

## Application Programming Interface (API)

### Developing applications and integrations communicating with the NICE Recording Solutions

The NICE Recording API is part of the NICE Recording Open Recording Environment (CORE). This environment allows 3<sup>rd</sup> party vendors and system integrators to develop applications that closely integrate with the heart of the NICE Recording Solutions.

#### Recorder API

With the NICE Recording System API, also known as the Recorder API, you can develop applications that can communicate with the NICE Recording System. Using a set of pre-defined API calls, your application can receive and send real-time recording events provided by the NICE Recording Tapping Boards (Parrot-DSC API) and can store data to and retrieve data from the embedded database of the NICE Recording Solutions.



Example Applications that can use the Open Call Controller Interface (OCCI)

#### Functions

The Recorder API provides system integrators with a comprehensive set of routines, protocols and tools for building software applications on NICE Recording's recording systems. It is designed to enable application developers to access the NICE Recording System in a conditioned way. For example, the Recorder API supports addition of custom data to call database records and creation of 3<sup>rd</sup> party search and replay applications.

The Recorder API is a client/server application. The server component resides on the NICE Recording System. The connected client components reside on client computers in the 'recorder network'.

The available functions via the Recorder API are:

##### Common functions

- Retrieving user, channel\*, call data information
- Registering and un-registering of free-seating users
- Retrieving audio from the recorder

##### Access to historical data

- Powerful and flexible search capabilities
- Access to all available call data

## Modify parts of historical data

- Modify parts of the stored information

## Real-time functions

- Receiving start and stop events\*
- Receiving call key events
- Start/stop recording on demand\*

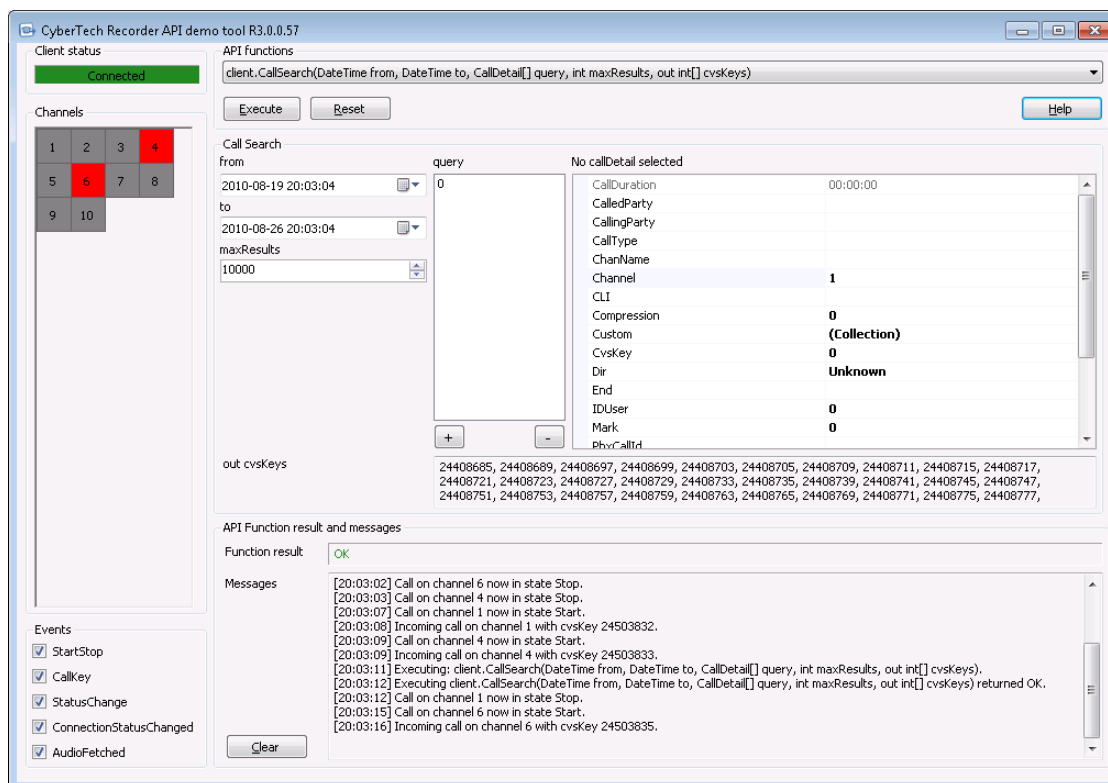
Access to the API functions is granted on basis of the user permissions configured in the connected NICE Recording system.

