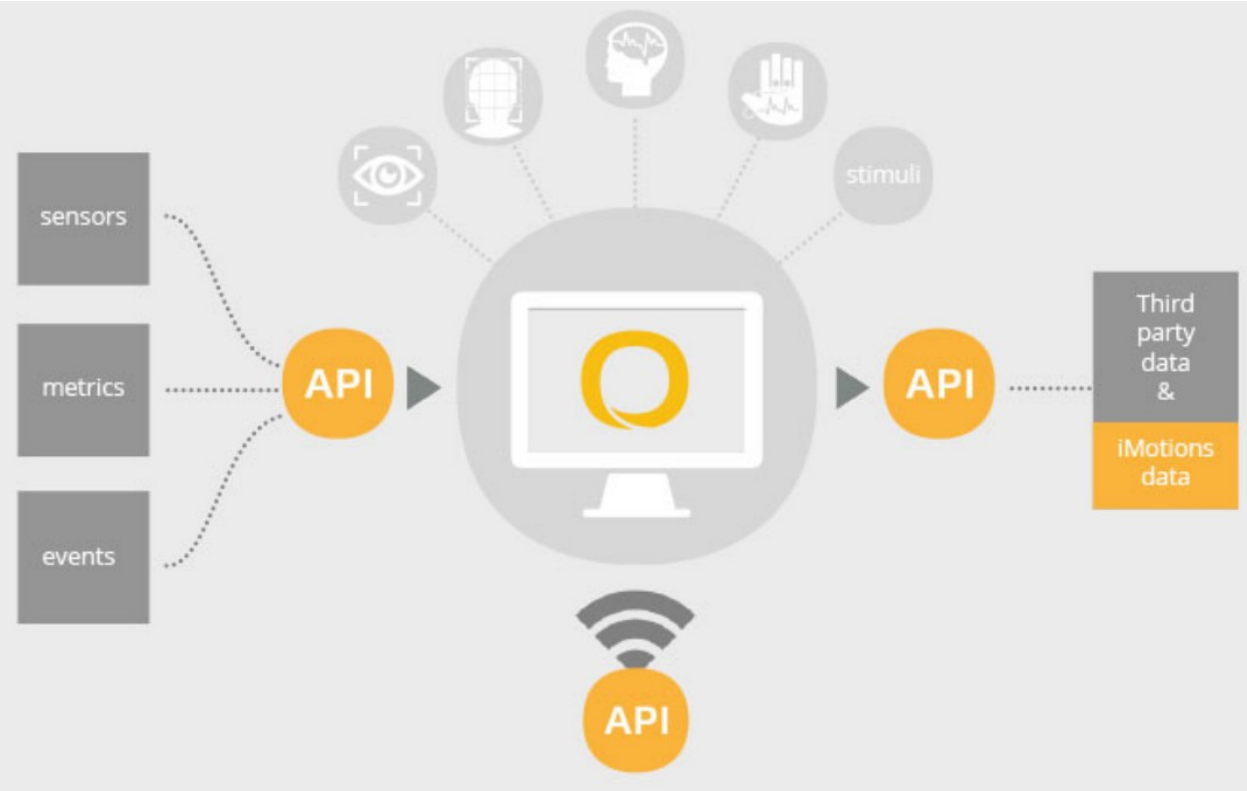
IMOTIONS
BIOMETRIC RESEARCH PLATFORM

iMotions Programmer's Guide



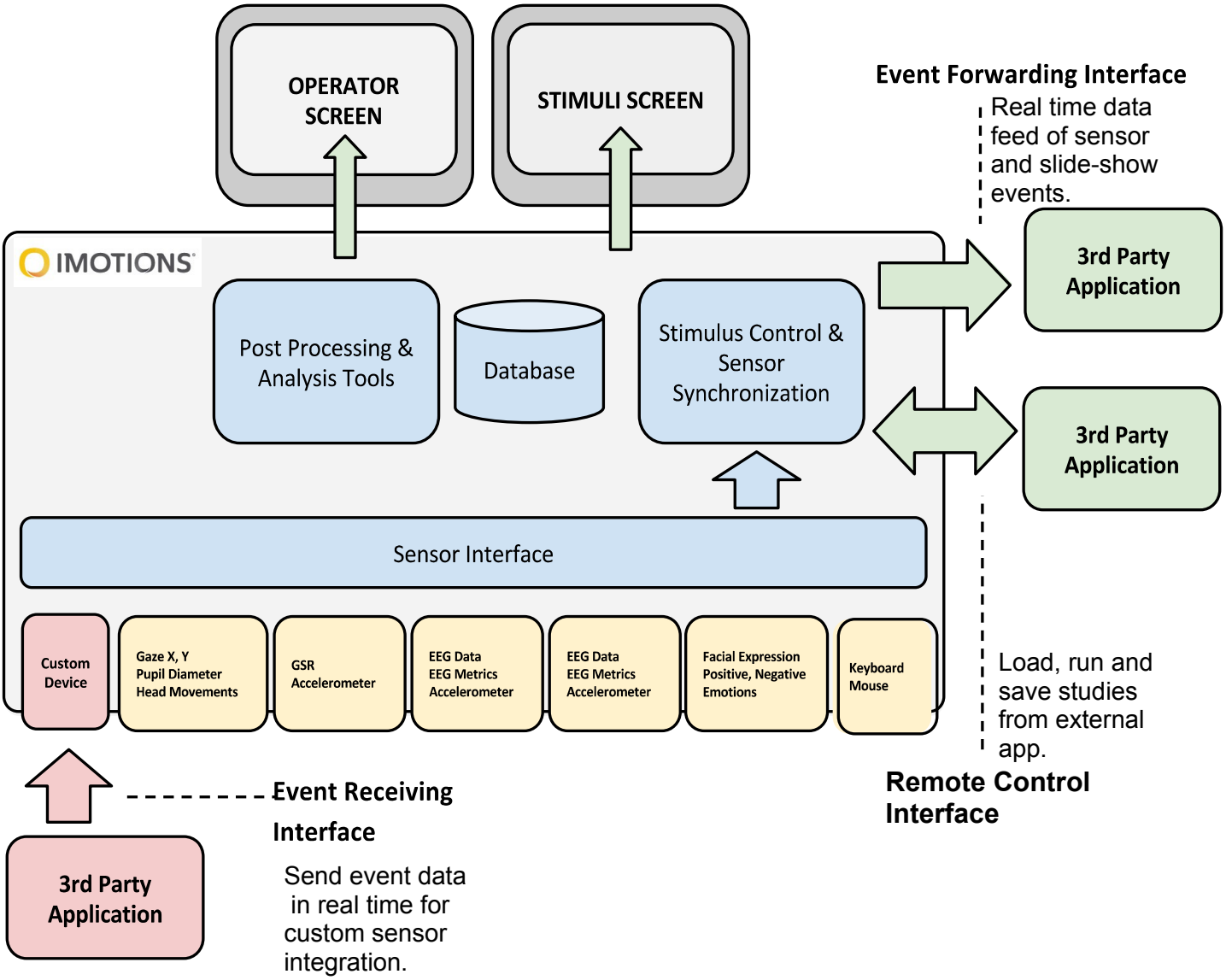
IMOTIONS
BIOMETRIC RESEARCH PLATFORM

Table of Contents

Change log

Version	Attention Tool Version Compatibility	Comment
1.0	5.1	Initial version
1.1	5.2.1	New RemoteControl API and gaze calibration events
1.2	5.3.2	Additional Remote API commands Export, NextSlide, Cancel
1.3	5.4	New segment marker events and auto-scene generation.
1.4	5.4	Add respondent as part of run command.
1.5	5.5	Launch face video post-processing
1.6	5.5.2	SAVE allows filtering of face videos. RUN allows disabling some UI dialogues.
1.7	5.7	QA metrics, timestamp in external events
1.8	5.7 6.0 6.1	Document changed to no refer to version in footer. New table added to show what versions of iMotions are supported.

Imotions Software Interfaces overview



iMotions Event Forwarding Interface

Overview

During a respondent test, the software collects various types of data from different sources. In a dual screen setup, this data is used to provide visualizations. It is also cached in memory and logged to the database at the end of the test. During study analysis, the data is loaded from the database and used to perform the analysis and paint various visuals e.g. gaze replay. The data can also be exported to a text file so that 3rd party applications can be used to perform custom analysis.

