CallCopy | Innovations in Call Recording and Contact Center Solutions

CallCopy API Manual

Version 4.5 R1

Contact Information:
888.922.5526 Option 2 – Support
support@callcopy.com
www.callcopy.com/support

Reference Guide

callcopy.com

Security Classification: CallCopy Confidential

Distribution: Approved internal CallCopy staff only and licensed CallCopy customers

Note: Applicable non-disclosure agreements must be in force for authorization.

Revision History			
Revision	**Change Description**	**Author**	**Effective Date**
0	Initial Release	Matt Madzia	5/18/2011
0.1	Corrected default Event Interface TCP Port (5621)	Matt Madzia	5/27/2011
0.2	Corrected typo in CALLSTOP description. Changed CALLSTART return to CALLSTOP.	JThomas	11/9/2011
1	Added reference to cc: Discover Admin Guide for TCP Settings configuration information.	MBuckingham	12/13/2012

Table of Contents

Chapter 1: Introduction

The CallCopy API lets you add custom data and integration seamlessly to external applications. For example, you may have customer specific information you would like to store with each call record, or perhaps you would like to specify when exactly to start and stop recording an agent, based off specific events.

Our API is extremely efficient and flexible. The CallCopy API can be used in any program written in any language that can communicate over TCP/IP. Messages are sent in a simple and descriptive XML format.

Additionally, commands can be sent via http requests using the WebAPI method.

CC:

The API is a standard blocking TCP server listening on port 5620. Usage is as simple as connecting to the port and sending a well-formatted XML request. You may then drop the connection, or stay connected to send another request.

Messages that are sent should end with a carriage return / line feed character (\r\n or ASCII character #13 [Hex $A]) to signal the end of data transmission.

Requests must be in a well-formed complete XML format. The XML may contain white spaces, as they are stripped out by the server.

The root XML tag for the message should be the message type you are sending. Child nodes of the XML message will be attributes to the request. One common attribute that can be sent in every message is REQUESTID.

Example:

```
<REQUEST>
  <TYPE>TEST</TYPE>
  <REQUESTID>1</REQUESTID>
</REQUEST>
```

In this example the message type is 'TEST'. This particular message can be used as an echo test to the server to ensure the availability of the server. The REQUESTID is a number that will be repeated back in the result message. The result for this message will look like the following:

```
<RESULT>
 <REQUESTID>1</REQUESTID>
 <REQUESTTYPE>TEST</REQUESTTYPE>
 <RESULTTYPE>API_OK</RESULTTYPE>
 <RESULTCODE>20</RESULTCODE>
 <RESULTMSG>API_OK</RESULTMSG>
</RESULT>
```

Please note that the CallCopy uses the DeviceID to track phone states. This is true even in a free seating environment.

Chapter 2: Handling Functions

The call handling functions are used either to replace CTI integration or to supplement an existing CTI integration. These functions can be used to update the system with call information pertaining to the current statuses of the devices on your phone system.

The following messages follow the normal call flow path as messages from a CTI integration do. This means that the scheduler handles call recording decisions. Because of this, statistics can naturally also be gathered by the system.

CALLSTART

Description: Allows a user to specify that a device has started a new phone call. This information is used to update live displays, as well in the scheduling system

Parameters:

- REQUESTID (required)
- DEVICEID *****
- CALLINSTANCE
- DEVICEALIAS *** +**
- GROUP
- GATE
- ANI
- DNIS
- USER1 – USER15
- CALLID
- PRIORITY
- CALLDIRECTION
- MAXRECORDSILENCE
- MAXRECORDTIME
- AGENTINITIATED
- CTIINITIATED
- ARCHIVEACTION
- STATIONNAME *****
- SYS_USER *** +**
- CRM_USER *** +**
- At least one of the parameters marked with an * is required for the request to be executed.
- **+** These parameters can only be used if a device ID has already been associated with an agent

Returns:

- **API_ERROR_RECORDER_NOT_RUNNING-**if the api could not send the command to any of the related cores

- **API _PARTIAL_CORE _RESPONSE-**if the api was able to connect to some cores but not others the RESULTMSG will be 'CALLSTART `could not be sent to the following cores:` #,#,#'
- **API_OK-**if the api was able to connect to all cores the RESULTMSG will be 'CALLSTART `sent to all of the following cores:` #,#,#'

When sending a callstart by DEVICEALIAS, SYS_USER, or CRM_USER, if a matching device isn't found, then this result is returned.

```
<RESULT>
<REQUESTTYPE>CALLSTART</REQUESTTYPE>
<REQUESTID>1</REQUESTID>
<RESULTTYPE>API_DEVICE_INVALID</RESULTTYPE>
<RESULTCODE>8</RESULTCODE>
<RESULTMSG>Could not find a device matching given parameters</RESULTMSG>
</RESULT>
```

CC:

STATIONNAME may be passed as the required item in place of DEVICEID. If DEVICEID is not passed or is passed as a blank value, CallCopy will look up the DEVICEID based on the value from the COMPUTERNAME field.

You may also pass the STATIONNAME to override the station mapping that is stored in the CallCopy database for the DEVICEID.

Callinstance is used to distinguish the calls on the same device at the same time. If a callinstance is not provided on callstart it will default to a blank callinstance.

Example:

```
<REQUEST>
  <TYPE>CALLSTART</TYPE>
  <REQUESTID>1</REQUESTID>
  <DEVICEID>555</DEVICEID>
  <DEVICEALIAS>3545</DEVICEALIAS>
  <GROUP>12</GROUP>
  <GATE>80</GATE>
  <ANI>6145551212</ANI>
  <DNIS>8889225526</DNIS>
  <USER1>Gold Level</USER1>
  <USER2>Customer 564582</USER2>
</REQUEST>

<RESULT>
<REQUESTTYPE>CALLSTART</REQUESTTYPE>
<REQUESTID>1</REQUESTID>
<RESULTTYPE>API_DEVICE_INVALID</RESULTTYPE>
<RESULTCODE>8</RESULTCODE>
<RESULTMSG>Could not find a device matching given parameters</RESULTMSG>
</RESULT>
```

CALLSTOP

Description: Allows a user to specify that a device has ended phone call. This information is used to update System Status and Live Monitoring screens, as well as stop recording if the device is currently being recorded.

Parameters:

- REQUESTID (required)
- DEVICEID *****
- CALLINSTANCE
- SYS_USER ***+**
- CRM_USER ***+**
- DEVICEALIAS ***+**
- STATIONNAME *****

***** At least one of the parameters marked with an * is required for the request to be executed.
+ These parameters can only be used if a device ID has already been associated with an agent

Returns:

- **API_ERROR_RECORDER_NOT_RUNNING-**if the api could not send the command to any of the related cores
- **API _PARTIAL_CORE _RESPONSE-**if the api was able to connect to some cores but not others the RESULTMSG will be 'CALLSTOP `could not be sent to the following cores:` #,#,#'
- **API_OK-**if the api was able to connect to all cores the RESULTMSG will be 'CALLSTOP `sent to all of the following cores:` #,#,#'

Example:

```
<REQUEST>
  <TYPE>CALLSTOP</TYPE>
  <REQUESTID>2</REQUESTID>
  <DEVICEID>555</DEVICEID>
</REQUEST>
```

Returns: Standard result code

CC:

DEVICEALIAS, CRM_USER, and SYS_USER may be passed as the required item in place of DEVICEID. If DEVICEID is not passed or is passed as a blank value, CallCopy will look up the DEVICEID based on the value provided. If the call instance is not provided the call instance of the last call on this device is used.

Callinstance is used to distinguish which call the callstop is being issued to. If no callinstance is provided on callstop then the callinstance of the last call on the device is used.

CALLUPDATE

Description: Allows a user to update information on a device that is currently on a phone call. This information is used to update live displays and reporting. This information will not be used by the scheduling system because it is assumed that the call is in progress at this point. Prior to this message, a CALLSTART message must be sent. If "update_if_not_recording" is set to 'Y' and the device is not in a call, then it will update the last call that was recorded for the given device.
If DeviceID is not present, it will be looked up by either CRM_User or Sys_User, whichever is present. If none are present, update will fail. (api_device_invalid)

Parameters:

- REQUESTID (required)
- DEVICEID *****
- CALLINSTANCE
- CRM_USER ***+**
- SYS_USER ***+**
- DEVICEALIAS ***+**
- STATIONNAME *****
- GROUP
- GATE
- ANI
- DNIS
- USER1 – USER15
- CALLDIRECTION
- ARCHIVEACTION
- KEEPDAYS
- UPDATE_IF_NOT_RECORDING

***** At least one of the parameters marked with an * is required for the request to be executed.
+ These parameters can only be used if a device ID has already been associated with an agent

Returns:

- **API_ERROR_RECORDER_NOT_RUNNING-**if the api could not send the command to any of the related cores
- **API _PARTIAL_CORE _RESPONSE-**if the api was able to connect to some cores but not others the RESULTMSG will be 'CALLSTART `could not be sent to the following cores:` #,#,#'
- **API_OK-**if the api was able to connect to all cores the RESULTMSG will be 'CALLSTART `sent to all of the following cores:` #,#,#'
- **API_RECORDID_INVALID** if *update_if_not_recording* is set and there was not a previous call to update.
- **API_DATABASE_QUERY_ERROR -**if the api doesn't successfully update a recording in the recordings table

Example:

```
<REQUEST>
  <TYPE>CALLUPDATE</TYPE>
  <REQUESTID>1</REQUESTID>
  <DEVICEID>555</DEVICEID>
  <USER3>Follow up on order</USER3>
</REQUEST>

<RESULT>
<REQUESTTYPE>CALLUPDATE</REQUESTTYPE>
<REQUESTID>1</REQUESTID>
<RESULTTYPE>API_OK</RESULTTYPE>
<RESULTCODE>20</RESULTCODE>
<RESULTMSG>CALLUPDATE  sent to all of the following cores: 1</RESULTMSG>
</RESULT>
```

```
<REQUEST>
  <TYPE>CALLUPDATE</TYPE>
  <REQUESTID>3</REQUESTID>
  <DEVICEID>555</DEVICEID>
  <USER3>Follow up on order</USER3>
  <UPDATE_IF_NOT_RECORDING>Y</UPDATE_IF_NOT_RECORDING>
</REQUEST>


<RESULT>
<REQUESTTYPE>CALLUPDATE</REQUESTTYPE>
<REQUESTID>3</REQUESTID>
<RESULTTYPE>API_OK</RESULTTYPE>
<RESULTCODE>20</RESULTCODE>
<RESULTMSG>Updated recording in database</RESULTMSG>
</RESULT>
```

Callinstance is used to distinguish which call the callupdate is being issued to. If no callinstance is provided on callupdate then the callinstance of the last call on the device is used.

