![DVSAnalytics]

Encore Web API Web Service

Encore Workforce Optimization Solution
Version 7.1 or later

June 20, 2020

encore

For Dealer and Customer Use Only

DVSAnalytics, Inc.
17255 N. 82nd Street, Suite 120, Scottsdale, AZ 85255
Phone: (480) 538-2020 • Fax: (480) 538-2021
Email: info@DVSAnalytics.com
Web: www.DVSAnalytics.com

TABLE OF CONTENTS

Introduction

The Encore Web API web service allows third-party software products to issue commands to the recorder, associate data with a recording, retrieve a URL link to a recording and send events to the Encore system. For example, a CRM application can send a customer account number, patient ID, or order number to be stored in the database record associated with the current call recording. It can also retrieve the URL link to the recording and store the link in a field in the CRM record so the recording can be played from the CRM record. There is no limit to the information that can be sent for a single recording.

The Web API web service is configured as either an HTTP or HTTPS web service. If the web service is accessed from an application outside the corporate network, then the web service must be configured as HTTPS for security reasons.

If the Encore system is configured for screen recording, then any request sent for an audio recording will also impact the screen recording.

If the system is configured for desktop recording, then only desktop recording requests impact the desktop recording.

Getting Started

The Encore system needs to be running Encore version 7.1 or later. Contact your Encore system administrator for the Web API web service URL.

If your Encore system has not been installed yet but you would like to view the web service WSDL, use the following URL: https://ecapturews.dvsanalytics.com/WebAPI/ECAPI.svc?wsdl.

For each request there is a response that must be received before the next request is sent. All requests have the following parameters in the response message:

RESPONSE MESSAGE

PARAMETER NAME	TYPE	DESCRIPTION
responseCode	Int	This contains the response code. Successful response is **100**.
responseMessage	String	This contains the text string associated with the response code. Successful response is **Command executed as expected**.

For more information on specific response codes, see "Appendix C: Response Codes" on page 20.

If a request generates a response with additional parameters, the response message is documented in the request section.

Initialization Requests

For all requests related to audio and screen recording, the Encore unit and port must be specified. The requests that require the unit and port are:

- AssignField
- CancelRecording
- GetField
- GetURL
- PauseRecording
- PortStatus
- PortStatusEX
- ResumeRecording
- StartRecording
- StopRecording

The application using the Web API web service must use either the user's extension or Windows login ID to obtain the unit and port assigned to that user.

- If the user's **extension** is available, use the **InitStation** request to get the Encore unit and port.
- If the user's **Windows login ID** is available, use the **InitUser** request to get the Encore unit and port. This is the easiest method if the application is a Windows application running on the user's machine.

If the application does not have access to either the user's extension or Windows login ID, then a lookup table must be created and maintained that maps the information in the application with either the user's extension or Windows Login ID. For example:

- If the application has a **User ID**, create a lookup table that maps the **User ID** to the **Windows Login ID** and use the **InitUser** request to obtain the Encore unit and port.
- If the application has an **ACD Login ID**, create a lookup table that maps the **ACD Login ID** to the **extension** and use the **InitStation** request to obtain the Encore unit and port.

Initialization requests are not used for desktop recording requests. See the "Desktop Recording Requests" section on page 11 for more information.

InitStation

This request should be issued when the application starts or anytime the extension number changes. It does not impact the recording state.

PARAMETER NAME	TYPE	DESCRIPTION
Extension	String	Specifies the user's phone extension. This is a required field.

RESPONSE MESSAGE

PARAMETER NAME	TYPE	DESCRIPTION
responseCode	Int	This contains the response code. Successful response is **100**.
responseMessage	String	This contains the text string associated with the response code. Successful response is **Command executed as expected**.
Unit	String	Recording unit
Port	String	Recording port

InitUser

This request should be issued when the application starts. It does not impact the recording state.

PARAMETER NAME	TYPE	DESCRIPTION
Machine	String	Specifies the name of the computer the user is logged into. This is a required field. This is only informational data. If the machine name is not known, just use the Windows Login ID for this field.
User	String	Specifies the Windows Login ID. This is a required field.