The event forwarding interface extends this model to allow third party applications to receive the data in real time as it is received by the iMotions system. When an event is received by the software e.g. a mouse click or an eyetracker sample, the application will now check to see if event forwarding is enabled. If it is, the event is serialized into a text string, and the string is forwarded as an event message. An external application can listen for these event messages.It would then process the event as part of performing some application specific task. Typically this would be used in a screen recording, where the application under test would be receiving the external events and adjusting behavior based on them e.g.

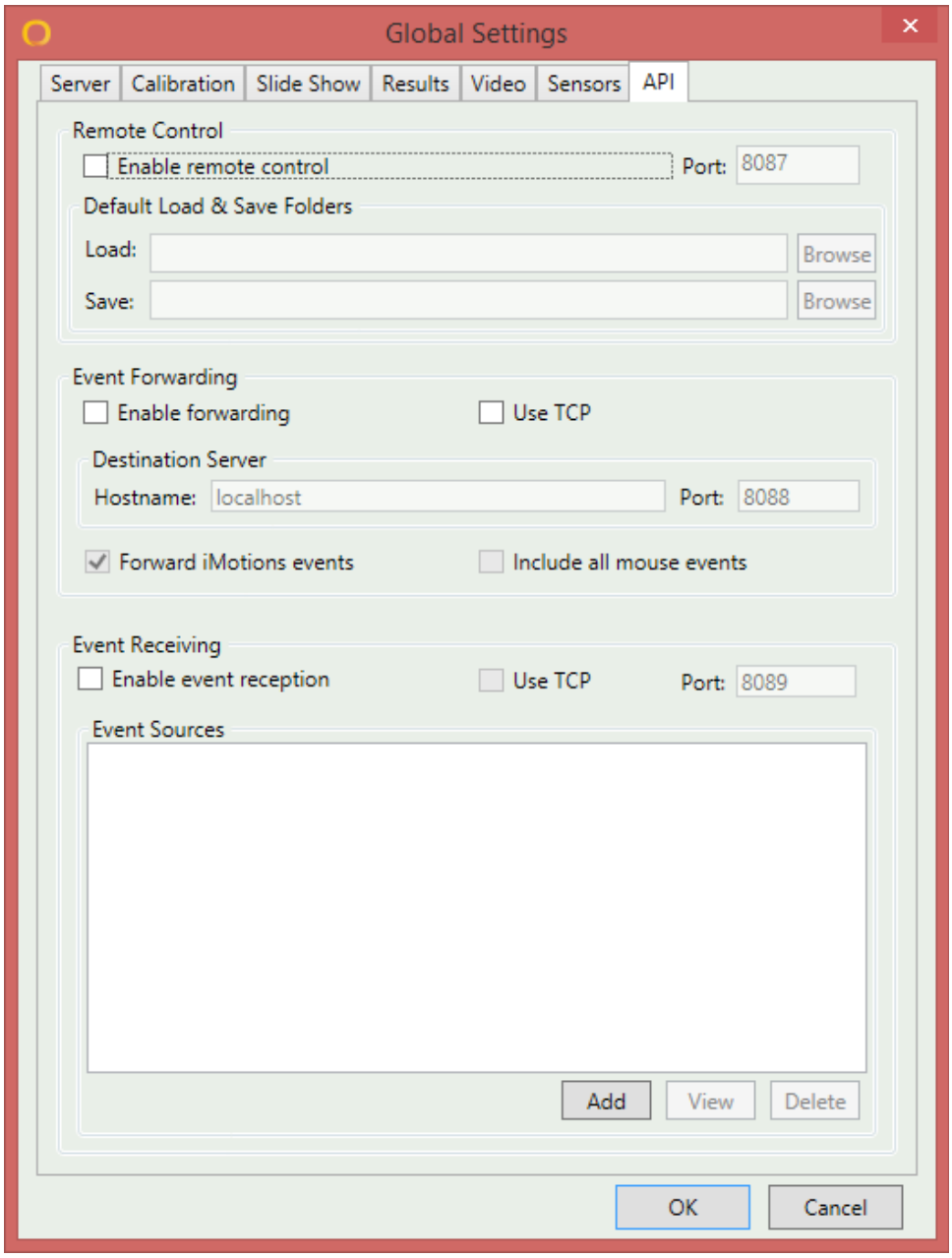
- The respondent could be shown different images based on where they looked in the previous image.
- If the last second of eyetracker data indicates that the respondent has looked away or closed their eyes, then an audible “wakeup” could be played to get the respondents attention.

Imotions software can send events to an external application using either TCP or UDP.

- When using TCP, the software acts as a server and listens for connections on a specific port. Once a connection has been established with a client, iMotions will forward all event messages to the client using this connection. The client application simply needs to read the event messages from the connection.
- When using UDP, iMotions will send each event message as a UDP datagram to a configured server/port combination.

Event Forwarding Setup

Event forwarding is enabled using the Global Settings dialog, within the API tab.



The middle section is used to configure the event forwarding feature.

- Enable/disable event forwarding and choosing whether you wish to communicate using TCP or UDP.
- The connection details. For UDP you will define the server and port to which the messages will be directed. For TCP, only the port number field is available and it will determine the port number on which AttentionTool will listen for connections.
- Check boxes allow general iMotions events to be enabled e.g. keyboard press events, mouse click events etc. Since mouse movement generates a lot of events, the default behavior is to suppress the sending of mouse move events. This can be overridden by checking 'Include all mouse events'.

Once event forwarding has been enabled, the Sensors tab should be use to individually mark those sensors for which events need forwarding. This allows for the forwarding of data for some sensors but not others e.g. you could choose to forward the GSR events but not the EEG data etc.

Common Event Format

All events are forwarded as UTF8 text strings. They all share a common header format, followed by event specific data fields. All fields are separated by a semi-colon character, and each record is terminated by a carriage-return, newline combination.

Field Name	Description	Example
Seq. Number	Incremented value for each event that is forwarded. Reset for each slide show.	`00000136`
Event source	Where the event has originated. This will either be the name of the sensor that sent the sample, or "AttentionTool" for internal events or PC input events.	`AttentionTool`
Sample Name	Identifies the type of event.	`Mouse`
Timestamp	Time since the start of the slideshow in ms	`7799`
Media Time	Position in the current video based stimuli in ms. This value is only applicable for videos and screen/web recordings. For other stimuli types, or when the video pipeline is not yet started or has finished -1 will be returned.	`6214`

E.g. the following shows some typical samples received from the UDP port. The common fields are underlined.

```
00000938;EyeTracker;EyeData;18332;15040;18319;1379;601;1379;601;…more fields…
00000939;AttentionTool;Keyboard;18348;15056;LControlKey, Shift
00000940;EyeTracker;EyeData;18352;15060;18339;1379;601;1379;601;…more fields…
```

In the last sample

1. Seq No. = `00000940`
2. Event Source = `EyeTracker`
3. Sample Name = `EyeData`
4. Timestamp = `18352`
5. Media Time = `15060`

Event Reference

Below are documented all the events that will be seen over the event forwarding interface. Events are only available when they are being collected as part of a slide show e.g. it is not possible to forward EEG events if you have not enabled them for data collection.

Slide show events

Gaze calibration and Slideshow events are always broadcast and indicate when a new test begins/ends and when individual slides in the slideshow begin/end.

Gaze calibration start

Event Source	`AttentionTool`	
Sample Name	`GazeCalibrationStart`	
Additional Fields	**Description**	**Example**
System time	Current date time according to the PC system clock formatted `YYYYMMDDhhmmssttt`	`20140513124944675`
Respondent	Name of the respondent being tested	`Anonymous 20-03-13 10h55m`
Gender	Respondent gender `MALE` or `FEMALE`	`MALE`
Age	Age of current respondent	`16`
Study	Name of the study that the respondent belongs to.	`Study 12-05-14 22h34m`

Gaze Calibration End

Event Source	`AttentionTool`	
Sample Name	`GazeCalibrationEnd`	
Additional Fields	**Description**	**Example**
System time	Current date time according to the PC system clock formatted `YYYYMMDDhhmmssttt`	`20140513124944675`
Calibration Status	Indication of status of the gaze calibration process. Failed, Aborted or Succeeded. If the status was succeeded, then details of the calibration result will follow, otherwise they will all be empty.	`Aborted`
Result Quality	Poor, Good, Excellent	`Excellent`

Result Points	Number of points used in the calibration	9
Result Offset	Average gaze position offset in pixels.	3
Result STD	Standard deviation of the offsets	3.292

Slide show start

Event Source	AttentionTool	
Sample Name	SlideshowStart	
Additional Fields	**Description**	**Example**
System time	Current date time according to the PC system clock formatted YYYYMMDDhhmmssttt	20130320105636662
Respondent	Name of the respondent being tested	Anonymous 20-03-13 10h55m
Gender	Respondent gender MALE or FEMALE	MALE
Age	Age of current respondent	16
Study	Name of the study	Study 12-05-14 22h34m

Slide show end

Event Source	AttentionTool	
Sample Name	SlideshowEnd	
Additional Fields	**Description**	**Example**
System time	Current date time according to the PC system clock formatted YYYYMMDDhhmmssttt	20130320105636662

Slide start

Event Source	AttentionTool	
Sample Name	SlideStart	
Additional Fields	**Description**	**Example**
System time	Current date time according to the PC system clock formatted YYYYMMDDhhmmssttt	20130320105636662
Stimulus name	Name of the current stimulus. If we are showing the light calibration	Happy Image

	slides at the start of a slideshow then this field will be empty.	

