

# COMPUTER TELEPHONY SERVER XML MESSAGES

---

---

DRAFT 33

**Parwan Electronics Corporation**  
**1230 Highway 34**  
**Aberdeen, NJ 07747**  
**(732)-290-1900 x 3100, Fax: (732)-692-6587**  
**[www.voicesaver.com](http://www.voicesaver.com)**

Confidential Information  
Page 1 of 89

DRAFT 33

Confidential Information  
Page 2 of 89

## Table of Contents

<i>Table of Contents</i> .....	3
<i>Description</i> .....	7
<i>TCP/IP Port</i> .....	7
<i>Trunk or Agent Port Offset</i> .....	7
<i>Computer Telephony Server Parameters</i> .....	7
<i>XML Message</i> .....	8
<i>Administrative Messages</i> .....	9
<b>Link Initialization and Termination Messages</b> .....	9
Initialize Link - <INIT_LINK><INFO> .....	9
Response - <INIT_LINK><RESPONSE>.....	10
Terminate Link - <TERMINATE_LINK><INFO>.....	11
Terminate Response - <TERMINATE_LINK><RESPONSE> .....	11
Pause Link Message - <PAUSE_LINK><INFO>.....	12
Pause Response - <PAUSE_LINK><RESPONSE>.....	12
Exit Message - <EXIT_LINK><INFO>.....	13
Exit Response - <EXIT_LINK><RESPONSE>.....	13
<b>Mailbox Administration - &lt;MAILBOX&gt; Element</b> .....	14
Update a Mailbox - <MAILBOX><UPDATE> .....	15
Update Response - <MAILBOX> <UPDATE_RESPONSE> .....	15
Update a Mailbox Record - <MAILBOX><UPDATE_SER> .....	16
Update Record Response - <MAILBOX> <UPDATE_SER_RESPONSE>.....	16
Add a New Mailbox - <MAILBOX> <ADD> .....	17
Add Response - <MAILBOX> <ADD_RESPONSE> .....	17
Mailbox Information - <MAILBOX> <INFO> .....	18
Information Response - <MAILBOX> <INFO_RESPONSE> .....	18
Delete Mailbox - <MAILBOX> <DELETE>.....	19
Delete Response - <MAILBOX> <DELETE_RESPONSE> .....	19
Check the Existence of a Mailbox - <MAILBOX> <EXISTS>.....	20
Existence Response - <MAILBOX> <EXISTS_RESPONSE>.....	20
<b>Message Waiting Signals - &lt;MSG_WAIT&gt; Element</b> .....	21
Message Waiting Information - <MSG_WAIT><INFO> .....	21
Message Waiting Response - <MSG_WAIT><RESPONSE> .....	22
<b>Tree Setup Messages</b> .....	23
<i>Outbound Call Control and Switching Messages</i> .....	24
<b>Dial-out Messages</b> .....	24
Outbound Dial-out - <DIALOUT><DIAL> .....	24
Immediate Response - <DIALOUT> <RESPONSE> .....	25
Call Progress Status - <DIALOUT><TERMINATION>.....	25
<b>Dial-out Messages with IVR Activation</b> .....	27
Outbound Dial-out IVR - <DIALOUT_IVR><DIAL> .....	27
Immediate Response - <DIALOUT_IVR> <RESPONSE>.....	29

Confidential Information

Page 3 of 89

Call Progress Status - <DIALOUT_IVR><TERMINATION> .....	29
Termination Status - <EXEC_IVR_MAILBOX><TERMINATION> .....	30
<b>Callback Messages .....</b>	<b>31</b>
Callback - <CALLBACK><INITIATE>.....	31
Immediate Response - <CALLBACK> <RESPONSE>.....	34
<b>Trunk Status Messages .....</b>	<b>35</b>
Applications Trunk Status - <APP_TRUNK_STATUS><INFO> .....	35
Response - <APP_TRUNK_STATUS><RESPONSE> .....	35
<b>Disconnect Messages .....</b>	<b>36</b>
Applications Disconnect - <APP_TRUNK_DISCONNECT><INFO>.....	36
Response - <APP_TRUNK_DISCONNECT><RESPONSE> .....	36
Network Initialed Disconnect - <NETWORK_TRUNK_DISCONNECT><INFO> .....	37
Applications Response - <NETWORK_TRUNK_DISCONNECT><RESPONSE> .....	37
<b>Initiate Digital Transfer, Mute, Hold, and Patching.....</b>	<b>38</b>
Transfer an Agent - <AGENT_TRANSFER><INFO> .....	38
Transfer Agent Response - <AGENT_TRANSFER><RESPONSE> .....	39
Mute Function - <MUTE><INFO> .....	42
Mute Function Response - <MUTE><RESPONSE> .....	43
Hold Customer Function - <HOLD><INFO> .....	44
Hold Customer Response - <HOLD><RESPONSE>.....	45
Dial and Patch - <DIAL_PATCH><INFO> .....	46
Dial and Patch Response - <DIAL_PATCH><RESPONSE> .....	46
Dial and Patch Done - <DIAL_PATCH><TERMINATION>.....	47
Patch Ports - <PATCH_PORTS><INFO> .....	48
Response Patch Ports - <PATCH_PORTS><RESPONSE>.....	48
Patching of Ports Done - <PATCH_PORTS><TERMINATION> .....	49
Tearing Down a Patch - <TEAR_PATCH><INFO>.....	50
Tear Down Patch Response - <TEAR_PATCH><RESPONSE>.....	50
Tear Down Patch Done - <TEAR_PATCH><TERMINATION>.....	51
<b>Initiate Conference Setup .....</b>	<b>52</b>
Conference Call Setup - <INIT_CONFERENCE><INFO> .....	52
Response Conference Call Setup - <INIT_CONFERENCE><RESPONSE> .....	52
Conference Link Setup - <INIT_CONFERENCE_LINK><INFO>.....	53
Response Conference Link Setup - <INIT_CONFERENCE_LINK><RESPONSE> .....	54
<b>Receive Call Conference .....</b>	<b>55</b>
Inbound Conference Call - <INBOUND_CONFERENCE><INFO> .....	55
Response Conference Call Setup - <INBOUND_CONFERENCE><RESPONSE>.....	55
<b>Change Conferee Mode .....</b>	<b>56</b>
Change Mode of the Conferee - <CHANGE_CONFEREE_MODE><INFO> .....	56
Response - <CHANGE_CONFEREE_MODE><RESPONSE> .....	57
<b>Active Talkers.....</b>	<b>58</b>
Report the List of Active Talkers - <ACTIVE_TALKERS_LIST><INFO> .....	58
Response - <ACTIVE_TALKER_LIST><RESPONSE> .....	58
<b>Agent Login and Logout.....</b>	<b>59</b>
Agent Log In - <AGENT_LOGIN><INFO>.....	59
Response Agent Log In - <AGENT_LOGIN><RESPONSE>.....	59
Agent Log Out - <AGENT_LOGOUT><INFO> .....	60
Response Agent Log Out - <AGENT_LOGOUT><RESPONSE> .....	60

.....	<b>60</b>
<b>Barging In (Optional Feature)</b> .....	<b>61</b>
Barge In Message - <BARGE_IN><INFO> .....	61
Response Barge In - <BARGE_IN><RESPONSE>.....	61
<b>Barging Out (Optional Feature)</b> .....	<b>63</b>
Barge Out Message - <BARGE_OUT><INFO> .....	63
Response Barge Out - <BARGE_OUT><RESPONSE> .....	63
<b><i>Inbound Call Control and Switching Messages</i></b> .....	<b>65</b>
<b>Inbound Call</b> .....	<b>65</b>
Receive Call - <RECEIVE_CALL><INFO> .....	65
Response Receive Call - <RECEIVE_CALL><RESPONSE>.....	65
Handle Inbound Call - <HANDLE_INBOUND_CALL><INFO>.....	66
Response Handle Inbound - <HANDLE_INBOUND_CALL><RESPONSE> .....	66
Accept the Call - <ACCEPT_CALL><INFO>.....	67
Accept Call Response - <ACCEPT_CALL><RESPONSE>.....	67
Termination Status Accept Call - <ACCEPT_CALL><TERMINATION>.....	67
Decline the Call - <DECLINE_CALL><INFO>.....	68
Decline Call Response - <DECLINE_CALL><RESPONSE>.....	68
Termination Status Decline Call - <DECLINE_CALL><TERMINATION>.....	68
<b><i>Voice Processing Messages</i></b> .....	<b>69</b>
<b>Playing and Stopping of Voice Message Files</b> .....	<b>69</b>
Play Voice File - <PLAY_VOICE_FILE><INFO> .....	69
Response Play File - <PLAY_VOICE_FILE><RESPONSE>.....	69
Termination Status Play File - <PLAY_VOICE_FILE><TERMINATION> .....	70
Stop Playing of Voice File - <STOP_PLAY_VOICE_FILE><INFO> .....	71
Response Play File - <STOP_PLAY_VOICE_FILE><RESPONSE> .....	71
Termination Status Play File - <STOP_PLAY_VOICE_FILE><TERMINATION> .....	71
<b>Play Number</b> .....	<b>73</b>
Play Number - <PLAY_NUMBER><INFO> .....	73
Play Number Response - <PLAY_NUMBER><RESPONSE>.....	73
Play Number Termination Status - <PLAY_NUMBER><TERMINATION>.....	73
<b>Recording and Stopping Voice Messages</b> .....	<b>74</b>
Record Voice File - <RECORD_VOICE_FILE><INFO> .....	74
Response Record Voice File - <RECORD_VOICE_FILE><RESPONSE> .....	74
Termination Record Voice File - <RECORD_VOICE_FILE><TERMINATION> .....	74
Stop Recording Voice - <STOP_RECORD_VOICE_FILE><INFO> .....	75
Response Stop Recording - <STOP_RECORD_VOICE_FILE><RESPONSE> .....	75
Termination Stop Record - <STOP_RECORD_VOICE_FILE><TERMINATION> .....	75
<b>Receive or Generate DTMF</b> .....	<b>76</b>
Receive DTMF Digits - <DTMF_RECEIVE><INFO>.....	76
Response Receive DTMF Digits - <DTMF_RECEIVE><RESPONSE>.....	76
Termination Receive DTMF Digits - <DTMF_RECEIVE><TERMINATION>.....	76
Generate DTMF Digits - <DTMF_GENERATE><INFO>.....	77
Generate DTMF Digits Response - <DTMF_GENERATE><RESPONSE> .....	77
Generate DTMF Digits Done - <DTMF_GENERATE><TERMINATION>.....	77
<b>Starting and Stopping of Two Way Voice Recordings</b> .....	<b>78</b>
Record Two Way Voice - <RECORD_TWO_WAY_VOICE><INFO> .....	78

Two Way Record Response - <RECORD_TWO_WAY_VOICE><RESPONSE> .....	78
Two Way Record Done - <RECORDED_TWO_WAY_VOICE><TERMINATION>.....	79
Stop Two Way Record - <STOP_RECORD_TWO_WAY_VOICE><INFO> .....	80
Stop Two Way Record Response - <STOP_RECORD_TWO_WAY_VOICE> <RESPONSE>.....	80
Stop Two Way Record Done - < STOP_RECORD_TWO_WAY_VOICE> <TERMINATION> .....	81
<b>Attach a Trunk to a Mailbox or Execute and IVR.....</b>	<b>82</b>
Message for a Mailbox - <ATTACH_MIALBOX><INFO> .....	82
Response - <ATTACH_MIALBOX><RESPONSE> .....	83
Execute IVR Mailbox Call Flow - <EXEC_IVR_MAILBOX><INFO> .....	84
Response - <EXEC_IVR_MAILBOX><RESPONSE> .....	85
Termination Status - <EXEC_IVR_MAILBOX><TERMINATION> .....	85
Stop IVR Mailbox Execution - <STOP_EXEC_IVR_MAILBOX><INFO>.....	87
Response Stop Call Flow - <STOP_EXEC_IVR_MAILBOX><RESPONSE>.....	87
<b>Appendix A .....</b>	<b>88</b>
Reading <EXEC_IVR_MAILBOX> Tag Variables by the IVR .....	88
Setting <EXEC_IVR_MAILBOX> Tag Variables from the IVR .....	89

DRAFT

## TECHNICAL MEMORANDUM

### Description

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

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

### TCP/IP Port

Computer Telephony 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 Computer Telephony Server and then communicate with it afterwards.

