CallCopy® | Innovations in Call Recording and Contact Center Solutions

CallCopy API Manual, v5.2 R1

June 2013

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	Updated entire manual to match 5.2 functionality of API server.	MBuckingham	2013-05-31
1	Updated DEVICESTATUS description based on new system behavior.	MBuckingham	2013-06-24

Table of Contents

Chapter 1: Introduction

The CallCopy API lets you add custom data to cc: Discover records and integrate third-party applications and cc: Discover. For example, you may want to store customer specific information with each call record, or perhaps you would like to specify when exactly to start and stop call recording based on specific events.

Message Structure

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.

> **Note** The CallCopy API Server 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.
>
> **Note** 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. (REQUESTID is required in every request message.)

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 API uses the DEVICEID to track phone states. This is true even in a free seating environment.

New Functions

No additional functions were added in this version of the CallCopy API Server.

Deprecated Functions

No additional functions were deprecated in this version of the CallCopy API Server. If a previously deprecated function is used, an error message will be returned stating the function is not supported.

Chapter 2: Call 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.

Parameters:

- REQUESTID (required)
- DEVICEID *****
- CALLINSTANCE
- DEVICEALIAS *** +**
- GROUP
- GATE
- ANI
- DNIS
- USER1 – USER15
- CALLID
- PRIORITY
- CALLDIRECTION
- MAXRECORDSILENCE
- MAXRECORDTIME
- AGENTINITIATED
- CTIINITIATED
- ARCHIVEACTION – This requires an archive action created in cc: Discover.
- STATIONNAME *****
- SYS_USER *** +**
- CRM_USER *** +**

At least one of the parameters marked with an asterisk (*) is required for the request to be executed.

Parameters marked with a plus sign (+) can be used only 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>
```

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

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 call recording schedule 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 and return a RESULTTYPE of API_DEVICE_INVALID.

Parameters:

- REQUESTID - Required
- DEVICEID *****
- CALLINSTANCE
- CRM_USER ***+**
- SYS_USER ***+**
- DEVICEALIAS ***+**
- STATIONNAME *****
- GROUP
- GATE
- ANI
- DNIS
- USER1 – USER15 ******
- CALLDIRECTION
- ARCHIVEACTION – This must an archive action created in cc: Discover.
- 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.
****** Using angle brackets (< >), some special characters, and symbols in these fields may not work correctly. Test all possible characters needed before using this function.

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>
```

Note 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 has finished recording. This information is used to update the CallCopy database. This information **will not** be used by the scheduling or archive systems because it is assumed that the call is completed at this point. By passing a RecordID parameter, it is assumed the call update is for a call post recording.

Parameters:

- REQUESTID - Required
- RECORDID - Required
- DEVICEID
- DEVICEALIAS
- GROUP
- GATE
- ANI
- DNIS
- USER1 – USER15 ******
- CALLDIRECTION

****** Using angle brackets (< >), some special characters, and symbols in these fields may not work correctly. Test all possible characters needed before using this function.

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>
```

RECORDSTART

Description: Allows a user to start an additional, partial recording for a call that is already being recorded.

Parameters:

- REQUESTID (required)
- DEVICEID *****
- CALLINSTANCE
- DEVICEALIAS *** +**
- GROUP
- GATE
- ANI
- DNIS
- CALLID
- PRIORITY
- CALLDIRECTION
- MAXRECORDSILENCE
- MAXRECORDTIME
- AGENTINITIATED
- CTIINITIATED
- ARCHIVEACTION – This must an archive action created in cc: Discover.
- STATIONNAME *****
- SYS_USER *** +**
- CRM_USER *** +**
- USER1-USER15 ******

At least one of the parameters marked with an * is required for the request to be executed.

Parameters marked with a plus sign (+) can be used only 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>RECORDSTART</REQUESTTYPE>
  <REQUESTID>1</REQUESTID>
  <RESULTTYPE>API_DEVICE_INVALID</RESULTTYPE>
  <RESULTCODE>8</RESULTCODE>
  <RESULTMSG>Could not find a device matching given parameters</RESULTMSG>
</RESULT>
```

Example Request

```
<REQUEST>
<TYPE>RECORDSTART</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>
```

Example Result

```
<RESULT>
<REQUESTTYPE>RECORDSTART</REQUESTTYPE>
<REQUESTID>2060</REQUESTID>
<RESULTTYPE>API_OK</RESULTTYPE>
<RESULTCODE>20</RESULTCODE>
<RESULTMSG>RECORDSTART sent to all of the following cores: 1</RESULTMSG>
</RESULT>
```

RECORDSTOP

Description: Used with RECORDSTART. 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 *****
- USER1-USER15**

***** 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.
**See parameter definitions.

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

```
<REQUEST>
  <TYPE>CALLSTOP</TYPE>
  <REQUESTID>2</REQUESTID>
  <DEVICEID>555</DEVICEID>
  <USER1>test</USER1>
  <USER3>1</USER3>
</REQUEST>
```

Example Result

```
<RESULT>
  <REQUESTTYPE>RECORDSTOP</REQUESTTYPE>
  <REQUESTID>2060</REQUESTID>
  <RESULTTYPE>API_OK</RESULTTYPE>
  <RESULTCODE>20</RESULTCODE>
  <RESULTMSG>RECORDSTOP sent to all of the following cores: 1</RESULTMSG>
</RESULT>
```

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. This function returns only one instance on a device. See DEVICERECORDINGSTATUS.