RESPONSE MESSAGE

PARAMETER NAME	TYPE	DESCRIPTION
responseCode	Int	This contains the response code. Successful response is **100**.
responseMessage	String	This contains the text string associated with the response code. Successful response is **Command executed as expected**.
Unit	String	Encore recording unit
Port	String	Encore recording port

Recording Control Requests

Call control requests are only valid if the current recording state is valid for the request. For example, it isn't valid to stop a recording that isn't started. It is a good idea to always issue a **PortStatus** request to make sure the call control request is valid for the current recording state. See the "PortStatus" section on page 11 for more information on recording states.

Depending on the PBX integration, a unit and port can have a call on hold (Recording State is SUSPENDED) and an active call being recorded (Recording State is RECORD). If a unit and port has both a call on hold and an active call, all API requests affect the active call. If a unit and port has only a call on hold, all API requests affect that call. Refer to the Integration Guide for your PBX to determine if "SUSPENDED" is a valid recording state in your system.

CancelRecording

CancelRecording stops the audio/screen recording for the given unit and port, and discards the associated recording and data.

- Valid Recording States: RECORD, PAUSE, SUSPENDED
- Exit State on Success: NOT RECORDING

PARAMETER NAME	TYPE	DESCRIPTION
Unit	String	Specifies the recording unit. This is a required field.
Port	String	Specifies the recording port. This is a required field.

PauseRecording

PauseRecording places the audio/screen recording in a PAUSE state and stops recording for the provided unit and port. The data and recording file name(s) are saved in memory so the recording can be re-started. A ResumeRecording, StartRecording, StopRecording, or CancelRecording request must be received to change the PAUSE state.

Because a PauseRecording command is typically used to prevent recording of sensitive information it affects both the call on hold (Recording State is SUSPENDED) and the active call (Recording State is RECORD). Until a StartRecording, StopRecording, or CancelRecording request is received, neither call will be recorded, even if the call on hold is taken off hold.

If you wish to pause a recording for to protect sensitive data, such as credit card data, and the Encore system has both audio recording and desktop recording, you will need to send both a **PauseRecording** and a **DesktopPause** to ensure the data is protected.

- Valid Entry State: RECORD, SUSPENDED
- Exit State on Success: PAUSE

PARAMETER NAME	TYPE	DESCRIPTION
Unit	String	Specifies the recording unit. This is a required field.
Port	String	Specifies the recording port. This is a required field.

ResumeRecording

ResumeRecording restarts the audio/screen recording if it is in a PAUSE state and places the recording in a RECORD state for the provided unit and port.

- Valid Entry State: PAUSE, RECORD, SUSPENDED
- Exit State on Success: RECORD, SUSPENDED

PARAMETER NAME	TYPE	DESCRIPTION
Unit	String	Specifies the recording unit. This is a required field.
Port	String	Specifies the recording port. This is a required field.

StartRecording

StartRecording begins an audio/screen recording for the provided unit and port. If the recording state is PAUSE, it will re-start the recording.

- Valid Recording States: NOT RECORDING, PAUSE, SUSPENDED
- Exit State on Success: RECORD

PARAMETER NAME	TYPE	DESCRIPTION
Unit	String	Specifies the recording unit. This is a required field.
Port	String	Specifies the recording port. This is a required field.
maxRecordingTimeInSeconds	Int	Specifies the maximum recording time in seconds. Normally set to 0, which uses the system default. This is a required field.

StopRecording

StopRecording ends the audio/screen recording for the provided unit and port, and adds the associated data to the database.

- Valid Recording States: RECORD, PAUSE, SUSPENDED
- Exit State on Success: NOT RECORDING

PARAMETER NAME	TYPE	DESCRIPTION
Unit	String	Specifies the recording unit. This is a required field.
Port	String	Specifies the recording port. This is a required field.

Data Assignment Requests

AssignField

AssignField writes a value to the specified database column in the audio/screen recording record. AssignField requests do not write a value to the Desktop recording records.

- Valid Recording States: RECORD, PAUSE, SUSPENDED
- Exit State on Success: This request does not change the recording state.

PARAMETER NAME	TYPE	DESCRIPTION
Unit	String	Specifies the recording unit. This is a required field.
Port	String	Specifies the recording port. This is a required field.
fieldName	String	Specifies the database column name. This is a required field.
fieldValue	String	The value to store. If the value is an empty string, an error is returned, and the field is not updated. This is a required field.

