What is the REX SDK ?

### 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 a 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

### What is the REX SDK?

The REX SDK is a development environment for Rapid Application Development purposes that helps to create, deploy 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
  + Deployment
  + Microsoft Installation project
* Documentation
  + Getting Started
  + User manual and Design guidelines
  + API documentation
  + Samples

### 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

### 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

### 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.

### 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.

### What products can this technology be applied to?

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

**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.

### How do I get started?

The following steps will help to create a first REX based add-in for Revit:

- Make sure that Microsoft Visual Studio 2010 is installed

- Make sure that Revit 2012 is installed

- Open the Getting Started Manual and follow these steps:

- Create a project template

- Develop necessary code

- Build the application

- Open Revit and run the Add-in