



# Neurotechnology Products Activation 12.4

User's guide

**Version:** 12.4.0.0.

**Release date:** 5/17/2023.

# Table of Contents

<b>1 About</b>	<b>1</b>
<b>2 License types</b>	<b>3</b>
2.1 Serial numbers (Single computer licenses)	4
2.2 Dongle (Volume license manager)	5
2.3 Internet licenses	6
2.4 Trial	7
<b>3 Activation options</b>	<b>8</b>
3.1 Activation wizard	9
3.1.1 Trial activation	9
3.1.2 Single computer licenses activation	11
3.1.3 Dongle activation	13
3.2 Manual activation	16
3.2.1 Trial	16
3.2.2 Serial numbers (Single computer licenses)	16
3.2.3 Volume license manager (dongle)	19
3.2.3.1 Generating licenses for single computers activation	19
3.2.3.2 Managing licenses on network	20
3.3 Activation for Android	22
<b>4 Advanced licensing</b>	<b>25</b>
4.1 Licenses obtain in your application	26
4.2 Licenses deactivation	28
4.3 Changing from trial to non-trial	29
4.4 Virtual environments licensing	30
4.5 Disaster recovery licenses	31
4.6 Licenses generation from a dongle	32
4.7 Licensed API functionality	35
4.7.1 MegaMatcher Standard SDK	35
4.7.2 MegaMatcher Extended SDK	41
4.7.3 MegaMatcher On Card SDK	47
4.7.4 VeriFinger Standard SDK	47

4.7.5 VeriFinger Extended SDK	48
4.7.6 VeriLook Standard SDK	50
4.7.7 VeriLook Extended SDK	50
4.7.8 VeriEye Standard SDK	51
4.7.9 VeriEye Extended SDK	52
4.7.10 VeriSpeak Standard SDK	53
4.7.11 VeriSpeak Extended SDK	53
4.7.12 SentiVeillance SDK	54
4.7.13 SentiSight SDK	55
4.7.14 SentiSight Embedded SDK	55
4.7.15 SentiGaze SDK	55
4.7.16 SentiSculpt SDK	56

## **5 Troubleshooting** **57**

# 1 About

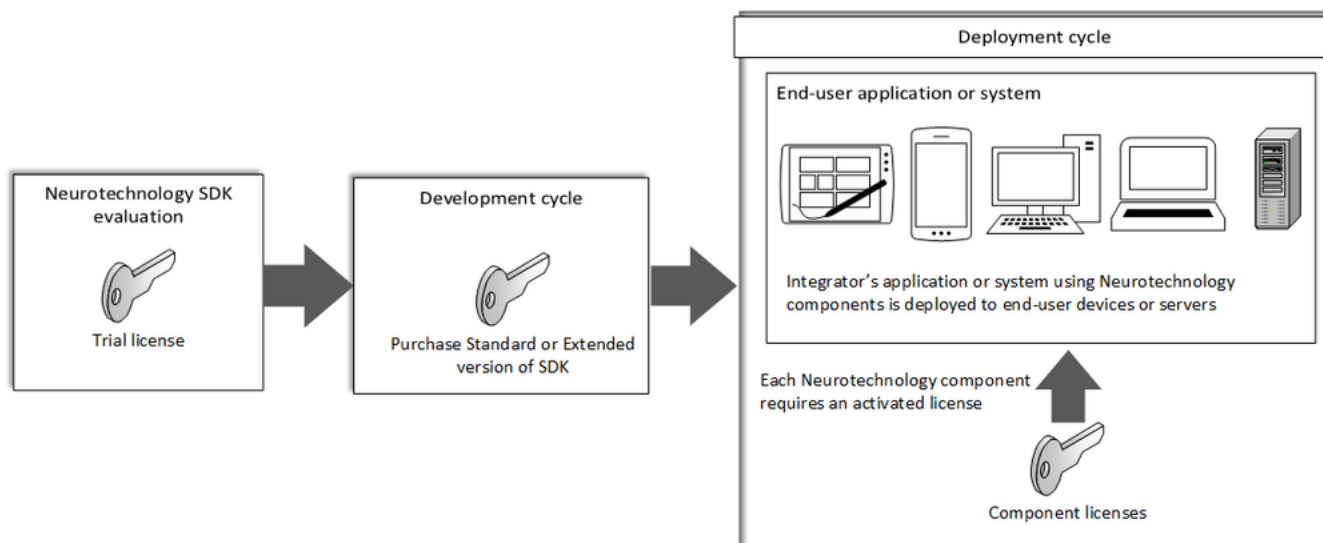
Neurotechnology allows to use the SDK on a development device and use the SDK components for an end user applications or large scale systems. Components and source code provided as samples and tutorials can be adapted by an integrator and redistributed as an end-user application when licenses were activated. We ask you to activate your product to verify that your installation is performed with a genuine Neurotechnology product. Also, product activation ensures that the product license has not been used on more devices than are permitted by the license agreement.

SDK activation is required for trial version and all purchased licenses, standard or extended versions of SDK. When you activate your SDK or license, no personal information is sent to Neurotechnology.

The SDK has the following licenses which should be activated:

- **Trial.** You can copy Neurotechnology SDK to use for 30 days free, no strings attached.
- **Development.** Products development using Neurotechnology technology requires a license for standard or extended version of SDK. SDK license includes several licenses for each component.
- **Deployment.** When Neurotechnology components are used in end-user applications, component licenses for each deployment device is required.
- **VAR.** This is the special licensing agreement between the Neurotechnology and the integrator, who wants to develop and sell the Neurotechnology SDK based development tool.
- **Enterprise license.** Individual licensing agreement for an unlimited use of Neurotechnology components.

The diagram below demonstrates which licenses are required during application life cycle:



## Notes

### Intellectual Property Notices

By buying, using trial version, copying to your computer or other device or activating a license of Neurotechnology SDK you indicate your acceptance of license agreement which is saved in \Documentation\SDK License.html. Neurotechnology software, documentation and other material in this SDK is Copyright © NEUROTECHNOLOGY. 1991–2023. ALL RIGHTS RESERVED.

### Link List

- [Pricing calculator](#) - online tool which helps to determine the cost of products and licenses.
- [Product advisor](#) - online tool which recommends Neurotechnology products based on your system requirements.
- [Product schema](#) - overviews the component licenses included into the SDK.

### Contacts

When you face troubles activating Neurotechnology licenses, please contact us by email [support@neurotechnology.com](mailto:support@neurotechnology.com)

When you need an advice about Neurotechnology products or to estimate required licenses for your project, please contacts us by email [info@neurotechnology.com](mailto:info@neurotechnology.com)

## 2 License types

Each particular SDK component has specific functionality and requires a license. You should note that a license is required for each computer or device that run a component. For example, if you are developing fingerprints enrollment system which will be deployed to 500 computer, you should have 500 fingerprint client licenses .

Typically, Neurotechnology products are activated using Single computer licenses. These licenses can be provided as a serial number, an internet license file or a dongle (a special hardware used to store licenses). Serial numbers are activated using internet connection or by email and after activation internet connection is not required.

Activation by serial number is not suitable for ARM-Linux, except BeagleBone Black and Raspberry Pi 3 devices, or virtual environments. In this case as well as in mobile devices internet licenses can be used. Neurotechnology provides a license file which is stored on a computer or a mobile/embedded device. This licensing option requires internet connection at least every 30 minutes. But the license can be generated (by Neurotechnology) with any suitable time up to 7 days.

When activation via internet is not suitable for your project, a convenient licenses management option is required or virtual environment is used, license may be stored in a dongle. Dongle also can be used in a computer distributing licenses across the devices in the same network.

## 2.1 Serial numbers (Single computer licenses)

Serial numbers are used to activate licenses for particular SDK components on a computer or a device. Each serial number requires activation for a SDK component to run. The activation uses internet connection. When internet connection is not possible, a serial number can be activated by sending an email. After a successful activation the network connection is not required for this licensing.

Neurotechnology provides a way to renew the license if the computer undergoes changes due to technical maintenance.

Serial numbers activation is 3 steps task:

1. Neurotechnology generates serial numbers when a customer purchases licenses for components.
2. Customer using provided serial number generates hardware Id.
3. Generated Id is provided to Neurotechnology. Using this Id, a license file is generated and provided to a customer.

Serial numbers can be activated manually following these steps or automatically skipping these steps. The available activation options are listed in section Activation options ([📄](#) page 8).

### Notes

- Activation by serial number is not suitable for ARM-Linux, except BeagleBone Black and Raspberry Pi 3 devices.
- Activation by serial number is not suitable for virtual environments.
- When a license was activated, client hardware cannot be changed. If hardware was updated or changed, a license should be deactivated ([📄](#) page 28).
- Generated hardware Id can be activated on [Neurotechnology website](#).

### See Also

Serial numbers activation using Activation Wizard ([📄](#) page 11)

Serial numbers manual activation ([📄](#) page 16)

Serial numbers activation for Android ([📄](#) page 22)



## 2.2 Dongle (Volume license manager)

Neurotechnology SDK licenses may be stored in a dongle (volume license manager), hardware-based protection lock storing purchased licenses. When licenses are saved in the USB dongle, activation may be performed without connection to the Internet and is suitable for virtual environments.

The main benefits when using dongle compared to serial numbers are these:

- One dongle can store many licenses for different Neurotechnology products.
- Provides licenses to local devices or other devices over TCP network.
- Dongle can be remotely updated.
- License files can be generated offline from a dongle.
- Licenses are not tied to device's hardware - when one device releases a license, it can be used by another device in the same network (after certain period of time - depending on a license type this period is 15 minutes to 12 hours).

Dongle is used on site by integrators or end users to manage licenses for SDK components in the following ways:

- **Single computer licenses activation.** An installation license for SDK component will be activated for using on a particular device. The license quantity for the SDK component in the license manager will be decreased by the amount of activated licenses.
- **Managing the single computer licenses on network.** The license manager allows to manage licenses for SDK components across the computers on network. The number of managed licenses for a SDK component is limited by the number of licenses in the license manager. No license activation is needed and the license quantity is not decreased. Once issued, the license is assigned to certain computer on the network.
- **Using a license manager as a dongle.** The volume license manager containing at least one license for a SDK component can be used as a dongle that allows to run SDK component installation on a particular computer.

### Notes

If licenses contained in dongle should be used from Docker (or other) containers, it is needed to launch License Activation Service (pgd) on host machine. For other licensing options License Activation service may be running on Docker. These restrictions apply only to License Activation Service, not on other Neurotechnology libraries and services.

### See Also

Dongle activation using Activation Wizard ([📄](#) page 13)

Dongle licenses manual activation ([📄](#) page 16)

## 2.3 Internet licenses

Special license file (called internet license) can be stored on a computer or a mobile/embedded device. This license is constantly checked over the Internet. Internet connection should be available periodically for a short amount of time.

Internet licenses are beneficial when you don't want to run Neurotechnology licensing service. Also, you do not need to activate these type of licenses. Only constant internet connection for a short period of time is required to check license. Also, internet licenses can be transferred to another computer or device by moving the license file there and waiting until the previous activation expires.

You've received internet licenses as \*.lic file(s). Place these file(s) at the root directory of your application (e.g., if using Neurotechnology sample applications, place these files to \Bin\Licenses directory. For Android location is sdcard/Neurotechnology/Licenses. When an application is launched, internet licenses for components are obtained.

The main differences of internet licenses when compared to other license types:

- Internet licenses are not tied to hardware, hardware or device can be changed (devices count is specified in a license; can only obtain from that many different machines, further obtains will fail).
- Internet connection is required to check a license. Network traffic usage: 5-10 minutes connection interval; for 2 licenses ~270kB/day data usage.
- Internet licenses can be used in virtual environments.
- Internet license has time limit for offline usage.
- License files are cached on server and are effective even if user deletes them.

### Notes

Port 80 is used to check licenses over the Internet.

### See Also

Internet licenses activation for Android (📄 page 22)

## 2.4 Trial

All Neurotechnology trial products allow 30 days trial period. After this period you will not be allowed to use trial product.

The following requirements should be met when using trial products:

- **Internet connection.** To use trial product, you must have constant internet connection. Otherwise, you will not be able to use trial product.
- **Activate trial version.** The activation can be performed by two methods: Activation Wizard (only for Windows) or manual activation (for all platforms). Read Activation options (📖 page 8) for activation instructions.
- **Use only trial product on a computer.** If you want to use one of the Neurotechnology trial products, you are not allowed to use any of Neurotechnology licensed products on the same computer at the same time. If you have several licensed products running on a computer, activation services should be stopped when using trial products. This is done during trial products activation.

### Notes

If there is at least one internet license file, SDK works as non-trial.

