KNX Cookbook

ETS App development

2

6

1

Summary

This document is a development help for KNX newcomers.

This document describes the development of an ETS App.

This document is part of the KNX Specifications v2.1.

Document updates

Version	Date	Modifications
01.01.01	2013.03.01	Final version.
01.01.02	2013.10.14	Editorial updates for the publication of KNX Specifications 2.1.

References

[01] KNX Flyer - “How to become an ETS App Developer.”
(Available from the website of KNX Association.)

Filename: 02_06_01 ETS App Development v01.01.02.docx
Version: 01.01.02
Status: Final version.
Save date: 2013.10.14
Number of pages: 12

Contents

1 Developer environment

1.1 Required software

- All supported development environments can be found under the ETS4 SDK requirements.
- The latest ETS4 version.
- The latest ETS4 SDK version.

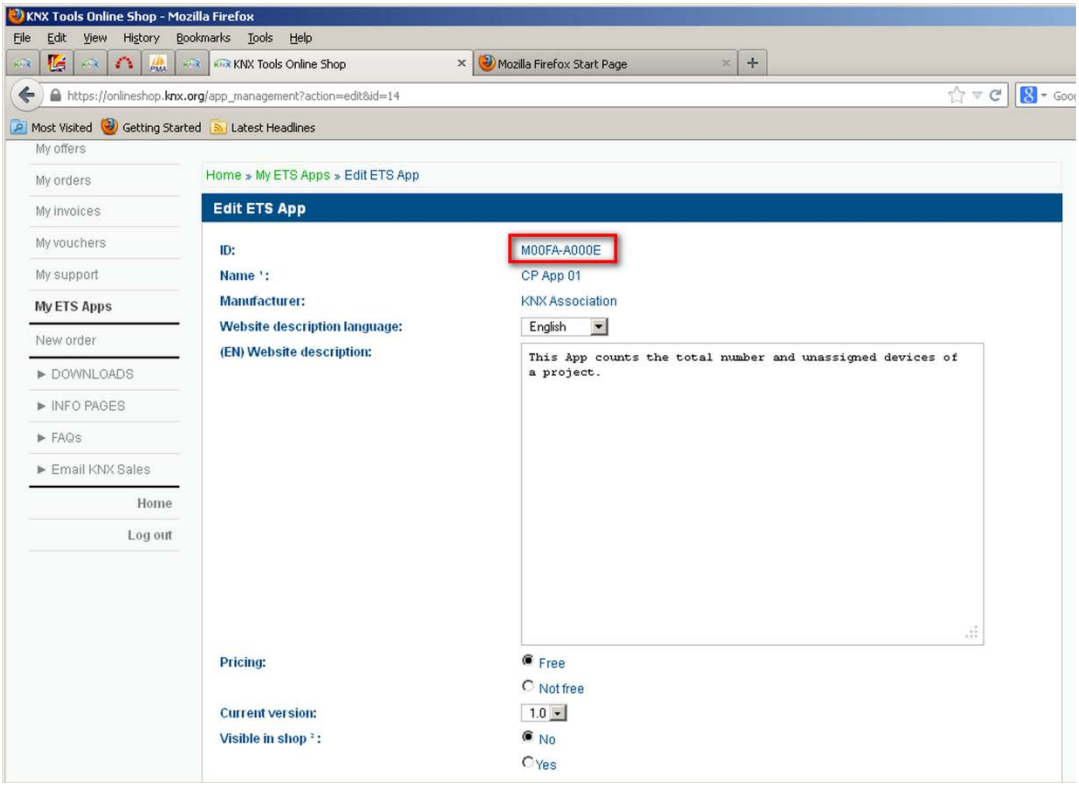
1.2 About the specific example in this document

- Development environment used: Microsoft® Visual Studio ® 2012 Professional.
- Programming language used: C#.

2 Steps

2.1 Before you start

- Install ETS4.
- Install a supported development environment.
- Install the ETS4 SDK.
- Use your KNX Online Shop account or make one if you do not have one yet.
- Make sure that your KNX Online Shop account has been set to customer type = “Member”. If this is not the case, open a support ticket via your KNX Online Shop account with this request.
- Apply to become an ETS App developer via your KNX Online Shop account, see [01].
- Declare your first App and get its ID, in this example the App ID = M00FA-A000E.



