications program if there are errors in the value of attributes.

Format:

```
<DIAL_PATCH>
  <RESPONSE MSGID="Message Id" STATUS="INVALID_TRUNK1,
    UNAVAILABLE_OUTBOUND_LINE">
  </RESPONSE>
</DIAL_PATCH>
```

Dial and Patch Done - <DIAL_PATCH><TERMINATION>

Call Ahead Server sends this message to the applications program after it dials the number and receives the call progress information from the hardware.

Format:

```
<DIAL_PATCH>
  <TERMINATION MSGID="Message Id" EVENT="CONNECTED, BUSY, NOANS,
    INTERCEPT" TRINK_PORT1="Port Number Patched to" TRINK_PORT2="Port
    Number Dialed On">
  </TERMINATION>
</DIAL_PATCH>
```


Patch Ports - <PATCH_PORTS><INFO>

The applications program sends this message to Call Ahead Server to patch two digital ports unconditionally.

Format:

```
<PATCH_PORTS>
  <INFO MSGID="Message ID" TRUNK_PORT1="Physical Port No"
    TRUNK_PORT2="Physical Port No">
  </INFO>
</PATCH_PORTS>
```

Attributes:

Attribute	Min	Max	Opt	Description
MSGID	4	20	N	Message ID
TRUNK_PORT1	2	4	N	First trunk digital port
TRUNK_PORT2	2	4	N	Second trunk digital port

Example:

```
<PATCH_PORTS>
  <INFO MSGID="783412" TRUNK_PORT1="04" TRUNK_PORT2="122">
  </INFO>
</PATCH_PORTS>
```

Response Patch Ports - <PATCH_PORTS><RESPONSE>

Call Ahead Server immediately sends this message to the applications program if it can not patch the ports specified in the <PORT_PATCH><INFO> message.

Format:

```
<PATCH_PORTS>
  <RESPONSE MSGID="Message Id" TRUNK_PORT1="Physical Port No"
    TRUNK_PORT2="Physical Port No" STATUS="INVALID_TRUNK_PORT1,
    INVALID_TRUNK_PORT2, TRUNK_PORT1_ALREADY_PATCHED,
    TRUNK_PORT2_ALREADY_PATCHED">
  </RESPONSE>
</PATCH_PORTS>
```

Example:

```
<PATCH_PORTS>
  <RESPONSE MSGID="783412" TRUNK_PORT1="04" TRUNK_PORT2="12672"
    STATUS="INVALID_TRUNK_PORT2">
  </RESPONSE>
</PATCH_PORTS>
```

In the above example, the value of attribute TRUNK_PORT2 is not valid.

Patching of Ports Done - <PATCH_PORTS><TERMINATION>

Call Ahead Server sends this message to the applications program after it has successfully patched the ports specified in the <PATCH_PORTS><INFO> message. Call Ahead Server performs a full duplex patch, i.e., the receive side of one port is patched to the transmit side of the other port. If the patching failed, Call Ahead Server sends the FAILED message.

Format:

```
<PATCH_PORTS>
  <TERMINATION MSGID="Message Id" TRUNK_PORT1="Physical Port No"
    TRUNK_PORT2="Physical Port No" EVENT="SUCCESS, FAILED">
  </TERMINATION>
</PATCH_PORTS>
```

Tearing Down a Patch - <TEAR_PATCH><INFO>

The applications program sends this message to Call Ahead Server and instructs it to tear down a patch that was previously established.

Format:

```
<TEAR_PATCH>
  <INFO MSGID="Message ID" TRUNK_PORT1="Leg1 Physical Port No"
    TRUNK_PORT2="Leg2 Physical Port No"
  </INFO>
</TEAR_PATCH>
```

Attributes:

Attribute	Min	Max	Opt	Description
MSGID	4	20	N	Message ID
TRUNK_PORT1	2	4	N	First trunk digital port
TRUNK_PORT2	2	4	N	Second trunk digital port

Example:

```
<TEAR_PATCH>
  <INFO MSGID="786512" TRUNK_PORT1="06" TRUNK_PORT2="23"
  </INFO>
</TEAR_PATCH>
```

In the above example, Call Ahead Server tears down the connection between trunk port 6 and trunk port 23. The trunk ports are not connected to each other, Call Ahead Server sends a error message to the applications program.

Tear Down Patch Response - <TEAR_PATCH><RESPONSE>

Call Ahead Server sends this message to the applications program after it has torn down the patch connection between two ports.

