



Uptivity WFO API Manual, v5.7

www.incontact.com

on Premises

Uptivity Desktop Analytics Administration Guide, v5.7

- **Version** — 5.7
- **Revision** — September 2015
- **About inContact** — inContact (NASDAQ: SAAS) is the cloud contact center software leader, helping organizations around the globe create high quality customer experiences with a complete workforce optimization portfolio (WFO). **Uptivity WFO** is part of the inContact portfolio and is a comprehensive WFO solution offering a powerful choice of deployment options. The portfolio also includes the WFO Suite powered by Verint[®], ECHO[™] Customer Feedback Survey, inView[™] Performance Dashboard.

inContact is the only provider to combine cloud software with an enterprise-class telecommunications network for a complete customer interaction solution. Winner of Frost & Sullivan 2012 North American Cloud Company of the Year in Cloud Contact Center Solutions, inContact has deployed over 1,300 cloud contact center instances. To learn more, visit www.inContact.com.

- **Copyright** — ©2015 inContact, Inc.
- **Disclaimer** — inContact reserves the right to update or append this document, as needed.
- **Contact** — Send suggestions or corrections regarding this guide to documentationsrequest-discover@incontact.com.

Table of Contents

| | |
|--------------------------------------|-----------|
| Introduction | 7 |
| Audience..... | 7 |
| Goals..... | 7 |
| Assumptions..... | 7 |
| Need-to-Knows | 7 |
| Uptivity WFO API Basics | 8 |
| Message Structure Overview | 8 |
| New Functions..... | 9 |
| Deprecated Functions..... | 9 |
| Call Handling Functions | 10 |
| CALLSTART..... | 10 |
| CALLSTOP..... | 13 |
| CALLUPDATE..... | 15 |
| CALLUPDATE – Post-Recording..... | 18 |
| RECORDSTART..... | 19 |
| RECORDSTOP | 21 |
| Status Functions | 23 |
| DEVICELIST | 23 |
| DEVICESTATUS | 26 |
| DEVICERECORDINGSTATUS | 29 |
| SYSTEMSTATUS | 33 |
| CHANNELS..... | 35 |

STATIONS 37

FETCHTERMINOLOGY 38

CALLLIST 40

Recording Control Functions 43

 CHATSTART 43

 CHATSTOP 45

 CHATUPDATE 46

 BLACKOUTSTART/AGENTBLACKOUTSTART 48

 BLACKOUTSTOP/AGENTBLACKOUTSTOP 50

Import Functions 52

 IMPORTUSER 52

 IMPORTAGENTPHONE 56

 IMPORTWORKSTATION 58

User, Group, and Role Functions 59

 ADDGROUP 59

 MODIFYGROUPPERMISSION 61

 MODIFYUSERGROUPPERMISSION 63

 MODIFYROLE 65

 MODIFYROLEGROUPPERMISSION 67

 MODIFYUSERROLE 69

 MODIFYROLEPERMISSION 71

 EMAIL 76

 COMBINEDEMAIL 78

 AGENTWINDOW 80

| | |
|---|------------|
| Results..... | 82 |
| Result Codes/Result Types..... | 82 |
| Parameter Definitions..... | 85 |
| Data Type Values..... | 90 |
| Call Handling Data Types | 90 |
| Status Data Types | 91 |
| Recording Control Data Types | 94 |
| Import Data Types..... | 95 |
| User, Group and Role Data Types | 96 |
| Uptivity WebAPI Service | 98 |
| General Architecture Overview | 98 |
| TCP Port Settings | 99 |
| WebAPI Call Syntax | 99 |
| Examples of WebAPI Calls | 100 |
| Update a Call Record | 100 |
| Export a Recorded Audio File | 101 |
| Export Multiple Recorded Audio Files..... | 103 |
| Export Multiple Recorded Audio Files as a ZIP File..... | 103 |
| Event Interface | 105 |
| Request Command..... | 105 |
| Request Result Schema | 105 |
| CALLSTART Event | 106 |
| CALLSTOP Event | 107 |
| RECORDINGSTARTED Event | 108 |

RECORDINGSTOPPED Event..... 109

RECORDINGUPDATE Event 109

RECORDINGPAUSED Event 110

RECORDINGRESUMED Event..... 111

Document Revision History 113

Introduction

Audience

This document is designed for developers, installation and support engineers, and customer or partner personnel who need to understand and use the Uptivity WFO API in a premises-based deployment.

Readers should be able to use a Windows PC and its common peripherals. They should also have intermediate to expert-level knowledge of Uptivity WFO, and have experience working with APIs and XML.

Goals

The goal of this document is to provide knowledge, reference, and procedural information necessary to understand and use the Uptivity WFO API for customizing an installation.

This document is NOT intended as a specific system or network design document. The guide also does not cover the Uptivity WFO services and components necessary for recording, PBX integration (such as CTI Core and Voice Boards), quality management, or other Uptivity WFO components.

Assumptions

This document assumes that Uptivity WFO has been installed and integrated with the PBX if applicable.

Need-to-Knows

Several Uptivity WFO features use menus and other windows that may be considered as “pop-ups” by some browsers. inContact recommends that you configure your browser to allow pop-ups for the Uptivity WFO site.

Uptivity WFO allows administrators to customize field names and terminology in the Web Portal to fit their unique environment. Therefore, screen examples and field names used in this manual may differ from those seen in your implementation.

In some cases, Uptivity WFO provides more than one way to accomplish a task or access a feature. The procedures in this manual explain the primary method, but also note the **Alternative** where applicable.

Uptivity WFO API Basics

Message Structure Overview

The Uptivity WFO 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. Commands can also be sent via http requests using the WebAPI method (see [Uptivity WebAPI Service](#)).

