**Migration Revit plug-in project to 2011 notes**

**Here is the steps to migrate Revit 2010 code to Revit 2011.**

Before migrating the code, please read what’s new section in RevitAPI.chm file to know what happened in the API. Then follow below steps.

1. Open the Revit code project in VS2008.
2. Remove the reference to old RevitAPI.dll
3. Reference to the new RevitAPI.dll and RevitAPIUI.dll, which are in Revit 2011 installation Program subfolder.
4. Replace old namespaces with the new namespace.
   1. For example:

//using Autodesk.Revit;

//using Autodesk.Revit.Elements;

//using Autodesk.Revit.Structural.Enums;

Using Autodesk.Revit.DB;

Using Autodesk.Revit.UI;

All namespace mappings are documented in ‘Revit 2011 API Namespace Remapping.xlsx’ file in SDK.

1. Add two external command attributes for each external command/Application.

[Autodesk.Revit.Attributes.Transaction(Autodesk.Revit.Attributes.TransactionMode.Automatic)]

Or [Autodesk.Revit.Attributes.Transaction(Autodesk.Revit.Attributes.TransactionMode.Manual)]

[Autodesk.Revit.Attributes.Regeneration(Autodesk.Revit.Attributes.RegenerationOption.Automatic)]

Or [Autodesk.Revit.Attributes.Regeneration(Autodesk.Revit.Attributes.RegenerationOption.Manual)]

1. Change the name space of IExternalCommand/IExternalApplication interface , ExternalCommandData class, Result enum to Autodesk.Revit.UI
2. Change the return value namespace of Result Execute method.
3. The way to get Document instance and Application instance changed.
4. UIApplication is introduced in API, to get database base level application, use below code.

ExternalCommandData.Application.Application

In Revit 2010 we get in this way:

ExternalCommandData.Application

1. Since UIDocument is introduced, to get database level document object use any one line below :

ExternalCommandData.Application.Application.Document

ExternalCommandData.Application.ActiveUIDocument.Document

In Revit 2010 we get in this way:

ExternalCommandData.Application.Document.

1. If project has element iteration process, need to migrate to use FilteredElementCollector class. Please refer to RevitAPI.chm and Revit 2011 API developer guide.pdf for information in detail.
2. Some classes were renamed, please read what’s new section in RevitAPI,chm for detailed information, and replace old name with the new class name.

**Other notes**

1. New ElementId() need argument . In Revit 2010 , no argument at all.
2. get\_Element(ElementId) needn’t ref. For ElementId is a class now.
3. TypeFilter change to ElementClassFilter. Creating instance change to use New ElementClassFilter
4. CategoryFilter becomes ElementCategoryFilter. Other element filter class are also renamed.
5. Get analytical model changes.

//AnalyticalModelWall anaWall = w.AnalyticalModel as AnalyticalModelWall;

Autodesk.Revit.DB.Structure.AnalyticalModel anaWall = w.GetAnalyticalModel();

1. ElementId doesn’t have Value property, change to IntegerValue. So replace with new property name.
2. When iterate the whole document element, we need to set a filter explicitly. If no filter was set at all, it will throw error.

FilteredElementCollector collector = new FilteredElementCollector(app.ActiveUIDocument.Document);

FilteredElementIterator iter = collector.OfClass(typeof(Element)).GetElementIterator();

while (iter.MoveNext())

{

}

1. If the command is with RegenerationOption.Manua mode, and the code changes model, we need to create transaction explicitly to update the model.

Transaction documentTransaction = new Transaction(commandData.Application.ActiveUIDocument.Document, "Document");

documentTransaction.Start();

……

documentTransaction.Commit();

1. Some other detailed items are not listed here, when you compile the project, their might be other errors. Please detect the possible reason from the error message and the API changes documentation.