2.2 Actual developments steps

2.2.1 Step 1: Create the first version and check its validation

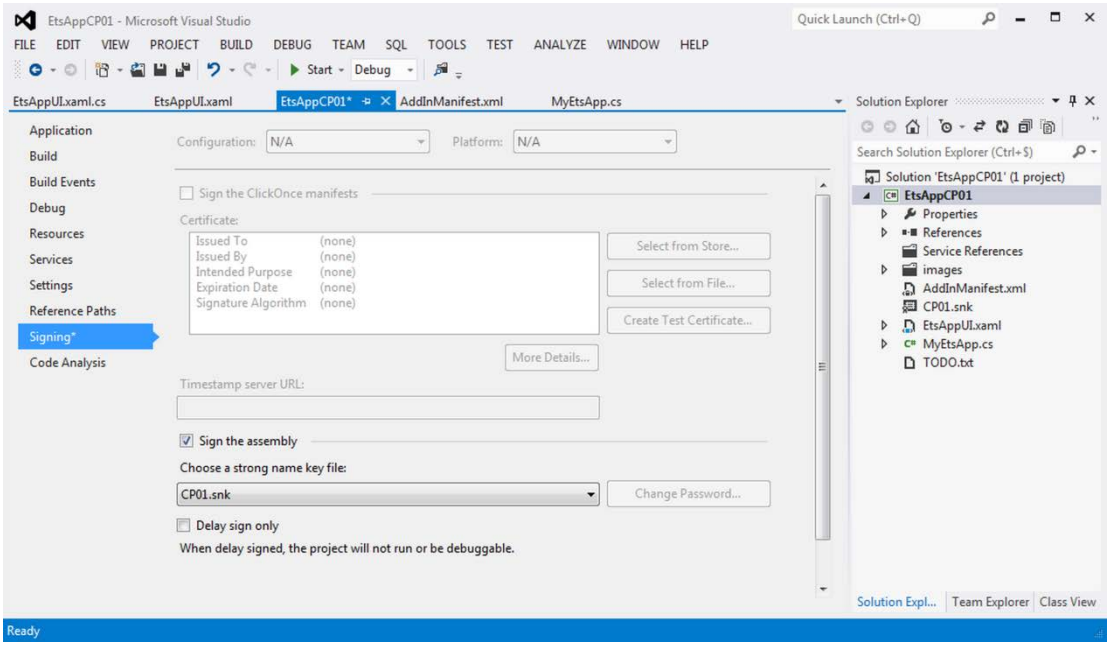
- Create a new (Visual Studio) project, e.g. ‘myEtsApp’ based on the ‘EtsApp’ template.
- Make the following changes in myEtsApp.cs.

```
[AddIn("CP App 01", Version = "1.0.0.0", Publisher = "stuff from KNX")]
```

```
public class MyEtsApp : IEts4AddIn, IDisposable
{
        private IHostNotification myHost;
        private Project myProject;
…
```

```
public string AppId
{
        get { return "M00FA-A000E"; }
}
```