The Uptivity WFO 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.

Messages 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 (this attribute is required in every request message).

Example:

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

In the preceding 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 in the result message.

The result for this example message would look like the following:

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

The Uptivity WFO 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 Uptivity WFO API Server.

Deprecated Functions

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

| Deprecated Function | Replacement Function | Effective Version |
|------------------------|----------------------|-------------------|
| IMPORTSTATION | IMPORTWORKSTATION | 5.3 |
| MODECHANGE | N/A | 5.0 |
| EXTENSIONPLAYBACKSTART | N/A | 5.0 |
| EXTENSIONPLAYBACKSTOP | N/A | 5.0 |
| RECORDINGDEVICE | N/A | 5.0 |
| RECORDINGDEVICEALIAS | N/A | 5.0 |
| IMPORTAGENT | IMPORTUSER | 5.0 |

Call Handling Functions

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 Uptivity WFO)

- 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.

STATIONNAME may be passed as the required item in place of DEVICEID. If DEVICEID is not passed or is passed as a blank value, Uptivity WFO 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 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.

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 a CALLSTART is sent by DEVICEALIAS, SYS_USER, or CRM_USER, and a matching device isn't found, 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>
```

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 asterisk (*) is required for the request to be executed.

Parameters marked with a plus sign (+) can only be used if a device ID has already been associated with an agent.

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, Uptivity WFO 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.

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>
```

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.

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.

Parameters:

- REQUESTID (required)
- DEVICEID *
- CALLINSTANCE
- CRM_USER *+
- SYS_USER *+
- DEVICEALIAS *+
- STATIONNAME *
- GROUP
- GATE
- ANI
- DNIS
- USER1 – USER15 **
- CALLDIRECTION
- ARCHIVEACTION (this must be an archive action created in Uptivity WFO)
- KEEP_DAYS
- UPDATE_IF_NOT_RECORDING

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 only be used if a device ID has already been associated with an agent.

Using angle brackets (< >), some special characters, and symbols in fields marked with a double asterisk (**) 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>
```

CALLUPDATE – Post-Recording

Description: Allows a user to update information about a call that has finished recording. This information is used to update the recording 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 fields marked with a double asterisk (**) 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 REQUESTID 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 be an archive action created in Uptivity WFO)
- STATIONNAME *
- SYS_USER * +
- CRM_USER * +
- USER1-USER15 **

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.

Using angle brackets (< >), some special characters, and symbols in fields marked with a double asterisk (**) 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: #,#,#`

When a callstart is sent by DEVICEALIAS, SYS_USER, or CRM_USER, and no matching device is found, 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>
  <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>

<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 phone call on a device has ended. 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 asterisk (*) is required for the request to be executed.

Parameters marked with a plus sign (+) can only be used if a device ID has already been associated with an agent.

Using angle brackets (< >), some special characters, and symbols in fields marked with a double asterisk (**) 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 ``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>
  <USER1>test</USER1>
  <USER3>1</USER3>
</REQUEST>

<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>
```

Status Functions

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 recording.

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 Uptivity WFO is aware of is kept in memory on the system. This function returns only one instance on a device. For all instances, see [DEVICERECORDINGSTATUS](#).

Parameters:

- DEVICELIST
- REQUESTID

Returns: RESULT section plus 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>

  <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
- CALLID
- DEVICEALIAS
- CRM_USER
- SYS_USER

Returns: RESULT section plus 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 (that is, recording or recording pause).

Parameters:

- REQUESTID
- DEVICEID
- DEVICEALIAS
- CRM_USER
- SYS_USER

Returns: RESULT section plus 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>jane.doe</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>00001000381342197107</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, disk space, and so forth.

Parameters:

- REQUESTID

Returns: RESULT section plus 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>
  <RECORDINGSPEACEINFO>223.66GB of 465.64GB</RECORDINGSPEACEINFO>
  <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 a CHANNEL section for each channel on the system containing:

- ID: This is the Channel ID from the recording database. The number of ID sections is the same as the COUNT.
- STATE
- 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 number of stations, plus a STATIONS section for each 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>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 TERMINOLOGY sections for each terminology variable specified in the system with:

- NAME
- VALUE

Example:

```

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

<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>
  </SETTINGS>
</RESULT>

```

```

</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)

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

Filters are combined as a Boolean "AND" operation.

Depending on which parameters are passed, the number of different parameters passed, and the number of results that match passed parameters, database performance may be impacted. Uptivity WFO's recording capability is designed to function normally during database outages, but proper consideration must be taken when designing call list queries to prevent excessive use of database resources.

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 all recording records matching the query criteria, listed in descending order by date/time recorded.

Example:

```

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

<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>
```

Recording Control Functions

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

Using these functions, you can create custom triggers in your software to start and stop recording. Potential applications include allowing 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. Separate instances of screen capture on the same system (for example, for separate chat windows/sessions) can be started and stopped by matching the SESSIONID with the CHATSTART and CHATSTOP commands in the requests sent. Performing more than three concurrent screen captures on a single system may result in reduced system performance.

Parameters:

- REQUESTID (required)
- SESSIONID (required)
- AGENTID

If AGENTID is not provided, Uptivity WFO 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 KEYNAME.

Although AGENTID is the parameter in this function, the field used to identify the agent is USERID.

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 Uptivity WFO 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 to 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 the 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, but 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

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

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. The blackout begins based on when the message is received plus any TIMEOFFSET.

While BLACKOUT is 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 is enabled by default. Either standard or real-time blackouts will occur, depending on your system configuration. For more information, contact your system administrator or Uptivity WFO Support.

Parameters:

- REQUESTID
- DEVICEID
- DEVICEALIAS
- CRM_USER
- SYS_USER
- TIMEOFFSET
- CALLINSTANCE (this parameter is only used for BLACKOUT, not AGENTBLACKOUT)

Returns: Standard result code. 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.

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>

```

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. The blackout stops based on when the message is received and the value of the time offset.