Parameters:

- DEVICELIST
- REQUESTID

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

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

Example Result:

```
<RESULT>
  <REQUESTTYPE>DEVICELIST</REQUESTTYPE>
  <REQUESTID>1</REQUESTID>
  <RESULTTYPE>API_OK</RESULTTYPE>
  <RESULTCODE>20</RESULTCODE>
  <RESULTMSG>API_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 current information pertaining to a specific device or devices. The system prioritizes the call instances based on which one is active or was most recently active. If the device is not currently on a call, the last call it was on will be returned. This function returns ONLY the highest priority (most recently active) instance for a device. For all instances, see DEVICERECORDINGSTATUS.

Parameters:

- REQUESTID
- DEVICEID
- DEVICEALIAS
- CRM_USER
- SYS_USER

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

- DEVICEID
- DEVICEALIAS
- 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:

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

<RESULT>

  <REQUESTTYPE>DEVICESTATUS</REQUESTTYPE>
  <REQUESTID>24</REQUESTID>
  <RESULTTYPE>API_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>
```

DEVICERECORDINGSTATUS

Description: Queries the system for a list of all instances of a device and displays status (i.e., recording or recording pause).

Parameters:

- REQUESTID
- DEVICEID
- DEVICEALIAS
- CRM_USER
- SYS_USER

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

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

Example:

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

```
<RESULT>
  <REQUESTTYPE>DEVICERECORDINGSTATUS</REQUESTTYPE>
  <REQUESTID>2</REQUESTID>
  <RESULTTYPE>API_OK</RESULTTYPE>
  <RESULTCODE>20</RESULTCODE>
  <RESULTMSG>Following active call instances found for specified deviceID,
devicealias, crm_user or sys_user</RESULTMSG>
<DEVICE>
  <DEVICEID>7008</DEVICEID>
  <DEVICEALIAS></DEVICEALIAS>
  <SYS_USER>vrunda.shah</SYS_USER>
  <CRM_USER></CRM_USER>
  <COREIDENT>8</COREIDENT>
  <RECORDING>
  <GROUP></GROUP>
  <GATE></GATE>
  <ANI>7460</ANI>
  <DNIS>7008</DNIS>
  <USER1></USER1>
  <USER2></USER2>
  <USER3></USER3>
  <USER4></USER4>
  <USER5></USER5>
  <USER6>30</USER6>
  <USER7>1</USER7>
  <USER8>00001000371342197048</USER8>
  <USER9></USER9>
  <USER10></USER10>
  <USER11></USER11>
  <USER12></USER12>
  <USER13></USER13>
  <USER14></USER14>
  <USER15></USER15>
  <CALLDIRECTION>I</CALLDIRECTION>
  <ISTRUNK></ISTRUNK>
  <STATION></STATION>
  <STARTTIME>7/13/2012 12:30:43 PM</STARTTIME>
  <STOPTIME></STOPTIME>
  <ISRECORDING>Y</ISRECORDING>
  <ISRECORDINGVIDEO>N</ISRECORDINGVIDEO>
  <FILENAME>C:\Recordings\20120713\7008\7008-12-30-43.cca</FILENAME>
  <CALLID></CALLID>
  <SESSIONID></SESSIONID>
  <CALLINSTANCE>37</CALLINSTANCE>
  <STATUS>recordpaused</STATUS>
  </RECORDING>
</DEVICE>
<RECORDING>
  <GROUP></GROUP>
  <GATE></GATE>
  <ANI>7008</ANI>
  <DNIS>7007</DNIS>
  <USER1></USER1>
  <USER2></USER2>
```

```
  <USER3></USER3>
  <USER4></USER4>
  <USER5></USER5>
  <USER6></USER6>
  <USER7></USER7>
  <USER8>0000100038134219710 7</USER8>
  <USER9></USER9>
  <USER10></USER10>
  <USER11></USER11>
  <USER12></USER12>
  <USER13></USER13>
  <USER14></USER14>
  <USER15></USER15>
  <CALLDIRECTION>O</CALLDIRECTION>
  <ISTRUNK></ISTRUNK>
  <STATION></STATION>
  <STARTTIME>7/13/2012 12:31:54 PM</STARTTIME>
  <STOPTIME></STOPTIME>
  <ISRECORDING>Y</ISRECORDING>
  <ISRECORDINGVIDEO>N</ISRECORDINGVIDEO>
  <FILENAME>C:\Recordings\20120713\7008\7008-12-31-54.cca</FILENAME>
  <CALLID></CALLID>
  <SESSIONID></SESSIONID>
  <CALLINSTANCE>38</CALLINSTANCE>
  <STATUS>recording</STATUS>
</RECORDING>
</DEVICE>
</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
- SYSTEMRAMUSED
- SYSTEMRAMUSEDPCT

Example:

```
<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

A successful call returns a RESULT section with a COUNT of the number of channels, plus an additional CHANNELS section containing a CHANNEL section per channel on the system containing:

- ID – This is the Channel ID from the CallCopy database. The number of ID sections is the same as the COUNT.
- STATE
- PLAYBACKEXTENSION – No longer supported.
- RECORDINGDEVICEALIAS – No longer supported.
- RECORDINGDEVICE – No longer supported.
- LASTSTATECHANGE
- LASTSTATECHANGEUNIXTIME

Example:

```
<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>1</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>2</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>3</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>4</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

A successful API call returns a RESULT section including a COUNT of the number of stations, plus an additional STATIONS section containing a STATION section per station on the system containing:

- STATIONNAME
- DEVICEID

Example:

