Status of the documentation:
Last modified: 03.02.2016
Printing date: 03.02.2016

Developer's Guide

ise smart connect KNX Programmable SDK

Making an app for the ise smart connect KNX Programmable

Table of contents

1 How to Use the ise smart connect KNX Programmable SDK

1.1 Preconditions

To use your ise smart connect KNX Programmable SDK you need:

- Microsoft Windows 7® or later operating system[1];
- Latest available version of ETS 4 or ETS 5 provided by KNX Association[2];
- Microsoft .NET Framework 4.0® or later[1] or optionally Mono v4.0.2[3];
- Microsoft Visual Studio 2010® or later[1], Visual Studio Community Edition[1], MonoDevelop or Xamarin Studio[3].

1.2 Setting up the ise smart connect KNX Programmable SDK

Extract the ise smart connect KNX Programmable SDK. The following files and directories are now available:

```
/
|- bin
|     |- IIscAppLoaderInterfaces.dll
|     |- IscAppConnectorHost.exe
|     |- IscAppConnectorHost.exe.config
|     |- log4net.config
|     |- log4net.dll
|     |- log4net.xml
|     |- Newtonsoft.Json.dll
|- doc
|     |- SampleApp
|     |     |- Properties
|     |     |     |- AssemblyInfo.cs
|     |     |- Serial
|     |     |     |- SerialPortUsageExample.cs
|     |     |- DataPointUsageExample.cs
|     |     |- IscApp.cs
|     |     |- SampleApp.csproj
|     |     |- SdCardUsageExample.cs
|     |     |- UserPartitionUsageExample.cs
|     |- iseSmartConnectKNXProgrammable.chm
|     |- SampleApp.sln
|     |- SampleAppCOConfiguration.xml
|- iseSmartConnectKnxEtsEditor
|     |- iseSmartConnectKnxEtsEditor.exe
|     |- KimDataTypes.xml
|- ProgrammableAppSigner
      |- ProgrammableAppSigner.exe
```

[1] Microsoft, .Net Framework, Visual Studio and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

[2] ETS and KNX are registered trademarks of KNX Association cvba.

[3] All text and image content on mono-project.com, unless otherwise specified, is licensed under a Creative Commons Attribution-Share Alike 3.0 United States License.
This does not include the Mono name, logo, or icon, which are registered trademarks of Novell, Inc.
This does not include Mono source code.

1.3 Workflow

This section describes the development for the ise smart connect KNX Programmable. Each step is described in more detail in a later section of the document.

1. Designing and creating the ETS project:
   To make the first step, determine which group objects and parameters serve the purpose of your app. To create them, use the *iseSmartConnectETSEditor.exe* provided with the ise smart connect Programmable SDK (see chapter 1.6 ise smart connect ETS Configuration Editor).

2. Developing the app:
   Determine the development environment you want to develop your app with (see chapter 1.1 Preconditions)

3. Create an app project:
   Using the selected development environment, create the app project (see chapter 1.4 Create ise smart connect KNX App Project). We recommend using the SampleApp as a start project first, which you can develop further later. This simplifies the development as well as the debugging on the device.

   If you like to develop with Mono, please refer to chapter 1.5 Developing with Mono.

4. Sign your App (see 1.9 Signing your App) to create a deployable app.
   Note: The ProgrammableAppSigner allows only ASCII characters as filenames. Empty directories in the project output directory are also prohibited.

5. Deploying to the device:
   a. Install the application ZIP via the web page of the device (see 1.10 Deploying an App to the ise smart connect KNX Programmable).

   b. Download the ETS project (see 1.7 ETS Download).

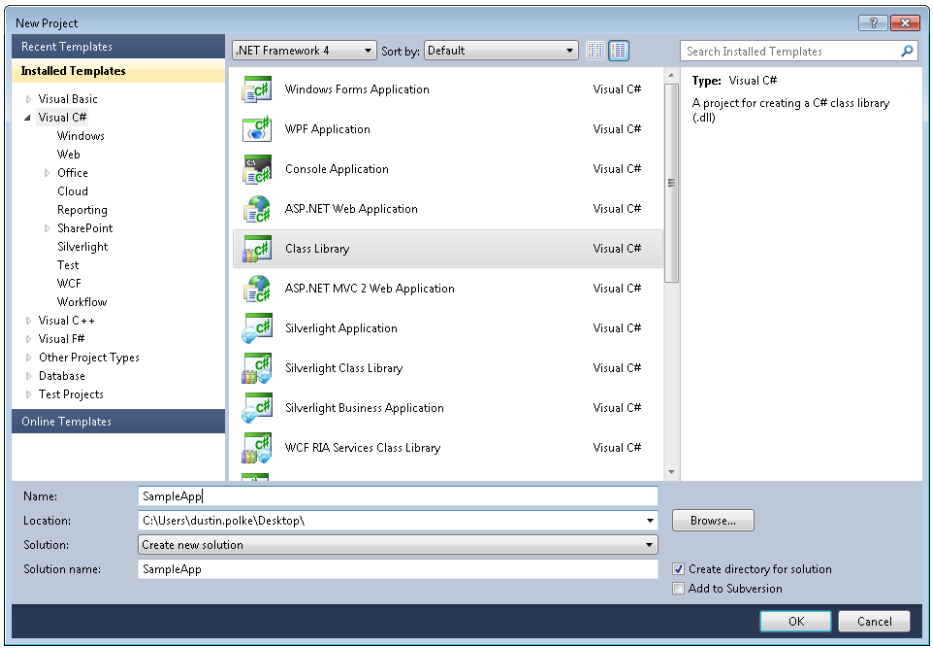
6. Enable debugging with the device:
   a. Set the device in debug mode using the webpage (see 1.11 Enabling and Disabling Development Mode).

   b. Set the project properties to launch the app on the PC (see 1.12 Launching the App on Your Development PC).

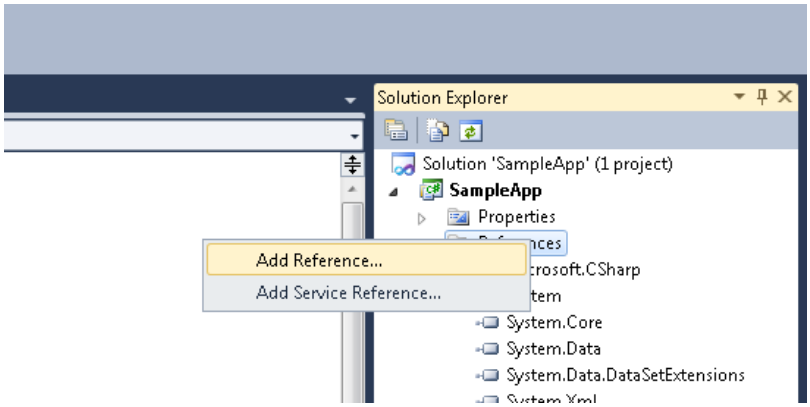
1.4 Create ise smart connect KNX App Project

1.4.1 Using Microsoft Visual Studio

1. Create a new Visual C# project from the Class Library template:
   The names of the C# project and the main namespace used in the project have to be identical to the AppId you entered in the iseSmartConnectETSEditor.