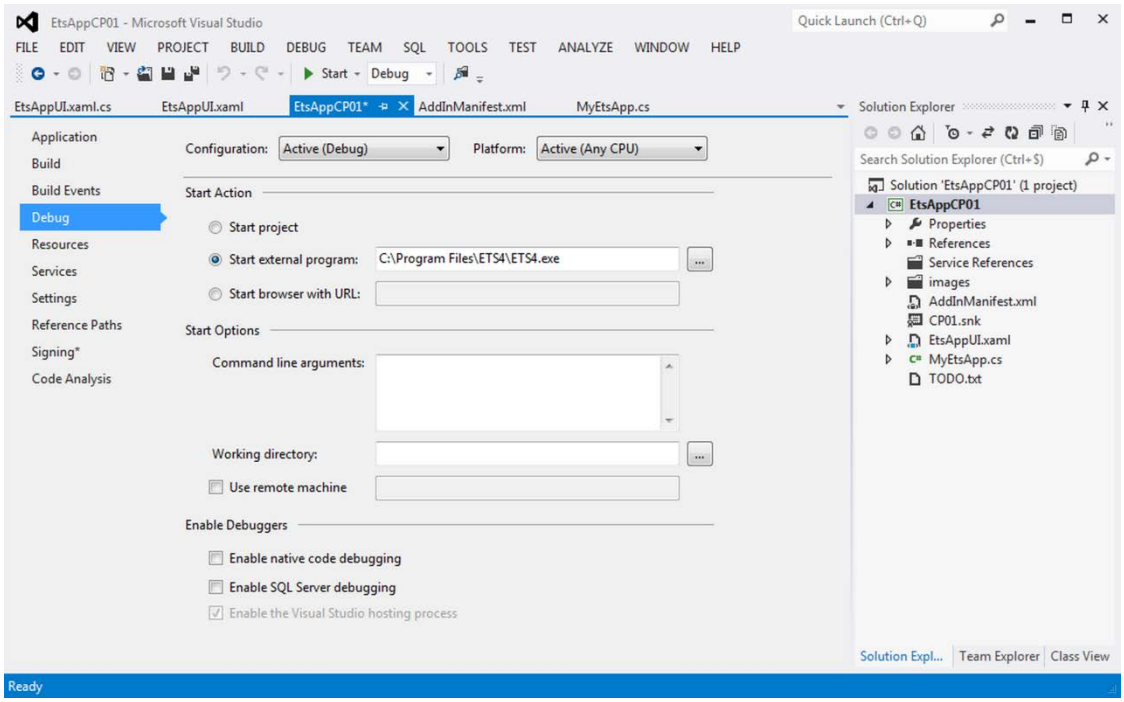
- Add a help file to the project (otherwise validation will fail).
- Modify the addInManifest.XML file accordingly.
- Sign the project (assembly) with a strong name.
  - Open the Properties page for the project.
  - Click the property page “Signing”.
  - Modify the property “Choose a strong name key”.



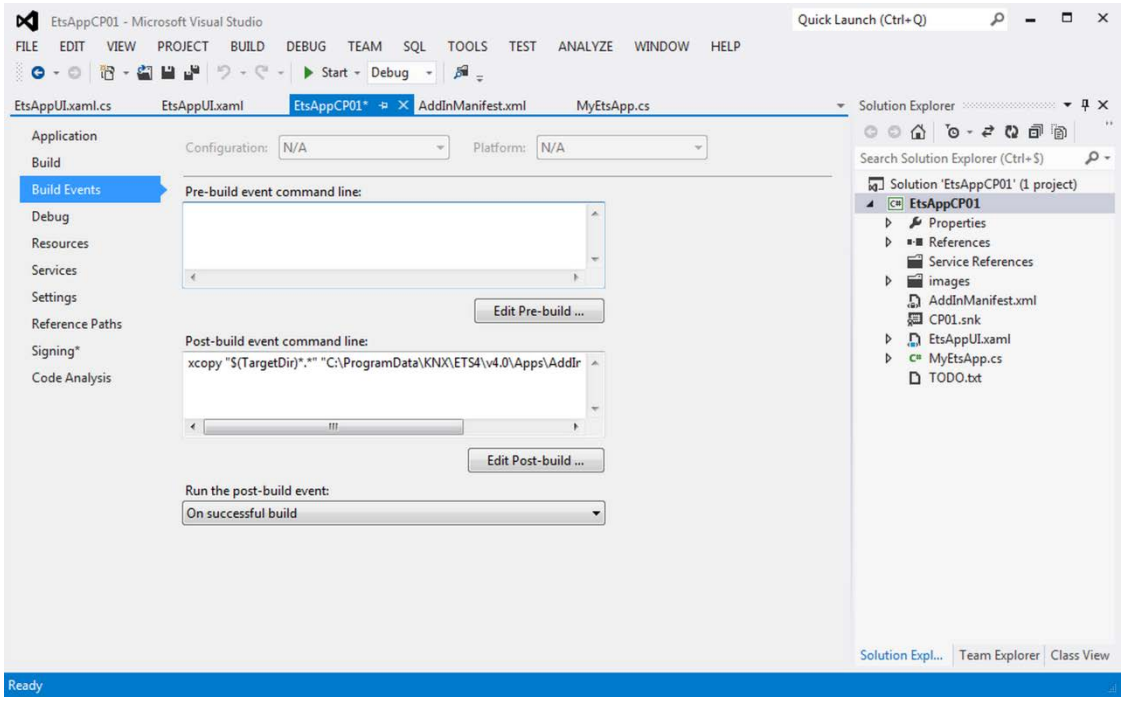
- Build the project and create a zip file including the actual help file, the image, the manifest file and the DLL. Rename the zip into .etsapp.
- Remark: ETS4 knows the .etsapp extension and will do the necessary if such file is provided.
- Upload it in the KNX Online Shop and check whether there are no validation errors; this is done automatically (“automatic validation”).
- Make sure there are no validation errors before going to the next step.

