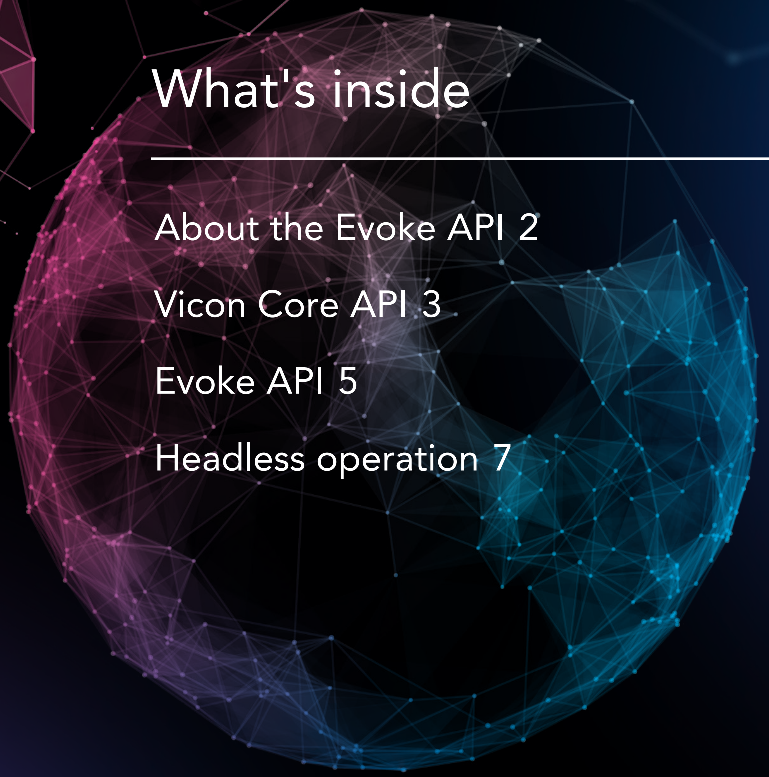
E V O K E

O R I G I N

VICON EVOKE API & AUTOMATION



What's inside

A RANGE OF LOCATION BASED VR PRODUCTS BY VICON

About the Evoke API

Evoke includes an extensive Application Programming Interface (API) for remote control, system monitoring and third-party application integration.

The API has two parts:

- Vicon Core API, page 3
- Evoke API, page 5

For information on running entirely via the API, without the Graphical User Interface (GUI), see Headless operation, page 7.

Vicon Core API

The Vicon Core API provides the core functionality for client/server communication, and schemas that define the types, functions and callbacks supported by the Vicon application. It is a generic JSON-based protocol operating over TCP.

Evoke opens the server socket on port 52800 by default. The port number can be specified by launching Evoke with a command-line argument:

```
C:\Program Files\Vicon\Evoke1.4\Evoke.exe --terminal-port=52800
```

A sample exchange may look like this (debug logging has been turned on for the ViconCoreAPI category, so all messages sent and received are logged):

ViconCoreAPI client-server exchange

```
2019.09.13 13:42:38.123 +1  Default ViconCoreAPI    Starting Vicon Core API server on port 52800...
2019.09.13 13:42:38.456 +1  Default ViconCoreAPI    Opened connection to 127.0.0.1:53926
2019.09.13 13:42:38.629 +1  Debug   ViconCoreAPI    Sending to 127.0.0.1:53926 [4,0][]
2019.09.13 13:42:38.631 +1  Debug   ViconCoreAPI    Received from 127.0.0.1:53926 [Terminal.AppInfo,5] []
2019.09.13 13:42:38.632 +1  Debug   ViconCoreAPI    Sending to 127.0.0.1:53926 [5,0]["Evoke","1.4.0.0","0"]
2019.09.13 13:42:38.645 +1  Debug   ViconCoreAPI    Received from 127.0.0.1:53926 [ApplicationServices.Shutdown,6] []
2019.09.13 13:42:38.730 +1  Debug   ViconCoreAPI    Sending to 127.0.0.1:53926 [6,0][""]
```

This trace shows two commands sent to the server as JSON-encoded strings, in the format `[SchemaName,CommandId] [Data]`:

- `Terminal.AppInfo` This is a built-in Vicon Core API command, which returns the current application information.
- `ApplicationServices.Shutdown` This is an Evoke API command that safely closes down the application, releasing all resources and writing settings files as required.

Each command is assigned a unique number by the client, and this number is sent back by the server with the reply, with the format `[CommandId,ResultCode][Data]`. This enables the client to match replies to the correct command.

Evoke API

The Evoke API is a client library that facilitates access to the services provided by Evoke. It includes all the schemas supported by the application, and classes for the data types exchanged via API. It is also responsible for serializing user data into JSON for transmission to the Vicon Core API server, and deserializing the replies back into user data.

Evoke API is compatible with Python 2.7 and Python 3.7 and later.

Services currently provided are:

- **Application services** - system file management, license information and application shutdown
- **Basic object services** - management of basic objects
- **Camera calibration services** - camera calibration operations and file management
- **Camera device services** - camera information, monitoring and control
- **Camera health services** - calibration assessment tools
- **Capture services** - data capture and control
- **Character from clusters services** - management of characters from clusters
- **Cluster device services** - smart cluster device information, monitoring and control
- **Log services** - logging control
- **Playback services** - data review and playback control
- **Radio device services** - radio device information, monitoring and control
- **Selection services** - selection and deselection of all device types

- **Smart object services** - management of smart objects
- **Subject services** - import and export of basic objects and overall tracking configuration
- **System health services** - data collection for system health reporting

Client libraries are supplied for C# and Python, located by default at:

```
C:\Program Files\Vicon\Evoke1.4\SDK
```

Instructions for use and full documentation are included in the packages. A limited dashboard application is provided for C#, together with a number of sample Python scripts.

Headless operation

Evoke may be operated entirely via API, in which case it may be advantageous to run without the Graphical User Interface (GUI). A separate executable is provided for this purpose:

```
C:\Program Files\Vicon\Evoke1.4\EvokeHeadless.exe
```

Headless operation may be more suitable for running Evoke as a service, and additionally benefits from slightly lower CPU usage without the overhead of running a GUI.

System and tracking setup must normally be completed with the GUI, as some setup functionality is not yet available via the API.

Evoke Headless shares the same settings and preferences as Evoke. In the event of an application crash, Evoke Headless is configured by default to send a report to Vicon, so that the problem can be investigated and fixed. The following command line options control this behavior:

- `--crash-handler-email=email.address@domain` - optional e-mail address for correspondence
- `--no-crash-upload` - disables crash report upload

If crash report upload is disabled, crash reports are instead written to:

```
C:\Users\Public\Documents\Vicon\ErrorReporting
```

These reports may be sent to Vicon at a later date with the OMG Error Reporting tool:

```
C:\Program Files\Vicon\Evoke1.4\OMGErrorReporting.exe --
upload-all
```