Slide type	Slide type within a stimulus. Possible values are one of the following:
	
	`BlackInterslide`
	`RandomInterslide`
	`TestImage`
		`TestImage`

Slide end

Event Source	`AttentionTool`	
Sample Name	`SlideEnd`	
Additional Fields	**Description**	**Example**
System time	Current date time according to the PC system clock formatted `YYYYMMDDhhmmssttt`	`20130320105636662`

Exposure Statistics

Event Source	`AttentionTool`	
Sample Name	`ExposureStatistics`	
Additional Fields	**Description**	**Example**
Study name	Name of the completed study	`Mytest`
Study ID	Unique ID for the study, used internally in iMotions software	`e0b8b75c-2be5-496a-bc72-2985a2948355`
Respondent Name	Name of the tested respondent	`Fred Smith`
Respondent ID	Unique ID for the respondent, used internally in iMotions software	`2361f9d4-31d1-4686-be4b-886c14b9728e`
Statistics	JSON string containing various statistics for the tested respondent.	See Below

The statistics field is a table of data containing various per-stimuli metrics calculated for the signals collected during the test.

The following example shows the statistics for a study with eyetracking, containing 2 stimuli named “Australian_Army_ceremonial_slouch_hat” and “Battleship”.

Note: the JSON below has been formatted for easier interpretation. The text included in any message will not include the new lines and other formatting.


```
[
    {
        "name":"Australian_Army_ceremonial_slouch_hat",
        "deviceSummaries":
        [
          {"displayName":"Eye tracking",
           "deviceName":"Eye tracking",
           "sampleRate":51.0,
           "quality":100.0,
           "iconType":"ET"}
        ]
     },
     {
         "name":"Battleship",
         "deviceSummaries":
         [
           {"displayName":"Eye tracking",
            "deviceName":"Eye tracking",
            "sampleRate":51.0,
            "quality":100.0,
            "iconType":"ET"}
         ]
      },
      {
          "name":"",
          "deviceSummaries":
          [
            {"displayName":"Eye tracking",
             "deviceName":"Eye tracking",
             "sampleRate":51.0,
             "quality":97.0,
             "iconType":"ET"}
          ]
      }
]
```


For each stimuli a `deviceSummaries` table is included with an entry for each device that was active during the slide-show. In the example above, only eyetracking was enabled, so there is only a single item in the `deviceSummaries` table.

The last entry in the list, with an empty string for a stimuli name, represents a summary for the whole slideshow.

The data in the `ExposureStatistics` event corresponds to the metrics that are available in the iMotions UI from the respondent Exposure Statistics tab.

Event Sequence:

Normally slide events will follow the pattern:

```
GazeCalibrationStart
GazeCalibrationEnd
SlideshowStart
SlideStart
SlideEnd
SlideStart
SlideEnd
....
SlideStart
SlideEnd
SlideshowEnd
ExposureStatistcs
```

NOTE: Gaze calibration events will only be included for eye-tracking studies.

If the test is not completed, but aborted by the operator, then no SlideEnd event will be seen and only the SlideshowEnd will be sent e.g.

```
SlideshowStart
SlideStart
SlideEnd
SlideStart
SlideshowEnd
```

Input events

Input events normally indicate user interaction using the keyboard and/or mouse. However they also include browser navigation events for web tests.

Key press

Event Source	`AttentionTool`	
Sample Name	`Keyboard`	
Additional Fields	**Description**	**Example**
Keyname	The name of the key that was pressed. If a modifier key was being held down at the same time, then it is appended to the key name.	`P` (P pressed on it's own) `P, Control` (P pressed whilst control is held down)

Mouse event

Event Source	`AttentionTool`	
Sample Name	`Mouse`	
Additional Fields	**Description**	**Example**
MouseEvent	Type of mouse event that was registered. e.g. `WM_MOVE` move event. Only forwarded if specifically enabled. `WM_MOUSEWHEEL` scrolling event with the mouse wheel `WM_RBUTTONDOWN/UP` `WM_LBUTTONDOWN/UP` button events See Windows API documentation for complete list.	`WM_LBUTTONUP`
Mouse X	X position of the mouse relative to the test display	`400`
Mouse Y	Y position of the mouse relative to the test display	`200`
ScrollDelta	Scroll delta for mouse wheel events	`-120`
XButton	Id of the button that triggered the event.For non-standard mice with more than 3 buttons.	

Browser navigation event

Event Source	`AttentionTool`	
Sample Name	`BrowserNav`	
Additional Fields	**Description**	**Example**

Navigation event	The browser event that triggered this
	
	`BeforeNavigate`
	`NavigateComplete`
	`DocumentComplete`
		`DocumentComplete` fired when the document is finished loading in the browser window.

URL	URL that we are navigating to	http://www.bbc.co.uk/news/
Name	Page name for DocumentComplete events only.	`BBC News – Home`

External Sensor Events

The following events originate from external sensor devices. The software receives data from them using device specific interfaces. The events are only available for forwarding if the device is hooked up to iMotions and iMotions has been configured to collect from the device.

Eyetracker event

Event Source	EyeTracker	
Sample Name	EyeData	
Additional Fields	**Description**	**Example**
Eyetracker timestamp	Timestamp assigned to the sample by the eyetracker. The value is adjusted by Attention Tool so that it is relative to the start of the slide show i.e. the timestamp of the first sample received after the start of the slide show is used as an offset for all subsequent samples.	1234
Gaze left X	Left eye gaze X coordinate relative to the display monitor in pixels. -1 indicates invalid sample.	145
Gaze left Y	Left eye gaze Y coordinate relative to the display monitor in pixels. -1 indicates invalid sample.	200
Gaze right X	Right eye gaze X coordinate relative to the display monitor in pixels. -1 indicates invalid sample.	165
Gaze right Y	Right eye gaze Y coordinate relative to the display monitor in pixels. -1 indicates invalid sample.	200
Left pupil diameter	Pupil diameter for left eye in mm	8
Right pupil diameter	Pupil diameter for right eye in mm	8
Left eye distance	Left eye distance from the tracker in mm.	700
Right eye distance	Right eye distance from the tracker in mm.	700
Left eye position X	X position of the left eye in the eyetracker camera as a ratio (0-1).	0.5
Left eye position Y	Y position of the left eye in the eyetracker camera as a ratio (0-1).	0.5
Right eye position X	X position of the right eye in the eyetracker camera as a ratio (0-1).	0.5
Right eye position Y	Y position of the right eye in the eyetracker camera as a ratio (0-1).	0.5

QSensor event

Event Source	QSensor	
Sample Name	AffectivaQSensor	
Additional Fields	**Description**	**Example**
Seq. Number	1 character sequence field 0-10 where 10 is represented by a letter e.g. 0-9, A	1
AccelZ	Accelerometer reading for the Z axis	-0.15
AccelY	Accelerometer reading for the Y axis	0.66
AccelX	Accelerometer reading for the X axis	-0.5
Battery	QSensor battery level in volts	4.1
Temperature	Skin temperature in celsius	35.2
EDA	Electro Dermal Activity	6.43

Emotiv EEG raw data

Event Source	Emotiv EEG	
Sample Name	EmotivEEG	
Additional Fields	**Description**	**Example**
AF3 contact value	Comma separated pair of values consisting of - Measurement for this electrode - Estimate of the quality of the electrode contact with the respondent's head. 0-4 where 0 is no contact and 4 is good contact.	
Contact values for F7, F3, FC5, T7, P7, O1, O2,P8, T8, FC6, F4, F8, AF4	See above	
Gyro X	Gyroscope measure in the X axis	
Gyro Y	Gyroscope measure in the Y axis	
Sequence No.	Sequence number generated by Emotive headset	

Emotiv EEG Affectiv Metrics

Event Source	Emotiv EEG	

Sample Name	`EmotivAffectiv`	
Additional Fields	**Description**	**Example**
Engagement	Emotiv SDK generated emotional response value in the range 0-1	`0.3`
Long term excitement	See above	
Short term excitement	See above	
Frustration	See above	
Meditation	See above	

ABM EEG raw data

Event Source	`ABM EEG`	
Sample Name	`ABMRawEEG`	
Additional Fields	**Description**	**Example**
Epoc	Epoch time (seconds)	`341`
Offset	Offset within epoch (0-255)	225
SDKTimeStamp	ABM time stamp in `hhmmsstt` (hoursminutessecmiliseconds)	000541879
X10: EKG, Poz, Fz, Cz, C3, C4, F3, F4, P3, P4 **X24:** F3, F1, Fz, F2, F4, C3, C1, Cz, C2, C4, CPz, P3, P1, Pz, P2, P4, Poz, O1, Oz, O2, EKG, AUX1, AUX2, AUX3	The channel data collected from the device. Varies by device. For X10, 10 floating values are returned.	