2.2.2 Step 2: Display general project information in the ETS4 status bar

- Prepare for debugging.
  - Open the Properties page for the project.
  - Click the property page “Debug”.
  - Modify the property “Start external program property” to “C:\Program Files\ETS4\ETS4.exe”.



- Click the page “Build events”.
- Modify the Post-build event command line property: xcopy "$(TargetDir)*.*" "C:\ProgramData\KNX\ETS4\v4.0\Apps\AddIns\M00FA-A000E\" /s/e/v/y.

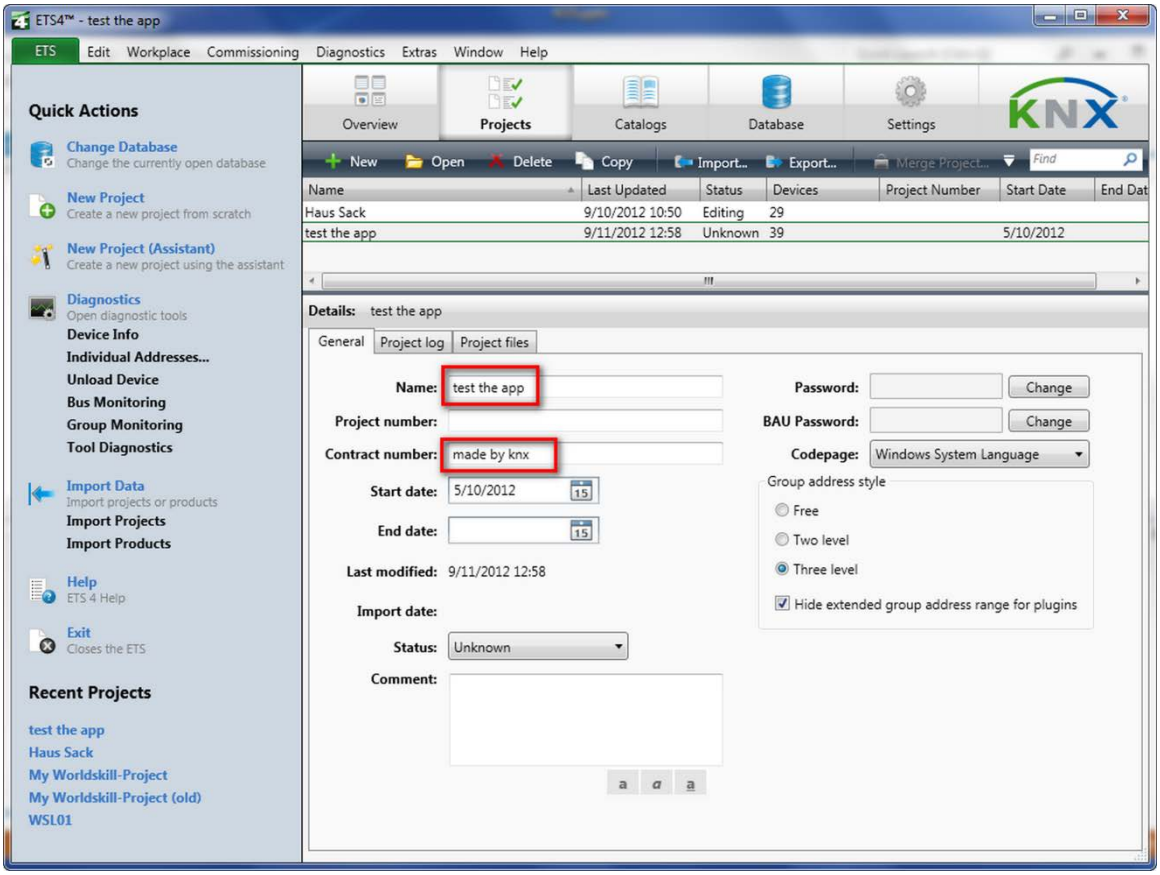