### See Also

Changing from trial to non-trial (📖 page 29)

Virtual environments licensing (📖 page 30)

Trial activation (📖 page 9)

## 3 Activation options

Neurotechnology licenses can be activated in these ways:

- Using Activation Wizard (📖 page 9) - the activation application for Windows.
- Manual activation (📖 page 16) for Windows, Linux, Mac OS X
- Activation for Android (📖 page 22)

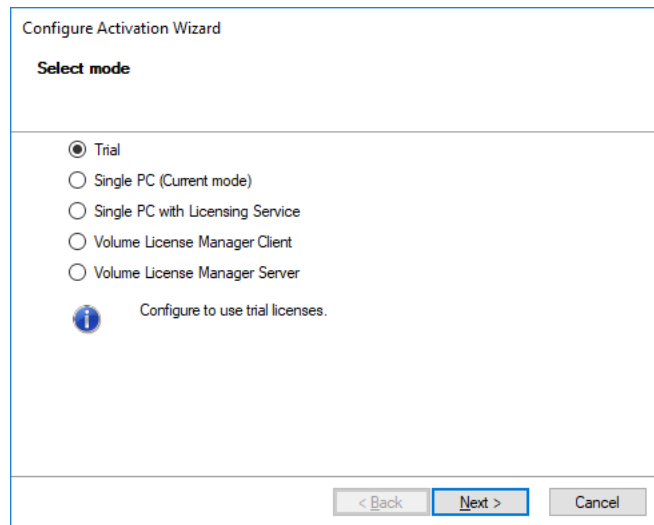
## 3.1 Activation wizard

Activation wizard is a tool for Windows OS which is used to activate purchased licenses or enable trial version of Neurotechnology SDK.

Activation wizard program (*ActivationWizard.exe*) can be started from these locations:

- \Bin\Win32\_x86\Activation\
- \Bin\Win64\_x64\Activation\

When you start Activation wizard, you should select activation mode (alternatively, mode can be selected from the main screen with *Configure* button):



Let's review these activation modes:

- Trial products activation (📄 page 9)
- Single computer licenses activation (📄 page 11)
- Licenses saved in a dongle activation (📄 page 13)

### 3.1.1 Trial activation

1. Select Trial activation mode and press *Next* button.
2. Select products which will be used for trial purposes. When products were selected, Activation wizard will generate *NLicensing.cfg* file in the same directory.

**Configure Activation Wizard**

**Products Selection**  
In order to use the trial, please select products you want to try out

<input checked="" type="checkbox"/> MegaMatcher <input type="checkbox"/> VeriFinger <input type="checkbox"/> VeriLook <input type="checkbox"/> VeriEye <input type="checkbox"/> VeriSpeak <input type="checkbox"/> SentiVigilance <input type="checkbox"/> SentiSight <input type="checkbox"/> SentiGaze <input type="checkbox"/> SentiSculpt <input type="checkbox"/> NCheck <input type="checkbox"/> NVeiler	<input checked="" type="checkbox"/> MegaMatcher Standard SDK <input checked="" type="checkbox"/> MegaMatcher Extended SDK <input type="checkbox"/> MegaMatcher On Card SDK
--	--

MegaMatcher technology is intended for large-scale AFIS and multi-biometric systems developers. The technology ensures high reliability and speed of biometric identification even when using large databases

Show Less

< Back   Next >   Cancel

Press *Next* button.

3. Configure proxy. Trial version of SDK requires constant internet connection to check licenses. If you haven't a direct connection to Internet, proxy should be configured. Otherwise, use *Proxy disabled* setting:

**Configure Activation Wizard**

**Configure Proxy**  
Trial requires constant internet connection. If you experience problems in using Trial, proxy configuration might be needed as some organizations block connections

☐ Proxy disabled (direct connection to Internet)  
☒ Proxy enabled  
 Please, contact your Network Administrator for proxy settings:

Address:

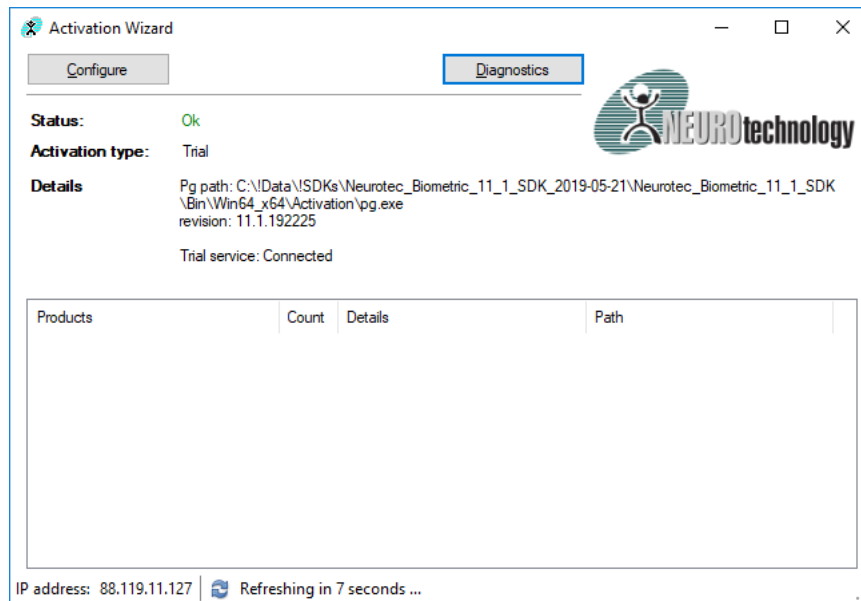
Port:

< Back   Finish   Cancel

By default *Disabled* option is selected. This means that your computer is connected to the internet directly. If you use proxy server for connecting the internet, enable proxy by entering these settings:

- **Address.** IP address of your proxy server (e.g., http://192.168.2.10).
- **Port.** Number of port for proxy server connections.

4. Press *Finish* button. General information about product is displayed: time left for trial products, local and external IP addresses, licensing information (location of licensing service, configuration file mode and licensing file log).



5. If you want to add, remove or change product trial licenses, you should click the *Configure* button. It will open a window identical to the trial product selection window. Note, that if you choose different trial product licenses they will replace the current trial licenses.

6. If you have other Neurotechnology products running on a computer, after starting *Activation Wizard* you will see dialog box prompting you to stop running licensing services (standard PC protection type).

If you choose Yes, running licensing services will be stopped and you'll be able to activate and use Trial product. But if standard PC protection type licensing services will be stopped, you will not be able to use licensed products. If you need to use licensed product again, stop trial product licensing service and start the one of licensed product.

When you finish activation, Neurotechnology licensing service (*pg.exe*) will be running in a background and fully functioning SDK will be available for the period of 30 days.

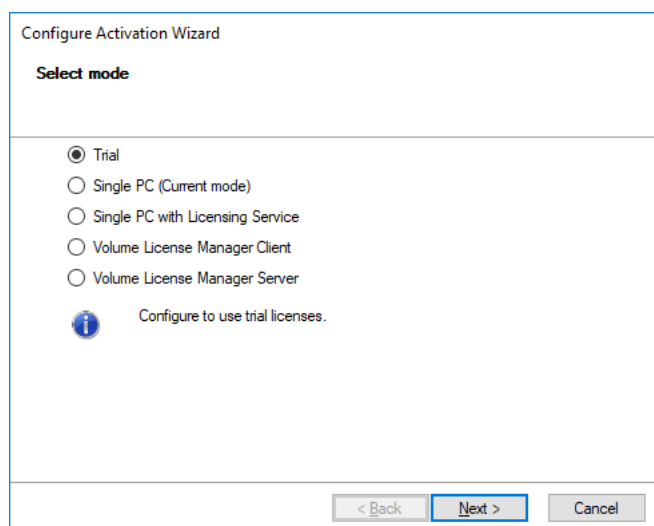
#### Notes

If you need to use licensed product, stop trial product licensing service and start the one of licensed product.

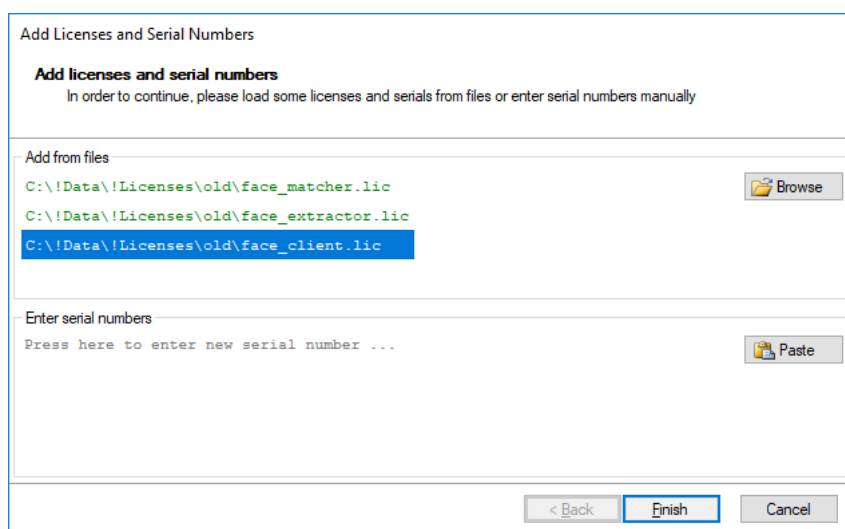
## 3.1.2 Single computer licenses activation

Serial numbers (single computer licenses) activation procedure:

1. Buy license(s) from Neurotechnology or your local distributor.
2. Start *Activation Wizard* application and select activation mode. Note, that if you have currently running Neurotechnology Activation service (*pg.exe*) and it doesn't match the one you are installing (e.g., you were using trial product), it will be removed.



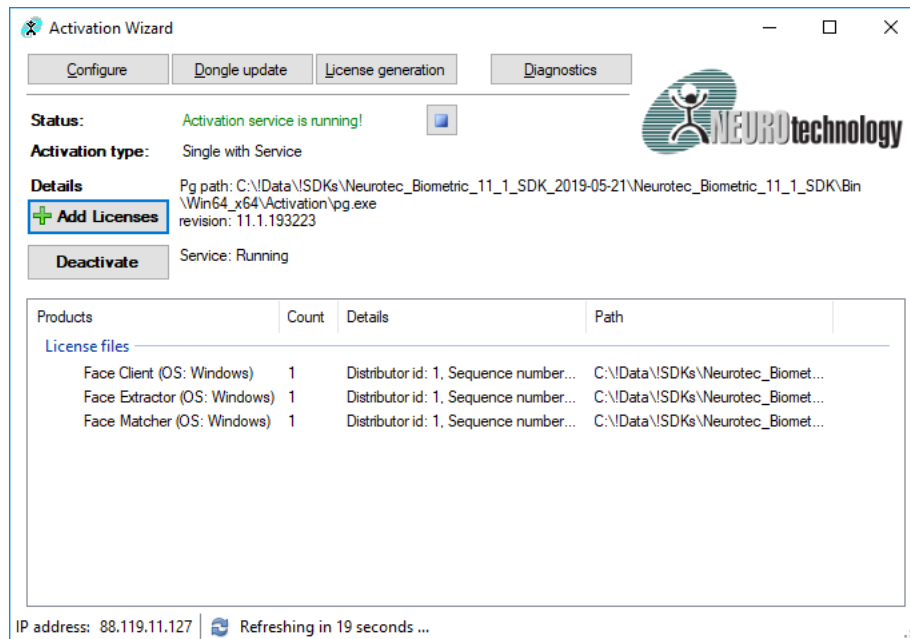
3. In this example let's choose *Single PC with Licensing Service* mode and activate serial numbers. Select *Add* licenses and specify path to files where serial numbers are saved or enter serial numbers manually:



Press *Finish* button.

4. After you hit Finish, you will see such window displaying activated licenses:





5. When licenses were activated, you can start using the SDK. Activated SDK functionality is displayed in the main window of Activation wizard (e.g. picture above shows that licenses for face extractor, matcher and client were purchased and activated). If you need to unlock additional functionality of the SDK, you should obtain additional licenses and activate them following the above procedure. E.g. if you need to extract and match faces and irises, you should buy and activate face and iris extractor, matcher or client licenses.

