TABLE OF CONTENTS

1. WHAT IS REX?

1.1. What is REX?

The primary goal of REX is to help Revit API developers concentrate on essential development aspects when creating various add-ins for Revit, by providing support for typical, commonly used functionalities.

The outcome of this approach is:

- Acceleration of add-ins development
- Better consistency across add-ins
- Seamless integration within Revit

REX is a technology or framework supporting development of add-ins for Revit and making them consistent and aligned with the way Revit interacts with users.

The primary goals for the REX framework can be defined as follows:

- Provide higher level of API interaction with Revit
- Provide components and tools for typical functionalities
- Enable seamless add-in behavior within the Revit environment
- Enable easy add-in activation and registration
- Support for Autodesk Exchange for Autodesk Revit.

1.2. What is the REX SDK?

The REX SDK is a development environment for Rapid Application Development purposes that helps to create and activate add-ins based on the REX technology.

The core part of the REX SDK is implemented in the form of a Microsoft Visual Studio C# template. Using the template provided, you can quickly build an add-in that has a similar look & feel to Autodesk Revit Extensions.

The REX SDK is composed of:

- Project Template (C#)
  - UI definition
  - Interactions
  - Localization support
  - Support for Autodesk Exchange.
- Documentation
  - Getting Started
  - User manual and Design guidelines
  - API documentation
  - Samples

1.3. To whom is it addressed?

The advantages of this technology can be useful to all Revit API developers making add-ins for Revit, but it's most efficient for those whom the following aspects are applicable:

- Developing multiple add-ins
- Advanced UI creation for add-ins
- Create commercial add-ins for further distribution
- Developing links with other products

1.4. What are the most interesting features for API developers?

- C# project templates, ready to build, distribution and activation in Revit
- Common UI controls as EditBox, ComboBox,IndexLabel, …
- Components for HTML report generation
- Units conversions and unit based parameters display and editing consistently with Revit
- Automated class data serialization and storage within BIM models

1.5. What are the benefits for Revit users?

The primary advantages for end-users are:

- Consistent look & feel across various add-ins
- Consistent behavior aligned with Revit product (e.g. unit sensitive values edit and display)
- Could be used on previous developed add-ins for Revit to take advantages of *dialogs, controls, units, and other utilities.*

1.6. What is the difference between the Revit and the REX API?

The REX SDK does not provide any additional access to internal Revit functionality. The REX API extends the Revit API functionality with a set of new components.

1.7. What products can this technology be applied to?

The REX based approach is applicable for Revit Architecture, Revit Structure and Revit MEP products.

1.8. When are alternative approaches recommended?

The REX SDK provides a set of tools to develop External Commands on top of the Revit API. But there are other common structures for Revit API applications not supported by the SDK, including:

- Dynamic model updaters
- Events and event driven applications
- Customized Ribbon panels and controls

In addition, it is not possible to define custom Failure definitions from a REX SDK application as they must be defined in the startup of an external application. For applications which need to use any of these API features, the REX SDK would not be appropriate.

Another limitation of the REX SDK lies in the supported language: project templates and examples are provided only in C# and are not applicable for development in other .NET compatible languages.

1.9. How do I get started?

This document describes how to create a first and simple extension for Revit. The next stage will be to create a more complex one. For this, the pre-defined Pyramid Generator sample and tutorial will be useful to understand features exposed via the Extensions SDK and how to use them.

2. SYSTEM REQUIREMENTS

- MS Visual Studio 2015
- .NET Framework 4.5.2
- Autodesk Revit 2017.

3. CONFIGURATION

To properly configure Visual Studio 2015, Extensions templates should be copied to dedicated folders