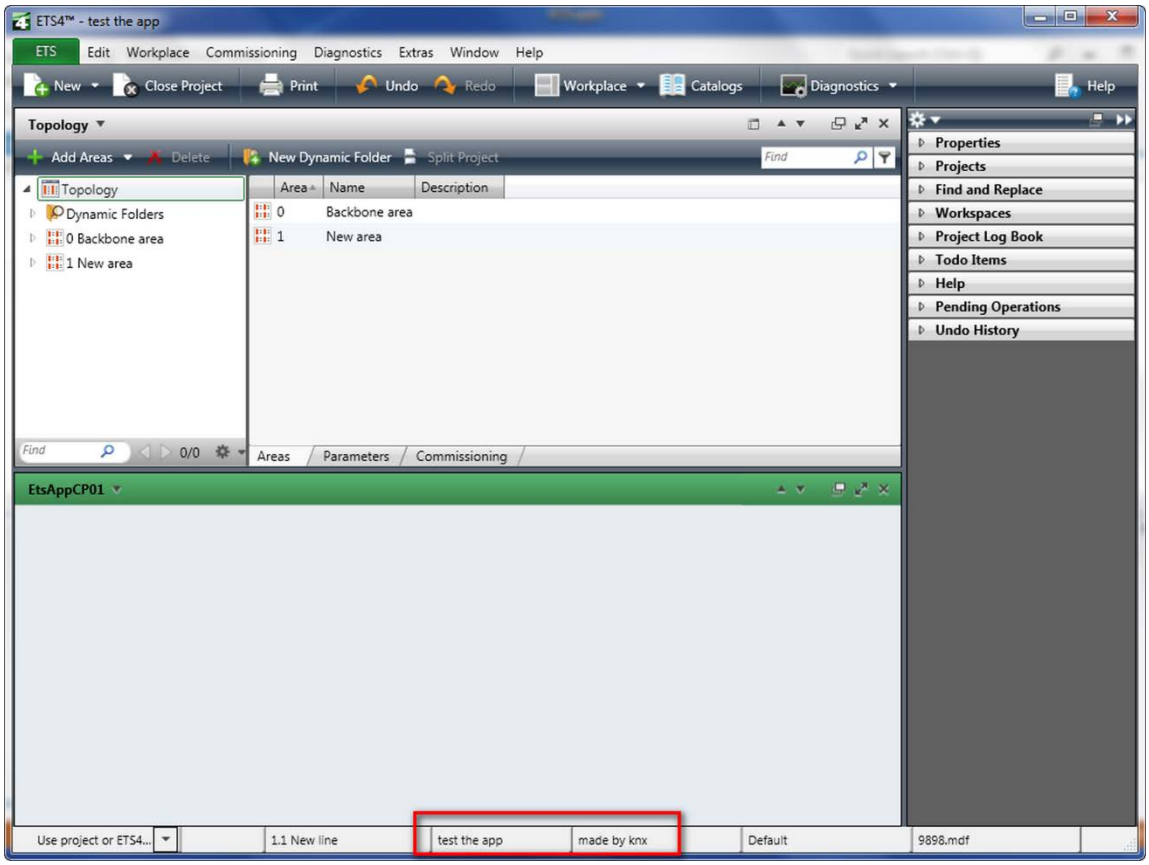
- Make the following changes in myEtsApp.cs.

```
public class MyEtsApp : IEts4AddIn, IDisposable
 {
        private IHostNotification myHost;
        private Project myProject;
        private EtsAppUI myUI;
…
```

```
public void Initialize(IInitializationContext initializationContext)
{
    myHost = initializationContext.HostNotification;
    myProject = initializationContext.Project;

    myHost.SetStatusbarText(myProject.Name, myProject.ContractNumber);
}
```