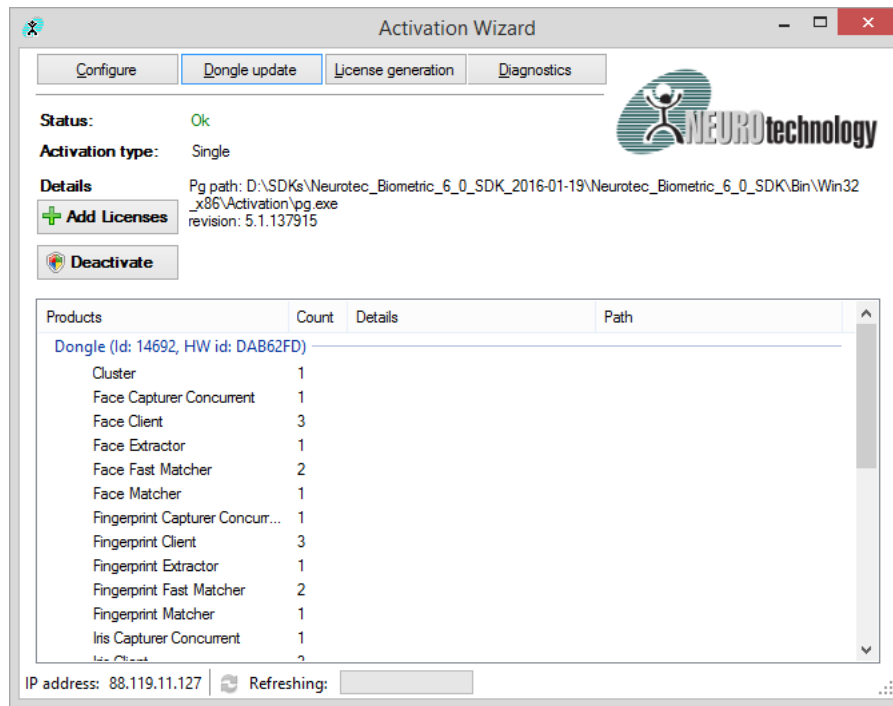
#### Note

When you select *Single PC* mode, licensing service does not run in the background. Use this mode for Single PC or dongle licensing. When using this mode activated licenses ([license\_name].lic files) should be copied to `\Bin\Licenses` folder of SDK. The preferred way is to call `NLicense.Add()` function/method from your application and manually set license content.

## 3.1.3 Dongle activation

### Single PC mode

Insert a Neurotechnology dongle into the USB-port. Wait until dongle drivers are installed. While the dongle is inserted, you have access to all licenses it provides:



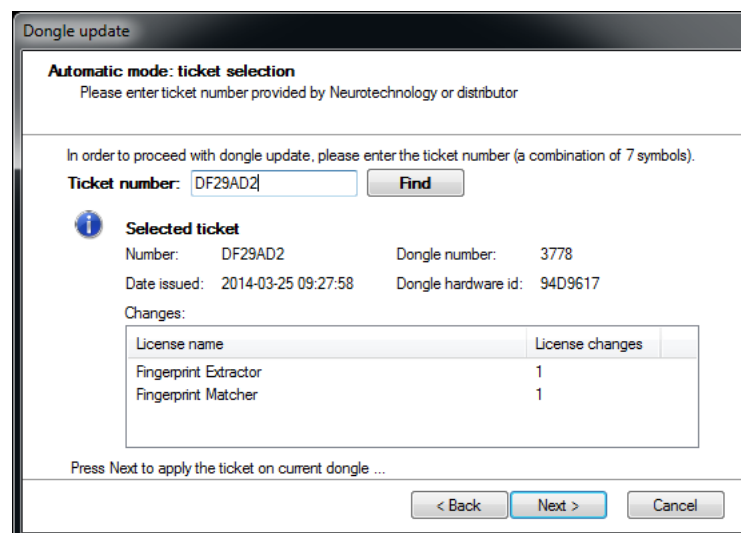
In this example *Single PC* activation mode was selected. This licensing option allows to install and run licensed component on a single PC or on one Server CPU. This means that licenses will be activated for a computer in which dongle was inserted. When licenses from the dongle appear in the *Products* window, no further steps are required. You can call this license in your application (see Licenses obtain in your application ( page 26)).

### Dongle update

Also, in the main *Activation wizard* window you can run *Dongle update*. This option allows to add new licenses to the dongle.

If you click *Dongle update* button, a dongle update wizard will open. Click *Next*. You will be presented with options for automatic and manual dongle update. Choose automatic update (manual update is potentially dangerous; consult Neurotechnology support if you plan on using manual update).

When you buy additional licenses for a dongle update, you will receive a ticket number. In the next window you will be prompted to enter this number, provided by Neurotechnology or distributor (each ticket can only be used once). Click *Find*.



Corresponding ticket data will be displayed, allowing you to add licenses to your dongle.

If all steps were successful you may test installation by running a sample application. Should you have any problems, please see Troubleshooting (page 57) section.

### Volume license manager configuration

Volume license manager allows to manage installation licenses for SDK components across the computers on LAN or internet. This licensing option can be useful when a computer needs a license only for a limited period of time (to perform a certain task). After this period this license can be assigned to another device. In this way the number of managed licenses for a SDK component is limited by the number of licenses in the license manager (dongle). No license activation is needed and the license quantity is not decreased. Once issued, the license is assigned to certain computer on the network.

In the Configuration menu, choose *Volume License Manager Client* (when you want to configure a computer to receive licenses from Volume license manager server) or *Volume License Manager Server* (when you want to configure a computer to act as the server for distributing licenses) and click *Next*.

1) When *Server* was selected, you should select network interface for incoming connections (if IP value 0.0.0.0 is entered the server uses all available interfaces). Port tells the server on what port to listen for client connections.

On a computer where server is running and there are unused licenses, Neurotechnology products will be registered.

The screenshot shows the 'Configure Activation Wizard' dialog box with the title 'Configure Volume License Manager Server'. Below the title is the instruction 'Please enter valid IP address and port to continue'. An information icon is followed by the text: 'Select network interface for incoming connections (if IP value 0.0.0.0 is entered the server uses all available interfaces). Port tells the server on what port to listen for client connections.' The 'IP address' field is a dropdown menu showing '0.0.0.0'. The 'Port' field is a text box containing '5000'. There is an unchecked checkbox for 'Enable broadcast configuration'. Below it, 'Broadcast config server group' is a dropdown showing 'default', and 'Broadcast config port' is a text box containing '16669'. At the bottom are three buttons: '< Back', 'Finish' (highlighted with a blue border), and 'Cancel'.

2) When *Client* was selected, you should enter a valid IP address and a port of the server. Client computer does not need to have licenses installed or a dongle plugged-in. It obtains licenses from the server.

The screenshot shows the 'Configure Activation Wizard' dialog box with the title 'Configure Volume License Manager Client'. Below the title is the instruction 'Please enter valid IP address and port to continue'. An information icon is followed by the text: 'Please enter a valid IP address and a port of the dongle (VLM) Server. Client computer does not need to have dongle plugged-in. It obtains licenses from the server.' The 'IP address' field is a text box containing '0.0.0.0'. The 'Port' field is a text box containing '5000'. There is an unchecked checkbox for 'Enable broadcast configuration'. Below it, 'Broadcast config server group' is a dropdown showing 'default', 'Broadcast config port' is a text box containing '16669', and 'Broadcast config IP address' is a text box containing '255.255.255.255'. At the bottom are three buttons: '< Back', 'Finish' (highlighted with a blue border), and 'Cancel'.

## 3.2 Manual activation

Neurotechnology products can also be activated manually by customizing configuration file and starting licensing service manually. This licensing option is used for **Linux** and **Mac OS X** operating systems (if product supports them) or when Activation wizard for **Windows** can not be used. In this case a client should edit configuration file (*pgd.conf*) manually. Activation configuration file (*pgd.conf*) should be saved in *Activation* folder of the SDK (Bin\[platform]\Activation).

### 3.2.1 Trial

Open *pgd.conf* you licensing configuration file. You will see these settings:

- **mode** – activation mode. 3 modes are available: single, server and gateway. When using trial product *Server* should be used.
- **address** - IP address of a proxy server. It is an advanced setting when you are using proxy server for internet connection.
- **trial** - enables trial period of SDK when *trial = true* is used.
- **Port** - number of port for proxy server connections.

If proxy server is not used, *address* and *port* fields are not required.

3 different *pgd.conf* file configurations are available:

1. Direct license usage:

```
Mode = single
```

2. Licensing server without proxy:

```
Mode = server
```

3. Proxy server licensing option:

```
Mode = gateway
address = 192.168.2.10
port = 80
```

In the most cases, direct licensing mode is used for trial licenses:

```
Mode = single
trial = true
```

Save configuration file with these settings to Bin\[platform]\Activation directory.

Start licensing service:

```
For Linux and Mac OS X: sudo ./run_pgd.sh start
For Windows: pg.exe -install
```

Using the same commands, licensing service can be stopped:

```
For Linux and Mac OS X: sudo ./run_pgd.sh stop
For Windows: pg.exe -uninstall
```

### 3.2.2 Serial numbers (Single computer licenses)

A single computer license allows running SDK component installation on one computer. Customers have to activate a license on each PC after before using SDK component.

Before license activation, you should edit configuration file (`pgd.conf`). `pgd.conf` file parameters:

Parameter	Values
Mode	<ul style="list-style-type: none"> <li><i>Single</i> - default for single computer licenses.</li> <li><i>Server, Gateway</i> - advanced activation option when licensing server or client is used.</li> <li><i>NoPg</i> - no licensing service running in the background. Use this mode for Single PC or dongle licensing. When using this mode activated licenses (<code>[license_name].lic</code> files) should be copied to <code>Licenses</code> folder. This folder should be placed in the root directory of your application. The preferred way is to call <code>NLicense.Add()</code> function/method from your application and manually set each activated license file content.</li> </ul>
LicenseFile	License file path. License file should be placed in the same directory as <code>pg.exe</code> or <code>pgd</code> or full path to a license file should be specified.
LicenceUsageLogFile	Used to specify a location of license usage logs file.
address	IP address of a proxy server. It is an advanced setting when you are using proxy server for activation.

## Linux and Mac OS X manual activation procedure

### 1. Generate computer Id file

To generate `Neurotechnology.id` file please use `id_gen` in appropriate directory:

`Bin/Linux_x86/Activation/id_gen` (32bit Linux platform)

`Bin/Linux_x86_64/Activation/id_gen` (64bit Linux platform)

`Bin/MacOSX_universal/Activation/id_gen` (OS X platform)

For example:

```
./id_gen sn.txt Neurotechnology.id
```

`sn.txt` - file with the serial number used to generate `Neurotechnology.id` (hardware Id file)

Note: these files should be saved in the same directory as `id_gen`.

### 2a. Get license file from Neurotechnology website

Generated computer Id file (`Neurotechnology.id`) should be activated online. Online activation is done by uploading Id file to site: <http://www.neurotechnology.com/cgi-bin/nla.cgi> Alternatively, Id file can be sent to Neurotechnology ([support@neurotechnology.com](mailto:support@neurotechnology.com)) or distributor from which SDK was acquired. Please send it only when online activation fails.

After a successful activation you will receive the activated license file. Generated license file should be saved in the same folder as `pgd` (`Bin\linux_x86\Activation`, `Bin\linux_x86_64\Activation` or `Bin\MacOSX_universal\Activation` depending on the platform).

If you need to save in another location, specify new location in `pgd.conf` file using `LicenseFile` variable. For example:

```
LicenseFile = C:/Neurotec_Biometric_6_0_SDK\Bin\Licenses\FaceClient_Linux.lic
Note: each license file should be specified separately.
```

### 2b. Run license activation tool

Another way to activate a license is `license_activation` tool. It is a command line program used to activate Neurotechnology products.

It can be found in appropriate directory:

`Bin/Linux_x86/Activation/license_activation` (32bit Linux platform)

`Bin/Linux_x86_64/Activation/license_activation` (64bit Linux platform)

Bin/MacOSX\_universal/Activation/license\_activation (OS X platform)

In order to use `license_activation`, enter this in command line:

```
license_activation [serial-number-file] [OPTION] ...
```

Available options:

Option	Description
-h	Prints usage information.
-s serial-number	Specifies serial number. Must be used together with -o parameter.
-o output-file	Specifies output file name.
-i	Generates device id.

#### Usage examples:

Obtains serial number from file `serial-number.txt`, activates it and saves license to `serial-number.lic` file:

```
license_activation serial-number.txt
```

Obtains serial number from file `serial-number.txt`, activates it and saves generated license to `license.lic` file:

```
license_activation serial-number.txt -o license.lic
```

Activates serial number XXXX-XXXX-XXXX-XXXX and saves generated license to `license.lic` file:

```
license_activation -s XXXX-XXXX-XXXX-XXXX -o license.lic
```

Generates device id and saves to `device.id` file:

```
license_activation -i -o device.id
```

### 3. Run Neurotechnology licensing service

Run this command and start activation service:

```
./run_pgd.sh start
```

Make sure that Neurotechnology service is running while using SDK ("`./run_pgd.sh log`" command can be used to test status) and `pgd.conf` shows the right license file place.