While BLACKOUT is 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 is enabled by default. Either standard or real-time blackouts will occur, depending on your system configuration. For more information, contact your system administrator or Uptivity WFO Support.

Parameters:

- REQUESTID (required)
- DEVICEID
- DEVICEALIAS
- CRM_USER
- SYS_USER
- TIMEOFFSET
- CALLINSTANCE (this parameter is only used for BLACKOUT, not AGENTBLACKOUT)

When the time offset value causes a BLACKOUTSTOP to be issued before the BLACKOUTSTART, the start and stop times are the same. For example, if a BLACKOUTSTOP is sent with a time offset of -60 and the blackout started only 30 seconds ago, the BLACKOUTSTOP time will be the same as the BLACKOUTSTART time.

CALLINSTANCE is used to distinguish which call the blackouts are issued to. If no CALLINSTANCE is provided on BLACKOUTSTART or BLACKOUTSTOP, then the CALLINSTANCE of the last call on the device is used.

Returns: Standard result code. 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.

Example:

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

Import Functions

Import functions allow you to write integration links between your systems and the Uptivity WFO system. Potential applications include writing your own front-end interface to edit certain information contained within Uptivity WFO, or writing automated scripts that update information in the server.

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. This command requires a password to import user account information. USERNAME is the Uptivity WFO username, while CRM_USER is the corresponding username associated with the user in an integrated system.

Parameters:

- REQUESTID
- ACTION (Add, Delete, Modify, List)
- USERID
- EMPLOYEEID
- FIRSTNAME
- LASTNAME
- STATUS (This parameter determines whether the USER is flagged as an AGENT in Uptivity WFO. Possible values are A – Active (flagged as an AGENT) or I – Inactive (not flagged as an AGENT).
- ADD_TIME (Date/time offset)
- MOD_TIME (Date/time offset)
- SYS_USER
- DOMAIN
- EMAIL
- CRM_USER
- PASSWORD
- SITEID
- USERNAME

The field USERID is ignored when the action is ADD.

Deleting a user will simply change the status flag of the user to "deleted" and free all phone ID associations (if any) with that user. The actual record will not be removed. This ensures that past records will still identify the correct user in case a phone ID is reused for a new user.

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

An error will be returned if the USERID does not exist, or if you try to add a new user with the same first and last name as an existing user. Even though an error is returned when you try to add a user with the same first and last name as an existing user, values are not changed. Thus, you can write an automated script that continuously calls ADD and ignore errors.

LIST should be used to get a list of all users currently in Uptivity WFO. No additional parameters are needed. Returned are USERID, EMPLOYEE_ID, FIRST_NAME, LAST_NAME, STATUS, ADD_TIME, MOD_TIME, SYS_USER, EMAIL, CRM_USER, and PHONES.

Returns:

- Standard Result
- USERID

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 is returned.
- **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 appear if the agent is not found. If it appears as the result of a query error, check the API logs for details.

Example:

```

<REQUEST>
  <TYPE>IMPORTUSER</TYPE>
  <REQUESTID>2</REQUESTID>
  <ACTION>ADD</ACTION>
  <AGENT_ID>2</USERID>
  <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</USERID>
      <EMPLOYEEID>5</EMPLOYEEID>
      <FIRSTNAME>John</FIRSTNAME>
      <LASTNAME>Doe</LASTNAME>
      <STATUS>A</STATUS>
      <ADD_TIME>10/30/2006 4:05:38 PM</ADD_TIME>
    </AGENT>
  </AGENTS>

```

```
<MOD_TIME>3/27/2008 3:34:43 PM</MOD_TIME>  
<SYS_USER>jdoe</SYS_USER>  
<DOMAIN>Uptivity</DOMAIN>  
<EMAIL>jdoe@uptivity.com</EMAIL>  
<CRM_USER>john</CRM_USER>  
<PHONES>1123,3343,3234</PHONES>  
</AGENT>  
</AGENTS>  
</RESULT>
```

IMPORTAGENTPHONE

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

Parameters:

- REQUESTID
- ACTION (Add, Delete)
- PHONEID
- AGENTID
- FIRSTNAME
- LASTNAME

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

Deleting a PHONEID has no effect on the agent record itself.

Deleting a PHONEID does not remove the call records for that PHONEID.

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.

Although AGENTID is the parameter in this function, the field used to identify the agent is USERID.

Returns:

- Standard Result
- AGENTID

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.

Example:

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

IMPORTWORKSTATION

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

Parameters:

- REQUESTID*
- DEVICEID*
- STATIONNAME*
- LOCATIONID*: the unique Location ID from the database.

Parameters marked with an asterisk (*) are required for the request to be executed.

Before the new information is added, 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, the result DEVICEID may be an old value. This design allows for easier troubleshooting.

Returns:

- Standard Result
- DEVICEID
- STATIONNAME

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.

Example:

```
<REQUEST>
  <TYPE>IMPORTWORKSTATION</TYPE>
  <REQUESTID>2</REQUESTID>
  <DEVICEID>1234</DEVICEID>
  <STATIONNAME>COMPUTER1</STATIONNAME>
  <LOCATIONID>1</LOCATIONID>
</REQUEST>
```

User, Group, and Role Functions

These functions allow you to control Uptivity WFO users, groups, and roles from an external resource.

ADDGROUP

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

Uptivity WFO 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.