LogEvent

LogEvent writes an event to the eCapture event table. If an audio recording is taking place, the system logs the recording identifier with the event. If no recording is taking place, the event will be logged without a recording identifier. Before using this request, contact DVSAnalytics Technical Support at **480.538.2020** or CustomerService@DVSAnalytics.com.

The recording state is not applicable to this request.

PARAMETER NAME	TYPE	DESCRIPTION
Unit	String	Specifies the recording unit. This is an optional field. If Unit and Port are not specified but the User is specified, the web service will use the value defined in User to look up the Unit and Port and populate these fields in the event record

PARAMETER NAME	TYPE	DESCRIPTION
		before the record is written to the database. If the Unit, Port, and User are left blank, there is no way to associate the record to a user or a recording.
Port	String	Specifies the recording port. This is an optional field. If Unit and Port are not specified but the User is specified, the web service will use the value defined in User to look up the Unit and Port and populate these fields in the event record before the record is written to the database. If the Unit, Port, and User are left blank, there is no way to associate the record to a user or a recording.
User	String	The windows user name. This is an optional field. If Unit and Port are not specified but the User is specified, the web service will use the value defined in User to look up the Unit and Port and populate these fields in the event record before the record is written to the database. If the Unit, Port, and User are left blank, there is no way to associate this event record to a user or a recording.
Machine	String	Name of the computer the user is logged into. This is an optional field.
eventTimestamp	String	The timestamp of the event. This is a required field.
eventType	String	This will specify what type of event is being logged. This is a required field.
Parm1	String	This field is optional. This is a 1024-character field that stores any string that should be stored with the event. This is an optional field.
Parm2	String	This field is optional. This is a 1024-character field that stores any string that should be stored with the event. This is an optional field.
Parm3	String	This field is optional. This is a 1024-character field that stores any string that should be stored with the event. This is an optional field.

Information Requests

GetField

GetField returns the value of the specified database column in the audio/screen recording record. GetField request does not return a value from the Desktop recording records.

- Valid Recording States: RECORD, PAUSE, SUSPENDED
- Exit State on Success: This request does not change the recording state.

PARAMETER NAME	TYPE	DESCRIPTION
Unit	String	Specifies the recording unit. This is a required field.
Port	String	Specifies the recording port. This is a required field.
fieldName	String	Specifies the database column name. This is a required field.

RESPONSE MESSAGE

PARAMETER NAME	TYPE	DESCRIPTION
responseCode	Int	This contains the response code. Successful response is **100**.
responseMessage	String	This contains the text string associated with the response code. Successful response is **Command executed as expected**.
fieldValue	String	The value of the field specified in fieldname. If fieldName is an invalid field or the field is not populated, an empty string is returned.

GetURL

This command returns a URL string that can be saved within the customer record of the third-party application as a link to the recording. When the URL is accessed, the Encore URL Player is launched and the recording starts playing. If a screen recording is associated with the audio recording and the default browser is Internet Explorer 11, the screen recording plays as well.

This request requires the URL Player to be configured on the system. For help configuring the URL Player, contact DVSAnalytics Technical Support at **480.538.2020** or CustomerService@DVSAnalytics.com.

- Valid Recording States: RECORD, PAUSE

- Exit State on Success: This request does not change the recording state.

PARAMETER NAME	TYPE	DESCRIPTION
Unit	String	Specifies the recording unit. This is a required field.
Port	String	Specifies the recording port. This is a required field.

RESPONSE MESSAGE

PARAMETER NAME	TYPE	DESCRIPTION
responseCode	Int	This contains the response code. Successful response is **100**.
responseMessage	String	This contains the text string associated with the response code. Successful response is **Command executed as expected**.
URL	String	The URL associated with the recording.

PortStatus

This command returns the state of the provided unit and port.

- Valid Recording States: Any state
- Exit State on Success: This request does not change the recording state.

PARAMETER NAME	TYPE	DESCRIPTION
Unit	String	Specifies the recording unit. This is a required field.
Port	String	Specifies the recording port. This is a required field.