### Trunk or Agent Port Offset

The Computer Telephony Server parameter eee81 specifies the trunk offset to be used. For example if eee81 is set to 256, then the trunk port 258 by the applications program means trunk port 2 on the Computer Telephony Server. This parameter is used for cases where there is one applications server and multiple Computer Telephone Servers.

### Computer Telephony Server Parameters

The following Computer Telephony 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
eee81	0	Trunk offset.

## **XML Message**

Computer Telephony 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

DRAFT 33



## Administrative Messages

The Computer Telephony Server Administrative XML messages are used to initialize the link, maintain mailboxes, set up trees, change email addresses, change follow-me phone numbers, and to perform other administrative functions.

### Link Initialization and Termination Messages

#### Initialize Link - <INIT\_LINK><INFO>

The applications program sends this message to Computer Telephony 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, Computer Telephony 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, Computer Telephony Server sends the <RECEIVE\_CALL> message to the applications program.

In order for Computer Telephony 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, Computer Telephony Server receives the message from the applications program and only establishes the link for making outbound calls.

## Response - <INIT\_LINK><RESPONSE>

Computer Telephony 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 Computer Telephony 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>

Computer Telephony 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 Computer Telephony 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>**

Computer Telephony 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>

Computer Telephony Server sends this message to the Application program after the operator enters the exit command from the Computer Telephony 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 Computer Telephony Server in response to the <EXIT\_LINK><INFO> message.

Format:

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

## **Mailbox Administration - <MAILBOX> Element**

The applications program instructs Computer Telephony Server to add, update, delete, or check a mailbox.

### **Acceptable Elements within MAILBOX**

<b>Element</b>	<b>Description</b>
ADD	Create a new mailbox
DELETE	Remove an existing mailbox
EXISTS	Check to see if a mailbox exists
INFO	Get the info of an existing mailbox
UPDATE	Modify an existing mailbox
UPDATE_SER	Modify an existing mailbox based on the absolute record number

### **Required Attributes for All Elements and Responses within <MAILBOX>**

<b>Attribute</b>	<b>Description</b>
MISGID	minimum length 4, maximum length 50, THIS IS AN ARBITRARY TRANSACTION ID USED FOR IDENTIFICATION
BOXNO	minimum length 1, maximum length 10, THIS IS THE MAILBOX NUMBER

## Update a Mailbox - <MAILBOX><UPDATE>

The application program sends this message to Computer Telephony Server to update the fields on an existing mailbox.

### Optional Attributes for <UPDATE> Element

Attribute	Description
BOXNAME	min length 1, max length 50, THIS IS THE MAILBOX NAME
BOXEMAIL	string variable, min length 3, max length 50, THIS IS THE EMAIL ADDRESS OF THE MAILBOX OWNER
BOXPWD	min length 4, max length 10, THIS IS THE PASSWORD TO ACCESS THIS MAILBOX VIA PHONE
EXTENS08	Hot line phone number for Call Back or another Follow-Me number.
EXTENS09	Call back phone number for the Call Back or Follow-Me service. All characters must be numeric

### Example:

```
<MAILBOX>
  <UPDATE MSGID="20040822112200ABC" BOXNO="156" BOXNAME="John Doe"
    BOXEMAIL="johndoe@aol.com" BOXPWD="9595" EXTENS09="011911123561241">
  </UPDATE>
</MAILBOX>
```

## Update Response - <MAILBOX> <UPDATE\_RESPONSE>

Computer Telephony Server sends this message to the application program in response to the <MAILBOX><UPDATE> message.

### Attributes:

Attribute	Description
STATUS	SUCCESS" or "FAILED
ERRORCODE	Optional attribute, 5 digits, THIS IS THE ERROR CODE IF MAILBOX UPDATE FAILED
ERRORMSG	Optional attribute, min. 1 char, max 100 char, THIS IS THE ERROR MESSAGE IF MAILBOX UPDATE FAILED

### Example:

```
<MAILBOX>
  <UPDATE_RESPONSE MSGID="20040822112200ABC" BOXNO="156"
    STATUS="FAILED" ERRORCODE="4015" ERRORMSG="BOX DOES NOT EXIST">
  </UPDATE_RESPONSE>
</MAILBOX>
```

## Update a Mailbox Record - <MAILBOX><UPDATE\_SER>

The application program sends this message to Computer Telephony Server to update the fields on an existing mailbox based on its absolute record number in the file.

### Optional Attributes for <UPDATE\_SER> Element

Attribute	Description
RECNO	Absolute record number in the Master.dat file
BOXNAME	min length 1, max length 50, THIS IS THE MAILBOX NAME
BOXEMAIL	string variable, min length 3, max length 50, THIS IS THE EMAIL ADDRESS OF THE MAILBOX OWNER
BOXPWD	min length 4, max length 10, THIS IS THE PASSWORD TO ACCESS THIS MAILBOX VIA PHONE
BOXMSGS	Number of Messages allowed in the Box. "0" means the mailbox is disabled.

### Example:

```
<MAILBOX>
  <UPDATE_SER MSGID="20040822112200ABJ" RECONO="232" BOXNO="156"
    BOXNAME="John Doe" BOXEMAIL="johndoe@aol.com" BOXPWD="9595"
    BOXMSGS="100">
  </UPDATE_SER>
</MAILBOX>
```

## Update Record Response - <MAILBOX> <UPDATE\_SER\_RESPONSE>

Computer Telephony Server sends this message to the application program in response to the <MAILBOX><UPDATE\_SER> message.

### Attributes:

Attribute	Description
STATUS	SUCCESS" or "FAILED
ERRORCODE	Optional attribute, 5 digits, THIS IS THE ERROR CODE IF MAILBOX UPDATE FAILED
ERRORMSG	Optional attribute, min. 1 char, max 100 char, THIS IS THE ERROR MESSAGE IF MAILBOX UPDATE FAILED

### Example:

```
<MAILBOX>
  <UPDATE_SER_RESPONSE MSGID="20040822112200ABJ" RECNO="232"
    BOXNO="156" STATUS="FAILED" ERRORCODE="4017" ERRORMSG="RECNO
    DOES NOT EXIST">
  </UPDATE_SER_RESPONSE>
</MAILBOX>
```



## Add a New Mailbox - <MAILBOX> <ADD>

The application program sends this message to Computer Telephony Server in order to add a new mailbox.

### Optional Attributes for <ADD> Element

Attribute	Description
SRCBOXNO	minimum length 1, maximum length 10, THIS IS THE MAILBOX NUMBER TO COPY THE DATA FROM.
BOXNAME	min length 1, max length 50, THIS IS THE MAILBOX NAME
BOXEMAIL	string variable, min length 3, max length 50, THIS IS THE EMAIL ADDRESS OF THE MAILBOX OWNER
BOXPWD	min length 4, max length 10, THIS IS THE PASSWORD TO ACCESS THIS MAILBOX VIA PHONE

### Example:

```
<MAILBOX>  
  <ADD MSGID="20040822112200ABC" SRCBOXNO="223" BOXNO="156"  
    BOXNAME="John Doe" BOXEMAIL="johndoe@aol.com" BOXPWD="9595">  
  </ADD>  
</MAILBOX>
```

## Add Response - <MAILBOX> <ADD\_RESPONSE>

Computer Telephony Server sends this message to the applications program in response to the <MAILBOX><ADD> message.

### Attributes:

Attribute	Description
STATUS	SUCCESS" or "FAILED
ERRORCODE	Optional attribute, 5 digits, THIS IS THE ERROR CODE IF MAILBOX ADD FAILED
ERRORMSG	Optional attribute, min. 1 char, max 100 char, THIS IS THE ERROR MESSAGE IF MAILBOX ADD FAILED

### Example:

```
<MAILBOX>  
  <ADD_RESPONSE MSGID="20040822112200ABC" BOXNO="156"  
    STATUS="FAILED" ERRORCODE="4018" ERRORMSG="BOX ALREADY EXISTS">  
  </ADD_RESPONSE>  
</MAILBOX>
```

## Mailbox Information - <MAILBOX> <INFO>

The applications program sends this message to Computer Telephony Server to receive information about the fields of a mailbox.

### Optional Arguments for <INFO> Function

There are no optional arguments for the <INFO> function

#### Example:

```
<MAILBOX>
  <INFO MSGID="20040822112200ABC" BOXNO="156" >
  </INFO>
</MAILBOX>
```

## Information Response - <MAILBOX> <INFO\_RESPONSE>

Computer Telephony Server sends this message to the applications program in response to the <MAILBOX><INFO> message.

#### Attributes:

Attribute	Description
STATUS	SUCCESS or FAILED
BOXNAME	min length 1, max length 50, THIS IS THE MAILBOX NAME
BOXEMAIL	string variable, min length 3, max length 50, THIS IS THE EMAIL ADDRESS OF THE MAILBOX OWNER
BOXPWD	min length 4, max length 10, THIS IS THE PASSWORD TO ACCESS THIS MAILBOX VIA PHONE
BOXTYPE	Mailbox type, e.g. W, U, E, J, or other Types
NEWMMSG	Number of new messages that have not been read.
TOTMSG	Total number of messages in the Box.
TOTFAXES	Total number of faxes received (Optional software).
ERRORCODE	5 digits, THIS IS THE ERROR CODE IF MAILBOX UPDATE FAILED
ERRORMSG	min. 1 char, max 100 char, THIS IS THE ERROR MESSAGE IF MAILBOX UPDATE FAILED

#### Example:

```
<MAILBOX>
  <INFO_RESPONSE MSGID="20040822112200ABC" BOXNO="156"
  STATUS="SUCCESS" BOXNAME="John Doe" BOXEMAIL="johndoe@aol.com"
  BOXPWD="9595" BOXTYPE="U" NEWMMSG="3" TOTMSG="53" >
  </INFO_RESPONSE>
</MAILBOX>
```

## Delete Mailbox - <MAILBOX> <DELETE>

The application program sends this message to Computer Telephony Server to delete a mailbox.

### Optional Attributes for <DELETE> Element

There are no optional attributes for the <DELETE> element

#### Example:

```
<MAILBOX>  
  <DELETE MSGID="20040822112200ABC" BOXNO="156" >  
  </DELETE>  
</MAILBOX>
```

## Delete Response - <MAILBOX> <DELETE\_RESPONSE>

Computer Telephony Server sends this message to the application program in response to the <MAILBOX><DELETE> message.

#### Attributes:

Attribute	Description
STATUS	SUCCESS or FAILED
RESPONSE	"EXISTSYES" or "EXISTSNO"
ERRORCODE	5 digits, THIS IS THE ERROR CODE IF MAILBOX UPDATE FAILED
ERRORMSG	min. 1 char, max 100 char, THIS IS THE ERROR MESSAGE IF MAILBOX UPDATE FAILED

#### Example:

```
<MAILBOX>  
  <DELETE_RESPONSE MSGID="20040822112200ABC" BOXNO="156"  
  STATUS="SUCCESS" RESPONSE="EXISTSYES">  
  </DELETE_RESPONSE>  
</MAILBOX>
```

## Check the Existence of a Mailbox - <MAILBOX> <EXISTS>

The applications program sends this message to Computer Telephony Server in order to find out if a mailbox exists.

### Optional Attributes for <EXISTS> Element

There are no optional attributes for the <EXISTS> element

#### Example:

```
<MAILBOX>  
  <EXISTS MSGID="20040822112200ABC" BOXNO="156" >  
  </EXISTS>  
</MAILBOX>
```

## Existence Response - <MAILBOX> <EXISTS\_RESPONSE>

Computer Telephony Server sends this message to the applications program in responses to the <MAILBOX><EXISTS> message.

#### Attributes:

Attribute	Description
STATUS	SUCCESS or FAILED
RESPONSE	"EXISTSYES" or "EXISTSNO"
ERRORCODE	5 digits, THIS IS THE ERROR CODE IF MAILBOX UPDATE FAILED
ERRORMSG	min. 1 char, max 100 char, THIS IS THE ERROR MESSAGE IF MAILBOX UPDATE FAILED

#### Example:

```
<MAILBOX>  
  <EXISTS_RESPONSE MSGID="20040822112200ABC" BOXNO="156"  
  STATUS="SUCCESS" RESPONSE="EXISTSYES">  
  </EXISTS_RESPONSE>  
</MAILBOX>
```

## Message Waiting Signals - <MSG\_WAIT> Element

### Message Waiting Information - <MSG\_WAIT><INFO>

Computer Telephony Server sends this message to the applications program to let it know that there are new messages for a subscriber.