`pgd.conf` example:

```
Mode = Single
LicenseFile = Neurotec_Biometric_6_0_SDK\Bin\Licenses\FaceClient_Linux.lic
```

## Windows manual activation procedure

### 1. Generate computer Id file

Please run `id_gen.exe` with parameters:

```
[Activation_directory*]/id_gen.exe <serial number file name> <computer id file name>
```

This command generates computer Id file `Neurotechnology.id`. Example:

```
id_gen.exe sn.txt Neurotechnology.id
```

`sn.txt` - file with the serial number used to generate `Neurotechnology.id` (hardware Id file)

Note: these files should be saved in the same directory as `id_gen.exe`.

### 2. Get license file

Generated computer Id file (`Neurotechnology.id`) should be activated online. Online activation is done by uploading Id file to site: <http://www.neurotechnology.com/cgi-bin/nla.cgi> Alternatively, Id file can be sent to Neurotechnology ([support@neurotechnology.com](mailto:support@neurotechnology.com)) or distributor from which SDK was acquired. Please send it only when online activation fails.

After a successful activation you will receive the activated license file. Generated license file should be saved in the same folder

as `pg.exe` (`Bin\Win32_x86\Activation` or `Bin\Win64_x64\Activation` depending on the platform).

If you need to save in another location, specify new location in `pgd.conf` file using `LicenseFile` variable. For example:

```
LicenseFile = C:/Neurotec_Biometric_6_0_SDK\Bin\Licenses\FaceClient_Windows.lic
```

Note: each license file should be specified separately.

### 3. Run Neurotechnology licensing service

After the configuration and license files were added to appropriate folders, licensing service should be started. To install licensing service on Windows, run this in command line (`pg.exe` is saved in *Activation* folder):

```
pg.exe -install
```

If you need to stop licensing service, you can run this command:

```
pg.exe -uninstall
```

**Note:** licensing service should be run with administrator privileges. If you are using command prompt, run it as an administrator.

It is recommended to use Activation wizard for Windows.

\* Activation directory - folder where activation files are saved. One of these:

- `Bin\Linux_x86\Activation`
- `Bin\Linux_x86_64\Activation`
- `Bin\MacOSX_universal\Activation`
- `Bin\Win32_x86\Activation`
- `Bin\Win64_x64\Activation`

#### Notes

After these 3 commands, Neurotechnology license(s) were activated and functionality of SDK was unlocked. You should note that a particular license unlocks not all functionality. For instance, if you have face client license and need to perform fingerprints matching, you should obtain additional fingerprint licenses. See [Product schema](#) for detailed information.

## 3.2.3 Volume license manager (dongle)

As described in the chapter dongle activation for Windows (page 13), volume license manager (dongle) is used to store purchased licenses. Dongle allows to manage licenses and activate them without internet connection. Dongle can be used in several ways for manual licenses management:

- Generating serial numbers for single computers activation.
- Managing licenses on the same network when the same license can be shared among devices.
- Set up of server for licenses distribution or client receiving licenses.

### 3.2.3.1 Generating licenses for single computers activation

Neurotechnology SDKs customers can generate single computer licenses independently from Neurotechnology or local distributor. Neurotechnology provides USB dongle with purchased licenses. In this case customers should develop their own licenses managing software which will be used to generate single computer licenses from a dongle. Licenses managing software is developed using `Licensing` library which is included into SDK. For more information about how to use this library, read SDK documentation.

Also, SDK package includes tutorials for C/C#/Java/VB.NET which demonstrates how to generate serial number from a dongle. These tutorials are saved in `Tutorials\Licensing` folder (*SerialNumberGenerationFromDongle*). Basically, you should call serial number generation function/method from `NLicenseManager` which is a part of *NLicensing* library and provides licenses

management functionality. For C# application will look like this:

```
//...
int sequenceNumber = int.Parse(args[0]);
uint productId = uint.Parse(args[1]);
int distributorId;
string serialNumber = NLicenseManager.GenerateSerial(productId, sequenceNumber, out
distributorId);
Console.WriteLine("serial number: {0}", serialNumber);
Console.WriteLine("distributor id: {0}", distributorId);
//...
```

NLicenseManager.GenerateSerial method uses provided product Id and user assigned sequence number to generate the serial number. Also distributor Id is returned. Product Id can be retrieved by calling NLicenseManager.GetProductIds() method.

#### Notes

For Windows serial numbers from a dongle can be generated using Activation wizard (🔗 page 13) (see *License generation using dongle*).

### 3.2.3.2 Managing licenses on network

The license manager allows sharing SDK installation licenses across the computers on the same local area network. The number of shared licenses is limited by the number of licenses in the attached dongle.

To share licenses on the network, Neurotechnology service configured as **server** (Volume license manager server) must run on one PC and dongle must be plugged in. In addition, Neurotechnology service configured as **client** (Volume license manager client) must run on all other PCs that are using SDK and need to retrieve a license.

Before starting license management services, you should edit license configuration file (pgd.conf) in each computer connected to the same network. This file is saved in Bin\[platform]\Activation folder.

**Note:** configurations for volume license manager (VLM) server and client differ. SDK distribution should be copied to each computer.

pgd.conf file parameters:

Parameter	Values for dongle (VLM) server	Values for dongle (VLM) client
mode	server	gateway
address	IP address or host name of network card for incoming connections. 0.0.0.0 - default	VLM server address or host name.
port	Port to listen. 5000 - default	License server port. 5000 - default

#### Volume license manager server setup

1. Plug in dongle into server computer
2. Configure pgd.conf file. This file should have the following structure:

```
mode = server # "server" is mandatory here to run as LAN manager(Dongle (VLM) Server)
address = 0.0.0.0 # <IP address of network card for incoming connections>
port = 5000 # <port, to listen>
```

**Note:** if 0.0.0.0 is used as address server will listen to all IP addresses used in networks cards.

3. Save pgd.conf file in Bin\[platform]\Activation folder.
4. Run Neurotechnology licensing service:

for Windows:

```
pg.exe -install
```



for Linux and Mac OS X:

```
./run_pgd.sh start
```

5. Make sure that Neurotechnology service is running while using SDK. To find the service in Windows: "Start -> Control Panel -> Administrative Tools -> Services -> Neurotechnology". Make sure that service is started. In Linux and Mac OS `./run_pgd.sh log` command can be used to test status.

### VLM client setup

When `pgd.conf` file for licenses distributing server was added to *Activation* folder and licensing service was started, client computers should be configured.

`pgd.conf` file for client should have the following structure:

```
mode = gateway # "gateway" is mandatory here to run as LAN client (Dongle (VLM) Client)
address = 192.168.1.1 # <license server address>
port = 5000 # <license server port number>
```

Using the same commands as for VLM server (`pg.exe -install` or `./run_pgd.sh start`), Neurotechnology licensing server should be started.

After this configuration application running in client computers can obtain licenses for components. Server will share licenses for these computers across the network (for configured client computers).

### Notes

When using this licensing option, the same license can be used only in one computer at the same time. When the license is not required any more in a client PC it is returned to the dongle. But you should note that it takes several hours. So if the amount of client computers exceeds the amount of licenses saved in a dongle, other computers will retrieve a license after several hours.

## 3.3 Activation for Android

Neurotechnology licenses for Android should be activated before using product. Activation methods varies depending on what license you have.

This section demonstrates how to activate a license using Multimodal sample for Android (`\Bin\Android\multibiometric-sample-android.apk`).

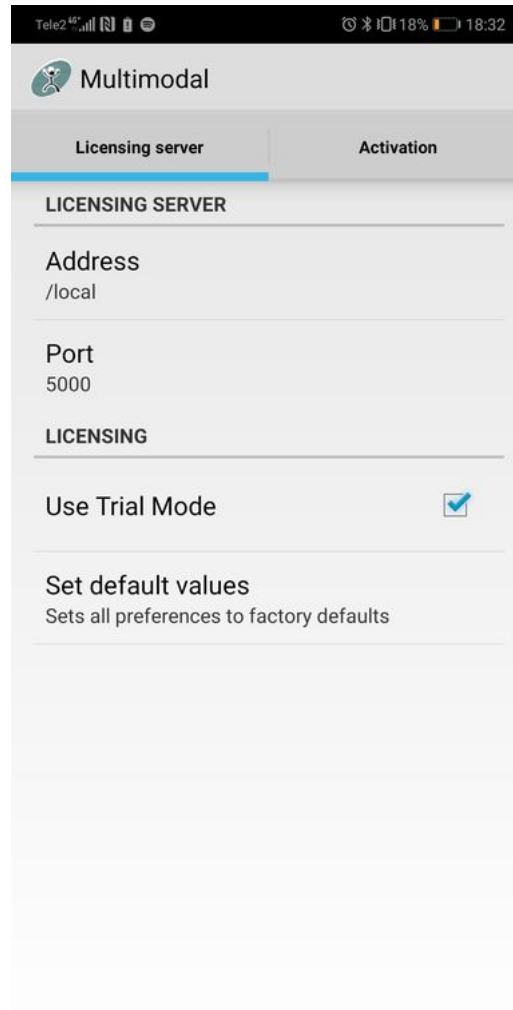
1. Copy `multibiometric-sample-android.apk` file to your Android device and and install it.

**Note:** installation can be run only when installation from *Unknown sources* is enabled. Go to your Android device settings and enable *Settings->Security->Unknown sources->OK->Trust*.

### Trial licenses activation

2. Launch an installed application. By default this application is configured to obtain a trial license. Internet connection is required for trial licenses activation. If you do not have direct access to the internet, you can set-up trial product to work through proxy server. Proxy server settings can be entered in *Connection Settings* window. Choose *Settings->Activation* in the right corner. In this window you can configure licensing server - specify address and port.

**Note:** usage of the Trial SDK is limited up-to 30 days. Trial SDK requires constant internet connection. WiFi connection is required to activate trial licenses.



### Purchased licenses activation

3. Disable trial mode (step 2 - uncheck "Use Trial Mode") and restart application.

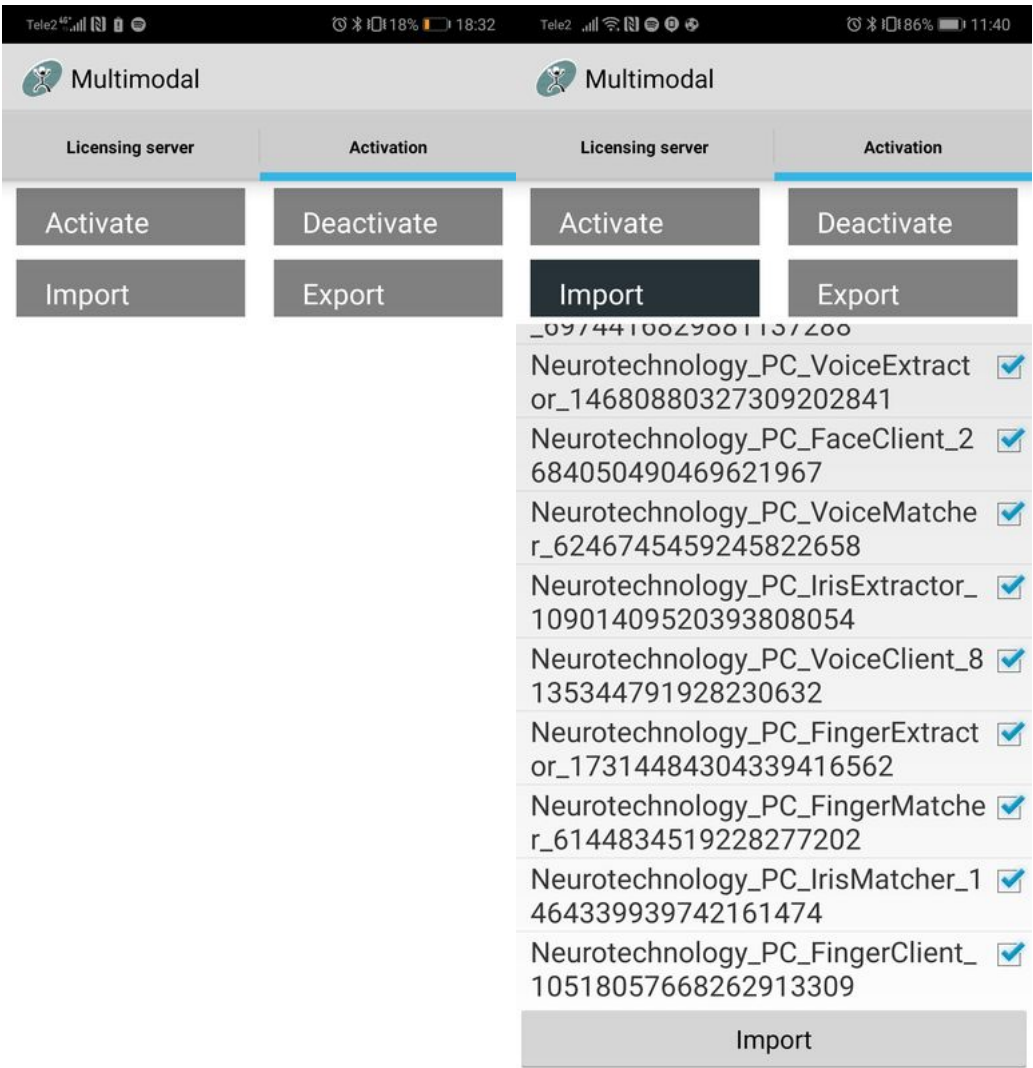
4. When you want to activate purchased licenses, you should copy license files to this default directory: [Internal storage]/Neurotechnology/Licenses. Serial numbers and internet licenses must be copied to this directory, otherwise they will not be available. Activated licenses are moved from *Licenses* directory to application's sandbox.