2. Add references to *IIscAppLoaderInterface.dll* and *log4net.dll* provided by the ise smart connect KNX Programmable SDK:

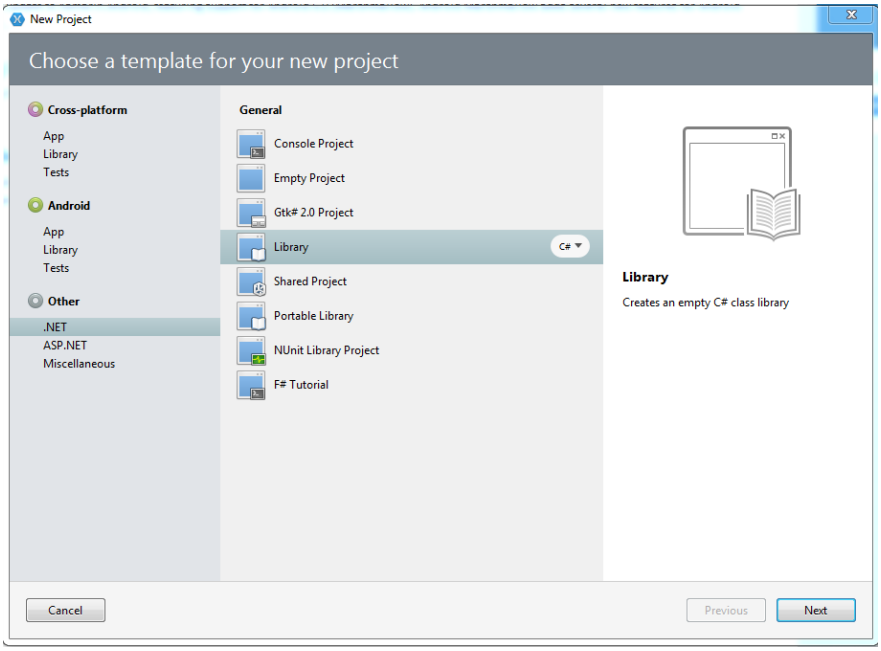


3. Now create a class named *IscApp* (attention: the name of the class is case sensitive and has to be *IscApp*) inheriting from *IIscApp* and implement the interface methods. You can use the *SampleApp.cs* provided in the *Example* folder of the SDK.
4. *log4net* can be used for logging. See */doc/SampleApp/IscApp.cs* for examples. The location of the log file during PC development can be configured in a configuration file as explained in chapter 1.16 Additional Configuration Settings.

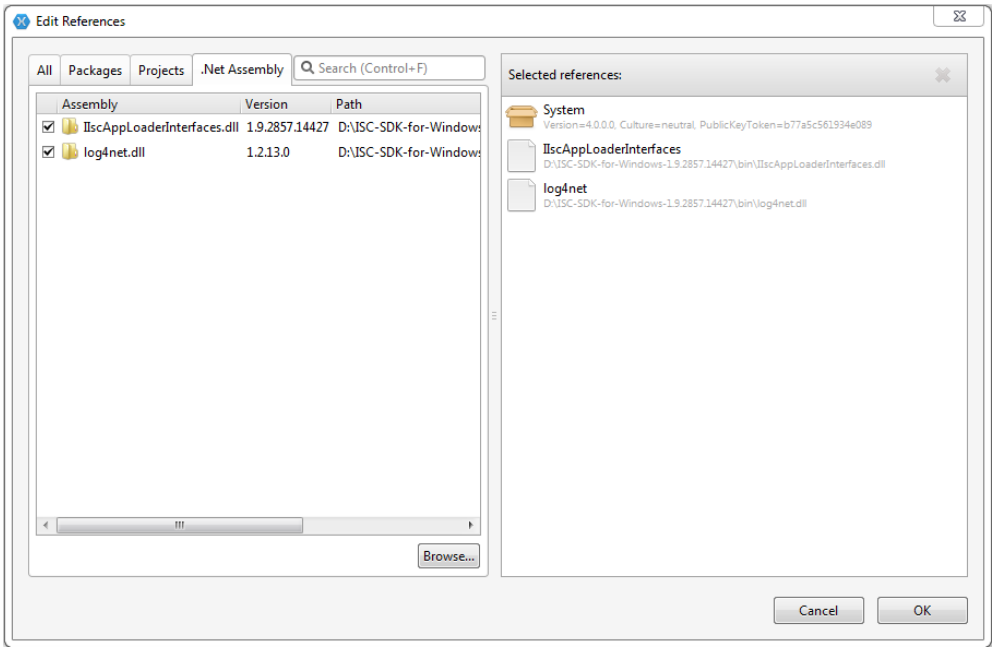
1.4.2 Using Xamarin Studio

The development with Mono Develop for Linux is similar to Xamarin Studio.

1. Create a new .NET C# class library. The project name must be identical with the main namespace and have to be identical with the *AppId*:



2. Add reference to IIscAppLoaderInterface.dll and log4net.dll provided by the ise smart connect KNX Programmable SDK:



3. Now create a class named *IscApp* (attention: the name of the class is case sensitive and has to be *IscApp*) inheriting from *IIscApp* and implement the interface methods. You can use the *SampleApp.cs* provided in the Example folder of the SDK.
4. *log4net* can be used for logging. See */doc/SampleApp/IscApp.cs* for examples. The location of the log file during PC development can be configured in a config file explained in chapter 1.16 Additional Configuration Settings.
5. Please set the target framework to *Mono / 4.0* or *.NET 4.0 Client Profile*.

1.5 Developing with Mono

The ise smart connect KNX Programmable uses the Mono runtime v4.0.2. The sources are available under http://download.mono-project.com/sources/mono/. The mono runtime setup for Windows can be found in http://download.mono-project.com/archive/4.0.2/.

Please refer to the following links for further information:

- http://www.mono-project.com/
  Web page of the Mono software platform.
- http://www.mono-project.com/docs/getting-started/application-portability/
  Detailed information on how to program portable applications for Windows/Linux, e.g. path separators.
- https://visualstudiogallery.msdn.microsoft.com/cb83d210-b09f-4e21-949e-81ad23684c78
  MonoHelper for XBuild integration into Visual Studio.

For a list of provided assemblies, please refer to the iseSmartConnectKNXProgrammable.chm.

1.6 ise smart connect ETS Configuration Editor

The ETS configuration file defines the KNX communication objects and ise smart connect KNX app parameters you want to use within your app. It is used to configure the ise smart connect KNX device in the ETS to run the specified app and make the communication object available in the ETS project.

1.6.1 Supported KNX Datapoint Types

The following table lists the .NET data types for the corresponding KNX datapoint type (DPT). If not all subtypes are supported, only the available DPT numbers are given.
Please refer to the *KNX Datapoint Type for Interworking* (chapter 3/7/2) in the *KNX Standard v2.1* for a detailed description on how to interpret the DPTs.

KNX DPT	.Net data type
1 bit - DPTs 1.*	bool
2 bit - DPTs 2.*	*DptControl*
4 bits - DPTs 3.007 and 3.008	*DptControlDimmingBlinds*
1 byte unsigned - DPTs 5.* and 17.001	byte
1 byte signed - DPTs 6.001 and 6.010	sbyte
1 byte DptSceneControl - DPT 18.001	*DptSceneControl*
1 byte DptSceneInfo - DPT 26.001	*DptSceneInfo*
2 bytes unsigned - DPTs 7.*	ushort
2 bytes signed - DPTs 8.*	short
2 bytes float - DPTs 9.*	double
3 bytes DptTimeOfDay - DPT 10.001	*DptTimeOfDay*
3 bytes Date - DPT 11.001	DateTime
3 bytes unsigned - DPT 232.600	*DptColourRGB*
4 bytes unsigned - DPTs 12.*	Uint
4 bytes signed - DPTs 13.*	int
4 bytes float - DPTs 14.*	double
14 bytes - DPTs 16.000 (ASCII) and 16.001 (ISO 8859-1)	string

To prevent the transmission of invalid bit patterns, it is necessary to use the supported .NET data types of the *IIscAppLoaderInterfaces* namespace for some DPTs. These .NET types provide functions to get/set the KNX specific DPT fields respecting the allowed range and/or combinations.

1.6.2 The iseSmartConnectKnxEtsEditor

The iseSmartConnectKnxEtsEditor tool from the SDK *(/iseSmartConnectKnxEtsEditor/ iseSmartConnectKnxEtsEditor.exe*) provides an easy interface to edit or create a configuration file for the ise smart connect KNX Programmable.