#### Format:

```
<MSG_WAIT>
  <INFO MSGID="Message ID" BOXNO="Mailbox No" BOXPWD="Passcode"
    BOXNAME="Name" DNIS="Dialed Number" ANI="Calling Number"
    FILE="Message File" FORMAT="WAV, VOX" UNREAD_MSGS="NEW MESSAGE
    COUNT" TOTAL_MSGS="Total Message Count">
  </INFO>
</MSG_WAIT>
```

#### Attributes:

Attribute	Min	Max	Opt	Description
MSGID	4	20	N	This is an arbitrary Transaction ID used for Identification.
BOXNO	1	10	N	This is the Mailbox number
BOXPWD	4	10	Y	This is the mailbox pass code.
BOXNAME	3	40	Y	Mailbox name
DNIS	3	14	N	Maximum 14 digits, the dialed number.
ANI	3	14	Y	Optional - Calling Party ID, Maximum 10 digits.
FILE	10	50	N	File name of the newest message.
FORMAT	3	3	Y	Record file format, WAV or VOX
UNREAD_MSGS	3	4	N	Count of unread messages
TOTAL_MSG	3	4	N	Count of total messages.

#### Example:

```
<MSG_WAIT>
  <INFO MSGID="67431" BOXNO="1015" BOXNAME="John Smith"
    DNIS="7322901900" ANI="7325365635"
    FILE="C:\PEC\UMSG\1015_10242004_02.VOX" FORMAT="vox"
    UNREAD_MSGS="001" TOTAL_MSGS="003">
  </INFO>
</MSG_WAIT>
```

The above example show that the caller called from the phone number 732-536-5635 and dialed the number 732-290-1900. He selected the mailbox 1015, recorded a message in the file c:\pec\umsg\1015\_10242004\_02.vox in the VOX format. The mailbox 1015 belongs to John Smith. At the end of recording Computer Telephony Server calculated that there is one unread message and a total of three messages in the mailbox.

## Message Waiting Response - <MSG\_WAIT><RESPONSE>

The applications program sends this message to Computer Telephony Server to let is received the <MSG\_WAIT><INFO> successfully or the data is not valid..

### Format:

```
<MSG_WAIT>
  <RESPONSE MSGID="Message ID" BOXNO="Mailbox No" STATUS="SUCCESS,
    FAILED">
  </INFO>
</MSG_WAIT>
```

### Attributes:

Attribute	Min	Max	Opt	Description
MSGID	4	20	N	This is an arbitrary Transaction ID used for Identification.
BOXNO	1	10	N	This is the Mailbox number
STATUS	6	8	N	The status from the application program.

### Example:

```
<MSG_WAIT>
  <RESPONSE MSGID="67431" BOXNO="1015" STATUS="SUCCESS">
  </RESPONSE>
</MSG_WAIT>
```

## ***Tree Setup Messages***

This is used for set up trees.

DRAFT 33

## Outbound Call Control and Switching Messages

### Dial-out Messages

The applications program sends this data to Computer Telephony Server to initiate an outbound call activity. The first response message Computer Telephony 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 Computer Telephony Server to initiate an outbound call and performs other activities after the called party answers the call.

#### 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.
AGENTS	1	20	Y	Optional – Agent lines to connect to when the called party answers. Example: 24-27
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.
RINGS	1	2	N	Number of rings to wait for No answer
MODE	5	10	N	BLIND or SUPERVISED

#### Example:

```
<DIALOUT>  
  <DIAL MSGID="781234A5" CAMPID="4H CLUB" DNIS="3014567812"  
    ANI="8009871212" AGENTS="25-34" MODE="SUPERVISED"> </DIAL>  
</DIALOUT>
```

Dial the phone number 3014567812 and if the system detects connect, then patch the customer trunk port to a free agent port from 25 to 34.



**Example:**

```
<DIALOUT>
  <DIAL MSGID="781234A6" CAMPID="ROTARY CLUB" DNIS="3013456677"
    ANI="8005551212" MODE="SUPERVISED">
  </DIAL>
</DIALOUT>
```

Dial the number 3013456677 and after detecting connect, activate functions issued by other XML messages.

**Example:**

```
<DIALOUT>
  <DIAL MSGID="781234A6" CAMPID="ROTARY CLUB" DNIS="3014568812"
    ANI="8005551212" LIVEMESSAGE="MEETING.VOX",
    ANSMESSAGE="CALL.VOX", MODE="SUPERVISED">
  </DIAL>
</DIALOUT>
```

Dial the number 13014568812 and if there is a live connect, then play the file: MEETING.VOX and if there is an answering machine, then play the file: CALL.VOX.

**Immediate Response - <DIALOUT> <RESPONSE>**

Computer Telephony Server checks the data received and sends the message immediately to the applications program.

**Attributes:**

Attribute	Description
STATUS	SUCCESS, NUMBER_TOO_LONG, FAILED, BUSY, NO_AGENT_FREE, NO_MSI_AGENT_FREE, NO_DIALOUT_PORT_FREE, LIVEMESSAGE_FILE_MISSING, ANSMESSAGE_FILE_MISSING, FAILED

**Example:**

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

**Call Progress Status - <DIALOUT><TERMINATION>**

After dialing the number, Computer Telephony Server returns the status of the call progress to the applications program.

**Attributes:**

Attribute	Description
EVENT	CONNECT, BUSY, NOANS, INTERCEPT, ANSMACHINE, FAX, or

	NODIALTONE, EOF
DTMF	The DTMF data if the called party pressed a DTMF key.
TRUNK_PORT	Port number Computer Telephony Server used to dial out the number.
AGENT	Optional: Agent port the call was transferred to.

**Example:**

```
<DIALOUT>
  <TERMINATION MSGID="672312" EVENT="CONNECT" TRUNK_PORT="14"
    AGENT="26">
  </TERMINATION>
</DIALOUT>
```

**Example:**

```
<DIALOUT>
  <TERMINATION MSGID="672312" EVENT="CONNECT" TRUNK_PORT="14">
  </TERMINATION>
</DIALOUT>
```

After playing the message, the called party press the DTMF key 5.

```
<DIALOUT>
  <TERMINATION MSGID="672312" EVENT="EOF" TRUNK_PORT="14" DTMF="5">
  </TERMINATION>
</DIALOUT>
```

**Note:**

In order to ignore Agent, please set eee30 to 0.

## Dial-out Messages with IVR Activation

The applications program sends this data to Computer Telephony Server to initiate an outbound call activity and after detecting Connect, it jumps to an IVR application. The first response message Computer Telephony Server sends is the immediate message and the second response message is the result of the call progress for Busy, No Answer, or Intercept. The Connect response is sent to the application by the IVR with variable information.

### Required Attributes for All Elements and Responses within <DIALOUT\_IVR>

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

### Outbound Dial-out IVR - <DIALOUT\_IVR><DIAL>

The applications program sends this message to Computer Telephony Server to initiate an outbound call and after connect performs activities based on the logic in the IVR.

#### 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.
AGENTS	1	20	Y	Optional – Agent lines to connect to when the called party answers. Example: 24-27
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.
BOXNO	1	12	N	Box Number of the IVR Call flow to be executed after Connect
VAR1	1	40	Y	Variable name used in IVR Call flow
VAR2	1	40	Y	Variable name used in IVR Call flow
VAR3	1	40	Y	Variable name used in IVR Call flow
VAR4	1	40	Y	Variable name used in IVR Call flow
VAR5	1	40	Y	Variable name used in IVR Call flow
RINGS	1	2	N	Number of rings to wait for No answer
MODE	5	10	N	BLIND or SUPERVISED

**Example:**

```
<DIALOUT_IVR>  
  <DIAL MSGID="781234A5" CAMPID="4H CLUB" DNIS="3014567812"  
    ANI="8009871212" AGENTS="25-34" MODE="SUPERVISED"> </DIAL>  
</DIALOUT_IVR>
```

Dial the phone number 3014567812 and if the system detects connect, then patch the customer trunk port to a free agent port from 25 to 34.

**Example:**

```
<DIALOUT_IVR>  
  <DIAL MSGID="781234A6" CAMPID="ROTARY CLUB" DNIS="3013456677"  
    ANI="8005551212" VAR="JOHN SMITH" BOXNO="3017654534"  
    MODE="SUPERVISED">  
  </DIAL>  
</DIALOUT_IVR>
```

Dial the number 3013456677 and after detecting connect, jump to the mailbox 3017654534.

**Example:**

```
<DIALOUT_IVR>  
  <DIAL MSGID="781234A6" CAMPID="ROTARY CLUB" DNIS="3014568812"  
    ANI="8005551212" LIVEMESSAGE="MEETING.VOX",  
    ANSMESSAGE="CALL.VOX", MODE="SUPERVISED">  
  </DIAL>  
</DIALOUT_IVR>
```

Dial the number 13014568812 and if there is a live connect, then play the file: MEETING.VOX and if there is an answering machine, then play the file: CALL.VOX.

## Immediate Response - <DIALOUT\_IVR> <RESPONSE>

Computer Telephony Server checks the data received and sends the message immediately to the applications program.

### Attributes:

Attribute	Description
STATUS	SUCCESS, NUMBER_TOO LONG, FAILED, BUSY, NO_AGENT_FREE, NO_MSI_AGENT_FREE, NO_DIALOUT_PORT_FREE, LIVEMESSAGE_FILE_MISSING, ANSMESSAGE_FILE_MISSING, FAILED, INVALID_MAILBOX, INVALID_CHANNEL_STATE

### Example:

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

## Call Progress Status - <DIALOUT\_IVR><TERMINATION>

After dialing the number, Computer Telephony Server returns the status of the call progress to the applications program. For the IVR logic, the XML variables are set to the data as passed. The VAR5 is set to either "LIVE\_CONNECT" or "ANSWERING\_MACHINE". The IVR logic can make use of the XML variable to process the rest of the call.

### Attributes:

Attribute	Description
EVENT	CONNECT, BUSY, NOANS, INTERCEPT, ANSMACHINE, FAX, or NODIALTONE, EOF
DTMF	The DTMF data if the called party pressed a DTMF key.
TRUNK_PORT	Port number Computer Telephony Server used to dial out the number.
AGENT	Optional: Agent port the call was transferred to.

### Example:

```
<DIALOUT_IVR>  
  <TERMINATION MSGID="672312" EVENT="CONNECT" TRUNK_PORT="14"  
    AGENT="26">  
    </TERMINATION>  
</DIALOUT_IVR>
```

### Example:

```
<DIALOUT_IVR>
  <TERMINATION MSGID="672312" EVENT="CONNECT" TRUNK_PORT="14">
  </TERMINATION>
</DIALOUT_IVR>
```

After playing the message, the called party press the DTMF key 5.

```
<DIALOUT_IVR>
  <TERMINATION MSGID="672312" EVENT="EOF" TRUNK_PORT="14" DTMF="5">
  </TERMINATION>
</DIALOUT_IVR>
```

### Termination Status - <EXEC\_IVR\_MAILBOX><TERMINATION>

Computer Telephony Server sends this message to the applications program after it has Executed the IVR Call flow, or if the flow execution stopped by the application

#### Format:

```
<EXEC_IVR_MAILBOX>
  <TERMINATION MSGID="Message ID" BOXNO="Mailbox number"
  TRUNK_PORT="Port Number" VAR1="Variable1 value" VAR2="Variable2
  value" VAR3="Variable3 value"
  VAR4="Variable4 value" VAR5="Variable5 value" EVENT="EOCS, EOCF,
  STOPPED">
  </TERMINATION>
</EXEC_IVR_MAILBOX>
```

The EVENT="EOCS" indicates that the call flow was executed successfully and EVENT="EOCF" indicates that the call flow encountered errors and EVENT="STOPPED" indicates that the call flow was terminated by the <STOP\_EXE\_IVR\_MAILBOX>.

#### Example 1:

```
<EXEC_IVR_MAILBOX>
  <TERMINATION MSGID="786123" BOXNO="7325667661" TRUNK_PORT="1"
  VAR1="1234" VAR2="rec1.vox" VAR3="5678" VAR4="3456" VAR5="3333"
  EVENT="EOCS">
  </TERMINATION>
</EXEC_IVR_MAILBOX>
```

In the example 1, the Computer Telephony Server has executed the call flow of mailbox number 7325667661. The EVENT="EOCS" indicates that End Of Call flow Success.

