



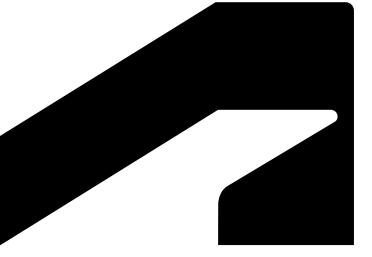
AUTODESK Platform Services

Design Automation API for Inventor

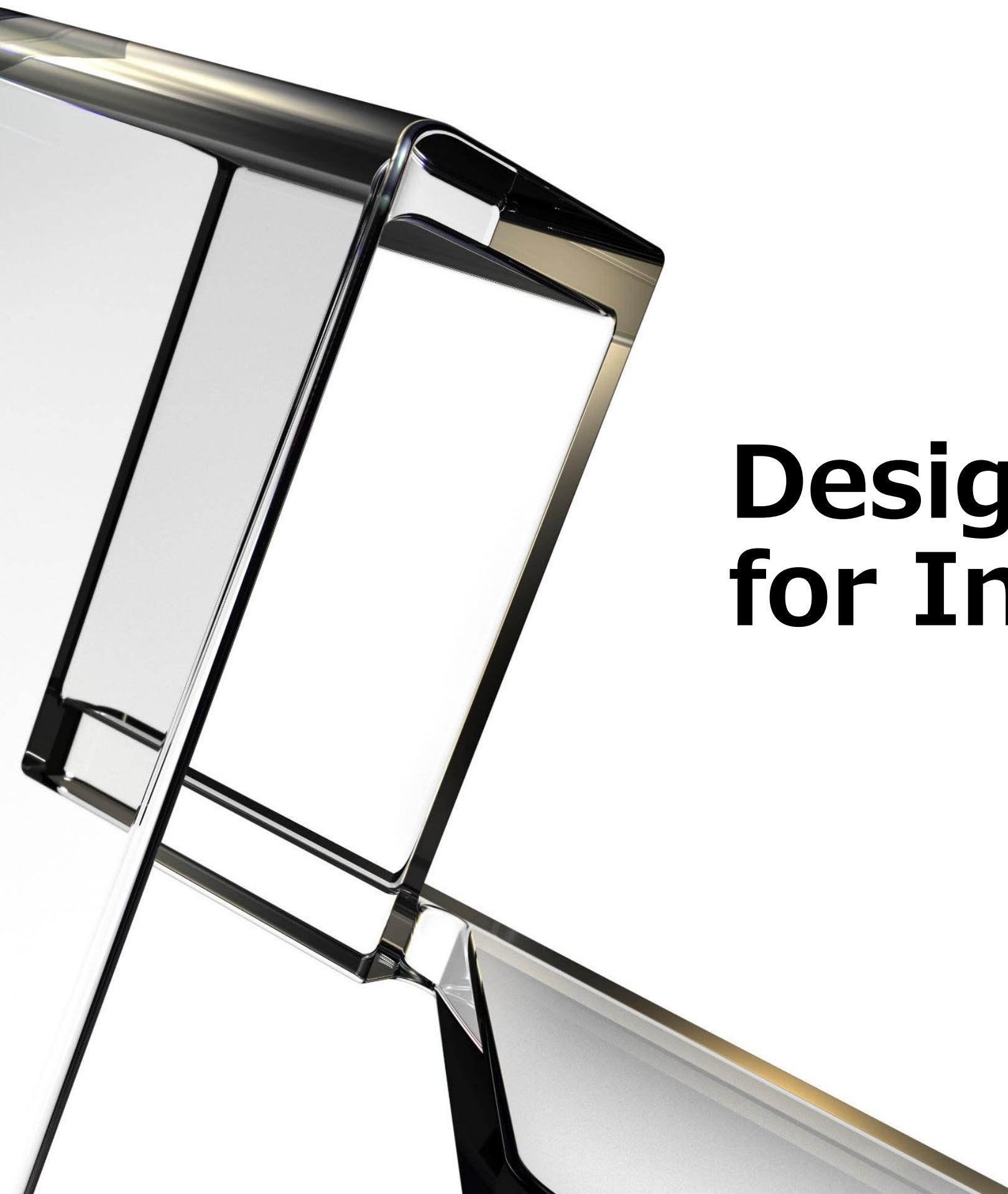
アップデート

加藤 丈博

Developer Advocacy & Support、Autodesk Developer Network

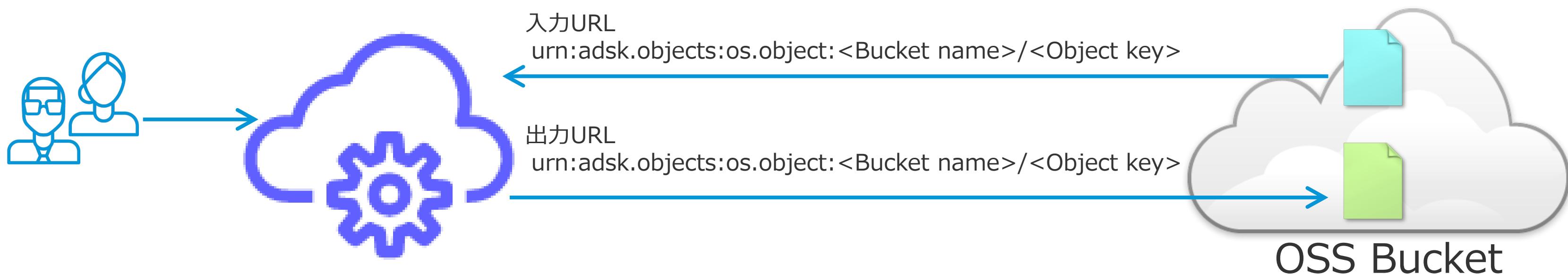


Design Automation API for Inventor アップデート



Inventor コアエンジンでのDirect-to-S3 対応

- Design Automation コアエンジンのファイル入出力にOSS Bucketを利用している場合の、URL指定を簡素化。
 - WorkItemで“urn:adsk.objects:os.object:<Bucket name>/<Object key>”形式で入力/出力 URLを指定
 - マルチパートアップロードをデフォルトでサポート
 - アクセストークンの期限延長にも対応

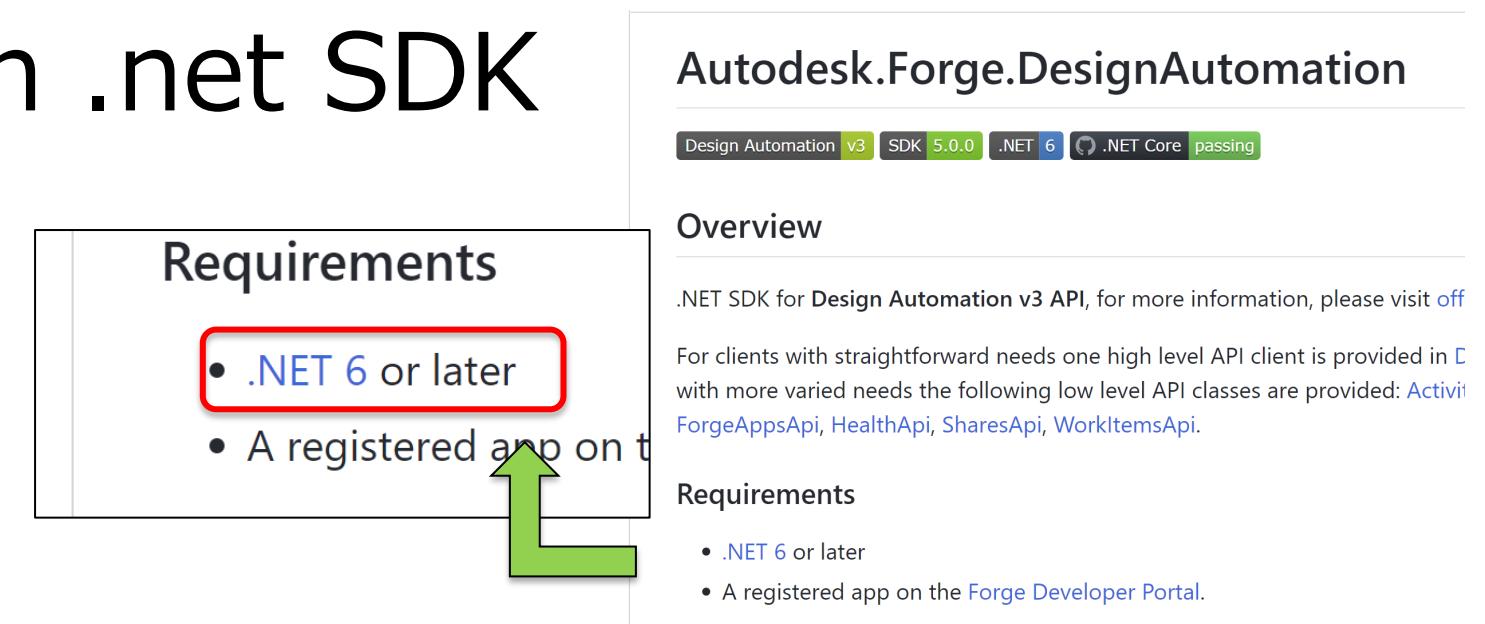


.net、node.js SDKでのDirect-to-S3 対応

- 既存のSDKを拡張する形での暫定対応
 - Direct-to-S3 に対応した公式 SDK を提供する取り組みは現在進行中
 - ✓ .net：“Autodesk.Forge” NuGetパッケージ version 1.9.7以降に更新
 - ✓ node.js：“forge-apis” npmパッケージ 0.9.4以降に更新

参考：Visual Studio 2019でDesign Automation .net SDKを利用する方法について

- 最新版Design Automation .net SDKは.net 6.0以降に対応



The screenshot shows the 'Requirements' section of the Autodesk Forge DesignAutomation NuGet package page. It lists two items: '.NET 6 or later' and 'A registered app on the Forge Developer Portal'. A red box highlights the first item, '.NET 6 or later'. A green arrow points from this highlighted requirement down to the corresponding requirement in the 'Requirements' section of the Visual Studio 2019 NuGet package page.

