Neurotechnology Products Activation

User's guide

Version: 2025.1.0.0.

Release date: 3/4/2025.

Table of Contents

1 About

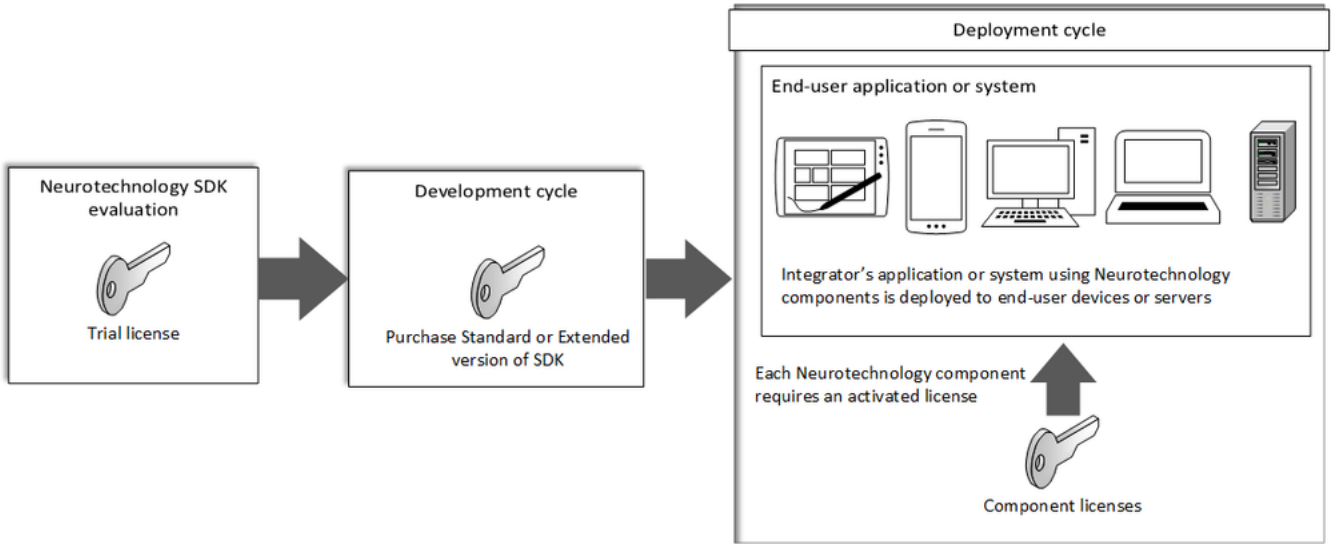
Neurotechnology provides the flexibility to utilize its products on a development device and integrate components into end-user applications or large-scale systems. Integrators can adapt the provided components and source code from samples and tutorials for redistribution as end-user applications upon license activation. Activation of your product is necessary to ensure the authenticity of the Neurotechnology product installation and to verify compliance with license agreements regarding device usage limits.

Neurotechnology products activation is mandatory for both trial versions and all purchased licenses, whether standard or extended. Activation does not involve the transmission of personal information to Neurotechnology.

Neurotechnology offers various licensing options:

- **Trial licensing**. Grants a 30-day free trial period without any obligations.
- **Development licensing**. An integrator should obtain one of Neurotechnology SDKs to develop a end-user product based on Neurotechnology technology. The SDK needs to be purchased just once and may be used for all projects and by all the developers within the integrator's company. Each SDK includes multiple licenses for each component. Integrators can obtain additional component licenses if more component licenses are required for the development process.
- **Deployment licensing**. Necessary when deploying Neurotechnology components in end-user applications, with separate licenses required for each deployment device.
- **VAR licensing**. A specialized licensing agreement between Neurotechnology and integrators interested in developing and selling Neurotechnology SDK-based development tools. Integrators who want to develop and sell a Neurotechnology-based development tool (with API, programming possibilities, programming samples, etc.), must obtain permission from Neurotechnology and sign a special VAR agreement.
- **Enterprise licensing**. Offers an individual licensing agreement allowing unlimited use of Neurotechnology components.
- **Disaster recovery licensing**. Disaster recovery licenses for server-side components 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.

Refer to the diagram below for a visualization of licensing throughout the application life cycle:



Notes

Intellectual Property Notices

By engaging with Neurotechnology products through purchase, trial, copying, or license activation, you implicitly agree to the terms of the license agreement stored in `\Documentation\SDK License.html`. All software, documentation, and materials within this product are copyrighted by NEUROTECHNOLOGY, © 1991–2025. ALL RIGHTS RESERVED.

Link List

- Pricing calculator - an online tool which helps to determine the cost of products and licenses.
- Product advisor - an online tool recommending Neurotechnology products based on your system requirements.
- Product schema - provides an overview of the component licenses included in the SDK.

Contacts

For assistance with Neurotechnology license activation issues, please contact us via email at support@neurotechnology.com.

To seek advice on Neurotechnology products or estimate the required licenses for your project, please email us at info@neurotechnology.com.

2 License types

Each specific SDK component possesses distinct functionalities and necessitates its own license. It's imperative to recognize that a license is indispensable for each individual computer or device running a component. For instance, if you're developing a fingerprint enrollment system slated for deployment across 500 computers, you'll require 500 fingerprint client licenses.

Traditionally, Neurotechnology products are activated via Single Computer Licenses. These licenses may be furnished as a serial number, an internet license file, or a dongle (a specialized hardware for license storage). Serial numbers are activated either through an internet connection or via email, and once activated, an internet connection is no longer necessary.

However, activation via serial number is unsuitable for ARM-Linux systems (excluding BeagleBone Black and Raspberry Pi 3 devices) or virtual environments. In such cases, as well as in mobile devices, internet licenses can be employed. Neurotechnology offers a license file that is stored on a computer or mobile/embedded device.

When internet activation isn't feasible for your project, or if a convenient license management solution is required, or if a virtual environment is employed, the license can be stored in a dongle. Dongles can also be utilized in distributing licenses across computers within the same network.

2.1 Serial numbers

Serial numbers allow to activate licenses for specific SDK components on a computer or device. Each serial number necessitates activation for its corresponding SDK component to function properly. Activation typically requires an internet connection. However, in cases where internet access is unavailable, activation can be initiated by sending an email. Following a successful activation, network connectivity is no longer necessary for this licensing type.

Neurotechnology offers a method to renew licenses in instances where computer alterations occur due to technical maintenance.

Activation via serial numbers involves a three-step process:

1. Neurotechnology generates serial numbers upon customer license purchases for various components.
2. Customers utilize the provided serial number to generate a hardware ID.
3. The generated ID is submitted to Neurotechnology, which then uses it to generate and provide a license file to the customer.

Serial number activation can be carried out manually by following these steps or automatically by skipping them. Details regarding available activation options are outlined in the section Activation options (page 8).

Notes

- Activation by serial number is not suitable for ARM-Linux, except BeagleBone Black, Raspberry Pi 3 and Nvidia Jetson 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 37).
- Generated hardware Id can be activated on Neurotechnology website.

See Also

Serial numbers activation for Android (page 31)

2.2 Dongle

Neurotechnology product licenses can be securely stored in a **dongle**, also known as a **volume license manager**, which acts as a hardware-based protection lock for purchased licenses. Storing licenses in a USB dongle enables activation without requiring an internet connection, making it particularly suitable for virtual environments.

The key advantages of using a dongle over serial numbers include:

1. **Storage for multiple licenses**. A single dongle can store numerous licenses for various Neurotechnology products.
2. **Licensing to local and remote devices**. It provides licenses to local devices as well as other devices over a TCP network.
3. **Remote update capability**. Dongles can be remotely updated, ensuring the latest license information is available.
4. **Offline license file generation**. License files can be generated offline directly from the dongle.
5. **Flexibility in license usage**. Licenses are not tied to a device's hardware permanently. When a device releases a license, it can be utilized by another device within the same network after a specified period (ranging from 15 minutes to 12 hours depending on the license type).

Dongles are utilized on-site by integrators or end-users to manage licenses for SDK components in the following ways:

- **Single computer license activation**. Activates an installation license for an SDK component on a specific device. The license quantity for the SDK component in the license manager decreases by the number of activated licenses.
- **Management of single computer licenses on a network**. The license manager enables the management of licenses for SDK components across networked computers. The number of managed licenses for an SDK component is limited by the number of licenses in the license manager. No further license activation is required, and the license quantity is not diminished. Once issued, the license is assigned to a specific computer on the network.
- **Using a license manager as a dongle**. A volume license manager containing at least one license for an SDK component can function as a dongle, allowing the execution of SDK component installations on particular computers.

Notes

If licenses stored in a dongle are intended for use from Docker or other containers, the License Activation Service (pgd) must be launched on the host machine. For alternative licensing options, the License Activation Service may run within Docker. These restrictions are specific to the License Activation Service and do not apply to other Neurotechnology libraries and services.

2.3 Internet licenses

A special type of license file, known as an **internet license**, can be stored on a computer or a mobile/embedded device. These licenses are continuously verified over the internet, requiring periodic internet connectivity for brief intervals.

Internet licenses offer several advantages:

- **No need for license activation**. Internet licenses do not require activation. Only a constant internet connection for short periods is necessary to check the license status.
- **Flexibility in license transfer**. Internet licenses can be transferred to another computer or device by moving the license file and waiting until the previous activation expires.
- **Ease of deployment**. Internet licenses are received as *.lic file(s) and can be placed at the root directory of the application. For example, in Neurotechnology sample applications, the files are typically placed in the \Bin\Licenses directory. For Android, they are located in sdcard/Neurotechnology/Licenses. When an application launches, it obtains internet licenses for its components.

The key differences of internet licenses compared to other license types include:

- **Hardware independence**. Internet licenses are not tied to specific hardware, allowing for changes in hardware or devices. However, the number of devices the license can be obtained from is limited by the license terms.
- **Internet connectivity requirement**. An internet connection is essential for license verification. The typical network traffic usage is minimal, with a short connection interval every 5-10 minutes, resulting in approximately 270kB/day data usage for two licenses.
- **Compatibility with virtual environments**. Internet licenses can be used in virtual environments.
- **Time limit for offline usage**. Internet licenses have a time limit for offline usage.
- **Server-side license file cache**. License files are cached on the server and remain effective even if the user deletes them.

Notes

Port 80 is used to check licenses over the Internet.

See Also

Internet licenses activation for Android (page 31)

2.4 Trial

All trial versions of Neurotechnology products come with a 30-day trial period. Once this period expires, you will no longer have access to the trial product.

When using trial products, the following requirements must be met:

- **Internet connection**. A constant internet connection is required to use the trial product. Without it, access to the trial product will be restricted.
- **Activation of trial version**. Trial versions can be activated through two methods: the Activation Wizard (only available for Windows) or manual activation (applicable to all platforms). Refer to the Activation options (page 8) section for detailed activation instructions.
- **Exclusive use of trial product**. If you opt to use one of Neurotechnology's trial products, you are prohibited from simultaneously using any licensed Neurotechnology products on the same computer. If multiple licensed products are running on a computer, activation services must be halted during the activation of trial products. This process is automatically managed during trial product activation.

Notes

If at least one internet license file is present, the SDK operates as a non-trial version.