Internet licenses do not require activation, only serial numbers should be activated.

**Note:** User can define other path to license file (source code changes are required). Also you should note that when using Standard licenses temporary internet connection may be required.

5. Go to *Settings->Activation*. These options for licensing are available:

- *Activate* - activates selected serial number licenses. After a successful serial number activation, application functionality is unlocked.
- *Deactivate* - deactivates selected licenses. You should note, that deactivated license can be used again for activation after several hours. Licenses deactivation require internet connection.
- *Import* - imports selected internet licenses. Licenses (\*.lic files) are imported from [Internal storage]/Neurotechnology/Licenses directory. Imported licenses unlock application's functionality. E.g., if you need to extract face template, you need to import *Face Client* and *Face Extractor* internet licenses.
- *Export* - exports selected internet licenses to [Internal storage]/Neurotechnology/Licenses. Exported licenses are not available for application until they are imported again.



**Notes**

If you are using Trial version of SDK, sample application will try to activate licenses automatically. Usage of Trial SDK is limited up-to 30 days. Trial SDK requires constant internet connection.

Android licenses are activated per application sandbox. This also enables to use license deactivation feature. Please note, that during development application should be updated instead of deleting and installing again, otherwise license could be lost. Also, other activation options such as using licenses from PC's dongle can be more convenient for development purposes. Read more about dongle activation (📖 page 20).

Each Android must contain specific license(s) activated depending on the functionality being used. If you get „Operation not activated“ message, that means there is/are no required license(s) activated depending on the functionality you use.

## 4 Advanced licensing

## 4.1 Licenses obtain in your application

When you have activated a license(s), you can start using licensed biometric components in your application. Before using licensed functionality in your application, you should obtain licenses for each component. Some usage examples:

- Fingerprint enrollment from a scanner requires licenses for these components: *Biometrics.FingerExtraction, Devices.FingerScanners*.
- Face verification requires licenses for these components: *Biometrics.FaceExtraction, Biometrics.FaceMatching*.
- Facial features detection: *Biometrics.FaceDetection, Biometrics.FaceExtraction, Biometrics.FaceSegmentsDetection*.

Basically, each Neurotechnology API component requires a license. Developer's guide (e.g. *Neurotechnology Biometric SDK.pdf*) saved in *Documentation* folder of the SDK has the chapter named *Licensed API functionality* (About->Licensing). This section lists down which API functionality is enabled by which license. Using the table from this section you can check if a particular component is unlocked by the license you have. For example, if you have bought *Fingerprint Client* license then you can use such components as *Biometrics.FingerExtraction, Biometrics.FingerDetection* or *Media*. But if you need to perform fingerprint segmentation which is accessed from *Biometrics.FingerSegmentation* component, you should purchase additional license *Fingerprint Segmenter*.

Sometimes it can be a tricky task to decide which component you should use and which licenses are required. SDK includes tutorials (*/Tutorials* folder) for C/C#/VB.NET/Java languages which demonstrate how to perform a biometric task and how to obtain and release licenses. Also previously mentioned Developer's guide includes *API Reference* documentation.

Let's see how licenses are obtained in *Detect facial features* tutorial for C#:

```
//...
// Let's specify licensed components names. These names are taken from the previously
// mentioned table
// Face detection and face features detection are performed by NBiometricClient.
// Face segments detection is defined as an additional component, because face
// detection/extraction and segmentation
// are separate tasks and may require separate licenses.
const string licenses =
    "Biometrics.FaceDetection,Biometrics.FaceExtraction,Biometrics.FaceSegmentsDetection";

//Checks whether TRIAL mode is enabled or not. To use purchased licenses, don't use below
//code line.
//GetTrialModeFlag() method takes value from "Bin/Licenses/TrialFlag.txt" file. So to
//easily change mode
//for all our examples, modify that file.
//Also you can just set TRUE to "TrialMode" property in code.
NLicenseManager.TrialMode = TutorialUtils.GetTrialModeFlag();
Console.WriteLine("Trial mode: " + NLicenseManager.TrialMode);

//Now let's try to obtain these licenses:
try
{
    // Obtains licenses for specified "licenses" from licenses manager server "local" using
    // "5000" server's port
    if (!NLicense.Obtain("/local", 5000, licenses))
    {
        throw new ApplicationException(string.Format("Could not obtain licenses : {0}",
licenses));
    }

    // Perform Facial features detection. See tutorial source code
}
//It is required to release licenses after biometric task was performed
finally
{
    NLicense.Release(licenses);
}
```

As you see, licenses are obtained using `NLicense.Obtain()` function/method and released when not used with `NLicense.Release()`. Licenses for components are obtained in the same way for other biometric tasks.

You should note that all activated licenses by default should be saved in Licenses folder in the root directory of your application (single licenses scenario). If you need to change the location of licenses (`[license_name].lic` files) you should call `NLicense.Add()` function/method from your application and manually set each activated license file content.

## 4.2 Licenses deactivation

Neurotechnology licenses work only in a specific device on which activation was carried out (unique hardware Id for this device is generated). But in some cases a license should be stopped (deactivated) on this device and activated again. Some typical situation when licenses deactivation is required:

- User tested a license in one device and needs to transfer it to other device (a license can work only in one device at the same time).
- Device in which license was activated is malfunctioned or hardware components such as processor or hard disk have been changed.
- User changed a device (a computer).
- Also it is strongly recommended to deactivate a license before reinstalling an operating system or installing a different OS version.

Deactivation should be performed on the same device where it was activated. When you have internet connection, a license will be deactivated automatically by using *Activation Wizard* or *LicenseDeactivation* (\Tutorials\Licensing\C) tutorial. When a device is not connected to the internet, you should perform manual deactivation and upload generated deactivation Id and license's file to Neurotechnology [website](#). If you generate deactivation Id using console (`id_gen -dp <product name>`), a license stops working on a device it was activated and it is required to complete license's deactivation procedure by submitting license file and deactivation id so that license could be reactivate on another device (or even the same computer/device).

The simplest way to deactivate a license is using *Activation wizard* in a device connected to the internet. Press the *Deactivate* button and choose a license. After a short period of time a license will be deactivated and removed from the licenses list. Such license can be activated again in other device.

Also licenses deactivation can be performed manually using the command line tool called `id_gen`. `id_gen` is a command line program used to generate computer identifier file for Neurotechnology components registration or licenses deactivation. This tool is saved in the SDK's `Bin\[platform]\Activation` directory. Run it with administrator privileges and use this command to deactivate a license (Windows OS):

```
id_gen -dp <product name> <deactivation Id file name>
e.g.:
id_gen.exe -dp <VeriFinger> <deactivation.id>
```

The *deactivation.id* file from this example will be created in the `id_gen` tool directory. This file and license file can be uploaded to Neurotechnology website for a deactivation. Note: upload deactivation file only when you do not have internet connection in a device (computer).



## 4.3 Changing from trial to non-trial

When you download and start using the SDK, trial mode is enabled by default. As described in section Trial (📄 page 7), trial mode is limited to 30 days. When you've finished exploring SDK functionality, your 30 days trial is over or you purchased licenses, you can keep using the same SDK, only trial mode parameter should be disabled.

`\Bin\Licenses` directory of SDK contains `TrialFlag.txt` file. Using `TRUE` or `FALSE` values within this file are the simplest way to enable or disable the trial mode of SDK (`TRUE` - trial mode is enabled, `FALSE` - non-trial mode of SDK is used).

Also, the trial and non-trial version of the SDK is controlled by using the following methods/parameters:

- C: `NLicManSetTrialMode()`
- C++: `NLicenseManager::SetTrialMode()`
- Java: `NLicenseManager.setTrialMode()`
- C# / VB.NET: `NLicenseManager.TrialMode`

So change from trial to non-trial version of the SDK can be achieved in these ways:

1. Delete `TrialFlag.txt` file.
2. Modify `TrialFlag.txt` to `FALSE`.
3. Modify the source code of a sample/tutorial you want to use.

The first 2 options will change to non-trial version of the SDK for all samples and tutorials.

When you change from trial to non-trial, you need to activate purchased licenses (📄 page 8) and correctly obtain them in your application (📄 page 26).

## 4.4 Virtual environments licensing

Neurotechnology SDKs can be used in virtual environments as well. Like a physical machine, a virtual machine running any of Neurotechnology product requires licenses activation.

### Trial version of SDK

The licensing for trial version of the SDK is not supported on virtual machine out of the box. It must be configured first. To use trial version of the SDK on a virtual machine you need:

1. Configure licensing service on a physical machine (VM host) and start it. Let's call it "Volume License Manager Server". See section Managing licenses on network (🔗 page 20) for more information.
2. When Volume license managers server is running, there are 2 options to use SDK from virtual machine:
  1. Use licensing service. In such case you need to configure licensing service to obtain licenses from "Volume License Manager Server".
  2. Do not use licensing service. In such case you need to change network address `/local` to an IP address of "Volume License Manager Server" when calling one of these functions/methods: `NLicenseObtain()` / `NLicense::Obtain()` / `Nlicense.obtain()` / `Nlicense.Obtain()`.

### Non-trial version of SDK

Virtual machine can get license from these sources:

1. *Dongle (🔗 page 5) connected to the host machine* if virtual environment is configured to access host machine's USB port. This is not recommended way as some virtualization software have USB issues and solution can be not stable.
2. *Dongle connected to some machine available over network* (or even the same machine where VM is running). This host machine acts like Volume license manager server. Virtual machine connects to this server to obtain licenses. Licenses sharing over the network is described in Managing licenses on network (🔗 page 20) section.
3. *Neurotechnology licensing server*. Internet licenses (🔗 page 6) can be used for virtual environments. In this case a virtual machine would communicate with Neurotechnology licensing server. If due some reason communication would be interrupted, a virtual machine can live without communication up to 1 week (if there is no need to restart it, but after restart internet connection is required immediately). By default internet based license requires internet connection every each 30 minutes.

### Notes

Licenses provided as serial numbers are not supported on virtual machines.

## 4.5 Disaster recovery licenses

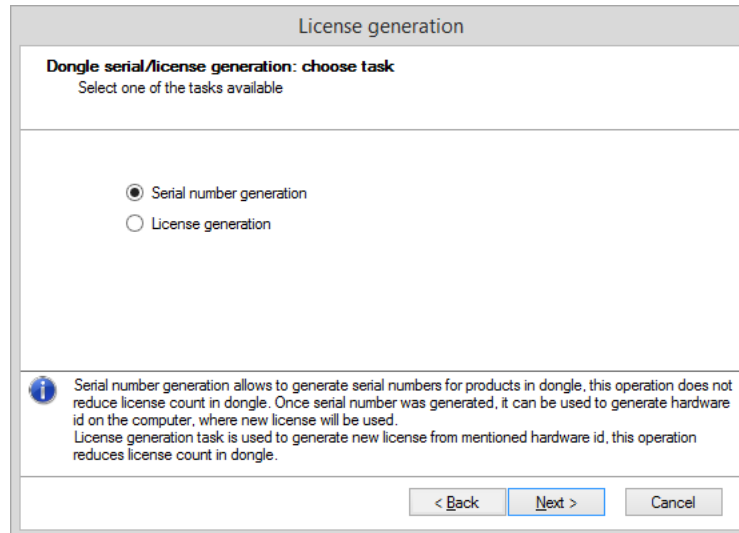
Disaster recovery licenses for server-side components of Neurotechnology products are intended for using in disaster recovery centers (DRC). A DRC is a location which has the same equipment as the primary site, completely mirrors the data environment of the primary site and is on standby while the primary site is working. If the primary site fails, the DRC takes over operations.