CALLUPDATE – Post Recording

Description: Allows a user to update information about a call that is done recording. This information is used to update the CallCopy database. This information will not be used by the scheduling or archive system because it is assumed that the call is completed at this point. By passing a RecordID parameter, it becomes assumed the call update is for a call post recording.

Parameters:

- REQUESTID (required)
- RECORDID (required)
- DEVICEID
- DEVICEALIAS
- GROUP
- GATE
- ANI
- DNIS
- USER1 – USER15
- CALLDIRECTION

Returns: Standard result code on success.

- **API_OK** if successful.
- **API_REQUESTID_INVALID** if the request id is invalid.
- **API_DATABASE_QUERY_ERROR -**if the api doesn't successfully update a recording in the recordings table

Example:

```
<REQUEST>
  <TYPE>CALLUPDATE</TYPE>
  <REQUESTID>1</REQUESTID>
  <RECORDID>554853</RECORDID>
  <USER4>Sales Call</USER4>
</REQUEST>
```

Chapter 3: Status Functions

The status functions allow you to receive current information from the system. This information can be used to track system, channel, or phone states and to write custom monitors for the system.

The following messages are informational only and do not affect the process of recording on the system

DEVICELIST

Description: Queries the system for a current list of phones/devices, and the current information associated with the devices. The current status of any device that CallCopy is aware of is kept in memory on the system.

Parameters:

- REQUESTID

Returns: RESULT section plus an additional section containing a DEVICE section containing:

- DEVICEID
- DEVICEALIAS
- CALLDISCRIMINATOR
- SYS_USER
- CRM_USER
- GROUP
- GATE
- ANI
- DNIS
- USER1 – USER15
- CALLDIRECTION
- ISTRUNK
- STATION
- STARTTIME *
- STOPTIME
- COREIDENT
- RECORDING section:
  - ISRECORDING
  - ISRECORDINGVIDEO
  - FILENAME
  - CALLID
  - SESSIONID
  - CALLINSTANCE

Example Result:

```
<RESULT>
  <REQUESTTYPE>DEVICELIST</REQUESTTYPE>
  <REQUESTID>1</REQUESTID>
```

```
  <RESULTTYPE>OK</RESULTTYPE>
  <RESULTCODE>20</RESULTCODE>
  <RESULTMSG>OK</RESULTMSG>
<DEVICES>
    <COUNT>1</COUNT>
    <DEVICE>
       <DEVICEID>1111</DEVICEID>
       <DEVICEALIAS>DESK1</DEVICEALIAS>
       <SYS_USER>gkerber</SYS_USER>
       <CRM_USER>CCGKERBER</CRM_USER>
       <GROUP>MANAGER</GROUP>
       <GATE>prize</GATE>
       <ANI></ANI>
       <DNIS></DNIS>
       <USER1>E</USER1>
       <USER2>winners</USER2>
       <USER3></USER3>
       <USER4></USER4>
       <USER5></USER5>
       <USER6></USER6>
       <USER7></USER7>
       <USER8></USER8>
       <USER9></USER9>
       <USER10></USER10>
       <USER11></USER11>
       <USER12></USER12>
       <USER13></USER13>
       <USER14></USER14>
       <USER15></USER15>
       <CALLDIRECTION>I</CALLDIRECTION>
       <ISTRUNK></ISTRUNK>
       <STATION></STATION>
       <STARTTIME>2/1/2011 11:49:32 AM</STARTTIME>
       <STOPTIME></STOPTIME>
       <COREIDENT>2</COREIDENT>
       <RECORDING>
         <ISRECORDING>Y</ISRECORDING>
         <ISRECORDINGVIDEO>Y</ISRECORDINGVIDEO>
         <FILENAME>C:\Recordings\20110131\DESK1\DESK1-17-03-14.wav</FILENAME>
         <CALLID></CALLID>
         <SESSIONID></SESSIONID>
         <CALLINSTANCE>2001</CALLINSTANCE>
       </RECORDING>
     </DEVICE>
   </DEVICES>
</RESULT>
```

DEVICESTATUS

Description: Queries the system for a current for information pertaining to a specific device or devices.

*Optional Parameters – if specified will only return a list of devices that match:*DEVICESTATUS*

- DEVICEID (optional)
- DEVICEALIAS (optional)
- CRM_USER (optional)

Parameters:

- REQUESTID
- DEVICEID
- DEVICEALIAS
- CRM_USER

Returns: RESULT section plus an additional section containing a DEVICE section containing:

- DEVICEID
- DEVICEALIAS
- CALLDISCRIMINATOR
- SYS_USER
- CRM_USER
- GROUP
- GATE
- ANI
- DNIS
- USER1 – USER15
- CALLDIRECTION
- ISTRUNK
- STATION
- STARTTIME
- STOPTIME
- COREIDENT
- RECORDING section:
  - ISRECORDING
  - ISRECORDINGVIDEO
  - FILENAME
  - CALLID
  - SESSIONID
  - CALLINSTANCE

Example Result:

```
<REQUEST>
<TYPE>DEVICESTATUS</TYPE>
```

```
<DEVICEALIAS>DESK1</DEVICEALIAS>
<REQUESTID>24</REQUESTID>
</REQUEST>


<RESULT>
  <REQUESTTYPE>DEVICESTATUS</REQUESTTYPE>
  <REQUESTID>24</REQUESTID>
  <RESULTTYPE>OK</RESULTTYPE>
  <RESULTCODE>20</RESULTCODE>
  <RESULTMSG>Found the following matching devices</RESULTMSG>
<DEVICES>
    <COUNT>1</COUNT>
    <DEVICE>
       <DEVICEID>1111</DEVICEID>
       <DEVICEALIAS>DESK1</DEVICEALIAS>
       <SYS_USER>gkerber</SYS_USER>
       <CRM_USER>CCGKERBER</CRM_USER>
       <GROUP>MANAGER</GROUP>
       <GATE>prize</GATE>
       <ANI></ANI>
       <DNIS></DNIS>
       <USER1>E</USER1>
       <USER2>winners</USER2>
       <USER3></USER3>
       <USER4></USER4>
       <USER5></USER5>
       <USER6></USER6>
       <USER7></USER7>
       <USER8></USER8>
       <USER9></USER9>
       <USER10></USER10>
       <USER11></USER11>
       <USER12></USER12>
       <USER13></USER13>
       <USER14></USER14>
       <USER15></USER15>
       <CALLDIRECTION>I</CALLDIRECTION>
       <ISTRUNK></ISTRUNK>
       <STATION></STATION>
       <STARTTIME>2/1/2011 11:49:32 AM</STARTTIME>
       <STOPTIME></STOPTIME>
       <COREIDENT>2</COREIDENT>
       <RECORDING>
         <ISRECORDING>Y</ISRECORDING>
         <ISRECORDINGVIDEO>Y</ISRECORDINGVIDEO>
         <FILENAME>C:\Recordings\20110131\DESK1\DESK1-17-03-14.wav</FILENAME>
         <CALLID></CALLID>
         <SESSIONID></SESSIONID>
         <CALLINSTANCE>2001</CALLINSTANCE>
       </RECORDING>
     </DEVICE>
   </DEVICES>
</RESULT>
```

SYSTEMSTATUS

Description: Queries the system for a system status. Information returned contains information pertaining to the health of the system, as well as, disk space, etc...

Parameters:

- REQUESTID

Returns: RESULT section plus an additional information tags:

- RECORDINGPATH
- RECORDINGSPACEINFO
- RECORDINGTOTALDISKCAPACITY
- RECORDINGTOTALDISKFREE
- RECORDINGTOTALDISKFREEPCT
- SYSTEMUPTIME
- PROCESSOR#
  - SYSTEMPROCESSORSPEED
  - SYSTEMPROCESSORTYPE
- SYSTEMRAMTOTAL
- SYSTEMRAMFREE
- SYSTEMRAMUSEDPCT

```
<REQUEST>
<TYPE>SYSTEMSTATUS</TYPE>
<REQUESTID>255</REQUESTID>
</REQUEST>

<RESULT>
<REQUESTTYPE>SYSTEMSTATUS</REQUESTTYPE>
<REQUESTID>255</REQUESTID>
<RESULTTYPE>API_OK</RESULTTYPE>
<RESULTCODE>20</RESULTCODE>
<RESULTMSG>API_OK</RESULTMSG>
<RECORDINGPATH>C:\Recordings\</RECORDINGPATH>
<RECORDINGSPACEINFO>223.66GB of 465.64GB</RECORDINGSPACEINFO>
<RECORDINGTOTALDISKCAPACITY>476819</RECORDINGTOTALDISKCAPACITY>
<RECORDINGTOTALDISKFREE>229032</RECORDINGTOTALDISKFREE>
<RECORDINGTOTALDISKFREEPCT>48.03</RECORDINGTOTALDISKFREEPCT>
<SYSTEMUPTIME>0 days, 18 hours, 24 minutes, 37 seconds.</SYSTEMUPTIME>
<PROCESSOR0>
 <SYSTEMPROCESSORSPEED>2394MHz</SYSTEMPROCESSORSPEED>
 <SYSTEMPROCESSORTYPE>x86 Family 6 Model 15 Stepping 11</SYSTEMPROCESSORTYPE>
</PROCESSOR0>
<PROCESSOR1>
 <SYSTEMPROCESSORSPEED>2394MHz</SYSTEMPROCESSORSPEED>
 <SYSTEMPROCESSORTYPE>x86 Family 6 Model 15 Stepping 11</SYSTEMPROCESSORTYPE>
</PROCESSOR1>
<PROCESSOR2>
 <SYSTEMPROCESSORSPEED>2394MHz</SYSTEMPROCESSORSPEED>
 <SYSTEMPROCESSORTYPE>x86 Family 6 Model 15 Stepping 11</SYSTEMPROCESSORTYPE>
</PROCESSOR2>
<PROCESSOR3>
 <SYSTEMPROCESSORSPEED>2394MHz</SYSTEMPROCESSORSPEED>
 <SYSTEMPROCESSORTYPE>x86 Family 6 Model 15 Stepping 11</SYSTEMPROCESSORTYPE>
</PROCESSOR3>
<SYSTEMRAMTOTAL>3325</SYSTEMRAMTOTAL>
<SYSTEMRAMFREE>1702</SYSTEMRAMFREE>
```

```
<SYSTEMRAMUSED>1623</SYSTEMRAMUSED>
<SYSTEMRAMUSEDPCT>48</SYSTEMRAMUSEDPCT>
</RESULT>
```

CHANNELS

Description: Queries the system for status of recorded channels.