```
<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: A RESULT section including a COUNT of the number of TERMINOLOGY sections, 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>
```

CALLLIST

Description: Queries the system for a list of call recordings matching criteria.

Parameters:

- REQUESTID – Required
- DEVICEID
- CRM_USER
- SYS_USER
- DEVICEALIAS
- GROUP
- GATE
- ANI
- DNIS
- USER1 - USER15 **
- CALLCOUNT – Number of calls that will be returned.
- CALLDATESTART - Beginning date for filter, in UNIX time
- CALLDATEEND - End date for filter, in UNIX time

Filters are combined as an "AND" Boolean operation.

> **Note** Depending on which parameters are passed, number of different parameters passed, and number of results that match passed parameters, database performance may be impacted. While CallCopy's recording capability is designed to function normally during database outages, proper consideration must be taken when designing call list queries in order to prevent excessive use of database resources. As an additional note, if both CALLCOUNT and CALLDATESTART/END are queried at the same time, the expected behavior is that CALLCOUNT will limit the query to showing the top N calls within the selected date range.

Returns: RESULT section including all information for the all recording records matching query criteria.

Example Request

```
<REQUEST>
<TYPE>CALLLIST</TYPE>
<REQUESTID>1</REQUESTID>
<DEVICEID>1111</DEVICEID>
<USER3>12345</USER3>
<CALLCOUNT>5</CALLCOUNT>
<CALLDATESTART>1312156800</CALLDATESTART>
<CALLDATEEND>1312934400</CALLDATEEND>
</REQUEST>
```

Example Results

```
<RESULT>
<TYPE>CALLLIST</TYPE>
<REQUESTID>1</REQUESTID>
<SEARCHRESULT>
       <CALLNUMBER>1</CALLNUMBER>
<DEVICEID>1111</DEVICEID>
<DEVICEALIAS>1234</DEVICEALIAS>
<ANI></ANI>
<DNIS></DNIS>
<CALLDIRECTION>I</CALLDIRECTION>
<GROUP></GROUP>
<GATE></GATE>
<USER1></USER1>
<USER2></USER2>
<USER3>12345</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>
<RECORDID>13101</RECORDID>
<RECORDINGTIME>1312921123</RECORDINGTIME>
<DURATION>60</DURATION>
<AGENTFIRSTNAME>John</AGENTFIRSTNAME>
<AGENTLASTNAME>Smith</AGENTLASTNAME>
<CRM_USER></CRM_USER>
<SYS_USER></SYS_USER>
       </SEARCHRESULT>
<SEARCHRESULT>
<CALLNUMBER>2</CALLNUMBER>
<DEVICEID>1111</DEVICEID>
<DEVICEALIAS>1234</DEVICEALIAS>
              ...
       </SEARCHRESULT>
<SEARCHRESULT>
<CALLNUMBER>3</CALLNUMBER>
<DEVICEID>1111</DEVICEID>
<DEVICEALIAS>1234</DEVICEALIAS>
              ...
       </SEARCHRESULT>
<SEARCHRESULT>
```

```
<CALLNUMBER>4</CALLNUMBER>
<DEVICEID>1111</DEVICEID>
<DEVICEALIAS>1234</DEVICEALIAS>
                ...
       </SEARCHRESULT>
<SEARCHRESULT>
<CALLNUMBER>5</CALLNUMBER>
<DEVICEID>1111</DEVICEID>
<DEVICEALIAS>1234</DEVICEALIAS>
                ...
       </SEARCHRESULT>
</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 - Required
- SESSIONID - Required
- AGENTID

If AGENTID is not provided, CallCopy can look up the agent using KEYNAME and KEYVALUE.

- KEYNAME - USERNAME, SYS_USER, EMPLOYEEID, CRM_USER, DEVICEALIAS (Phone ID)
- KEYVALUE - The username or employee ID as dictated in the keyname.

Returns:

A standard result code is returned:

- API_SESSIONID_UNSPECIFIED – If the API doesn't successfully connect to the database.
- API_AGENT_NOT_FOUND – 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 a Core.

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 - Required
- SESSIONID - Required

Returns:

A standard result code is returned:

- 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>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

Description: Used to update the record for a desktop-screen-capture recording session. 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 recording is in progress at this point. Prior to this message, a CHATSTART message must be sent.

Parameters:

- REQUESTID - Required
- SESSIONID - Required
- DEVICEALIAS
- GROUP
- GATE
- ANI
- DNIS
- USER1-USER15 ******
- CALLDIRECTION
- CALLID

Returns:

A standard result code is returned:

- 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/AGENTBLACKOUTSTART

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.

Where BLACKOUT blacks out based on call instance, AGENTBLACKOUT will blackout all call and screen data for a specific time range for the specified agent. AGENTBLACKOUT is preferred over BLACKOUT, and AGENTBLACKOUT must be enabled in the cc_APIServer.ini. This is configured during installation. Refer to the *cc: Discover Installation Guide*'s "API Server" section for more information.

Comment [MB1]: Changed from N to Y for default when AgentBlackout was introduced by Vrunda. Jeff/Matt are meeting sometime next week (week of 7/29) to decide what the default behavior should be going forward. DOC-589

Parameters:

- REQUESTID
- DEVICEID
- DEVICEALIAS
- CRM_USER
- SYS_USER
- TIMEOFFSET
- CALLINSTANCE – This parameter is only used for BLACKOUT, not AGENTBLACKOUT.

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_AGENT_NOT_FOUND – Request needs to find information for either the device or the agent. If neither is found, this error appears.
- 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: #,#,#'
- API_REQUESTID_INVALID – REQUESTID specified is not valid.

BLACKOUTSTOP/AGENTBLACKOUTSTOP

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.

Where BLACKOUT blacks out based on call instance, AGENTBLACKOUT will blackout all call and screen data for a specific time range for the specified agent. AGENTBLACKOUT is preferred over BLACKOUT, and AGENTBLACKOUT must be enabled in the cc_APIServer.ini. This is configured during installation. Refer to the *cc: Discover Installation Guide*'s "API Server" section for more information.

Parameters:

- REQUESTID - Required
- DEVICEID
- DEVICEALIAS
- CRM_USER
- SYS_USER
- TIMEOFFSET
- CALLINSTANCE – This parameter is only used for BLACKOUT, not AGENTBLACKOUT.