Autodesk.Forge.DesignAutomation

Design Automation v3 SDK 5.0.0 .NET 6 .NET Core passing

Overview

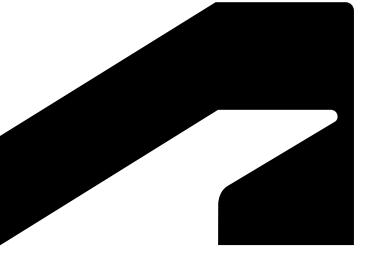
.NET SDK for Design Automation v3 API, for more information, please visit [off](#)

For clients with straightforward needs one high level API client is provided in [C](#) with more varied needs the following low level API classes are provided: [Activit](#), [ForgeAppsApi](#), [HealthApi](#), [SharesApi](#), [WorkItemsApi](#).

Requirements

- .NET 6 or later
- A registered app on the [Forge Developer Portal](#).

- Visual Studio 2019は.net 5.0までをサポート
⇒Visual Studio 2019のプロジェクトでは最新版Design Automation .net SDK をNuGetパッケージインストールが出来ない状況



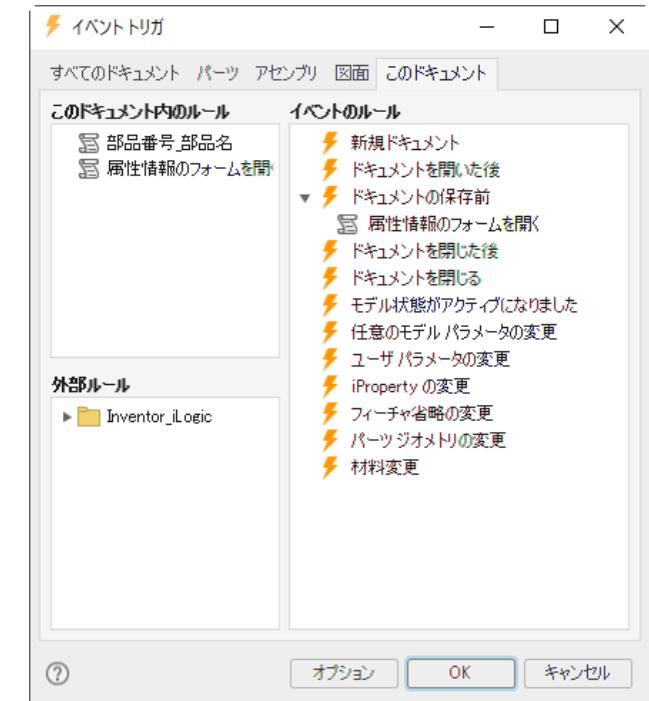
Inventor iLogicを用いた Design Automation の活用



Design AutomationでのiLogicの活用パターン

1. Inventorのイベントに設定したiLogicを実行

⇒既存のiLogicフォームを用いたコンフィギュレータ
をWebアプリ化 等



2. ドキュメント内のiLogicルールを外部から実行

⇒定型処理を行うiLogicをWebアプリから実行 等

3. コアエンジンの入力パラメータにiLogicルール (ソースコード) を指定して実行

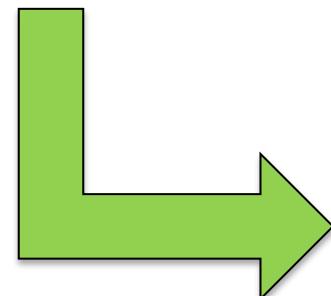
⇒AppBundle無しでのWorkItemの実行。PoC開発 等

Inventorコアエンジンの/s オプション

Inventorコアエンジンの/s オプション

InventorCoreConsole.exe

- ✓ Design Automation for Inventor用のコアエンジン
- ✓ クライアント版Inventorには同梱されていない
- ✓ 起動引数で指定されたAppBundleをロードし、AppBundleのメソッド（のどちらか）を実行
 - Run(Document doc) : 所定起動引数のみの場合
 - RunWithArguments(Document doc, NameValueMap map) : 所定引数以外を指定した場合。所定以外の引数はNameValueMapにキー"-1", "-2"…で順に格納される
- ✓ InventorCoreConsole.exeの所定起動引数
 - /al : ロードするAppBundleのパス
 - /i : ロードするドキュメントのパス
 - /s : 実行するスクリプトファイルへのパス