Parameters:

- REQUESTID

Returns:

- **API_DATABASE_QUERY_ERROR -**if the api doesn't successfully connect to the database

Successful: RESULT section plus an additional CHANNELS section containing a CHANNEL section per channel on the system containing:

- ID
- STATE
- PLAYBACKEXTENSION
- RECORDINGDEVICEALIAS
- RECORDINGDEVICE
- LASTSTATECHANGE
- LASTSTATECHANGEUNIXTIME

ID is the integer index of the channel starting with 1 and ending with the value contained in COUNT

```
<REQUEST>
<TYPE>CHANNELS</TYPE>
<REQUESTID>255</REQUESTID>
</REQUEST>

<RESULT>
<REQUESTTYPE>CHANNELS</REQUESTTYPE>
<REQUESTID>26</REQUESTID>
<RESULTTYPE>API_OK</RESULTTYPE>
<RESULTCODE>20</RESULTCODE>
<RESULTMSG>API_OK</RESULTMSG>
<COUNT>4</COUNT>
<CHANNELS>
<CHANNEL>
 <ID>0</ID>
 <STATE>PreparingToRecord</STATE>
 <PLAYBACKEXTENSION></PLAYBACKEXTENSION>
 <RECORDINGDEVICEALIAS></RECORDINGDEVICEALIAS>
 <RECORDINGDEVICE>7002</RECORDINGDEVICE>
 <LASTSTATECHANGE>1/31/2011 5:50:33 PM</LASTSTATECHANGE>
 <LASTSTATECHANGEUNIXTIME>1296496234</LASTSTATECHANGEUNIXTIME>
</CHANNEL>
<CHANNEL>
```

```
 <ID>1</ID>
 <STATE>Idle</STATE>
 <PLAYBACKEXTENSION></PLAYBACKEXTENSION>
 <RECORDINGDEVICEALIAS></RECORDINGDEVICEALIAS>
 <RECORDINGDEVICE>7003</RECORDINGDEVICE>
 <LASTSTATECHANGE>1/31/2011 6:15:52 PM</LASTSTATECHANGE>
 <LASTSTATECHANGEUNIXTIME>1296497752</LASTSTATECHANGEUNIXTIME>
</CHANNEL>
<CHANNEL>
 <ID>2</ID>
 <STATE>Idle</STATE>
 <PLAYBACKEXTENSION></PLAYBACKEXTENSION>
 <RECORDINGDEVICEALIAS></RECORDINGDEVICEALIAS>
 <RECORDINGDEVICE>7055</RECORDINGDEVICE>
 <LASTSTATECHANGE>1/31/2011 6:11:40 PM</LASTSTATECHANGE>
 <LASTSTATECHANGEUNIXTIME>1296497501</LASTSTATECHANGEUNIXTIME>
</CHANNEL>
<CHANNEL>
 <ID>3</ID>
 <STATE>Idle</STATE>
 <PLAYBACKEXTENSION></PLAYBACKEXTENSION>
 <RECORDINGDEVICEALIAS></RECORDINGDEVICEALIAS>
 <RECORDINGDEVICE>7444</RECORDINGDEVICE>
 <LASTSTATECHANGE>1/31/2011 6:11:40 PM</LASTSTATECHANGE>
 <LASTSTATECHANGEUNIXTIME>1296497501</LASTSTATECHANGEUNIXTIME>
</CHANNEL>
</CHANNELS>
</RESULT>
```

STATIONS

Description: Queries the system for a list of programmed stations

Parameters:

- REQUESTID

Returns:

- **API_DATABASE_QUERY_ERROR -**if the api doesn't successfully connect to the database

If successful RESULT section including:

- COUNT

Plus an additional STATIONS section containing a STATION section per station on the system containing:

- STATIONNAME
- DEVICEID

```
<REQUEST>
<TYPE>STATIONS</TYPE>
<REQUESTID>29</REQUESTID>
</REQUEST>


<RESULT>
<REQUESTTYPE>STATIONS</REQUESTTYPE>
```

```
<REQUESTID>29</REQUESTID>
<RESULTTYPE>API_OK</RESULTTYPE>
<RESULTCODE>20</RESULTCODE>
<RESULTMSG>API_OK</RESULTMSG>
<COUNT>2</COUNT>
<STATIONS>
 <STATION>
  <STATIONNAME>BOBDABUILDER</STATIONNAME>
  <DEVICEID>2222</DEVICEID>
 </STATION>
 <STATION>
  <STATIONNAME>JamesBond</STATIONNAME>
  <DEVICEID>7007</DEVICEID>
 </STATION>
</STATIONS>
</RESULT>
```

FETCHTERMINOLOGY

Description: Queries the system for a list of terminology settings

Parameters:

- REQUESTID

Returns: RESULT section including:

- COUNT

Plus additional TERMINOLOGY sections for each terminology variable specified in the system:

- NAME
- VALUE

Example Request:

```
<REQUEST>
  <TYPE>FETCHTERMINOLOGY</TYPE>
  <REQUESTID>34</REQUESTID>
</REQUEST>
```

Example Result:

```
<RESULT>
  <REQUESTTYPE>FETCHTERMINOLOGY</REQUESTTYPE>
  <REQUESTID>34</REQUESTID>
  <COUNT>10</COUNT>
  <SETTINGS>
    <TERMINOLOGY>
      <NAME>agent</NAME>
      <VALUE>Agent</VALUE>
    </TERMINOLOGY>
    <TERMINOLOGY>
      <NAME>ani</NAME>
      <VALUE>CallerID ANI</VALUE>
    </TERMINOLOGY>
    <TERMINOLOGY>
      <NAME>callcopygroup</NAME>
      <VALUE>CallCopy group</VALUE>
    </TERMINOLOGY>
```

```
    <TERMINOLOGY>
      <NAME>devicealias</NAME>
      <VALUE>Agent Number</VALUE>
    </TERMINOLOGY>
    <TERMINOLOGY>
      <NAME>deviceid</NAME>
      <VALUE>Voice Port</VALUE>
    </TERMINOLOGY>
    <TERMINOLOGY>
      <NAME>dnis</NAME>
      <VALUE>Number Called DNIS</VALUE>
    </TERMINOLOGY>
    <TERMINOLOGY>
      <NAME>gate</NAME>
      <VALUE>ACD Gate</VALUE>
    </TERMINOLOGY>
    <TERMINOLOGY>
      <NAME>group</NAME>
      <VALUE>Group</VALUE>
    <TERMINOLOGY>
      <NAME>user1</NAME>
      <VALUE>Account Number</VALUE>
    </TERMINOLOGY>
    <TERMINOLOGY>
      <NAME>user2</NAME>
      <VALUE>CSN</VALUE>
    </TERMINOLOGY>
  </SETTINGS>
</RESULT>
```

Chapter 4: Recording Control Functions

The recording control functions allow you to write software to control when the system should record. The system allows you to start and stop recording at will

Using these functions you can create your own custom triggers in your software to start and stop recording. Some ideas for this are: To allow agents to record harassing callers or recording the part of a call containing credit card verification.

These functions will not cause problems if the device is already being recorded, or is not being recorded. In other words, you do not need to first check the status of a device before sending this message.

CHATSTART

Description: Used to initiate a desktop-screen capture only recording session. The recorder assumes there is no audio with this call. This is commonly used with web-based chat applications.

Parameters:

- REQUESTID
- SESSIONID - session id (REQUIRED)
- AGENTID

If AGENTID isn't provided CallCopy can look up the agent using KEYNAME and KEYVALUE.

- KEYNAME – USERNAME (sysusername) EMPLOYEEID, CRMUSERNAME, DEVICEALIAS (Phone ID)
- KEYVALUE - The username or employee id as dictated in the keyname.

Returns:

- **API_SESSIONID_UNSPECIFIED -**if the api doesn't successfully connect to the database
- **API_AGENT_NOT_FOUND –** is returned If CallCopy cannot locate an agent based on the information provided, detail description will be in the RESULTMSG
- **API_ERROR_RECORDER_NOT_RUNNING-**if the api could not send the command to any of the related cores
- **API_OK-**if the api was able to connect to core

Standard result codes

Example:

```
<REQUEST>
  <REQUESTID>37</REQUESTID>
  <TYPE>CHATSTART</TYPE>
  <SESSIONID>82738</SESSIONID>
  <KEYNAME>USERNAME</KEYNAME>
  <KEYVALUE>JDOE</KEYVALUE>
</REQUEST>


<RESULT>
```

```
 <REQUESTTYPE>CHATSTART</REQUESTTYPE>
 <REQUESTID>37</REQUESTID>
 <RESULTTYPE>API_OK</RESULTTYPE>
 <RESULTCODE>20</RESULTCODE>
 <RESULTMSG>CHATSTART  sent to all of the following cores: 1</RESULTMSG>
</RESULT>
```

CHATSTOP

Description: Used to stop a desktop-screen capture recording session. This information is used to update live displays, as well as stop recording if the device is currently being recorded.