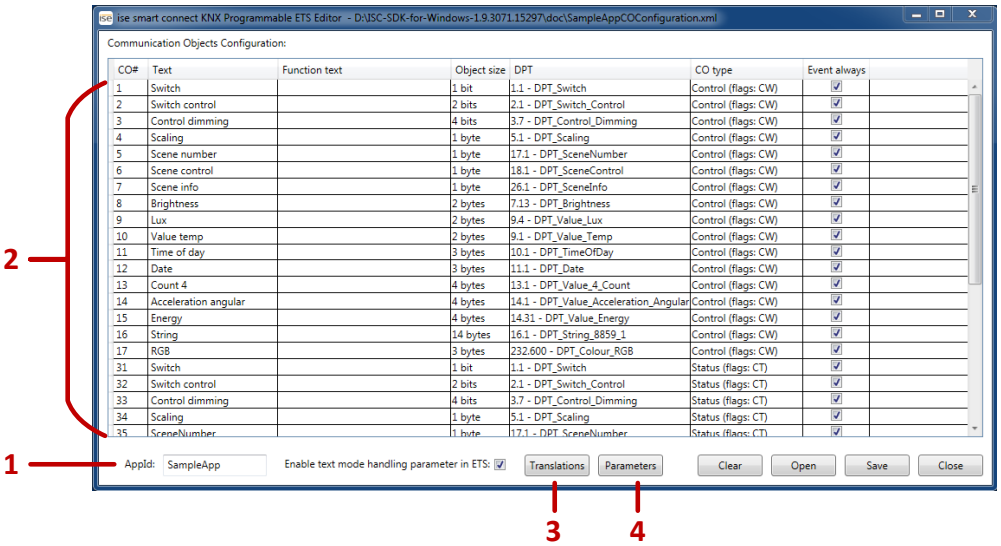


Figure 1: The iseSmartConnectKnxEtsEditor tool

(1) Please note that the application project in your development environment (e.g. C# project) has to be named exactly as the *AppId*. Otherwise your application will not run.

(2) The communication objects can be added in the main window with an editable number, text and function text. The selectable object size defines the available DPTs.
The maximum number of communication objects, which can be imported into the ETS using the plugin, is 64. The highest communication object number, which can be used, is 64. That means that the communication objects which have numbers higher than 64 will not be imported, even if the total number of the communication objects configured in the iseSmartConnectKnxEtsEditor is less than 64.

The CO type can be set to *Control*, *Status* or *Status & Control*, setting the corresponding flags for the ETS configuration.

Additionally, for each communication object you can configure the flag *Event always* (see column Event always in the iseSmartConnectKnxEtsEditor).
If this flag is unset, telegrams will only be sent/received if the value of the communication object has changed. If this flag is set, telegrams will always be sent/received even if the value of the communication object has not changed. Therefore, during the development, if you see telegram that you do not expect, or if you miss telegrams that you expect, this flag could be the cause for such behaviour.

A CO entry can be deleted by right-click the CO row and press the delete key.

(3) For each communication object, a translation text and function text could be added in a supported ETS product language.

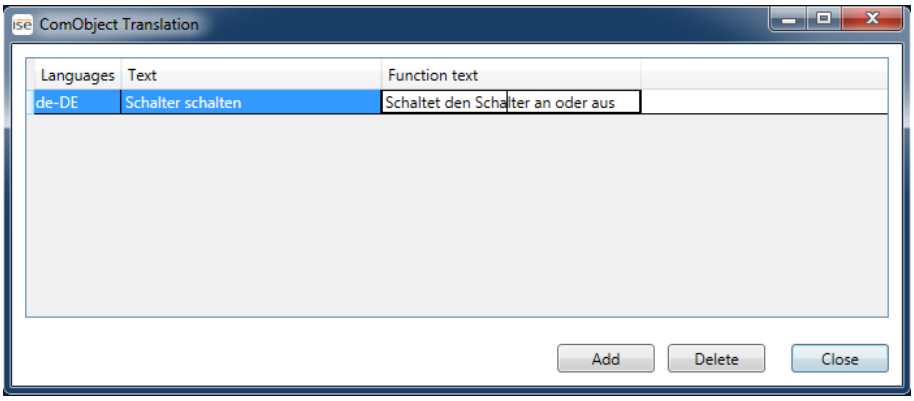


Figure 2: Communication object translations editor window.

(4) Three types of parameters can be configured: IP parameter, number parameter and text parameter. The maximum number of parameters is 30 and at most 10 parameters of each type.

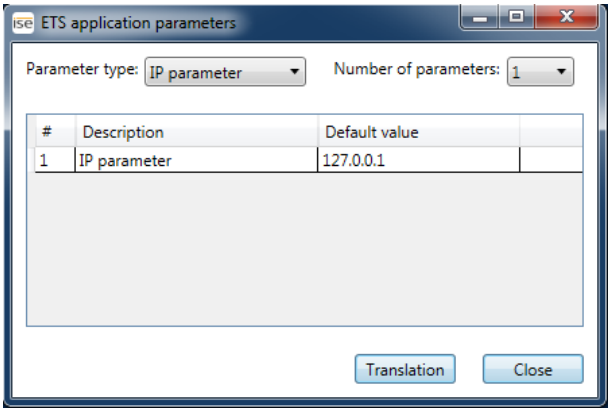


Figure 3: ETS application parameters window.

For each parameter, a translation could be added via the translation button.

The application program configuration is stored as an XML file via the save button.

1.6.3 Configure the ise smart connect KNX Programmable

To configure the *ise smart connect KNX Programmable* device, right-click the device in the ETS and select *Plug-In -> Import configuration*. In the following dialog, select the configuration file.

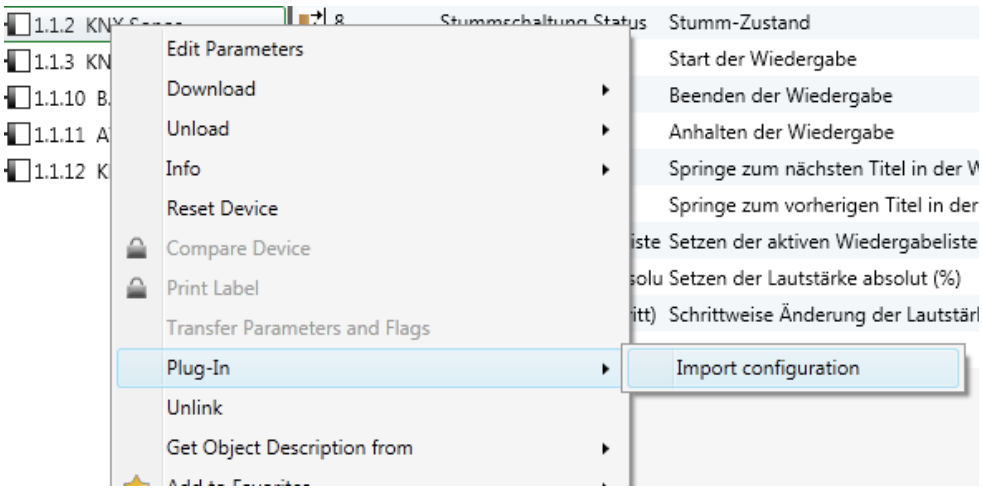


Figure 4: Import

For a quick start, the SDK contains an example configuration at */doc/SampleAppCOConfiguration.xml*.

1.6.4 Update Configuration File

To update your configuration (like adding, deleting, or renaming a CO and/or parameter, or even changing the data point type), open the configuration file again and make your changes, save it and re-import it via the plug-in again. The changes you made will be obtained in the ETS. If you have made an association (assigned a group address to a CO) and updated its CO, like renaming it for example, the association will not be changed.

A group address association is removed in the following cases:

- CO was deleted;
- CO number was changed;
- The object size of the CO was changed, and the group address is already assigned to another CO.

If you update your ETS project, you may have to update your Programmable application to be compatible with the changes you have made to your project.

If you intend to change parameters, be aware that the application retrieves the actual values by index of the parameter, not by its name. If you change the meaning of a parameter, remove a parameter, or add a required parameter, you may have to update your Programmable application accordingly.

It is recommended to test your application after you made any changes to the ETS configuration.

1.7 ETS Download

After creating a configuration and loading it with the plugin in the ETS, a download of the project with the ETS is required.