Parameters:

- REQUESTID
- GROUPNAME*

Returns:

- Standard Result
- GROUPID
- GROUPNAME

Possible error returns are:

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

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>
```

MODIFYGROUPPERMISSION

Description: Allows an administrator to add or remove agents and their device assignments to/from Uptivity Groups.

Parameters:

- REQUESTID
- ACTION
- GROUPNAME
- GROUPLD
- PHONEID OR DEVICEID
- FIRSTNAME
- LASTNAME
- AGENTID

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

You must supply either GROUPNAME or GROUPLD, but you do not have to supply both.

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

Although AGENTID is the parameter in this function, the field used to identify the agent is USERID.

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

Returns:

- Standard Result
- Parameters Sent

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.

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>
```

MODIFYUSERGROUPPERMISSION

Description: Allows an administrator to add or remove permissions for users to view an Uptivity Group and manage data (for example, call records, evaluations) related to agents in that group.

Parameters:

- REQUESTID
- ACTION
- GROUPNAME
- GROUPLD
- DEVICEID
- USERNAME

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

You must supply either GROUPNAME or GROUPLD, but you do not have to supply both.

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

Returns:

- Standard Result
- Parameters Sent

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 appear if the agent is not found. If it appears as the result of a query error, check the API logs for details.

Example:

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


MODIFYROLE

Description: Allows an administrator to add an Uptivity WFO Role.

Parameters:

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

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

You must supply REQUESTID.

Uptivity WFO will attempt to add the new role. ROLENAME and 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.

Returns:

- Standard Result
- Parameters Sent

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.

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>
```

MODIFYROLEGROUPPERMISSION

Description: Used to assign an Uptivity Group to a Role.

Parameters:

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

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

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

You must supply REQUESTID and GROUPNAME.

Returns:

- Standard Result
- Parameters Sent

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.

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>
```

MODIFYUSERROLE

Description: Used to assign Roles to Users. You can use any of these to associate user info: USERID, LASTNAME, FIRSTNAME, SYSTEMUSERNAME, OR USERNAME.

Parameters:

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

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

Returns:

- Standard Result
- Parameters Sent

Possible error returns are:

- **API_DATABASE_QUERY_ERROR:** Represents a range of possible query problems. Check 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.

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>
```

MODIFYROLEPERMISSION

Description: Used to modify permissions assigned to a Role. The PERMISSION NAME used in the API expression differs from the way the permission is listed in the Discover Web Portal (see table). For details regarding permissions, see the *Uptivity WFO Administration Manual*.

| PERMISSION NAME | Value in Discover Web Portal |
|----------------------|---|
| ALLOWCHANGEPASSWORD | Allow Password Changes |
| ALLOWQAVIEW | Allow Viewing of QA Evaluations |
| CALLREPORTING | Allow Viewing Call Reports |
| QAREPORTING | Allow Viewing QA Reports |
| VIEWOWNRECORDS | Allow Viewing of User's Own Records |
| ALLOWAGENTADMIN | Allow User Administration |
| ALLOWAPIAUTH | Allow API Authentication |
| ALLOWARCHIVEADMIN | Allow Archive Administration |
| ALLOWCOMMENT | Allow Bookmarking |
| ALLOWEDITCOMPLETEDQA | Allow Editing of Completed QA Evaluations |
| ALLOWEXPORTDOWNLOAD | Allow Downloading of Export |
| ALLOWEXPORTEMAIL | Allow Emailing of Export |
| ALLOWFILEDELETE | Allow Recording Record and File Deletes |
| ALLOWGROUPADMIN | Allow Group Administration |
| ALLOWLIVEMONITOR | Allow Live Monitoring of Calls |
| ALLOWQAEDIT | Allow QA Form Administration (cannot be granted unless user also has Allow Viewing of QA Evaluations/ALLOWQAVIEW) |
| ALLOWQADELETE | Allow Deletion of Completed QA Evaluations |
| ALLOWSYSTEMCONFIG | Allow System Configuration |
| ALLOWSYSTEMREPORTS | Allow Viewing System Reports |
| ALLOWVIEWALLRECORDS | Allow Viewing All Call Records & QA Evaluations |
| AUDITREPORTING | Allow Viewing Audit Reports |

| | |
|--|----------------------------------|
| CONTENTUPLOAD | Allow Content Library Management |
| REPORTSUBSCRIPTION | Allow Report Subscriptions |
| ALLOWWIDGETADMINISTRATION | Allow Widget Administration |
| The following applies only if you use Uptivity Screen Recording. | |
| ALLOWVIEWVIDEO | Allow Viewing of Video |
| The following apply only if you use Uptivity On-Demand. | |
| ALLOWBLACKOUT | Allow Blackout Start and Stop |
| ALLOWRECORDINGALIAS | Allow Recording by Device Alias |
| ALLOWRECORDDESKTOP | Allow Desktop Recording |
| ALLOWRECORDID | Allow Recording by Device ID |
| ALLOWRECORDSTOP | Allow Recording Stop |
| ALLOWUPDATEAUDIO | Allow Call Updates |
| PROMPTID | Prompt for Device at Login |
| SECUREALL | Prevent Settings Changes |
| SECUREID | Prevent Device ID Changes |
| ALLOWWEBONDEMAND | Allow Web On Demand |
| The following apply only if you use Uptivity Speech Analytics. | |
| ANALYTICSREPORTING | Allow Viewing Analytics Reports |
| ALLOWVIEWANALYTICS | Allow Analytics View |
| ALLOWANALYTICSADMIN | Allow Analytics Administration |
| The following apply only if you use Uptivity Survey. | |
| SURVEYREPORTING | Allow Viewing Survey Reports |
| ALLOWSURVEYEDIT | Allow Editing Surveys |
| ALLOWSURVEYADMIN | Allow Survey Administration |
| ALLOWSURVEYDELETE | Allow Deleting Surveys |
| ALLOWSURVEYVIEW | Allow Viewing Surveys |
| The following applies only if you use Uptivity Desktop Analytics. | |
| ALLOWFUSIONADMIN | Allow Fusion Administration |

| The following apply only if you use Clarity WFM. | |
|--|-----------------------------|
| CONFIGURATIONSECTION | Configuration Section |
| EDITNEWSWIDGET | Edit News Widget |
| EMPLOYEECREATE | Employee Create |
| EMPLOYEEPROFILEALLVIEW | Employee Profile All View |
| EMPLOYEEPROFILETEAMVIEW | Employee Profile Team View |
| EMPLOYEEESCHEDULEALLEDIT | Employee Schedule All Edit |
| EMPLOYEEESCHEDULEALLVIEW | Employee Schedule All View |
| EMPLOYEEESCHEDULETEAMEDIT | Employee Schedule Team Edit |
| EMPLOYEEESCHEDULETEAMVIEW | Employee Schedule Team View |
| EMPLOYEESEARCH | Employee Search |
| EMPLOYEESECTION | Employee Section |
| EMPLOYEESELFEDIT | Employee Self Edit |
| FORECASTACQUIRE | Forecast Acquire |
| FORECASTPREDICT | Forecast Predict |
| FORECASTSECTION | Forecast Section |
| FORECASTTREND | Forecast Trend |
| HISTORICALWIDGETS | Historical Widgets |
| HOMEPAGEWIDGETS | Home Page Widgets |
| LEAVEREQUESTAPPROVALALL | Leave Request Approval All |
| LEAVEREQUESTAPPROVALTEAM | Leave Request Approval Team |
| REALTIMEWIDGETS | Real Time Widgets |
| REPORTSHISTORICAL | Reports Historical |
| REPORTSPROCESSES | Reports Processes |
| REPORTSREALTIME | Reports Real Time |
| REPORTSSECTION | Reports Section |
| ROSTERALL | Roster All |
| ROSTERTEAM | Roster Team |
| SCHEDULECREATE | Schedule Create |

| | |
|-------------------------|----------------------------|
| SCHEDULELOAD | Schedule Load |
| SCHEDULEPUBLISH | Schedule Publish |
| SCHEDULESECTION | Schedule Section |
| SWAPREQUESTAPPROVALALL | Swap Request Approval All |
| SWAPREQUESTAPPROVALTEAM | Swap Request Approval Team |

Parameters:

- REQUESTID
- ROLEID
- ROLENAME
- PERMISSIONNAME (see table)
- ACTION (add/delete)

You must supply either ROLEID or ROLENAME, but you do not have to supply both. If ROLENAME is used, the system will look up the associated ROLEID.

Returns:

- Standard Result
- Parameters Sent

Possible error returns are:

- **API_DATABASE_QUERY_ERROR:** Represents a range of possible query problems. Check API logs for details.
- **API_ERROR_INVALID_ACTION:** Allowed actions are ADD or 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.

Example:

```

<REQUEST>
  <REQUESTTYPE> MODIFYROLEPERMISSION </REQUESTTYPE>
  <REQUESTID>1</REQUESTID>
  <ACTION>ADD</ACTION>
  <ROLENAME>DefaultRole</ROLENAME>
  <PERMISSIONNAME>HOMEPAGEWIDGETS</PERMISSIONNAME>
</REQUEST>

```

```
<RESULT>  
  <REQUESTTYPE>MODIFYROLEPERMISSION</REQUESTTYPE>  
  <REQUESTID>1</REQUESTID>  
  <RESULTTYPE>API_OK</RESULTTYPE>  
  <RESULTCODE>20</RESULTCODE>  
  <RESULTMSG>Permission Added</RESULTMSG>  
</RESULT>
```

EMAIL

Description: Prompts Uptivity WFO 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

Possible error returns are:

- **API_EMAIL_NOT_VALID:** The email address in the request is incorrect or could not be found.
- **API_ERROR_CLIENT_NOT_FOUND:** The email server is not configured or cannot be found.
- **API_FILE_NOT_FOUND:** The specific file to be exported and emailed cannot be found.
- **API_INVALID_MESSAGE_FORMAT:** Appears if parameters are missing or the request is not formatted correctly.

Example:

```

<REQUEST>
<TYPE>EMAIL</TYPE>
<REQUESTID>14</REQUESTID>
<TOADDRESS>jdoe@uptivity.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>

```

COMBINEDEMAIL

Description: Prompts Uptivity WFO 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

Possible error returns are:

- **API_EMAIL_NOT_VALID:** The email address in the request is incorrect or could not be found.
- **API_ERROR_CLIENT_NOT_FOUND:** The email server is not configured or cannot be found.
- **API_FILE_NOT_FOUND:** A specific file to be exported and emailed cannot be found.
- **API_INVALID_MESSAGE_FORMAT:** Appears if parameters are missing or the request is not formatted correctly.

Example:

```

<REQUEST>
<TYPE>COMBINEDEMAIL</TYPE>
<REQUESTID>1</REQUESTID>
<TOADDRESS>jane.doe@uptivity.com</TOADDRESS>
<TONAME>jane</TONAME>
<FROMADDRESS>jane.doe@uptivity.com</FROMADDRESS>
<FROMNAME>jane</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>

```

AGENTWINDOW

Description: Provides latest information on the agent's window as captured by Uptivity Screen Recording.

Parameters:

- REQUESTID
- COMPUTERNAME or
- USERNAME

Returns:

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

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.

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>uptivity.com</DOMAIN>
<IPADDRESS>10.100.5.91</IPADDRESS>
<STATUS>AVAILABLE</STATUS>
<APPLICATIONTITLE>Internet Explorer</APPLICATIONTITLE>
<LASTUPDATE>Feb 2 2011 4:53PM</LASTUPDATE>
</RESULT>

```

Results

All messages reply with a minimum of a root tag called RESULT. This tag contains 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

| RESULTCODE Return Value | 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 |

Parameter Definitions

ALLOWSELFRECORD: Permission given to an agent allowing them to record their own calls. Possible values (Y/N).

