License

Tested? You want it?

License Model Prices Quotation Order Now

Licensing

License Models

The SDK can be purchased for a single developer (Single Developer) and for all developers at one location (Branch or Site). The license obtained in this way is valid for lifetime, even without active support. At the same time, the license may be used for an unlimited number of products (royalty-free). The development of own SDK-like products on the basis of the SDK licensed here is prohibited.

Single Developer License[1]
A Single Developer License is valid exclusively for one developer, whose name, contact details and address are specified when the license is purchased. The use of the SDK is limited exclusively to this one developer. For further information, please refer to our current license terms, which we will be glad to send you in advance.

Branch or Site License[2]
A Branch License is valid exclusively for all developers at one location, which is specified with its address and the contact details of the local contact person when the license is purchased. The use of the SDK is limited exclusively to this one location. For further information, please refer to our current license terms, which we will be glad to send you in advance.

License Types

For the development of OPC UA applications, the SDK is provided in the form of different packages. Depending on the package, a smaller or larger range of functions is included. Regardless of the package used, the interfaces provided there can be enabled using the corresponding license.

Client: OPC UA Client SDK
A client license allows the use of the interfaces that are provided exclusively for client development. OPC UA (Unified Architecture) and all the client facets defined by OPC UA such as Data Access (DA), Historical Data Access (HDA), Alarms & Events (AE) or Alarms & Conditions and many more are supported.

The packages `Opc.UaFx.Client` and `Opc.UaFx.Advanced` can be used for development. The activation of the client interfaces is done by the class `Opc.UaFx.Client.Licenser`. For this license the *OPC Classic Option* can be purchased additionally.

Server: OPC UA Server SDK
A server license allows the use of the interfaces that are provided exclusively for server development. OPC UA (Unified Architecture) and all the "server facets" defined by OPC UA such as Data Access (DA), Historical Data Access (HDA), Alarms & Events (AE) or Alarms & Conditions and many more are supported.

Only the `Opc.UaFx.Advanced` package can be used for development. The activation of the server interfaces is done using the class `Opc.UaFx.Server.Licenser`.

Bundle: OPC UA Client & Server SDK
A bundle license allows the use of the interfaces that are provided for client and server development. OPC UA (Unified Architecture) and all the "client & server facets" defined by OPC UA such as Data Access (DA), Historical Data Access (HDA), Alarms & Events (AE) or Alarms & Conditions and many more are supported.

For client development the "Opc.UaFx.Client" package or the "Opc.UaFx.Advanced" package can be used. For server development only the "Opc.UaFx.Advanced" package can be used. The activation of the client

and server interfaces is done by the class `Opc.UaFx.Licenser`. For this license the *OPC Classic Option* can be purchased additionally.

OPC Classic Option
With the "OPC Classic Option" the SDK offers the possibility to develop OPC Classic client applications. For this purpose the client interfaces use an (internal) OPC Classic wrapper server. This option is only supported under Windows operating systems, because OPC Classic uses the DCOM interfaces from Microsoft for client/server communication. Additionally the runtime of the OPC Foundation for OPC Classic is required (see Downloads).

For OPC Classic Client development only the "Opc.UaFx.Advanced" package can be used. The activation of the OPC Classic Client interfaces is done using the class `Opc.UaFx.Client.Licenser` for a "Client License + OPC Classic Option" and using the class `Opc.UaFx.Licenser` for a "Bundle License + OPC Classic Option".

License Code

General

To unlock the corresponding interfaces and options, you will receive a license key after purchasing the license. This license key (a coded contiguous string of characters) must be entered in your application via the corresponding `Licenser` class. Only if the `Licenser` has received a valid license key, the acquired interfaces and options will be available.

Information

Your license key contains all information about the purchased license. This includes the data of the licensee, the licensor, the serial number, the product information, the product features/options and the intended use including data on the term, the provision, the validity or the end of support. This information is used by our products to unlock the corresponding purchased features.

Validation

The validity of a license is checked at runtime. No remote services or similar interfaces beyond the local hardware are used for this. Apart from hardware-bound licenses, no other parameters are used for validation. Whether a particular version of the product can be used is determined by comparing the release date with the support end date.

Usage

The license key is proof of the license agreement concluded between Traeger and you for the use of a product with the corresponding purchased functions. Passing on the license key thus corresponds to sublicensing, which is not provided for under the license agreement and to which the licensor is exclusively entitled. The use of a "wrapper" product and its transfer to third parties is equivalent to sublicensing and is therefore also prohibited.

Security

The license key is provided as a contiguous encoded string. Strings which are embedded in an application can always be read from the application's memory and thus copied. Any other process, such as branding the product for a licensee or providing a license file, is transferable in the same way. For this reason, the provision of the license as a string is a widely used procedure, while its use can only be ensured by trust and random verification.

Prevention

According to the license agreement, the customer is obligated to take all technically reasonable measures to secure the license key and to prevent its disclosure to third parties. Prerequisites or recommendations for such measures are not prescribed by the Licensor. If an application is already obfuscated, the license key may also be obfuscated without restrictions and embedded in the application resources in encrypted form like other character strings. Less technical measures can also be used to protect disclosure. One such option is to prevent reverse engineering of one's own products by their users, as part of one's own license agreement(s).

Negligence

It is the responsibility of the licensee to keep the license key confidential. The (deliberately) obvious disclosure of the license to third parties leads to a breach of the license conditions and thus to the consequences described therein. If a license key becomes publicly known, the original licensee can be traced via its information and held responsible. Negligence exists in particular if the license key is stored in publicly (easily) accessible locations, for example. This includes public repositories, download sites, forums, social media, file sharing networks and similar platforms.

Verification

Based on the information about the licensee in the license key, it is possible at any time to trace a license key back to the actual licensee and thus to check the validity of the use of the license. License rights are also checked (randomly) in the course of support requests. While the contact data of the inquirer(s) is being checked, the support employee can ask for further information. It is therefore advisable to provide the serial number of the license or to keep it available for faster processing of a request.

[1] , [2] There is no guarantee that the following information is up to date.

Table of Contents