Please read the "**Insight IVR XML External Tagged String Variables**" technical memorandum (**Appendix A**) to learn how to use the XML variables.

## **Callback Messages**

The applications program sends this data to Computer Telephony Server to initiate a Callback. Computer Telephony Server sends an immediate response after checking the message.

### **Callback - <CALLBACK><INITIATE>**

The applications program sends this message to Computer Telephony Server to invoke a callback to a subscriber. Since a callback can be initiated from many different sources, including a phone call, a website, an SMS, and an email, many options must be made available to allow the system to be flexible. Please notice the scenario remarks specified under each example given in order to utilize the call back function correctly.

Also, please note that when a call back message is sent to the Computer Telephony Server, Computer Telephony Server registers the transaction and takes an action based on the availability of system resources.

DRAFT

**Attributes:**

Attribute	Min	Max	Opt	Description
MSGID	4	50	N	This is the message ID.
BOXNO	1	10	N	This is the Mailbox or Subscriber account number. Generally, in a call back scenario, this is the DID.
LEG_ONE_NO	3	14	N	Subscriber's call back number. This is referred to as first leg of the call or LEG A.
LEG_TWO_NO	1	14	Y	The destination number the subscriber wishes to reach. This is referred to as second leg of the call or LEG B. If this field is not defined, then LEG A will be called and played a dial-tone.
ANI	4	10	Y	Generally the same as BOXNO, but could be different.
ACCT_TYPE	1	1	N	This is the account type. Values: 1 = PREPAID 2 = POSTPAID.
PLAY_BALANCE	1	1	Y	This specifies whether or not you wish a prepaid customer to hear a balance message. Values: 0 = NO 1 = YES
BALANCE	1	9	Y	If the account type is prepaid and LEG B is not specified, a balance can be specified. The balance should be sent as an integer. (i.e. One Dollar = 100 units)
LANGUAGE	1	2	Y	If any messages are to be played to the subscriber, this specifies which language to use. If this is not set, the language will be "1" for English. Values: 0 = English 1 = Spanish 2 = French 3 = German 4 = Arabic 5 = Farsi 6 = Chinese
CURRENCY	1	2	Y	If any currency related messages are to be played to the subscriber, this specified which currency type to play. Values: 0 = Dollars and Cents 1 = Pesos and Centavos 2 = Euros and Cents 3 = Pounds and Pence 4 = Francs and Centimes 5 = Rupees and Paisa 6 = Dinar and Fils
DURATION	1	9	Y	This is the duration allowed for this call in seconds. If both LEG A and LEG B of the call are specified, then this is the duration of the call incorporating the costs of both LEG A and LEG B. If only LEG A is specified, then this is the duration of the call incorporating ONLY LEG A.

**Example 1**



**Scenario:** A customer logs on to a website and initiates a call back by putting both the LEG A phone number and the LEG B phone number. The customer has a Prepaid account with \$10.00 balance. The rate to call prefix 01191\*\* is 20 cents per minute, and the rate to call 1212\*\* is 5 cents per minute.

```
<CALLBACK>
  <INITIATE MSGID="781234A5" LEG_ONE_NO="011911124123412"
    LEG_TWO_NO="12123404600" ANI="2124541212" ACCT_TYPE="1"
    DURATION="2400">
  </INITIATE>
</CALLBACK>
```

**NOTE:** If both LEG A and LEG B of the call are specified for a Prepaid account, only the fields shown in Example 1 are necessary. All other fields will be ignored.

## Example 2

**Scenario:** A customer sends an SMS from his cell phone and initiates a call back by putting ONLY the LEG A phone number. The customer has a Prepaid account with \$20.00 balance. You want the Computer Telephony Server to play the customer his account balance message in Spanish using Euros and Cents before he receives the dial-tone. The rate to call prefix 01191\*\* is 20 cents per minute.

```
<CALLBACK>
  <INITIATE MSGID="781234A5" LEG_ONE_NO="011911124123412"
    ANI="2124541212" ACCT_TYPE="1" PLAY_BALANCE="1" BALANCE="2000"
    LANGUAGE="1" CURRENCY="2" DURATION="6000" >
  </INITIATE>
</CALLBACK>
```

**NOTE:** If only LEG A is specified, Computer Telephony Server will call the subscriber at the phone number specified by LEG A and play a dial-tone. If "PLAY\_BALANCE" is set to 1, then the Computer Telephony Server will first play the customer his balance in the language and currency specified. If "PLAY\_BALANCE" is set to 0 then the BALANCE, LANGUAGE, and CURRENCY fields will be ignored.

## Example 3

**Scenario:** A Postpaid customer wants to make an international call to initiate a call back to his number 011911188998899. The rate to call prefix 01191\*\* is 20 cents per minute.

```
<CALLBACK>
  <INITIATE MSGID="781234A5" LEG_ONE_NO="01191188998899"
    ANI="2124541212" ACCT_TYPE="2" DURATION="50000">
  </INITIATE>
</CALLBACK>
```

**NOTE:** If the account is specified as Postpaid, only the fields specified above are used. All other fields, if specified, are ignored. Please note that DURATION is not necessary, but if it is specified, this will be the maximum duration allowed for this call. After the call is completed, Computer Telephony Server will send a Call Detail Record (CDR) to the billing server.

### Immediate Response - <CALLBACK> <RESPONSE>

Computer Telephony Server checks the data received and sends the message immediately to the applications program.

#### Attributes:

Attribute	Description
STATUS	SUCCESS, NUMBER_TOO LONG, FAILED, INVALID_BOX_NO, INVALID_LEG_ONE_NO

#### Example:

```
<CALLBACK>  
  <INITIATE_RESPONSE MSGID="781234A5" STATUS="SUCCESS">  
  </INITIATE_RESPONSE>  
</CALLBACK>
```

## Trunk Status Messages

Computer Telephony Server upon request from the applications software return the status of the trunk.

### Applications Trunk Status - <APP\_TRUNK\_STATUS><INFO>

The applications program sends this message to Computer Telephony Server to request the status of a port. After Computer Telephony Server receives this message, it sends the disconnect signal to the telephone network.

#### Attributes:

Attribute	Description
TRUNK_PORT	Port number Computer Telephony Server used to dial out the number.

#### Example:

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

### Response - <APP\_TRUNK\_STATUS><RESPONSE>

Computer Telephony Server sends this message to the applications program after determining the status of the port. This message is send in response to the <APP\_TRUNK\_STATUS><INFO> message.

#### Attributes:

Attribute	Description
STATUS	WAITING, PLAYING, RECORDING, DIALING, INVALID_TRUNKPORT

#### Example:

```
<APP_TRUNK_STATUS>
  <RESPONSE MSGID="786512M3" TRUNK_PORT="14" STATUS="PLAYING">
  </RESPONSE>
</APP_TRUNK_STATUS>
```

## Disconnect Messages

Computer Telephony Server handles the network generated and applications generated disconnects, in each case the signaling must be handled appropriately.

### Applications Disconnect - <APP\_TRUNK\_DISCONNECT><INFO>

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

#### Attributes:

Attribute	Description
TRUNK_PORT	Port number Computer Telephony 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>

Computer Telephony 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.

#### 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 Initialed Disconnect - <NETWORK\_TRUNK\_DISCONNECT><INFO>

Computer Telephony Server informs the application program when the Network sends a disconnect signal.

### 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>
```

## Applications Response - <NETWORK\_TRUNK\_DISCONNECT><RESPONSE>

### 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, Mute, Hold, and Patching

### Transfer an Agent - <AGENT\_TRANSFER><INFO>

The applications program sends this message to Computer Telephony Server and instructs it to transfer the call from one agent to the other agent.

#### Format:

```
<AGENT_TRANSFER>
  <INFO MSGID="Message ID" SOURCE_AGENT_PORT="Agent Number"
    DESTINATION_AGENT_PORT="Agent Number" ON_HOLD_FILE="On hold voice
    file" MODE="INITIATE, BLIND, ACCEPT, CANCEL">
  </INFO>
</AGENT_TRANSFER>
```

#### Attributes:

Attribute	Min	Max	Opt	Description
MESSAGEID	4	20	N	
SOURCE_AGENT_PORT	2	4	N	The agent port in active conversation with a customer.
DESTINATION_AGENT_PORT	2	4	N	The agent port to take over the call.
ON HOLD MESSAGE	10	64	Y	Message voice file to be played to the customer while the source agent talks to the destination agent.
MODE	5	8	N	Instruct the Computer Telephony Server what to do.

There are basically two steps to the operation.

The first step instructs the Computer Telephony Server to initiate the call transfer. In this case the call is transferred from the source agent to the destination agent for consultation purposes. The system plays the message on hold file to the customer port number.

#### Example:

```
<AGENT_TRANSFER>
  <INFO MSGID="GM-232" SOURCE_AGENT_PORT="34"
    DESTINATION_AGENT_PORT="38" ON_HOLD_FILE="Music.Vox"
    MODE="INITIATE">
  </INFO>
</AGENT_TRANSFER>
```

In the second step the applications program instructs the Computer Telephony Server to accept or cancel the call transfer. If the instruction is to accept, the system stops playing of the on hold message, and connects the destination agent port to the customer port.

**Example:**

```
<AGENT_TRANSFER>
  <INFO MSGID="GM-232" SOURCE_AGENT_PORT="34"
    DESTINATION_AGENT_PORT="38", MODE="ACCEPT">
  </INFO>
</AGENT_TRANSFER>
```

In case of cancel, the system stops playing the on hold message and reconnects the customer port to the source agent port.

**Example:**

```
<AGENT_TRANSFER>
  <INFO MSGID="GM-232" SOURCE_AGENT_PORT="34"
    DESTINATION_AGENT_PORT="38" ON_HOLD_FILE="" MODE="CANCEL">
  </INFO>
</AGENT_TRANSFER>
```

The case where the MODE is set to BLIND, the system simply transfers the call without the consultation feature. This is a one step process and no message on hold file is played to the customer port.

**Example:**

```
<AGENT_TRANSFER>
  <INFO MSGID="GM-232" SOURCE_AGENT_PORT="34"
    DESTINATION_AGENT_PORT="38" ON_HOLD_FILE="" MODE="BLIND">
  </INFO>
</AGENT_TRANSFER>
```

**Transfer Agent Response - <AGENT\_TRANSFER><RESPONSE>**

Computer Telephony Server sends this message to the applications program if there are errors or the transfer takes place. The following cases are supported:

**Case:**

```
<AGENT_TRANSFER>
  <RESPONSE MSGID="Message Id" SOURCE_AGENT_PORT="port number"
    DESTINATION_AGENT_PORT="port number" ON_HOLD_FILE="file name"
    STATUS="CONNECTED, BUSY, NOANS, INVALID_AGENT">
  </RESPONSE>
</AGENT_TRANSFER>
```

**Case:**

```
<AGENT_TRANSFER>
  <RESPONSE MSGID="Message Id" SOURCE_AGENT_PORT="port number"
    DESTINATION_AGENT_PORT="port number" ON_HOLD_FILE="file name"
    MODE="Mode entered" STATUS="INVALIDE_MODE" >
  </RESPONSE>
</AGENT_TRANSFER
```

**Case:**

```
<AGENT_TRANSFER>
  <RESPONSE MSGID="Message Id" SOURCE_AGENT_PORT="port number"
    DESTINATION_AGENT_PORT="port number" ON_HOLD_FILE="file name"
    MODE="Mode entered" STATUS="INVALID_SOURCE_AGENT_PORT" >
  </RESPONSE>
</AGENT_TRANSFER>
```

**Case:**

```
<AGENT_TRANSFER>
  <RESPONSE MSGID="Message Id" SOURCE_AGENT_PORT="port number"
    DESTINATION_AGENT_PORT="port number" ON_HOLD_FILE="file name"
    MODE="Mode entered" STATUS="INVALIDE_DESTINATION_AGENT_PORT" >
  </RESPONSE>
</AGENT_TRANSFER>
```

**Case:**

```
<AGENT_TRANSFER>
  <RESPONSE MSGID="Message Id" SOURCE_AGENT_PORT="port number"
    DESTINATION_AGENT_PORT="port number" ON_HOLD_FILE="file name"
    MODE="Mode entered" STATUS="SOURCE_AGENT_NOT_CONNECTED" >
  </RESPONSE>
</AGENT_TRANSFER>
```

**Case:**