ALLOWSELFDOWNLOAD: Permission given to an agent allowing them to download their recordings. 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 Uptivity WFO. This ID is the internal system identifier used by Uptivity WFO to associate agents to phone IDs, calls, and so forth.

AGENTINITIATED: Indicates whether a recording was initiated by the agent. Possible values (Y/N).

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

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

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

CALLDIRECTION: Denotes whether the call was inbound or outbound. Possible values (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 Uptivity WFO Core used to record a call. This value is found in the Discover Web Portal.

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

CRM_USER: Name of the user in a customer relationship management system.

CTIINITIATED: Indicates whether a recording was initiated by a CTI event. Calls can also be initiated by API calls using Uptivity On-Demand or by other applications. Possible values (Y/N).

DEVICEID: 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: 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 the caller dialed to reach 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 Uptivity WFO.

FILENAME: Path name of a call recording file.

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

GATE: An identifier for the call type as identified in the PBX. This is also known as the skill set, vector, or application.

GROUP: The ACD group to which a device belongs.

GROUPID: Uptivity Group ID number.

GROUPNAME: Uptivity Group name.

ID: Channel ID number in the Discover Web Portal.

IDENT: Uptivity WFO recording ID number.

ISRECORDING: Value that indicates if a device is recording.

ISTRUNK: Value that 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 (for example, from 'Recording' to 'Available').

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

LOCATIONID: Unique identification key for a Location, stored in the database.

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 to conserve disk space in the event a stop message is not received.

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

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

RECORDINGPATH: The system variable value specifying the default path for recordings.

RECORDINGSPACEINFO: A description of the amount of free disk space on the recording drive. For example: 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 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 a return error if the value is higher.

SESSIONID: For CHATSTART, CHATUPDATE, and 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 are:

- 0 – On Hook
- 1 – Off Hook
- 2 – Recording
- 3 – Playback
- 4 – Waiting
- 5 – Error
- 6 – Resetting
- 7 – Recording Prep
- 8 – Finish Recording
- 9 – Dialing For 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, Uptivity WFO 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 recording database for the DEVICEID. This allows you to define your station mappings externally without having to store them in the local Uptivity WFO system database.

STOPTIME: Time that recording stopped.

SYSTEMPROCESSORSPEED: A measured system processor speed. The 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 free physical RAM 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 network user to an agent defined in the Uptivity WFO system and should equal the network username (not the Uptivity WFO username) of the agent. This association is used to grant them the proper access rights (such as phone IDs that they may record and download permissions).

TIMEOFFSET: Number of seconds to offset (that is, 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 Uptivity WFO 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 to be stored with the call record. These values are stored as ASCII text.

⚠ ()** 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 Uptivity WFO.

Data Type Values

Charts in this section specify how data is stored in the Uptivity WFO database. REQUEST and RESPONSE XML messages to and from the API Server pass these values as strings. The API converts the items to the appropriate data type when writing values to the database.

Call Handling Data Types

| Functions: CALLSTART, CALLSTOP, CALLUPDATE, RECORDSTART, RECORDSTOP | |
|---|-------------------------|
| Parameter | Value Type |
| REQUESTID | integer (signed 32-bit) |
| DEVICEID | char(64) |
| CALLINSTANCE | 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 |

Status Data Types

| Functions: DEVICELIST, DEVICESTATUS, DEVICERECORDINGSTATUS | |
|---|-------------------------|
| Parameter | Value Type |
| 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) |
| Function: SYSTEMSTATUS | |
| Parameter | Value Type |
| REQUESTID | integer (signed 32-bit) |
| RECORDINGPATH | string |
| RECORDINGSPEACEINFO | 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 |
| Function: CHANNELS | |
| Parameter | Value Type |
| REQUESTID | integer (signed 32-bit) |
| COUNT | Integer (signed 32-bit) |
| ID | integer (signed 32-bit) |
| STATE | String |
| LASTSTATECHANGE | String |
| LASTSTATECHANGEUNIXTIME | String |
| Function: STATIONS | |
| Parameter | Value Type |
| REQUESTID | integer (signed 32-bit) |
| COUNT | integer (signed 32-bit) |
| STATIONNAME | char(50) |
| DEVICEID | char(64) |
| Function: FETCHTERMINOLOGY | |
| Parameter | Value Type |
| REQUESTID | integer (signed 32-bit) |
| COUNT | integer (signed 32-bit) |
| NAME | varchar(255) |
| VALUE | varchar(255) |

| Function: CALLLIST | |
|---------------------------|-------------------------|
| Parameter | Value Type |
| REQUESTID | integer (signed 32-bit) |
| DEVICEID | char(64) |
| 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) |

Recording Control Data Types

| Function: CHATSTART, CHATSTOP, CHATUPDATE | |
|--|------------------------------------|
| Parameter | Value Type |
| REQUESTID | integer (signed 32-bit) |
| SESSIONID | char(255) |
| AGENTID (USERID) | 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) |
| Function: BLACKOUTSTART, BLACKOUTSTOP | |
| Parameter | Value Type |
| REQUESTID | integer (signed 32-bit) |
| DEVICEID | varchar(64) |
| DEVICEALIAS | varchar(100) |
| CRM_USER | varchar(50) |
| SYS_USER | varchar(255) |
| CALLINSTANCE | TBD |