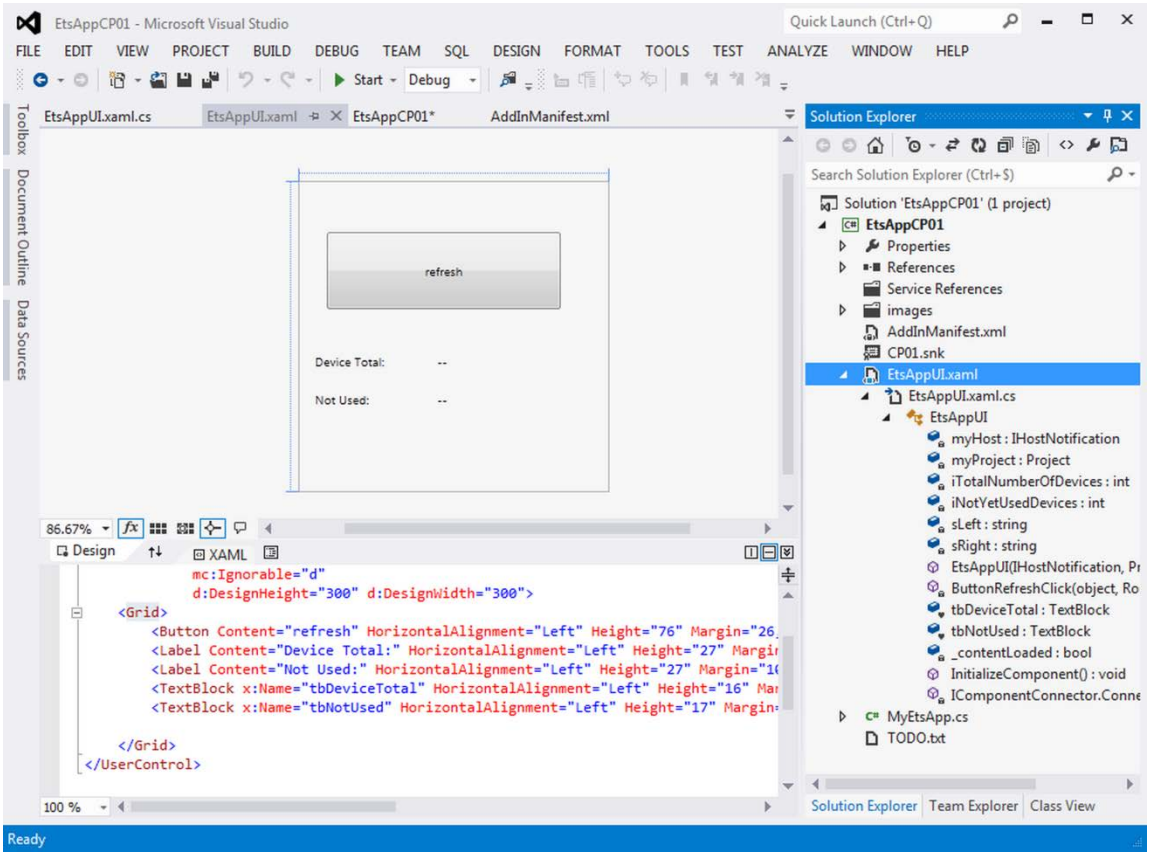
- Build, debug & test.





2.2.3 Step 3: Display project specific information in the App UI

- Add a user control to the project.
  - Right click the project in the solution explorer.
  - Click “Add”.
  - Click “User Control…”.
  - This will add an .xaml and .xaml.cs file to the project.
- Add a button, e.g. “refresh”.
- Add labels and text blocks, e.g. “Device Total” and “Not Used”.