```
<AGENT_TRANSFER>
  <RESPONSE MSGID="Message Id" SOURCE_AGENT_PORT="port number"
    DESTINATION_AGENT_PORT="port number" ON_HOLD_FILE="file name"
    MODE="Mode entered" STATUS="DESTINATION_AGENT_ALREADY_CONNECTED"
  >
  </RESPONSE>
</AGENT_TRANSFER>
```

**Case:**

```
<AGENT_TRANSFER>
  <RESPONSE MSGID="Message Id" SOURCE_AGENT_PORT="port number"
    DESTINATION_AGENT_PORT="port number" ON_HOLD_FILE="file name"
    MODE="Mode entered" STATUS="FILE_MISSING">
  </RESPONSE>
```



</AGENT\_TRANSFER>

DRAFT 33

Confidential Information  
Page 41 of 89

## Mute Function - <MUTE><INFO>

The applications program sends this message to Computer Telephony Server and instructs it to active the mute function. The customer does not hear the agent conversation, while the agent hears the customer conversation.

### Format:

```
<MUTE>
  <INFO MSGID="Message ID" AGENT_PORT="Agent Number" MODE="ON,
  OFF">
  </INFO>
</MUTE>
```

### Attributes:

Attribute	Min	Max	Opt	Description
MESSAGEID	4	20	N	
AGENT_PORT	2	4	N	The agent port in active conversation with a customer.
MODE	2	3	N	Instruct the Computer Telephony Server what to do.

### Example:

```
<MUTE>
  <INFO MSGID="GM-232" AGENT_PORT="34" MODE="ON">
  </INFO>
</AGENT_TRANSFER>
```

After this message is processed, the customer port does not hear the agent conversation. The agent hears the customer port conversation.

### Example:

```
<MUTE>
  <INFO MSGID="GM-232" AGENT_PORT="34" MODE="OFF">
  </INFO>
</AGENT_TRANSFER>
```

After this message is processed, the customer port and the agent each other's conversation.

## Mute Function Response - <MUTE><RESPONSE>

Computer Telephony Server sends this message to the applications program if there are errors or the action takes place. The following cases are supported:

### Case:

```
<MUTE>
  <RESPONSE MSGID="Message Id" AGENT_PORT="port number" MODE="ON,
  OFF" STATUS="SUCCESS">
  </RESPONSE>
</MUTE>
```

### Case:

```
<MUTE>
  <RESPONSE MSGID="Message Id" AGENT_PORT="port number" MODE="mode"
  STATUS="INVALID_MODE">
  </RESPONSE>
</MUTE>
```

### Case:

```
<MUTE>
  <RESPONSE MSGID="Message Id" AGENT_PORT="port number" MODE="Mode
  entered" STATUS="INVALID_AGENT_PORT" >
  </RESPONSE>
</MUTE>
```

### Case:

```
<MUTE>
  <RESPONSE MSGID="Message Id" AGENT_PORT="port number" MODE="Mode
  entered" STATUS="AGENT_NOT_CONNECTED" >
  </RESPONSE>
</MUTE>
```

## Hold Customer Function - <HOLD><INFO>

The applications program sends this message to Computer Telephony Server to keep the customer on hold and play a message file. Neither the customer nor the agent hears each other's conversation.

### Format:

```
<HOLD>  
  <INFO MSGID="Message ID" AGENT_PORT="Agent Number"  
    ON_HOLD_FILE="On hold voice file" MODE="ON, CANCEL" >  
  </INFO>  
</HOLD>
```

### Attributes:

Attribute	Min	Max	Opt	Description
MESSAGEID	4	20	N	
SOURCE_AGENT_PORT	2	4	N	The agent port in active conversation with a customer.
ON HOLD MESSAGE	10	64	N	Message voice file to be played to the customer port while the agent talks to some one else.
MODE	2	5	N	Instruct the Computer Telephony Server what to do.

### Example:

```
<HOLD>  
  <INFO MSGID="GM-232" AGENT_PORT="34" ON_HOLD_FILE="Music.Vox"  
    MODE="ON" >  
  </INFO>  
</HOLD>
```

After this message is processed, the customer port hears the Music.Vox file, and neither the agent nor the customer port hear each other.

### Example:

```
<HOLD>  
  <INFO MSGID="GM-232" AGENT_PORT="34" ON_HOLD_FILE=""  
    MODE="CANCEL" >  
  </INFO>  
</HOLD>
```

After this message is processed, the system stops the play and reconnects the customer port to the agent. The agent and the customer ports are place in a full duplex conversation.

## Hold Customer Response - <HOLD><RESPONSE>

Computer Telephony Server sends this message to the applications program if there are errors or the function has been performed successful. The following cases are supported:

### Case:

```
<HOLD>
  <RESPONSE MSGID="Message Id" AGENT_PORT="port number"
    ON_HOLD_FILE="file name" MODE="Mode entered" STATUS="SUCESS">
  </RESPONSE>
</HOLD>
```

### Case:

```
<HOLD>
  <RESPONSE MSGID="Message Id" AGENT_PORT="port number"
    ON_HOLD_FILE="file name" MODE="Mode entered"
    STATUS="INVALID_MODE" >
  </RESPONSE>
</HOLD>
```

### Case:

```
<HOLD>
  <RESPONSE MSGID="Message Id" AGENT_PORT="port number"
    ON_HOLD_FILE="file name" MODE="Mode entered"
    STATUS="INVALID_AGENT_PORT" >
  </RESPONSE>
</HOLD>
```

### Case:

```
<HOLD>
  <RESPONSE MSGID="Message Id" AGENT_PORT="port number"
    ON_HOLD_FILE="file name" MODE="Mode entered"
    STATUS="AGENT_NOT_CONNECTED" >
  </RESPONSE>
</HOLD>
```

### Case:

```
<HOLD>
  <RESPONSE MSGID="Message Id" AGENT_PORT="port number"
    ON_HOLD_FILE="file name" MODE="Mode entered"
    STATUS="FILE_MISSING">
  </RESPONSE>
</HOLD>
```

## Dial and Patch - <DIAL\_PATCH><INFO>

The applications program sends this message to Computer Telephony 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 agent port specified in the message. If the called party disconnects, the Server breaks the connection between the called party and the agent port.

### Format:

```
<DIAL_PATCH>  
  <INFO MSGID="Message ID" AGENT_PORT="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
AGENT_PORT	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" AGENT_PORT="27" DNIS="17322901900"  
  ANI="2126793400">  
  </INFO>  
</DIAL_PATCH>
```

In the above example, Computer Telephony 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, Computer Telephony Server patches the agent port 27 to the called party's trunk port.

## Dial and Patch Response - <DIAL\_PATCH><RESPONSE>

Computer Telephony 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="AGENT_PORT_ALREADY_CONNECTED, INVALID_AGENT_PORT,  
NO_DIALOUT_PORT_FREE, SUCCESS, FAILED">  
</RESPONSE>  
</DIAL_PATCH>
```

### **Dial and Patch Done - <DIAL\_PATCH><TERMINATION>**

Computer Telephony 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, AGENT_BUSY, FAILED_TO_GET_DIALOUT_PORT"  
TRINK_PORT="Port Number Dialed On">  
  </TERMINATION>  
</DIAL_PATCH>
```

## Patch Ports - <PATCH\_PORTS><INFO>

The applications program sends this message to Computer Telephony 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>

Computer Telephony 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>**

Computer Telephony Server sends this message to the applications program after it has successfully patched the ports specified in the <PATCH\_PORTS><INFO> message.

Computer Telephony 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, Computer Telephony 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 Computer Telephony 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, Computer Telephony Server tears down the connection between trunk port 6 and trunk port 23. The trunk ports are not connected to each other, Computer Telephony Server sends a error message to the applications program.

## Tear Down Patch Response - <TEAR\_PATCH><RESPONSE>

Computer Telephony 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>**

Computer Telephony 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, Computer Telephony Server successfully tore down the connection between trunk port 6 and trunk port 23.

## **Initiate Conference Setup**

### **Conference Call Setup - <INIT\_CONFERENCE><INFO>**

The applications program instructs Computer Telephony Server to establish a conference call. The applications program must assign a unique conference ID to the conference.

Format:

```
<INIT_CONFERENCE>
  <INFO MSGID="Message ID" CONFERENCE_ID="Number"
    AGENT_NO="Agent_Number" TRUNK_PORT="Phone Port Number"
    MODE="FULL, LISTEN, REMOVE">
  </INFO>
</INIT_CONFERENCE>
```

### **Response Conference Call Setup - <INIT\_CONFERENCE><RESPONSE>**

Computer Telephony Server sends this message to the applications program after it has setup the conference.

Format:

```
<INIT_CONFERENCE>
  <RESPONSE MSGID="Message Id" STATUS="CONF_CREATED, FULL, ADDED">
  </RESPONSE>
</INIT_CONFERENCE>
```

## Conference Link Setup - <INIT\_CONFERENCE\_LINK><INFO>

The applications program instructs Computer Telephony Server to establish a conference service link. The applications program must assign a unique conference ID to the conference.

### Format:

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

### Attributes:

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

### Example 1:

```
<INIT_CONFERENCE_LINK>  
  <INFO MSGID="PAN-123" NAME="GERMAN-101" DNIS="7322901900"  
    PASS_CODE="2424">  
  </INFO>  
</INIT_CONFERENCE_LINK>
```

In the above example, Computer Telephony Server receives the message from the applications program and sets the mailbox 7322901900 to be notified for conference call activities. The pass code 2424 is the pass code for the mailbox. The mailbox name is set to GERMAN-101. When a caller dials 7322901900, Computer Telephony Server sends the <INBOUND\_CONFERENCE> message to the applications program.

In order for Computer Telephony 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\_CONFERENCE\_LINK> message per DNIS. The applications program must have the correct pass codes for the mailboxes.

## Response Conference Link Setup - <INIT\_CONFERENCE\_LINK><RESPONSE>

Computer Telephony Server sends this message to the applications program after it has established the service link.

### Format:

```
<INIT_CONFERENCE_LINK>  
  <RESPONSE MSGID="Message Id" DNIS="Mail Box No."  
    PASS_CODE="Pass word" NAME="Conference Name"  
    STATUS=" CONFERENCE_CALLS_NO_SUPPORTED, INVALID_PASS_CODE,  
    INVALID_DNIS, SUCCESS">  
  </RESPONSE>  
</INIT_CONFERENCE_LINK>
```

### Example:

```
<INIT_CONFERENCE_LINK>  
  <RESPONSE MSGID=" PAN-123" DNIS="7322901900"  
    NAME="GERMAN-101" PASS_CODE="2222" STATUS="SUCCESS">  
  </RESPONSE>  
</INIT_CONFERENCE_LINK>
```

## Receive Call Conference

### Inbound Conference Call - <INBOUND\_CONFERENCE><INFO>

When Computer Telephony 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 is set up with the <INIT\_LINK> message. Generally this is a C type mailbox with the name containing the socket id of the applications program link.

If the conference is not established, then the Computer Telephony Server establishes the conference. If there is only one conferee, then it plays a message to the caller informing him that he is the only conferee in the conference group.

The following message is sent to the applications program:

#### Format:

```
<INBOUND_CONFERENCE>
  <INFO MSGID="Message ID" DNIS="Phone Number" ANI="Phone Number"
  REDIRECTING_NO="Phone Number"
  BOXNO="Mail Box Number" TRUNK_PORT="Port Number"
  CONFERENCE_ID="Number" TOTAL_CONFEREES="total number of
  conferees">
  </INFO>
</INBOUND_CONFERENCE >
```

#### Example:

```
<INBOUND_CONFERENCE>
  <INFO MSGID="5634" DNIS="7325668767" ANI="405674345"
  REDIRECTING_NO="" BOXNO="6325668767" TRUNK_PORT="5"
  CONFERENCE_ID="4" TOTAL_CONFEREES="3">
  </INFO>
</INBOUND_CONFERENCE >
```

### Response Conference Call Setup - <INBOUND\_CONFERENCE><RESPONSE>

The application program responds with the following message:

#### Format:

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

## Change Conferee Mode

### Change Mode of the Conferee - <CHANGE\_CONFEREE\_MODE><INFO>

The applications program sends this message to the Computer Telephony Server to change the mode of the conferee.

#### Format:

```
<CHANGE_CONFEREE_MODE>
  <INFO MSGID="Message ID" TRUNK_PORT="Port Number"
    CONFERENCE_ID="Number" BOXNO="Mail Box No" MODE="MONITOR, FULL,
    REMOVE, ADD">
  </INFO>
</CHANGE_CONFEREE_MODE>
```

#### Example 1 – Removing a Conferee from a Conference Group:

```
<CHANGE_CONFEREE_MODE>
  <INFO MSGID="5634" TRUNK_PORT="5" CONFERENCE_ID="4"
    BOXNO="7325667100" MODE="REMOVE">
  </INFO>
</CHANGE_CONFEREE_MODE>
```

In the above example, the Computer Telephony Server removes the caller on port 5 from the conference group associated with the mailbox 7325667100, but does not disconnect him.

#### Example 2 – Adding a Conferee to a Conference Group:

```
<CHANGE_CONFEREE_MODE>
  <INFO MSGID="5634" TRUNK_PORT="5" CONFERENCE_ID="4"
    BOXNO="7325667811" MODE="ADD">
  </INFO>
</CHANGE_CONFEREE_MODE>
```

In the above example, the Computer Telephony Server adds the caller on port 5 to the Conference Group associated with the mailbox 7325667811. If the Conference Group 5 is full, the CONFERENCE\_FULL status is returned.

#### Example 3 – Making a Conferee Mute:

```
<CHANGE_CONFEREE_MODE>
  <INFO MSGID="5634" TRUNK_PORT="5" CONFERENCE_ID="4"
    BOXNO="7325667811" MODE="MONITOR">
  </INFO>
</CHANGE_CONFEREE_MODE>
```

In the above example, the Computer Telephony Server allows the caller on port 5 to only listen to other conferees.



### Example 3 – Allowing the Conferee Talk and Listen:

```
<CHANGE_CONFEREE_MODE>
  <INFO MSGID="5634" TRUNK_PORT="5" CONFERENCE_ID="4"
    BOXNO="7325667811" MODE="FULL">
  </INFO>
</CHANGE_CONFEREE_MODE>
```

In the above example, the Computer Telephony Server allows the caller on port 5 to talk and listen to other conferees.

### Response - <CHANGE\_CONFEREE\_MODE><RESPONSE>

The Computer Telephony Server responds with the following message:

#### Format:

```
<CHANGE_CONFEREE_MODE>
  <RESPONSE MSGID="Message Id" CONFERENCE_ID="Conference ID"
    BOXNO="Mailbox Number", TRUNK_PORT="Line Number" STATUS="SUCCESS,
    INVALID_TRUNK_PORT, INVALID_CONFERENCE_ID, INVALID_MAIL_BOX_NO,
    INVALID_MOD, FAILURE, CONFERENCE_FULL">
  </RESPONSE>
</CHANGE_CONFEREE_MODE>
```

## Active Talkers

### Report the List of Active Talkers - <ACTIVE\_TALKERS\_LIST><INFO>

The applications program sends this message to the Computer Telephony Server to receive the list of active talkers in a conference group. The application program should send this message every 100 to 200 milliseconds to get the status of the active talkers.

#### Format:

```
<ACTIVE_TALKERS_LIST>
  <INFO MSGID="Message ID" CONFERENCE_ID="Number" BOXNO="Mailbox">
  </INFO>
</ACTIVE_TALKERS_LIST>
```

#### Example:

```
<ACTIVE_TALKERS_LIST>
  <INFO MSGID="5634" CONFERENCE_ID="4" BOXNO="7325667100">
  </INFO>
</ACTIVE_TALKERS_LIST>
```

In the above example, the applications program requests the Computer Telephony Server to report the status of active of talkers in the conference group associated with the mailbox 7325667100.

### Response - <ACTIVE\_TALKER\_LIST><RESPONSE>

The Computer Telephony Server sends the list of active talkers in a conference group by sending the following message:

#### Format:

```
<ACTIVE_TALKERS_LIST>
  <RESPONSE MSGID="Message Id"
  BOXNO="Mailbox" TOTAL_TALKERS="Number of Conferees Talking"
  LIST="ports" STATUS="SUCCESS, FAILURE, INVALID_MAIL_BOX_NO" >
  </RESPONSE>
</ACTIVE_TALKERS_LIST>
```

#### Example:

```
<ACTIVE_TALKERS_LIST>
  <RESPONSE MSGID="78654" BOXNO="7325667100" TOTAL_TALKERS="2"
  LIST="3 21" STATUS="SUCCESS" >
  </RESPONSE>
</ACTIVE_TALKERS_LIST>
```

In the above example, the Computer Telephony Server informs the applications program that the callers on trunk port numbers 3 and 21 were the active talkers. There were two active talkers.

## Agent Login and Logout

### Agent Log In - <AGENT\_LOGIN><INFO>

The applications program must send this message to Computer Telephony Server when an Agent logs in.

Format:

```
<AGENT_LOGIN>
  <INFO MSGID="Message ID" AGENT_NO="Agent Number"
    AGENT_NAME="Agent Name" CAMP_ID="Campaign Id" MODE="MANUAL,
    INBOUND, PREDICTIVE, BLENDED, BUSY">
  </Info>
</AGENT_LOGIN>
```

Attribute	Min	Max	Opt	Description
MSGID	4	20	N	Message ID
AGENT_NO	2	4	N	Valid agent number who handles calls
AGENT_NAME	2	30	Y	Agent name for tracking purposes
CAMP_ID	2	10	Y	Campaign ID used for file names
MODE	4	10	N	Agent function modes MANUAL – Do not route predictive or inbound calls INBOUND – Route only inbound calls PREDICTIVE – Route dial out calls BLENDED – Route inbound and dial out calls BUSY – Do not route any calls

### Example:

```
<AGENT_LOGIN>
  <INFO MSGID="786512" AGENT_NO="6" AGENT_NAME="John" CAMP_ID="IBM"
    MODE="PREDICTIVE">
  </INFO>
</AGENT_LOGIN>
```

### Response Agent Log In - <AGENT\_LOGIN><RESPONSE>

Computer Telephony Server sends this message in response to the applications' <AGENT\_LOGIN><INFO> message.

Format:

```
<AGENT_LOGIN>
  <RESPONSE MSGID="Message Id" STATUS="SUCCESS, ALREADY_LOGGED_IN,
    INVALID_AGENT_NO">
  </RESPONSE>
</AGENT_LOGIN>
```

## Agent Log Out - <AGENT\_LOGOUT><INFO>

The applications program must send this message to Computer Telephony Server when an Agent logs out.

Format:

```
<AGENT_LOGOUT>
  <INFO MSGID="Message ID" AGENT_NO="Agent Number"
    CAMP_ID="Campaign Id">
  </Info>
</AGENT_LOGOUT>
```

### Example:

```
<AGENT_LOGOUT>
  <INFO MSGID="786512" AGENT_NO="7" CAMP_ID="IBM"
  </INFO>
</AGENT_LOGOUT>
```

## Response Agent Log Out - <AGENT\_LOGOUT><RESPONSE>

Computer Telephony Server sends this message in response to the applications' <AGENT\_LOGOUT><INFO> message.

Format:

```
<AGENT_LOGOUT>
  <RESPONSE MSGID="Message Id" STATUS="SUCCESS, ALREADY_LOGGED_OUT,
    INVALID_AGENT_NO">
  </RESPONSE>
</AGENT_LOGOUT>
```

**Note:**

Make sure the .\Agent\_Status directory is not created.

## Barging In (Optional Feature)

### Barge In Message - <BARGE\_IN><INFO>

The applications program sends this message to Computer Telephony Server and instructs it to let a supervisor port barge into an established connection between an agent and a customer.

The customer port is known to the system through previous actions, therefore it is not necessary to specify the customer port number.

The feature require the use conferencing hardware that must support the coach and pupil features.

#### Format:

```
<BARGE_IN>
  <INFO MSGID="Message ID" SUPERVISOR_PORT="PortNo"
    AGENT_PORT="Agent Port Number" MODE="COACH, MONITOR, FULL">
  </Info>
</BARGE_IN>
```

#### Example:

```
<BARGE_IN>
  <INFO MSGID="GE_13451" SUPERVISOR_PORT="27" AGENT_PORT="23"
    MODE="FULL">
  </INFO>
</BARGE_IN>
```

The agent on Port 23 must be connected to a customer port. After this XML message, the system creates a conference group with ports 23, 27, and the customer port number as members.

### Response Barge In - <BARGE\_IN><RESPONSE>

Computer Telephony Server sends this message to the applications program after it adds the trunk port or agent port to an already established connection.

#### Format:

```
<BARGE_IN>
  <RESPONSE MSGID="Message Id" STATUS="SUCCESS, FAILED">
  </RESPONSE>
</BARGE_IN>
```

## Error Messages:

### Case 1:

```
<BARGE_IN>  
  <RESPONSE MSGID="Message ID" AGENT_PORT="Port Number"  
    STATUS="AGENT_NOT_CONNECTED, INVALID_AGENT_PORT ">  
  </RESPONSE>  
</BARGE_IN>"
```

### Case 2:

```
<BARGE_IN>  
  <RESPONSE MSGID="Message Id" MODE="Mode" STATUS="INVALID_MODE">  
  </RESPONSE>  
</BARGE_IN>"
```

### Case 3:

```
<BARGE_IN>  
  <RESPONSE MSGID="Message Id" SUPERVISOR_PORT="Port Number"  
    STATUS="INVALID_SUPERVISOR_PORT">  
  </RESPONSE>  
</BARGE_IN>"
```

## Barging Out (Optional Feature)

### Barge Out Message - <BARGE\_OUT><INFO>

The applications program sends this message to Computer Telephony Server and instructs it to break a previously established barge in case.

#### Format:

```
<BARGE_OUT>
  <INFO MSGID="Message ID" SUPERVISOR_PORT="PortNo"
    AGENT_PORT="Agent Port Number"">
  </Info>
</BARGE_OUT>
```

#### Example:

```
<BARGE_OUT>
  <INFO MSGID="GE_13451" SUPERVISOR_PORT="27" AGENT_PORT="23"
  </INFO>
</BARGE_OUT>
```

After this XML message, the system removes the ports 23, 27, and the customer ports from the barge in conference group.

### Response Barge Out - <BARGE\_OUT><RESPONSE>

Computer Telephony Server sends this message to the applications program after the barge conference has been broken. The server joins the agent and customer ports.

#### Format:

```
<BARGE_OUT>
  <RESPONSE MSGID="Message Id" STATUS="SUCCESS, FAILED">
  </RESPONSE>
</BARGE_OUT>
```

#### Error Messages:

##### Case 1:

```
<BARGE_OUT>
  <RESPONSE MSGID="Message ID" AGENT_PORT="Port Number"
    STATUS="AGENT_NOT_CONNECTED, INVALID_AGENT_PORT ">
  </RESPONSE>
</BARGE_OUT>
```

**Case 2:**

```
<BARGE_OUT>  
  <RESPONSE MSGID="Message Id" SUPERVISOR_PORT="Port Number"  
    STATUS="INVALID_SUPERVISOR_PORT">  
  </RESPONSE>  
</BARGE_OUT>"
```

DRAFT 33



## Inbound Call Control and Switching Messages

### Inbound Call

#### Receive Call - <RECEIVE\_CALL><INFO>

When Computer Telephony 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 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 Computer Telephony 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>
```

## Handle Inbound Call - <HANDLE\_INBOUND\_CALL><INFO>

The applications program instructs the Computer Telephony Server to handle and inbound call from the Network. The application program passes the DNIS number corresponding to a mailbox. In addition to the DNIS, the application program must send the ANI, and Redirecting Number. The application program is responsible for sending the disconnect message when the caller disconnects. For other terminating conditions, the Computer Telephone Server sends the proper message to the applications program.

### Format:

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

## Response Handle Inbound - <HANDLE\_INBOUND\_CALL><RESPONSE>

The Computer Telephony Server sends this response to the applications program.

### Format:

```
<HANDLE_INBOUND_CALL>  
  <RESPONSE MSGID="Message Id" STATUS="SUCCESS, INVALID_DNIS,  
    INVALID_MAILBOX, DISABLED_MAILBOX"  
  </RESPONSE>  
</HANDLE_INBOUND_CALL>
```

## Accept the Call - <ACCEPT\_CALL><INFO>

The applications program instructs the Computer Telephony Server to accept the call inbound call from the Network.