Import Data Types

| Function: IMPORTUSER, IMPORTAGENTPHONE, IMPORTWORKSTATION | |
|--|--|
| Parameter | Value Type |
| 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) |
| LOCATIONID | integer (signed 32-bit) |

User, Group and Role Data Types

| Function: ADDGROUP, MODIFYGROUPPERMISSION, MODIFYUSERGROUPPERMISSION | |
|---|---------------------------|
| Parameter | Value Type |
| 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) |
| Function: EMAIL, COMBINEDEMAIL | |
| Parameter | Value Type |
| 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) |

| Function: AGENTWINDOW | |
|------------------------------|-------------------------|
| Parameter | Value Type |
| 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) |

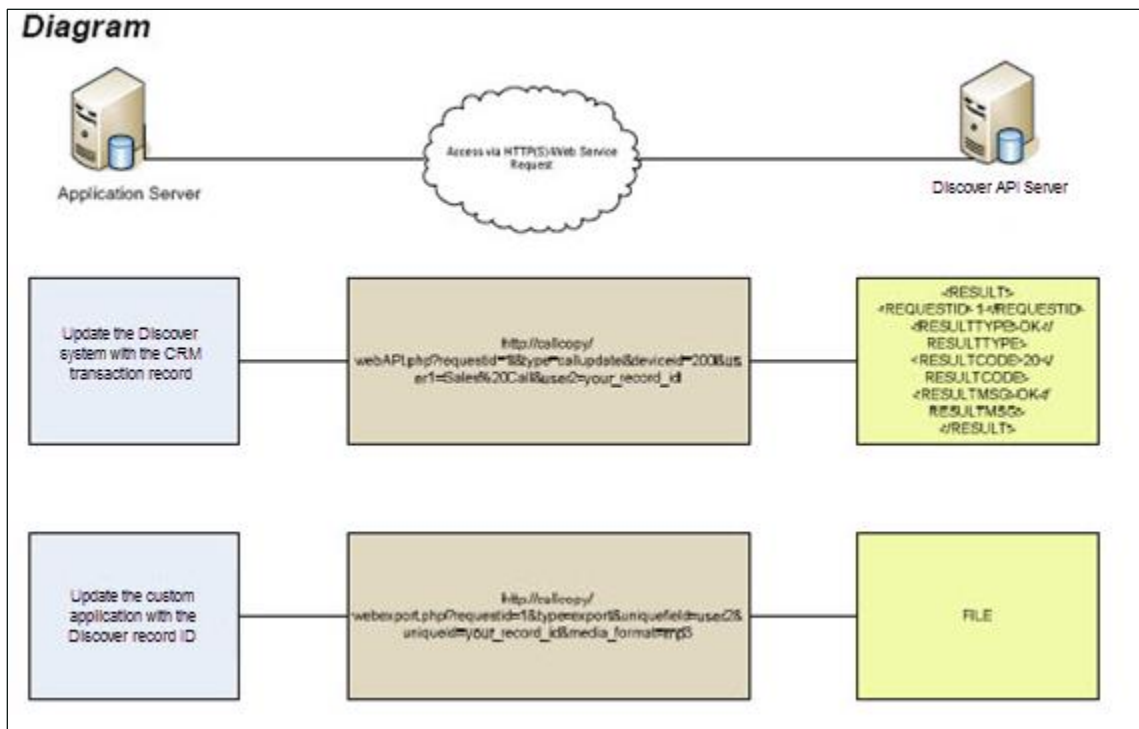
Uptivity WebAPI Service

General Architecture Overview

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

The WebAPI service accepts specially-formatted URLs that contain API data and translates the URL into an Uptivity WFO 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 WebAPI service calls is shown in this diagram.



The WebAPI service is a built-in component on every Uptivity WFO 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 runs on port 2012 (2013 if SSL is used). These port settings can be modified if needed. See the "API Server Settings" section of the *Uptivity WFO Administration Manual* for more information on API Server configuration settings.

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

Non SSL-Enabled Connections Example:

```
http://discoverserveraddress:2012/webAPI.aspx
```

SSL-Enabled Connections Example:

```
https://discoverserveraddress:2013/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&deviceid=200&user1=SpecialCall&user2=10121968&keepdays=3650
```

Each request call must include the following:

- **Protocol Identifier:** This determines whether the call will be sent as plain text or will use SSL encryption. For plain text, set the value to **HTTP**. For SSL encryption, set the value to **HTTPS**.
- **Host Address:** This is the hostname or IP Address of the server running the Uptivity WFO API Service.
- **TCP Port:** This is the port number on which the API Service is listening for web requests.
- **API Page:** Calling this page allows an application to access any functions in the Uptivity WFO 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. The 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 for the specified API function as defined for each function call type (call handling, status, and so forth). All API functions are available via the WebAPI method.

To make a successful WebAPI call, the API function parameters must 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 equal (=) sign, followed directly with the desired value for that parameter. Parameters are case-insensitive.

You cannot use the same parameter more than once in a single WebAPI call as the parameters will be combined before passing to the Uptivity WFO API. For example:

- Your WebAPI call contains "deviceid=1234&deviceid=5555".
- The values are combined into the comma-separated list "deviceid=1234, 5555" before the Uptivity WFO API receives the command.
- The Uptivity WFO API treats this as a single value "1234,5555" and not two separate DeviceIDs.