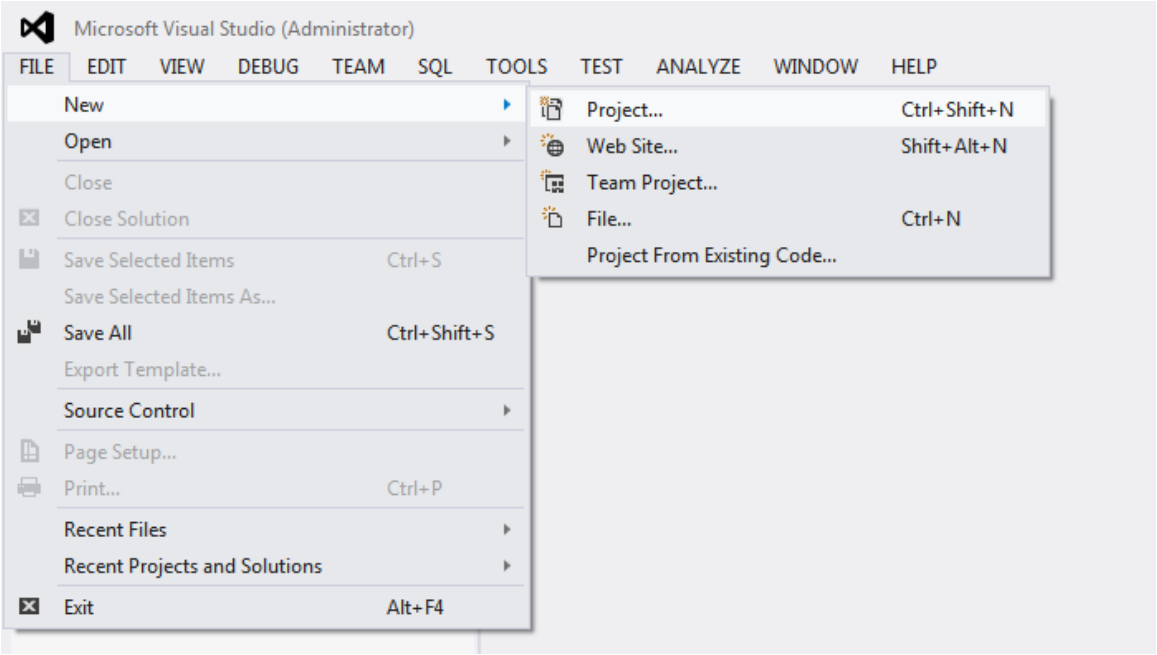
- The content of the folder "..\Software Development Kit\REX SDK\Visual Studio templates\Items\" should be copied to "C:\Users\<current user>\Documents\Visual Studio 2015\Templates\ItemTemplates\Visual C#\"

- The content of the folder "..\Software Development Kit\REX SDK\Visual Studio templates\Projects\" should be copied to "C:\Users\<current user>\Documents\Visual Studio 2015\Templates\ProjectTemplates\Visual C#