Disaster recovery units require the same licenses as the main units. Licenses for disaster recovery components are sold with a discount.

## 4.6 Licenses generation from a dongle

### License generation using Activation Wizard

Licenses saved in a dongle can be used to generate a serial number or a license file. Select *License generation* in the top of Activation wizard. New license generation window appears:



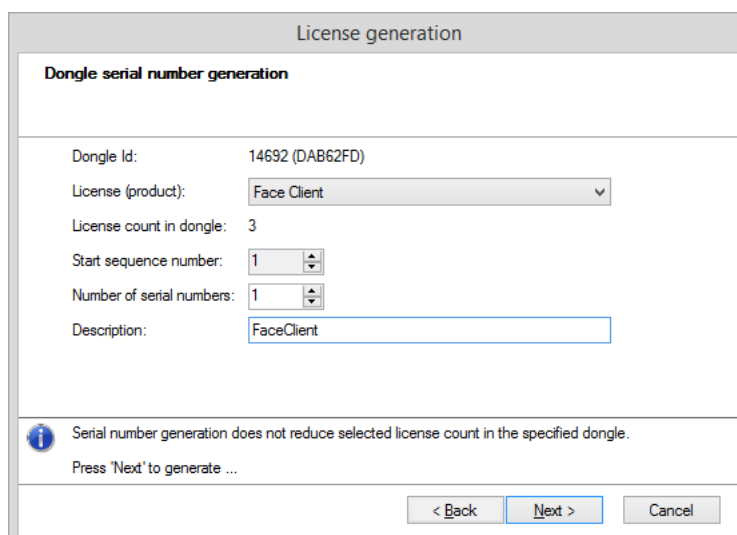
In this window you can generate:

- *Serial number*. Serial numbers generation allows to generate serial numbers for products in a dongle, this operation does not reduce license count in dongle. Once serial number was generated, it can be used to generate hardware Id on a device, where new license will be used.
- *License*. When you have generated the serial number and used it to generate the hardware Id, *License generation* option should be used to generate a new license. License generation is performed using generated hardware Id from the previous step. Also, this option reduces license count in a dongle, because new license is generated.

License generation from a dongle is used by customers who need to manage licenses without connecting to the internet. Also it is an easy way to save many licenses.

Let's see how to generate a license:

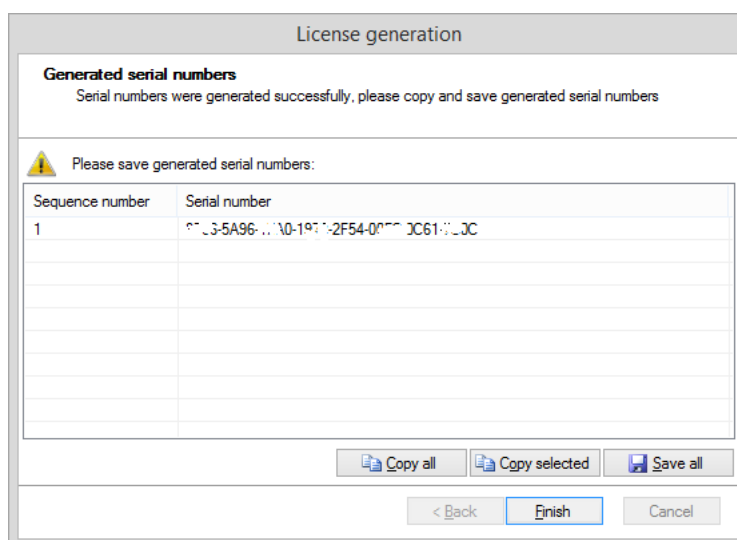
1. Select *Serial number generation* and press *Next*. In the next window select the license (product) and number of serial numbers to generate (can not exceed the license count in dongle). Also you can add a description for licenses and change the sequence number (used for licenses enumeration):



The 'License generation' dialog box contains the following fields and controls:

- Dongle Id:** 14692 (DAB62FD)
- License (product):** Face Client (dropdown menu)
- License count in dongle:** 3
- Start sequence number:** 1 (spin box)
- Number of serial numbers:** 1 (spin box)
- Description:** FaceClient (text box)
- Information:** Serial number generation does not reduce selected license count in the specified dongle. Press 'Next' to generate ...
- Buttons:** < Back, Next >, Cancel

2. Press *Next*. In the next window will be displayed generated serial number. Save it to your computer and press *Finish*.



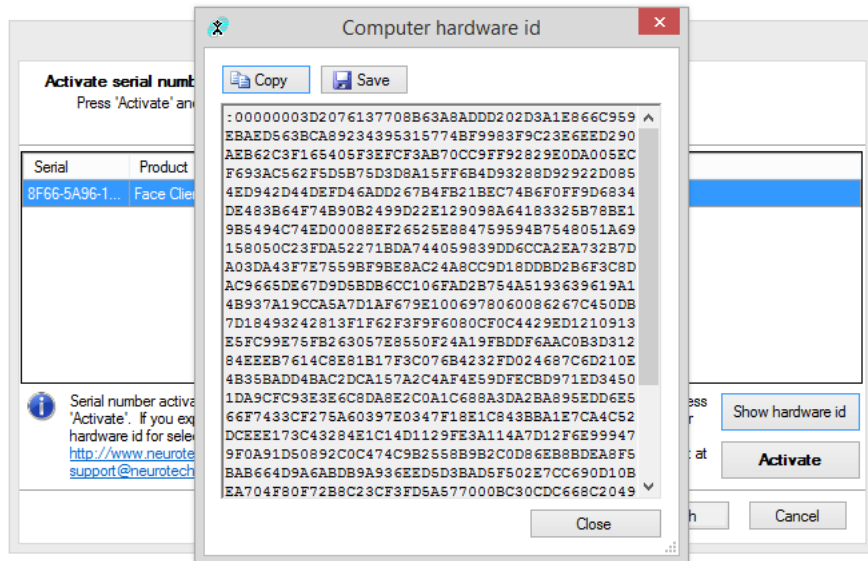
The 'Generated serial numbers' dialog box displays the following information:

- Header:** Generated serial numbers
- Message:** Serial numbers were generated successfully, please copy and save generated serial numbers
- Warning:** Please save generated serial numbers:
- Table:**

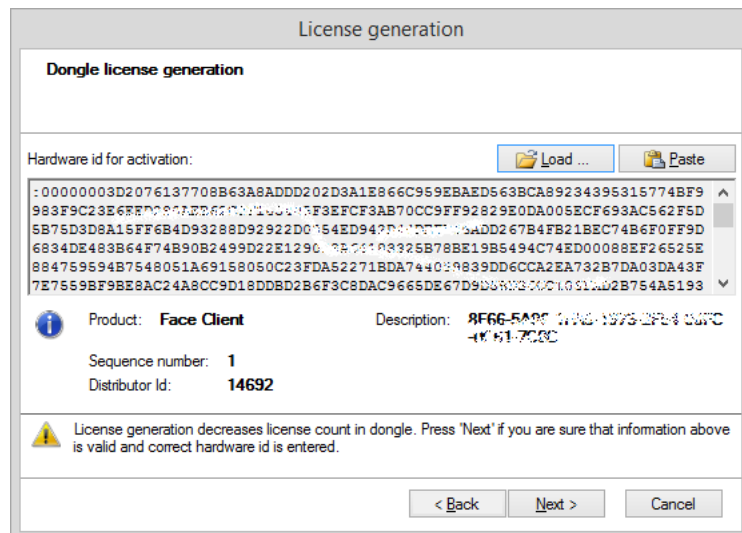
Sequence number	Serial number
1	67_3-5A96_...10-191_2F54-00...DC61...DC

**Buttons:** Copy all, Copy selected, Save all, < Back, Finish, Cancel

4. Generate hardware Id using serial number from the step 3. Open Activation wizard on a computer where you'll use the license. In the main Activation wizard window press *Add licenses* button and specify path to the saved serial number file or manually enter serial number(s) and press *Next* button. In the next window press *Show hardware Id*. Copy or save this Id. It will be used to activate previously generated serial number.



5. Press *License generation* in the main window and open Dongle license generation wizard again. But this time choose *License generation* option. We will use the hardware Id from the step 4 to activate the serial number from the step step 2. Load or paste generated hardware ID and press next button:



License file will be generated. This license can be used in the device where hardware Id was generated. In this device open Activation wizard and press *Add licenses* button.

**Note:** Generated license will not work on a different computer, only where hardware Id was generated.

## 4.7 Licensed API functionality

This section provides the list of Neurotechnology SDKs and which licenses are required to unlock a certain API component.

### Notes

You can check your licenses in *Activation Wizard* tool *Bin/Licenses* directory.

### 4.7.1 MegaMatcher Standard SDK

**MegaMatcher Standard SDK** license names and licensed API components:

License name	API Component (functionality)
Fingerprint Client	Media
	Devices.FingerScanners
	Biometrics.FingerDetectionBase
	Biometrics.FingerDetection
	SmartCards
	Images.Processing.FFT
	Biometrics.FingerExtractionBase
	Biometrics.FingerExtraction
	Biometrics.FingerSegmentsDetectionBase
	Biometrics.FingerSegmentationBase
	Images.WSQ
	Images.IHead
	Images.JPEG2000
	Images.LosslessJPEG
	Biometrics.Standards.Base
	Biometrics.Standards.FingerTemplates
	Biometrics.Standards.FingerCardTemplates
	Biometrics.Standards.Fingers
	Biometrics.Standards.Other
	Biometrics.FingerDetectionMedium
	Biometrics.FingerExtractionMedium
	Biometrics.FingerSegmentsDetection
	Biometrics.FingerSegmentation
	Biometrics.FingerQualityAssessmentBase
	Biometrics.FingerQualityAssessment
	BioAPI.Base
	BioAPI.Fingers

Fingerprint Extractor	Media Devices.FingerScanners Biometrics.FingerDetectionBase Biometrics.FingerDetection SmartCards Images.Processing.FFT Biometrics.FingerExtractionBase Biometrics.FingerExtraction Biometrics.FingerSegmentsDetectionBase Biometrics.FingerSegmentationBase
Fingerprint Fast Extractor	Media Devices.FingerScanners Biometrics.FingerDetectionBase Biometrics.FingerDetection SmartCards Images.Processing.FFT Biometrics.FingerExtractionBase Biometrics.FingerExtraction Biometrics.FingerSegmentsDetectionBase Biometrics.FingerSegmentationBase Biometrics.FingerDetectionMedium Biometrics.FingerExtractionMedium Biometrics.FingerDetectionFast Biometrics.FingerExtractionFast
Fingerprint Matcher	Biometrics.FingerMatching Biometrics.MatchingFusion Cluster.ClusterNode
Fingerprint Fast Matcher	Biometrics.FingerMatching Biometrics.MatchingFusion Cluster.ClusterNode Biometrics.FingerMatchingFast
Fingerprint Capturer	Media Devices.FingerScanners Biometrics.FingerDetectionBase Biometrics.FingerDetection
Fingerprint Fast Segmenter	Biometrics.FingerSegmentsDetectionBase Biometrics.FingerSegmentationBase Biometrics.FingerSegmentsDetection Biometrics.FingerSegmentation Biometrics.FingerSegmentsDetectionFast Biometrics.FingerSegmentationFast



Face Client	Media SmartCards Images.IHead Images.JPEG2000 Images.LosslessJPEG Biometrics.Standards.Base Biometrics.Standards.Other BioAPI.Base Devices.Cameras Biometrics.FaceDetectionBase Biometrics.FaceDetection Biometrics.FaceExtractionBase Biometrics.FaceExtraction Biometrics.FaceSegmentsDetectionBase Biometrics.FaceSegmentationBase Biometrics.Standards.Faces Biometrics.FaceDetectionMedium Biometrics.FaceExtractionMedium Biometrics.FaceSegmentsDetection Biometrics.FaceSegmentation Biometrics.FaceQualityAssessmentBase Biometrics.FaceQualityAssessment BioAPI.Faces
Face Extractor	Media SmartCards Devices.Cameras Biometrics.FaceDetectionBase Biometrics.FaceDetection Biometrics.FaceExtractionBase Biometrics.FaceExtraction Biometrics.FaceSegmentsDetectionBase Biometrics.FaceSegmentationBase
Face Fast Extractor	Media SmartCards Devices.Cameras Biometrics.FaceDetectionBase Biometrics.FaceDetection Biometrics.FaceExtractionBase Biometrics.FaceExtraction Biometrics.FaceSegmentsDetectionBase "Biometrics.FaceSegmentationBase Biometrics.FaceDetectionMedium Biometrics.FaceExtractionMedium Biometrics.FaceDetectionFast Biometrics.FaceExtractionFast
Face Matcher	Biometrics.MatchingFusion Cluster.ClusterNode Biometrics.FaceMatching