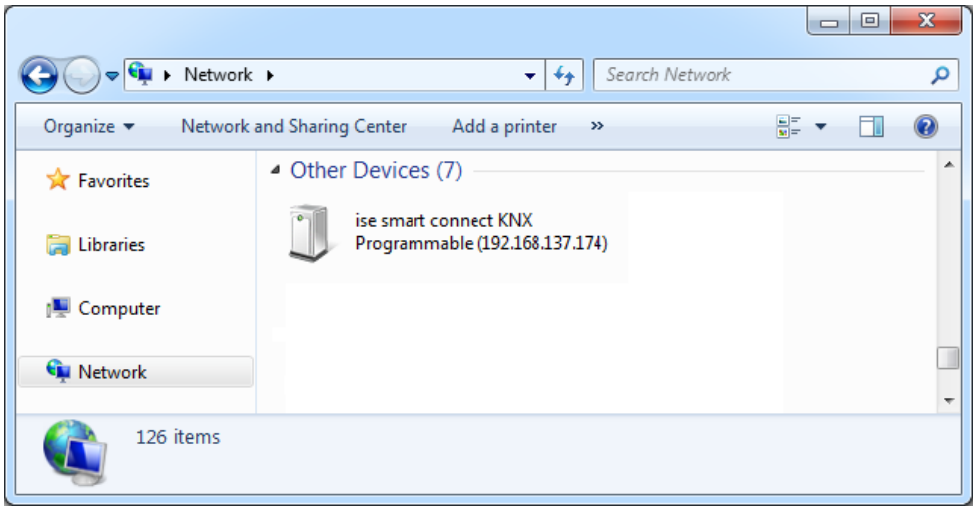
To program the physical address, the ise smart connect KNX Programmable device can be put into programming mode by pressing the programming button shortly or by pressing the button *Enable Programming Mode* in the web interface (see chapter 1.8

Displaying the Website of the Device). The red programming LED indicates programming mode. Then, using the ETS, download the physical address. After configuring the physical address, download the application program.

1.8 Displaying the Website of the Device

The website of the device provides useful information about the ise smart connect KNX Programmable, like the software version, physical address, IP address, development mode, etc.

- In order to open the website, double-click the icon of the device in the *Other Devices* area in the Windows network environment.



- Alternatively, you can also enter the IP address of the device in your browser.

1.9 Signing your App

To sign your app, use the ProgrammableAppSigner (provided by the SDK).

The ProgrammableAppSigner requires a PKCS#12 file with a certificate signed by ise Individuelle Software-Entwicklung GmbH. To obtain a certificate, you have to send us a certificate request generated from an X.509 private key.

We recommend using the program *X Certificate and Key Management*, which you can download for various operating systems from http://sourceforge.net/projects/xca.

1.9.1 Creating a Certificate with *X Certificate*

1. Select *File->New DataBase* [sic] to create a new (OS independent) storage for the certificates or select *File->Open DataBase* [sic] to open an existing database. Each database is protected with by password. Keep your database and password safe to prevent abuse.

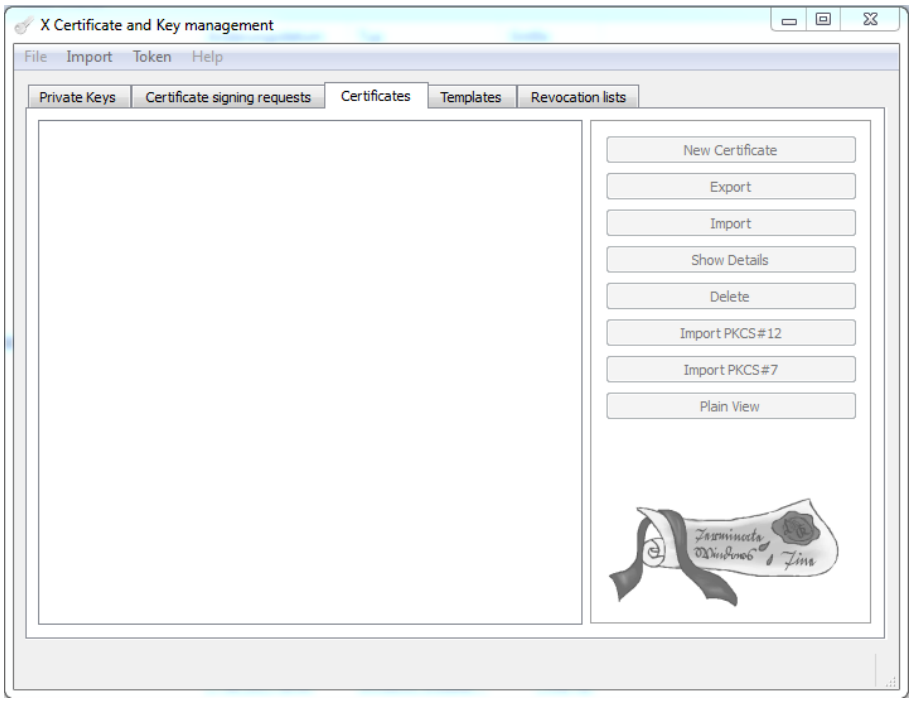


Figure 5: Creating a certificate with X Certificate.

2. Set *File->Options* to use SHA256 as default hash algorithm:

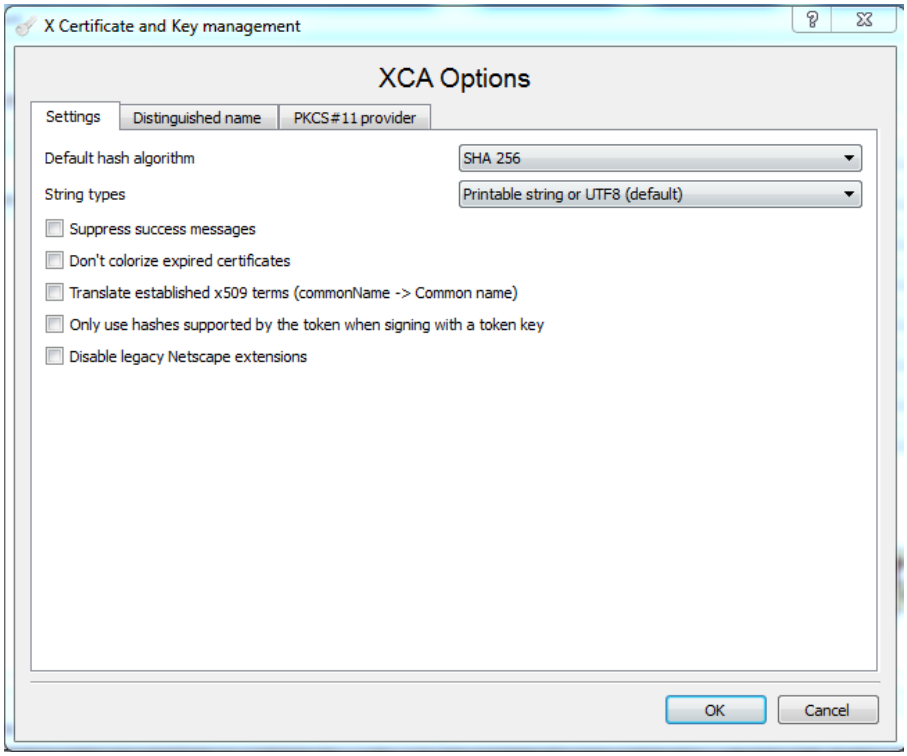


Figure 6: Using SHA256 as default hash algorithm.

1.9.2 SDK user part

1. Create a private key: Select the *Private Keys* tab and click the *New Key* button. Now enter a key name and use key type RSA with a size of 4096 bit. Finally click *Create*.

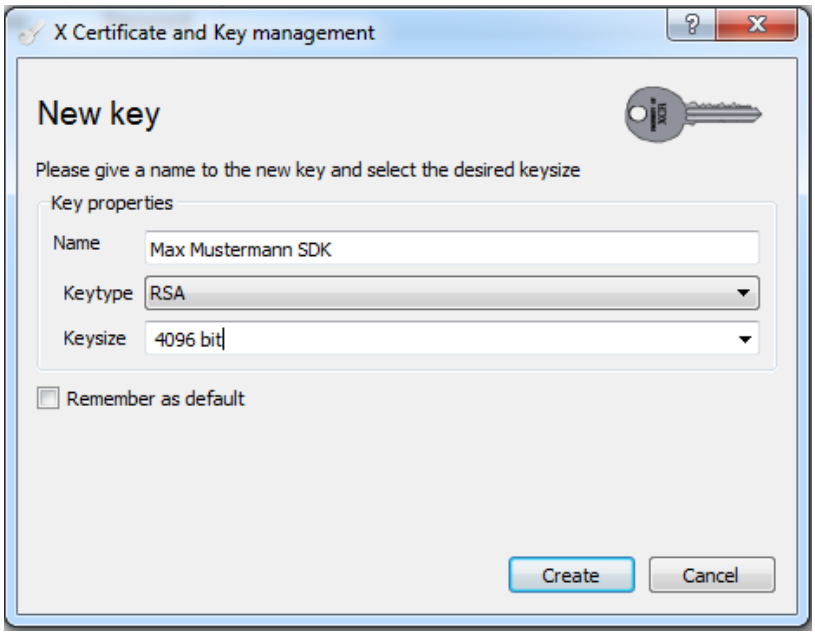


Figure 7: Creating a new key.