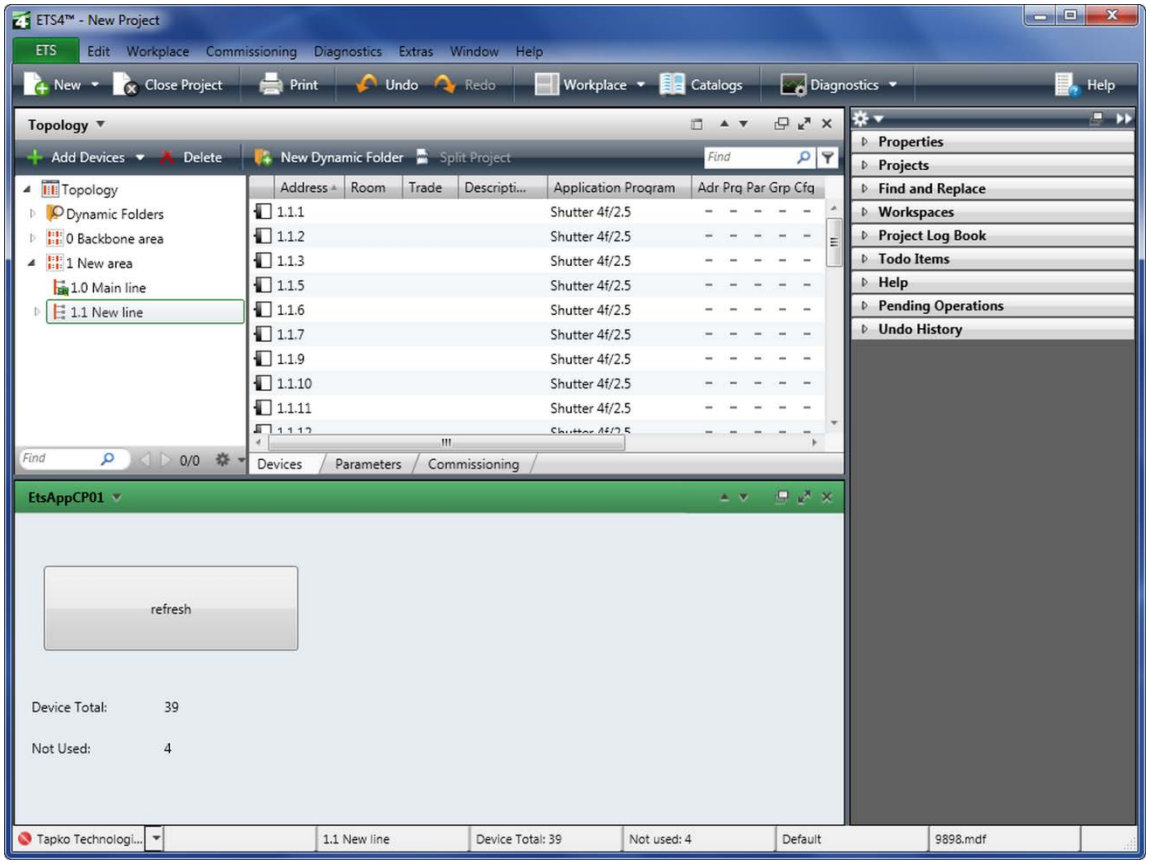
- Make the following changes in myEtsApp.cs.

```
public FrameworkElement GetAddInUI()
        {
                myUI = new EtsAppUI(myHost, myProject);

                return myUI;
        }

public void Initialize(IInitializationContext initializationContext)
        {
                myHost = initializationContext.HostNotification;
                myProject = initializationContext.Project;
        }
```

- Make the following changes in EtsAppUI.xaml.cs.

```
public partial class EtsAppUI : UserControl
{
	private IHostNotification myHost;
	private Project myProject;

	int iTotalNumberOfDevices;
	int iNotYetUsedDevices;
	String sLeft;
	String sRight;

	public EtsAppUI(IHostNotification thisHost, Project thisProject)
	{
		InitializeComponent();
		myHost = thisHost;
		myProject = thisProject;

		iTotalNumberOfDevices = myProject.DefaultInstallation.AllDevices.Count;
		iNotYetUsedDevices = myProject.DefaultInstallation.UnassignedDevices.Count;

		sLeft = String.Format("Device Total: {0}", iTotalNumberOfDevices);
		sRight = String.Format("Not used: {0}", iNotYetUsedDevices);

		myHost.SetStatusbarText(sLeft, sRight);

		sLeft = String.Format("{0}", iTotalNumberOfDevices);
		sRight = String.Format("{0}", iNotYetUsedDevices);

		tbDeviceTotal.Text = sLeft;
		tbNotUsed.Text = sRight;
	}

	private void ButtonRefreshClick(object sender, RoutedEventArgs e)
	{
		iTotalNumberOfDevices = myProject.DefaultInstallation.AllDevices.Count;
		iNotYetUsedDevices = myProject.DefaultInstallation.UnassignedDevices.Count;

		sLeft = String.Format("Device Total: {0}", iTotalNumberOfDevices);
		sRight = String.Format("Not used: {0}", iNotYetUsedDevices);

		myHost.SetStatusbarText(sLeft, sRight);

		sLeft = String.Format("{0}", iTotalNumberOfDevices);
		sRight = String.Format("{0}", iNotYetUsedDevices);

		tbDeviceTotal.Text = sLeft;
		tbNotUsed.Text = sRight;
		}
}
```

- Build, debug & test.



2.2.4 Step 4: Release

- Replace the dll in the .etsapp file.
- Upload it in the KNX Online Shop.
- Request manual validation after the automatic validation.