3 Activation options

Neurotechnology licenses can be activated in these ways:

- Using Activation Wizard (page 9) - the activation application for Windows.
- Manual activation (page 23) for Windows, Linux, macOS
- Activation for Android (page 31)

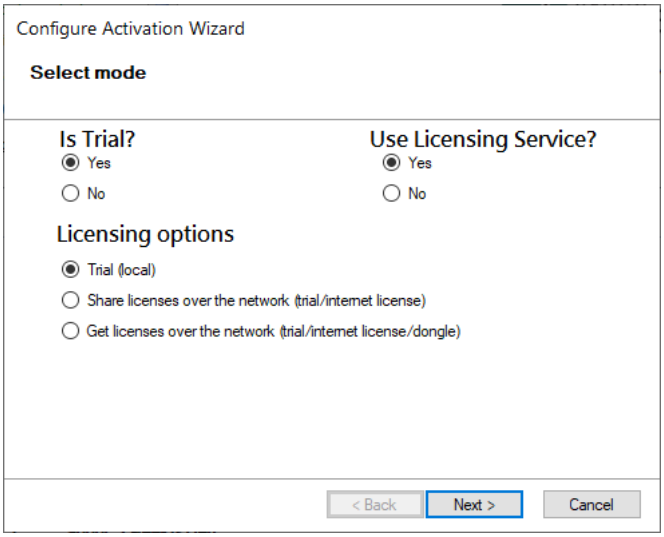
3.1 Activation wizard

The **Activation Wizard** is an utility included in the Neurotechnology SDK specifically designed for automatic product activation on Windows operating systems. It serves the purpose of activating purchased licenses or enabling trial versions of Neurotechnology SDK.

The Activation Wizard program, denoted as *ActivationWizard.exe*, can be initiated from the following location:

- `\Bin\Win64_x64\Activation\`

Upon launching the Activation Wizard, users are prompted to select an activation mode. Alternatively, the activation mode can be chosen from the main screen using the "*Configure*" button.



Let's review these activation modes:

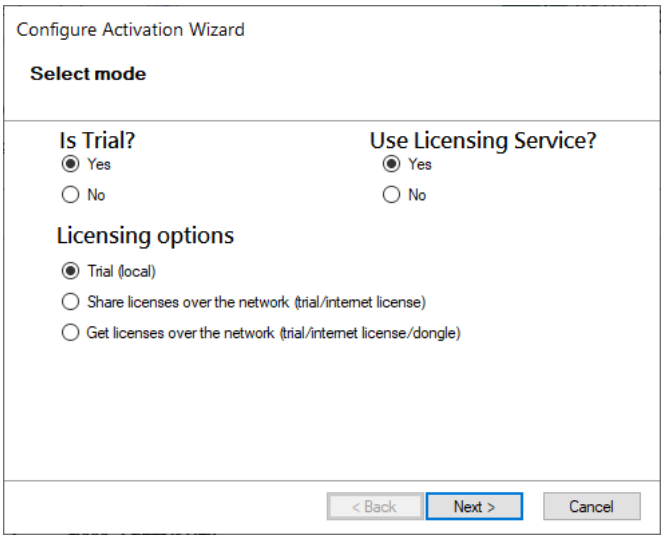
- Trial products activation (page 9)
- Purchased licenses activation (page 12)

3.1.1 Trial activation

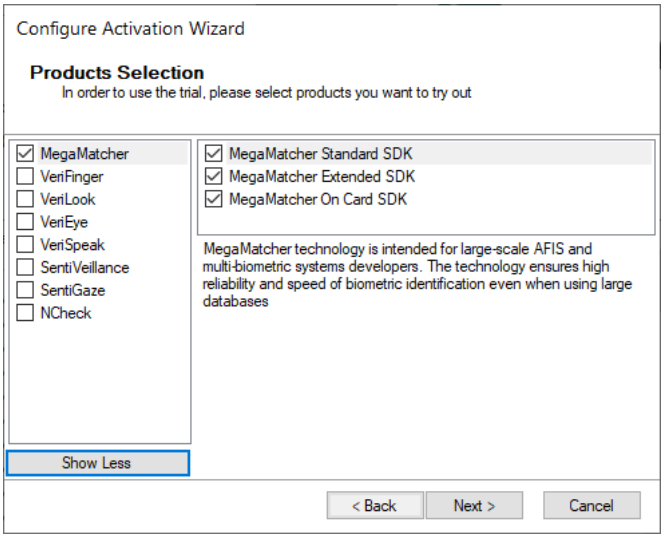
To activate Neurotechnology products for a trial period, follow these steps:

1. **Select *Trial* activation mode**. Launch the Activation Wizard and choose the Trial activation mode. Press the "Next" button to proceed.

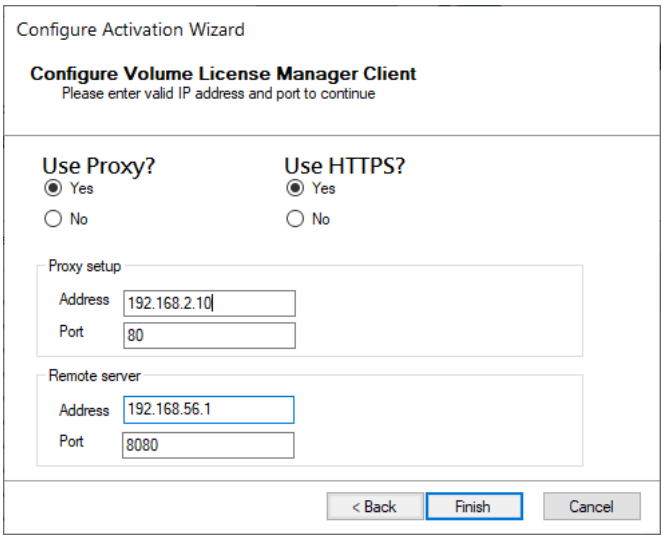
Additionally, you have the option to enable or disable the Neurotechnology licensing service. Enabling this service allows it to run in the background.



2. **Select trial products**. Choose the Neurotechnology products that you wish to trial. Once selected, the Activation Wizard will generate an *NLicensing.cfg* file in the same directory. Click "Next" to continue.



3. **Configure proxy server** (if necessary). The trial version of the SDK requires a constant internet connection to check licenses. If you are not directly connected to the internet and require a proxy, configure the proxy settings. Alternatively, if no proxy is needed, select the "Use Proxy -> No" option. If using a proxy server, enter the server's IP address and port. If HTTPS/SSL is needed, enable it by selecting "Use HTTPS - Yes". Click "Finish" when done.



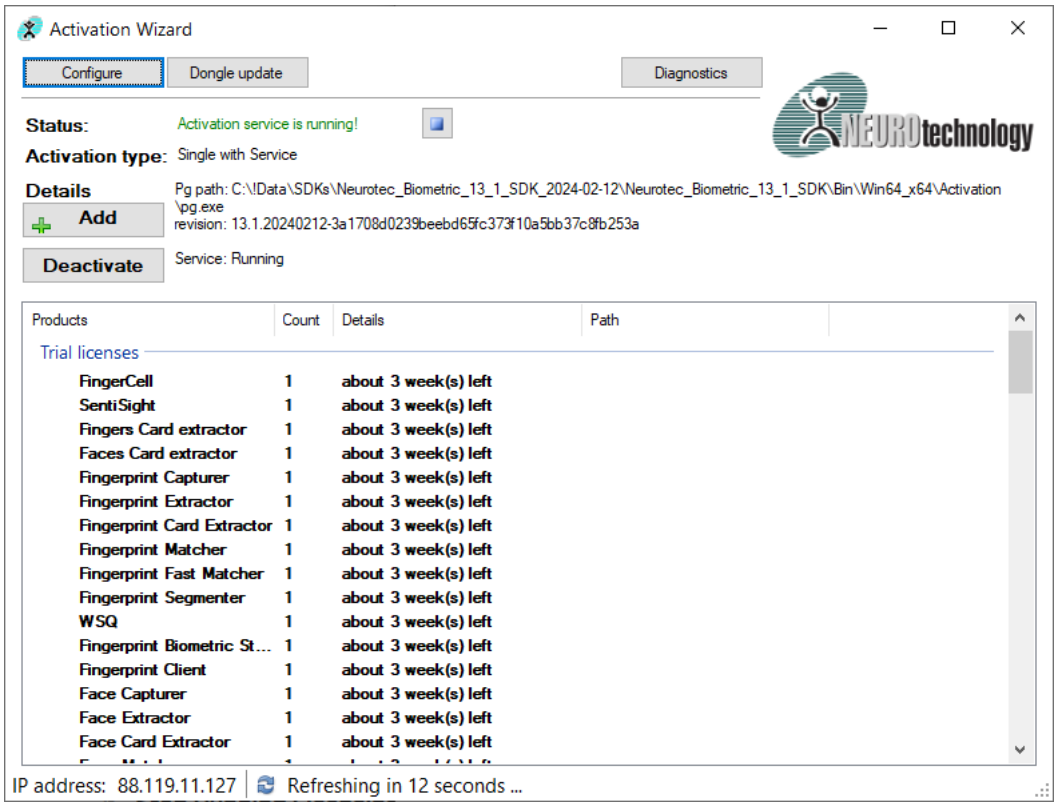
By default, the *No* (disabled) option is selected. This means that your computer is connected to the internet directly. If you use a proxy server for connecting the internet, enable proxy by entering these settings under the *Proxy setup*:

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

If more security is required, it is possible to use HTTPS/SSL (port 443). It can be enabled by selecting *Yes* under *Use HTTPS*?

Also, if remote licensing server is used, you should specify address and port of it.

4. **Review activation information**. Press the *Finish* button. Upon finishing the activation process, the Activation Wizard will display general information about the product trial, including the remaining trial period, local and external IP addresses, and licensing details.



5. **Configure trial product licenses**. If you need to add, remove, or change trial product licenses, click the "Configure" button. This will open a window where you can modify trial product selections. Note that selecting different trial product licenses will

replace the current ones.

6. **Stop running licensing services** (if necessary). If other Neurotechnology products are running on your computer, the Activation Wizard may prompt you to stop their licensing services. Choose whether to stop these services to activate and use the trial product. If you stop the licensing services for licensed products, you won't be able to use them until you start their respective licensing services again.

Once activation is complete, you'll have access to the fully functioning SDK for a period of 30 days.

Notes

If you need to use licensed products, press the "Configure" button and select the "Is Trial - No" option. More details are provided in the section Licenses activation (page 12).