Returns: Standard result code.

Example:

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

Errors: Possible error returns are:

- API_AGENT_NOT_FOUND – Request needs to find information for either the device or the agent. If neither is found, this error appears.
- 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:` #,#,#'
- API_REQUESTID_INVALID – Request id specified is not valid.

  Note 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.

  Note 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

EXTENSIONPLAYBACKSTART

This function has been deprecated.

EXTENSIONPLAYBACKSTOP

This function has been deprecated.

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

This function has been deprecated. See IMPORTUSER.

IMPORTUSER

Description: This function allows you to import users and add, modify and delete information pertaining to a user. Once a user is added, you may use IMPORTAGENTPHONE to associate phone IDs to the user. The previous IMPORTAGENT command did not require a password since agents and users were separate entities. In v5.0, agents and users were combined, and this command now requires the password to import user account information. USERNAME is the CallCopy username, while CRM_USER is the corresponding username associated with the user within an integrated system.

Parameters:

- REQUESTID
- ACTION – Add, Delete, Modify, List
- AGENT_ID
- EMPLOYEEID
- FIRSTNAME
- LASTNAME
- STATUS (A – Active, I – Inactive)
- ADD_TIME – Date/time offset
- MOD_TIME – Date/time offset
- SYS_USER
- DOMAIN
- EMAIL
- CRM_USER
- PASSWORD
- SITEID
- USERNAME

Returns:

- Standard Result
- AGENTID

Example:

```
<REQUEST>
  <TYPE>IMPORTUSER</TYPE>
  <REQUESTID>2</REQUESTID>
  <ACTION>ADD</ACTION>
  <AGENT_ID>2</AGENT_ID>
  <EMPLOYEEID>5</EMPLOYEEID>
  <FIRSTNAME>James</FIRSTNAME>
  <LASTNAME>Doe</LASTNAME>
  <STATUS>A</STATUS>
  <ADD_TIME></ADD_TIME>
  <MOD_TIME></MOD_TIME>
  <SYS_USER>doe.1</SYS_USER>
  <DOMAIN>somecompany</DOMAIN>
  <EMAIL>jdoe@somecompany.com</EMAIL>
  <CRM_USER>CRMUsername</CRM_USER>
  <PASSWORD>Password</PASSWORD>
  <SITEID>1</SITEID>
  <USERNAME>CallCopyUsername</USERNAME>
</REQUEST>
```

List Example:

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


<RESULT>
  <REQUESTTYPE>IMPORTUSER</REQUESTTYPE>
  <REQUESTID>5</REQUESTID>
  <RESULTTYPE>API_OK</RESULTTYPE>
  <RESULTCODE>20</RESULTCODE>
  <RESULTMSG>API_OK</RESULTMSG>
  <AGENTS>
    <COUNT>1</COUNT>
    <AGENT>
      <AGENT_ID>2</AGENT_ID>
      <EMPLOYEEID>5</EMPLOYEEID>
      <FIRSTNAME>John</FIRSTNAME>
      <LASTNAME>Doe</LASTNAME>
      <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>
      <DOMAIN>CallCopy</DOMAIN>
      <EMAIL>jdoe@callcopy.com</EMAIL>
      <CRM_USER>john</CRM_USER>
      <PHONES>1123,3343,3234</PHONES>
    </AGENT>
  </AGENTS>
</RESULT>
```

The possible values for ACTION are: ADD, DELETE, MODIFY, 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. Thus, you can 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.

Errors: Possible error returns are:

- API_DATABASE_QUERY_ERROR – Represents a range of possible query problems. Check the API logs for details.
- API_ERROR_INVALID_ACTION – Allowed actions are ADD, MODIFY, DELETE. If the requested action is not one of these, this error comes up.
- API_INVALID_MESSAGE_FORMAT – This message appears if parameters are missing or the request is not formatted correctly.
- API_USER_ALREADY_EXIST – Indicates that a user with the same username already exists.
- API_USER_NOT_FOUND – When modifying or deleting an agent, this may come up if the agent is not found. If it appears as the result of a query error, check the API logs for details.

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.

Errors: Possible error returns are:

- API_AGENT_NOT_FOUND – Request needs to find information for either the device or the agent. If neither is found, this error appears.

IMPORTSTATION

Description: Allows an administrator to add or delete/add station to device mappings.

Parameters:

- REQUESTID
- DEVICEID
- STATIONNAME

Returns:

- Standard Result
- DEVICEID
- STATIONNAME

Example:

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

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.

Errors: Possible error returns are:

- API_DATABASE_QUERY_ERROR – Represents a range of possible query problems. Check the API logs for details.
- API_INVALID_MESSAGE_FORMAT – This message appears if parameters are missing or the request is not formatted correctly.

Chapter 7: User, Group, and Role Functions

These functions allow you to control cc: Discover users, groups, and roles from an external resource.

ADDGROUP

Description: Allows an administrator to add a CallCopy group. An asterisk (*) denotes a required parameter.

Parameters:

- REQUESTID
- GROUPNAME*

Returns:

- Standard Result
- GROUPID
- GROUPNAME

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>
```

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.

Errors: Possible error returns are:

- API_INVALID_MESSAGE_FORMAT – This message appears if parameters are missing or the request is not formatted correctly.

MODIFYGROUPPERMISSION

Description: Allows an administrator to add or remove agents and their device assignments to/from CallCopy 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>
```

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.

Errors: Possible error returns are:

- API_AGENT_NOT_FOUND – Request needs to find information for either the device or the agent. If neither is found, this error appears.
- API_DATABASE_QUERY_ERROR – Represents a range of possible query problems. Check the API logs for details.

MODIFYUSERGROUPPERMISSION

Description: Allows an administrator to add or remove permissions for users to view a CallCopy group and manage data (e.g., call records, evaluations) related to agents in that 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>
```

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.