### Format:

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

## Accept Call Response - <ACCEPT\_CALL><RESPONSE>

Computer Telephony Server sends this message immediately to the applications program that the trunk port is correct.

### Format:

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

## Termination Status Accept Call - <ACCEPT\_CALL><TERMINATION>

Computer Telephony Server sends this message to the applications program after it has completed the accept call process.

### Format:

```
<ACCEPT_CALL>
  <TERMINATION MSGID="Message Id" TRUNK_PORT="Port Number"
  EVENT="EOF">
  </TERMINATION>
</ACCEPT_CALL>
```

## Decline the Call - <DECLINE\_CALL><INFO>

The applications program instructs the Computer Telephony Server to decline and incoming call from the Network.

### Format:

```
<DECLINE>
  <INFO MSGID="Message ID" TRUNK_PORT="Port Number"
    CAUSE_CODE="NORMAL_CLEARING"
  </INFO>
</DECLINE >
```

## Decline Call Response - <DECLINE\_CALL><RESPONSE>

Computer Telephony Server sends this message immediately to the applications program that the trunk port is correct and also there is a cause code.

### Format:

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

## Termination Status Decline Call - <DECLINE\_CALL><TERMINATION>

Computer Telephony Server sends this message to the applications program after it has completed the decline call process.

### Format:

```
<DECLINE_CALL>
  <TERMINATION MSGID="Message Id" TRUNK_PORT="Port Number"
    EVENT="EOF">
  </TERMINATION>
</DECLINE_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 Computer Telephony 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" FILE3="file name" FILE4="file name"
  ANUMBER="Phone Number" BNUMBER="Phone Number"
  MODE="INTERRUPTABLE, UNINTERRUPTABLE, UNINTCOLLECTDIG">
</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
FILE3	3	40	Y	Optional third message file path
FILE4	3	40	Y	Optional Fourth message file path
ANUMBER	3	15	Y	Used for UNITCOLLECTDIG file name
BNUMBER	3	15	Y	Used for UNITCOLLECTDIG file name
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>
```

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

#### Response Play File - <PLAY\_VOICE\_FILE><RESPONSE>

Computer Telephony 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, FILE3_MISSING , FILE4_MISSING, VOICE_CHANNEL_BUSY,
    INVALID_CHANNEL_STATE">
  </RESPONSE>
</PLAY_VOICE_FILE>
```

**Termination Status Play File - <PLAY\_VOICE\_FILE><TERMINATION>**

Computer Telephony 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, STOPPED">
  </TERMINATION>
</PLAY_VOICE_FILE>
```

**Example 1:**

```
<PLAY_VOICE_FILE>
  < TERMINATION MSGID="786123" TRUNK_PORT="15" EVENT="EOF
    EVENT="3">
  </ TERMINATION >
</PLAY_VOICE_FILE>
```

In the example 1, the listener pressed the DTMF key 3 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 Computer Telephony 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>**

Computer Telephony 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>**

Computer Telephony Server sends this message to the applications program after it has successfully stopped 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>
```

**Example bpl:**

```
<PLAY_VOICE_FILE>  
<INFO MSGID="7865343" ANUMBER="7325367811" BNUMBER="7322901900"  
TRUNK_PORT="34" FILE="C:\PEC\SARI_RAAT.VOX" MODE="UNINTCOLLECTDIG"  
</INFO>  
</PLAY_VOICE_FILE>
```

**Output could be:**

File name:

C:\Pec\Dayanand\PLAY\_RESP\_0206200813.Txt

Data:

MSGID="7865343" ANUMBER="7325367811" BNUMBER="7322901900" DTMF="343"

Shows that 343 was pressed by the caller during the playing of the file:

C:\PEC\SARI\_RAAT.VOX in uninterruptable mode.

DRAFT 33



## **Play Number**

The applications program sends this message to Computer Telephony Server to initiate a voice number play.

### **Play Number - <PLAY\_NUMBER><INFO>**

The applications program sends this message to Computer Telephony 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>**

Computer Telephony 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>**

Computer Telephony 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 Computer Telephony 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>**

Computer Telephony Server sends this message immediately to the applications program after verifying the data 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>**

Computer Telephony 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 Computer Telephony Server to stop a recording session in progress.

Format:

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

The optional OPTION="STAS" tells the software to stop the play synchronously, generate an ENOFFILE and send the RESPONSE message only after the stop has been completed.

## Response Stop Recording - <STOP\_RECORD\_VOICE\_FILE><RESPONSE>

Computer Telephony 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>

Computer Telephony 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 Computer Telephony 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>

Computer Telephony 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>

Computer Telephony 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 Computer Telephony 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>

Computer Telephony 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>

Computer Telephony 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>
```

## Starting and Stopping of Two Way Voice Recordings

### Record Two Way Voice - <RECORD\_TWO\_WAY\_VOICE><INFO>

The applications program instructs Computer Telephony Server to initiate a two way voice record session. The two way record receives voice from the transmit channels of two trunk ports and after adding records it in a file. If there is no activity for 600 seconds the recording session stops.

#### Format:

```
<RECORD_TWO_WAY_VOICE>
  <INFO MSGID="Message ID" TRUNK_PORT1="Port Number 1",
    TRUNK_PORT2="Port Number 2" FILE_NAME="file name"
  </INFO>
</RECORD_TWO_WAY_VOICE>
```

#### Attributes:

Attribute	Min	Max	Opt	Description
MSGID	4	20	N	Message ID
TRUNK_PORT1	2	4	N	1 <sup>st</sup> Trunk port to listen to
TRUNK_PORT2	2	4	N	2nd Trunk port to listen to
FILE_NAME	3	40	N	The file name for recordings

#### Example:

```
<RECORD_TWO_WAY_VOICE>
  <INFO MSGID="674312" TRUNK_PORT1="08", TRUNK_PORT2="21"
    FILE_NAME="C:\TWR\JOHN_SMITH.VOX">
  </INFO>
</RECORD_TWO_WAY_VOICE>
```

In the above example, Computer Telephony Server will listen to trunk ports 8 and 21 and record the two way conversation in the file c:\twr\john\_smith.vox. The recording will stop once Computer Telephony Server receives the STOP\_VOICE\_TWO\_WAY\_RECORD message.

### Two Way Record Response - <RECORD\_TWO\_WAY\_VOICE><RESPONSE>

Computer Telephony Server sends this message immediately to the applications program after checking the validity of the data.

#### Format:

```
<RECORD_TWO_WAY_VOICE>
  <RESPONSE MSGID="Message Id" TRUNK_PORT1="Port Number 1",
    TRUNK_PORT2="Port Number 2" STATUS="INVALID_TRUNK_PORT1,
    INVALID_TRUNK_PORT2, TRUNK_PORTS_THE_SAME, SUCCESS">
```

```
</RESPONSE>
</RECORD_TWO_WAY_VOICE>
```

### **Two Way Record Done - <RECORDED\_TWO\_WAY\_VOICE><TERMINATION>**

Computer Telephony Server sends this message to the applications program after the recording for the two ports have been completed.

#### **Format:**

```
<RECORD_TWO_WAY_VOICE>
  <TERMINATION MSGID="Message Id" TRUNK_PORT1="Port Number 1",
    TRUNK_PORT2="Port Number 2" EVENT="FAILD_FILE_CREATE, EOF">
  </TERMINATION>
</RECORD_TWO_WAY_VOICE>
```

#### **Example:**

```
<RECORD_TWO_WAY_VOICE>
  <TERMINATION MSGID="674312" TRUNK_PORT1="08", TRUNK_PORT2="21"
    EVENT="EOF">
  </TERMINATION>
</RECORD_TWO_WAY_VOICE>
```

In the above example, Computer Telephony Server informs the applications program that the two way recording for the trunk port has been stopped.

## Stop Two Way Record - <STOP\_RECORD\_TWO\_WAY\_VOICE><INFO>

The applications program instructs Computer Telephony Server to stop the two recording in progress.

### Format:

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

### Attributes:

Attribute	Min	Max	Opt	Description
MSGID	4	20	N	Message ID
TRUNK_PORT	2	4	N	1 <sup>st</sup> Trunk port to listen to

### Example:

```
<STOP_RECORD_TWO_WAY_VOICE>
  <INFO MSGID="674312" TRUNK_PORT="09">
  </INFO>
</STOP_RECORD_TWO_WAY_VOICE>
```

In the above example, Computer Telephony Server will stop the two way recording of the conversation between on trunk ports 9.

## Stop Two Way Record Response – <STOP\_RECORD\_TWO\_WAY\_VOICE><RESPONSE>

Computer Telephony Server sends this message immediately to the applications program after checking the validity of the data.

### Format:

```
<STOP_RECORD_TWO_WAY_VOICE>
  <RESPONSE MSGID="Message Id" TRUNK_PORT1="Port Number 1",
  STATUS="INVALID_TRUNK_PORT">
  </RESPONSE>
</STOP_RECORD_TWO_WAY_VOICE>
```



## Stop Two Way Record Done - < STOP\_RECORD\_TWO\_WAY\_VOICE> <TERMINATION>

Computer Telephony Server sends this message to the applications program after the recording session is complete.

### Format:

```
<STOP_RECORD_TWO_WAY_VOICE>  
  <TERMINATION MSGID="Message Id" TRUNK_PORT="Port Number 1",  
    EVENT="SUCCESS">  
  </TERMINATION>  
</STOP_RECORD_TWO_WAY_VOICE>
```

### Example:

```
<STOP_RECORD_TWO_WAY_VOICE>  
  <TERMINATION MSGID="674312" TRUNK_PORT1="08", EVENT="SUCCESS">  
  </TERMINATION>  
</STOP_RECORD_TWO_WAY_VOICE>
```

In the above example, Computer Telephony Server informs the applications program that the stop function was performed successfully.

## Attach a Trunk to a Mailbox or Execute and IVR

### Message for a Mailbox - <ATTACH\_MIALBOX><INFO>

The applications program instructs Computer Telephony Server to attach the port to a mailbox and take a message or allow the subscriber to pick up his messages. Please make sure the mailbox number is a valid mailbox.

#### Format:

```
<ATTACH_MAILBOX>  
  <INFO MSGID="Message ID" BOXNO="Mail Box No" BOXPWD="Mail Box  
  Pass Code" TRUNK_PORT="Port Number"  
  </INFO>  
</ATTACH_MAILBOX>
```

#### Attributes:

Attribute	Min	Max	Opt	Description
MSGID	4	20	N	Message ID
BOXNO	7	10	N	Mail box number
BOXPWD	4	4	Y	Mail box pass code used for picking up messages.
TRUNK_PORT	2	4	N	Trunk port to attach to

#### Example 1:

```
<ATTACH_MAILBOX>  
  <INFO MSGID="Message ID" BOXNO="2123404600" TRUNK_PORT="41"  
  </INFO>  
</ATTACH_MAILBOX>
```

This allows the connected party to leave a message for the mailbox number 2123404600. Please make sure that 2123404600 is a valid U-type mailbox.

#### Example 1:

```
<ATTACH_MAILBOX>  
  <INFO MSGID="Message ID" BOXNO="2123404600" BOXPWD="2222"  
  TRUNK_PORT="41"  
  </INFO>  
</ATTACH_MAILBOX>
```

This allows the connected party to enter the pass code and read the messages from his mail box.

## Response - <ATTACH\_MIALBOX><RESPONSE>

Computer Telephony Server sends this message immediately to the applications program after checking the validity of the data.

