1. Install Blender <https://blenderbim.org/docs/users/installation.html> and download BlenderBim addon (blender-4.0.0-windows-x64 and blenderbim-231104-py310-win)
2. Add addon BlenderBim in blender

Blender Preferences 
Themes 
VI ewport 
Lights 
Editing 
Animation 
d-ons 
Inp ut 
Navigation 
Keym ap 
System 
Save & Load 
File Paths 
Anin 
Enabled Add-ons Only 
Import-export: Atomic Blender PDB/XYZ 
System: Blender ID authentication 
System: BlenderBIM 
p blender 

1. Edit environment variables in windows and set these two paths as system variables under Path:

C:\Users\{YOUR\_USER}\AppData\Roaming\BlenderFoundation\X.XX\

C:\Program Files\Blender Foundation\Blender X.XX

Make sure to change the red colloured parts so they correspond to your case. You can have a look here on how to do it:

<https://www.imatest.com/docs/editing-system-environment-variables/#Windows>

In order to ensure that you have done it correctly open a new command line window and run ‘blender –help’. If that doesn’t return error it means you have configured it correctly.

**Modifications in blenderBim code:**

**C:\Users\{user}\AppData\Roaming\Blender Foundation\Blender\4.0**

**Operator.py (scripts\addons\blenderbim\bim\module\tester\operator.py)**

            if props.generate\_html\_report:

                engine = ifctester.reporter.Html(specs)

                engine.report()

                engine.to\_file(output)

                webbrowser.open("file://" + output)

            if props.generate\_html\_report:

                engine = ifctester.reporter.Html(specs)

                engine.report()

                engine.to\_file(output)

                # webbrowser.open("file://" + output)

**Operator.py (scripts\addons\blenderbim\bim\module\tester\operator.py)**

        with tempfile.TemporaryDirectory() as dirpath:

            start = time.time()

            output = os.path.join(dirpath, "{}.html".format(props.specs))

            ifc = tool.Ifc.get()

            if props.should\_load\_from\_memory and tool.Ifc.get():

                ifc = tool.Ifc.get()

            else:

                ifc = ifcopenshell.open(props.ifc\_file)

    def execute(self, context):

        props = context.scene.IfcTesterProperties

        with tempfile.TemporaryDirectory() as dirpath:

            start = time.time()

            output = os.path.join(dirpath, "{}.json".format(props.specs))

            ifc = tool.Ifc.get()

            if props.should\_load\_from\_memory and tool.Ifc.get():

                ifc = tool.Ifc.get()

            else:

                ifc = ifcopenshell.open(props.ifc\_file)

**Operator.py (scripts\addons\blenderbim\bim\module\tester\operator.py)**

            if props.generate\_html\_report:

                engine = ifctester.reporter.Html(specs)

                # engine = ifctester.reporter.Json(specs)

                engine.report()

                engine.to\_file(output)

                # webbrowser.open("file://" + output)

            if props.generate\_html\_report:

                # engine = ifctester.reporter.Html(specs)

                engine = ifctester.reporter.Json(specs)

                engine.report()

                engine.to\_file(output)

                # webbrowser.open("file://" + output)

**facet.py (scripts\addons\blenderbim\libs\site\packages\ifctester\facet.py)**

        self.applicability\_templates = [

            "Elements with {name} data of {value} in the dataset {propertySet}",

            "Elements with {name} data in the dataset {propertySet}",

        ]

        self.requirement\_templates = [

            "{name} data shall be {value} and in the dataset {propertySet}",

            "{name} data shall be provided in the dataset {propertySet}",

        ]

        self.applicability\_templates = [

            "Elements with {name} data of {value} in the dataset {propertySet}",

            "Elements with {name} data in the dataset {propertySet}",

        ]

        self.requirement\_templates = [

            "{name}",

            "{name} data shall be provided in the dataset {propertySet}",

        ]

**Information on how-to run a script via command line:**

You can run your own script by calling blender and it’s addon blenderbim functioncalities by running the command bellow:

blender --background --addons blenderbim --python "C:\{FOLDER}\script.py"

The documentation from the command line arguments is here: <https://docs.blender.org/manual/en/latest/advanced/command_line/arguments.html>

<https://docs.blender.org/manual/en/latest/advanced/command_line/render.html>