See Also

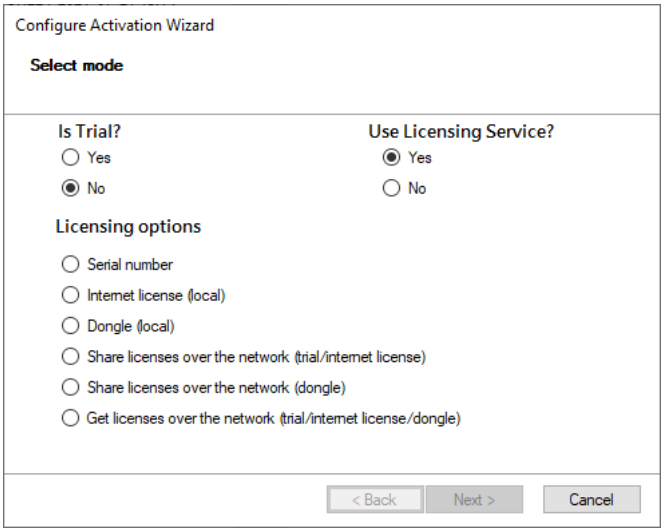
Trial (page 7)

3.1.2 Licenses activation

Neurotechnology components are protected against copying. To utilize them, you must purchase licenses and activate them. The available license activation options are as follows:

1. Serial Numbers (page 12)
2. Internet Licenses (page 15)
3. Dongle (page 17)

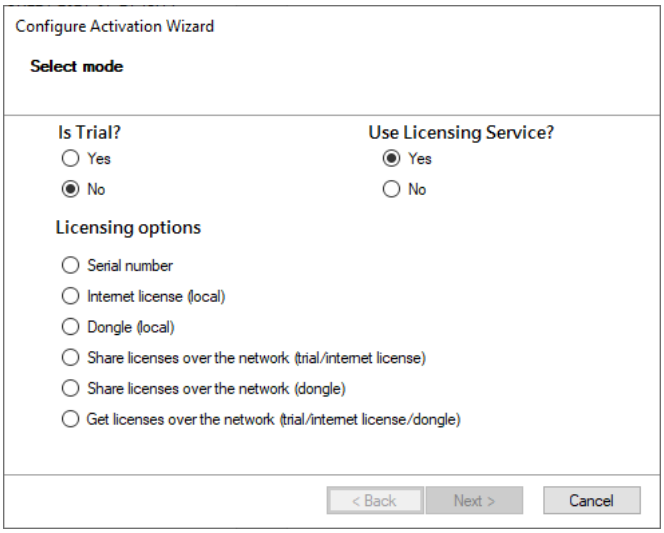
To begin the activation process, launch the *Activation Wizard* application and select the appropriate licensing option. Additionally, ensure that you select "*No*" for the "*Is Trial*" option when activating purchased licenses. This ensures that the activation is recognized as a purchased license rather than a trial version.



3.1.2.1 Serial Numbers activation

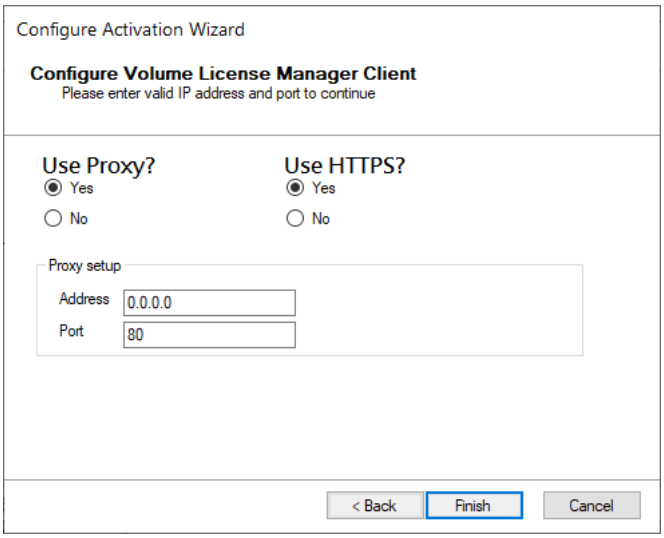
The procedure for activating serial numbers (single computer licenses) is as follows:

1. **Select serial number licensing option**. Choose the "Serial number" option from the Licensing menu.



2. **Choose licensing service**. Decide whether to use the Neurotechnology licensing service by selecting "Yes" or "No." If you opt for "Yes," the service will run in the background. Press "Next" to proceed.

3. **Configure proxy** (if necessary). If your computer is not directly connected to the internet and requires a proxy server, enable proxy settings. Otherwise, select "Proxy disabled." If using a proxy server, enter the server's IP address and port. If HTTPS/SSL is needed, enable it by selecting "Use HTTPS - Yes". Click "Finish" when done.



By default, the *No* (disabled) option is selected. This means that your computer is connected to the internet directly. If you use a proxy server for connecting the internet, enable proxy by entering these settings under the *Proxy setup*:

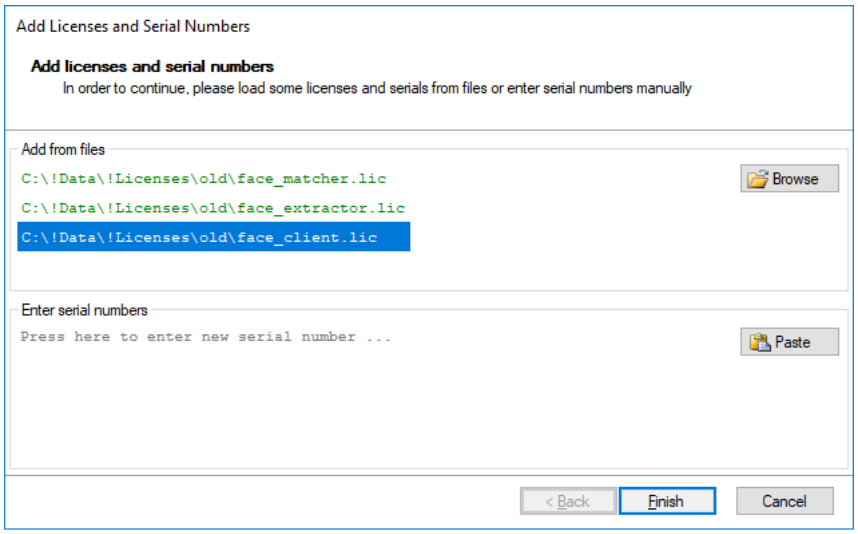
- **Address**. The IP address of your proxy server (e.g., http://192.168.2.10).
- **Port**. The number of a port for proxy server connections.

If more security is required, it is possible to use HTTPS/SSL (port 443). It can be enabled by selecting *Yes* under *Use HTTPS*?

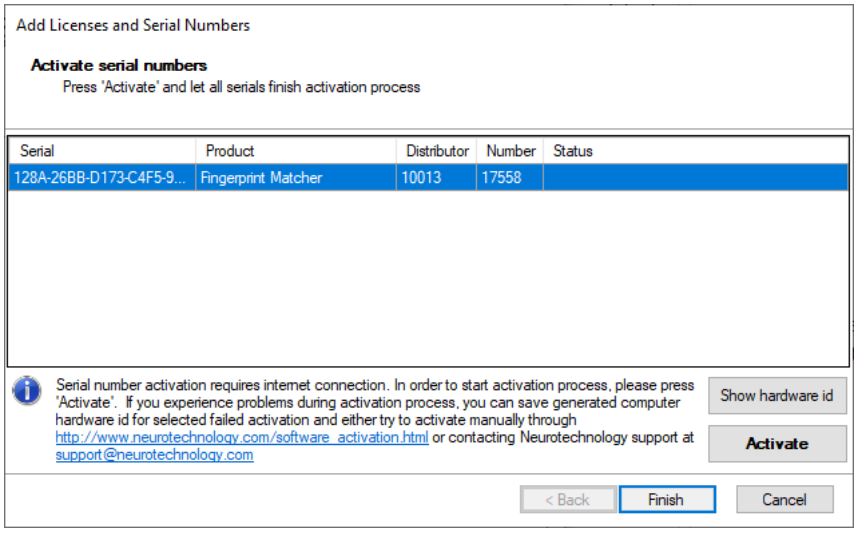
Press the *Finish* button.

4. **Add serial numbers**. From the main screen of the Activation Wizard, select "Add" Specify the path to the files where the serial numbers are saved, or enter them manually. Details of the serial numbers will be displayed.

5. **Review the provided information**. When serial number were added, press the "Finish" button:

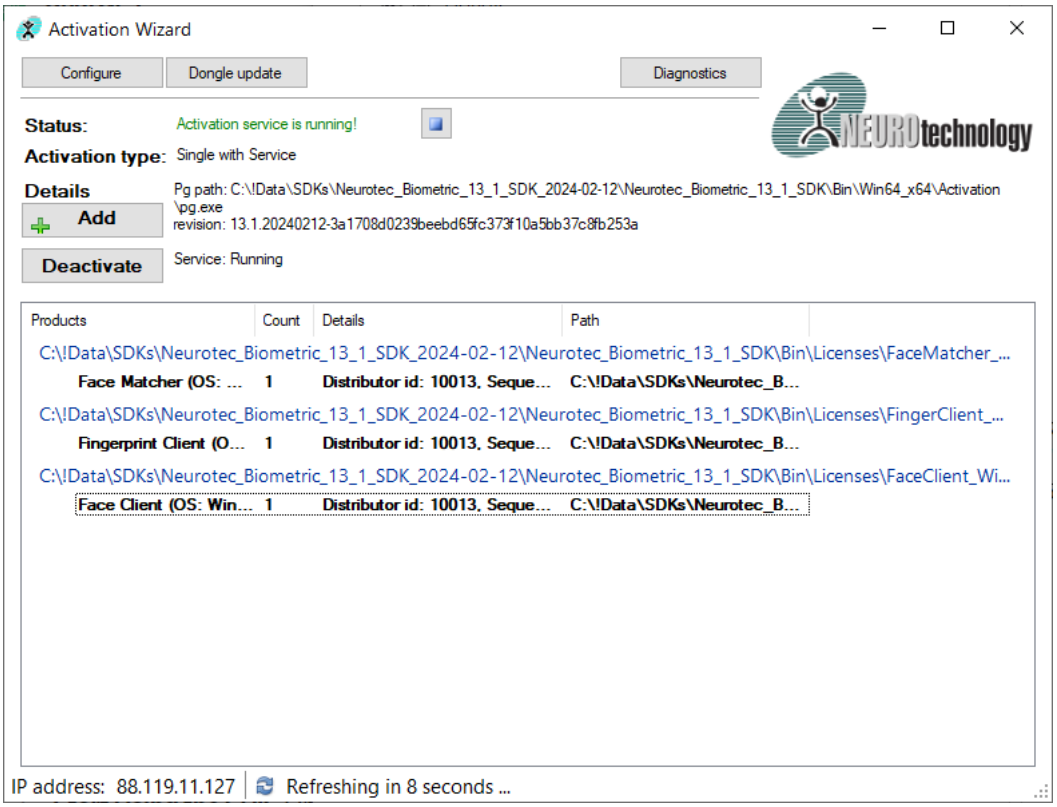


Serial number details are displayed:



Review the provided serial numbers and click "Activate." Upon successful activation, you'll see a confirmation message stating "Activated." Click "Finish" to complete the process.

5. **View activated licenses**. The main window of the Activation Wizard will display the details and count of activated licenses.



6. **Start using the SDK**. Once all licenses are activated, you can begin using the SDK. The activated SDK functionality will be visible in the main window of the Activation Wizard.

If you require additional functionality from the SDK, obtain additional licenses and activate them using the same procedure. For example, if you need to extract and match faces and irises, purchase and activate the corresponding face and iris extractor, matcher, or client licenses.

Notes

Activated licenses are copied to the *\Bin\Licenses* folder of the SDK. It's recommended to use the NLicense.Add() function/method from your application to manually set the license content.