Parameters:

- REQUESTID - request id
- SESSIONID - session id (REQUIRED)

Returns:

- **API_SESSIONID_UNSPECIFIED -**if the api doesn't successfully connect to the database
- **API_ERROR_RECORDER_NOT_RUNNING-**if the api could not send the command to any of the related cores
- **API_OK-**if the api was able to connect to core

Standard result code

Example:

```
<REQUEST>
  <REQUESTID>52</REQUESTID>
  <TYPE>CHATSTOP</TYPE>
  <SESSIONID>82738</SESSIONID>
</REQUEST>

<RESULT>
 <REQUESTTYPE>CHATSTOP</REQUESTTYPE>
 <REQUESTID>46</REQUESTID>
 <RESULTTYPE>API_OK</RESULTTYPE>
 <RESULTCODE>20</RESULTCODE>
 <RESULTMSG>CHATSTOP  sent to all of the following cores: 1</RESULTMSG>
</RESULT>
```

CHATUPDATE

Parameters:

- REQUESTID - request id
- SESSIONID - session id (REQUIRED)
- DEVICEALIAS
- GROUP
- GATE
- ANI
- DNIS
- USER1-USER15

- CALLDIRECTION
- CALLID

Returns:

- **API_SESSIONID_UNSPECIFIED -**if the api doesn't successfully connect to the database
- **API_ERROR_RECORDER_NOT_RUNNING-**if the api could not send the command to any of the related cores
- **API_OK-**if the api was able to connect to core

Example:

```
<REQUEST>
  <REQUESTID>64</REQUESTID>
  <TYPE>CHATUPDATE</TYPE>
  <SESSIONID>82738</SESSIONID>
  <GROUP>53</GROUP>
</REQUEST>

<RESULT>
 <REQUESTTYPE>CHATUPDATE</REQUESTTYPE>
 <REQUESTID>64</REQUESTID>
 <RESULTTYPE>API_OK</RESULTTYPE>
 <RESULTCODE>20</RESULTCODE>
 <RESULTMSG>CHATUPDATE  sent to all of the following cores: 1</RESULTMSG>
</RESULT>
```

BLACKOUTSTART

Description: Used to start a blackout period for a recording. Only specify one of the following: deviceid, devicealias, crm_user, or sys_user. Based on when the message is received, and any time offset is when the blackout will start.

Note: This is an optional feature. The feature requires the system option for "Blackouts" to be enabled.

Parameters:

- REQUESTID - request id
- DEVICEID
- DEVICEALIAS
- CRM_USER
- SYS_USER
- STATIONNAME
- CALLINSTANCE
- TIMEOFFSET – Number of seconds to offset the start time of the blackout (can be a positive or negative integer)

Returns: Standard result code

Example:

```
<REQUEST>
  <REQUESTID>124</REQUESTID>
```

```
  <TYPE>BLACKOUTSTART</TYPE>
  <SYS_USER>JSMITH</SYS_USER>
  <TIMEOFFSET>-5</TIMEOFFSET>
</REQUEST>


<RESULT>
 <REQUESTTYPE>BLACKOUTSTART</REQUESTTYPE>
 <REQUESTID>124</REQUESTID>
 <RESULTTYPE>API_OK</RESULTTYPE>
 <RESULTCODE>20</RESULTCODE>
 <RESULTMSG>BLACKOUTSTART  sent to all of the following cores: 1</RESULTMSG>
</RESULT>
```

Errors: Possible error returns are:

- API_REQUESTID_INVALID – request id specified is not valid.
- API_DEVICE_INVALID – could not find a valid device from the given parameters.
- **API_ERROR_RECORDER_NOT_RUNNING-**if the api could not send the command to any of the related cores
- **API_OK-**if the api was able to connect to some cores but not others the RESULTMSG will be 'BLACKOUTSTART could not be sent to the following cores: #,#,#'

BLACKOUTSTOP

Description: Used to stop a blackout period for a recording. Only specify one of the following: deviceid, devicealias, crm_user, or sys_user. Based on when the message is received and time offset is when the blackout will stop.

CC: This is an optional feature. The feature requires the system option for "Blackouts" to be enabled.

Parameters:

- REQUESTID - request id
- DEVICEID
- DEVICEALIAS
- CRM_USER
- SYS_USER
- STATIONNAME
- CALLINSTANCE
- TIMEOFFSET - Number of seconds to offset the stop time of the blackout (can be a positive or negative integer)

Returns: Standard result code

Example:

```
<REQUEST>
  <REQUESTID>125</REQUESTID>
  <TYPE>BLACKOUTSTOP</TYPE>
```

```
  <SYS_USER>JSMITH</SYS_USER>
</REQUEST>
```

Errors: Possible error returns are:

- API_REQUESTID_INVALID – request id specified is not valid.
- API_DEVICE_INVALID – could not find a valid device from the given parameters.
- **API_ERROR_RECORDER_NOT_RUNNING-**if the api could not send the command to any of the related cores
- **API_OK-**if the api was able to connect to some cores but not others the RESULTMSG will be 'BLACKOUTSTART could not be sent to the following cores: #,#,#'

When a Time Offset causes a BLACKOUTSTOP to be issued before the start it will result in the start and stop times being the same. For example: if an BLACKOUTSTOP is sent with a time offset of -60 and the blackout started only 30 seconds ago the BLACKOUTSTOP time will be the same as the BLACKOUTSTART time.

Callinstance is used to distinguish which call the blackouts are issued to. If no callinstance is provided on blackoutstart or blackoutstop then the callinstance of the last call on the device is used.

Chapter 5: Playback Functions

The playback functions allow you to play audio files from the CallCopy system to an extension on your phone system.

Only certain audio formats are supported and additional hardware may be required for these functions to operate so contact CallCopy prior to using these functions.

1. Playback has 2 modes of operation:
   Using no barge code. When not passing a barge code, the CallCopy server will call the device to playback the call. The device must be answered before playback is started. When not using barge, the playback channel will be freed when the device is hung up.

2. Using a barge code. When passing a barge code, the CallCopy server will dial the barge code and the device. This will cause CallCopy to be automatically connected into the call currently in progress on the device and playback will start immediately. (This feature varies between phone systems) When using barge, the playback channel will be freed when playback is done, when EXTENSIONPLAYBACKSTOP is called, or when CallCopy detects a call stop for the device.

EXTENSIONPLAYBACKSTART

Description: Used to initiate a playback of a file or record. You may specify either a filename (which must be accessible from the CallCopy server) or a uniquefield/uniqueid pair which will be used to look up the file from the CallCopy database.

When DEVICE is specified, that will be the number used to dial for playback. If DEVICE is not specified, CRM_USER or SYS_USER must be present, then a look up will be done based on CRM_USER or SYS_USER to find what number that agent is on.

Parameters:

- REQUESTID – request id
- DEVICE – number to dial for playback
- BARGECODE – dial code to barge a call
- CRM_USER – username to look agent up by
- SYS_USER – username to look agent up by
- UNIQUE_FIELD – database column name to lookup by
- UNIQUE_ID – data to find from the UNIQUE_FIELD
- FILENAME – filename of the audio
- PLAYBACKTYPE – this is the type of playback method that will be used. Possible values:
  - BARGE – Playback using barge code method
  - SSC – Playback using single step conferencing method
  - DIAL – Playback using a simple dial to station method

Returns: Standard result code

Example:

```
<REQUEST>
  <TYPE>EXTENSIONPLAYBACKSTART</TYPE>
  <REQUESTID>1</REQUESTID>
  <CRM_USER>j_doe</CRM_USER >
  <UNIQUEFIELD>user1</UNIQUEFIELD>
  <UNIQUEID>122203</UNIQUEID>
</REQUEST>
```

Errors: Possible error returns are:

- API_REQUESTID_INVALID – request id specified is not valid.
- API_DEVICE_INVALID – could not find a valid device from the given parameters.
- API_FILE_NOT_FOUND – cannot find the file for playback

EXTENSIONPLAYBACKSTOP

Description: Used to stop playback of a file that was previously started with EXTENSIONPLAYBACKSTART.

When DEVICE is specified, that will be the number playback is stopped on. If DEVICE is not specified, CRM_USER or SYS_USER must be present, then a look up will be done based on CRM_USER or SYS_USER to find what number that agent is on.

Parameters:

- REQUESTID – request id
- DEVICE – number to dial for playback
- BARGECODE – dial code to barge a call
- CRM_USER – username to look agent up by
- SYS_USER – username to look agent up by
- PLAYBACKTYPE – this is the type of playback method that will be used. Possible values:
  - BARGE – Playback using barge code method
  - SSC – Playback using single step conferencing method
  - DIAL – Playback using a simple dial to station method

Returns: Standard result codes

Example:

```
<REQUEST>
  <TYPE>EXTENSIONPLAYBACKSTOP</TYPE>
  <REQUESTID>2</REQUESTID>
  <CRM_USER>j_doe</CRM_USER >
</REQUEST>
```

Errors: Possible error returns are:

- API_REQUESTID_INVALID – request id specified is not valid.
- API_DEVICE_INVALID – could not find a valid device from the given parameters.

Chapter 6: Import Functions

The import functions allow you to write integration links between your systems and the CallCopy system.

These functions can be used to write your own front end interface to edit certain information contained within CallCopy, or to write automated scripts that update information in the server.

IMPORTAGENT

Description: Allows a user to import an agent and the agents associated data (such as an employee id) into CallCopy. This function allows you to add, modify and delete information pertaining to an agent. Once an agent is added, you may use IMPORTAGENTPHONE to associate phones ID's to the agent.

Parameters:

- REQUESTID
- ACTION
- AGENTID
- FIRSTNAME
- LASTNAME
- SYSUSERNAME
- DOMAIN
- EMAIL
- EMPLOYEEID
- CRMUSERNAME
- ALLOWDUPLICATE (Y – Yes, N – No)
- STATUS (A – Active, I – Inactive)
- SYS_DOMAIN
- SITE_ID
- MOBILE_ID

Returns:

- Standard Result
- AGENTID

Example:

```
<REQUEST>
 <TYPE>IMPORTAGENT</TYPE>
<REQUESTID>2</REQUESTID>
<ACTION>ADD</ACTION
><FIRSTNAME>James</FIRSTNAME>
<LASTNAME>Doe</LASTNAME>
<SYSUSERNAME>doe.1</SYSUSERNAME>
<STATUS>A</STATUS>
<DOMAIN>somecompany</DOMAIN>
<EMAIL>jdoe@somecompany.com</EMAIL>
<EMPLOYEEID>1234</EMPLOYEEID>
</REQUEST>
```

List Example:

```
<REQUEST>
  <TYPE>IMPORTAGENT</TYPE>
  <REQUESTID>5</REQUESTID>
  <ACTION>LIST</ACTION>
</REQUEST>

<RESULT>
  <REQUESTTYPE>IMPORTAGENT</REQUESTTYPE>
  <REQUESTID>5</REQUESTID>
  <RESULTTYPE>OK</RESULTTYPE>
  <RESULTCODE>20</RESULTCODE>
  <RESULTMSG>OK</RESULTMSG>
  <AGENTS>
    <COUNT>1</COUNT>
    <AGENT>
      <AGENT_ID>2</AGENT_ID>
      <EMPLOYEE_ID>5</EMPLOYEE_ID>
      <FIRST_NAME>John</FIRST_NAME>
      <LAST_NAME>Doe</LAST_NAME>
      <STATUS>A</STATUS>
      <ADD_TIME>10/30/2006 4:05:38 PM</ADD_TIME>
      <MOD_TIME>3/27/2008 3:34:43 PM</MOD_TIME>
      <SYS_USER>jdoe</SYS_USER>
      <SYS_DOMAIN>CallCopy</SYS_DOMAIN>
      <EMAIL>jdoe@callcopy.com</EMAIL>
      <CRM_USER>john</CRM_USER>
      <PHONES>1123,3343,3234</PHONES>
    </AGENT>
  </AGENTS>
</RESULT>
```

CC:

The possible values for action are: ADD, MODIFY, DELETE, and LIST.

