The Building Coder Programming Forge, BIM and the Revit API le a region, curve or planar surface.") « Bounding Boxes Axis Alignment and Transformation I Main I Resx Language Management and APS DevCons » July 27, 2023 Export, gbXML and Python Tips Looking at several useful discussions on Python, handling DLLs, and various aspects of exporting to gbXML, FBX and MongoDB today: • DLL paradise in Python Follow @jeremytammik • Multiple gbXML export About - Topics - Index - Source • GbXML energy settings Automate FBX export with SendKeys Q~ Search • RFA export to MongoDB Subscribe to this blog's feed **DLL Paradise in Python** Jake of Ripcord Engineering shared several useful Revit API discussion forum solutions recently. Many thanks to Jake for his support! Categories One is a possible approach to handle DLL Hell using the Python subprocess module for .NET 360 2010 2011 2012 2013 2014 2015 2016 disentanglement without need for any IPC, e.g., for CPython and pyRevit: 2017 2018 2019 2020 2021 2022 2023 2024 2025 3dwc ACA ACC Accelerator Adva AI Algorithm Question: I need to use CPython via pyRevit to have access to libraries such as numpy and Analysis Apps AppStore APS Architecture ARX AU AU pandas. At the same time, I want to take advantage of pyRevit's capabilities such as forms etc. As 2008 AU 2009 AU 2010 AU 2011 Audio Automation far as I understood, I can't have both of these in a single script file. If I got this correctly, is there AVF Batch BIM Book BPA C++ ChatGPT Climate any way to do this? Climbing Clipper Cloud COM CompHound Content The numpy part is quite decoupled since it is meant to help me with the data exchange process cURL DA4R Data Access DataDays from other data sources; after that point, everything would be focused on Revit APIs. Debugging Deep Learning Deletion Deployment Desktop DevBlog DevCon DevDays DevTV **Answer:** I dealt with the same challenge a little while back. Please look at the pyRevit issue 1731 Dimensioning Discipline Diving DMU Docs DragDrop on Dynamo incompatibility: two versions of Same DLL for a short discussion on using Python DWF DWG DXF Dynamo Elasticsearch Element subprocess module for subprocess management in the Revit/pyRevit context. **Creation Element Relationships** While I am not a Revit API / Python / pyRevit expert I can report that subprocess worked well Events Exchange Export External F# Failure enough. Learning subprocess should be a productive use of time assuming the underlying Family FBX Filters Forge Forma Fun Fusion characteristics are a good match for your application. gbxml Geometry Getting Started GIF GIS Git Glue Group Hackathon HTML I18n Idling IFC **Response:** Thanks, Jake. I tried the same approach, and it also worked perfectly for my case. Insight360 Installation JavaScript Job Journal JSON Appreciate it. Labs Library Links LLM Logging M3U Mac Macro **Answer:** Thanks for giving it a go. And thanks for the feedback. Markdown Material Meetup Migration Mobile Modeless MongoDB MP3 Music NavisWorks News Multiple GbXML Export Node NoSQL OBJ Open Source Parameters Jake also helped answer the question on export of multiple gbXML models: Parts PDF Performance Philosophy Photo Physics Question: For my university thesis work I have to create a lot of different GBXML models (around Plugin Precision Print Prompt Properties Purge Python Q4R4 Rebar ReCap Regen Regex REST 18000). No way I can do that without code. This is what I came up with (I attached only a part of it; RevitLookup REX Ribbon RME Roadmap RST RTC FloorR, WallsR, RoofR are lists to set R value of corresponding elements): Ruby Sailing SAT Schedule SDK Samples ### Setting Energy Analysis parameters ### Security Selection Server Settings STL Storage Storm Sustainability SVG TED Template Testing Threejs opt=Analysis.EnergyAnalysisDetailModelOptions() Training Transaction Travel Units Unity Update opt.EnergyModelType=Analysis.EnergyModelType.BuildingElement User Interface Utilities va3c Vasari VB VDC opt.ExportMullions=False opt.IncludeShadingSurfaces=False View Viewer VSTA WCF WebGL Win32 Wizard opt.SimplifyCurtainSystems=True Workset WPF XAML opt.Tier=Analysis.EnergyAnalysisDetailModelTier.SecondLevelBoundaries ### loop over all R-value combinations and create models ### **Recent Posts** t=Transaction(doc, "R change") c=Transaction(doc, "model creation") Revit Gen AI: MCP, RAG and Vibe Docs, Local APS LLM and Parallel Task for i in range(len(FloorR)): for j in range(len(WallsR)): Orchestration for k in range(len(RoofR)): DevCon, API Docs and Fabrication Part Service t.Start() Floor.Set(FloorR[i]/0.3048) #R-value change for floor Access UIApplication and Bounding Box on Wall.Set(WallsR[j]/0.3048)#R-value change for Walls Sheet Roof.Set(RoofR[k]/0.3048)#R-value change for roof t.Commit() Unit Testing and More Serious Matters t.Dispose() Tools for Extensible Storage and OAuth Auth0 c.Start() API Context and Extensible Storage in DA4R model=Analysis.EnergyAnalysisDetailModel.Create(doc, opt) model.TransformModel() Access to UIApplication, Tags and LLM API GBopt=GBXMLExportOptions() Support GBopt.ExportEnergyModelType=ExportEnergyModelType.BuildingElement doc.Export("C:\Users\Mиша\Desktop\ASD","0"+","+str(0.2/FloorR[i])+","+str(0.3/WallsR[j])+","+ Wall Layer Voodoo and Prompt Optimisation |str(0.3/RoofR[k]), GBopt) c.Commit() LLM Prompting, RAG Ingestion and New **Projects** This creates models, but I ran into a problem I don't fully understand: as the process continues, it slows down and stops at about 170-175 created models. Apparently, something is taking up the memory. I tried calling doc.Delete(model) at the end of each for loop, but that didn't help **Archives** either. March 2025 What could be a solution? February 2025 **Answer:** The behaviour you describe is completely expected and as designed. January 2025 Revit is an end