Format:

```
<TEAR_PATCH>
  <RESPONSE MSGID="Message Id" TRUNK_PORT1="Leg1 Physical Port No"
    TRUNK_PORT2="Leg2 Physical Port No" STATUS="INVALID_TRUNK_PORT1,
    INVALID_TRUNK_PORT2, TRUNK_PORTS_NOT_PATCHED">
  </RESPONSE>
</TEAR_PATCH>
```

Tear Down Patch Done - <TEAR_PATCH><TERMINATION>

Call Ahead Server sends this message to the applications program after it has torn down the patch connection between two ports.

Format:

```
<TEAR_PATCH>
  <TERMINATION MSGID="Message Id" TRUNK_PORT1="Leg1 Physical Port
  No" TRUNK_PORT2="Leg2 Physical Port No" EVENT="SUCCESS">
  </TERMINATION>
</TEAR_PATCH>
```

Example:

```
<TEAR_PATCH>
  < TERMINATION MSGID="786512" EVENT="SUCCESS" TRUNK_PORT1="06"
  TRUNK_PORT2="23"
  </ TERMINATION >
</TEAR_PATCH>
```

In the above example, Call Ahead Server successfully tore down the connection between trunk port 6 and trunk port 23.

Inbound Call Control and Switching Messages

Inbound Call

Receive Call - <RECEIVE_CALL><INFO>

When Call Ahead Server receives a call from the Network it informs the application program and passes the DNIS number received from the Network. The mailbox corresponding to the DNIS number be set up with the <INIT_LINK> message. Generally this is an X type mailbox with the name containing the socket id of the applications program link.

Format:

```
<RECEIVE_CALL>
  <INFO MSGID="Message ID" DNIS="Phone Number" ANI="Phone Number"
    REDIRECTING_NO="Phone Number" TRUNK_PORT="Port Number"
  </INFO>
</RECEIVE_CALL>
```

Response Receive Call - <RECEIVE_CALL><RESPONSE>

The applications program must respond back to Call Ahead Server and instruct it in what actions to take.

Format:

```
<RECEIVE_CALL>
  <RESPONSE MSGID="Message Id" STATUS="DISCONNECT, PLAY, TRANSFER,
    RECORD" AGENT_NO="Agent" PLAY_FILE="file" RECORD_FILE="file"
  </RESPONSE>
</RECEIVE_CALL>
```

Voice Processing Messages

Playing and Stopping of Voice Message Files

Play Voice File - <PLAY_VOICE_FILE><INFO>

The applications program sends this message to Call Ahead Server to initiate the playing of a voice message file.

Format:

```
<PLAY_VOICE_FILE>
  <INFO MSGID="Message ID" TRUNK_PORT="Port Number" FILE1="file
    name" FILE2="file name" MODE="INTERRUPTABLE, UNINTERRUPTABLE">
  </INFO>
</PLAY_VOICE_FILE>
```

Attributes:

Attribute	Min	Max	Opt	Description
MSGID	4	20	N	Message ID
TRUNK_PORT	2	4	N	Trunk port to play the message file.
FILE1	3	40	N	The first message file path.
FILE2	3	40	Y	Optional second message file path
MODE	13	15	N	Allow play to be interrupted by DTMF or not

Example:

```
<PLAY_VOICE_FILE>
  <INFO MSGID="786123" TRUNK_PORT="15" FILE1="C:\PROMPTS\HELLO.VOX"
    MODE="INTERRUPTABLE">
  </INFO>
</PLAY_VOICE_FILE>
```

Call Ahead Server will play the voice message file c:\prompts\hello.vox. If the listener pressed a DTMF key, then Call Ahead Server will stop playing.

Response Play File - <PLAY_VOICE_FILE><RESPONSE>

Call Ahead Server sends this message immediately to the applications program after it has determined that it will be able to play the file or it is unable to proceed.

Format:

```
<PLAY_VOICE_FILE>
  <RESPONSE MSGID="Message Id" TRUNK_PORT="Port Number"
    STATUS="SUCCESS, INVALID_TRUNK_PORT, FILE1_MISSING,
    FILE2_MISSING, VOICE_CHANNEL_BUSY, INVALID_CHANNEL_STATE">
  </RESPONSE>
</PLAY_VOICE_FILE>
```

Termination Status Play File - <PLAY_VOICE_FILE><TERMINATION>

Call Ahead Server sends this message to the applications program after it has played the voice file, or if the play gets interrupted by the called party when he pressed a DTMF key.