RESPONSE MESSAGE

PARAMETER NAME	TYPE	DESCRIPTION
responseCode	Int	This contains the response code. Successful response is **100**.
responseMessage	String	This contains the text string associated with the response code. Successful response is **Command executed as expected**.
portStatus	String	If the responseCode = 100, then this contains the state of the port. Possible values:

PARAMETER NAME	TYPE	DESCRIPTION
		• **RECORD** – Recording • **NOT RECORDING** – Recording has stopped or never started • **PAUSE** – The recording is paused • **SUSPENDED** – The recording is suspended because the call is on hold. Refer to the Integration Guide for your PBX to determine if "SUSPENDED" is a valid recording state in your system. If the responseCode <> 100, this field is not returned.

PortStatusEX

This command returns the state of the provided unit and port and if the state is RECORD, PAUSE, or SUSPENDED, it will return the duration of the current state. Each change of state resets the duration to zero.

- Valid Recording States: Any state
- Exit State on Success: This request does not change the recording state.

PARAMETER NAME	TYPE	DESCRIPTION
Unit	String	Specifies the recording unit. This is a required field.
Port	String	Specifies the recording port. This is a required field.

RESPONSE MESSAGE

PARAMETER NAME	TYPE	DESCRIPTION
responseCode	Int	This contains the response code. Successful response is **100**.
responseMessage	String	This contains the text string associated with the response code. Successful response is **Command executed as expected**.
portStatus	String	If the responseCode = 100, then this contains the state of the port. Possible values: • **RECORD** – Recording • **NOT RECORDING** – Recording has stopped or never started • **PAUSE** – The recording is paused • **SUSPENDED** – The recording is suspended

PARAMETER NAME	TYPE	DESCRIPTION
		because the call is on hold. Refer to the Integration Guide for your PBX to determine if "SUSPENDED" is a valid recording state in your system.
		If the responseCode <> 100, this field is not returned.
stateDuration		If the responseCode = 100, then this contains the duration of the current state. If the current state is NOT RECORDING, this parameter is 0.
		If the responseCode <> 100, this field is not returned.

Desktop Recording Requests

Desktop recording is a feature of the Encore system that allows a user's desktop to be recording continuously in ten minute segments while the user is logged into the desktop. Since it is not tied to an audio recording, it doesn't have an associated recording unit and port. Instead, desktop recording require the Windows login ID to be specified for all desktop requests.

If the application does not have access to the user's Windows login ID, then a lookup table must be created and maintained that maps the information in the application with the user's Windows Login ID. For example:

- If the application has a user's **User ID**, create a lookup table that maps the user's **User ID** to the user's **Windows Login ID** and use the **Windows Login ID** from the lookup table for desktop recording requests.
- If the application has a user's **Email Address**, create a lookup table that maps the **Email Address** to the **Windows Login ID** and use the **Windows Login ID** from the lookup table for desktop recording requests.

DesktopPause

DesktopPause stops and saves the desktop recording for the provided Windows Login ID. Since recordings are in ten minute segments, a DesktopPause stops the current segment and does not start the next ten minute segment until a DesktopResume is received.

If you wish to pause a recording to protect sensitive data, such as credit card data, and the Encore system has both audio recording and desktop recording, you will need to send both a **PauseRecording** and a **DesktopPause** to ensure the data is protected.

- Valid Entry State: RECORD
- Exit State on Success: NOT RECORDING

PARAMETER NAME	TYPE	DESCRIPTION
User	String	Specifies the user's Windows login ID. This is a required field.

DesktopResume

DesktopResume starts the desktop recording for the provided Windows Login ID. Since recordings are in ten minute segments, a DesktopPause stops the current segment and does not start the next ten minute segment until a DesktopResume is received.

- Valid Entry State: NOT RECORDING
- Exit State on Success: RECORD

PARAMETER NAME	TYPE	DESCRIPTION
User	String	Specifies the user's Windows login ID. This is a required field.

Appendix A: Simulator

DVSAnalytics has a test Web API web service that can be used for testing your application without impacting your production system: https://ecapturews.dvsanalytics.com/WebAPI/ECAPI.svc.

If you have an issue working with the test system, please contact DVSAnalytics Technical Support at **480.538.2020** or CustomerService@DVSAnalytics.com.