user product designed to be driven by a human being. Human beings are not November 2024 expected to sit down and create 18000 models in one sitting. I suggest you implement an external executable that drives Revit using the code you shared above and monitors progress as you October 2024 export results from the models you create. September 2024 Whenever Revit starts slowing down, take note of how far you got in processing, kill the process, August 2024 restart Revit and continue from where you left off. This is a common approach to July 2024 programmatically drive processes in batch mode that were not designed for it. You can also search The Building Coder for further hints on batch processing Revit documents. Alternatively, June 2024 you could generate your 18000 models online using APS and DA4R. May 2024 Furthermore, based on the code snippet provided, it appears that only R-values are manipulated and not the underlying model geometry. If that's the case, it might be best to use Revit to export a More... single gbXML seed file. Then, iterate over the desired seed file parameters (like R-value) in an environment like Python which is excellent for large scale text operations. Two utilities that would help with the route described above: • XmlNotepad – to build familiarity with gbXML structure and mechanization • xgbxml – Python library for gbXML parsing and manipulation Export gbXML - Settings General Details Parameter Value **Building Type** School or University Copenhagen, Denmark Location Ground Plane Level 00 Export Category Rooms Complex **Export Complexity** Include Thermal Properties Project Phase **New Construction** Sliver Space Tolerance 500.0 6 Save Settings Cancel **GbXML Energy Settings** Jake points to the same solution to answer another question as well, on gbXml export using energy settings: **Question:** Sorry to revive a thread which has been solved more than one year ago, however the solution provided is not working for me as I am programming in Python. I am using pyRevit to program functions in Python. When I use the code presented above, I get the following error: EnergyAnalysisDetailModelOptions.ExportMullions = False Traceback (most recent call last): File "<stdin>", line 1, in <module> AttributeError: static property 'ExportMullions' of 'EnergyAnalysisDetailModelOptions' can only be assigned to through a type, not an instance If I understand this correctly, I am having a problem due to the type of variable in my code. However, Python does not allow the declaration of variables. How can I make the statement to become a type and not an instance? Any idea how I can get past this issue without moving on to another language? **Answer:** A nice example of Python EnergyAnalysisDetailModelOptions administration is discussed in the export of multiple gbXML models. The relevant code snippet is this: ### Setting Energy Analysis parameters ### opt=Analysis.EnergyAnalysisDetailModelOptions() opt.EnergyModelType=Analysis.EnergyModelType.BuildingElement opt.ExportMullions=False opt.IncludeShadingSurfaces=False opt.SimplifyCurtainSystems=True opt.Tier=Analysis.EnergyAnalysisDetailModelTier.SecondLevelBoundaries Automate FBX Export with SendKeys We already shared a C# solution to handle a Revit dialogue using Idling, DialogBoxShowing and **SendKeys** to implement the TwinMotion dynamic link export FBX automatically. Now Onur Er cleaned it up further in his updated answer: Question: I want to export FBX using TwinMotion Dynamic Link. I would like to export FBX files from many Revit files. How I can use PostCommand and then handle the Windows forms on the export panel? **Answer:** Thank you for sharing your solution. It saved me unbelievable amount of time, maybe days or weeks. Thank you VERY VERY MUCH!!! I cleaned the code and made it more readable in case someone needs it. My own Revit plugin calls this Twinmotion macro automatically after Revit starts up like this: using System.Threading.Tasks; using Autodesk.Revit.UI; using System.Windows.Forms; using Autodesk.Revit.UI.Events; namespace YourNamespaceHere public class Class2 : IExternalApplication UIControlledApplication UIControlledApplication; public Result OnStartup(UIControlledApplication Application) UIControlledApplication = Application; UIControlledApplication.Idling += Application_Idling; return Result.Succeeded; public Result OnShutdown(UIControlledApplication Application) => Result.Succeeded; void Application_Idling(object Sender, IdlingEventArgs E) UIControlledApplication.Idling -= Application_Idling; var UIApplication = (UIApplication)Sender; MyMacro(UIApplication); //TaskDialog.Show("Application_Idling", Sender.GetType().FullName); void OnDialogBoxShowing(object Sender, DialogBoxShowingEventArgs Args) => ((TaskDialogShowing EventArgs)Args).OverrideResult((int)TaskDialogResult.Ok); static async void RunCommands(UIApplication UIapp, RevitCommandId Id_Addin) UIapp.PostCommand(Id_Addin); await Task.Delay(400); SendKeys.Send("{ENTER}"); await Task.Delay(400); SendKeys.Send("{ENTER}"); await Task.Delay(400); SendKeys.Send("{ENTER}"); await Task.Delay(400); SendKeys.Send("{ESCAPE}"); await Task.Delay(400); SendKeys.Send("{ESCAPE}"); void MyMacro(UIApplication UIapp) try var Name = "CustomCtrl_%CustomCtrl_%Twinmotion 2020%Twinmotion Direct Link%ExportButton"; var Id_Addin = RevitCommandId.LookupCommandId(Name); if (Id_Addin != null) UIapp.DialogBoxShowing += OnDialogBoxShowing; RunCommands(UIapp, Id_Addin); catch TaskDialog.Show("Test", "error"); finally UIapp.DialogBoxShowing -= OnDialogBoxShowing; Thank you, Onur Er! RFA Export to MongoDB To wrap up, Eduardo Lalo Ibarra of Mexico City shares one of his favourite classes built with #VSC and #MongoDB to facilitate the export of data from Revit families: The class implementation is encoded in the attached image files on LinkedIn:

using System; using System.Collections.Generic; using System.Linq; using System.Text; using System.Threading.Tasks; namespace exportcloudMDB /// Door class with get/set autoproperties /// </summary> public class Door public string _id { get; set; }
public string FamilyType { get; set; }
public string Mark { get; set; }
public string DoorFinish { get; set; } using System; using System.Collections.Generic; using System.Linq; using System.Text; using System.Threading.Tasks; using RestSharp; using System.Net; public class GoorAFI /// HTTP access constant to toggle /// between local and cloud server /// </summary> public static bool useCloudServer = false; /// (summary)
/// Base url for local testing and cloud url for production /// </wramary>
const string baseUrlLocal = "http://localhost:8980/";
const string baseUrlCloud = "https://mongorevit.herokuapp.com/"; /// Base url for local testing and cloud url for production /// </summary> public static string RestAPIBaseUrl get { return useCloudServer ? besetrlCloud : besetrltocal; } /// GET 350M data from /// the specified mongoOB collection. /// (/summary) sublic static List@cor> Get(string collectionHame) RestClient client = new RestClient(RestAPIBaseUrl);
RestRequest request = new RestRequest("/api" + "/" + collectionName, INestResponsed.istdoor>> response = client.Executed.istdoor>>(request); /// Batch POST 150N document data into /// (/summary) public static HttpStatusCode PostBatch(out string content, out string string collectionName, List(Door> doorOata) RestClient client = new RestClient(RestAPIBaseU+1); RestRequest request = new RestRequest("/apl" + "/" + collectionName + "/" "batch", Method.POST); request.RequestFormat = DataFormat.Json; request.AddBody(doorData); IRestResponse response = client.Execute(request);
content = response.Content;
errorMessage = response.ErrorMessage; return response.StatusCode; using System; using System.Collections.Generic; using System.Linq; using System.Text; using System.Threading.Tasks; using Autodesk.Revit.DB;

/// Implements the Revit add-in interface IExternalCommand

//Get the Current Session / Project from Revit

//Get the Current Document from the Current Session Document doc - ulapp.ActiveUIDocument.Document

Result result = ExportBatch(collector, ref message)

public Result Execute(ExternalCommandData commandData, ref string message,

TaskDialog.Show("Door Data Export", "Door Data successfuly

TankDialog.Show("Door Data Export", "Something went wrong...");

public static Result ExportBatch(FilteredElementCollector doors, ref string

/// </summary> [Transaction(TransactionMode.ReadOnly)] public class ExportCommand : TEsternalCommand

//Get all doors from project FilteredElementCollector collector = new FilteredElementCollector(doc).OfClass(typeof(FamilyInsta .OfCategory(BuiltInCategory.OST_Doors);

if(result -- Result.Succeeded)

/// <param name="doors"></param> /// <param name="message"></param>

HttpStatusCode statusCode;

using System; using System.Collections.Generic; using System.Linq; using System.Text; using System.Threading.Tasks;

namespace exportcloudMDG

foreach(tlement element in doors)

List<Door> doorData = new List<Door>();

doorData.Add(new DoorData(element));

/// Implements the Revit add-In interface IExternalCommand /// c/summaryo [Transaction(TransactionMode.Manual)]

//Get the Current Session / Project from Revit

//REST request to GET door data LIst<Door> doors = DoorAPI.Get("doors"); //Set Door finish values using unique

foreach(Door door in doors)

doorFinish.Set(dFinish);

I share the construction of the class. I will give myself some time to share the

Parameter doorFinish = element.get_Parameter(BuiltInParameter.Door_Finish);

return Result.Succeeded;

Eduardo also provides it as a PDF, from which I extracted a text file:

whole process.

mongodb_export.pdf

Here is his useful list of assets:

MongoDB Documents

Revit API docs

Revit SDK

repository

0 Comments

Many thanks, Eduardo!

• Visual Studio Community 2022

My First Revit Plug-in Overview

Create account in GitHub

Tags: akn_include, Jeremy Tammik, Revit API

Start the discussion...

A Privacy

topics

index

source

about

LOG IN WITH

Share

⊠ Subscribe

home

The Building Coder

Privacy settings

archives

mongodb_export.txt

//Get the Current Document from the Current Session

public Result Execute(ExternalCommandData commandData, ref string message,

using(Transaction trans = new Transaction(doc, "Import Door Data"))

string uId = door._id; Element element = doc.GetElement(uId); string dFinish = door.DoorFinish;

TaskDialog.Show("Import Door Data", "Door Data successfuly imported");

• The Builder Coder Visual Studio Revit Add-in Templates – recommendation: clone the Posted on July 27, 2023 at 11:00 in Analysis, Export, FBX, gbXML, MongoDB, Python, Win32 I Permalink

> OR SIGN UP WITH DISQUS (?) Name Best Newest Be the first to comment. **DISQUS** Do Not Sell My Data

> > subscribe

code

forum

Login ▼

Oldest

devguide

3dwc

Submit