Format:

```
<PLAY_VOICE_FILE>  
  <TERMINATION MSGID="Message Id" TRUNK_PORT="Port Number"  
    EVENT="DTMF, LOOPDROP, NOISE, EOF, FASTBUSY">  
  </TERMINATION>  
</PLAY_VOICE_FILE>
```

Example 1:

```
<PLAY_VOICE_FILE>  
  < TERMINATION MSGID="786123" TRUNK_PORT="15" EVENT="DTMF">  
  </ TERMINATION >  
</PLAY_VOICE_FILE>
```

In the example 1, the listener pressed DTMF key and interrupted the playing of the message file.

Example 2:

```
<PLAY_VOICE_FILE>  
  < TERMINATION MSGID="786123" TRUNK_PORT="15" EVENT="EOF">  
  </ TERMINATION >  
</PLAY_VOICE_FILE>
```

In the example 2, the listener listened to the entire message..

Stop Playing of Voice File - <STOP_PLAY_VOICE_FILE><INFO>

The applications program sends this message to Call Ahead Server to stop the playing of a voice file in progress.

Format:

```
<STOP_PLAY_VOICE_FILE>
  <INFO MSGID="Message ID" TRUNK_PORT="Port Number">
  </INFO>
</STOP_PLAY_VOICE_FILE>
```

Response Play File - <STOP_PLAY_VOICE_FILE><RESPONSE>

Call Ahead Server sends this message immediately to the applications program if for any reason it can not stop the playing.

Format:

```
<STOP_PLAY_VOICE_FILE>
  <RESPONSE MSGID="Message Id" TRUNK_PORT="Port Number"
  STATUS="SUCCESS, INVALID_TRUNK_PORT, VOICE_CHANNEL_NOT_PLAYING,
  INVALID_CHANNEL_STATE">
  </RESPONSE>
</STOP_PLAY_VOICE_FILE>
```

Termination Status Play File - <STOP_PLAY_VOICE_FILE><TERMINATION>

Call Ahead Server sends this message to the applications program after it has successfully sopped the playing of the voice file.

```
<STOP_PLAY_VOICE_FILE>
  <TERMINATION MSGID="Message Id" TRUNK_PORT="Port Number"
  EVENT="SUCCESS">
  </TERMINATION>
</STOP_PLAY_VOICE_FILE>
```


Play Number

The applications program sends this message to Call Ahead Server to initiate a voice number play.

Play Number - <PLAY_NUMBER><INFO>

The applications program sends this message to Call Ahead Server and instructs it to play a number in different formats.

Format:

```
<PLAY_NUMBER>
  <INFO MSGID="Message ID" NUMBER="Number Data" TRUNK_PORT="Port
  Number" MODE="INTERRUPTABLE, UNINTERRUPTABLE", FORMAT="INTEGERS,
  DECIMAL">
  </INFO>
</PLAY_NUMBER>
```

Play Number Response - <PLAY_NUMBER><RESPONSE>

Call Ahead Server sends this message immediately to the applications program with information related to the validity of the message.

Format:

```
<PLAY_NUMBER>
  <RESPONSE MSGID="Message Id" TRUNK_PORT="Port Number"
  STATUS="SUCCESS, NOT_NUMBER">
  </RESPONSE>
</PLAY_NUMBER>
```

Play Number Termination Status - <PLAY_NUMBER><TERMINATION>

Call Ahead Server sends this message to the applications program after the number has been played to the caller or called party.

Format:

```
<PLAY_NUMBER>
  <TERMINATION MSGID="Message Id" TRUNK_PORT="Port Number"
  EVENT="DTMF, LOOP_DROP, NOISE, EOF, FAST_BUSY">
  </TERMINATION>
</PLAY_NUMBER>
```

Recording and Stopping Voice Messages

Record Voice File - <RECORD_VOICE_FILE><INFO>

The applications program sends this message to Call Ahead Server to initiate a voice record function.

Format:

```
<RECORD_VOICE_FILE>
  <INFO MSGID="Message ID" TRUNK_PORT="Port Number"
    FILE_NAME="file_name", TERM_CODE="DTMF MASK, SILENCE, LOOP_DROP,
    FAST_BUSY">
  </INFO>
</RECORD_VOICE_FILE_ >
```

Response Record Voice File - <RECORD_VOICE_FILE><RESPONSE>

Call Ahead Server sends this message immediately to the applications program after verifying the date sent by through the <INFO> message.

Format:

```
<RECORD_VOICE_FILE>
  <RESPONSE MSGID="Message Id" TRUNK_PORT="Port Number"
    STATUS="SUCCESS, FAILED_TO_CREATE_FILE, INVALID_TRUNK_PORT,
    INVALID_TERMCODE">
  </RESPONSE>
</RECORD_VOICE_FILE>
```

Termination Record Voice File - <RECORD_VOICE_FILE><TERMINATION>

Call Ahead Server sends this message to the applications program after the recording of the voice file has been completed.