Errors: Possible error returns are:

- API_DATABASE_QUERY_ERROR – Represents a range of possible query problems. Check the API logs for details.
- API_ERROR_INVALID_ACTION – Allowed actions are ADD, MODIFY, DELETE. If the requested action is not one of these, this error will be displayed.
- API_USER_NOT_FOUND – When modifying or deleting an agent, this may come up if the agent is not found. If it appears as the result of a query error, check the API logs for details.

MODIFYROLE

Description: Allows an administrator to add a CallCopy role.

Parameters:

- REQUESTID
- ACTION (add/modify/delete)
- ROLENAME
- DESCRIPTION

Returns:

- Standard Result
- Parameters Sent

Example:

```
<REQUEST>
  <REQUESTTYPE>MODIFYROLE</REQUESTTYPE>
  <REQUESTID>1</REQUESTID>
  <ACTION>ADD</ACTION>
  <ROLENAME>DefaultRole</ROLENAME>
  <DESCRIPTION>Adding Default Role</DESCRIPTION>
</REQUEST>

<RESULT>
  <REQUESTTYPE>MODIFYROLE</REQUESTTYPE>
  <REQUESTID>45</REQUESTID>
  <RESULTTYPE>API_OK</RESULTTYPE>
  <RESULTCODE>20</RESULTCODE>
  <RESULTMSG>Role Added</RESULTMSG>
</RESULT>
```

You must pass either ADD, MODIFY, or DELETE as the ACTION type.

You must supply REQUESTID.

CallCopy will attempt to add the new role. The ROLENAME and the RESULTCODE will be returned as the result. If a role with the same name already exists, the function will return as such without modifying any data.

Errors: Possible error returns are:

- API_DATABASE_QUERY_ERROR – Represents a range of possible query problems. Check the API logs for details.
- API_ERROR_INVALID_ACTION – Allowed actions are ADD, MODIFY, DELETE. If the requested action is not one of these, this error will be displayed.
- API_INVALID_MESSAGE_FORMAT – This message appears if parameters are missing or the request is not formatted correctly.
- API_ROLE_ALREADY_EXISTS – The role the request is attempting to create already exists.
- API_ROLE_NOT_FOUND – The requested role does not exist or could not be located.

MODIFYROLEGROUPPERMISSION

Description: Used to assign a Group in cc: Discover to a Role.

Parameters:

- REQUESTID
- ROLEID
- ROLENAME
- GROUPNAME
- ACTION (add/delete)

Returns:

- Standard Result
- Parameters Sent

Example:

```
<REQUEST>
  <REQUESTTYPE> MODIFYROLEGROUPPERMISSION </REQUESTTYPE>
  <REQUESTID>1</REQUESTID>
  <ACTION>ADD</ACTION>
  <ROLENAME>DefaultRole</ROLENAME>
  <GROUPNAME>CALLCOPYTESTGROUP</GROUPNAME>
</REQUEST>

<RESULT>
  <REQUESTTYPE>MODIFYROLEGROUPPERMISSION</REQUESTTYPE>
  <REQUESTID>1</REQUESTID>
  <RESULTTYPE>API_OK</RESULTTYPE>
  <RESULTCODE>20</RESULTCODE>
  <RESULTMSG>Group Group01 does not exist or it is not active</RESULTMSG>
</RESULT>
```

You must pass either ADD or DELETE as the ACTION type.

You must supply either ROLEID or ROLENAME, but you do not have to pass both.

You must supply REQUESTID and GROUPNAME.

Errors: Possible error returns are:

- API_DATABASE_QUERY_ERROR – Represents a range of possible query problems. Check the API logs for details.
- API_ERROR_INVALID_ACTION – Allowed actions are ADD, MODIFY, DELETE. If the requested action is not one of these, this error will be displayed.
- API_INVALID_MESSAGE_FORMAT – This message appears if parameters are missing or the request is not formatted correctly.

MODIFYUSERROLE

Description: Used to assign a Role to a user. USERID, LASTNAME, FIRSTNAME, SYSTEMUSERNAME, OR USERNAME (any of these) can be used to associate user info.

Parameters:

- REQUESTID
- ROLEID
- ROLENAME
- USERID
- LASTNAME
- FIRSTNAME
- SYSTEMUSERNAME
- USERNAME
- ACTION (add/delete)

Returns:

- Standard Result
- Parameters Sent

Example:

```
<REQUEST>
  <REQUESTTYPE> MODIFYUSERROLE </REQUESTTYPE>
  <REQUESTID>1</REQUESTID>
  <ACTION>ADD</ACTION>
  <ROLENAME>DefaultRole</ROLENAME>
  <USERNAME>TestUser</USERNAME>
</REQUEST>

<RESULT>
  <REQUESTTYPE>MODIFYUSERROLE</REQUESTTYPE>
  <REQUESTID>1</REQUESTID>
  <RESULTTYPE>API_OK</RESULTTYPE>
  <RESULTCODE>20</RESULTCODE>
  <RESULTMSG>User 10 is assigned role Test</RESULTMSG>
</RESULT>
```

You must supply either ROLEID or ROLENAME, but you do not have to pass both.

Errors: Possible error returns are:

- API_DATABASE_QUERY_ERROR – Represents a range of possible query problems. Check the API logs for details.
- API_ERROR_INVALID_ACTION – Allowed actions are ADD, MODIFY, DELETE. If the requested action is not one of these, this error will be displayed.
- API_INVALID_MESSAGE_FORMAT – This message appears if parameters are missing or the request is not formatted correctly.
- API_ROLE_NOT_FOUND – The requested role does not exist or could not be located.
- API_USER_NOT_FOUND – When modifying or deleting an agent, this message indicates that the agent could not be found. If it appears as the result of a query error, check the API logs for details.