The field AGENTID is ignored when the action is ADD.

Deleting an agent will simply change the status flag of the agent to “deleted”, but the record will not actually be removed. This ensures that past records will still identify the correct agent in case a phone ID is reused for a new agent. Also, all phone ID associations to the agent are freed.

Modify should only be used to add additional information to the agent, or to change the name of the agent as long as it is still the same employee.

An error will be returned if the AGENTID does not exist, or if you try to add a new agent with the same first and last name as an existing agent. Even though an error is returned when you try to add an agent with the same first and last name as an existing agent, values are not changes. Because of this you may write an automated script that continuously calls ADD and ignore errors.

List should be used to get a list of all agents currently in CallCopy. No additional parameters are needed. Returned are agent_id, employee_id, first_name, last_name, status, add_time, mod_time, sys_user, email, crm_user, and phones.

IMPORTAGENTPHONE

Description: Allows a user to control the association of a phone device to an agent

Parameters:

- REQUESTID
- ACTION
- PHONEID
- AGENTID
- FIRSTNAME
- LASTNAME

Returns:

- Standard Result
- AGENTID

Example:

```
<REQUEST>
  <TYPE>IMPORTAGENTPHONE</TYPE>
  <REQUESTID>2</REQUESTID>
  <ACTION>ADD</ACTION>
  <PHONEID>1234</PHONEID>
  <AGENTID>0</AGENTID>
  <FIRSTNAME>John</FIRSTNAME>
  <LASTNAME>Doe</LASTNAME>
</REQUEST>
```

The possible values for action are: ADD, and DELETE.

You must pass either the AGENTID or the FIRSTNAME and LASTNAME of the agent. The system will first try to use the AGENTID. Passing both will not cause a problem; however in this case FIRSTNAME and LASTNAME will be ignored.

Deleting a phone ID will have no effect on the agent record itself.

Deleting a phone ID will not remove the call records for that phone ID.

An error will be returned if the AGENTID does not exist, or if you try to add a new phone ID that is already assigned to another agent.

IMPORTSTATION

Description: Allows an administrator to add or modify station to device mappin gs

Parameters:

- REQUESTID
- DEVICEID
- STATIONNAME

Returns:

- Standard Result
- DEVICEID
- STATIONNAME

Example:

```
<REQUEST>
  <TYPE>IMPORTSTATION</TYPE>
  <REQUESTID>2</REQUESTID>
  <DEVICEID>1234</DEVICEID>
  <STATIONNAME>COMPUTER1</STATIONNAME>
</REQUEST>
```

CC:

Before the insert, a delete will be done for entries containing either the DEVICEID or STATIONNAME to guarantee that the new entry is unique.

The result will contain the STATIONNAME that was sent from the request, however the DEVICEID that is returned is the value that is actually stored in the database. If an error occurs adding the new entry, please know that the result DEVICEID may be an old value. This design allows for easier troubleshooting.

Chapter 7: Permission Functions

The permission functions allow you to control the aspects of the CallCopy permissioning system from an external resource.

ADDGROUP

Description: Allows an administrator to add a CallCopy permissioning group

Parameters:

- REQUESTID
- GROUPNAME

Returns:

- Standard Result
- GROUPID
- GROUPNAME ADDED GROUP TO MATCH OTHER

Example:

```
<REQUEST>
  <TYPE>ADDGROUP</TYPE>  <REQUESTID>2</REQUESTID>
  <GROUPNAME>Sales & Marketing</GROUPNAME>
</REQUEST>

<RESULT>
 <REQUESTTYPE>ADDGROUP</REQUESTTYPE>
 <REQUESTID>2</REQUESTID>
 <RESULTTYPE>API_OK</RESULTTYPE>
 <RESULTCODE>20</RESULTCODE>
 <RESULTMSG>group: Sales & Marketing has been added</RESULTMSG>
 <GROUPID>11</GROUPID>
 <GROUPNAME>Sales & Marketing</GROUPNAME>
</RESULT>
```

CC: CallCopy will attempt to add the new group. The GROUPNAME and the GROUPID of the new group will be returned as the result. If a group with the same name already exists, the function will still return as normal without modifying any data.

MODIFYGROUPPERMISSION

Description: Allows an administrator to add or remove device assignments to/from groups

Parameters:

- REQUESTID
- ACTION
- GROUPNAME
- GROUPID

- PHONEID OR DEVICEID
- FIRSTNAME
- LASTNAME
- AGENTID

Returns:

- Standard Result
- Parameters Sent

Example:

```
<REQUEST>
  <TYPE>MODIFYGROUPPERMISSION</TYPE>
  <REQUESTID>2</REQUESTID>
  <ACTION>ADD</ACTION>
  <GROUPNAME>Sales & Marketing</GROUPNAME>
  <FIRSTNAME>John</FIRSTNAME>
  <LASTNAME>Doe</LASTNAME>
  <DEVICEID>1234</DEVICEID>
</REQUEST>
```

CC:

You must pass either ADD or DELETE as the ACTION type

You must supply either GROUPNAME or GROUPID, but you do not have to pass both

You must supply either FIRSTNAME and LASTNAME or AGENTID, but you do not have to pass both

The system will first look to see if you supplied the textual representation of the group name or the agent name and lookup the corresponding ID's for you. In either case the ID is then queried to verify that the correct group and agent entries are found.

The DEVICEID can be:

A PhoneID that is assigned to the agent

The result will include all data including the GROUPID and AGENTID if you did not pass them to the function

MODIFYUSERGROUPPERMISSION

Description: Allows an administrator to add or remove permissions for users to view items accessible by a particular CallCopy group.

Parameters:

- REQUESTID
- ACTION
- GROUPNAME
- GROUPID
- DEVICEID
- USERNAME

Returns:

- Standard Result
- Parameters Sent

Example:

```
<REQUEST>
  <TYPE>MODIFYUSERGROUPPERMISSION</TYPE>
  <REQUESTID>2</REQUESTID>
  <ACTION>ADD</ACTION>
  <GROUPNAME>Sales & Marketing</GROUPNAME>
  <USERNAME>qasupervisor</USERNAME>
</REQUEST>
```

CC:

You must pass either ADD or DELETE as the ACTION type

You must supply either GROUPNAME or GROUPID, but you do not have to pass both

The system will first look to see if you supplied the textual representation of the group name the corresponding ID for you. In either case the Grouped and the username are then queried to verify that the correct entries exist.

EMAIL

Description: Prompts CallCopy to export the audio recording to the email address provided.

Parameters:

- REQUESTID
- UNIQUEFIELD
- UNIQUEID
- TOADDRESS
- MEDIA_FORMAT [mp3, wav,vox, m4a, or cav]

Returns:

Standard Result

Example:

```
<REQUEST>
<TYPE>EMAIL</TYPE>
<REQUESTID>14</REQUESTID>
<TOADDRESS>gkerber@callcopy.com</TOADDRESS>
<MEDIA_FORMAT>wav</MEDIA_FORMAT>
<UNIQUEID>46</UNIQUEID>
<UNIQUEFIELD>ident</UNIQUEFIELD
</REQUEST>

<RESULT>
<REQUESTTYPE>EMAIL</REQUESTTYPE>
<REQUESTID>14</REQUESTID>
<RESULTTYPE>API_OK</RESULTTYPE>
<RESULTCODE>20</RESULTCODE>
<RESULTMSG>API_OK</RESULTMSG>
</RESULT
```

AGENTWINDOW

Description: Provides latest information on the agent's window provided by screencapture

Parameters:

- REQUESTID
- COMPUTERNAME or
- USERNAME

Returns:

- Standard Result
- With screen information
  - Domain
  - IPADDRESS
  - STATUS
  - APPLICATIONTITLE
  - LASTUPDATE

Example:

```
<REQUEST>
<TYPE>AGENTWINDOW</TYPE>
<REQUESTID>7</REQUESTID>
<COMPUTERNAME>ceddy</COMPUTERNAME>
<USERNAME>ceddy</USERNAME>
</REQUEST>


<RESULT>
<REQUESTTYPE>AGENTWINDOW</REQUESTTYPE>
<REQUESTID>7</REQUESTID>
<RESULTTYPE>API_OK</RESULTTYPE>
<RESULTCODE>20</RESULTCODE>
<RESULTMSG>API_OK</RESULTMSG>
<DOMAIN>callcopy.com</DOMAIN>
<IPADDRESS>10.100.5.91</IPADDRESS>
<STATUS>AVAILABLE</STATUS>
<APPLICATIONTITLE>Internet Explorer</APPLICATIONTITLE>
<LASTUPDATE>Feb  2 2011  4:53PM</LASTUPDATE>
</RESULT>
```

Chapter 8: Results

The following section possible results that you will receive from the messaging

All messages will reply with a minimum of a root tag called RESULT containing the REQUESTID that was sent, as well as a RESULTTYPE, RESULTCODE and a RESULTMSG.

```
<RESULT>
  <REQUESTID>1</REQUESTID>
  <RESULTTYPE>OK</RESULTTYPE>
  <RESULTCODE>20</RESULTCODE>
  <RESULTMSG>OK</RESULTMSG>
</RESULT>
```

If the result is an error condition, the RESULTTYPE will be ERROR, the RESULTCODE will contain a code for the error, and the RESULTMSG may contain more detail about the error.

```
<RESULT>
  <REQUESTID>1</REQUESTID>
  <RESULTTYPE>ERROR</RESULTTYPE>
  <RESULTCODE>17</RESULTCODE>
  <RESULTMSG> API_ERROR_NO_DEFAULT_RECORDING_PATH</RESULTMSG>
</RESULT>
```

If the result is a non-error condition, the RESULT node will also contain a sub node containing data specific to the request that was made.