Format:

```
<RECORD_VOICE_FILE>
  <TERMINATION MSGID="Message Id" TRUNK_PORT="Port Number"
    EVENT="DTMF, LOOP_DROP, NOISE, EOF, FAST_BUSY">
  </TERMINATION>
</RECORD_VOICE_FILE>
```

Stop Recording Voice - <STOP_RECORD_VOICE_FILE><INFO>

The applications program sends this message to Call Ahead Server to stop a recording session in progress.

Format:

```
<STOP_RECORD_VOICE_FILE>
  <INFO MSGID="Message ID" TRUNK_PORT="Port Number">
  </INFO>
</STOP_RECORD_VOICE_FILE>
```

Response Stop Recording - <STOP_RECORD_VOICE_FILE><RESPONSE>

Call Ahead Server sends this message immediately to the applications program after verifying the data sent by through the <INFO> message.

Format:

```
<STOP_RECORD_VOICE_FILE>
  <RESPONSE MSGID="Message Id" TRUNK_PORT="Port Number"
  STATUS="SUCCESS, INVALID_TRUNK_PORT, VOICE_CHANNEL_NOT_RECORDING,
  INVALID_CHANNEL_STATE">
  </RESPONSE>
</STOP_RECORD_VOICE_FILE>
```

Termination Stop Record - <STOP_RECORD_VOICE_FILE><TERMINATION>

Call Ahead Server sends this message to the applications program after the recording of the voice file has been stopped.

Format:

```
<STOP_RECORD_VOICE_FILE>
  <TERMINATION MSGID="Message Id" TRUNK_PORT="Port Number"
  EVENT="SUCCESS">
  </TERMINATION>
</STOP_RECORD_VOICE_FILE>
```

Receive or Generate DTMF

Receive DTMF Digits - <DTMF_RECEIVE><INFO>

The applications program instructs Call Ahead Server to receive DTMF digits.

Format:

```
<DTMF_RECEIVE>
  <INFO MSGID="Message ID" TRUNK_PORT="Port Number"
    NUMBER_OF_DIGITS="count"
  </INFO>
</DTMF_RECEIVE>
```

Response Receive DTMF Digits - <DTMF_RECEIVE><RESPONSE>

Call Ahead Server sends this message to the applications program after verifying the data in the <INFO> message.

Format:

```
<DTMF_RECEIVE>
  <RESPONSE MSGID="Message Id" TRUNK_PORT="Port Number"
    STATUS="SUCCESS, TOO_MANY_DIGITS">
  </RESPONSE>
</DTMF_RECEIVE>
```

Termination Receive DTMF Digits - <DTMF_RECEIVE><TERMINATION>

Call Ahead Server sends this message to the applications program after it has receive all the DTMF digits.

Format:

```
<DTMF_RECEIVE>
  <TERMINATION MSGID="Message Id" TRUNK_PORT="Port Number"
    DATA="DTMF data" EVENT="DTMF, LOOP_DROP, NOISE, EOF, FAST_BUSY">
  </TERMINATION>
</DTMF_RECEIVE>
```

Generate DTMF Digits - <DTMF_GENERATE><INFO>

The applications program instructs Call Ahead Server to generate DTMF digits.

Format:

```
<DTMF_GENERATE>
  <INFO MSGID="Message ID" TRUNK_PORT="Port Number"
    DTMF_STRING="Data">
  </INFO>
</DTMF_GENERATE>
```

Generate DTMF Digits Response - <DTMF_GENERATE><RESPONSE>

Call Ahead Server sends this message immediately to the application program after determining the validity of the data.

Format:

```
<DTMF_GENERATE>
  <RESPONSE MSGID="Message Id" TRUNK_PORT="Port Number"
    STATUS="SUCCESS, TOO_MANY_DIGITS">
  </RESPONSE>
</DTMF_GENERATE>
```

Generate DTMF Digits Done - <DTMF_GENERATE><TERMINATION>

Call Ahead Server sends this message to the applications program after it has completed generating the DTMF digits.