ABM EEG decon data

Event Source	`ABM EEG`	

Sample Name	ABMDeconEEG	
Additional Fields	**Description**	**Example**
Epoc	The fields layout is the same as above for raw data. Unlike the raw data, the decontaminated data is received in batches with some delay. However both the Attention Tool timestamp and the ABM time fields reflect the timestamp of the raw data on which the decontaminated sample is derived from.	

Emotient FACET

Event Source	FACET	
Sample Name	EmotientFACET	
Additional Fields	**Description**	**Example**
Frame Index	The frame number in the video recording of the face for this slide.	1
Frame Time	The frame timing in 10,000 Milliseconds	3333333
Faces Count	Number of faces detected. -1 indicates the frame was not processed, 0 indicates no face was detected. In either case the remaining fields will be empty.	1
Face X	Location X coordinate of the face detection rectangle.	212
Face Y	Location Y coordinate of the face detection rectangle.	120
Face Width	Width of the face detection rectangle.	180
Face Height	Height of the face detection rectangle.	185
Joy, Anger, Surprise, Fear, Neutral, Contempt, Disgust, Sadness, Positive, Negative	Each emotion has 2 fields: -Evidence for the presence of the emotion. Logarithmic scale, typically between -5 and +5. -Intensity of the emotion, value between 0-1	

Example Listener

The following Powershell example listens for incoming events on the UDP socket and simply prints anything it receives to the standard output.

Contents of DumpUDP.ps1

```
function UDPListen( $port )
{
	$endpoint = new-object System.Net.IPEndPoint ([IPAddress]::Any,$port)
	$udpclient=new-Object System.Net.Sockets.UdpClient $port
	while ($true ) {
		$content=$udpclient.Receive([ref]$endpoint)
		[Text.Encoding]::ASCII.GetString($content)
	}
}
```

Open a powershell prompt (Start -> Accessories -> Windows PowerShell -> Windows Powershell)

Enable script loading and load up the script – in the example below, the script file was saved to the desktop. Then execute UDPListen(8088). We will start listening for incoming messages from AttentionTool, and anything received will be printed to the console.

```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\Users\myuser> cd Desktop
PS C:\Users\myuser\Desktop> Set-ExecutionPolicy -scope CurrentUser -force unrestricted
PS C:\Users\myuser\Desktop> . .\DumpUDP
PS C:\Users\myuser\Desktop> UDPListen(8088)
00000001;AttentionTool;SlideshowStart;0;-1;20130321130651148;Anonymous 20-03-13
10h55m;MALE;16
00000002;AttentionTool;SlideStart;0;-1;20130321130651148;BBC;BlackInterslide
00000003;AttentionTool;SlideEnd;1503;-1;20130321130652651
00000004;AttentionTool;SlideStart;1506;-1;20130321130652654;BBC;TestImage
00000005;AttentionTool;BrowserNav;2119;-1;NavigateComplete;http://www.bbc.co.uk/news/;
```

```
00000006;AttentionTool;BrowserNav;4738;2305;DocumentComplete;http://www.bbc.co.uk/news
/;BBC News - Home
00000007;AttentionTool;Mouse;10321;7888;WM_LBUTTONDOWN;681;275;0;0
00000008;AttentionTool;Mouse;10473;8040;WM_LBUTTONUP;681;275;0;0
00000009;AttentionTool;BrowserNav;10538;8105;BeforeNavigate;http://www.bbc.co.uk/news/
world-europe-21874427;
00000010;AttentionTool;BrowserNav;10717;8284;NavigateComplete;http://www.bbc.co.uk/new
s/world-europe-21874427;
00000011;AttentionTool;BrowserNav;12965;10533;DocumentComplete;http://www.bbc.co.uk/ne
ws/world-europe-21874427;BBC News
 - Turkey Kurds: PKK chief Ocalan calls for ceasefire
00000012;AttentionTool;Mouse;16384;13952;WM_LBUTTONDOWN;1580;1;0;0
00000013;AttentionTool;Mouse;16616;14184;WM_LBUTTONUP;1580;1;0;0
00000014;AttentionTool;SlideEnd;16678;-1;20130321130707826
00000015;AttentionTool;SlideshowEnd;16681;-1;20130321130707850
```

Stop the script by closing the powershell window.
Examples of UDP listener code in other languages are readily available on the internet.

Event Receiving Interface

Overview

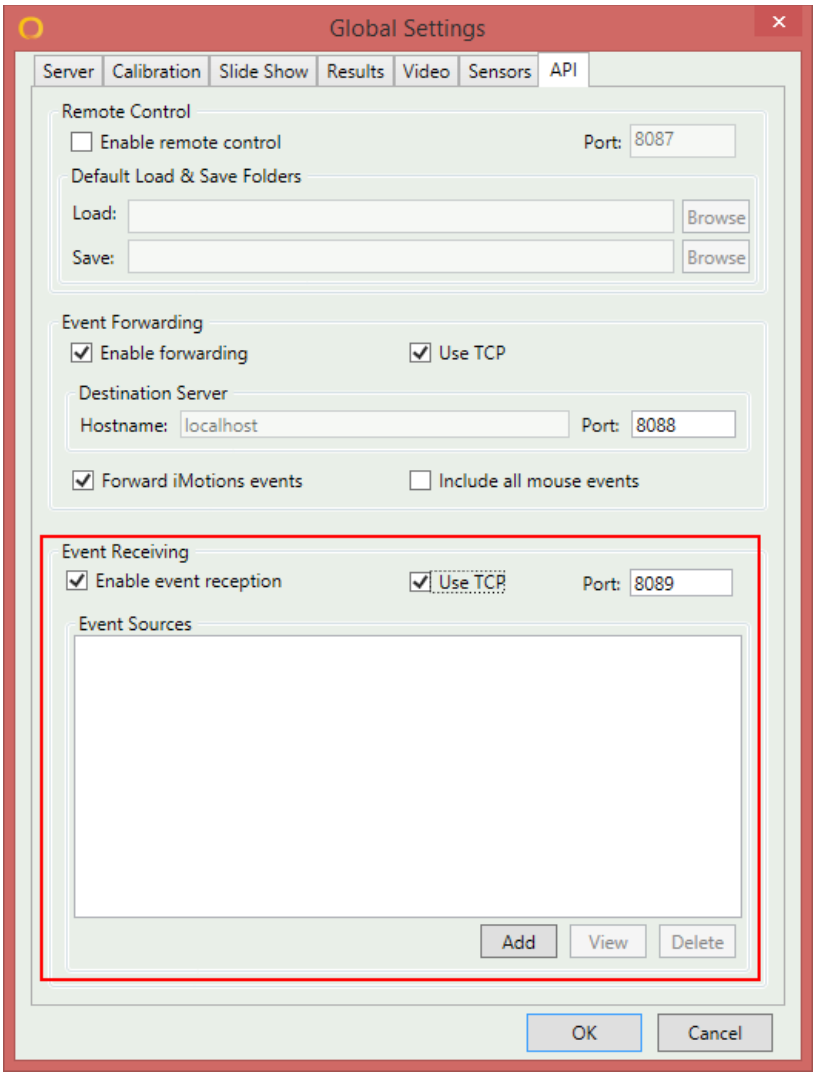
iMotions currently supports a number of sensors for measuring a respondent's state during a test e.g. Emotiv EEG, Affectiva QSensor. However there are many more sensor devices that a customer could use, and it is impractical for iMotions to support every such device. Instead it is envisaged that a 3rd party application would be created that interfaces with their desired device set, and then passes the data thus collected to iMotions software. This data will be treated in the same way as data collected from the sensors with built in support e.g. it will be synced with all other collected data, it can be visualised on a graph, will be stored in the database and saved with the study, will be available in the text data export for further analysis etc.