Result Codes

The following are the constants for the RESULTCODE return values.

0	API_RECORDING_STARTED
1	API_ERROR_ALREADY_RECORDING
2	API_ERROR_OUT_OF_DISK_SPACE
3	API_ERROR_NO_FREE_CHANNELS
4	API_ERROR_UNKNOWN
5	API_DEVICE_ALIAS_UNSPECIFIED
6	API_ERROR_SCHEDULER_NOT_RUNNING
7	API_DEVICE_NOT_IN_CALL
8	API_DEVICE_INVALID
9	API_DID_NOT_MATCH_A_SCHEDULE
10	API_UNKNOWN_MESSAGE_TYPE
11	API_ERROR_PROCESSING_MESSAGE
12	API_DEVICE_NOT_RECORDING
13	API_RECORDING_STOPPED
14	API_REQUESTID_INVALID
15	API_INVALID_REQUEST_TYPE
16	API_INVALID_MESSAGE_FORMAT
17	API_INVALID_MESSAGE_FORMAT
18	API_ERROR_NO_DEFAULT_RECORDING_PATH
20	API_OK
21	API_ERROR_INVALID_ACTION
22	API_ERROR_KEYS_NOT_UNIQUE
23	API_DATABASE_QUERY_ERROR
24	API_AGENT_NOT_FOUND
25	API_AGENT_NOT_ACTIVE
26	API_BLANK_REQUIRED_VALUE
27	API_NO_GROUP
28	API_USER_NOT_FOUND
29	API_MODE_INVALID
30	API_EVENT_TYPE_INVALID
31	API_ERROR_CLIENT_NOT_FOUND
32	API_ERROR_ALREADY_MONITORING
33	API_SESSIONID_UNSPECIFIED
34	API_ERROR_RECORDER_NOT_RUNNING
35	API_ERROR_BLOCKED_RECORDING_FILTER
36	API_ERROR_LOGGER_NOT_RUNNING
37	API_RECORDING_UPDATED
38	API_BLACKOUT_INVALID_STATE
39	API_UNLICENSED
40	API_RECORDID_INVALID
41	API_FILE_NOT_FOUND

Chapter 9: Parameter Definitions

The following section contains definitions of the allowed parameters above, as well as the data types that they may contain.

ALLOWSELFRECORD - Permission given to an agent allowing them to record their own calls using the CallCopy Agent Recorder. Possible values (Y/[N])

ALLOWSELFDOWNLOAD – Permission given to an agent allowing them to download their recorded calls recorded through using the CallCopy Agent Recorder. Possible values (Y/[N])

ACTION – Action associated with the request type. See the individual notes for the request type.

AGENTID – The identifier assigned to an agent by CallCopy. This ID is the internal system identifier used by CallCopy to associate agents to phone Ids, Calls, etc.

AGENTINITIATED – Did the agent initiate the recoding? Possible values (Y/[N])

ANI – (Automatic Number Identification). The number of the person calling the device. Caller ID

CALLDIRECTION – Denoted whether the call was inbound or outbound ([I]/O)

CALLDISCRIMINATOR – Differentiates between multiple calls on the same device.

USECTICALLINFO – When set to 'Y', the callstart message will use cached CTI information for fields not being passed in. (Group, Gate, ANI, DNIS, User1-15, CallDirection)

DEVICEID – A number denoting the physical number identifying a phone. This is usually the phone extension. In a free seating environment, this is the voice port or the DN of the phone. In other words this is the fixed number identifying a device

DEVICEALIAS – A number denoting the logical number identifying a phone. In a free seating environment, this is usually the sign on number for the agent. This number is the variable number for a phone.

DIALTOMONITOR – A string that may be passed to the system to specify exactly what the system must dial to the phone switch to be able to monitor the device. This is usually a service observe code used in a T1/E1 based recording environment (Optional)

DNIS - (Dialed Number Identification Service). Typically the number that the caller dialed to get to the device that took the call. This can also be a code that the carrier assigns to the call to identify it as it comes in over the circuits.

EMPLOYEEID – An optional employee ID that you can assign to an agent. This value is used for reporting only and is not used internally by CallCopy

FIRSTNAME – First name (or given name) of an agent.

GATE – An identifier for the call type. This is also known as the skill set, vector, or application.

GROUP – The ACD group that the device belongs to

KEEPDAYS – The number of days to store a record and associated files before purging.

LASTNAME – Last name (surname or family name) of an agent.

MAXRECORDSILENCE – A time in seconds that the call may have silence before recording is ended. This option is only available on certain hardware types.

MAXRECORDTIME – A time in seconds specifying how long the call may last before recording is automatically ended. This limits the amount of time a recording will continue for a single call. Uses can be to automatically stop recording on a long call, or possibly a call that the stop message was not received for to ensure the disk space is not eaten up.

PRIORITY – A number from 0-100 denoting the priority of the call for recording purposes.

RECORDINGPATH – The system variable value specifying the default path for recordings

RECORDINGSPACEINFO – A description of the amount of free disk space on the recording drive,

Example Value: 22.35GB of 37.21GB

RECORDINGTOTALDISKCAPACITY – The size of the disk specified on the recording path

RECORDINGTOTALDISKFREE – The amount of free space on the disk specified on the recording path

RECORDINGTOTALDISKFREEPCT – The percent of free disk space specified on the recording path

REQUESTID- A number identifying a request. This number will be repeated back in the result message as the request ID. This number must be a positive integer value and does not necessarily need to be unique. 2147483647 is the maximum REQUESTID that can be sent to the API. The API will send API_REQUESTID_INVALID as an return error if the value is higher.

STATE- An integer value identifying the line state of a recording channel. Possible values:

0 – On Hook

1 – Off Hook

2 – Recording

3 – Playback

4 – Waiting

5 – Error

6 – Resetting

7 – Recording Prep

8 – Finish Recording

9 – Dialing Fort Playback

STATEDESC- A description of the line state

STATIONNAME- This field can simultaneously serve two purposes:

STATIONNAME may be passed as the required item in place of DEVICEID in CALLSTART. If DEVICEID is not passed or is passed as a blank value, CallCopy will look up the DEVICEID based on the value from the COMPUTERNAME field.

You may also pass the STATIONNAME to override the station mapping that is stored in the CallCopy database for the DEVICEID. This allows you to define your station mappings externally without having to store them in the local CallCopy system database.

SYSTEMPROCESSORSPEED – A measured system processor speed-reading in MHz

SYSTEMPROCESSORTYPE – The description for the system processor that is stored in the system registry

SYSTEMPROCESSORSPEEDREG- The processor speed that is stored in the system registry

SYSTEMRAMTOTAL – Total amount of physical RAM in the system in megabytes

SYSTEMRAMUSED - Total amount of physical RAM used in megabytes

SYSTEMRAMFREE - Total amount of physical RAM free in megabytes

SYSTEMRAMUSEDPCT – Percent of physical RAM used

SYSTEMUPTIME – Uptime for the system in days, hours, minutes and seconds

SYSUSERNAME – This value is used to associate a user to an agent. This value should equal the username (of your network, not the CallCopy username) of the agent. The value is used by the CallCopy Agent Recorder to associate a network user to an agent defined in the CallCopy system. This association is used to grant them the proper access rights to the CallCopy Agent Recorder (such as phone Ids that they may record, and download permissions).

UPDATE_IF_NOT_RECORDING – When this parameter is present and set to ‘Y’ the update command will update the last call recorded for the device if the device is not currently being recorded when the update command is called. Possible values (Y/[N])

USER1-15 – Variables that you may use to add your own custom information in to be stored with the call record. These values are stored as ASCII text. 1-3 has a max length of 20 chars. 4 and 5 has a max length of 255 chars.

Chapter 10: Datatype Values

The following are the value types for the CALLSTART, CALLSTOP, CALLUPDATE and STARTRECORDINGAUTO parameters:

REQUESTID	integer (signed 32 bit)
DEVICEID	char(64)
DEVICEALIAS	char(15)
GROUP	char(20)
GATE	char(20)
ANI	char(20)
DNIS	char(20)
USER1	char(20)
USER2	char(20)
USER3	char(20)
USER4	char(255)
USER5	char(255)
USER6	char(255)
USER7	char(255)
USER8	char(50)
USER9	char(50)
USER10	char(50)
USER11	char(50)
USER12	char(50)
USER13	char(50)
USER14	char(50)
USER15	char(50)
CALLID	char(16)
DIALTOMONITOR	char(20)
PRIORITY	integer (signed 32 bit)
CALLDIRECTION	char(1) (‘I’,’O’,’?’)
MAXRECORDSILENCE	integer (signed 32 bit)
MAXRECORDTIME	integer (signed 32 bit)
AGENTINITIATED	char(1) (‘Y’,’N’)
CTIINITIATED	char(1) (’Y’,’N’)
STATIONNAME	char(50)
OVERRIDEDISKLOCATION	char(150)
UPDATE_IF_NOT_RECORDING	char(1) (‘Y’,’N’)
USECTICALLINFO	char(1) (‘Y’,’N’)

The following are the value types for the MODECHANGE parameters:

REQUESTID	integer (signed 32 bit)
DEVICEID	char(64)
DEVICEALIAS	char(15)
MODE	char(255) values: MODE_SIGNEDON, MODE_SIGNEDOFF, MODE_READY, MODE_NOTREADY, MODE_INCALL, MODE_CALLEND, MODE_LUNCH, MODE_BREAK, MODE_WRAP, MODE_OTHER

The following are the value types for the SYSTEMSTATUS parameters:

REQUESTID	integer (signed 32 bit)
RECORDINGPATH	string
RECORDINGSPACEINFO	string
RECORDINGTOTALDISKCAPACITY	float
RECORDINGTOTALDISKFREE	float
RECORDINGTOTALDISKFREEPCT	float (%2f)
SYSTEMPROCESSORSPEED	string
SYSTEMPROCESSORTYPE	string
SYSTEMPROCESSORSPEEDREG	string
SYSTEMRAMTOTAL	integer (signed 64 bit)
SYSTEMRAMFREE	integer (signed 64 bit)
SYSTEMRAMUSEDPCT	integer (unsigned 32 bit)
SYSTEMUPTIME	string

The following are the value types for the DEVICELIST and DEVICESTATUS parameters:

REQUESTID	integer (signed 32 bit)
DEVICEID	char(64)
DEVICEALIAS	char(15)
SYS_USER	char(30)
CALLDISCRIMINATOR	char(64)
CRM_USER	char(30)
GROUP	char(20)
GATE	char(20)
ANI	char(20)
DNIS	char(20)
USER1	char(20)
USER2	char(20)
USER3	char(20)
USER4	char(255)
USER5	char(255)
USER6	char(255)
USER7	char(255)
USER8	char(50)
USER9	char(50)
USER10	char(50)
USER11	char(50)
USER12	char(50)
USER13	char(50)
USER14	char(50)
USER15	char(50)
CALLDIRECTION	char(1) ('I','O','?')
MODE	integer (unsigned 32-bit) values: (0..9) = (SignedOn, SignedOff, Ready, NotReady, InCall, CallEnd, Lunch, Break, Other, Wrap)
MODESTR	char(255)
ISTRUNK	boolean
STATION	char(128)
LASTMODECHANGE	char dateformat "yyyy-mm-dd hh:nn:ss"
ISRECORDING	boolean
FILENAME	char(255)
CALLID	char(30)

The following are the value types for the CHANNELS parameters:

REQUESTID	integer (signed 32 bit)
ID	integer (signed 32 bit)
STATE	string

PLAYBACKEXTENSION	char(15)
RECORDINGDEVICEALIAS	char(15)
RECORDINGDEVICE	char(64)
LASTSTATECHANGE	string
LASTSTATECHANGEUNIXTIME	string

The following are the value types for the STATIONS parameters:

REQUESTID	integer (signed 32 bit)
COUNT	integer (signed 32 bit)
STATIONNAME	char(50)
DEVICEID	char(64)

The following are the value types for the CHATSTART, CHATSTOP, and CHATUPDATE parameters:

REQUESTID	integer (signed 32 bit)
SESSIONID	char(255)
KEYNAME	string values USERNAME, EMPLOYEEID
KEYVALUE	char(255)
DEVICEID	char(64)
DEVICEALIAS	char(15)
GROUP	char(20)
GATE	char(20)
ANI	char(20)
DNIS	char(20)
USER1	char(20)
USER2	char(20)
USER3	char(20)
USER4	char(255)
USER5	char(255)

CALLDIRECTION	char(1) (‘I’,’O’,’?’)
CALLID	char(16)

The following are the value types for the BLACKOUTSTART and BLACKOUTSTOP parameters:

REQUESTID	integer (signed 32 bit)
DEVICEID	char(64)
DEVICEALIAS	char(15)
CRM_USER	char(30)
SYS_USER	char(30)

The following are the value types for the IMPORTAGENT and IMPORTAGENTPHONE parameters:

REQUESTID	integer (signed 32 bit)

ACTION	string values ADD, DELETE, MODIFY
AGENTID	integer (signed 32 bit)
FIRSTNAME	char(20)
LASTNAME	char(30)
SYSUSERNAME	char(30)
DOMAIN	char(30)
EMAIL	char(255)
EMPLOYEEID	char(20)
ALLOWSELFRECORD	char(1) ('Y','N')
ALLOWSELFDOWNLOAD	char(1) ('Y','N')
PHONEID	char(10)
DEVICEID	char(15)
STATIONNAME	char(50)

The following are the value types for the ADDGROUP, MODIFYGROUPPERMISSION, and MODIFYUSERGROUPPERMISSION parameters:

REQUESTID	integer (signed 32 bit)
ACTION	string values ADD, MODIFY
GROUPNAME	char(255)
GROUPID	integer (signed 32 bit)
DEVICEID	char(64)
USERNAME	char(16)
PHONEID	char(10)
FIRSTNAME	char(20)
LASTNAME	char(30)
AGENTID	integer (signed 32 bit)

The following are the value types for the EXTENSIONPLAYBACKSTART and EXTENSIONPLAYBACKSTOP parameters:

REQUESTID	integer (signed 32 bit)
DEVICE	char(25)
BARGECODE	char(25)
CRM_USER	char(30)
SYS_USER	char(30)
UNIQUE_FIELD	string
UNIQUE_ID	string
FILENAME	char(200)

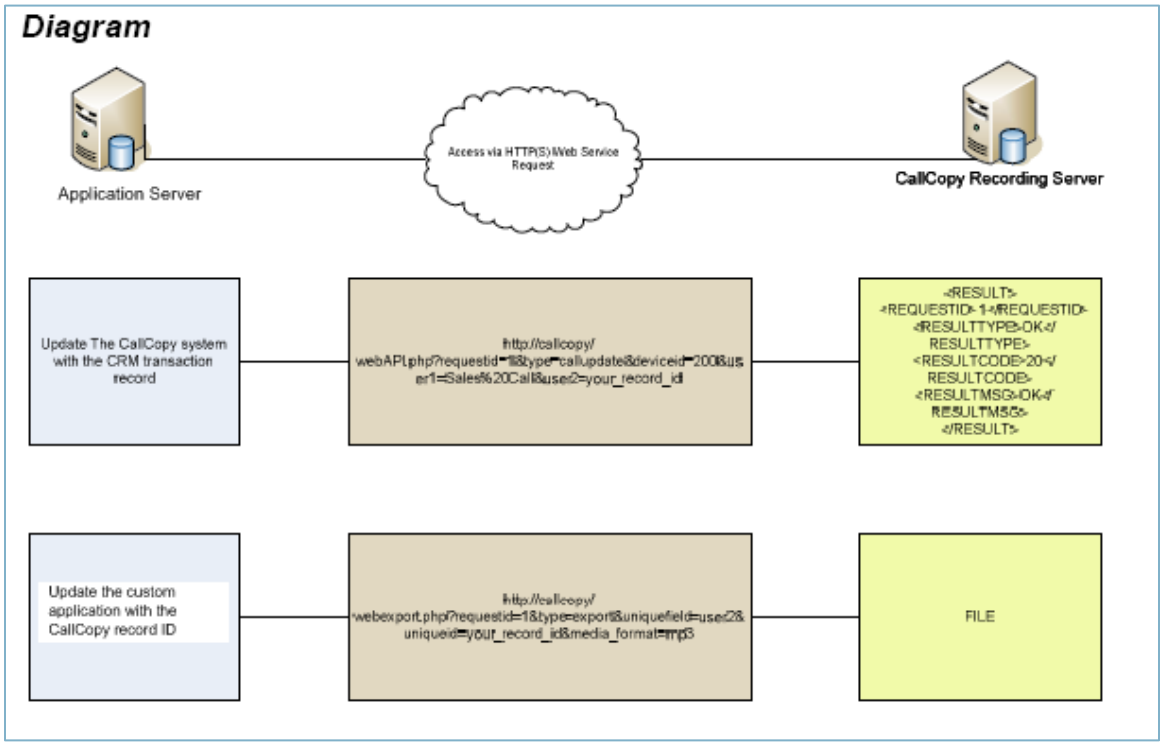
Chapter 11: CallCopy WebAPI Service

General Architecture

The CallCopy Web API Service runs on the CallCopy server via a custom HTTP listener. Proper network firewall, security, and access measures will need to be in place to allow a custom or third-party application to access the CallCopy server.

The WebAPI service accepts specially formatted URLs that contain API data and translates the URL into a CallCopy API function call which is then executed by the API Server. This allows the same API functionality as the API socket service provider for users more familiar with or applications only capable of making HTTP requests.

An example of the WebAPI service calls are demonstrated in the diagram below.



The WebAPI service is a built in component on every cc:Discover system. The service has custom configuration options that must be set, and all calls to the service must follow the syntax that is accepted by the WebAPI.

TCP Port Settings

By default, the web service will run on port 2012, or optionally 2013 if SSL is used. The port settings in use by the API module can be modified from the defaults if needed. See the API Server settings section of the *CallCopy Administration Manual* for more information on API Server configuration settings.

To call the WebAPI properly, you must pass the port number the API Server is listening on in the URL. The port number syntax is highlighted in the examples below

Example: `http://callcopyserveraddress:2012/webAPI.aspx`

Or for SSL enabled connections:

Example: `https://callcopyserveraddress:2012/webAPI.aspx`

WebAPI Call Syntax

To use the WebAPI functions, requests are passed to a specially formatted URL that is accepted by the WebAPI service. An example of a WebAPI request call is below:

```
http://ccserveraddress:2012/webAPI.aspx?type=callupdate&requestid=18213567&devi
ceid=200&user1=SpecialCall&user2=10121968&keepdays=3650
```

Each request call must be composed of the following components:

- **Protocol Identifier:** This determines if the call will be sent as plain text or utilize SSL encryption. For plain text the value will be set to 'HTTP' . For SSL encryption the value will be 'HTTPS'
- **Host Address:** This is the hostname or IP Address of the server running the CallCopy API Service.
- **TCP Port:** This is the port number the API Service is listening for web requests. 2012 is the default value.
- **API Page:** Calling this page allows an application to access any functions in the CallCopy API, or allows an application access to download and/or retrieve recorded audio files for any purpose, be it immediate playback or long term storage. Value can be either **/webAPI.php** or **/webAPI.aspx**.
- **Function Type:** This indicates which API function you are passing to the API Service.
- **Function Parameters:** These are the required and optional parameters that need to be supplied for the specified API function. These parameters are defined for each function call type in the function chapters of this manual. All API functions are available via the WebAPI method.

To make a successful WebAPI call, the API function parameters need to be passed to the WebAPI page. The parameters are passed using the *question mark* '**?**' character, and each parameter is delimited by the *ampersand* '**&**' character. The desired value for each API parameter is set using an equals '**=**' sign, followed directly with the desired value for that parameter.

Example WebAPI Calls

Updating a Call Record

This example request will update/mark the record with the listed data using the CALLUPDATE function. This function must be passed while the call is being actively recorded, and must follow all requirements defined in the CALLUPDATE function description listed in Chapter 2 of this manual.

The WebAPI call should be formatted in the following manner:

```
http://ccserveraddress:2012/webAPI.aspx?type=callupdate&requestid=18213567&deviceid=200&user1=SpecialCall&user2=10121968&keepdays=3650
```

This example URL was constructed from the following components:

- Protocol: http
- Host Address: ccserveraddress
- TCP Port: 2012
- API Page: WebAPI.aspx
- Function Type: callupdate
- Parameters:
  - RequestID: A unique identifier for the transaction generated by the user making the WebAPI call. The value must be an integer.
    - Value: 18213567
  - Deviceid *(required)*: The phone extension for the call that is being recorded.
    - Value: 200
  - User1: Custom defined field *(in this example, used to display a text phrase)*
    - Value: SpecialCall
  - User2: Custom defined field *(in this example, a transaction ID number)*
    - Value: 10121968
  - Keepdays: The number of days this record will be maintained before being purged.
    - Value: 3650

Exporting a recorded audio file

A function unique to the WebAPI is the **Export** call. This function allows the recorded audio from a call to be exported from the CallCopy system at any time after the recording has completed.