Format:

```
<DTMF_GENERATE>
  <TERMINATION MSGID="Message Id" TRUNK_PORT="Port Number"
    EVENT="NORMAL, FAILED">
  </TERMINATION>
</DTMF_GENERATE>
```

Table of Contents

Introduction	1
Call Ahead Server XML Protocol	2
TCP/IP Port	2
Call Ahead Server Parameters	3
XML Message	3
Administrative Messages	4
Link Initialization and Termination Messages	4
Initialize Link - <INIT_LINK><INFO>	4
Response - <INIT_LINK><RESPONSE>	5
Terminate Link - <TERMINATE_LINK><INFO>	6
Terminate Response - <TERMINATE_LINK><RESPONSE>	6
Pause Link Message - <PAUSE_LINK><INFO>	7
Pause Response - <PAUSE_LINK><RESPONSE>	7
Exit Message - <EXIT_LINK><INFO>	8
Exit Response - <EXIT_LINK><RESPONSE>	8
Outbound Call Control and Switching Messages	9
Dial-out Messages	9
Outbound Dial-out - <DIALOUT><DIAL>	9
Immediate Response - <DIALOUT> <RESPONSE>	10
Call Progress Status - <DIALOUT><TERMINATION>	10
Disconnect Messages	12
Application Disconnect - <APP_TRUNK_DISCONNECT><INFO>	12
Response - <APP_TRUNK_DISCONNECT><RESPONSE>	12
Network Initiated Disconnect - <NETWORK_TRUNK_DISCONNECT><INFO>	14
Application Disconnect Response - <NETWORK_TRUNK_DISCONNECT><RESPONSE>	14
Initiate Digital Transfer and Patching	15
Dial and Patch - <DIAL_PATCH><INFO>	15
Dial and Patch Response - <DIAL_PATCH><RESPONSE>	16
Dial and Patch Done - <DIAL_PATCH><TERMINATION>	16
Patch Ports - <PATCH_PORTS><INFO>	17
Response Patch Ports - <PATCH_PORTS><RESPONSE>	17
Patching of Ports Done - <PATCH_PORTS><TERMINATION>	18
Tearing Down a Patch - <TEAR_PATCH><INFO>	19
Tear Down Patch Response - <TEAR_PATCH><RESPONSE>	19
Tear Down Patch Done - <TEAR_PATCH><TERMINATION>	20
Inbound Call Control and Switching Messages	21
Inbound Call	21
Receive Call - <RECEIVE_CALL><INFO>	21
Response Receive Call - <RECEIVE_CALL><RESPONSE>	21
Voice Processing Messages	22
Playing and Stopping of Voice Message Files	22
Play Voice File - <PLAY_VOICE_FILE><INFO>	22

Response Play File - <PLAY_VOICE_FILE><RESPONSE> _____	22
Termination Status Play File - <PLAY_VOICE_FILE><TERMINATION> _____	23
Stop Playing of Voice File - <STOP_PLAY_VOICE_FILE><INFO> _____	24
Response Play File - <STOP_PLAY_VOICE_FILE><RESPONSE> _____	24
Termination Status Play File - <STOP_PLAY_VOICE_FILE><TERMINATION> _____	24
Play Number _____	25
Play Number - <PLAY_NUMBER><INFO> _____	25
Play Number Response - <PLAY_NUMBER><RESPONSE> _____	25
Play Number Termination Status - <PLAY_NUMBER><TERMINATION> _____	25
Recording and Stopping Voice Messages _____	26
Record Voice File - <RECORD_VOICE_FILE><INFO> _____	26
Response Record Voice File - <RECORD_VOICE_FILE><RESPONSE> _____	26
Termination Record Voice File - <RECORD_VOICE_FILE><TERMINATION> _____	26
Stop Recording Voice - <STOP_RECORD_VOICE_FILE><INFO> _____	27
Response Stop Recording - <STOP_RECORD_VOICE_FILE><RESPONSE> _____	27
Termination Stop Record - <STOP_RECORD_VOICE_FILE><TERMINATION> _____	27
Receive or Generate DTMF _____	28
Receive DTMF Digits - <DTMF_RECEIVE><INFO> _____	28
Response Receive DTMF Digits - <DTMF_RECEIVE><RESPONSE> _____	28
Termination Receive DTMF Digits - <DTMF_RECEIVE><TERMINATION> _____	28
Generate DTMF Digits - <DTMF_GENERATE><INFO> _____	29
Generate DTMF Digits Response - <DTMF_GENERATE><RESPONSE> _____	29
Generate DTMF Digits Done - <DTMF_GENERATE><TERMINATION> _____	29

Call Ahead[®] XML

Message Formats

Technical Memorandum

May 1, 2011

Parwan Electronics Corporation
1230 Highway 34
Aberdeen, New Jersey, 07747 - U.S.A.

Tel: (732) 290-1900

Fax: (732) 566-8771

Confidential Information – www.voicesaver.com
***DRAFT 2 *** Page 32/32