The test system uses the simulation mode of the Web API. This mode tries to simulate a real system as much as possible. The requests respond as follows:

Initialization Requests

- **InitStation** – As long as the request is properly formed, it returns a successful response with a unit of 1 and a port from 1 up to 288. Each time an InitStation is sent with a different extension, the port number is the next available port. If an InitStation request is sent for extension "1234", it will return a unit of 1 and a port number that is the next available port number, for example, port 3. For that day, any time an InitStation request is sent for extension "1234", the unit and port returned is 1 and 3 respectively. Each night the simulator is reset back to port 1.

- **InitUser** – Just like the InitStation, as long as the request is properly formed, it returns a successful response with a unit of 1 and a port from 1 up to 288. Each time an InitUser is sent with a different User ID, the port number is the next available port. The computer name does not cause the port to be different. If an InitUser request is sent for User ID "smith", it will return a unit of 1 and a port number that is the next available port number, for example, port 4. For that day, any time an InitUser request is sent for User ID "smith", the unit and port returned is 1 and 4 respectively. Each night the simulator is reset back to port 1.

Recording Control Requests

- **CancelRecording** – It returns a successful response.

- **PauseRecording** – If the port status is "RECORD", then it returns a successful response. Otherwise, it returns a response code of 608, port not recording. If your application is only pausing and resuming a recording, you will always get a response code of 608 unless you put in a start recording request in the application (before the pause) while you are testing with the simulator.

- **ResumeRecording** – If the port status is "PAUSE", then it returns a successful response. Otherwise, it returns a response code of 627, port not in pause mode. If your application is only pausing and resuming a recording, you will always get a response code of 627 unless you put in a start recording request in the application (before the pause) while you are testing with the simulator.

- **StartRecording** – If the port status is "NOT RECORDING" or "PAUSE", then it returns a successful response. Otherwise, it returns a response code of 609, port already recording.

- **StopRecording** – If the port status is "RECORD", then it returns a successful response. Otherwise, it returns a response code of 608, port not recording.

Data Assignment Requests

- **AssignField** – If the port status is "RECORD" or "PAUSE", then it returns a successful response. Otherwise, it returns a response code of 608, port not recording.
- **LogEvent** – As long as the request is properly formed, it returns a successful response.

Information Requests

- **GetField** - If the port status is "RECORD" or "PAUSE", then it returns a successful response and populated the field Value with "123". Otherwise, it returns a response code of 608, port not recording.
- **GetURL** – If the port status is "RECORD" or "PAUSE", then it returns a successful response. Otherwise, it returns a response code of 608, port not recording. The URL returned looks something like:

  https://encore.dvsanalytics.com/encore/urlplayer.aspx?fldCallIndex=429504.2640&siteid=1

  While the URL is properly formed, it is not a working URL.

- **PortStatus** – As long as the request is properly formed, it returns a successful response. It remembers the last command you sent, so if you issue a StartRecording and then select PortStatus, it returns "RECORD". If you issue a StopRecording and then select PortStatus, it returns "NOT RECORDING".
- **PortStatusEX** – As long as the request is properly formed, it returns a successful response. It remembers the last command you sent, so if you issue a StartRecording and then select PortStatus, it returns "RECORD". If you issue a StopRecording and then select PortStatus, it returns "NOT RECORDING". If the state is "NOT RECORDING", the stateDuration is 0. If any other state is returned, the stateDuration is 20.
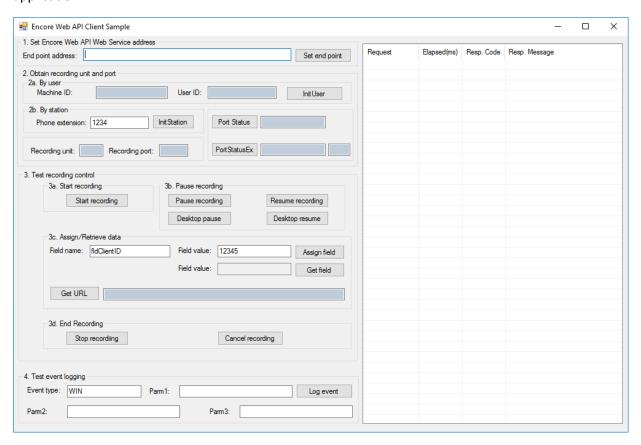
Desktop Recording Requests

- **DesktopPause** – As long as the request is properly formed, it returns a successful response.
- **DesktopResume** – As long as the request is properly formed, it returns a successful response.