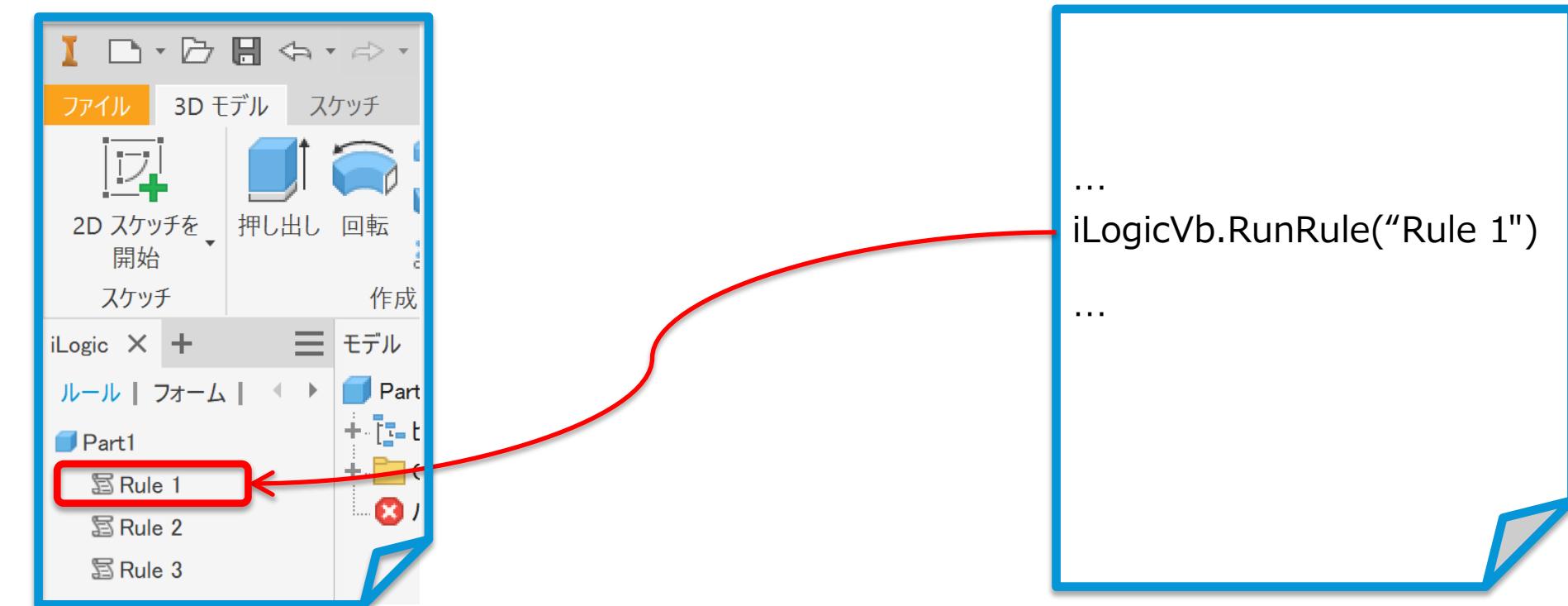
✓ InventorCoreConsole.exeの所定起動引数

- /al : ロードするAppBundleのパス
- /i : ロードするドキュメントのパス
- /s : 実行するスクリプトファイルへのパス

Inventorファイル内のiLogicの実行

- iLogicVb.RunRule(name As String) 関数
 - ✓ name 実行するiLogic ルール名

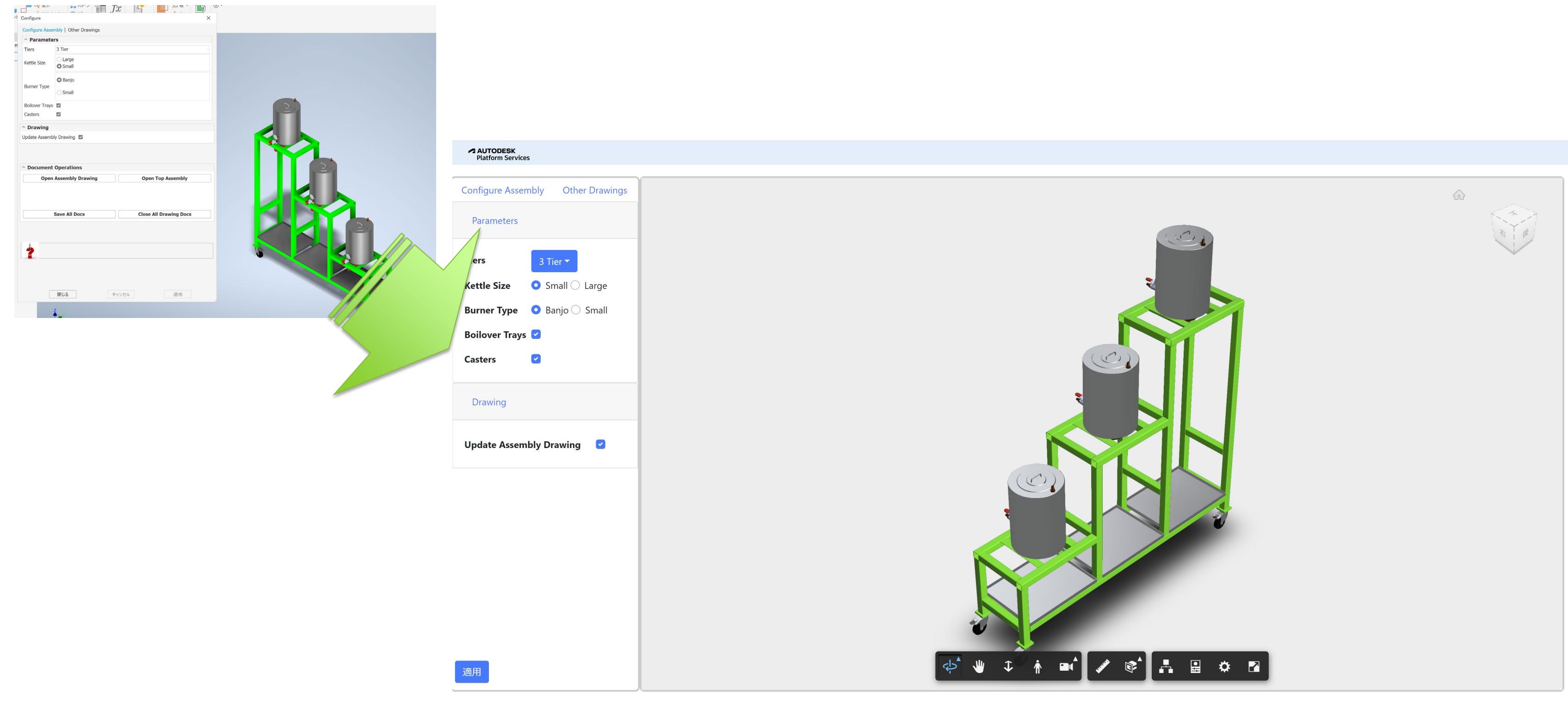
InventorCoreConsole.exe /i <*inventor file*> /s <*script file*>



Design AutomationでiLogicを利用する際の注意点

1. ThisApplicationではなくThisServerを
ThisApplication.ActiveDocumentではなく、
ThisDoc.Documentを使用する
2. UI表示を行う処理は記述しない
⇒Try～Catchのメッセージボックスにも注意
3. ログ出力はLoggerではなくTraceを使用する
⇒ TraceはWorkItemの実行ログに出力される

Design AutomationでのiLogicを活用した コンフィギュレータサンプル



コンフィギュレータアプリとAPS APIの対応

GUI

• • •



Viewer



パラメータ反映

• • •



Design
Automation

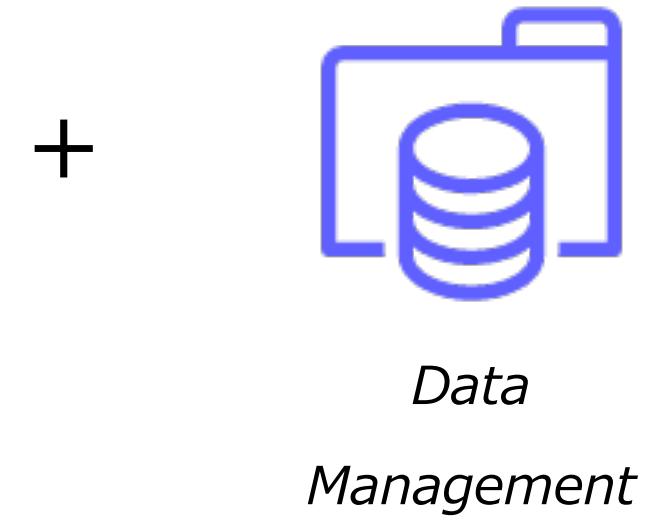


SVF変換

• • •



Model
Derivative



パラメータ変更⇒ビューワー反映のレスポンスについて

- Design Automation

クラウド上で実行され、パラメータに応じた元データと、View (SVF) の生成を都度行っている

⇒あらかじめパラメータのパターンに応じたView (SVF) を作成しておく

コンフィギュレータアプリとAPS APIの対応

GUI

• • •



Viewer



パラメータ反映

• • •



*Design
Automation*



*Data
Management*

SVF変換

• • •



**Inventor プラグイン
(VBA、Addin等をLocalで実行)**

Model

Derivative



Data

Management

Inventor SVF アドインによるSVFの出力

Autodesk App Manager ウィンドウヘルプ

General Declarations

```
'@Folder("ApplicationProject")
Option Explicit

Public Sub ExprtAllSVFs()
```

Public Function exportSVF(path As String)
 Dim oAddin As TranslatorAddIn
 Set oAddin = ThisApplication.ApplicationAddIns.ItemById("{C200B99B-B7DD-4114-A5E9-6557AB5ED8EC}")

 Dim oContext As TranslationContext
 Set oContext = ThisApplication.TransientObjects.CreateTranslationContext()

 oContext.Type = IOMechanismEnum.kFileBrowseIOMechanism

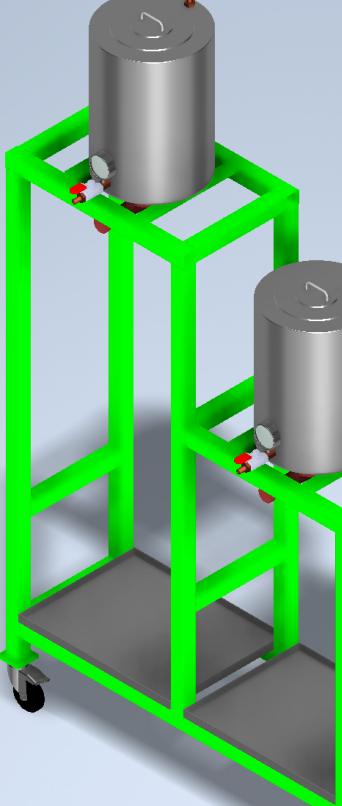
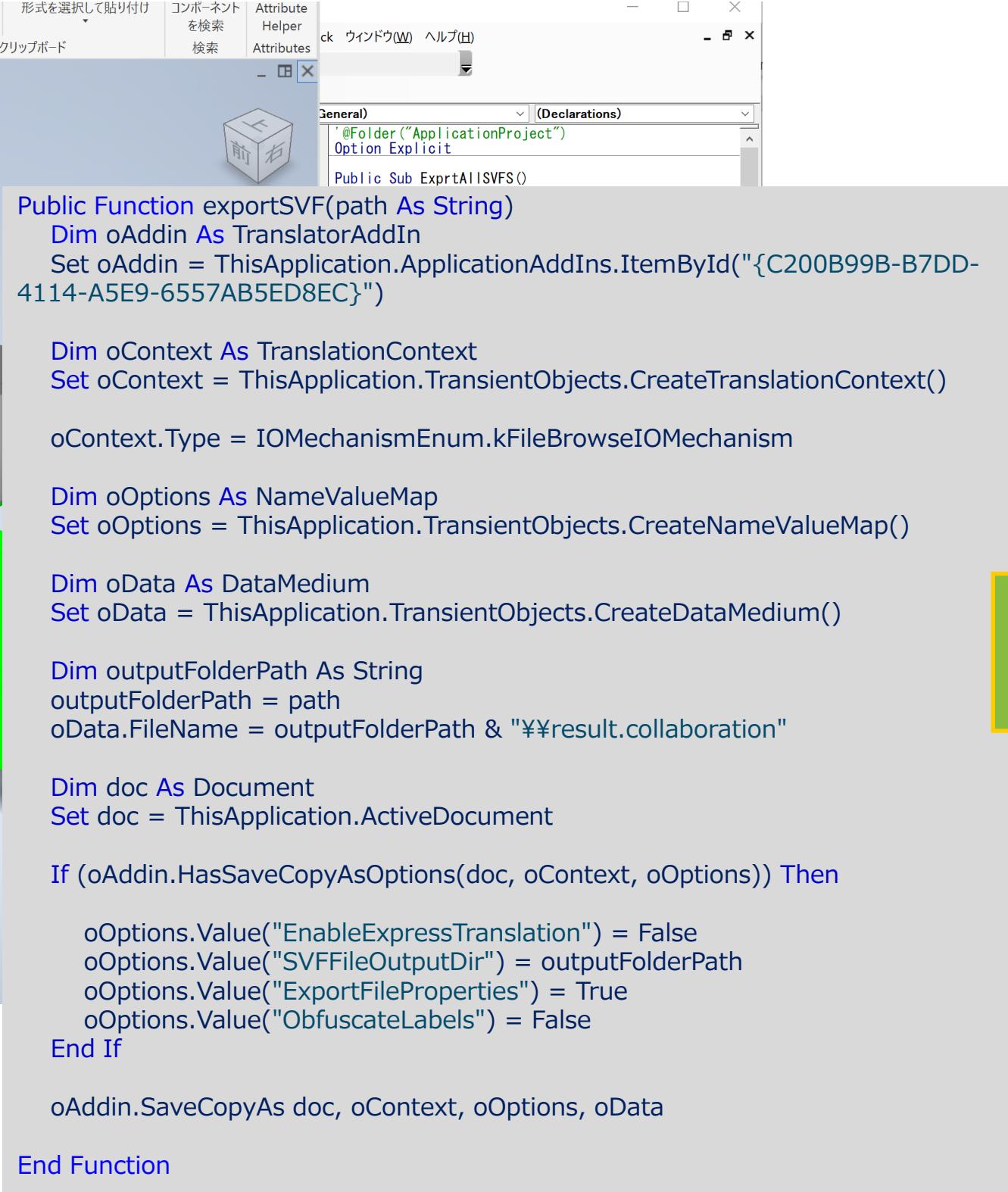
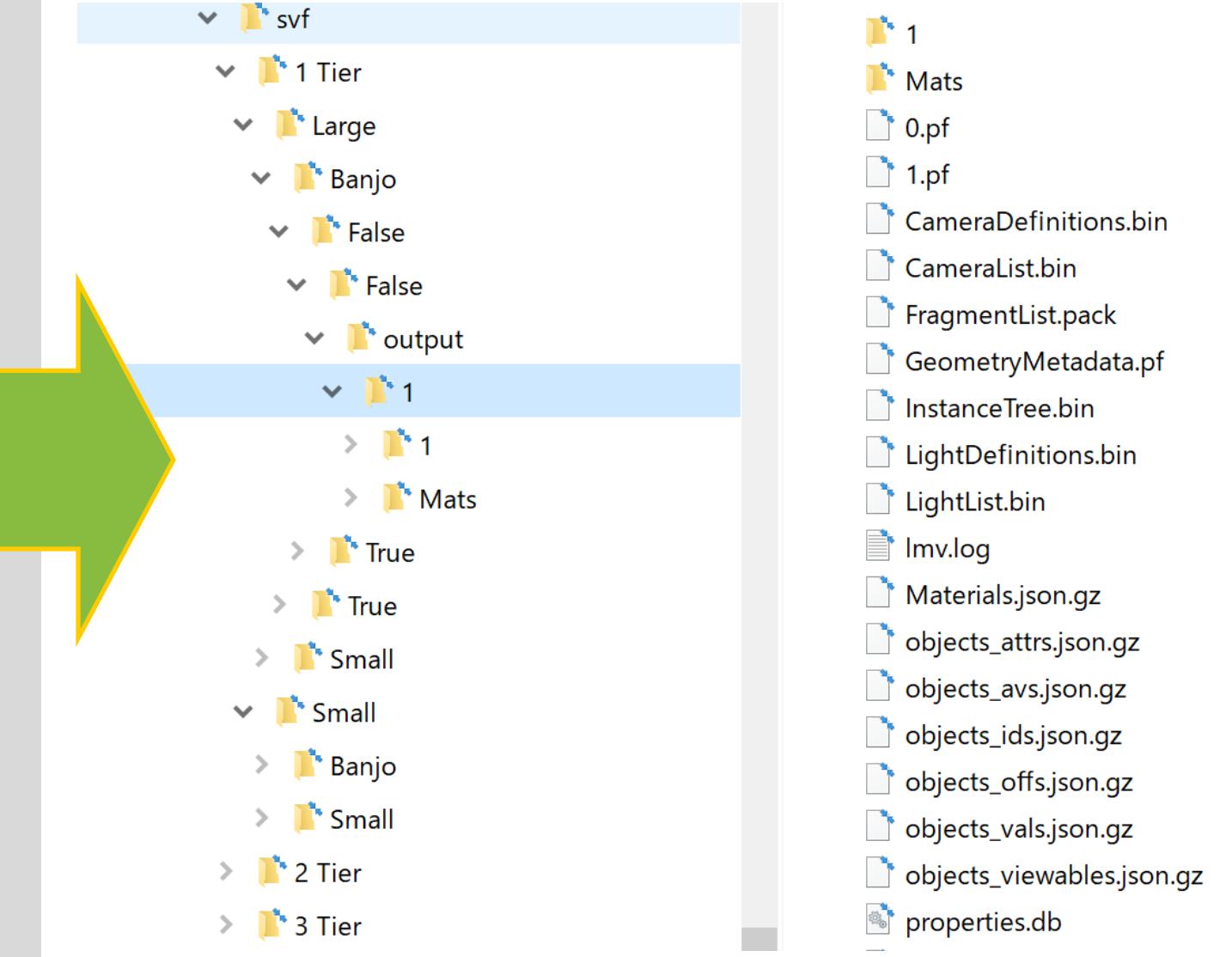
 Dim oOptions As NameValueMap
 Set oOptions = ThisApplication.TransientObjects.CreateNameValueMap()

 Dim oData As DataMedium
 Set oData = ThisApplication.TransientObjects.CreateDataMedium()

 Dim outputFolderPath As String
 outputFolderPath = path
 oData.FileName = outputFolderPath & "\result.collaboration"

 Dim doc As Document
 Set doc = ThisApplication.ActiveDocument

 If (oAddin.HasSaveCopyAsOptions(doc, oContext, oOptions)) Then
 oOptions.Value("EnableExpressTranslation") = False
 oOptions.Value("SVFFileOutputDir") = outputFolderPath
 oOptions.Value("ExportFileProperties") = True
 oOptions.Value("ObfuscateLabels") = False
 End If

 oAddin.SaveCopyAs doc, oContext, oOptions, oData
End Function




A large green arrow points from the code window to the file explorer window, indicating the flow of the process.

- svf
 - 1 Tier
 - Large
 - Banjo
 - False
 - False
 - output
 - 1
 - 1
 - Mats
 - True
 - True
 - Small
 - Small
 - Small
 - Banjo
 - Small
 - 2 Tier
 - 3 Tier
- 1
- Mats
- 0.pf
- 1.pf
- CameraDefinitions.bin
- CameraList.bin
- FragmentList.pack
- GeometryMetadata.pf
- InstanceTree.bin
- LightDefinitions.bin
- LightList.bin
- Imv.log
- Materials.json.gz
- objects_attrs.json.gz
- objects_avs.json.gz
- objects_ids.json.gz
- objects_offs.json.gz
- objects_vals.json.gz
- objects_viewables.json.gz
- properties.db

Forge ViewerでLocal ServerのSVFの表示

```
function launchViewer(index) {  
  var options = {  
    env: 'Local'  
};
```

envに'Local'を指定

```
Autodesk.Viewing.Initializer(options, () => {  
  viewer3d = new  
Autodesk.Viewing.GuiViewer3D(document.getElementById('forgeViewer3d'));  
viewer3d.start();
```

...

```
//svfURL is local server .svf file's URL, e.g. './svf/1//output/1/result.svf'  
viewer3d.loadModel(svfURL, {}, (model) => {});  
};
```

loadModelでローカルサーバ
のsvfファイルのURLを指定
(Encode不要)

.net Coreでコンテンツの許可

Startup.cs

```
// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.  
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)  
{  
    app.UseStaticFiles(new StaticFileOptions  
    {  
        ServeUnknownFileTypes = true,  
        DefaultContentType = "image/png"  
    });  
  
    app.UseRouting();  
  
    app.UseEndpoints(routes =>
```



.svfファイルは、.Net coreの既定のコンテンツタイプに含まれていないため、デフォルトではブロックされてしまうため、ブロックされないための対応。

Design Automation for Inventorでアセンブリ、図面ファイルを利用する場合

- Inventor の図面、アセンブリは複数の Inventor ファイルで構成されている

⇒親ファイル内に、参照しているファイルのパス情報を持つておらず、Inventorで開く際にパス情報を元に子ファイルをロード

- Design Automation の処理は、クラウド上にアップロードしたデータを、Design Automation 実行環境（サーバ）にダウンロードして実行する

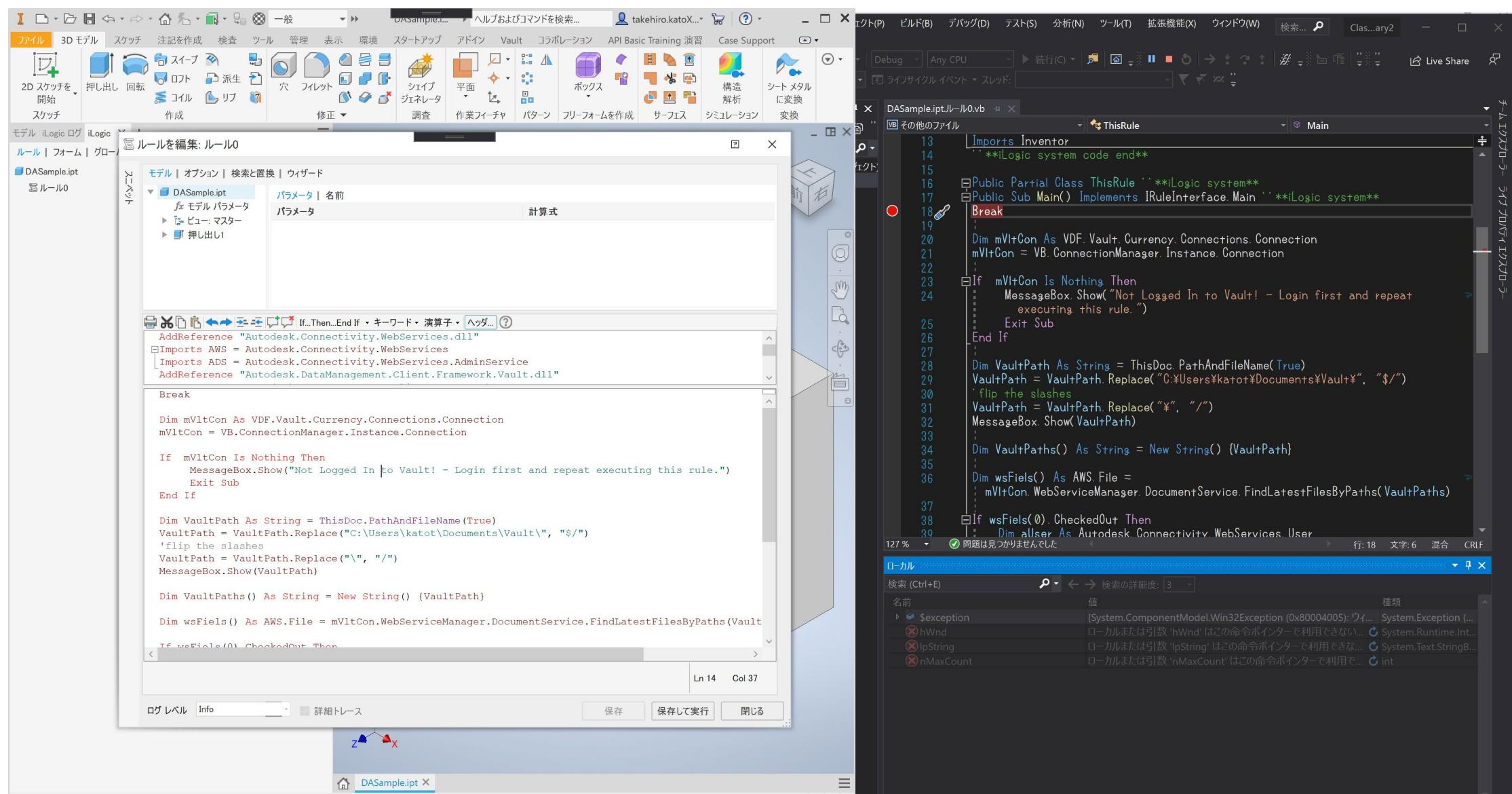
⇒アセンブリ、図面の場合、参照する全てのファイルをフォルダ配下に配置し、ZIP でまとめてアップロード

Inventor の Pack and Go を活用



参考：Visual StudioによるiLogicのデバッグ実行

Visual Studio 2019から、Inventorのプロセスにアタッチして、iLogicをデバッグ実行



- Break Pointの設定
- ステップ実行
- ローカル変数の値の参照等が利用可能



Autodesk and the Autodesk logo are registered trademarks or trademarks of Autodesk, Inc., and/or its subsidiaries and/or affiliates in the USA and/or other countries. All other brand names, product names, or trademarks belong to their respective holders. Autodesk reserves the right to alter product offerings and specifications at any time without notice, and is not responsible for typographical or graphical errors that may appear in this document.

© 2022 Autodesk. All rights reserved.