See Also

Serial numbers (page 4)

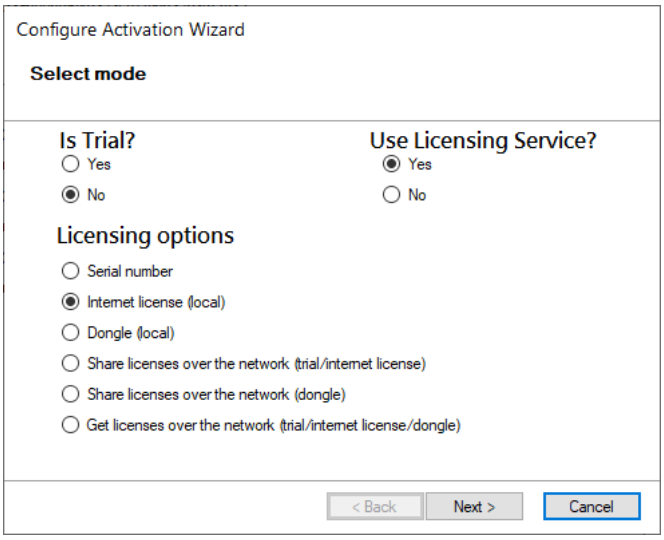
3.1.2.2 Internet Licenses activation

The procedure for activating internet licenses using the Activation Wizard is outlined below:

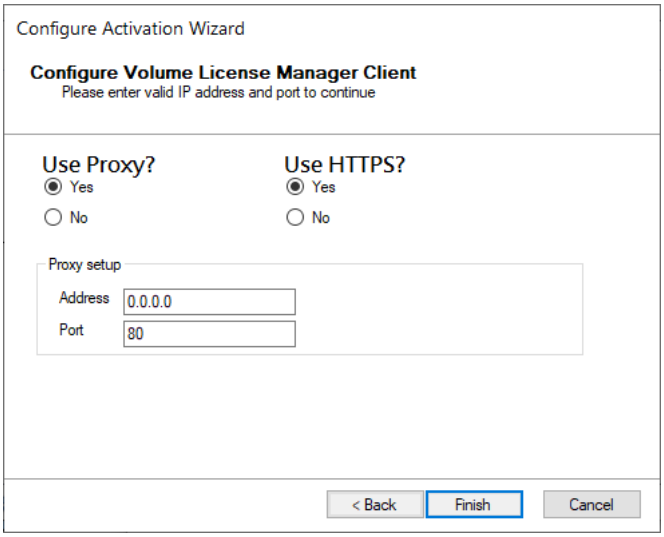
1. **Receive internet license files**. Obtain internet licenses in the form of *.lic file(s). Place these file(s) in the *\Bin\Licenses* directory of your Neurotechnology SDK installation.

2. **Select trial status**. Ensure that the "Is Trial?" option is set to "No" in the Activation Wizard.

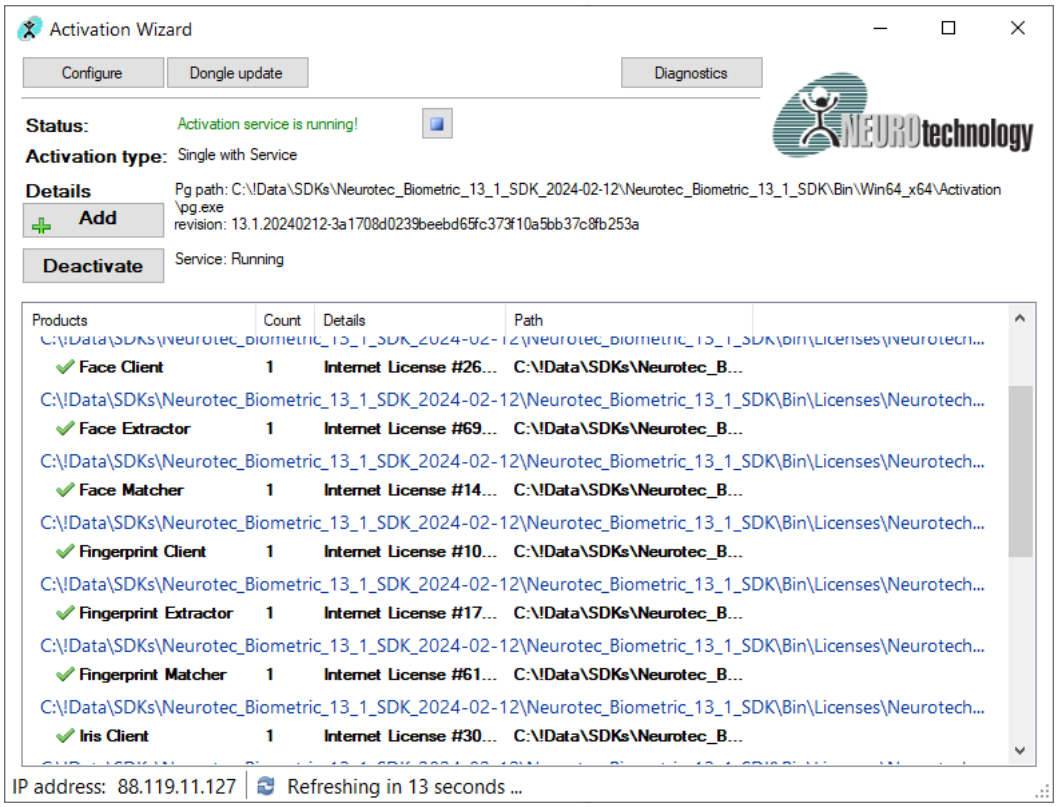
3. **Choose internet license** (Local). In the Activation Wizard, select "Internet license (local)" under Licensing options. Additionally, decide whether to use the licensing service by selecting the appropriate option. Press the "Next" button to proceed.



4. **Configure Volume license manager client**. If your computer is not directly connected to the internet and requires a proxy server, enable proxy settings. Otherwise, select "Proxy disabled." If using a proxy server, enter the server's IP address and port. If HTTPS/SSL is needed, enable it by selecting "Use HTTPS - Yes". Click "Finish" when done.



4. **Activation process**. The internet licenses will be activated in the background, and the activated licenses will be displayed on the main screen of the Activation Wizard.



Once these steps are completed, your internet licenses will be successfully activated and ready for use with the Neurotechnology SDK.

See Also

Internet licenses (page 6)

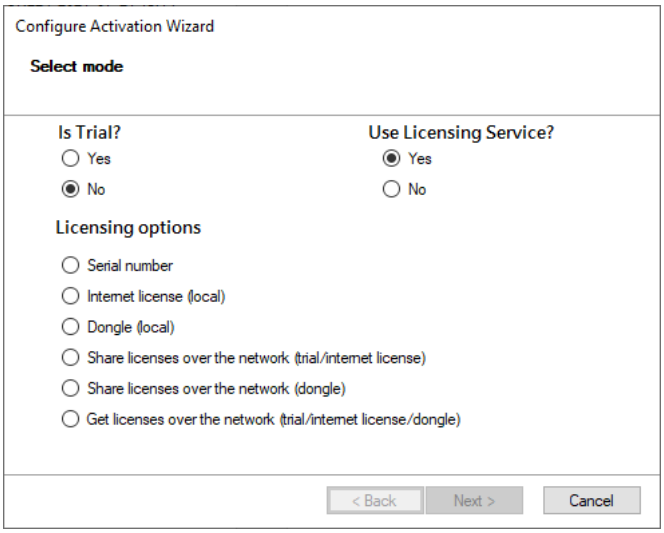
3.1.2.3 Dongle activation

Neurotechnology SDK licenses may be stored in a dongle (volume license manager), hardware-based protection lock storing purchased licenses. By following these steps, you can activate Neurotechnology SDK licenses using a dongle without requiring an internet connection, making it suitable for use in virtual environments or environments without internet access:

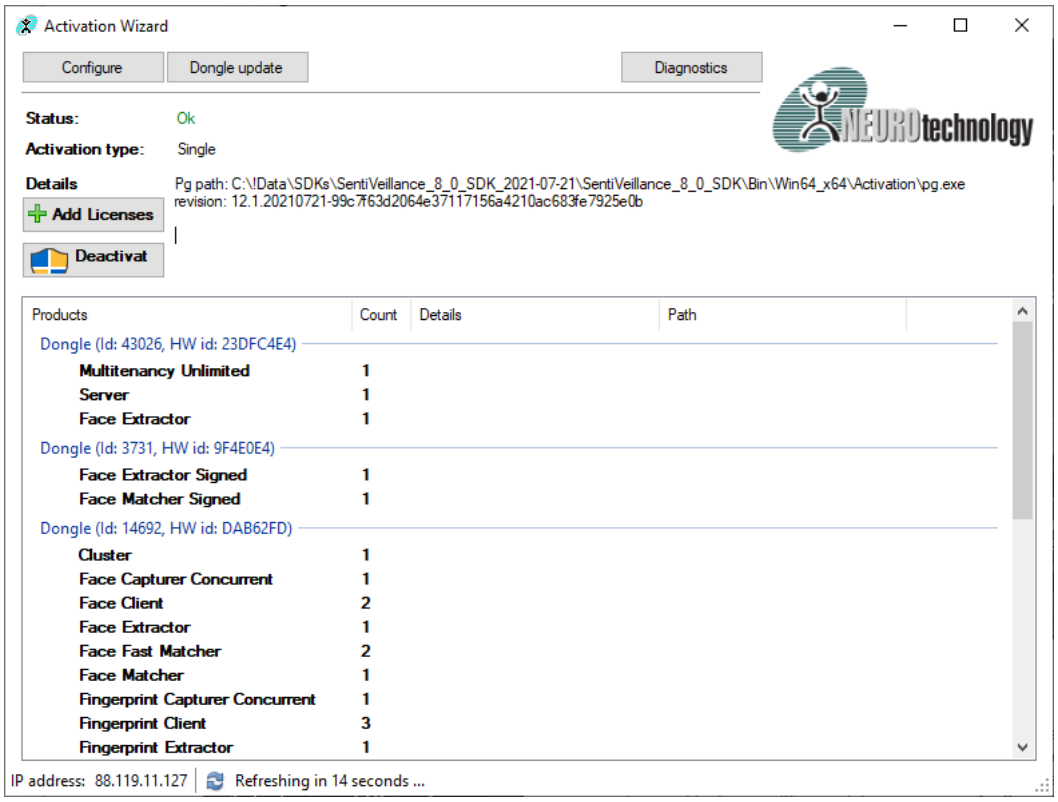
1. **Insert dongle**. Insert the Neurotechnology dongle into a USB port on your computer. Wait for the dongle drivers to be installed. Once the dongle is inserted and recognized by the system, you'll have access to all the licenses stored on it.

2. **Select licensing mode**. After inserting the dongle, you need to select the appropriate licensing mode in the Activation wizard. There are two options:

- **Dongle (local) mode**. When the dongle is connected directly to your PC, select "Dongle (local)" mode. This mode allows you to install and run the licensed component on a single PC or one Server CPU. Licenses will be activated for the computer in which the dongle was inserted.
- **Share licenses over network using dongle**. Alternatively, you can use dongle to share licenses over the network. This mode allows you to configure licensing service that enables sharing licenses over the network. Licenses stored on the dongle can be shared across multiple computers connected to the same network. This enables flexibility in licensing for environments where multiple users or devices need access to the licensed components.



3. **Check license availability**. After inserting the dongle, check the Products window to ensure that the licenses from the dongle are visible. If they appear in the Products window, it means that the licenses have been successfully activated for the computer, and no further steps are required.



4. **Integrating licensing into your application**. Once the licenses are available on your computer through the dongle, you can integrate them into your application. This involves calling the appropriate license within your application code to enable the licensed features provided by the Neurotechnology SDK. See Licenses obtain in your application (page 35).