Appendix B: Sample Application

To install the sample application, copy the **Client Install** directory to the target machine. Run **setup.exe** to install the application. The installation puts a shortcut to the application in **Program Files | dvsAnalytics**.

Microsoft .NET Framework 4 Client Profile or the full .NET Framework 4 is required to run the application.

Section 1 - Set Encore Web API Web Service Address

The first time you open the application, the **End point address** is blank. This specifies the URL of the Encore Web API that receives the requests from the sample application. This must be populated before any requests are sent. Once populated, click the **Set end point** button. You can use the URL associated with the Web API of your Encore system or you can use the test URL provided by DVS: **https://ecapturews.dvsanalytics.com/WebAPI/ECAPI.svc**. If you use the URL provided by DVS, it is in simulation mode. You should review "Appendix A: Simulator" on page15 for responses when in simulation mode.

Section 2 – Obtain Recording Unit and Port

After you set the **End point address**, specify the **Recording unit** and **Recording port**. There are three ways to specify the unit and port:

2a. By User

The application has populated the **Machine ID** field with the name of the machine where the sample application is running, and the **User ID** field with your **Windows Login ID**. The **User ID** field must match an entry in the **Computer Login ID** field in an Encore **Recorded User** record. The Machine ID is used for informational logging and does not need to match information in the recording account.

You can overwrite the **User ID** with an entry that matches an existing test recording account, or your **Encore System Administrator** can create a test recording account for you using your **Windows Login ID**.

When you select the **InitUser** button, the request is sent to the Web API. If successful, the **Recording unit** and **Recording port** fields are populated with the values from the response. If not successful, a 0 is written to both fields. Each response is written to the **Result** grid.

2b. By Station

The **Phone extension field** must match an entry in the **Extension** field in an Encore **Recorded User** record.

You can enter a value that matches an existing recording account or your **Encore System Administrator** can create a test recording account for you using your phone extension number.

When you select the **InitStation** button, the request is sent to the Web API. If successful, the **Recording unit** and **Recording port** fields are populated with the values from the response. If not successful, a **0** is written to both fields. Each response is written to the **Result** grid.

Manual Entry

If you know the **Recording unit** and **Recording port** you want to use for testing, you can manually enter the values. You should select the **Port Status** button to verify the unit and port you entered are valid.

Port Status Button

Sends a **PortStatus** request and displays the port status in the field to the right of the button.

Port StatusEx Button

Sends a **PortStatusEx** request and displays the port status and status duration in the fields to the right of the button.

Section 3 - Test Recording Control

The buttons in this section all correspond to one of the available API requests and most are self explanatory. For more details, see the associated section on the specific request.

- **Start recording** button – Sends a StartRecording request.
- **Pause recording** button – Sends a PauseRecording request.
- **Resume recording** button – Sends a ResumeRecording request.
- **Desktop pause** button – Sends a DesktopPause request. When testing the Desktop pause button, the **User ID** must match an entry in the **Computer Login ID** field in an Encore **Recorded User** record.

- **Desktop resume** button – Sends a DesktopResume event. When testing the Desktop resume button, the **User ID** must match an entry in the **Computer Login ID** field in an Encore **Recorded User** record.

- **Assign field** button – Sends an AssignField request. You must enter a valid field name. If using the DVS-provided test Web API address, the following field names can be used:
  - fldCampaign (varchar 24)
  - fldClientID (varchar 16)
  - fldCustomerAccountNum (varchar 64)
  - fldCallCategory (varchar 24)
  - fldAccountingNum (smallint)

- **Get field** button – Sends a GetField request. You must enter a valid field name. If using the DVS-provided test Web API address, the following field names can be used:
  - fldCampaign (varchar 24)
  - fldClientID (varchar 16)
  - fldCustomerAccountNum (varchar 64)
  - fldCallCategory (varchar 24)
  - fldAccountingNum (smallint)

- **Get URL** button – Sends a GetURL request. If the unit and port is currently being recorded, the URL of the recording is populated in the entry field. You cannot use the URL until the recording has stopped and the record is written to the database. To use the URL, open an Internet Explorer 11 browser and paste the URL into the browser.

- **Stop recording** button – Sends a StopRecording request.

- **Cancel recording** button – Sends a CancelRecording request.

Section 4 - Test Event Logging

- **Log event** button – Sends a LogEvent request.