2. Create a certificate signing request: Select the *certificate signing requests* tab and click the *new request* button. In the *Create Certificate signing request* window, select the *Subject* tab and enter the following information:

- countryName: the country code for your location, e.g. DE for Germany.
- localityName: your city name, e.g. Oldenburg.
- organizationName: the name of your company.
- organizationUnitName: Development or specific name of your department.
- commonName: your name.
- emailAddress: your contact email.
- Private key: select the previously generated private key.

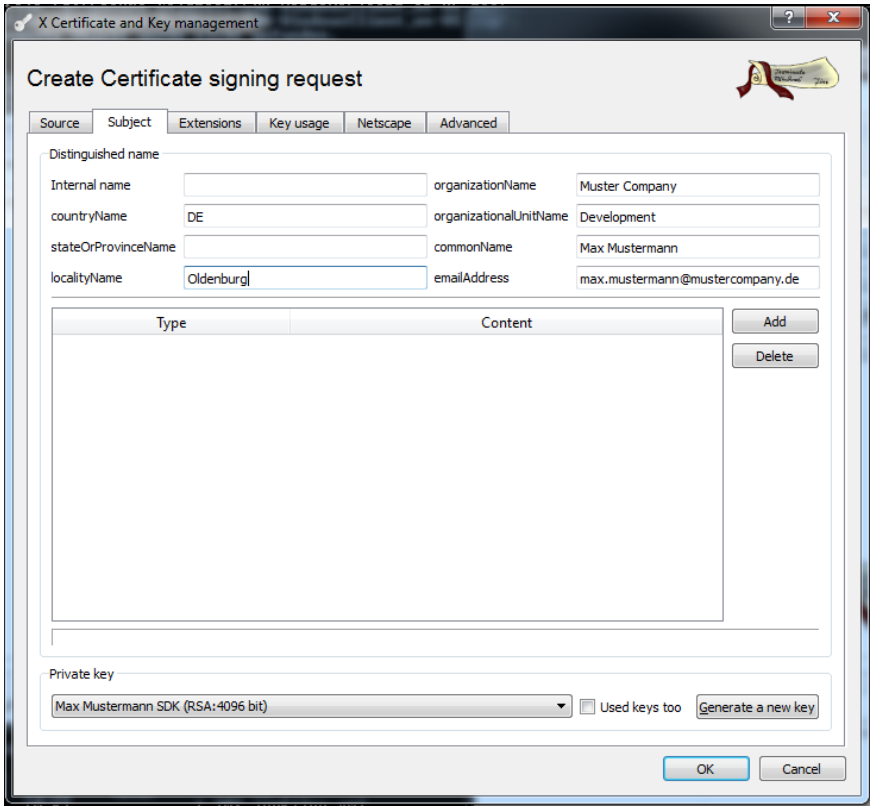


Figure 8: Creating certificate signing request.

Now select the *Key usage* tab and select *Digital Signature* and *Non Repudiation* and click on *OK* to generate the request.

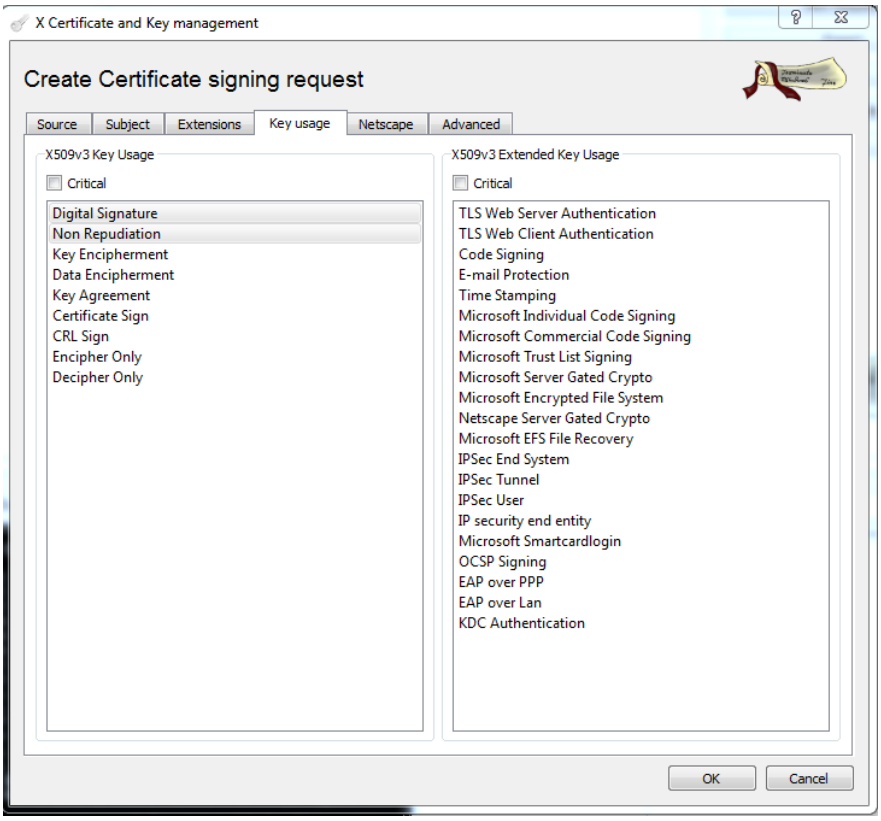


Figure 9: Selecting usage type of the key.

Now export the request via the *Export* button in *PEM* format and send the generated file to ise Individuelle Software-Entwicklung GmbH via email. Please do not send your X Certificate Database, your private key or any of your passwords to us. Also, do not give these to anyone else who should not be able to release applications under your name.

3. The ise Individuelle Software-Entwicklung GmbH will check your request and send you a signed software development certificate via email and our public root certificate. To import these certificates, select the *Certificates* tab and first import the ise certificate, then your developer certificate via the *Import* button. To trust our certificate, right-click the certificate and select *Trust->Always trust this certificate*.
4. Create a p12 certificate with your private key: Finally select your certificate and click the *Export* button. Now choose *PKCS #12* format for the exported file and press *Ok*. ***Choose a safe password, this is your private password, we will never ask for the p12 file or your password! Be aware, that everybody with your p12 file and your password can sign applications in your name!***

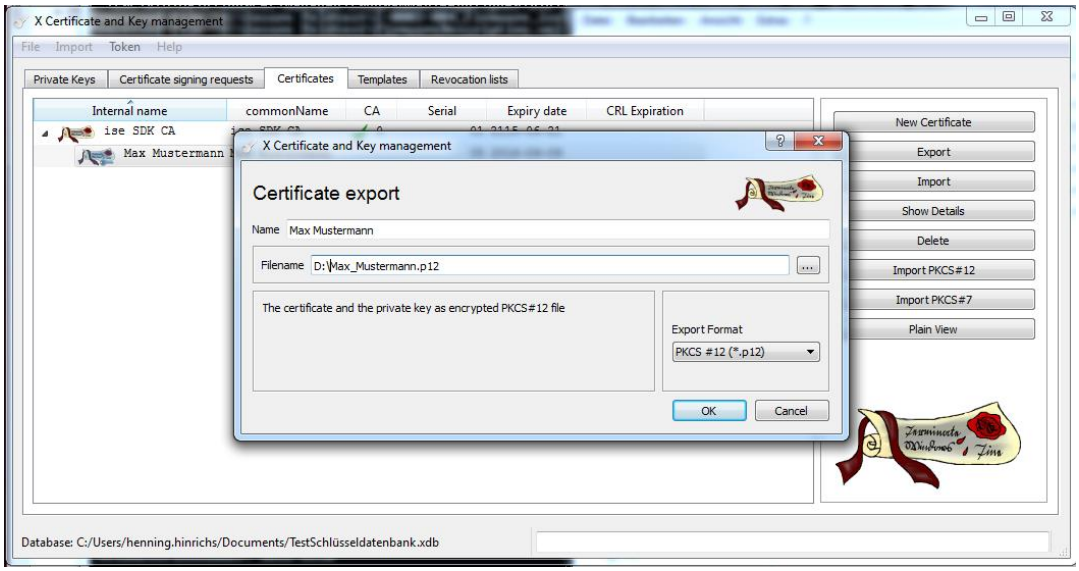


Figure 10: Creating a p12 and exporting certificate.