See Also

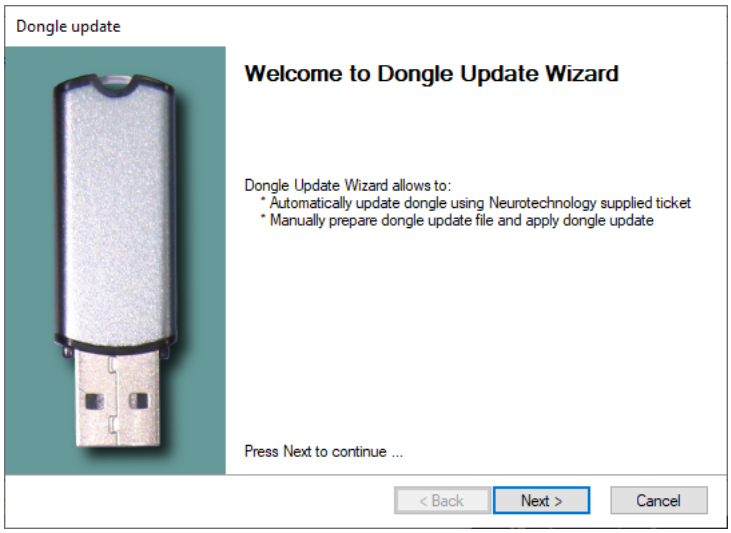
Dongle (page 5)

3.1.2.3.1 Dongle update

Activation wizard allows you to run *Dongle update*. This option allows adding new licenses to the dongle. Here's a step-by-step

guide for performing a Dongle update in the Activation wizard:

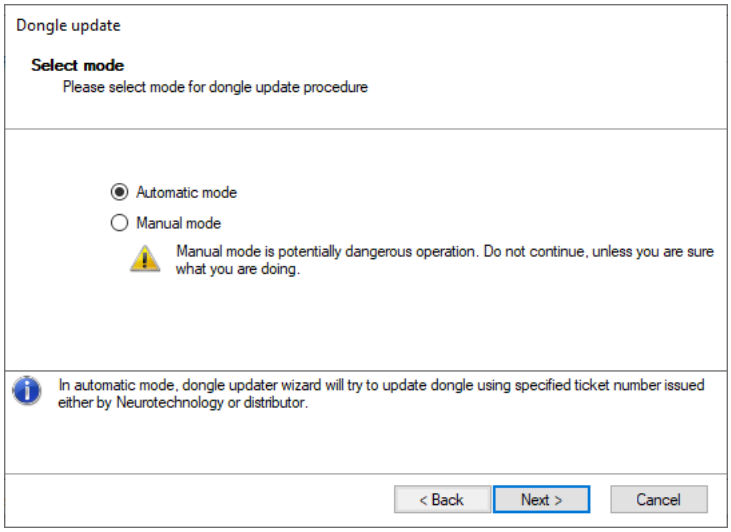
1. **Open Activation Wizard and click Dongle Update button**. Launch the Activation wizard and locate the Dongle update button on the main screen. Click on it to initiate the update process. This action will open the Dongle update wizard:



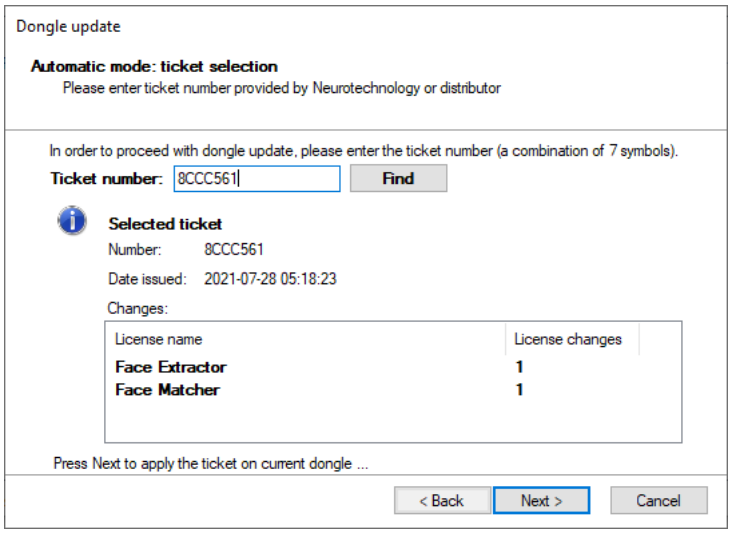
2. **Choose dongle update mode.** In the Dongle update wizard, you'll be presented with options for automatic or manual dongle update.

- **Automatic mode** (recommended). In automatic mode, the Dongle updater wizard will attempt to update the dongle using a specified ticket number issued by either Neurotechnology or a distributor.
- **Manual mode** (advanced mode). In manual mode, the wizard allows the user to manually prepare dongle dumps and apply dongle updates. This mode is considered an advanced feature and carries potential risks if not executed properly.

It's recommended to choose automatic update to ensure a smoother process. Manual update carries potential risks, so it's advised to consult with Neurotechnology support if you intend to use this method. Their expertise can help ensure the update process is carried out correctly and minimize the risk of errors or complications.



3. **Enter ticket number for additional licenses**. If you've purchased additional licenses for the dongle update, you should have received a ticket number. Enter this ticket number when prompted in the next window of the Dongle update wizard. Each ticket can only be used once. After entering the ticket number, click "Find" to proceed. The corresponding ticket data will be displayed, allowing you to add licenses to your dongle:



4. **Verify successful update**. If all steps were successful, you can test the installation by running a sample application that utilizes the licensed components. If any issues arise during the process, refer to the Troubleshooting (page 63) section for assistance.

3.1.2.3.2 Share dongle licenses over the network

Neurotechnology licensing service allows managing licenses for SDK components across multiple computers or mobile/embedded devices in a LAN or over the Internet. This licensing option is useful when a computer or device needs a license for a limited period to perform a specific task, after which the license can be reassigned to another device.

The number of managed licenses for an SDK component is restricted by the number of licenses stored in the dongle. This ensures that the number of concurrent users or devices using the licensed component does not exceed the available licenses.

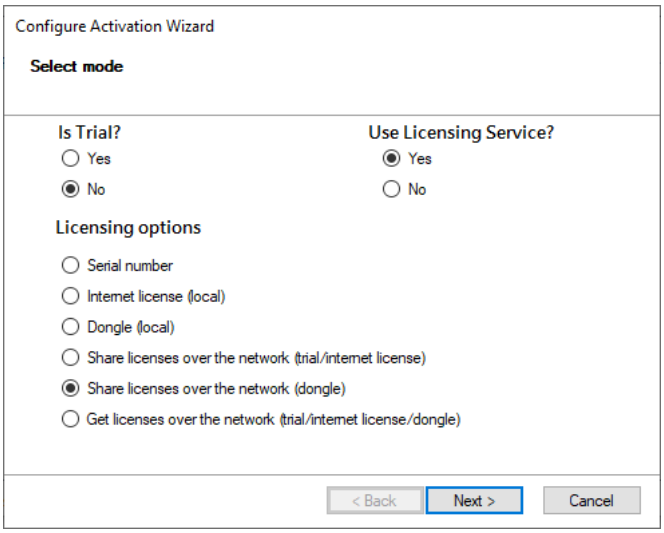
Unlike traditional licensing methods where activation is necessary on each device, with this approach, no license activation is required. Once issued, the license is assigned to a specific computer or device on the network. Once a license is issued, it is assigned to a particular computer or device within the network. This allows for efficient utilization of licenses across multiple devices, ensuring that licenses are not wasted or underutilized.

Activation wizard can be used to configure licensing service to share licenses from a dongle over the network. Here are the steps required to configure licensing service to share dongle licenses over the network:

1. **Select configuration**. On the main screen of the Activation Wizard, select "Configure."

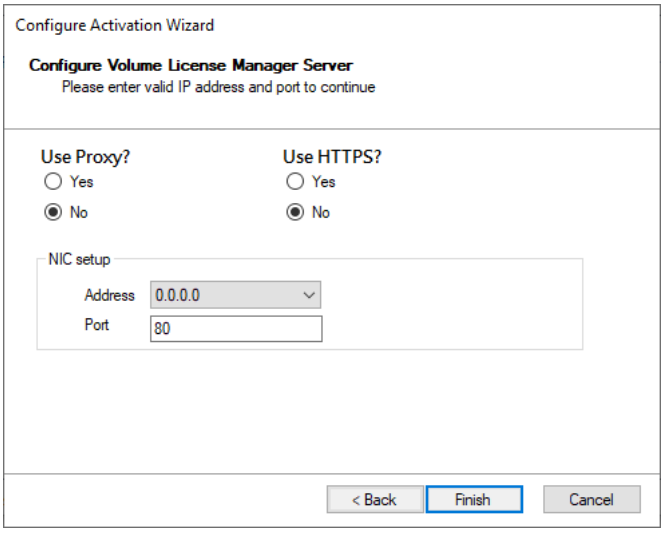
2. **Configure activation mode**:

- Set "Is Trial" to "No."
- Enable "Use Licensing Service."
- Enable "Share licenses over the network (dongle)".

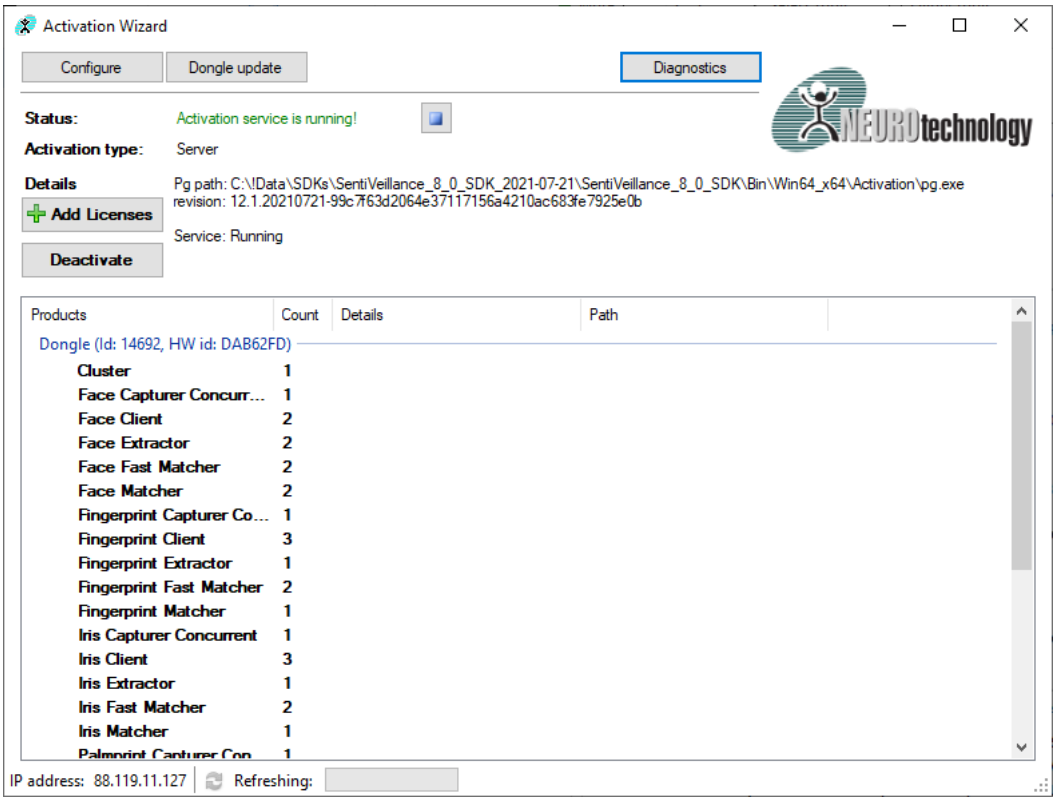


3. **Configure network interface**:

- Enter the IP address and port of the computer where the dongle is connected and the licensing server is running.
- If you enter "0.0.0.0" as the IP value, the server will use all available interfaces.
- Specify the port number to which the server should listen for client connections.



4. **Finish configuration**. After entering the necessary settings, click "Finish." The Activation Wizard will display the "Activation service is running" status on the main screen.