EMAIL

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

Parameters:

- REQUESTID
- UNIQUEFIELD
- UNIQUEID
- TOADDRESS
- MEDIA_FORMAT – MP3, WAV, VOX, M4A, or CAV
- ENCRYPT – Used to encrypt the exported CAV file, requires PASSWORD.
- PASSWORD – Password used to access the encrypted exported 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>
<ENCRYPT>Y</ENCRYPT>
<PASSWORD>test</PASSWORD>
</REQUEST>

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

Errors: Possible error returns are::

- API_EMAIL_NOT_VALID – This means the email address in the request is incorrect or could not be found.
- API_ERROR_CLIENT_NOT_FOUND – Email server is not configured or cannot be found.
- API_FILE_NOT_FOUND – If it cannot find the file you're trying to export and email.
- API_INVALID_MESSAGE_FORMAT – This message appears if parameters are missing or the request is not formatted correctly.

COMBINEDEMAIL

Description: Prompts cc: Discover to export one or more recording files as a single file to the email address provided.

Parameters:

- REQUESTID
- TOADDRESS
- TONAME
- FROMADDRESS
- FROMNAME
- IDENT – Recording ID; Multiple IDs can be separated by a comma.
- UNIQUEID

Returns:

- Standard Result

Example:

```
<REQUEST>
<TYPE>COMBINEDEMAIL</TYPE>
<REQUESTID>1</REQUESTID>
<TOADDRESS>vrunda.shah@callcopy.com</TOADDRESS>
<TONAME>vrunda</TONAME>
<FROMADDRESS>vrunda.shah@callcopy.com</FROMADDRESS>
<FROMNAME>vrunda</FROMNAME>
<IDENT>1509</IDENT>
<UNIQUEID>IDENT</UNIQUEID>
</REQUEST>

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

Errors: Possible error returns are:

- API_EMAIL_NOT_VALID – This means the email address in the request is incorrect or could not be found.
- API_ERROR_CLIENT_NOT_FOUND – Email server is not configured or cannot be found.
- API_FILE_NOT_FOUND – If it cannot find the file you're trying to export and email.
- API_INVALID_MESSAGE_FORMAT – This message appears if parameters are missing or the request is not formatted correctly.

AGENTWINDOW

Description: Provides latest information on the agent's window provided by cc: Screen Capture.

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>
```

Errors: Possible error returns are:

- API_AGENT_NOT_FOUND – Request needs to find information for either the device or the agent. If neither is found, this error appears.
- API_DATABASE_QUERY_ERROR – Represents a range of possible query problems. Check the API logs for details.

Chapter 8: Results

The following section explains 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>API_OK</RESULTTYPE>
  <RESULTCODE>20</RESULTCODE>
  <RESULTMSG>API_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/Result Types

The following are the RESULTCODE return values (i.e., numbers) and the corresponding RESULTTYPE label.

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
42	API_AGENT_ALREADY_EXIST
43	API_STREAMING_STARTED
44	API_COMMAND_NOT_ALLOWED
45	API_EMAIL_NOT_VALID
46	API_PARTIAL_CORE_RESPONSE
47	API_COMMAND_QUEUE_FULL
48	API_TOO_MANY_CLIENTS
49	API_COMMAND_NOT_SUPPORTED
50	API_ROLE_ALREADY_EXISTS
51	API_PERMISSIONIDENT_NOT_FOUND
52	API_ROLE_NOT_FOUND
53	API_SITEID_NOT_VALID
54	API_USER_ALREADY_EXIST
55	API_CONFIG_ERROR

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.

ARCHIVEACTION – An archive action in cc: Discover that specifies how long call recording files are maintained. The archive action ID number must be used.

CALLCOUNT – Number of calls that a user requests when using the CALLLIST function.

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

CALLID – The call identifier that originally comes from the PBX.

CALLINSTANCE – Each call on a phone is considered an instance. For example, an agent is on a call with a customer. The agent puts the customer on hold to call a supervisor for assistance. The phone has two call instances.

COREIDENT – ID of the cc: Discover core used to record a call. This value is found in the cc: Discover Web portal.

COUNT – Varies by function. Number of items for a particular function such as number of channels.

CRM_USER – Name of the user in the Customer Relationship Management system.

CTIINITIATED – 'Y' or 'N' value that indicates if a recording was initiated by a CTI event. Calls can also be initiated by API calls using CallCopy On-Demand or other applications.

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.

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.

DOMAIN – The Active Directory domain of a user's Windows network account.

EMAIL – An email account address.

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

FILENAME – Path name of a call recording file.

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 to which a device belongs.

GROUPID – CallCopy Group ID number.

GROUPNAME – CallCopy Group name.

ID – Channel ID number in cc: Discover Web portal.

IDENT – CallCopy recording ID number.

ISRECORDING – Value that indicates if a device is recording.

ISTRUNK – This value indicates if the recording device is a trunk.

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

KEYNAME – See CHATSTART.

KEYVALUE – See CHATSTART.

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

LASTSTATECHANGE – Date and time that a device's state changed (e.g., from 'Recording' to 'Available'.)

LASTSTATECHANGEUNIXTIME – Date and time that a device's state changed in the UNIX date/time format.

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.

PHONEID – User's phone number, extension, device ID.

PLAYBACKEXTENSION – This parameter is no longer supported.

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

RECORDINGDEVICE – This parameter is no longer supported.

RECORDINGDEVICEALIAS – This parameter is no longer supported.

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.