5. The created p12 file can be used by our ProgrammableAppSigner application to sign your programmable apps.

1.9.3 ProgrammableAppSigner

An app can be signed for a specific instance or for any instance of an ise smart connect KNX Programmable depending on the MAC address given to the ProgrammableAppSigner.

If you want to sign your app to run on any ise smart connect KNX Programmable, you need to specify 00:00:00:00:00:00 as the MAC address.

To sign an app, call the ProgrammableAppSigner on the command line and invoke it with the following arguments:

ProgrammableAppSigner.exe *<App> <p12 file> <p12 password> <MAC address> (<output archive>)*

The five command line options have the following meanings:

- <*App*>: absolute path to the directory containing your application files including the main DLL, e.g. *C:\SampleApp*;
- <*p12 file*>: absolute path to the p12 file, e.g. *C:\isc_sdk\app.p12*;
- <*p12 password*>: the password you have chosen for your p12 file;
- <*MAC address*>: mac address of the ise smart connect KNX Programmable device for which the certificate is to be created; use *00:00:00:00:00:00* if an app should run on any device;
- (optional) <*output archive*>: the relative (to the current working directory) or absolute file name of the resulting ZIP archive; if a file already exits it will be overwritten. Defaults to <*App*>*/app.zip*. If you omit the .zip file ending, it will be added automatically.

The ProgrammableAppSigner will produce four files when the arguments are correct:

- <App>/appfiles.xml
- <App>/appfiles.xml.sig
- <App>/sdk.crt
- <output archive>(.zip)

Existing files of these names will be irrevocably destroyed.

1.10 Deploying an App to the ise smart connect KNX Programmable

You can deploy debug or release binaries to the device. To deploy a .NET app to the ise smart connect KNX Programmable, use the ZIP archive that was created by the ProgrammableAppSigner. The other generated files are not needed anymore. Be aware that the ZIP file size should not be greater than 30MB to avoid problems with future app updates and firmware updates.

On the website, click *Application Installation*, then click the *Search* button to navigate to the folder containing the ZIP archive. After selecting the archive, click the *Application Installation* button beneath the search button.

To start the deployed app, click the button to disable the development mode. If it is already disabled, enable it and disable it again. This will start your new app.

The *appinfo.txt* content is displayed on the device web page as optional application information. It is useful to display some information, like app version, app name and/or manufacturer.

Please do not add the *log4net.dll* and *IIscAppLoaderInterface.dll* files to the ZIP archive because they are already available on the device.

1.11 Enabling and Disabling Development Mode

Development mode is disabled by default. To enable or disable it, use the corresponding button of the web interface of the device. By default the button is named *Enable development mode*. Clicking on the button starts development mode, which allows the developer to debug the application on their PC. If development mode is active, the name of the button changes to *Disable development mode*.

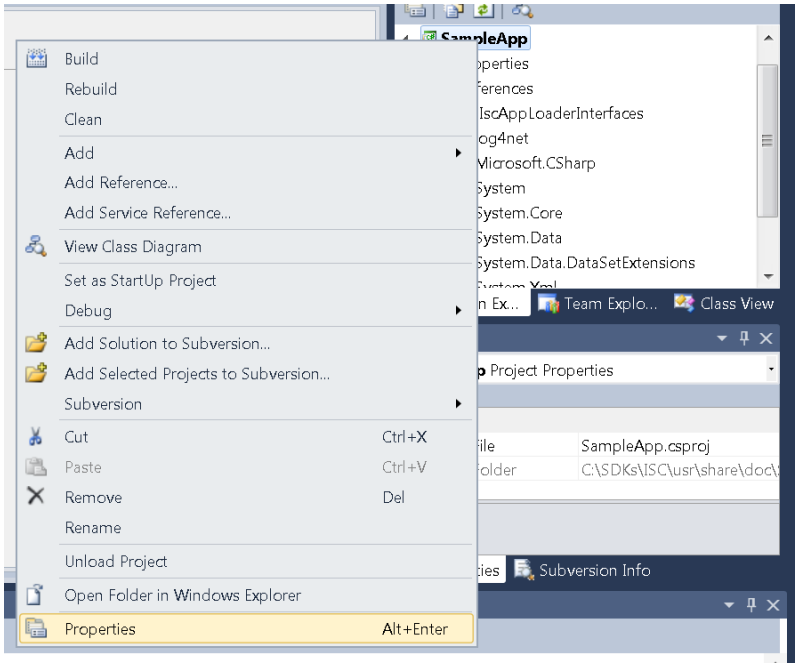
By disabling development mode, the application is started on the device. The development mode will also be disabled if you

- restart the device, e.g. reboot or power loss,
- reset the device via ETS, or
- download the ETS project.

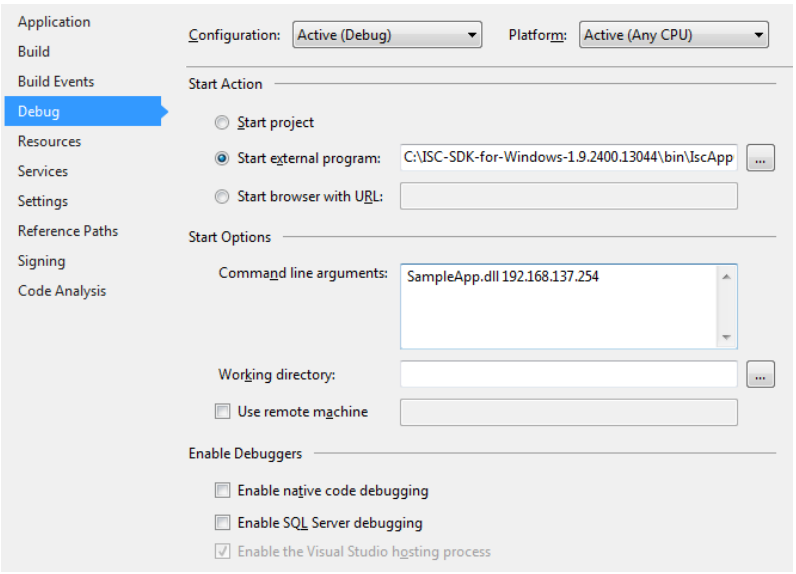
1.12 Launching the App on Your Development PC

To ease development of a smart connect application, the developer can start their library using the executable *IscAppConnectorHost.exe* from the SDK on the development PC. By doing this, you can start and debug your code like any other local application. To do that, please follow the instructions below for Microsoft Visual Studio:

1. Enable development mode, see 1.11.
2. Reload webpage until the App State displays *stopping* then *running*.
3. Open the *Properties* of the app project and go to the tab *Debug*:



4. Select *Start external program* and choose *IscAppConnectorHost.exe* from the SDK bin directory.
5. Set the command line arguments:
   - Case insensitive name of the assembly of your created library including the extension (“.dll”)
   - IP address of the *ise smart connect KNX Programmable* to connect to



6. After configuring the start procedure, you can begin to debug your application or if additional resources like a serial port or an SD Card are needed, please check first the following chapters.

1.13 Supervision of the Health State of your App

The ise smart connect KNX Programmable watches the Mono process that runs your application and tries to communicate with it regularly. The communication happens automatically in a background thread of the application process. If the process does not respond within one minute, the Programmable assumes that the application is dead and reboots the whole operating system as an attempt to return to a working state. For this reason you should never exit the Mono process your application is running in (e.g. by Environment.Exit).

If your application is running on a PC in development mode, this health check is inactive. The device will not reboot when you close the process. When you disable the development mode, the application will start on the Programmable and the health check will be active again.