This document describes the method whereby a 3rd party application can send captured sensor readings to iMotions software.

NOTE:
Whilst we tend to focus on the "unsupported external sensor" scenario, it should be noted that there is no limitation to where the data arriving from the 3rd party application was sourced. It will be stored in the software as a stream of external events. Whether they originated from external sensor devices, or the values were generated internally by the 3rd party application is not important to iMotions.

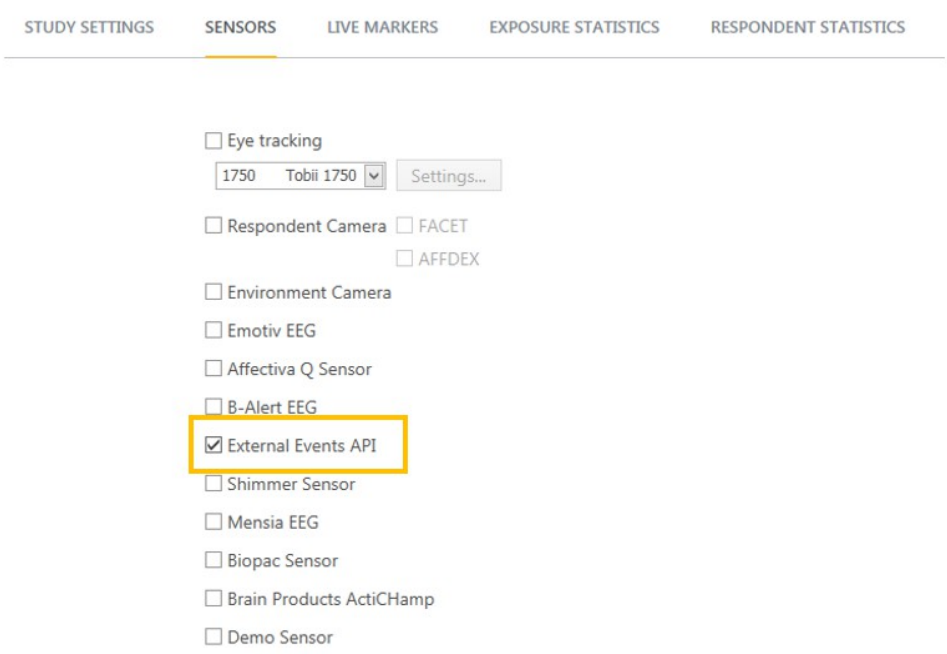
iMotions Setup

Event reception is enabled using the Global Settings dialog, within the API tab.



As with event forwarding, the operator can choose to connect to iMotions using TCP or UDP. In the screenshot above, the TCP option has been enabled.

When adding a new study, also make sure that “External Events API” is enabled.



Event Interface

The software utilises a UDP or TCP network connection to implement the event receiving interface.

- UDP – iMotions listens on a specified port for incoming packets
- TCP - iMotions will listen for connections on a specified port. Once a connection is accepted, the software will read incoming events from the resulting TCP stream.

The incoming data packet must conform to the specification detailed later in this document. When a packet is received it is processed and checked against the registry of configured event sources. Assuming it is recognised, it will be parsed and the resulting event object will be handed over to the iMotions data collection pipeline for further processing e.g. live graph update, storage etc.

Event Sources

iMotions receives events from event sources. An event source is a logical grouping of different sample types that can be received from a particular source. It would typically correspond to an external sensor.

iMotions can receive data from many event sources, and each event source can support many different sample types. In order for the software to accept a sample, it must first be configured using an event source definition file. This is simply an XML text file describing the sorts of samples that can be received from this source, and the structure of the fields in each sample.

Versioning

In order to enable 3rd party applications to evolve the Source definition supports a version attribute. This will allow the definitions to expand over time in a backwards compatible fashion. So if a 3rd party enhances their support for an external sensor and needs to add some extra fields to the sample definition, the definition file would be updated with the latest sample descriptions, and the version number would be incremented. This new definition would then be loaded up into iMotions.
The older defintions are retained in the software so that existing data can still be decoded correctly, and indeed older versions of the 3rd party app could still run happily with iMotions.

Example Source Definitions

The following example shows a definition file for an event source of the Affectiva QSensor.

```
<EventSource Version="1" Id="QSensor" Name="Affectiva Q Sensor">
    <Sample Id="AffectivaQSensor" Name="QSensor">
        <Field Id="SeqNo" />
        <Field Id="AccelZ" />
        <Field Id="AccelY" />
        <Field Id="AccelX" />
        <Field Id="Battery" />
        <Field Id="Temperature" Range="Fixed" Min="30" Max="40" />
        <Field Id="EDA" Range="Variable" Min="0" Max="0.2" />
    </Sample>
</EventSource>
```

A definition file for the Emotiv EEG headset could look something like this.

```
<EventSource Version="1" Id="EmotivEEG" Name="Emotiv EEG">
    <Sample Id="EmotivAffectiv" Name="Emotiv Affectivity" >
        <Field Id="Engagement" Range="Fixed" Min="0" Max="1" />
        <Field Id="ExcitementLongTerm" Range="Fixed" Min="0" Max="1" />
        <Field Id="ExcitementShortTerm" Range="Fixed" Min="0" Max="1" />
        <Field Id="Frustration" Range="Fixed" Min="0" Max="1" />
        <Field Id="Meditation" Range="Fixed" Min="0" Max="1" />
    </Sample>

    <Sample Id="EmotivRawData" Name="Emotiv Raw Data">
        <Field Id="EEG Timestamp" />
        <Field Id="AF3_Quality" />
        <Field Id="AF3_Value" />
        <!— other EEG channel fields -->
        <Field Id="GyroX" />
        <Field Id="GyroY" />
        <Field Id="SeqNo" />
    </Sample>

</EventSource>
```

Only fields with the Range attribute will be shown on a line graph. Therefore if a sample does not contain any such fields, then no graph for that sample type will be shown. Fields marked for graph display must be numeric, all other fields can contain any arbitrary data, the values will not be checked/validated.

Multiple Instances

It is possible that there are multiple instances of a given event source generating data e.g. multiple EMG sensors monitoring different muscles. The API requires only one source definition to be loaded into iMotions, the different instances should be identified by a field in the event message.

Incoming Messages

External applications communicate with iMotions by sending messages over a network 'connection'. This allows the 3rd party application to run on the iMotions system, but also on any other LAN connected machine.

Each packet will consist of a UTF-8 text string, with fields separated with a semi-colon character. All packets must start with a fixed header that will allow iMotions to identify the type of the message. Since semi-colon is used as a field separator, it is important that any 3rd party application ensures that this character does not appear in any of the event fields e.g. check any text field for semi-colon and replace it with a comma if found etc.

A packet MUST be terminated by a carriage-return, line-feed combination, similar to HTTP headers, denoted as "\r\n" in many programming languages. Similarly with the semi-colon character, the sending application must ensure that the \r\n combination does not appear anywhere in the message body.

Common Header

Field No.	Name	Description	Example
1	Type	Single character that identifies the type of message. Two message types are supported. E – Sensor Event M – Discrete Marker	`E`
2	Version	Message version. Digits identifying the version of the message , in cases where a message has evolved over time. Currently only the Marker message has more than one version.	`1`

Sensor Event Fields

Field No.	Name	Description	Example
3	Source	Event Source ID. Must match an event source definition	`MyEMG`
4	Source Definition Version	Version of the Source definition the event data conforms to. The version in the definition file must match this value. If blank, then it is assumed that the event is for the latest definition.	
5	Instance	Optional instance ID. Typically left blank, but can be included if there are multiple instances of this event source sending data to AT. For example, if the respondent has an EMG sensor on each arm, the instance field would indicate	`LeftBicep`

		which of the sensors this event relates to. Contains arbitrary text with a maximum length of 15 characters.	
6	Elapsed Time	Optional timestamp indicating the time in ms since the start of the test. Typically left empty unless the sample is generated from data already received from AT.	`1000` or `20151225141059555`
7	Media Time	Optional media-time indicating the video file position that this sample should be synced with. Typically left empty unless the sample is generated from data received from AT.	
8	Sample Type	ID indicating the sample type. This must match a sample description from the event source definition file.	`EMGData`
9….		Data fields for the sample type. The number of fields must match the sample definition.	

Elapsed and Media Time

Typically the external application will leave these timestamp fields empty. The iMotions software will timestamp the event when it is received by the system. The latency between the sending and receiving is typically much smaller than the 1ms precision that iMotions uses for time-stamping events. Therefore it is envisaged that the timestamp fields would only be filled in by the external application in the following cases.

1. The external events are generated from data received from the iMotions software e.g. some synthesized metric. In this case the external application would use the timestamp(s) on the source samples to fill in the timestamp fields in the generated sample.
2. There is a significant delay between the external application acquiring the data, and the data being forwarded to iMotions. In this case the ElapsedTime should be included to maintain the synchronization between the external signal and the running slideshow. Imotions will use the ElapsedTime field to calculate a corresponding media-time if appropriate, so the media-time can usually be left blank.
   In these cases it may be more convenient for the external application to send a clock-time rather than a millisecond offset from the start of the slide-show. Therefore if the ElapsedTime field matches the following pattern "`yyyyMMddHHmmssfff`", it will interpreted as a full data-time string e.g. 20151225141059234
   If the slide-show began at 14:00 on the 25th December 2015, then this time-stamp would be converted to an ellapsed time of 65234 milliseconds.

Example

The following shows what a message representing a QSensor sample would look like (see the example QSensor Source definition earlier).

```
E;1;QSensor;1;;;;AffectivaQSensor;002;0.1232;-0.123;0.321;4;37;0.223\r\n
```

Note `\r\n` denote the end of record terminator sequence i.e. 2 bytes 0x13 0x10.

Marker Event Fields

Markers are used to annotate a recording, allowing significant events to be stored with the collected data. Subsequently the markers will be available in the time-line for navigation purposes. If a segment of the time-line is marked, then the segment can be used to generate a scene that allows for aggregation of data across respondents.

Message Version 1

Field No.	Name	Description	Example
3	Elapsed Time	Optional timestamp indicating the time in ms since the start of the test. Typically left empty unless the sample is generated from data already received from AT.	
4	Media Time	Optional media-time indicating the video file position that this sample should be synced with. Typically left empty unless the sample is generated from data received from AT	
5	Short Text	Short name indicating the markers meaning	`CheckOut`
6	Description	Optional longer descriptive text describing the purpose of the marker. Special care should be taken to ensure that this field does not contain a semi-colon character.	`Respondent completed the check out task.`

Message Version 2

The following additional fields were introduced in AttentionTool 5.4 and must be included if the version in the message header is set to 2, or is omitted.

Field No.	Name	Description	Example
7	Marker Type	One of the following:- D – Discrete marker. Marks a single standalone event. S – Marks the start of a recording segment. E – Marks the end of a recording segment. If no matching start segment message has been received, the message will be discarded. N – Marks the start of the next segment, automatically closing any currently open segment.	S
8	SceneType	If the event represents the start of a segment range (S or N marker type), then the scene type can be included to indicate what sort of scene should be created if the marker range is to be used to auto-generate scenes. Possible values are:- V – Video segment. The marked region of the recording represents a section of video that is common across respondents. I – Image segment. The marked region of the recording represents a static image.	I

The purpose of the new fields is to facilitate the following :-

- Mark a segment of interest in a screen recording. This is achieved by sending matched pairs of start/end markers. Matching markers are paired based on the ShortText field.
- Use a marked segment to automatically create a scene. By including the Scene Type field with the start marker, the 3rd party application indicates to iMotions that the segment should be used to create a scene. The value of the scene type field will determine if a video or image scene is generated. The scene type will be set based on the content that is displayed on the screen during the segment :- V for dynamic content (video clip), and I for static content (poster ads).

Game Test Scenario

Consider using iMotions to test a game. The game could be updated to send marker events to the software when interesting events occur during the game play e.g. when the player dies. In addition, if the user is exposed to video sequences (cut scenes etc), then a start/end pair of markers could be sent at the start and end of the video clip. iMotions will then automatically generate a scene fragment for the marked region. The data for the marked region can then be aggregated across all respondents who were exposed to the cut scene.

Discrete Examples

The following shows an example discrete marker message.

```
M;1;;;CheckOut;Respondent completed the check out task\r\n
```

The following shows the same discrete marker message using the new v2 interface.

```
M;2;;;CheckOut;Respondent completed the check out task;D;\r\n
```

Segment Examples

The following messages show a start/end sequence that would mark out a segment of a recording.

```
M;2;;;HotClip1;Show trailer;S;V\r\n

.....30 seconds later

M;2;;;HotClip1;;E;\r\n
```

The following messages illustrate how a sequence of questions could be marked out with the API. Since all segments need to run sequentially, we can simplify the interaction a little by just sending a sequence of Next segment markers.

```
M;2;;;Question1;Shown question1;N;I\r\n

.....30 seconds later

M;2;;;Video1;Shown clip1;N;V\r\n
```

```
.....2 minutes later

M;2;;;Question2;Shown question2;N;I\r\n
```

iMotions Remote Control Interface

Overview

The remote control API allows external programs to control the iMotions software i.e. perform certain tasks without interacting with the user interface. The following commands are available.

Command	Action
Minimize	If idle, causes iMotions to minimize to the task bar. The software will not minimize if it is currently running a test.
Maximize	iMotions window will be restored.
Shutdown	iMotions will close. If running a test, then the command is rejected.
Run <Study> <Respondent>	iMotions will use the command parameters to select the study and respondent to test. Then a test will be run. If a test is already in progress, the command will be rejected.
Next Slide	Requests the slide-show to proceed to the next slide. Equivalent to the Shift-Space keyboard combination
Cancel Slide-show	Requests the slide-show to abort. Equivalent to Shift-F1.
Status	The Status command can be sent to iMotions at any time. It will send a response which will indicate if the software is idle or currently running a test. If a test is ongoing, then details of the current test will be included in the response.
Save <DateToSave> <Study> <Respondent>	If idle, iMotions will save the selected Study/Respondent to file. The resulting zip file will be saved in a configurable folder in the file system. The location of the zip file will be included in the command response.
Load <ZipFile> <TargetStudy> <AddMerge>	Loads the contents of a study export into iMotions. This will either create a new study or merge the data for the tested respondents in the zip file into an existing study.
Export Sensor Data	Exports the study data to a raw text file that can be loaded into Excel, Matlab etc.
Export Video Recordings	Exports the screen or face-camera recordings to a folder.

This limited set of commands is targeted at allowing a customer, with a large data collection network, to integrate iMotions software into their existing study distribution infrastructure.

Distributing Studies In iMotions

The software allows the user to export the data for a study to a zip file. This file can then be loaded into a different system for analysis. If the study already exists on the analysis system, then the data for the

respondents in the zip file are added to the existing study. The Save and Load commands allow these operations to be executed by a third-party software rather than from the UI.

Example Workflow

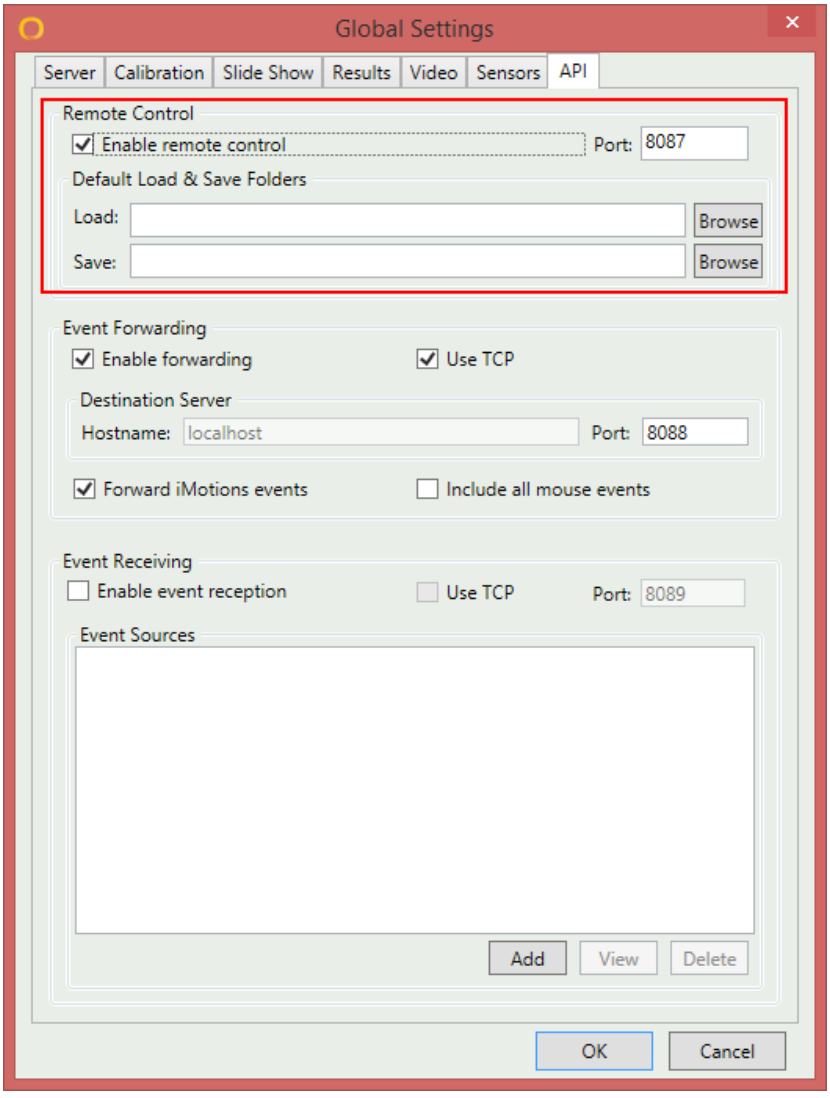
The following example, shows how the Save and Load commands could be used in a distributed environment, where there is a "master" analysis instance of iMotions, and a number of "slave" data collection instances.

1. Master study is set up on the analysis iMotions system. Typically this would be done using a test plan.
2. Operator uses some custom software on the master system to distribute the study.
    - The software requests iMotions to execute a Save command.
    - iMotions creates a zip file containing the study definition and includes the full path of the exported study in the response to the Save command.
    - The software copies the zip file to numerous slave systems e.g. using windows network shares.
3. Custom software on the slave system detects the zip file, e.g. by searching in a specific "incoming" folder every few minutes. Once a new study is found, the software tries to load it into iMotions
    - Check if iMotions is busy using the Status command.
    - If the software is not currently running a study, then execute the Load command.
    - iMotions uses the filename passed in the Load command, to import the study definition contained in the zip file.
4. Custom software on the slave system is used to run the study for individual respondents, and subsequently save the collected data to zip files.
    - The software requests iMotions to run a study for a particular respondent.
    - The software listens on the AT Event Forwarding API for slideshow events.
    - When the "End Slideshow" event is received the software knows that the test has been completed.
    - Software sends a Save request to iMotions with the name of the last tested respondent.
    - iMotions creates a data export for the named respondent.
    - The software takes the exported zip file and copies it back to some "Tested" folder on the master system.
5. Managing software on the master system detects the incoming data in the "Tested" folder and uses the Load command to merge the respondent test data into the master iMotions study.

iMotions Setup

The remote control interface can be activated in 2 ways.

1. iMotions can be started with a /REMOTECONTROL switch. When started like this, the software will not display any splash screen, and will initially be minimized in the Windows task bar. The remote control functionality will be activated, and the remote control properties in global settings will be disabled.
2. If started in the normal manner, the remote control interface is by default deactivated. It can be enabled using the API tab in global settings.



The Event Forwarding and Receiving interfaces, only work in one direction i.e. events are written from iMotions or read by iMotions. By contrast the remote control interface is based around a request/response protocol - a request message is sent to iMotions, and the software answers with a response message indicating the results of the command. Therefore the remote control has been implemented to work over TCP connections only. The settings dialog allows the port that the software listens on to be configured, and also the default folders iMotions will use when executing Load or Save commands.

iMotions Operation

When the remote control interface is activated, iMotions will listen on the specified TCP port and wait for a client to connect. Once a client connection is accepted, the resulting TCP stream is used to exchange messages. The client is expected to send command messages, in response to which iMotions will execute the requested command and reply, possibly some time later, with a response message indicating if the command was successfully executed.

If the client is also responsible for starting iMotions, it would typically run the software with the /REMOTECONTROL option, and then wait for iMotions to become available. The easiest way to do this would be to periodically attempt to connect to the remote control TCP port until the connection request is successful.

Command Reference

This section describes in detail the format of the remote-control messages, and their corresponding response messages.

Common Message Formats

Incoming command messages will use a common header, which will identify the sort of message and the command that is being sent. Following this will be zero or more command specific parameters. The command messages are similar to the event receiving message format.

The response messages will follow a similar pattern, with a common response header and additional command specific response parameters.

Common Command Header

Field No.	Name	Description	Example
1	Type	Single character that identifies the type of message. R – Remote control command	R
2	Version	Message version. Digits identifying the version of the API comamnd that the message conforms to. The initial version is 1.	1
3	Id	Arbitrary command identifier that will be included in any response.	0000123
4	Command	Command Id. One of the following • MIN • MAX • RUN • SLIDESHOWNEXT • SLIDESHOWCANCEL • STATUS • SAVE • LOAD • EXPORTSENSORDATA • EXPORTRECORDEDVIDEOS • FACEVIDEOPROCESSING • SHUTDOWN	STATUS

Common Response Header

Field No.	Name	Description	Example
1	Seq. No	This is a simple counter incrememented every time a message is sent over this interface. For the UDP interface, then the count is updated for every outgoing UDP event. For the TCP interface, the count will be updated for every response on the current TCP remote-control connection.	R
2	Event Source	Identifies the message originating from the imotions remote control system: RemoteControl	RemoteControl
3	Sample Name	Will contain the name of the command that this response refers to	MAX
4	Timestamp	If a test is in progress, this field will contain the time in ms since the test was started.	7788
5	Media	Position in the current video based stimuli in ms. This	1234

	Time	value is only applicable for videos and screen/web recordings. For other stimuli types, or when a test is not executing, -1 will be returned.	
6	Id	Arbitrary command identifier that was included with the command message.	0000123
7	Status	1 = Success, 0 = Failed to execute command	0
8	Fail Reason	String describing the fail. Empty if successful command	Eyetracker not connected

Commands Requiring No Parameters

The following commands do not support any additional parameters.

MIN Command

On receipt of this command the system will minimize the UI.
The command fails if a test is in progress. No additional data is returned in the response message.

MAX Command

On receipt of this command the system will maximize the UI and bring it to the front making it the active application.
The command fails if a test is in progress. No additional data is returned in the response message.

SHUTDOWN Command

On receipt of this command the system will initiate a shutdown of the software.
The command fails if a test is in progress. An OK response will be sent before the program exits, so a client should always see a response before the connection closes. No additional data is returned in the response message.

The following 3 commands are the only ones that are supported when a slide-show is in progress. The first two change the behavior of a slide-show, and as such will fail if a study is not currently being executed.

SLIDESHOWNEXT Command

Equivalent to the respondent/operator pressing the next-slide hot-key combination (Shift-space by default) during a slide-show. An error will be returned if the current slide has not been configured to support manual slide change.

SLIDESHOWCANCEL Command

Equivalent to the respondent/operator pressing the cancel slide-show hot-key combination (Shift-F1 by default) during a slide-show. The slide-show is aborted, and no data is saved for the respondent.

STATUS Command

The status command acts as a kind of Ping message. It can be sent in or outside of a test.

The Version 2 message will also detect if the software is busy performing some form of processing and will report the status messages that are shown by the "Busy" screen.

Additional response message fields

Field No.	Name	Description	Example
9	Study	Currently executing study. Empty if a test is not in progress	Cola Test
10	Respondent	Current respondent. Empty if a test is not in progress	Bill Gaze
11	Stimuli	Current stimuli being displayed to the respondent. Empty if a test is not in progress	Santa Cola

Additional Version 2 response message fields

Field No.	Name	Description	Example
12	IsBusy	0 = system is not showing the Busy UI. 1 = system is busy performing some long running task and is blocked from executing any new command.	1
13	Status Text	When busy – the text displayed in the status bar of the software, typically a short description of the current task. Empty if the system is not performing processing.	Exporting Study
14	Progress Text	When busy – the text displayed by the Busy UI, otherwise empty.	Fetching raw data from the database

Other Commands

The following commands require the client to supply additional command parameters in the command request message.

RUN Command

On receipt of this command the system will start a test for the named study and respondent included in the command. The command will fail if a test is already in progress, or if the named study is not set up in iMotions.

If version 1 of this message is received, then the named respondent must also exist in iMotions, otherwise the operation fails.

If version 2 of the message is received then the sender has the option to include respondent properties in an additional field, in which case the respondent will be created with these properties.

If version 3 of the message is received then the sender has the option to request the software to disable certain dialogues.

No additional data is returned in the response message.

Additional command message parameters – Message Version 1

Field No.	Name	Description	Example
5	Study	Name of the study	Cola Test
6	Respondent	Name of the respondent to be tested	Bill Gaze

Additional command message parameters – Message Version 2

Field No.	Name	Description	Example
7	Respondent Properties	Space separated list of name/value pairs, specifying the age and gender of the respondent to be added. If this field is blank, then a respondent will not be added, and the named respondent must already exist. Properties that must be specified:- Gender=Male|Female Age=<int>	Gender=Male Age=26

Additional command message parameters – Message Version 3

Field	Name	Description	Example

No.			
8	UI Prompt Options	Specify how the software should react if an exceptional condition is detected as part of initiating the slide show. `NormalPrompt:` Default behavior where the operator is prompted to confirm continuing when certain conditions are detected e.g. an expected sensor is not active. `NoPromptIgnoreWarnings:` The operator is not prompted on warnings, it is assumed that the continue option is desired. `NoPrompt:` The operator is not prompted on warnings, they are treated as errors, and the study will not be run.	NormalPrompt

NOTE:
Typically it is assumed that there is an operator monitoring the execution of the software. If iMotions is requested to run a study and it detects some exceptional condition, then it will use on screen dialogues to query the operator e.g. if the study has been configured to use GSR, but the software is not currently connected to a Shimmer device, then the operator will be prompted if they wish to go ahead and run the study anyway.

The new version 3 option, allows this behavior to be changed – typically for use where the software is being used as a data collection middleware, and the caller does not want it to be creating UI dialogs.

SAVE Command

On receipt of this command the system will save the data for the named study / respondent combination into a zip file. The data in the file is equivalent to using the "Save Study To File" option from the iMotions UI i.e. it is intended for use by iMotions's "Load Study From File" feature, not by third party applications. Unlike the save option available from the menus, the remote command will allow the client to specify a value indicating what data to include in the export. This last option can be used if you want to minimize the size of the exported data e.g. by excluding any stimuli images or videos that are already available on the master analysis iMotions system.
The command will fail if a test is in progress, or if the named study/respondent does not exist or has not been tested.

Additional command message parameters

Field No.	Name	Description	Example
5	Data to include	This option allows you to limit the data that will be exported. If left blank, the default will be 1. 0 - Data only, i.e. meta-data about the study and respondent, and any collected sensor and eyetracking data. 1 - Additionally include any media files generated for the respondents i.e. face recordings, screen recordings.	1

		2 - Include all data, including stimuli images and videos. 3 – Similar to (1) but exclude face recordings.	
6	Study	Name of the study	Cola Test
7	Respondent	Name of the respondent to be tested, leave blank for all tested respondents.	Bill Gaze
8	Folder	Optional name of a folder where the data should be saved. If not supplied, then AT chooses a default location.	N:\SharedFolder\AT

Additional response message fields

Field No.	Name	Description	Example
9	Path	Full path to the generated zip file	N:\Shared\Rsp01.zip

LOAD Command

On receipt of this command the system will load the data contained in the specified zip file. The data in the file will have been created using the iMotions "Save Study To File" option, or via the SAVE command. If performing a Merge operation, the system will check to ensure that the existing study, and the study in the zip file are compatible i.e. they must contain the same stimuli and have the same screen resolution setup.

Additional command message parameters

Field No.	Name	Description	Example
5	File	Path to the zip file containing the study export data.	N:\SharedFolder\AT\Study1.zip
6	Study	Optional name of the Study that the data is to be merged into, or the name to use when creating a study. The default is the name of the study saved in the zip file.	Cola Study
7	Mode	Optional flag indicating if this is a Merge or a Create study operation, default is Auto. Merge: Merge data only. If no existing study is found with a matching name and properties, then the Load fails. Create: Create a new study. If an existing study with a matching name is found, then the Load fails. Auto: If a matching study is found, then do a merge, otherwise create a new study. If a matching name is found but the study properties don't match, then the load will fail.	Create

Additional response message fields

Field No.	Name	Description	Example
9	Mode	Indicates whether a Merge or Create was executed	Merge

EXPORTSENSORDATA Command

On receipt of this command the system will save the data for the named study / respondent(s) combination into text file(s). The data in the file is equivalent to using the table formatted "Export Sensor Data" option from the study library export context menu. The remote command supports the selection of particular respondents. This can be done by specifying the name of a single respondent, or by specifying a date range. If a date range is supplied, then only tests that were executed within the date range will be exported.
The command will fail if a test is in progress, or if the named study/respondent does not exist or has not been tested.

Additional command message parameters

Field No.	Name	Description	Example
5	Data to include	This option allows you select what data you want exporting using a space seperated list of values. If not specified, then all sensor data is included. Available values are:- GazeCalibration – Gaze calibration details Eyetracker – Eyetracker data UserEvents – Mouse, Keyboard, browser events CameraDevice – FACET analysis EmotivEEGDevice – Emotiv EEG raw data and metrics ShimmerDevice – Shimmer GSR etc QsensorDevice – Q Sensor GSR BalertDevice – ABM EEG raw data & metrics UTC – Additional UTC timestamp column	Eyetracker QSensorDevice
6	Study	Name of the study	Cola Test
7	Respondent	Name of the respondent to be tested, leave blank for all tested respondents.	
8	Folder	Optional name of a folder where the data should be saved. If not supplied, then AT chooses a default location.	N:\SharedFolder\AT
9	From	From date time yyyymmddHHMMss	20141025193000
10	To	To date time yyyymmddHHMMss	20141025203000

Additional response message fields

Field No.	Name	Description	Example
9	Path	Full path to the folder where the text files are generated	N:\Shared\AT

EXPORTRECORDEDVIDEOS Command

On receipt of this command, the system will copy recorded respondent videos into a named folder. The videos are named based on the respondent name. An individual respondent can be specified, or a tested date range can be used.
The command will fail if a test is in progress, or if the named study/respondent does not exist or has not been tested.

Additional command message parameters

Field No.	Name	Description	Example
5	Data to include	Sort of video to export. Choose one of the following :- Screen FaceCam If left blank, then Screen is assumed.	FaceCam
6	Study	Name of the study	Cola Test
7	Respondent	Name of the respondent to be tested, leave blank for all tested respondents.	Bill Gaze
8	Folder	Optional name of a folder where the data should be saved. If not supplied, then AT chooses a default location.	N:\SharedFolder\AT
9	From	From date time using yyyymmddHHMMss format. Leave blank for all respondents.	
10	To	To date time, using yyyymmddHHMMss format. Leave blank for all respondents.	

Additional response message fields

Field No.	Name	Description	Example
9	Path	Path to the folder where the video files were copied to.	N:\Shared\AT

FACEVIDEOPROCESSING Command

On receipt of this command, the system will initiate the face video post processing.
The command will fail if a test is in progress, or if the named study/respondent does not exist.
NOTE: The processing of face videos is extremely resource intensive and can take a considerable amount of time, depending on how many respondents require processing and the length of the recordings.
The enhanced STATUS command can be used to periodically poll iMotions to check on the progress of the task.

Additional command message parameters

Field No.	Name	Description	Example
5	Study	Name of the study	Cola Test
6	Respondent	Name of the respondent to be tested, leave blank for all tested respondents. A list of respondent names can also be supplied, separating each name with a | character.	Bill Gaze|Bob Smith
7	FaceSize	Optional minimum face size as a percentage of the full video frame width. Default is 20%.	15
8	Skip	Optional frame skipping factor. The process can be sped up by not processing all frames. Default is to process all unprocessed frames.	2
9	Overwrite	Optional - Reprocess all frames (1), regardless if they were previously processed e.g. during the slideshow. Default is not to reprocess frames (0).	1
10	CPU Count	Optional engine count. Default is the number of CPUs detected on the system, up to a max of 4 or 8, depending on strategy.	2
11	Multi-process strategy	Optional strategy for using multi-core processing. respondent (default)– multiple respondents are processed in parallel. A maximum of 8 respondents can be processed at the same time. Use this strategy if many respondents are outstanding. file – multiple frames of a video are processed in parallel. A maximum of 4 concurrent engines can execute frame processing. Use this strategy to process a single respondent as quickly as possible.	respondent