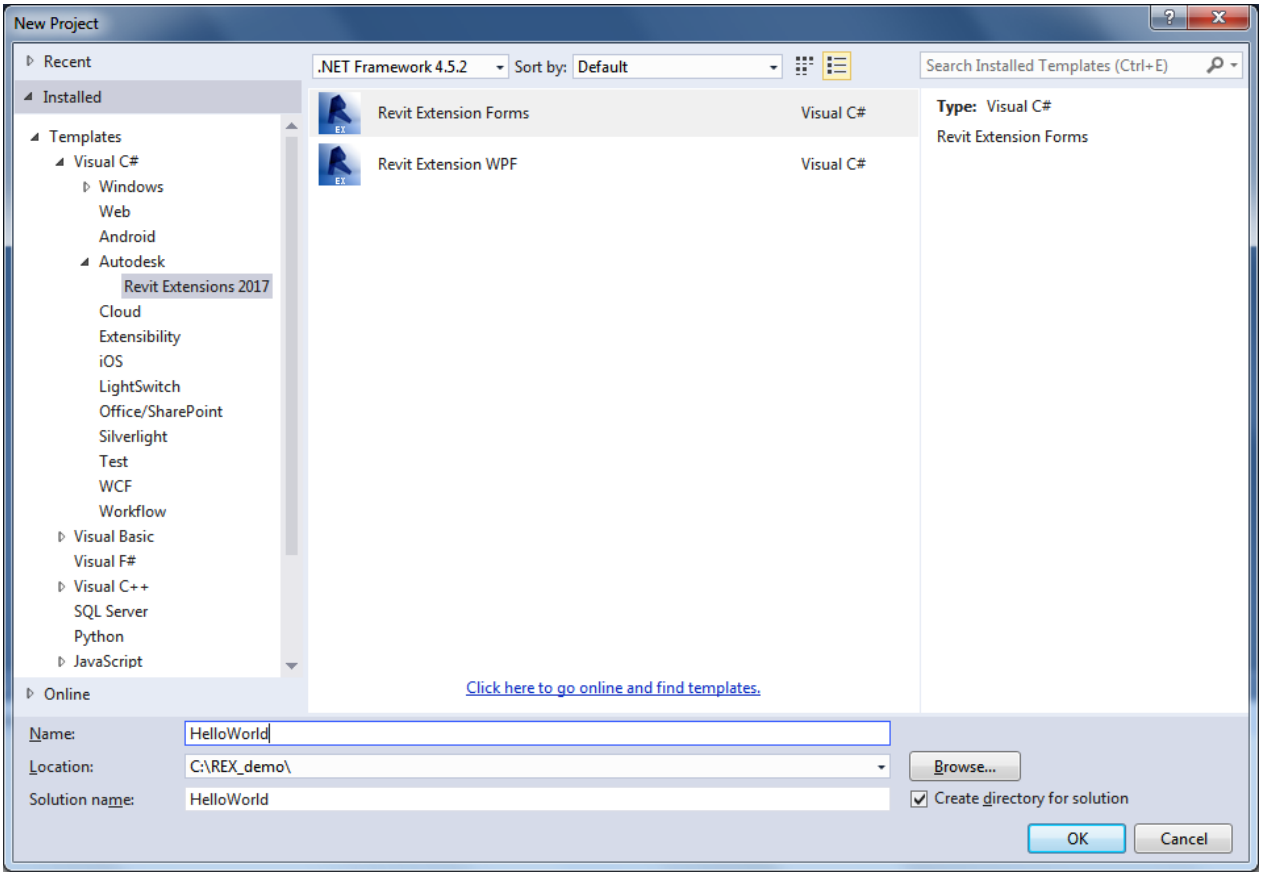
4. FIRST PROJECT CREATION

4.1. Visual Studio solution

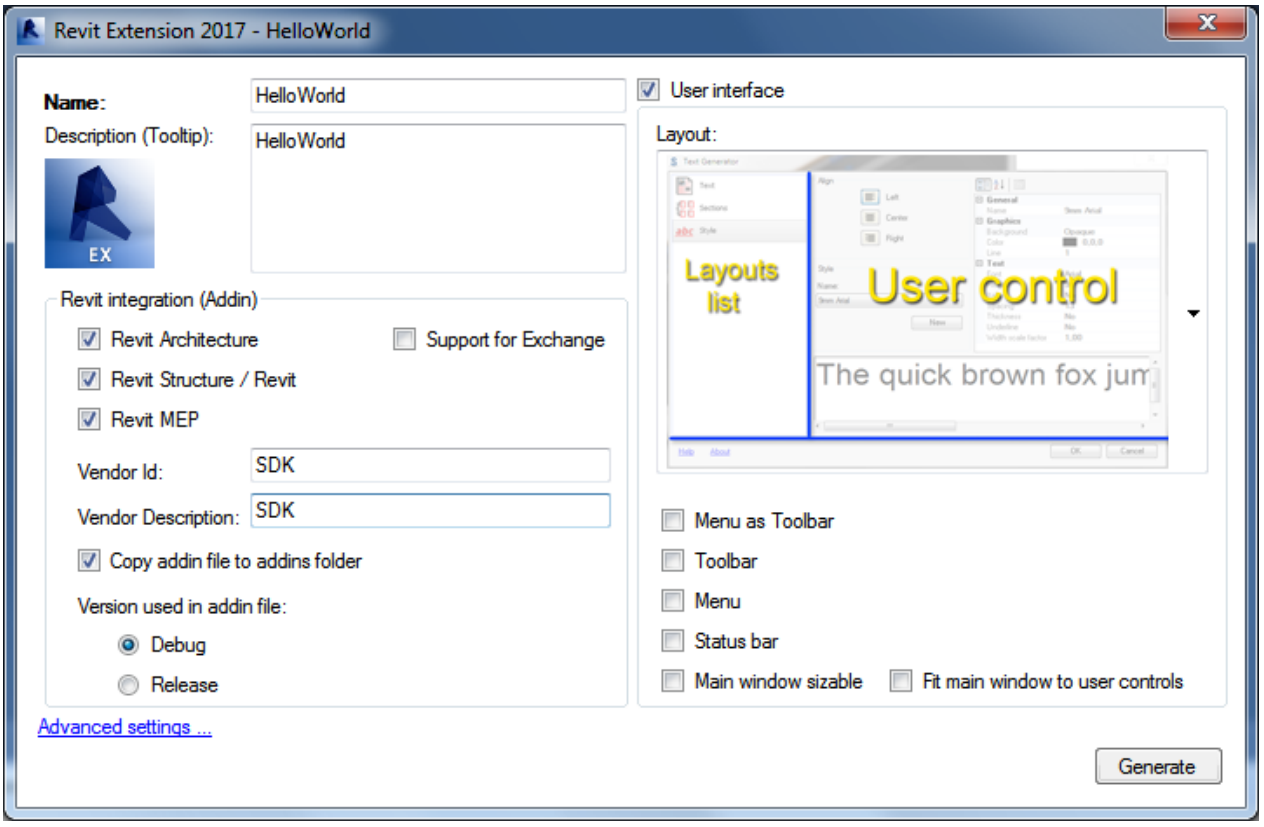
After starting visual studio, you need to choose Autodesk\ Revit Extensions Templates as a new project (**File / New / Project**).