Notes

If there's no direct connection to the Internet, you'll need to configure the proxy settings in step 3. Select whether to use a Proxy server and specify whether HTTPS should be used.

3.2 Manual activation

When the Activation Wizard for Windows cannot be utilized or when using Linux and macOS operating systems, Neurotechnology products can still be activated manually.

Required steps for manual licenses activation:

1. **Locate activation folder**. Navigate to the Activation folder within the SDK directory. This folder is typically located at *Bin\[platform]\Activation*, where *[platform]* represents the specific operating system:

- *Linux*: \Bin\Linux_x86_64\Activation
- *Linux ARM*: Bin\Linux_arm64\Activation\
- *macOS*: Bin\macOS\Activation\
- *Windows*: Bin\Win64_x64\Activation\

2. **Edit configuration file** (*pgd.conf*). In the Activation folder, locate the configuration file named *pgd.conf*. This file contains settings and parameters related to activation.

3. **Customize configuration file**. Open the *pgd.conf* file using a text editor. Customize the settings within this file according to your activation requirements. This may include specifying activation mode, server addresses, and any other relevant parameters.

4. **Save configuration file**. After customizing the configuration file, save the changes.

5. **Start licensing service manually**. Start the licensing service manually.

3.2.1 Trial

To manually activate trial licenses for Neurotechnology products, follow these steps:

1. **Open pgd.conf configuration file**. Locate and open the pgd.conf file (saved in the "Activation" folder), which is the licensing configuration file for Neurotechnology products.

2. **Configure activation mode and trial settings**:

- **Mode**. Set the *Mode* parameter to specify the activation mode. Choose from four available modes: *NoPg*, *Single*, *Server*, or *Gateway*.
- **Trial**. If you want to enable a trial period for the SDK, set Trial to true.
- **Licensing server address and port**. Optionally, specify the *Address* and *Port* parameters if using a licensing server. These settings are required for network licensing and can be left blank if not applicable.

3. **Configure proxy settings** (if required). If you're using a proxy server, set the *TrialProxyServer*, *TrialProxyPort*, *TrialProxyUsername*, and *TrialProxyPassword* parameters accordingly.

4. **Enable or disable secure connection**. When required, set *https* parameter to *true*. This will enable secure HTTPS/SSL connection.

5. **Save configuration file**. Save the *pgd.conf* file with the licensing service settings in the *Bin\[platform]\Activation* directory within the SDK installation directory.

6. **Start Licensing Service**:

- *For Linux and macOS*: `sudo ./run_pgd.sh start`
- *For Windows*: `pg.exe -install`

7. (Optional) **Stop licensing service**. If you need to stop licensing service, run the following command:

- *For Linux and macOS*: `sudo ./run_pgd.sh stop`
- *For Windows*: `pg.exe -uninstall`

`pgd.conf` file configuration examples:

1. Local licenses usage (with licensing service running in the machine):

```
Mode = Single
Trial = true
https = false
```

2. Local licenses usage (without licensing service running in the machine):

```
Mode = NoPg
Trial = true
https = false
```

3. Sharing licenses over network when licensing server with proxy is used (Https/SSL enabled):

```
Address = 192.168.56.1
Port = 8080
Mode = Server
TrialProxyServer = 192.168.2.10
TrialProxyPort = 80
TrialProxyUsername =
TrialProxyPassword =
Trial = true
https = true
```

4. Get licenses over network (using proxy):

```
Address = 192.168.56.1
Port = 8080
Mode = Gateway
TrialProxyServer = 192.168.2.10
TrialProxyPort = 80
TrialProxyUsername =
TrialProxyPassword =
Trial = true
https = false
```


3.2.2 Serial numbers (Single computer licenses)

Serial numbers allow to activate licenses for specific SDK components on a computer or device. Each serial number necessitates activation for its corresponding SDK component to function properly. Activation typically requires an internet connection. However, in cases where internet access is unavailable, activation can be initiated by sending an email. Following a successful activation, network connectivity is no longer necessary for this licensing type.

Before license activation, you should edit configuration file (*pgd.conf*). Locate and open the *pgd.co*nf file (saved in the "Activation" folder), which is the licensing configuration file for Neurotechnology products.

Configure activation mode and licensing service settings as described in the previous sections. The main parameters of the licensing configuration file:

Parameter	Description
Mode	Set the *Mode* parameter to specify the licensing service and activation mode. Choose from four available modes: *NoPg*, *Single*, *Server*, or *Gateway*. • *Single* - the default licensing service mode for single computer licenses. The licensing service is running in the same machine where licensed components are executed. • *Server* - the licensing service is running as a server allowing sharing licenses over the network. • *Gateway* - the licensing service is running in a gateway mode allowing to get licenses over the network. • *NoPg* - no licensing service running 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 the `Licenses` folder. This folder should be placed in the root directory of your application. The preferred way is to call the `NLicense.Add()` function/method from your application and manually set each activated license file content.
LicenseFile	The license file path. The license file should be placed in the same directory as `pg.exe` or `pgd` or full path to a license file should be specified. If several license files are used, the path for each license file should be added.
LicenceUsageLogFile	Path to licenses usage log file.
Address	IP address of a proxy server. It is an advanced setting when you are using proxy server for activation.
Port	Port of a proxy server.

Below is a step-by-step manual activation procedure for both Linux/macOS and Windows platforms:

Linux and macOS Manual Activation Procedure

1. Generate computer Id file. Run the `id_gen` tool to generate the `Neurotechnology.id` file from one of these directories (depends on the platform you use):

- `\Bin\Linux_arm64\Activation\`
- `\Bin\Linux_armhf\Activation\`
- `\Bin\Linux_x86_64\Activation\`
- `\Bin\macOS\Activation\`

For example:

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

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

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

2a. Obtain license file from Neurotechnology website. Activate the generated `Neurotechnology.id` file online or send it to Neurotechnology or your distributor if online activation fails.

- **Online activation**. Online activation is performed by uploading Id file to the Neurotechnology online activation website:
  - http://www.neurotechnology.com/cgi-bin/nla.cgi
- **Offline activation**. Alternatively, Id file can be sent to Neurotechnology (support@neurotechnology.com) or distributor from which SDK was acquired. Please send it only when online activation fails.
- **Save the activated license file**. After a successful activation you will receive the activated license file (e.g., *FaceClient_Linux.lic*). The received license file should be saved in the `Activation` folder (see step 1).

- (Optional) **Set license file location**. If you need to save in another location, specify new location in the `pgd.conf` file using `LicenseFile` parameter. For example:

```
LicenseFile =  C:/Neurotechnology_SDK\Bin\Licenses\FaceClient_Linux.lic
```

Note: each license file should be specified separately.

2b. (Optional) **Run licenses activation script**. Use this method if the one described in step 2a does not fit your business rules.

Another method for activating a license is by using the ***license_activation*** tool. This command-line program is designed specifically for activating Neurotechnology products and can be found in the appropriate directory within the SDK installation:

- For Linux ARM64: `/Bin/Linux_arm64/Activation/`
- For Linux ARMHF: `/Bin/Linux_armhf/Activation/`
- For Linux x86_64: `/Bin/Linux_x86_64/Activation/`
- For macOS Universal: `/Bin/macOS/Activation/`

To utilize *license_activation*, follow these steps:

- Open a terminal or command prompt.
- Navigate to the directory containing the *license_activation* tool.
- Enter the following command in the command line:

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

Available options:

Option	Description
-h	This option prints usage information for the *license_activation* tool. It provides details on how to use the tool and its available options.
`-s serial-number`	Use this option to specify the serial number required for activation. It should be used together with the -o parameter, which specifies the output file name.
`-o output-file`	This option specifies the name of the output file where the generated license will be saved.
`-i`	This option generates a device ID. It may be used independently or in conjunction with other options, depending on the activation requirements.

Usage examples:

1. This command activates the license using the serial number provided in the *serial_number.txt* file and saves the generated license to *license.lic*:

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

2. 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
```

3. 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
```

4. 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_13_1_SDK\Bin\Licenses\FaceClient_Linux.lic
```

Windows manual activation procedure

1. Generate computer Id file

Run ***id_gen.exe*** with the following parameters:

```
Bin/Win64_x64/Activation/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 containing 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. Obtain license file from Neurotechnology website. Activate the generated *Neurotechnology.id* file online or send it to Neurotechnology or your distributor if online activation fails.

- **Online activation**. Online activation is performed by uploading Id file to the Neurotechnology online activation website:
- http://www.neurotechnology.com/cgi-bin/nla.cgi

- **Offline activation**. Alternatively, Id file can be sent to Neurotechnology (support@neurotechnology.com) or distributor from which SDK was acquired. Please send it only when online activation fails.
- **Save the activated license file**. After a successful activation you will receive the activated license file (e.g., *FaceClient_Windows.lic*). The received license file should be saved in the `Activation` folder (see step 1).
- (Optional) **Set license file location**. If you need to save in another location, specify new location in the *pgd.conf* file using *LicenseFile* parameter. For example:
  - `LicenseFile = C:/Neurotechnology_SDK\Bin\Licenses\FaceClient_Linux.lic`

Note: each license file should be specified separately.

3. Run Neurotechnology licensing service. After adding the configuration and license files to the appropriate folders, start the licensing service.

To install the licensing service on Windows, run the following command in the command line:

```
pg.exe –install
```

To stop the licensing service, use the following command:

```
pg.exe –uninstall
```

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

It is recommended to use Activation wizard (page 9) for Windows.

Notes

After completing the activation procedure outlined above, Neurotechnology license(s) will be activated, and the functionality of the SDK will be unlocked. However, it's important to note that a particular license may not unlock all functionalities. For example, if you have a face client license and need to perform fingerprint matching, you will need to obtain additional fingerprint licenses. Refer to the Product schema for detailed information on the functionalities covered by each license.

3.2.3 Volume license manager (dongle)

As outlined in the section on dongle activation for Windows (page 17), a volume license manager (dongle) serves as a repository for purchased licenses. The dongle offers the flexibility to manage and activate licenses without an internet connection. It can be employed in various scenarios for manual license management:

1. **Generating serial numbers for single computer activation**. The dongle facilitates the generation of serial numbers for activating licenses on individual computers, ensuring secure and reliable license allocation.

2. **Managing licenses on the same network**. In networked environments, the dongle enables centralized license management, allowing the same license to be shared among multiple devices on the network. This ensures efficient utilization of licenses across the organization.

3. **Setting up a server for license distribution or client license reception**. The dongle can be configured as a server to distribute licenses across the network or as a client to receive licenses from a centralized server. This setup streamlines the license distribution process, enhancing license management efficiency.

3.2.3.1 Generating licenses for single computers activation

Neurotechnology SDK customers have the option to independently generate single computer licenses using a USB dongle provided by Neurotechnology or a local distributor. In this scenario, customers are required to develop their own license management software to generate licenses from the dongle. This software utilizes the Licensing library included in the SDK, which offers functionality for managing licenses.

To assist developers in implementing this functionality, the SDK package includes tutorials for C/C#/Java/VB.NET, demonstrating how to generate serial numbers from a dongle. These tutorials can be found in the `Tutorials\Licensing` folder (*SerialNumberGenerationFromDongle*). Typically, developers will call the serial number generation function/method from NLicenseManager, which is part of the NLicensing library and facilitates license management tasks.