SESSIONID – For CHATSTART, CHATUPDATE, CHATSTOP, this value is provided by the client. For DEVICESTATUS and DEVICELIST, this value is based on the call and device.

STARTTIME – Time that recording started.

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

STATION – Windows workstation. See also STATIONNAME.

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.

STOPTIME – Time that recording stopped.

SYSTEMPROCESSORSPEED – A measured system processor speed-reading in MHz

SYSTEMPROCESSORTYPE – The description for the system processor 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

SYS_USER – 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).

TIMEOFFSET – Number of seconds to offset (i.e., increase/decrease) the stop time of the blackout. This value can be a positive or negative integer.

UNIQUEFIELD – This can be any column from the CallCopy database's Recording table, such as:

agent_id	ani	archive_action_id	audio_hash	audio_size
Calldirection	callid	channel	device_alias	device_id
dnis	duration	filename	filename_video	gate
globalid	Ident	keepdays	locked	listenedto
mediaLabel	overridden	priority	qa_forms	qa_group_id
qa_poss_score	qa_score	recording_time	recording_type	screen_capture
sgroup	user1	user2	user3	user4
user5	user6	user7	user8	user9
user10	user11	user12	user13	user14
user15	video_hash	video_size	wrap_duration	

UNIQUEID – This is a value in the column specified by the UNIQUEFIELD parameter. Together, these parameters identify a recording record in the Recording table.

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.

> **Note (**)** Updates to these fields using some special characters and symbols may not work correctly. Test all possible characters needed before using this function.

USERNAME – Username in cc: Discover.

Chapter 10: Data Type Values

The parameter data types and sizes in the charts below specify how data is stored in the CallCopy database. The REQUEST and RESPONSE XML messages to and from the API server pass these values as strings. The API converts the necessary items to the appropriate data type when writing values to the database.

The following are the value types for the CALLSTART, CALLSTOP, CALLUPDATE, RECORDSTART, and RECORDSTOP functions' parameters:

REQUESTID	integer (signed 32-bit)
DEVICEID	char(64)
CALLINSTATNCE	TBD
DEVICEALIAS	char(15)
GROUP	char(20)
GATE	char(20)
ANI	char(20)
DNIS	char(20)
USER1-USER3	varchar(20)
USER4-USER8	varchar(255)
USER9-USER15	varchar(50)
CALLID	char(16)
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)
UPDATE_IF_NOT_RECORDING	char(1) ('Y','N')
ARCHIVEACTION	bigint(8)
SYS_USER	char(30)
CRM_USER	char(30)
KEEPDAYS	TBD

The following are the value types for the DEVICELIST, DEVICESTATUS, and DEVICERECORDINGSTATUS 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-USER3	varchar(20)
USER4-USER8	varchar(255)
USER9-USER15	varchar(50)
CALLDIRECTION	char(1) (‘I’,’O’,’?’)
ISTRUNK	boolean
STATION	char(128)
ISRECORDING	boolean
FILENAME	char(255)
CALLID	char(30)

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
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 CHANNELS parameters.

*These parameters are no longer supported.

REQUESTID	integer (signed 32-bit)
COUNT	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 FETCHTERMINOLOGY parameters:

REQUESTID	integer (signed 32-bit)
COUNT	integer (signed 32-bit)
NAME	varchar(255)
VALUE	varchar(255)

The following are the value types for the CALLLIST parameters:

REQUESTID	integer (signed 32-bit)
DEVICEID	
CRM_USER	char(30)
SYS_USER	char(30)
DEVICEALIAS	char(15)
GROUP	char(20)
GATE	char(20)
ANI	char(20)
DNIS	char(20)
CALLCOUNT	NA
CALLDATESTART	bigint(8)
CALLDATEEND	bigint(8)
USER1- USER3	varchar(20)
USER4-USER8	varchar(255)
USER9- USER15	varchar(50)

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

REQUESTID	integer (signed 32-bit)
SESSIONID	char(255)
AGENTID	integer (signed 32-bit)
KEYNAME	string values USERNAME, EMPLOYEEID
KEYVALUE	char(255)
DEVICEALIAS	char(15)
GROUP	char(20)
GATE	char(20)
ANI	char(20)
DNIS	char(20)
USER1-USER3	varchar(20)
USER4-USER8	varchar(255)
USER9-USER15	varchar(50)
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	varchar(64)
DEVICEALIAS	varchar(100)
CRM_USER	varchar(50)
SYS_USER	varchar(255)
CALLINSTANCE	TBD

The following are the value types for the IMPORTUSER, IMPORTAGENTPHONE, and IMPORTSTATION parameters:

REQUESTID	integer (signed 32-bit)
ACTION	string values ADD, DELETE, MODIFY
AGENTID	integer (signed 32-bit)
FIRSTNAME	char(20)
LASTNAME	char(30)
SYS_USER	varchar(255)
DOMAIN	char(30)
EMAIL	char(255)
EMPLOYEEID	char(20)
ALLOWSELFRECORD	char(1) (‘Y’,’N’)
ALLOWSELFDOWNLOAD	char(1) (‘Y’,’N’)
ALLOWDUPLICATE	char(1) (‘Y’,’N’)
STATUS	char(1) (‘A’ - Active, ’I’ - Inactive)
SYS_DOMAIN	TBD

SITE_ID	TBD
MOBILE_ID	TBD
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 EMAIL and COMBINEDEMAIL parameters:

REQUESTID	integer (signed 32-bit)
UNIQUEFIELD	varchar(255)
UNIQUEID	varchar(255)
TOADDRESS	varchar(255)
TONAME	varchar(255)
FROMADDRESS	varchar(255)
FROMNAME	varchar(255)
MEDIA_FORMAT	varchar(3)
IDENT	varchar(255)