The dialog below will appear.



At this stage, developer needs to specify the new extension name and the folder where he wants to save his source code.
After validation, a set of options could be defined:



Options available:

- Addin generation
- Revit Extensions with GUI or not and menu configuration
- Some advanced settings as the integration in Revit Extensions Ribbon and start-up project.
- Support for Autodesk Exchange.

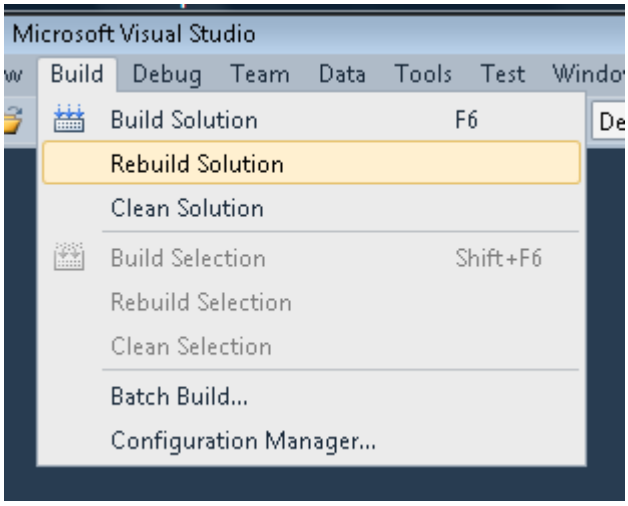
The Visual Studio solution will be created after clicking on the Generate button.

4.2. Development

To create the classical HelloWorld in Revit, the code of the method OnRun() in ExtensionsRevit.cs from Main\Revit directory is as following:

```
  public override void OnRun()
        {
            Autodesk.Revit.UI.TaskDialog.Show("My first Extension", "Hello
World");
        }
```

To build the solution and create binaries, we use the build menu from Visual studio.



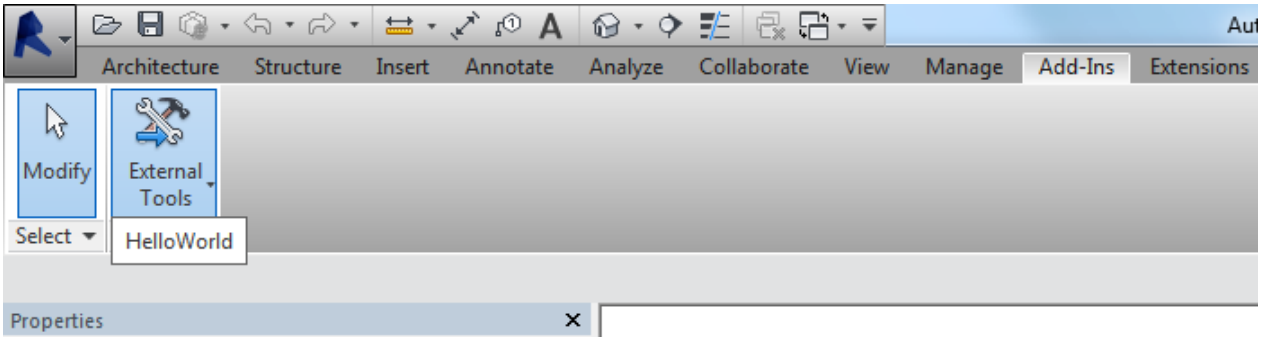
4.3. Extension installation

During compilation and using the post build, a Revit addin file has been generated inside this folder: C:\Users\<current user>\AppData\Roaming\Autodesk\Revit\Addins\2017\HelloWorld.addin.
If “Support for Exchange” is checked, new folder with binaries and Exchange configuration files is generated inside folder: C:\Users\<current user>\AppData\Roaming\Autodesk\ApplicationPlugins.

5. RUN HELLO WORLD EXTENSION

To run it:

- Launch Revit
- Go to addin tab
- Select Hello World form the external tools dropdown list



- A Revit TaskDialog should appear as follow