Appendix C: Sample Source Code

The SDK provides the source code to the sample C# application name **Encore Web API Client Sample** that allows you to test all the API requests. The Microsoft Visual Studio 2017 project files are in the **Client Sample Code** directory. Since the application is C#, Microsoft .NET Framework 4 Client Profile or the full .NET Framework 4 is required to run the application.

Appendix D: Response Codes

RESPONSE CODE	RESPONSE MESSAGE	DESCRIPTION
100	Command executed as expected	Request successful.
606	Error: Recording unit does not exist	Request unsuccessful, the **Unit** parameter received is not valid.
608	Warning: Port was not recording	Request unsuccessful, a request to stop, cancel, or pause recording is received and the specified unit and port is not recording.
609	Warning: Port already recording	Request unsuccessful, a request to start recording is received and the specified unit and port is already recording.
610	Error: Issue creating transaction record	Request unsuccessful, a start recording request is received and a problem occurred creating the associated transaction file.
611	Error: Port does not exist	Request unsuccessful, the **Port** parameter received is not valid.
616	Error: Port not in record or pause mode	Request unsuccessful, a request to stop or cancel recording was received and the specified unit and port is not recording or paused.
618	Error: Start recording failed, all licenses in use	Request unsuccessful, a request to start recording is received and all licenses are in use.
621	Error: Recording Engine is not connected to the recording server	Request unsuccessful, a request is received, and an internal communication error occurred between the Recording Engine and the Recording Server.
626	Error: Recording Engine determined there are no ports available on a Recording Server that supports load balancing.	Request unsuccessful, a request to start recording is received and a Recording Server without load balancing is at 100% utilization, or all Recording Servers in a load balancing group are at 100% utilization.

RESPONSE CODE	RESPONSE MESSAGE	DESCRIPTION
651	Error: ECAPI Web Service is not connected to the recorder	Request unsuccessful, a request is received, and the web service is not configured correctly so it cannot send a request to the recorder.
652	Error: Field name and field value must be provided	Request unsuccessful, an assign field request is received and either the **fieldName** or **fieldValue** are empty.
653	Error: An unexpected error occurred while processing request	Request unsuccessful, a request is received and an unexpected error occurs while processing the request.
654	Error: Both User name and machine name must be provided	Request unsuccessful, an **InitUser** request is received and either the **userName** or **machineName** are empty.
655	Error: No recording port is defined for the specified machine name and user name	Request unsuccessful, an **InitUser** request is received and no recording port is defined for the machine name and user name combination.
656	Error: More than one recording port is defined for the specified machine name and/or user name	Request unsuccessful, an **InitUser** request is received and there is more than one port defined for the user name and machine name combination.
657	Error: EventType is not provided	Request unsuccessful, a **LogEvent** request is received and the **EventType** field, which is required, is not provided.
658	Error: URL player address is not defined in ECAPI Web service's Web.config	Request unsuccessful, a **GetURL** request is received and the address for the URL player is not specified in the Web.config file.
659	Error: Extension must be provided	Request unsuccessful, an **InitStation** request is received and an extension is not provided.
660	Error: No recording port is defined for the specified station	Request unsuccessful, an **InitStation** request is received and the recording port is not defined for the station.
661	Error: More than one recording port is defined for the specified station	Request unsuccessful, an **InitStation** request is received and more than one recording port is defined for the station.
662	Error: User name must be provided	Request unsuccessful, a **DesktopPause** or **DesktopResume** request is received and the **userName** is empty.

RESPONSE CODE	RESPONSE MESSAGE	DESCRIPTION
663	Error: Desktop recording is not possible	Request unsuccessful, a **DesktopPause** or **DesktopResume** request is received and Desktop recording is not allowed or the recording client ID is not available.
664	Error: Desktop Pause recording failed	Request unsuccessful, a **DesktopPause** request is received but a failure occurred. The current desktop recording segment is not stopped.
665	Error: Desktop Resume recording failed	Request unsuccessful, a **DesktopResume** request is received but a failure occurred. A new desktop recording segment is not started.
801	System Error: Error writing to disk (disk is full, path does not exist, no write permissions, etc.)	Request unsuccessful, a request is received and the system experiences an error while writing to the disk due to a full disk, non-existent path, no write permissions, etc.