The Export function supports audio exporting only, in either WAV (PCM) format or MP3.

The Export function has three required parameters, **uniquefield**, **uniqueid**, and **media_format**. These parameters are used for identifying the desired audio file for exporting.

- **Uniquefield:** This is the parameter name that contains the identifying information for the record. This can be any parameter inserted from a Call Handling API function, or any parameter returned from a DEVICELIST or DEVICESTATUS API function. The parameter used must contain data that is unique to an individual record. Any parameters that will not contain unique data *(such as deviceid or devicealias)* will return the most recent record that contains matching data.
- **Uniqueid:** This is the value that will be contained in the 'uniquefield' parameter that uniquely identifies the record.
- **Media_format:** This is the file format of the exported audio file. Can be set to 'wav' for PCM WAV audio, or 'mp3' for MP3 audio format also m4a, cav, vox

The WebAPI call should be formatted in the following manner:

```
http://ccserveraddress:2012/webAPI.aspx?type=export&requestid=10482345&uniquefi
eld=user2&uniqueid=10121968&media_format=mp3
```

This example URL was constructed from the following components:

- Protocol: http
- Host Address: ccserveraddress
- TCP Port: 2012
- API Page: WebAPI.aspx
- Function Type: export
- Parameters:
  - RequestID: A unique identifier for the transaction generated by the user making the WebAPI call. The value must be an integer.
    - Value: 10482345
  - uniquefield *(required)*: the parameter name that contains the identifying information for the record. *(in this example, a user generated transaction ID)*
    - Value: user2
  - uniqueid: The value that will be contained in the ‘uniquefield’ parameter for the desired record.
    - Value: 10121968
  - Media_format: This is the file format of the exported audio file.
    - Value: mp3

Chapter 12: Event Interface

CallCopy can also broadcast events about known devices. Other applications can subscribe to those events to watch for state changes.

The Event Proxy is a standard blocking TCP server listening on port 5620. Usage is as simple as connecting to the port and sending a well-formatted XML request. See the API Server Settings section of the *cc: Discover Administration Manual* for more information on API Server configuration settings.

Messages that are sent should end with a carriage return / line feed character (\r\n or ASCII character #13 [Hex $A]) to signal the end of data transmission

Request Result Schema

```
<RESULT>
  <REQUESTTYPE>EVENTS</REQUESTTYPE>
  <REQUESTID>0</REQUESTID>
  <RESULTTYPE>OK</RESULTTYPE>
  <RESULTCODE>20</RESULTCODE>
  <RESULTMSG>OK</RESULTMSG>
  <EVENTXREF>0</EVENTXREF>
  <EVENTCLASS>ALL</EVENTCLASS>
  <EVENTTYPE></EVENTTYPE>
</RESULT>
```

CALLSTART Event

The CALLSTART event signals that CallCopy has received a call start notification from a CTI or API source.

```
<CALLCOPYEVENT>
<EVENTCLASS>CALL</EVENTCLASS>
<EVENTTYPE>CALLSTART</EVENTTYPE>
<MODULE>Api</MODULE>
<EVENTXREF>0</EVENTXREF>
<EVENTDATA>
      <GLOBALCALLID></GLOBALCALLID>
      <DEVICEID>1111</DEVICEID>
      <DEVICEALIAS></DEVICEALIAS>
      <SYS_USER></SYS_USER>
      <CRM_USER></CRM_USER>
      <GATE></GATE>
      <ANI></ANI>
      <DNIS></DNIS>
      <AGENTID>16</AGENTID>
      <USER1></USER1>
      <USER2></USER2>
      <USER3></USER3>
      <USER4></USER4>
      <USER5></USER5>
      <USER6></USER6>
```

```
        <USER7></USER7>
        <USER8></USER8>
        <USER9></USER9>
        <USER10></USER10>
        <USER11></USER11>
        <USER12></USER12>
        <USER13></USER13>
        <USER14></USER14>
        <USER15></USER15>
        <GROUP></GROUP>
        <CALLINSTANCE></CALLINSTANCE>
        <CALLDIRECTION>I</CALLDIRECTION>
        <STATION></STATION>
        <ISTRUNK></ISTRUNK>
        <KEEPDAYS>30</KEEPDAYS>
        <ARCHIVEACTION>-1</ARCHIVEACTION>
        <STARTTIME>1/31/2011 2:39:49 PM</STARTTIME>
        <COREIDENT>1</COREIDENT>
</EVENTDATA>
</CALLCOPYEVENT>
```

CALLSTOP Event

The CALLSTOP event signals that CallCopy has received a call stop or call end notification from a CTI or API source.

```
<CALLCOPYEVENT>
<EVENTCLASS>CALL</EVENTCLASS>
<EVENTTYPE>CALLSTOP</EVENTTYPE>
<MODULE>Api</MODULE>
<EVENTXREF>0</EVENTXREF>
<EVENTDATA>
        <DEVICEALIAS></DEVICEALIAS>
        <SYS_USER></SYS_USER>
        <CRM_USER></CRM_USER>
        <GATE></GATE>
        <ANI></ANI>
        <DNIS></DNIS>
        <AGENTID>16</AGENTID>
        <USER1></USER1>
        <USER2></USER2>
        <USER3></USER3>
        <USER4></USER4>
        <USER5></USER5>
        <USER6></USER6>
        <USER7></USER7>
        <USER8></USER8>
        <USER9></USER9>
        <USER10></USER10>
        <CALLINSTANCE></CALLINSTANCE>
        <USER11></USER11>
        <USER12></USER12>
        <USER13></USER13>
        <USER14></USER14>
        <USER15></USER15>
        <GROUP></GROUP>
        <KEEPDAYS>-1</KEEPDAYS>
        <ARCHIVEACTION>-1</ARCHIVEACTION>
        <GLOBALCALLID></GLOBALCALLID>
        <DEVICEID>1111</DEVICEID>
        <COREIDENT>1</COREIDENT>
```

```
</EVENTDATA>
</CALLCOPYEVENT>
```

RECORDINGSTARTED Event

The RECORDINGSTARTED event signals that CallCopy is recording a call.

```
<CALLCOPYEVENT>
<EVENTCLASS>RECORD</EVENTCLASS>
<EVENTTYPE>RECORDINGSTARTED</EVENTTYPE>
<MODULE>Api</MODULE>
<EVENTXREF>0</EVENTXREF>
<EVENTDATA>
	<DEVICEID>1111</DEVICEID>
	<FILENAME>C:\Recordings\20110131\1111\1111-14-39-50.au</FILENAME>
	<CHANNEL>1</CHANNEL>
	<ERRORREASON>NONE</ERRORREASON>
	<GLOBALCALLID></GLOBALCALLID>
	<DEVICEALIAS></DEVICEALIAS>
	<GATE></GATE>
	<ANI></ANI>
	<DNIS></DNIS>
	<AGENTID>16</AGENTID>
	<USER1></USER1>
	<USER2></USER2>
	<USER3></USER3>
	<USER4></USER4>
	<USER5></USER5>
	<USER6></USER6>
	<USER7></USER7>
	<USER8></USER8>
	<USER9></USER9>
	<USER10></USER10>
	<USER11></USER11>
	<USER12></USER12>
	<USER13></USER13>
	<CALLINSTANCE></CALLINSTANCE>
	<USER14></USER14>
	<USER15></USER15>
	<GROUP></GROUP>
	<KEEPDAYS>30</KEEPDAYS>
	<ARCHIVEACTION>-1</ARCHIVEACTION>
	<COREIDENT>1</COREIDENT>
</EVENTDATA>
</CALLCOPYEVENT>
```

RECORDINGSTOPPED Event

The RECORDINGSTOPPED event signals that CallCopy has stopped recording a call

```
<CALLCOPYEVENT>
<EVENTCLASS>RECORD</EVENTCLASS>
<EVENTTYPE>RECORDINGSTOPPED</EVENTTYPE>
<MODULE></MODULE>
<EVENTXREF>0</EVENTXREF>
<EVENTDATA>
	<GLOBALCALLID>000100000000092</GLOBALCALLID>
	<DEVICEID>1111</DEVICEID>
	<DEVICEALIAS></DEVICEALIAS>
	<GATE></GATE>
```

```
        <ANI></ANI>
        <DNIS></DNIS>
        <AGENTID>16</AGENTID>
        <USER1></USER1>
        <USER2></USER2>
        <USER3></USER3>
        <USER4></USER4>
        <USER5></USER5>
        <KEEPDAYS>30</KEEPDAYS>
        <RECORDID>13101</RECORDID>
        <COREIDENT>1</COREIDENT>
</EVENTDATA>
</CALLCOPYEVENT>
```

RECORDINGUPDATE Event

The RECORDINGUPDATE event signals that CallCopy has saved a recording to the database.

```
<CALLCOPYEVENT>
<EVENTCLASS>RECORD</EVENTCLASS>
<EVENTTYPE>RECORDINGUPDATE</EVENTTYPE>
<MODULE></MODULE>
<EVENTXREF>0</EVENTXREF>
<EVENTDATA>
        <GLOBALCALLID>000100000000092</GLOBALCALLID>
        <DEVICEID>1111</DEVICEID>
        <KEEPDAYS>30</KEEPDAYS>
        <FILENAME>C:\Recordings\20110131\1111\1111-14-39-50.au</FILENAME>
        <RECORDID>13101</RECORDID>
        <COREIDENT>1</COREIDENT>
</EVENTDATA>
</CALLCOPYEVENT>
```

About CallCopy

CallCopy, a leading provider of innovative call recording and contact center solutions, is dedicated to ensuring the highest standards of customer and employee satisfaction. The award-winning, enterprise-proven cc: Discover suite delivers advanced call recording, screen capture, quality management, speech analytics, performance management, customer survey and workforce management capabilities to organizations of all sizes and industries across the globe.

CallCopy empowers these organizations to gather business intelligence, which is leveraged to maximize operational performance, reduce liability, achieve regulatory compliance and increase customer satisfaction.

For more information, visit www.callcopy.com.