The following are the value types for the AGENTWINDOW parameters:

REQUESTID	integer (signed 32-bit)
COMPUTERNAME	varchar(255)
USERNAME	varchar(255)
DOMAIN	varchar(255)
IPADDRESS	varchar(255)
STATUS	varchar(255)
APPLICATIONTITLE	varchar(255)
LASTUPDATE	varchar(255)

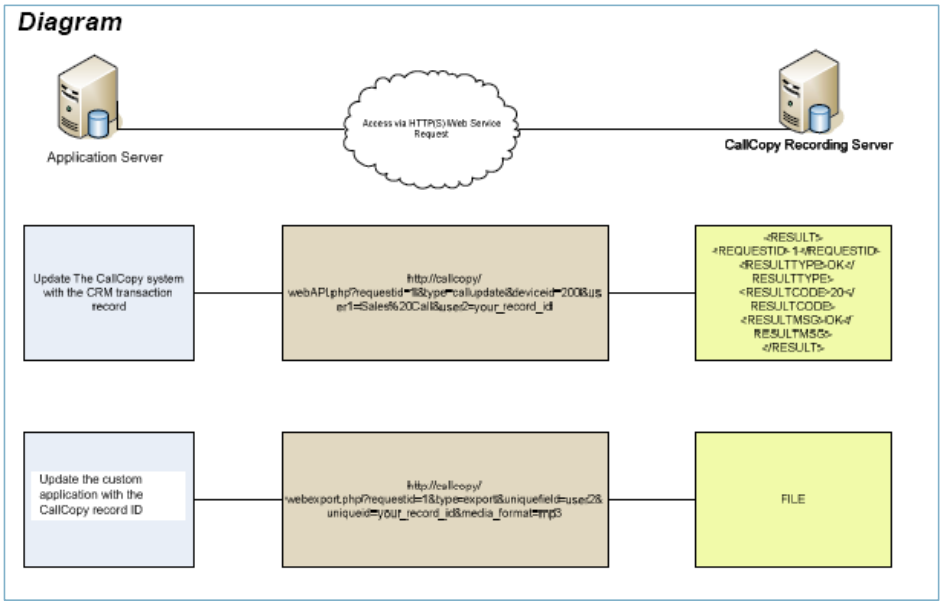
Chapter 11: CallCopy WebAPI Service

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 *cc: Discover 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

Update 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&devi
ceid=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

Export a Recorded Audio File

A function type 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 CAV, WAV (PCM), or MP3 format.

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. The possible values for UNIQUEFIELD are listed below.
- **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

Optional parameters **ENCRYPT** and **PASSWORD** may be used to encrypt the exported CAV file.

- **ENCRYPT** – Used to encrypt the exported CAV file, requires PASSWORD.
- **PASSWORD** – Password used to access the encrypted exported CAV.

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&encrypt=Y&password=test
```

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

Errors: Possible error returns are:

- API_FILE_NOT_FOUND – If it cannot find the file you're trying to export and email.
- API_INVALID_MESSAGE_FORMAT – This message appears if parameters are missing or the request is not formatted correctly.

Export Multiple Recorded Audio Files

The COMBINEDEXPORT function type combines multiple audio files into one file and exports it. It only uses these values:

- **IDENT:** These are comma-separated call record IDs.

The COMBINEDEXPORT call should be formatted in the following manner:

```
http://ccserveraddress:2012/webAPI.aspx?type=combinedexport&requestid=10482345&
ident=10121967,10121968,10121969&media_format=mp3
```

Errors: Possible error returns are:

- API_FILE_NOT_FOUND – If it cannot find the file you're trying to export and email.
- API_INVALID_MESSAGE_FORMAT – This message appears if parameters are missing or the request is not formatted correctly.

Export Multiple Recorded Audio Files as a ZIP File

The MULTIEXPORT function type combines multiple audio files into a ZIP file and exports that file. It only uses these values:

- **IDENT:** These are comma-separated call record IDs.

The MULTIEXPORT call should be formatted in the following manner:

```
http://ccserveraddress:2012/webAPI.aspx?type=multiexport&requestid=10482345&ide
nt=10121967,10121968,10121969&media_format=mp3
```

Errors: Possible error returns are:

- API_FILE_NOT_FOUND – If it cannot find the file you're trying to export and email.
- API_INVALID_MESSAGE_FORMAT – This message appears if parameters are missing or the request is not formatted correctly.

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 #1310 [Hex $DA]) to signal the end of data transmission.

Request Command

This is the command that any client application connecting directly to our API will send to initiate event monitoring. The internal CallCopy API passes these events to the Event Proxy.

```
<REQUEST>
  <REQUESTID>0</REQUESTID>
  <TYPE>EVENTS</TYPE>
  <EVENTCLASS>ALL</EVENTCLASS>
  <EVENTVALUE></EVENTVALUE>
</REQUEST>
```

Request Result Schema

```
<RESULT>
  <REQUESTTYPE>EVENTS</REQUESTTYPE>
  <REQUESTID>0</REQUESTID>
  <RESULTTYPE>API_OK</RESULTTYPE>
  <RESULTCODE>20</RESULTCODE>
  <RESULTMSG>API_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>
```

RECORDINGPAUSED Event

The RECORDINGPAUSED event signals that CallCopy has paused recording a call.

```
<CALLCOPYEVENT>
<EVENTCLASS>RECORD</EVENTCLASS>
<EVENTTYPE> RECORDINGPAUSED</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>
```

RECORDINGRESUMED Event

The RECORDINGRESUMED event signals that CallCopy has resumed recording a call from pause.

```
<CALLCOPYEVENT>
<EVENTCLASS>RECORD</EVENTCLASS>
<EVENTTYPE> RECORDINGRESUMED</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>
```

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.