### Format:

```
<ATTACH_MAILBOX>  
  <RESPONSE MSGID="Message ID" BOXNO="Mail Box No"  
    TRUNK_PORT="Port Number" BOXPWD="Mail Box Pass Code"  
    STATUS="SUCCESS, INVALID_BOX_NO, INVALID_TRUNK_PORT,  
    INVALID_PASS_CODE, TRUNK_PORT_IS_PATCHED, CALL_NOT_CONNECTED,  
    UNAVAILABLE_VOICE_RESOURCE"  
  </RESPONSE>  
</ATTACH_MAILBOX>
```

DRAFT 33

## Execute IVR Mailbox Call Flow - <EXEC\_IVR\_MAILBOX><INFO>

The applications program instructs Computer Telephony Server to attach the port to an Interactive Voice Response (IVR) mailbox and perform functions based on the IVR logic created the Applications Generator. Please make sure the mailbox number is a valid mailbox. (Please see Appendix A)

### Format:

```
<EXEC_IVR_MAILBOX>  
  <INFO MSGID="Message ID" BOXNO="Mailbox number" TRUNK_PORT="Port  
  Number" VAR1="Variable1 value" VAR2="Variable2 value"  
  VAR3="Variable3 value" VAR4="Variable4 value" VAR5="Variable5  
  value">  
  </INFO>  
</EXEC_IVR_MAILBOX>
```

### Attributes:

Attribute	Min	Max	Opt	Description
MSGID	4	20	N	Message ID
BOXNO	1	12	N	Box Number of the IVR Call flow to be executed
TRUNK_PORT	2	4	N	Trunk port to play the message file
VAR1	1	40	Y	Variable name used in IVR Call flow
VAR2	1	40	Y	Variable name used in IVR Call flow
VAR3	1	40	Y	Variable name used in IVR Call flow
VAR4	1	40	Y	Variable name used in IVR Call flow
VAR5	1	40	Y	Variable name used in IVR Call flow

### Example 1:

```
<EXEC_IVR_MAILBOX>  
<INFO MSGID="786123" BOXNO="7325667661" TRUNK_PORT="1" VAR1="5555"  
VAR2="4444" VAR3="8888" VAR4="1111" VAR5="2222" >  
</INFO>  
</EXEC_IVR_MAILBOX>
```

Computer Telephony Server will execute the IVR Call flow. The values specified in the execute command will be available within the call flow. After call flow execution it returns the execution status and variables.

## Response - <EXEC\_IVR\_MAILBOX><RESPONSE>

Computer Telephony Server sends this message immediately to the applications program after it has determined that it will be able to execute the IVR call flow specified or it is unable to proceed.

### Format:

```
<EXEC_IVR_MAILBOX>
  <INFO MSGID="Message ID" BOXNO="Mailbox number" TRUNK_PORT="Port
Number" VAR1="Variable1 value" VAR2="Variable2 value"
VAR3="Variable3 value" VAR4="Variable4 value" VAR5="Variable5
value"> STATUS="SUCCESS, INVALID_TRUNK_PORT, INVALID_MAILBOX,
VOICE_CHANNEL_BUSY, INVALID_CHANNEL_STATE">
  </INFO>
</EXEC_IVR_MAILBOX>
```

## Termination Status - <EXEC\_IVR\_MAILBOX><TERMINATION>

Computer Telephony Server sends this message to the applications program after it has Executed the IVR Call flow, or if the flow execution stopped by the application

### Format:

```
<EXEC_IVR_MAILBOX>
  <TERMINATION MSGID="Message ID" BOXNO="Mailbox number"
TRUNK_PORT=" Port Number" VAR1="Variable1 value" VAR2="Variable2
value" VAR3="Variable3 value"
VAR4="Variable4 value" VAR5="Variable5 value" EVENT="EOCS, EOCF,
STOPPED">
  </TERMINATION>
</EXEC_IVR_MAILBOX>
```

The EVENT="EOCS" indicates that the call flow was executed successfully and EVENT="EOCF" indicates that the call flow encountered errors and EVENT="STOPPED" indicates that the call flow was terminated by the <STOP\_EXE\_IVR\_MAILBOX>.

### Example 1:

```
<EXEC_IVR_MAILBOX>
  <TERMINATION MSGID="786123" BOXNO="7325667661" TRUNK_PORT="1"
VAR1="1234" VAR2="rec1.vox" VAR3="5678" VAR4="3456" VAR5="3333"
EVENT="EOCS">
  </TERMINATION>
</EXEC_IVR_MAILBOX>
```

In the example 1, the Computer Telephony Server has executed the call flow of mailbox number 7325667661. The EVENT ="EOCS" indicates that End Of Call flow Success

### Example 1:

```
<EXEC_IVR_MAILBOX>
  <TERMINATION MSGID="786123" BOXNO="7325667661" TRUNK_PORT="1"
    VAR1="1234" VAR2="rec1.vox" VAR3="5678" VAR4="3456" VAR5="3333"
    EVENT="EOCS">
  </TERMINATION>
</EXEC_IVR_MAILBOX>
```

In the example 1, the Computer Telephony Server has executed the call flow of mailbox number 7325667661. The EVENT ="EOCS" indicates that End Of Call flow Success

Note: The VAR1, VAR2, VAR3, VAR4 and VAR5 contains the modified values within the call flow.

### Example 2:

```
<EXEC_IVR_MAILBOX>
  <TERMINATION MSGID="786123" MAIL_BOX="7325667541" TRUNK_PORT="1"
    VAR1="1234" VAR2="rec1.vox" VAR3="5678" VAR4="3456" VAR5="3333"
    EVENT="EOCF">
  </TERMINATION>
</EXEC_IVR_MAILBOX>
```

In the example 2, the Computer Telephony Server has executed the call flow 7325667541. The EVENT ="EOCF" indicates that End Of Call flow failure.

### Example 3:

```
<EXEC_IVR_MAILBOX>
  <TERMINATION MSGID="786123" MAIL_BOX="7325667541" TRUNK_PORT="1"
    VAR1="1234" VAR2="rec1.vox" VAR3="5678" VAR4="3456" VAR5="3333"
    EVENT="STOPPED">
  </TERMINATION>
</EXEC_IVR_MAILBOX>
```

In the example 3, the Computer Telephony Server has executed the call flow 7325667541. The EVENT ="STOPPED" indicates the call flow was stopped by the <STOP\_EXEC\_IVR\_MAILBOX> XML message. .

### Stop IVR Mailbox Execution - <STOP\_EXEC\_IVR\_MAILBOX><INFO>

The applications program sends this message to Computer Telephony Server to stop the execution of an IVR call flow in progress.

Format:

```
<STOP_EXEC_IVR_MAILBOX>
  <INFO MSGID="Message ID" BOXNO="Mailbox number" TRUNK_PORT="Port
  Number">
  </INFO>
</STOP_EXEC_IVR_MAILBOX>
```

### Response Stop Call Flow - <STOP\_EXEC\_IVR\_MAILBOX><RESPONSE>

Computer Telephony Server sends this message immediately to the applications program if for any reason it can not stop the execution.

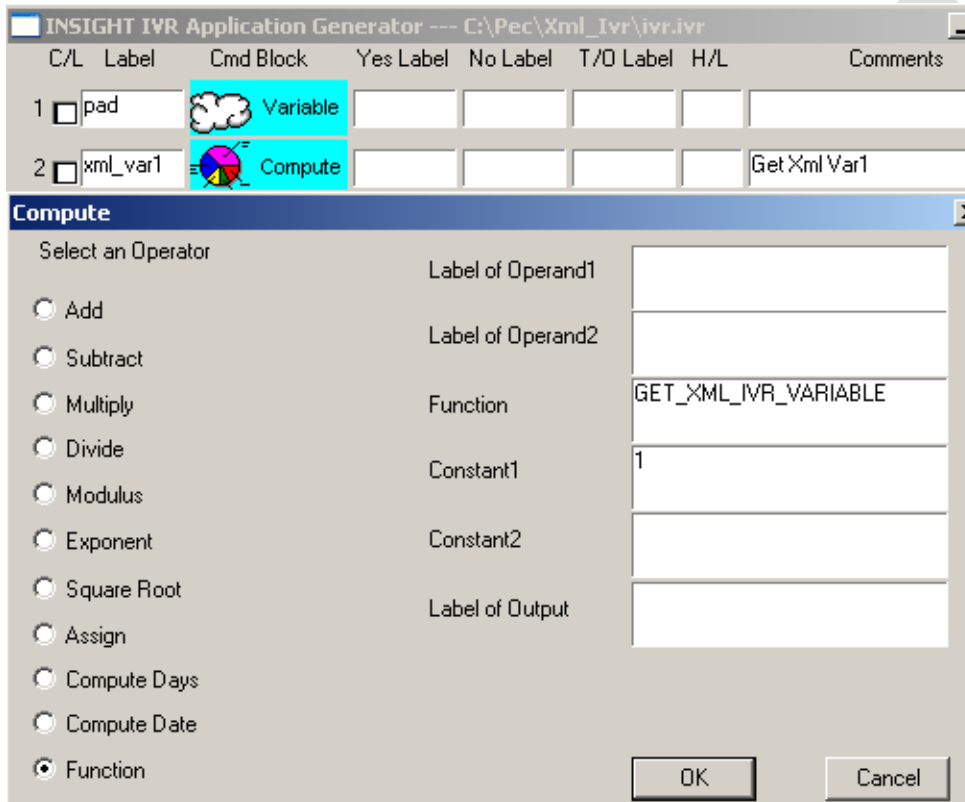
Format:

```
<STOP_EXEC_IVR_MAILBOX>
  <RESPONSE MSGID="Message Id" BOXNO="Mailbox number"
  TRUNK_PORT="Port Number" STATUS="SUCCESS, INVALID_MAIL_BOX,
  INVALID_TRUNK_PORT, MAIL_BOX_NOT_EXECUTING">
  </RESPONSE>
</STOP_EXEC_IVR_MAILBOX>
```

## Appendix A

### Reading <EXEC\_IVR\_MAILBOX> Tag Variables by the IVR

The IVR Applications Developer must use the Compute block of the Applications Generator to get the <EXEC\_IVR\_MAILBOX> tag variables into a label variable. Please see the following example:

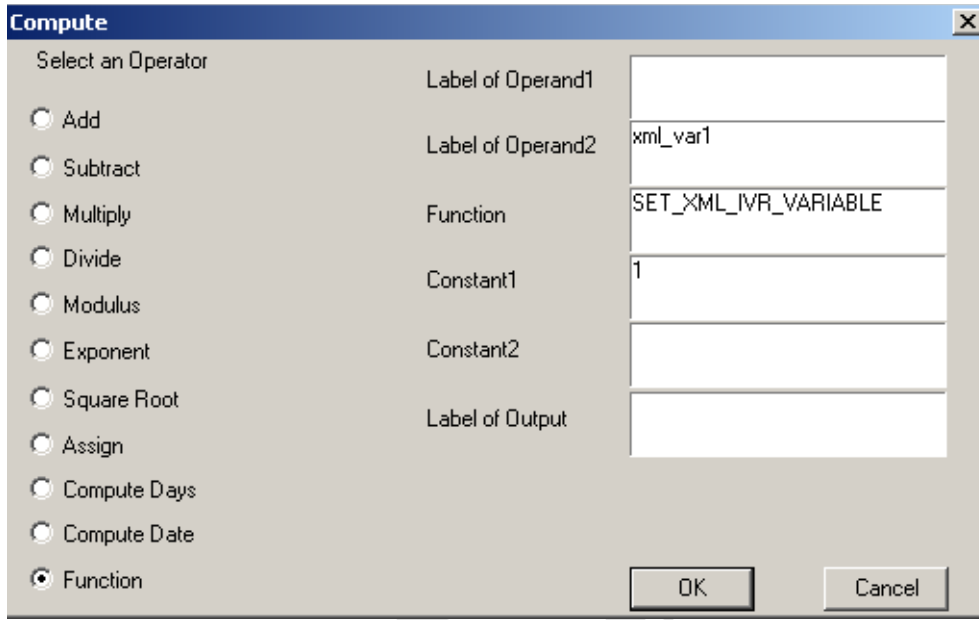


The above compute block will get the <EXEC\_IVR\_MAILBOX> tag Var1 to the xml\_var1 label. In order to get the <EXEC\_IVR\_MAILBOX> tag Var2, set the Constant1 to 2.



### Setting <EXEC\_IVR\_MAILBOX> Tag Variables from the IVR

The IVR Applications Developer must use the Compute block of the Applications Generator to set the <EXEC\_IVR\_MAILBOX> tag variables. Please see the following example:



Compute	
Select an Operator	Label of Operand1
<input type="radio"/> Add	Label of Operand2
<input type="radio"/> Subtract	Function
<input type="radio"/> Multiply	Constant1
<input type="radio"/> Divide	Constant2
<input type="radio"/> Modulus	Label of Output
<input type="radio"/> Exponent	
<input type="radio"/> Square Root	
<input type="radio"/> Assign	
<input type="radio"/> Compute Days	
<input type="radio"/> Compute Date	
<input checked="" type="radio"/> Function	

The above compute block sets the <EXEC\_IVR\_MAILBOX> tag var1 to the data in the xml\_var1 label. To set the <EXEC\_IVR\_MAILBOX> tag var2, then set Constant1 to 2.