1.14 Using an SD Card

To provide additional storage for the application, there is the possibility to use an SD card. The SD card has to be FAT32 formatted; otherwise, the device will not be able to recognize it. Please note that the capacity of the SD Cards must not exceed 32 GB.
To simulate an SD card when debugging on your development PC, the location to be accessed needs to be configured. The configuration of this step is described in the next chapter.

An example how to use the SD card can be found in the *SampleApp* project in the class *SdCardUsageExampleClass*.

1.15 Using the USB Port

This section is relevant if your ise smart connect KNX Programmable has a USB port.

The USB port is designed to support only one USB device with 300mA maximum current. The adapter will be reconnected automatically if you pull and reinsert it again. Please note that the name may have changed (e.g. from */dev/ttyUSB0* to */dev/ttyUSB1*). Therefore, it is necessary to get the name (using the property *SerialPortName*) again.

Nevertheless, an EMC-compatible error handling has to be implemented by the app developer to warrant the CE conformity. To do so, you have to use the method *ResetUsbConfiguration()* of the interface *IIscAppLoaderInterfaces*.

In the *SampleApp* delivered with the ise smart connect KNX Programmable SDK, you find an example of how to implement this error handling using the method *ResetUsbConfiguration()*.

1.15.1 Supported USB-to-Serial Adapters

The ise smart connect KNX Programmable supports USB-to-Serial adapter based on FTDI and Prolific 2303 chipsets.

The following products have been tested:

- ATEN USB-to-Serial Converter model UC-232A
- LogiLink USB2.0 to Serial Adapter
- pluggable USB Serial Adapter model PL2303-DB9
- Trendnet TU-S9

1.15.2 Supported USB CEC Adapter

The ise smart connect KNX Programmable supports the Pulse-Eight USB – CEC adapter model P8-USBCECv1 via a kernel module. Examples and further description can be found in our forum at http://forum.ise.de/wlbb/index.php/Board/18-Programmable-App-Developer/.

1.16 Additional Configuration Settings

In some cases, your application might need some additional resources like a serial port or an SD card. These resources are located on your PC and you have to make sure that the starting application *IscAppConnectorHost.exe* knows the location of these resources. Therefore, in the directory of the *IscAppConnectorHost.exe*, an additional file called *IscAppConnectorHost.exe.config* can be found. Please change this file to reflect the resource locations on your pc, which you want to use from your app.

When do you need to change the user setting in *IscAppConnectorHost.exe.config*?

- If you are using serial communication, you need to configure the name of the serial port, e.g. *COM2*. Otherwise, make sure that no value has been entered.
- If you are using the SD card, you need to configure the path where to simulate the SD card, e.g. *G:*\. You can simulate the presence of the SD card as well by setting a Boolean value (true/false) to the value tag of *ConfigSDCardPresentDevelopment* variable.
- If you want to store app-specific data on the device, you need to provide a path where to store this data on your development PC. This path will be returned if *GetConfigRootPath()* is called within your app. It is important to use the function *GetConfigRootPath()* to get the appropriate path for the configuration data. Never use a hard coded path in your application.
- If you would like to change the creation path of the log file.

Example Configuration:

```
<?xml version="1.0"?>
<configuration>
  <userSettings>
    <IscAppConnectorHost.Properties.Settings>
      <setting name="ConfigRootPathDevelopment" serializeAs="String">
        <value>.\</value>
      </setting>
      <setting name="SerialInterfaceDevelopment" serializeAs="String">
        <value>COM3</value>
      </setting>
      <setting name="ConfigSDCardPresentDevelopment" serializeAs="String">
        <value>True</value>
      </setting>
      <setting name="ConfigSDCardPathDevelopment" serializeAs="String">
        <value>G:\</value>
      </setting>
    </IscAppConnectorHost.Properties.Settings>
  </userSettings>
</configuration>
```

If you would like to change the creation path and file name of the log file, please change the xml attribute value of the following node:

```
<param name="File" value="IscAppConnectorHost.log"/>.
```

This node is located within the log4net config file section:

```
 <log4net>
    <appender name="LogFileAppender" type="log4net.Appender.RollingFileAppender">
      <param name="File" value="IscAppConnectorHost.log"/>
      <lockingModel type="log4net.Appender.FileAppender+MinimalLock" />
      <appendToFile value="true" />
      <rollingStyle value="Size" />
      <maxSizeRollBackups value="2" />
      <maximumFileSize value="1MB" />
      <staticLogFileName value="true" />
      <layout type="log4net.Layout.PatternLayout">
        <param name="ConversionPattern" value="%d [%t] %-5p %c %m%n"/>
      </layout>
    </appender>

    <root>
      <level value="ALL" />
      <appender-ref ref="LogFileAppender" />
    </root>
  </log4net>
```

1.17 Pitfalls during Development

During the development of a new application, you should be aware of some pitfalls.

First, there is always the problem with missing telegrams. Either you have sent a KNX telegram or you expect to receive a KNX telegram and the result is that this is not happening. The cases for telegram loss are manifold, but here is a small checklist to avoid this from the perspective of your application.

There are two scenarios upon other, which could lead to this behaviour. First, check if the datatype of the value you are sending is correct and identical with the datatype of your communication object in the ETS. Second, there is a flag called *Event always*, which has to be configured for the communication object you are sending the value to. Please refer to the documentation in chapter 1.6.2 The iseSmartConnectKnxEtsEditor to accomplish this task.

1.18 How to Log Using the SDK

The ise smart connect KNX Programmable SDK provides a logging mechanism, which allows you to easily log your application.

1.18.1 Defining the Log Object

Since you have referenced *log4net.dll* provided with the ise smart connect KNX Programmable SDK, you can define your logging object, as following, to use the provided logging mechanism:

```
private static readonly ILog Logger = LogManager.GetLogger(typeof(IscApp));
```

For more information about the logging code, see the C# project *SampleApp*, which is provided with the SDK.

1.18.2 Logging Modes

The ise smart connect KNX Programmable supports two logging modes: info mode and debug mode.

1. Info log mode:
   The info level is the default logging level. Only in case of an error, the given log information will be written into the log file.

2. Debug log mode:
   All information of debug and above log levels will be written into the log file.

Debug mode is activated by clicking the button *activate* next to *log4net Debug-logging* on the *Application Installation* webpage of the device. Deactivating debug log mode starts the info log mode (the default mode) again. Please note that activating and/or deactivating the debug log mode causes the app to restart.

A factory reset as well as a firmware update deactivates debug log mode.

1.18.3 Important Log File on ise smart connect KNX Programmable

An archive with log files can be downloaded via the device web page. The name of the important file for debugging and development is *IscAppConnectorHost.log* which is located within */var/log* in the archive. This file contains the log output of your application and its host process on the device. The files in the directory */var/log* contains only log messages since the last device start. A persistent log file of fatal log messages is *IscAppConnectorHost_Fatal.log* within */opt/userdata/log*. To be able to view the log file correctly formatted on your PC, please use a viewer like Notepad++ or Visual Studio.

1.19 Troubleshooting and Support

If you have a problem with your ise smart connect KNX Programmable and require support, please send an email with a detailed error description and the whole archive of the log files (see next chapter) created after the error occurred to support@ise.de.

1.20 Status page of the ise smart connect KNX Programmable

You can call up the device status on the website of the ise smart connect KNX Programmable. Among other things, it displays the installed software version and other useful information about ise smart connect KNX Programmable.

2 License agreement ise smart connect KNX Programmable software

Hereinafter are the contract terms for your use of the software as the "Licensee".
By accepting this agreement and installing the ise smart connect KNX Programmable software or putting the ise smart connect KNX Programmableinto use, you conclude an agreement with ise Individuelle Software-Entwicklung GmbH and agree to be legally bound to the terms of this agreement.

2.1 Definitions

Licensor: ise Individuelle Software-Entwicklung GmbH, Oldenburg, Osterstraße 15, Germany.
Licensee: The legal recipient of the ise smart connect KNX Programmable software.
Firmware: Software which is embedded on the ise smart connect KNX Programmable hardware and enables operation of the ise smart connect KNX Programmable.
ise smart connect KNX Programmable software: The ise smart connect KNX Programmable software designates all of the software provided for the ise smart connect KNX Programmable product, including the operating data. This includes, in particular, the firmware and the product database.