Face Fast Matcher	Biometrics.MatchingFusion Cluster.ClusterNode Biometrics.FaceMatching Biometrics.FaceMatchingFast
Face Capturer	Media Devices.Cameras Biometrics.FaceDetectionBase Biometrics.FaceDetection
Face Fast Token Image	Biometrics.FaceSegmentsDetectionBase Biometrics.FaceSegmentationBase Biometrics.FaceSegmentsDetection Biometrics.FaceSegmentation Biometrics.FaceQualityAssessmentBase Biometrics.FaceQualityAssessment Biometrics.FaceSegmentsDetectionFast Biometrics.FaceSegmentationFast Biometrics.FaceQualityAssessmentFast
Iris Client	Media SmartCards Images.WSQ Images.IHead Images.JPEG2000 Images.LosslessJPEG Biometrics.Standards.Base Biometrics.Standards.Other BioAPI.Base Devices.IrisScanners Biometrics.IrisDetectionBase Biometrics.IrisDetection Biometrics.IrisExtractionBase Biometrics.IrisExtraction Biometrics.IrisSegmentsDetectionBase Biometrics.IrisSegmentationBase Biometrics.Standards.Irises Biometrics.IrisDetectionMedium Biometrics.IrisExtractionMedium Biometrics.IrisSegmentsDetection Biometrics.IrisSegmentation Biometrics.IrisQualityAssessmentBase Biometrics.IrisQualityAssessment BioAPI.Irises
Iris Extractor	Media SmartCards Devices.IrisScanners Biometrics.IrisDetectionBase Biometrics.IrisDetection Biometrics.IrisExtractionBase Biometrics.IrisExtraction Biometrics.IrisSegmentsDetectionBase Biometrics.IrisSegmentationBase

Iris Fast Extractor	Media SmartCards Devices.IrisScanners Biometrics.IrisDetectionBase Biometrics.IrisDetection Biometrics.IrisExtractionBase Biometrics.IrisExtraction Biometrics.IrisSegmentsDetectionBase Biometrics.IrisSegmentationBase Biometrics.IrisDetectionMedium Biometrics.IrisExtractionMedium Biometrics.IrisDetectionFast Biometrics.IrisExtractionFast
Iris Matcher	Biometrics.MatchingFusion Cluster.ClusterNode Biometrics.IrisMatching
Iris Fast Matcher	Biometrics.MatchingFusion Cluster.ClusterNode Biometrics.IrisMatching Biometrics.IrisMatchingFast
Iris Capturer	Media Devices.IrisScanners Biometrics.IrisDetectionBase Biometrics.IrisDetection
Palmprint Client	Media SmartCards Images.Processing.FFT Images.Wsq Images.IHead Images.JPEG2000 Images.LosslessJPEG Biometrics.Standards.Base Biometrics.Standards.Other BioAPI.Base Devices.PalmScanners Biometrics.PalmDetectionBase Biometrics.PalmDetection Biometrics.Standards.Palms Biometrics.Standards.PalmTemplates" Biometrics.PalmDetectionMedium Biometrics.PalmExtractionBase Biometrics.PalmExtraction Biometrics.PalmExtractionMedium Biometrics.PalmSegmentsDetectionBase Biometrics.PalmSegmentsDetection Biometrics.PalmSegmentationBase Biometrics.PalmSegmentation Biometrics.PalmQualityAssessmentBase Biometrics.PalmQualityAssessment BioAPI.Palms

Palmprint Matcher	Biometrics.MatchingFusion Cluster.ClusterNode Biometrics.PalmMatching
Palmprint Capturer	Media Devices.PalmScanners Biometrics.PalmDetectionBase Biometrics.PalmDetection
Voice Client	Media SmartCards Biometrics.Standards.Base Biometrics.Standards.Other BioAPI.Base Devices.Microphones Biometrics.VoiceDetectionBase Biometrics.VoiceDetection Biometrics.VoiceExtractionBase Biometrics.VoiceExtraction Biometrics.VoiceSegmentsDetectionBase Biometrics.VoiceSegmentationBase Biometrics.VoiceDetectionMedium Biometrics.VoiceExtractionMedium Biometrics.VoiceSegmentsDetection Biometrics.VoiceSegmentation Biometrics.VoiceQualityAssessmentBase Biometrics.VoiceQualityAssessment BioAPI.Voices
Voice Extractor	Media SmartCards Devices.Microphones Biometrics.VoiceDetectionBase Biometrics.VoiceDetection Biometrics.VoiceExtractionBase Biometrics.VoiceExtraction Biometrics.VoiceSegmentsDetectionBase Biometrics.VoiceSegmentationBase
Voice Fast Extractor	Media SmartCards Devices.Microphones Biometrics.VoiceDetectionBase Biometrics.VoiceDetection Biometrics.VoiceExtractionBase Biometrics.VoiceExtraction Biometrics.VoiceSegmentsDetectionBase Biometrics.VoiceSegmentationBase Biometrics.VoiceDetectionMedium Biometrics.VoiceExtractionMedium Biometrics.VoiceDetectionFast Biometrics.VoiceExtractionFast

Voice Matcher	Biometrics.MatchingFusion Cluster.ClusterNode Biometrics.VoiceMatching
Voice Capturer	Media Devices.Microphones Biometrics.VoiceDetectionBase Biometrics.VoiceDetection

## 4.7.2 MegaMatcher Extended SDK

**MegaMatcher Extended SDK** license names and licensed API components:

License name	API Component (functionality)
Fingerprint Client	Media Devices.FingerScanners Biometrics.FingerDetectionBase Biometrics.FingerDetection SmartCards Images.Processing.FFT Biometrics.FingerExtractionBase Biometrics.FingerExtraction Biometrics.FingerSegmentsDetectionBase Biometrics.FingerSegmentationBase Images.WSQ Images.IHead Images.JPEG2000 Images.LosslessJPEG Biometrics.Standards.Base Biometrics.Standards.FingerTemplates Biometrics.Standards.FingerCardTemplates Biometrics.Standards.Fingers Biometrics.Standards.Other Biometrics.FingerDetectionMedium Biometrics.FingerExtractionMedium Biometrics.FingerSegmentsDetection Biometrics.FingerSegmentation Biometrics.FingerQualityAssessmentBase Biometrics.FingerQualityAssessment BioAPI.Base BioAPI.Fingers

Fingerprint Extractor	Media Devices.FingerScanners Biometrics.FingerDetectionBase Biometrics.FingerDetection SmartCards Images.Processing.FFT Biometrics.FingerExtractionBase Biometrics.FingerExtraction Biometrics.FingerSegmentsDetectionBase Biometrics.FingerSegmentationBase
Fingerprint Fast Extractor	Media Devices.FingerScanners Biometrics.FingerDetectionBase Biometrics.FingerDetection SmartCards Images.Processing.FFT Biometrics.FingerExtractionBase Biometrics.FingerExtraction Biometrics.FingerSegmentsDetectionBase Biometrics.FingerSegmentationBase Biometrics.FingerDetectionMedium Biometrics.FingerExtractionMedium Biometrics.FingerDetectionFast Biometrics.FingerExtractionFast
Fingerprint Matcher	Biometrics.FingerMatching Biometrics.MatchingFusion Cluster.ClusterNode
Fingerprint Fast Matcher	Biometrics.FingerMatching Biometrics.MatchingFusion Cluster.ClusterNode Biometrics.FingerMatchingFast
Fingerprint Capturer	Media Devices.FingerScanners Biometrics.FingerDetectionBase Biometrics.FingerDetection
Fingerprint Fast Segmenter	Biometrics.FingerSegmentsDetectionBase Biometrics.FingerSegmentationBase Biometrics.FingerSegmentsDetection Biometrics.FingerSegmentation Biometrics.FingerSegmentsDetectionFast Biometrics.FingerSegmentationFast

Face Client	Media SmartCards Images.IHead Images.JPEG2000 Images.LosslessJPEG Biometrics.Standards.Base Biometrics.Standards.Other BioAPI.Base Devices.Cameras Biometrics.FaceDetectionBase Biometrics.FaceDetection Biometrics.FaceExtractionBase Biometrics.FaceExtraction Biometrics.FaceSegmentsDetectionBase Biometrics.FaceSegmentationBase Biometrics.Standards.Faces Biometrics.FaceDetectionMedium Biometrics.FaceExtractionMedium Biometrics.FaceSegmentsDetection Biometrics.FaceSegmentation Biometrics.FaceQualityAssessmentBase Biometrics.FaceQualityAssessment BioAPI.Faces
Face Extractor	Media SmartCards Devices.Cameras Biometrics.FaceDetectionBase Biometrics.FaceDetection Biometrics.FaceExtractionBase Biometrics.FaceExtraction Biometrics.FaceSegmentsDetectionBase Biometrics.FaceSegmentationBase
Face Fast Extractor	Media SmartCards Devices.Cameras Biometrics.FaceDetectionBase Biometrics.FaceDetection Biometrics.FaceExtractionBase Biometrics.FaceExtraction Biometrics.FaceSegmentsDetectionBase "Biometrics.FaceSegmentationBase Biometrics.FaceDetectionMedium Biometrics.FaceExtractionMedium Biometrics.FaceDetectionFast Biometrics.FaceExtractionFast
Face Matcher	Biometrics.MatchingFusion Cluster.ClusterNode Biometrics.FaceMatching

Face Fast Matcher	Biometrics.MatchingFusion Cluster.ClusterNode Biometrics.FaceMatching Biometrics.FaceMatchingFast
Face Capturer	Media Devices.Cameras Biometrics.FaceDetectionBase Biometrics.FaceDetection
Face Fast Token Image	Biometrics.FaceSegmentsDetectionBase Biometrics.FaceSegmentationBase Biometrics.FaceSegmentsDetection Biometrics.FaceSegmentation Biometrics.FaceQualityAssessmentBase Biometrics.FaceQualityAssessment Biometrics.FaceSegmentsDetectionFast Biometrics.FaceSegmentationFast Biometrics.FaceQualityAssessmentFast
Iris Client	Media SmartCards Images.WSQ Images.IHead Images.JPEG2000 Images.LosslessJPEG Biometrics.Standards.Base Biometrics.Standards.Other BioAPI.Base Devices.IrisScanners Biometrics.IrisDetectionBase Biometrics.IrisDetection Biometrics.IrisExtractionBase Biometrics.IrisExtraction Biometrics.IrisSegmentsDetectionBase Biometrics.IrisSegmentationBase Biometrics.Standards.Irises Biometrics.IrisDetectionMedium Biometrics.IrisExtractionMedium Biometrics.IrisSegmentsDetection Biometrics.IrisSegmentation Biometrics.IrisQualityAssessmentBase Biometrics.IrisQualityAssessment BioAPI.Irises
Iris Extractor	Media SmartCards Devices.IrisScanners Biometrics.IrisDetectionBase Biometrics.IrisDetection Biometrics.IrisExtractionBase Biometrics.IrisExtraction Biometrics.IrisSegmentsDetectionBase Biometrics.IrisSegmentationBase



Iris Fast Extractor	Media SmartCards Devices.IrisScanners Biometrics.IrisDetectionBase Biometrics.IrisDetection Biometrics.IrisExtractionBase Biometrics.IrisExtraction Biometrics.IrisSegmentsDetectionBase Biometrics.IrisSegmentationBase Biometrics.IrisDetectionMedium Biometrics.IrisExtractionMedium Biometrics.IrisDetectionFast Biometrics.IrisExtractionFast
Iris Matcher	Biometrics.MatchingFusion Cluster.ClusterNode Biometrics.IrisMatching
Iris Fast Matcher	Biometrics.MatchingFusion Cluster.ClusterNode Biometrics.IrisMatching Biometrics.IrisMatchingFast
Iris Capturer	Media Devices.IrisScanners Biometrics.IrisDetectionBase Biometrics.IrisDetection
Palmprint Client	Media SmartCards Images.Processing.FFT Images.Wsq Images.IHead Images.JPEG2000 Images.LosslessJPEG Biometrics.Standards.Base Biometrics.Standards.Other BioAPI.Base Devices.PalmScanners Biometrics.PalmDetectionBase Biometrics.PalmDetection Biometrics.Standards.Palms Biometrics.Standards.PalmTemplates" Biometrics.PalmDetectionMedium Biometrics.PalmExtractionBase Biometrics.PalmExtraction Biometrics.PalmExtractionMedium Biometrics.PalmSegmentsDetectionBase Biometrics.PalmSegmentsDetection Biometrics.PalmSegmentationBase Biometrics.PalmSegmentation Biometrics.PalmQualityAssessmentBase Biometrics.PalmQualityAssessment BioAPI.Palms