For example, in a C# application, the process may 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);
//...
```

The `NLicenseManager.GenerateSerial` method utilizes the provided product ID and user-assigned sequence number to generate the serial number, with the distributor ID returned as well. The product ID can be obtained by calling the NLicenseManager.GetProductIds() method.

Additionally, for Windows, serial numbers from a dongle can also be generated using the Activation wizard (page 17).

3.2.3.2 Managing licenses on network

The Neurotechnology License Manager allows the sharing of SDK licenses across computers within the same local area network (LAN). The number of shared licenses is limited by the licenses stored in the attached dongle.

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

Configuration for Volume License Manager Server (VLM)

To enable license sharing on the network, you'll need to set up the Neurotechnology licensing service as a **server** (Volume License Manager server) on one PC, while also configuring it as a **client** (Volume License Manager client) on all other devices that require access to the SDK licenses.

Here's how to configure volume license manager server:

1. Plug the dongle into the server computer.

2. Configure the *pgd.conf* file in the *Bin\[platform]\Activ*ation folder with the following parameters:

```
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>, change to your
own IP address
Port = 5000 # <port, to listen>
```

Note: Using 0.0.0.0 as the address allows the server to listen on all IP addresses used in network cards.

3. Save *pgd.conf* file in *Bin\[platform]\Activation* folder.

4. Start Neurotechnology licensing service:

- For Windows:

```
pg.exe -install
```

- For Linux and macOS:

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

5. Ensure that the Neurotechnology service is running. In Windows, navigate to "Start -> Control Panel -> Administrative Tools -> Services -> Neurotechnology" to verify the service status. In Linux and macOS, use `./run_pgd.sh log` command to check the service status.

Configuration for Volume License Manager Client

1. Copy the SDK distribution to each client computer.

2. Configure `pgd.conf` file for the client with the following parameters:

```
Mode = Gateway # "gateway" is mandatory here to run as LAN client (Dongle VLM Client)
Address = 192.168.1.1 # <IP address of network card for incoming connections>, change to
your own IP address
Port = 5000 # <licensing server port number>
```

3. Save `pgd.conf` file in `Bin\[platform]\Activation` folder.

4. Start the Neurotechnology licensing service using the same commands as for the VLM server:

- For Windows:

```
pg.exe -install
```

- For Linux and macOS:

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

5. Ensure that the Neurotechnology service is running. In Windows, navigate to "Start -> Control Panel -> Administrative Tools -> Services -> Neurotechnology" to verify the service status. In Linux and macOS, use ./run_pgd.sh log command to check the service status.

After configuring the server and client, applications running on client computers can obtain licenses for components. The server will share licenses across the network for the configured client computers.

Notes

When using this licensing option, each license can only be used in one computer at a time. When a license is no longer required on a client PC, it is returned to the dongle, although this process may take several hours. Therefore, if the number of client computers exceeds the number of licenses stored in the dongle, other computers will retrieve a license after some time.

3.3 Activation for Android

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

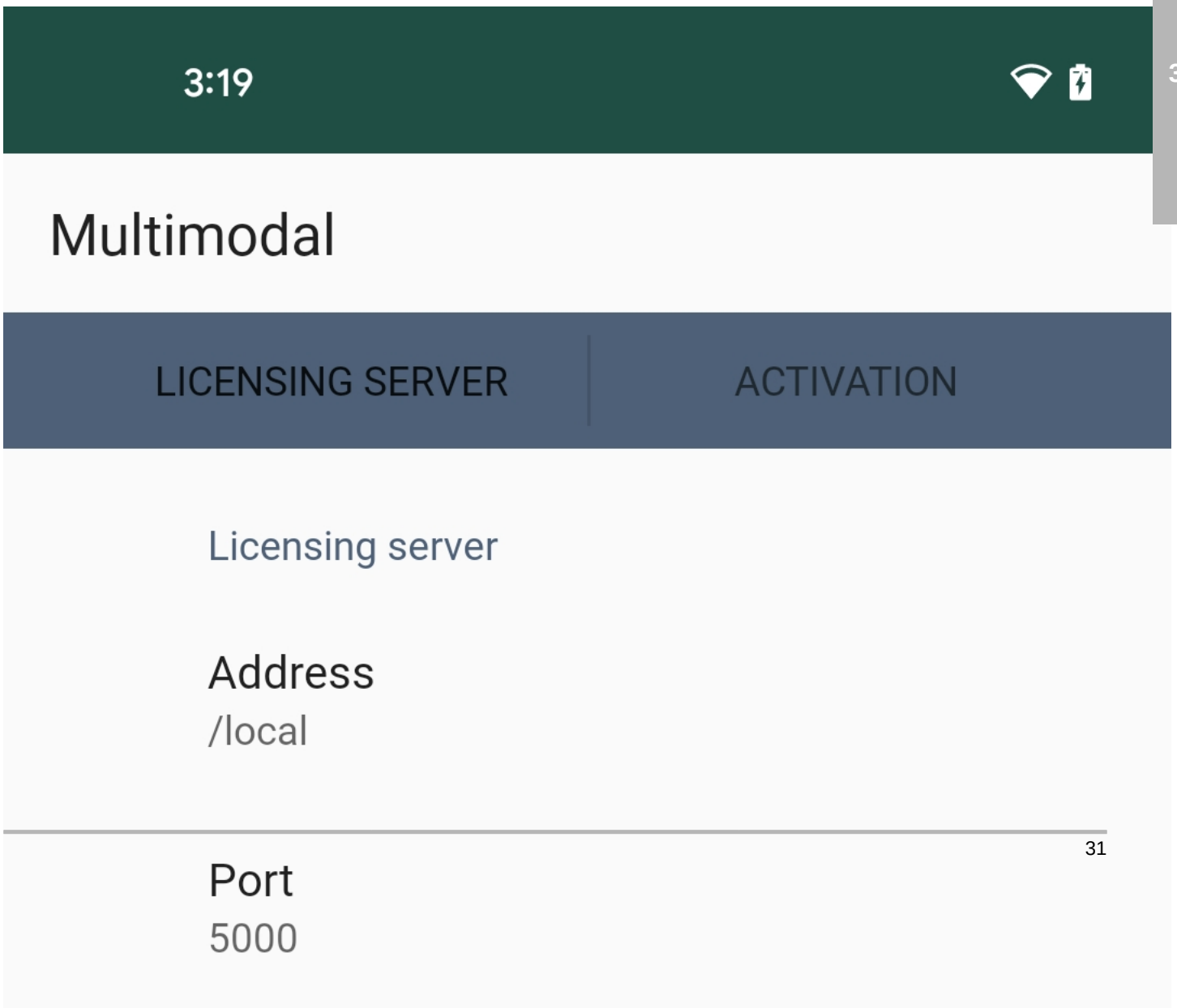
This section demonstrates how to activate a license using Multimodal sample for Android (`\Bin\Android\multibiometric-sample-android.apk`).

Copy the *multibiometric-sample-android.apk* file to your Android device and install it. Ensure that installation from *Unknown* sources is enabled in your device settings: *Settings -> Security -> Unknown sources -> OK -> Trust*.

Trial licenses activation

1. **Launch the Multimodal sample application**. By default, it's set to obtain a trial license, requiring an internet connection for activation. If direct internet access is unavailable, set up the trial product to work through a proxy server by configuring the proxy settings in the Connection Settings window: *Settings -> Activation* (in the right corner). Ensure a Wi-Fi connection for trial license activation.

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

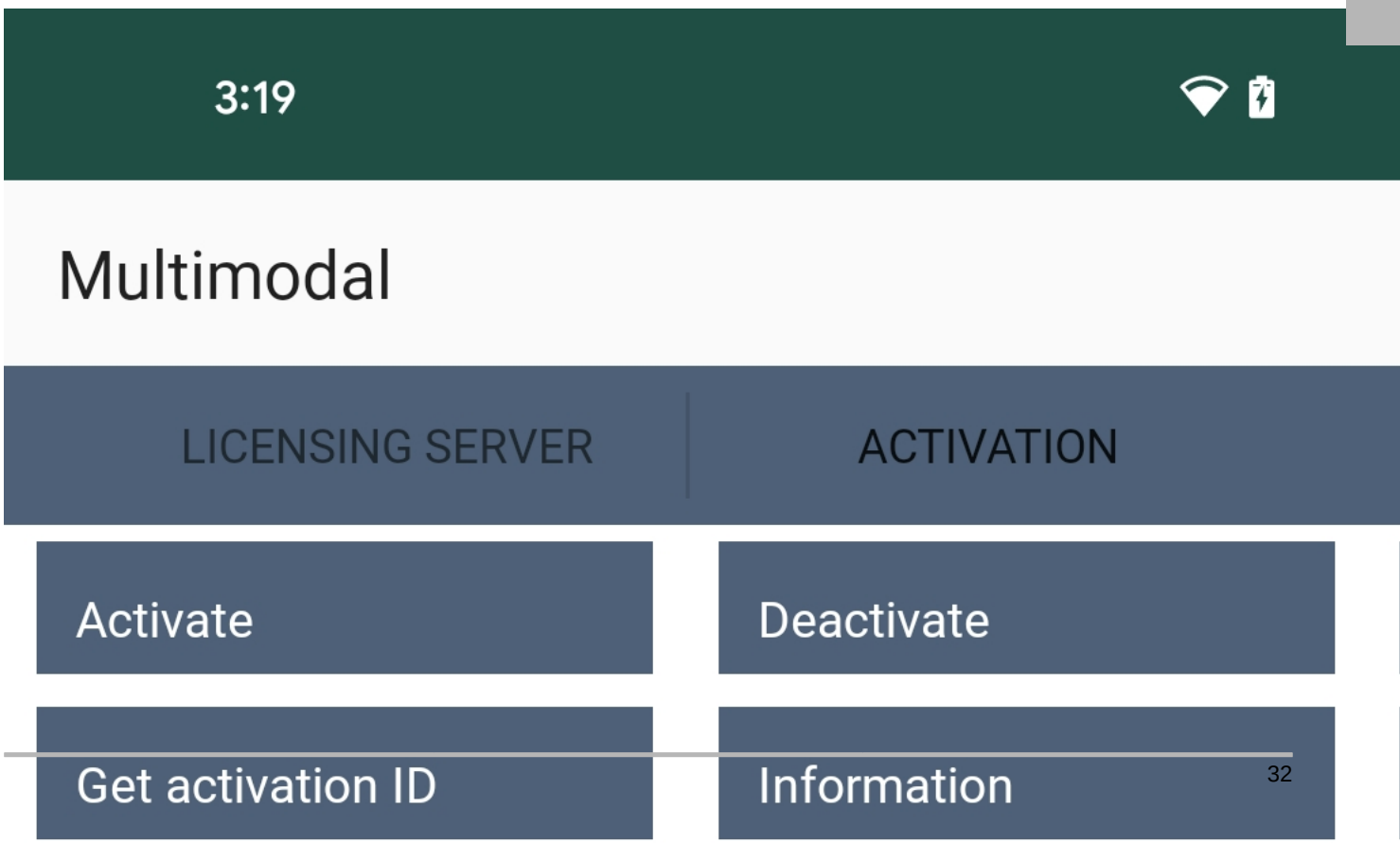
1. **Disable trial mode** by unchecking "*Use Trial Mode*" in the application settings, then restart the application.

2. **Copy licenses to Android device**. To activate purchased licenses, copy the license files to the device. The default licenses directory: *[Internal storage]/Neurotechnology/Licenses*. Serial numbers and internet licenses must be placed in this directory to be accessible. Activated licenses are moved from the Licenses directory to the application's sandbox.

- Internet licenses do not require activation; only serial numbers need to be activated. Note that temporary internet connection may be required for Standard licenses.
- 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.

3. **Access licensing settings**. Navigate to *Settings -> Activation* in the application. The following licensing options are available:

- *Activate*
  - This option allows you to select a directory containing licenses or serial numbers.
  - Licenses will be moved to the application, and serial numbers will be activated online.
  - Generated licenses will be placed in the application.
  - If no internet connection is available, activation IDs from serial numbers will be generated.
  - Users can then use these activation IDs to activate licenses using the Neurotechnology activation website.
  - Application functionality is unlocked after successful activation.
- *Deactivate*
  - This option deactivates selected licenses, allowing them to be reused for activation later.
  - Deactivation requires an internet connection.
- *Get activation ID*
  - This option retrieves previously generated activation IDs.
  - The activation IDs will be saved in the selected directory as a zip file.



Notes

- If using the Trial version of the SDK, the sample application will attempt to activate licenses automatically.
- Android licenses are activated per application sandbox, enabling the use of the license deactivation feature.
- During development, it's recommended to update the application instead of deleting and reinstalling to avoid losing licenses.
- Ensure that each Android device contains the specific licenses activated depending on the functionality being used. If you encounter an "Operation not activated" message, it means that the required license for the functionality is not activated.

4 Advanced licensing

4.1 Licenses obtain in your application

To utilize licensed biometric components in your application, you must first activate the necessary licenses. Below are guidelines for obtaining and managing licenses, along with examples of usage.

Note: Refer to the documentation for detailed instructions on how to activate licenses for the biometric components.

Trial mode

You can utilize a trial mode for testing purposes. Follow these instructions to enable trial mode:

```
NLicenseManager.TrialMode = TutorialUtils.GetTrialModeFlag();
Console.WriteLine("Trial mode: " + NLicenseManager.TrialMode);
```

Obtaining licenses

Try to obtain licenses using the following code snippet (C#):

```
try
{
    if (!NLicense.Obtain("/local", 5000, license))
    {
        throw new ApplicationException(string.Format("Could not obtain licenses : {0}",
license));
    }
    // Perform biometric operations
}
```

Licenses management

By default, activated licenses should be saved in the "*Licenses*" folder within your application's root directory. If you need to change the location of licenses, utilize the `NLicense.Add()` method to manually set the content of each activated license file.

Usage examples

Ensure you obtain the appropriate licenses for each biometric functionality required in your application. Examples include:

- Fingerprint template extraction/creation requires one of these licenses:
  - *FingerExtractor*
  - *FingerClient*
  - *FingerFastExtractor*
- Face template extraction/creation requires one of these licenses:
  - *FaceExtractor*
  - *FaceClient*
  - *FaceFastExtractor*
- Face verification requires one of these licenses (if you want to verify faces from images, you also need to create a face template, thus one of the licenses mentioned above is required):
  - *FaceMatcher*
  - *FaceFastMatcher*

SDK includes tutorials (*/Tutorials* folder) for C/C#/VB.NET/Java languages that demonstrate how to perform a biometric task and how to obtain required licenses. Also, the Developer's Guide ("*/Documentation/Neurotechnology Biometric SDK.pdf*") includes *API Reference* documentation.

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

```
// CHOOSE LICENCES !!!
// ONE of the below listed licenses is required for unlocking this sample's functionality.
//Choose a license that you currently have on your device.
// If you are using a TRIAL version - choose any of them.
const string license = "FaceClient";
//const string license = "FaceFastExtractor";
//const string license = "SentiVeillance";


// TRIAL MODE
// Below code line determines whether TRIAL 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 the licenses manager server "local"
using "5000" server's port
  if (!NLicense.Obtain("/local", 5000, license))
  {
    // If you get a "NotActivated" exception on some operation, it means that you have not
obtained
    // required licenses. So please check which licenses you have obtained and if they were
obtained successfully.
    throw new ApplicationException(string.Format("Could not obtain licenses : {0}",
license));
  }

  // Perform Facial features detection. See the tutorial source code

}
```

4.2 Licenses deactivation

Neurotechnology licenses are bound to specific devices upon activation, with a unique hardware ID generated for each device. However, there are situations where licenses need to be deactivated and reactivated. Typical scenarios requiring license deactivation include:

1. **Device transfer**. When a user needs to transfer a license from one device to another, as licenses can only operate on one device at a time.

2. **Device malfunction or hardware changes**. If the device on which the license was activated experiences malfunctions or undergoes hardware changes such as processor or hard disk replacements.

3. **Device change**. When a user switches to a different device or computer.

4. **Operating system reinstallation or version upgrade**. It is strongly recommended to deactivate a license before reinstalling the operating system or upgrading to a different OS version.

Deactivation should be carried out on the same device where activation occurred. If the device has internet connectivity, deactivation can be done automatically using the *Activation Wizard* or *LicenseDeactivation* tutorial located at "*Tutorials**Licensing*". In cases where the device is not connected to the internet, manual deactivation is required. This involves generating a deactivation ID using the console (`id_gen -dp <product name>`), which then needs to be submitted along with the license file to the Neurotechnology website. Once deactivated, the license can be reactivated on another device, or even the same computer.

The simplest method for deactivating a license is through the *Activation Wizard* on a device connected to the internet. By pressing the "Deactivate" button and selecting the license, it will be deactivated and removed from the licenses list after a short period. The deactivated license can then be activated on another device.

Alternatively, licenses can be deactivated manually using the command-line tool called `id_gen`. This tool, located in the SDK's Bin\[platform]\Activation directory, generates a computer identifier file for license deactivation or Neurotechnology components registration. Run id_gen with administrator privileges and use the following command to deactivate a license on Windows OS:

```
id_gen -dp <product name> <deactivation Id file name>
```

For example:

```
id_gen.exe -dp <VeriFinger> <deactivation.id>
```

The *deactivation.id* file generated by this command should be uploaded to the Neurotechnology website along with the license file for deactivation.

Note: Upload the deactivation file only when the device (computer) does not have an internet connection.

4.3 Changing from trial to non-trial

When you download and initiate usage of the SDK, the trial mode is automatically enabled by default. As detailed in the Trial (page 7) section, the trial mode operates under a 30-day limitation. Once you've concluded your exploration of the SDK functionalities or have acquired licenses, you can continue utilizing the same SDK by disabling the trial mode parameter.

The *\Bin\Licenses* directory of the SDK contains a file named *TrialFlag.txt*. This file can be manipulated using *False* or *True* values to enable or disable the trial mode of the SDK (*True* signifies that trial mode is enabled, while *False* indicates the use of the non-trial mode of the SDK).

Additionally, the trial and non-trial versions of the SDK can be controlled through the following methods or parameters:

- C: `NLicManSetTrialMode`
- C++: `NLicenseManager::SetTrialMode`
- Java: `NLicenseManager.setTrialMode`
- C#: `NLicenseManager.TrialMode`

To transition from trial to non-trial versions of the SDK, you have several options:

1. **Delete the *TrialFlag.txt* file**. This action will switch all samples and tutorials to the non-trial version of the SDK.
2. **Modify *TrialFlag.txt* to *False***. Changing the content of *TrialFlag.txt* to *False* achieves the same result, transitioning to the non-trial version for all samples and tutorials.
3. **Modify the source code of a sample/tutorial**. Alternatively, you can directly modify the source code of the sample or tutorial you intend to use, adjusting it to disable the trial mode.
    1. For C# you can enable or disable trial mode using this property: `NLicenseManager.TrialMode`
    2. For C, you can enable or disable trial mode using this function: `NResult N_API NLicManSetTrialMode(NBool value)`

It's important to note that when transitioning from trial to non-trial mode, you'll need to activate purchased licenses (page 8) and ensure they are correctly obtained within your application (page 35). This ensures seamless integration and usage of the licensed functionalities provided by the SDK.

4.4 Virtual environments licensing

Neurotechnology SDKs are compatible with virtual environments, but licensing within such environments requires specific configuration. Here's a breakdown of trial and non-trial versions of the SDK for virtual machines:

Trial version of SDK

The trial version of the SDK isn't inherently supported on virtual machines. However, it can be enabled with the following steps:

1. **Configure licensing service on host machine (VM Host)**:
    1. Set up a licensing service on a physical machine, referred to as the "Volume License Manager Server". Refer to the "Managing licenses on network (page 28)" section for detailed instructions.
2. **Options for using SDK from Virtual Machine**.
    1. **Use licensing service**. Configure the licensing service to obtain licenses from the "Volume License Manager Server".
    2. **Do not use licensing service**. If not using the licensing service, modify the network address /local to the IP address of the "Volume License Manager Server" when calling relevant functions/methods like `NLicenseObtain`.

Non-trial version of SDK

Virtual machines can obtain licenses from the following sources:

1. **Dongle (page 5) connected to host machine**. If the virtual environment is set up to access the host machine's USB port, the virtual machine can utilize a dongle connected to the host machine. However, this method is not recommended due to potential USB issues with certain virtualization software.
2. **Dongle connected to machine over network**. Another option is to connect a dongle to a machine accessible over the network. This host machine serves as the Volume License Manager Server, and the virtual machine connects to it to obtain licenses. Refer to the "Managing licenses on network (page 28)" section for details on license sharing over the network.
3. **Neurotechnology licensing server**. Internet licenses (page 6) can be utilized for virtual environments, allowing the virtual machine to communicate with the Neurotechnology licensing server. If internet communication is interrupted, the virtual machine can operate without connection for up to one week. Note that internet-based licenses typically require an internet connection every 30 minutes by default.

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 specifically designed for utilization in disaster recovery centers (DRCs). A DRC refers to a designated location equipped with identical infrastructure to the primary site, effectively mirroring the data environment of the primary site and remaining on standby while the primary site is operational. In the event of a failure at the primary site, the DRC seamlessly assumes operations.

It's crucial to note that disaster recovery units necessitate identical licenses as their corresponding main units. However, licenses for disaster recovery components are offered at a discounted rate. This ensures that DRCs are fully equipped and licensed to promptly resume operations in case of a primary site failure, thereby minimizing downtime and ensuring business continuity.

4.6 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.

4.6.1 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

4.6.2 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

4.6.3 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

4.6.4 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

4.6.5 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

4.6.6 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

4.6.7 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

4.6.8 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

4.6.9 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

4.6.10 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

4.6.11 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

4.6.12 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

4.6.13 SentiSight SDK

SentiSight SDK license names and licensed API components:

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

4.6.14 SentiSight Embedded SDK

SentiSight Embedded SDK license names and licensed API components:

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

4.6.15 SentiGaze SDK

SentiGaze SDK license names and licensed API components:

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

4.6.16 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 any issues during the installation or usage of the SDK, please don't hesitate to reach out to the Neurotechnology Support Department at support@neurotechnology.com for assistance. Providing additional information about your PC configuration and installation details can help expedite the resolution process.

Here's how you can obtain and share PC information based on your operating system:

For Windows:

1. Open the Activation Wizard.
2. Navigate to the "Diagnostic" window.
3. Press the "Generate" button.
4. Copy the contents of the window to the clipboard or save the text to a file.
5. Paste the information into the email message or attach the file with the saved text.

For Linux:

1. Navigate to the appropriate directory based on your system architecture:
    1. *Linux*: \Bin\Linux_x86_64\Activation
    2. *Linux ARM*: Bin\Linux_arm64\Activation\
2. Run the `diagnostic_report.sh` script as the superuser (root).
3. Copy the output to the clipboard or save it to a file.
4. Paste the information into the email message or attach the file with the saved text.

By providing detailed PC information, you'll assist the support team in diagnosing and resolving any issues you may encounter effectively.