If your call contains duplicate parameters with different casing (for example, "deviceid=1234&DEVICEID=5555"), the API returns an error message: "parameters must be unique but duplicates were found." However, if you use multiple parameters with the same casing (for example, "deviceid=1234&deviceid=5555"), the command simply will not return the correct results.

Examples of WebAPI Calls

Update a Call Record

This example 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. For parameter requirements, see [CALLUPDATE](#).

The WebAPI call should be formatted as follows:

```
http://ccserveraddress:2012/webAPI.aspx?type=callupdate&requestid=18213567&deviceid=200&user9=SpecialCall&user10=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. In the example, the value is **18213567**.
 - **DeviceID (required):** The phone extension for the call being recorded. In the example, the value is **200**.
 - **User9:** Custom-defined field (in this example, used to display a text phrase). In the example, the value is **SpecialCall**.
 - **User10:** Custom-defined field (in this example, a transaction ID number). In the example, the value is **10121968**.
 - **Keepdays:** The number of days this record will be maintained before being purged. In the example, the value is **3650**.

Export a Recorded Audio File

EXPORT is a function type unique to the WebAPI. This function allows Uptivity WFO to export recorded media from a call any time after recording has completed.

The EXPORT function supports exporting audio only in the following formats: WAV (PCM), MP3, and VOX. You can also export the entire recording, including metadata, using the proprietary CAV format.

The EXPORT function has three required parameters: **UNIQUEFIELD**, **UNIQUEID**, and **MEDIA_FORMAT**. These parameters are used to identify the desired audio file and specify the export format.

- **UNIQUEFIELD:** 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:** Value in the UNIQUEFIELD parameter that uniquely identifies the record.
- **MEDIA_FORMAT:** File format of the exported audio file.

Optional parameters **ENCRYPT** and **PASSWORD** may be used to encrypt an 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&uniquefield=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:** Unique identifier for the transaction generated by the user making the WebAPI call. The value must be an integer. In the example: **10482345**.
 - **UNIQUEFIELD:** Required parameter name that contains identifying information for the record. In the example: user-generated transaction ID **user2**.
 - **UNIQUEID:** Value contained in the UNIQUEFIELD parameter for the record. In the example: **10121968**.
 - **MEDIA_FORMAT:** File format of exported audio file. In the example: **mp3**.
 - **ENCRYPT:** Optional parameter used to encrypt exported CAV file. In the example: **Y**.
 - **PASSWORD:** Required parameter if encryption is used. In the example: **test**.

Possible error returns are:

- **API_FILE_NOT_FOUND:** Appears if the specific file to be exported and emailed cannot be found.
- **API_INVALID_MESSAGE_FORMAT:** Appears if parameters are missing or the request is not formatted correctly.

Export Multiple Recorded Audio Files

COMBINEEXPORT is unique to the WebAPI. This function type combines multiple audio files into a single file (post-recording) and exports the file in the specified format. It uses these values:

- **IDENT:** These are comma-separated call record IDs.
- **MEDIA_FORMAT:** File format of the exported audio file.

The COMBINEEXPORT call should be formatted in the following manner:

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

This example URL was constructed from the following components:

- **Protocol:** http
- **Host Address:** ccserveraddress
- **TCP Port:** 2012
- **API Page:** WebAPI.aspx
- **Function Type:** combinedexport
- **Parameters:**
 - **REQUESTID:** Unique identifier for the transaction generated by the user making the WebAPI call. The value must be an integer. In the example: **10482345**.
 - **IDENT:** Required parameter specifying the recordings to be combined. In the example: comma-separated call record IDs **10121967,101219168,10121969**.
 - **MEDIA_FORMAT:** File format of exported audio file. In the example: **mp3**.

Possible error returns are:

- **API_FILE_NOT_FOUND:** Appears if the specific file to be exported and emailed cannot be found.
- **API_INVALID_MESSAGE_FORMAT:** Appears if parameters are missing or the request is not formatted correctly.

Export Multiple Recorded Audio Files as a ZIP File

MULTIEXPORT is unique to the WebAPI. This function type combines multiple audio files into a ZIP file (post-recording) and exports that file. It only uses these values:

- **IDENT:** These are comma-separated call record IDs.
- **MEDIA_FORMAT:** File format of the exported audio file.

The MULTIEXPORT call should be formatted in the following manner:

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

This example URL was constructed from the following components:

- **Protocol:** http
- **Host Address:** ccserveraddress
- **TCP Port:** 2012
- **API Page:** WebAPI.aspx
- **Function Type:** multiexport
- **Parameters:**
 - **REQUESTID:** Unique identifier for the transaction generated by the user making the WebAPI call. The value must be an integer. In the example: **10482345**.
 - **IDENT:** Required parameter specifying the recordings to be combined. In the example: comma-separated call record IDs **10121967,101219168,10121969**.
- **MEDIA_FORMAT:** File format of exported audio file. In the example: **mp3**.

Possible error returns are:

- **API_FILE_NOT_FOUND:** Appears if the specific file to be exported and emailed cannot be found.
- **API_INVALID_MESSAGE_FORMAT:** Appears if parameters are missing or the request is not formatted correctly.

Event Interface

Uptivity WFO can 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 *Uptivity WFO 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 command must be sent by any client application to initiate event monitoring. The internal Uptivity WFO 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 Uptivity WFO 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 Uptivity WFO 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 Uptivity WFO 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 Uptivity WFO 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 Uptivity WFO 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 Uptivity WFO 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 Uptivity WFO has resumed recording a call after a 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>
```


Document Revision History

| Revision | Change Description | Effective Date |
|----------|----------------------------------|----------------|
| 0 | Initial version for this release | 2015-09-01 |