Palmprint Matcher	Biometrics.MatchingFusion Cluster.ClusterNode Biometrics.PalmMatching
Palmprint Capturer	Media Devices.PalmScanners Biometrics.PalmDetectionBase Biometrics.PalmDetection
Voice Client	Media SmartCards Biometrics.Standards.Base Biometrics.Standards.Other BioAPI.Base Devices.Microphones Biometrics.VoiceDetectionBase Biometrics.VoiceDetection Biometrics.VoiceExtractionBase Biometrics.VoiceExtraction Biometrics.VoiceSegmentsDetectionBase Biometrics.VoiceSegmentationBase Biometrics.VoiceDetectionMedium Biometrics.VoiceExtractionMedium Biometrics.VoiceSegmentsDetection Biometrics.VoiceSegmentation Biometrics.VoiceQualityAssessmentBase Biometrics.VoiceQualityAssessment BioAPI.Voices
Voice Extractor	Media SmartCards Devices.Microphones Biometrics.VoiceDetectionBase Biometrics.VoiceDetection Biometrics.VoiceExtractionBase Biometrics.VoiceExtraction Biometrics.VoiceSegmentsDetectionBase Biometrics.VoiceSegmentationBase
Voice Fast Extractor	Media SmartCards Devices.Microphones Biometrics.VoiceDetectionBase Biometrics.VoiceDetection Biometrics.VoiceExtractionBase Biometrics.VoiceExtraction Biometrics.VoiceSegmentsDetectionBase Biometrics.VoiceSegmentationBase Biometrics.VoiceDetectionMedium Biometrics.VoiceExtractionMedium Biometrics.VoiceDetectionFast Biometrics.VoiceExtractionFast

Voice Matcher	Biometrics.MatchingFusion Cluster.ClusterNode Biometrics.VoiceMatching
Voice Capturer	Media Devices.Microphones Biometrics.VoiceDetectionBase Biometrics.VoiceDetection
Cluster	Cluster.ClusterServer

## 4.7.3 MegaMatcher On Card SDK

**MegaMatcher On Card SDK** license names and licensed API components:

License name	API Component (functionality)
Fingerprint Card Extractor	Media SmartCards Devices.FingerScanners Biometrics.Standards.FingerCardTemplates Biometrics.FingerDetectionBase Biometrics.FingerExtractionBase .FingerSegmentsDetectionBase Biometrics.FingerSegmentationBase Biometrics.FingerQualityAssessmentBase
Face Card Extractor	Media SmartCards Devices.Cameras Biometrics.FaceDetectionBase Biometrics.FaceExtractionBase Biometrics.FaceSegmentsDetectionBase Biometrics.FaceSegmentationBase
Iris Card Extractor	Media SmartCards Devices.IrisScanners Biometrics.IrisDetectionBase Biometrics.IrisExtractionBase Biometrics.IrisSegmentsDetectionBase Biometrics.IrisSegmentationBase

## 4.7.4 VeriFinger Standard SDK

**VeriFinger Standard SDK** license names and licensed API components:

License name	API Component (functionality)
Fingerprint Extractor	Media Devices.FingerScanners Biometrics.FingerDetectionBase Biometrics.FingerDetection SmartCards Images.Processing.FFT Biometrics.FingerExtractionBase Biometrics.FingerExtraction Biometrics.FingerSegmentsDetectionBase Biometrics.FingerSegmentationBase
Fingerprint Matcher	Biometrics.FingerMatching Biometrics.MatchingFusion Cluster.ClusterNode

## 4.7.5 VeriFinger Extended SDK

**VeriFinger Extended SDK** license names and licensed API components:

License name	API Component (functionality)
Fingerprint Client	Media Devices.FingerScanners Biometrics.FingerDetectionBase Biometrics.FingerDetection SmartCards Images.Processing.FFT Biometrics.FingerExtractionBase Biometrics.FingerExtraction Biometrics.FingerSegmentsDetectionBase Biometrics.FingerSegmentationBase Images.WSQ Images.IHead Images.JPEG2000 Images.LosslessJPEG Biometrics.Standards.Base Biometrics.Standards.FingerTemplates Biometrics.Standards.FingerCardTemplates Biometrics.Standards.Fingers Biometrics.Standards.Other Biometrics.FingerDetectionMedium Biometrics.FingerExtractionMedium Biometrics.FingerSegmentsDetection Biometrics.FingerSegmentation Biometrics.FingerQualityAssessmentBase Biometrics.FingerQualityAssessment BioAPI.Base BioAPI.Fingers
Fingerprint Extractor	Media Devices.FingerScanners Biometrics.FingerDetectionBase Biometrics.FingerDetection SmartCards Images.Processing.FFT Biometrics.FingerExtractionBase Biometrics.FingerExtraction Biometrics.FingerSegmentsDetectionBase Biometrics.FingerSegmentationBase
Fingerprint Matcher	Biometrics.FingerMatching Biometrics.MatchingFusion Cluster.ClusterNode
Fingerprint Capturer	Media Devices.FingerScanners Biometrics.FingerDetectionBase Biometrics.FingerDetection

## 4.7.6 VeriLook Standard SDK

**VeriLook Standard SDK** license names and licensed API components:

License name	API Component (functionality)
Face Extractor	Media SmartCards Devices.Cameras Biometrics.FaceDetectionBase Biometrics.FaceDetection Biometrics.FaceExtractionBase Biometrics.FaceExtraction Biometrics.FaceSegmentsDetectionBase Biometrics.FaceSegmentationBase
Face Matcher	Biometrics.MatchingFusion Cluster.ClusterNode Biometrics.FaceMatching

## 4.7.7 VeriLook Extended SDK

**VeriLook Extended SDK** license names and licensed API components:

License name	API Component (functionality)
Face Client	Media SmartCards Images.IHead Images.JPEG2000 Images.LosslessJPEG Biometrics.Standards.Base Biometrics.Standards.Other BioAPI.Base Devices.Cameras Biometrics.FaceDetectionBase Biometrics.FaceDetection Biometrics.FaceExtractionBase Biometrics.FaceExtraction Biometrics.FaceSegmentsDetectionBase Biometrics.FaceSegmentationBase Biometrics.Standards.Faces Biometrics.FaceDetectionMedium Biometrics.FaceExtractionMedium Biometrics.FaceSegmentsDetection Biometrics.FaceSegmentation Biometrics.FaceQualityAssessmentBase Biometrics.FaceQualityAssessment BioAPI.Faces
Face Extractor	Media SmartCards Devices.Cameras Biometrics.FaceDetectionBase Biometrics.FaceDetection Biometrics.FaceExtractionBase Biometrics.FaceExtraction Biometrics.FaceSegmentsDetectionBase Biometrics.FaceSegmentationBase
Face Matcher	Biometrics.MatchingFusion Cluster.ClusterNode Biometrics.FaceMatching
Face Capturer	Media Devices.FingerScanners Biometrics.FingerDetectionBase Biometrics.FingerDetection

## 4.7.8 VeriEye Standard SDK

VeriEye Standard SDK license names and licensed API components:

License name	API Component (functionality)
Iris Extractor	Media SmartCards Devices.IrisScanners Biometrics.IrisDetectionBase Biometrics.IrisDetection Biometrics.IrisExtractionBase Biometrics.IrisExtraction Biometrics.IrisSegmentsDetectionBase Biometrics.IrisSegmentationBase
Iris Matcher	Biometrics.MatchingFusion Cluster.ClusterNode Biometrics.IrisMatching

## 4.7.9 VeriEye Extended SDK

VeriEye Extended SDK license names and licensed API components:

License name	API Component (functionality)
Iris Client	Media SmartCards Images.WSQ Images.IHead Images.JPEG2000 Images.LosslessJPEG Biometrics.Standards.Base Biometrics.Standards.Other BioAPI.Base Devices.IrisScanners Biometrics.IrisDetectionBase Biometrics.IrisDetection Biometrics.IrisExtractionBase Biometrics.IrisExtraction Biometrics.IrisSegmentsDetectionBase Biometrics.IrisSegmentationBase Biometrics.Standards.Irises Biometrics.IrisDetectionMedium Biometrics.IrisExtractionMedium Biometrics.IrisSegmentsDetection Biometrics.IrisSegmentation Biometrics.IrisQualityAssessmentBase Biometrics.IrisQualityAssessment BioAPI.Irises



Iris Extractor	Media SmartCards Devices.IrisScanners Biometrics.IrisDetectionBase Biometrics.IrisDetection Biometrics.IrisExtractionBase Biometrics.IrisExtraction Biometrics.IrisSegmentsDetectionBase Biometrics.IrisSegmentationBase
Iris Matcher	Biometrics.MatchingFusion Cluster.ClusterNode Biometrics.IrisMatching

## 4.7.10 VeriSpeak Standard SDK

**VeriSpeak Standard SDK** license names and licensed API components:

License name	API Component (functionality)
Voice Extractor	Media SmartCards Devices.Microphones Biometrics.VoiceDetectionBase Biometrics.VoiceDetection Biometrics.VoiceExtractionBase Biometrics.VoiceExtraction Biometrics.VoiceSegmentsDetectionBase Biometrics.VoiceSegmentationBase
Voice Matcher	Biometrics.MatchingFusion Cluster.ClusterNode Biometrics.VoiceMatching

## 4.7.11 VeriSpeak Extended SDK

**VeriSpeak Extended SDK** license names and licensed API components:

License name	API Component (functionality)
Voice Client	Media SmartCards Biometrics.Standards.Base Biometrics.Standards.Other BioAPI.Base Devices.Microphones Biometrics.VoiceDetectionBase Biometrics.VoiceDetection Biometrics.VoiceExtractionBase Biometrics.VoiceExtraction Biometrics.VoiceSegmentsDetectionBase Biometrics.VoiceSegmentationBase Biometrics.VoiceDetectionMedium Biometrics.VoiceExtractionMedium Biometrics.VoiceSegmentsDetection Biometrics.VoiceSegmentation Biometrics.VoiceQualityAssessmentBase Biometrics.VoiceQualityAssessment BioAPI.Voices
Voice Extractor	Media SmartCards Devices.Microphones Biometrics.VoiceDetectionBase Biometrics.VoiceDetection Biometrics.VoiceExtractionBase Biometrics.VoiceExtraction Biometrics.VoiceSegmentsDetectionBase Biometrics.VoiceSegmentationBase
Voice Matcher	Biometrics.MatchingFusion Cluster.ClusterNode Biometrics.VoiceMatching
Voice Capturer	Media Devices.Microphones Biometrics.VoiceDetectionBase Biometrics.VoiceDetection

## 4.7.12 SentiVeillance SDK

**SentiVeillance SDK** license names and licensed API components:

License name	API Component (functionality)
SentiVeillance	Devices.Cameras Surveillance Biometrics.FaceDetectionBase Biometrics.FaceDetection Biometrics.FaceExtractionBase Biometrics.FaceExtraction Biometrics.FaceSegmentsDetection Biometrics.FaceMatching Devices.Cameras

## 4.7.13 SentiSight SDK

SentiSight SDK license names and licensed API components:

License name	API Component (functionality)
SentiSight	Media Devices.Cameras SentiSight

## 4.7.14 SentiSight Embedded SDK

SentiSight Embedded SDK license names and licensed API components:

License name	API Component (functionality)
SentiSight	Media Devices.Cameras SentiSight

## 4.7.15 SentiGaze SDK

SentiGaze SDK license names and licensed API components:

License name	API Component (functionality)
SentiGaze	Media Devices.Cameras SentiGaze

## 4.7.16 SentiSculpt SDK

**SentiSculpt SDK** license names and licensed API components:

License name	API Component (functionality)
SentiSculpt	Media Devices.Cameras SentiSculpt

# 5 Troubleshooting

If you encounter problems while installing or using the SDK, please contact Neurotechnology Support Department [support@neurotechnology.com](mailto:support@neurotechnology.com) or your local distributor. It is desirable to send us additional information about your PC configuration and installation details.

On **Windows**, the PC information should be obtained through Activation Wizard:

1. Switch to the "Diagnostic" window in the Activation Wizard.
2. Copy the contents of the window to the clipboard or save the text to a file.
3. Paste the information to the email message or attach the file with the saved text.

On **Linux**, the PC information should be obtained in following way:

1. Run the `diagnostic_report.sh` script in `Bin/Linux_x86/Activation` or `Bin/Linux_x86_64/Activation` directory (use directory for your architecture). The script must be run as superuser (root).
2. Copy the output to the clipboard or save to a file.
3. Paste the information to the email message or attach the file with the saved text.