**\*) Note:** The following API functions are only available for recorder channels in the same system that runs the Recorder API server component. (Stand Alone systems):

- Detailed channel information (in-sync/recording)
- Real-time start and stop events
- Starting and stopping recording

## Tools

The Recorder API comes with testing and demonstration application. This application allows you to run all the available Recorder API functions and check the results. When developing applications, this tool is also helpful for learning about the API functions behaviour.



## API Functions

### Coverages

All rights are controlled by a single access right in the “Call Permissions” section in the recording system. Coverage is assigned to this right either directly for the user, or inherited from the user’s group. The recorder API coverages can be applied on 3 types of information:

1. Call information
2. User information
3. Channel information

Call information and the applied coverages have impact on the following API functions:

- Client.GetCallDetails()
- Client.BulkRetrieve()
- Client.GetCall()
- Client.MarkCall()

- Client.GetCallAudio()
- Client.CallSearch()
- Client.CallUpdate()

User information and the applied coverages have impact on the following API functions:

- Client.GetUserProfile()
- Client.GetUserInfo()
- Client.GetUserID()

- Client.RegFreeSeat()
- Client.UnRegFreeSeat()

Channel information and the applied coverages have impact on the following API functions:

- Client.GetAllChannels()
- Client.GetChannelInfo()
- Client.GetChannelID()

- Client.StartCallRecord()
- Client.StopCallRecord()

Some functions and properties are not impacted by coverages:

- Client.IsConnected
- Client.Disconnect()
- Client.GetWindowsUser()
- Client.Status()
- Client.GetChannel()

- Client.GetUserProfile()
- Client.BaseCustom
- Client.GetAvailFields()
- Client.MaxKeysPerBulkRetrieve
- Client.MaxResultsPerCallSearch

## Searching for calls

The method overloads for `CallSearch()` can be used for searching for calls in the recording system. Two overloads exist; one with and one without options. The `CallSearch()` overload without the option can be regarded to be the same as calling:

```
client.CallSearch(from,
                 to,
                 query,
                 out cvsKeys,
                 new CallSearchOption[]
                 {
                     new PagingCallSearchOption(0, maxResults),
                     new SortingCallSearchOption (
                         CallDetailFieldSelection.Start,
                         SortDirection.Ascending),
                     new CommandTimeoutCallSearchOption(30)
                 });
```

## Updating calls

The following pieces of information about a recorded call can be updated:

- Mark
- Remarks
- Custom data (CVSCXX extra database fields in the Recording System)

- `Client.CallUpdate()`
- `Client.MarkCall()`

## Installation & Documentation

The installation of the Recorder API is easy and consists of three files:

- NICE Recording.RecorderAPI.dll
- NICE Recording.RecorderAPI.Forms.Controls.dll
- NICE Recording.dll

Two types of documentation are provided with the recorder API client. The first is a CHM help file in the same style as the Microsoft documentation. The second is a set of XML Documentation files that, for development, can be installed with the assemblies. This allows visual studio to generate help with IntelliSense.

## More information

More information on the NICE Recording Recorder API is available on the NICE Extranice



*Sede Milano:* 20132 MILANO - Via Crescenzago, 105 Tel. (02) 26.111.942 r.a. Fax (02) 26.111.937 e-mail: [sales@resnet.it](mailto:sales@resnet.it)  
*Filiale Roma:* 00131 ROMA - Via Roiate, 7 Tel. (06) 41.91.633 Fax (06) 41.31.467 e-mail: [resrm@resnet.it](mailto:resrm@resnet.it)

[www.resnet.it](http://www.resnet.it)