2.2 Object of the agreement

The object of this agreement is the ise smart connect KNX Programmable software provided on data media or through downloads, as well as the corresponding documentation in written and electronic form.

2.3 Rights of use of the ise smart connect KNX Programmable software

The Licensor grants the Licensee the non-exclusive, non-transferable right to use the ise smart connect KNX Programmable software for an unlimited time in accordance with the following conditions for the purposes and applications specified in the valid version of the documentation (which shall be provided in printed form or also as online help or online documentation).
The Licensee is obliged to ensure that each person who uses the program only does so as part of this license agreement and observes this license agreement.

2.4 Restriction of rights of use

2.4.1 Copying, modification and transmission

The Licensee is not authorised to use, copy, modify or transfer the ise smart connect KNX Programmable software in whole or in part in any way other than as described herein. Excluded from this is one (1) copy produced by the Licensee exclusively for archiving and backup purposes.

2.4.2 Reverse engineering and conversion technologies

The licensee is not authorised to apply reverse-engineering techniques to the ise smart connect KNX Programmable software or to convert the ise smart connect KNX Programmable software to another form. Such techniques include, in particular, disassembly (conversion of the binary-coded computer instructions of an executable program into an assembler language which can be read by humans) or decompilation (conversion of binary-coded computer instructions or assembler instructions into source code in the form of high-level language instructions).

2.4.3 Firmware and hardware

The firmware may only be installed and used on the hardware (ise smart connect KNX Programmable) approved by the Licensor.

2.4.4 Transfer to a third party

The ise smart connect KNX Programmable software may not be passed on to third parties, nor may it be made accessible to third parties.

2.4.5 Renting out, leasing out and sub-licensing

The Licensee is not authorised to rent or lease the ise smart connect KNX Programmable software or grant sub-licenses to the program.

2.4.6 Software creation

The Licensee requires written approval from the Licensor to create and distribute software which is derived from the ise smart connect KNX Programmable software.

2.4.7 The mechanisms of license management and copy protection

The mechanisms of the license management and copying protection of the ise smart connect KNX Programmable software may not be analysed, published, circumvented or disabled.

2.5 Ownership, confidentiality

2.5.1 Documentation

The ise smart connect KNX Programmable software and the documentation (which shall be provided in printed form or also as online help or online documentation) are business secrets of the Licensor and/or the object of copyright and/or other rights and shall continue to belong to the Licensor. The Licensee shall observe these rights.

2.5.2 Transfer to a third party

Neither the software nor the data backup copy nor the documentation (which shall be provided in printed form or also as online help or online documentation) may be passed on to third parties at any point in time, in whole or in part, for a charge or free of charge.

2.6 Changes, additional deliveries

The ise smart connect KNX Programmable software and the documentation (which shall be provided in printed form or additionally as online help or online documentation) shall be subject to possible changes by the licensor.

2.7 Warranty

The ise smart connect KNX Programmable software shall be delivered together with software from third parties as listed in Chapter 3 – *Open Source Software*. No warranty is provided for software from third parties.

2.7.1 Software and documentation

The ise smart connect KNX Programmable software and the documentation (which shall be provided in printed form or additionally as online help or online documentation) shall be provided to the licensee in the respective valid version. The warranty period for the ise smart connect KNX Programmable software is twenty-four (24) months. During this time, the licensor shall provide the following warranty:

- The software shall be free of material and manufacturing defects when turned over to the customer.
- The software shall function in accordance with the documentation included with it in the respective valid version.
- The software shall be executable on the computer stations specified by the Licensor.

The warranty shall be fulfilled with the supply of spare parts.

2.7.2 Limitation of warranty

Otherwise, no warranty shall be provided for the freedom from faults of the ise smart connect KNX Programmable software and its data structures from defects. Nor does the warranty cover defects due to improper use or other causes outside the influence of the Licensor. Any additional warranty claims shall be excluded.

2.8 Liability

The Licensor shall not be liable for damages due to loss of profit, data loss or any other financial loss resulting as part of the use of the ise smart connect KNX Programmable software, even if the Licensor is aware of the possibility of damage of that type.

This limitation of liability is valid for all damage claims of the Licensee, regardless of the legal basis. In any case, liability is limited to the purchase price of the product.
The exclusion of liability does not apply to damage caused by premeditation or gross negligence on the part of the Licensor. Furthermore, claims based on the statutory regulations for product liability shall remain intact.

2.9 Applicable law

This agreement is subject to the laws of the Federal Republic of Germany.
The place of jurisdiction is Oldenburg.

2.10 Termination

This agreement and the rights granted herein shall end if the Licensee fails to fulfil one or more provisions of this agreement or terminates this agreement in writing. The ise smart connect KNX Programmable software and the documentation turned over (which is provided in printed form or also as online help or online documentation) including all copies shall in this case be returned immediately and without being requested to do so. No claim to reimbursement of the price paid shall be accepted in this case.
The license for use of the ise smart connect KNX Programmable software shall expire upon termination of the agreement. In this case, the ise smart connect KNX Programmable product must be taken out of operation. Further use of the ise smart connect KNX Programmable without a license is precluded.
The commissioning software and visualisation software must be uninstalled and all copies must be destroyed or returned to the Licensor.

2.11 Subsidiary agreements and changes to the agreement

Subsidiary agreements and changes to the agreement shall only be valid in writing.

2.12 Exception

All rights not expressly mentioned in this agreement are reserved.

3 Open Source Software

This product uses software from third-party sources used within the scope of the GNU General Public License (GPL) or Lesser GNU General Public License LGPL and within the scope of the Berkeley Software Distribution (BSD) and the MIT license.

The software packages used in this product which are licensed within the scope stated here are described in the following.

Software package	U-Boot
Version of the software	2012.07
Source	http://www.denx.de/wiki/U-Boot/WebHome
License	GNU GPL, Version 2, June 1991
Copyright notice	Copyright © 2000-2012 by Wolfgang Denk et al.

Software package	GNU/Linux
Version of the software	3.2.20
Source	http://kernel.org
License	GNU GPL, Version 2, June 1991
Copyright notice	Copyright © 1992-2013 by Linus Torvalds et al.

Software package	Buildroot
Version of the software	2012.11
Source	http://buildroot.org
License	GNU GPL, Version 2, June 1991
Copyright notice	Copyright © 1999-2005 Erik Andersen, 2006-2012 The Buildroot developers

Software package	GNU C Library (GLIBC)
Version of the software	2.30.3
Source	http://www.gnu.org/s/libc/
License	GNU LGPL, Version 2.1, February 1999
Copyright notice	Copyright © 1996-2012 by Roland McGrath et al.

Software package	Boost C++ Libraries
Version of the software	1.49.0
Source	http://www.boost.org
License	Boost Software Licence, version 1.0
Copyright notice	Copyright 2012 Boost.org

Software package	libupnp
Version of the software	1.6.17
Source	http://sourceforge.net/projects/pupnp/files/pupnp/
License	BSD
Copyright notice	Copyright (c) 2000-2003, Intel Corporation. All rights reserved.

Software package	Websocketpp
Version of the software	0.3.x
Source	http://www.zaphoyd.com/websocketpp
License	BSD
Copyright notice	Copyright (c) 2013, Peter Thorson. All rights reserved.

Software package	jQuery
Version of the software	1.11.1
Source	https://jquery.org
License	MIT Licence
Copyright notice	Copyright 2014 The jQuery Foundation

Software package	Newtonsoft.Json
Version of the software	7.0.1
Source	http://www.newtonsoft.com/json
License	MIT Licence
Copyright notice	Copyright 2007 James Newton-King

Software package	Apache log4net for .NET Framework
Version of the software	1.2.13.0
Source	https://logging.apache.org/log4net/
License	Apache License, Version 2.0
Copyright notice	Copyright 2004-2013 The Apache Software Foundation

The license texts of the GPL and LGPL are available via the following web page: http://www.gnu.org/licenses/licenses.html

The source code for this software can be obtained via the email address info@ise.de.

This offer is valid for 3 years after the discontinuation of the service for this product.

4 GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of

it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

> **a)** You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
> **b)** You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
> **c)** If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES

OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS