



Forge Online

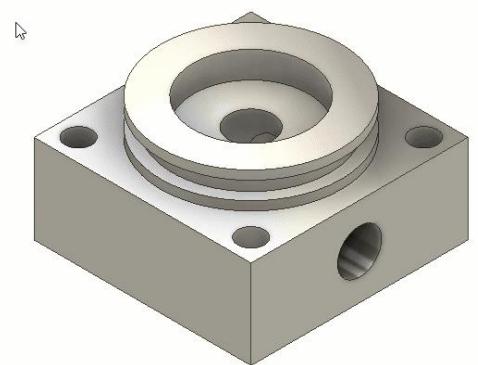
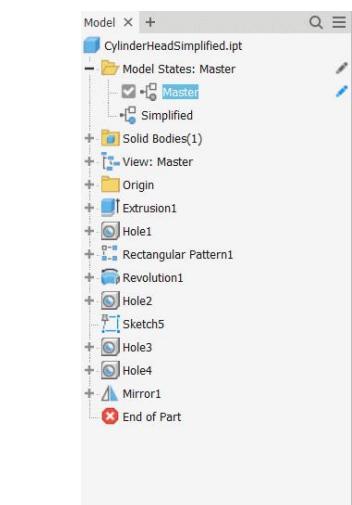
# Inventor 2022新機能 モデル状態とForge API

オートデスク 株式会社

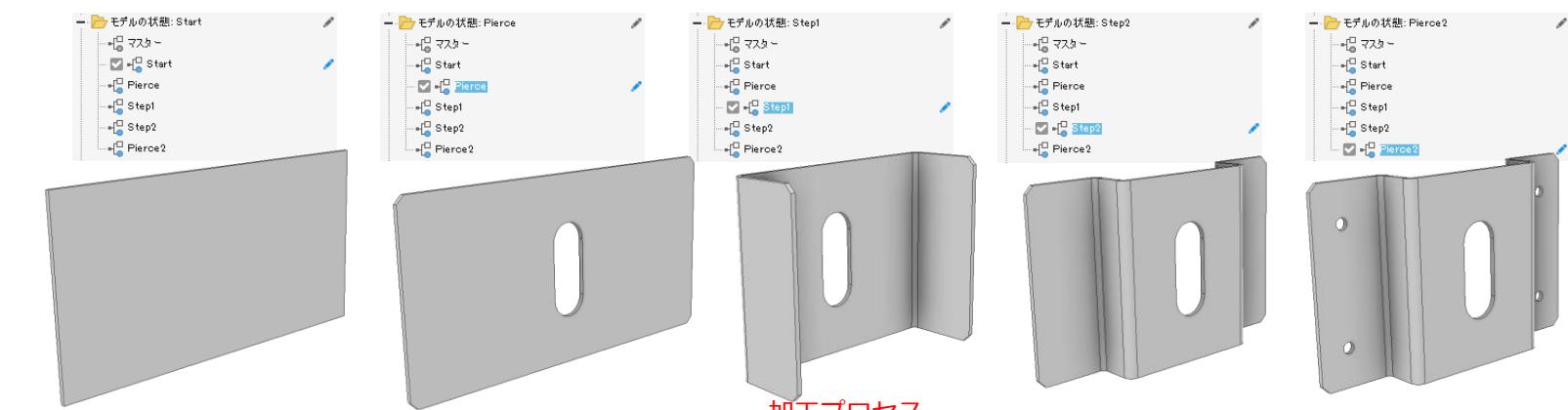
# Design Automation for Inventorによるモデル状態のViewer(SVF)ファイル出力

# Inventor 2022 新機能 モデル状態

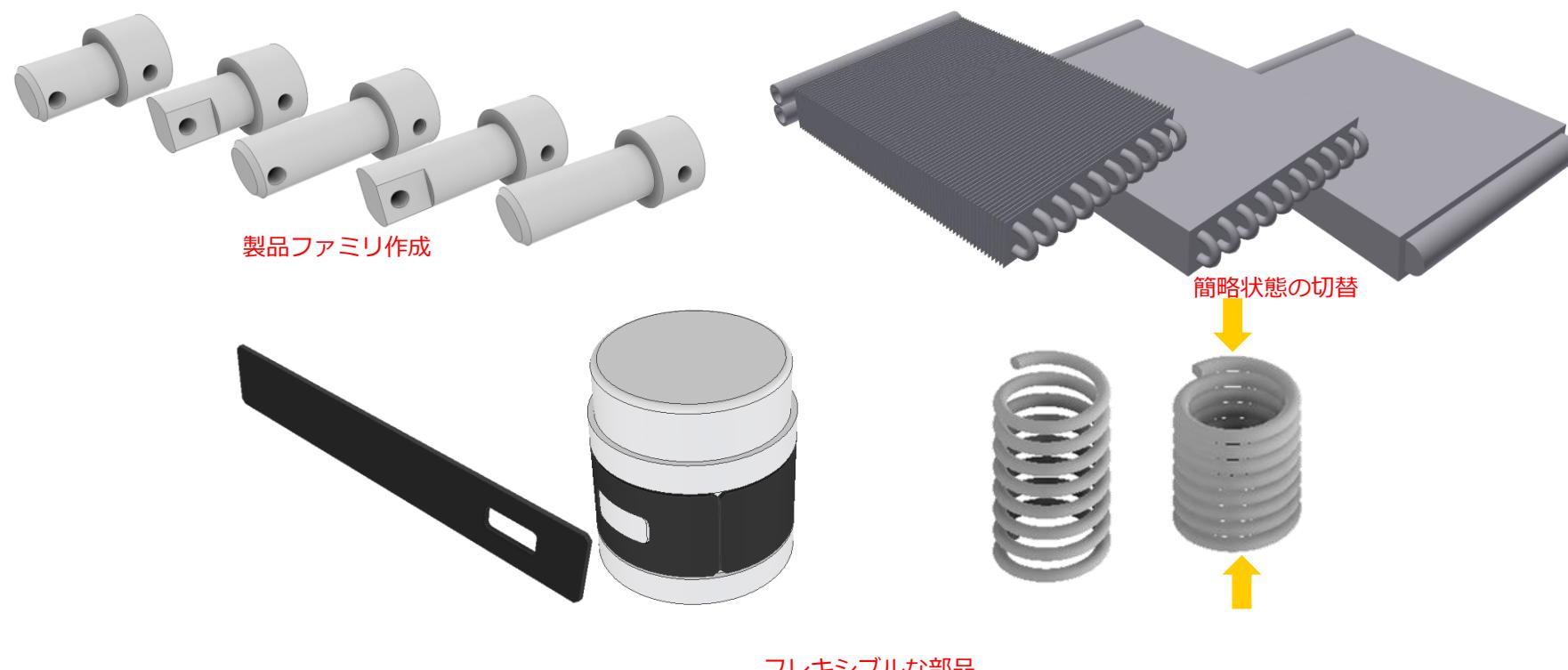
- 1つのドキュメント内に、パートまたはアセンブリの複数の表現を作成することが可能



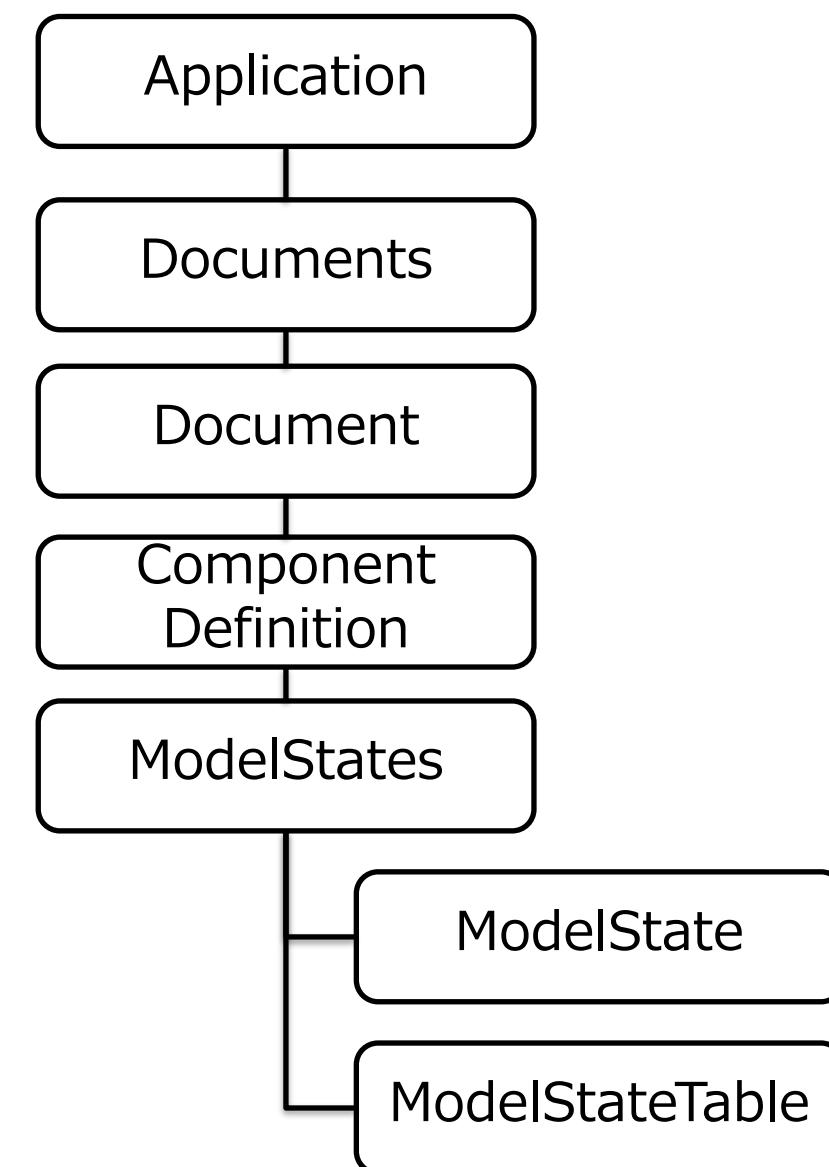
- 製造プロセスごとの形状
- 簡略化レベルの切り替え
- 製品ファミリ作成
- アジャスト可能部品
- フレキシブルな部品の表現
- その他



を1つのドキュメントで表現可能



# Inventor 2022 新機能 モデル状態 API

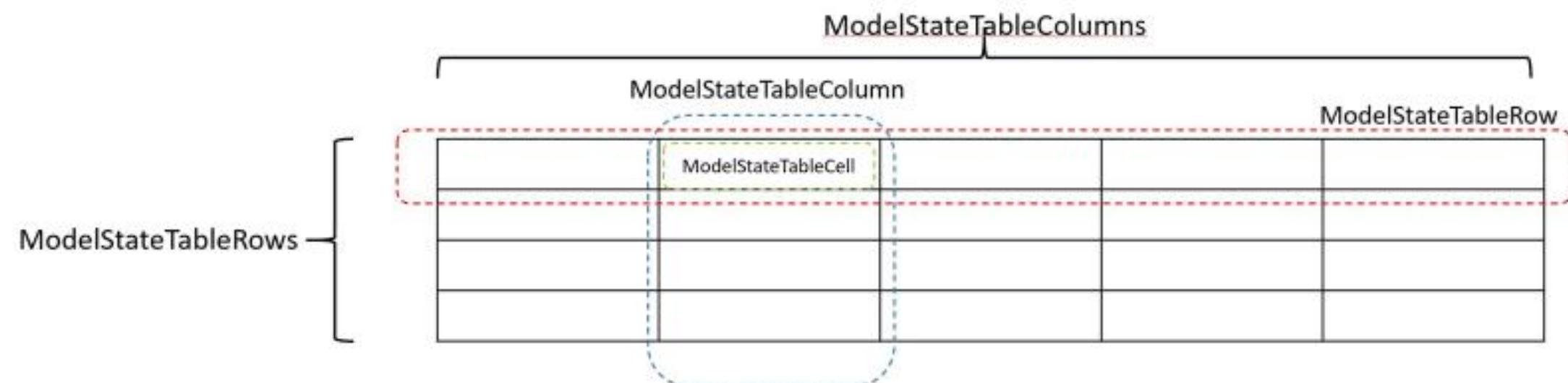


## ModelStateオブジェクト

- ・ 個々のモデル状態を表すオブジェクト。
- ・ モデル状態名や、アクティブ化の状態を取得・設定が可能

## ModelStateTableオブジェクト

- ・ モデル状態毎に、異なる設定を表形式で保持しているオブジェクト
- ・ 表の行が“モデル状態”に、列が“設定”に対応

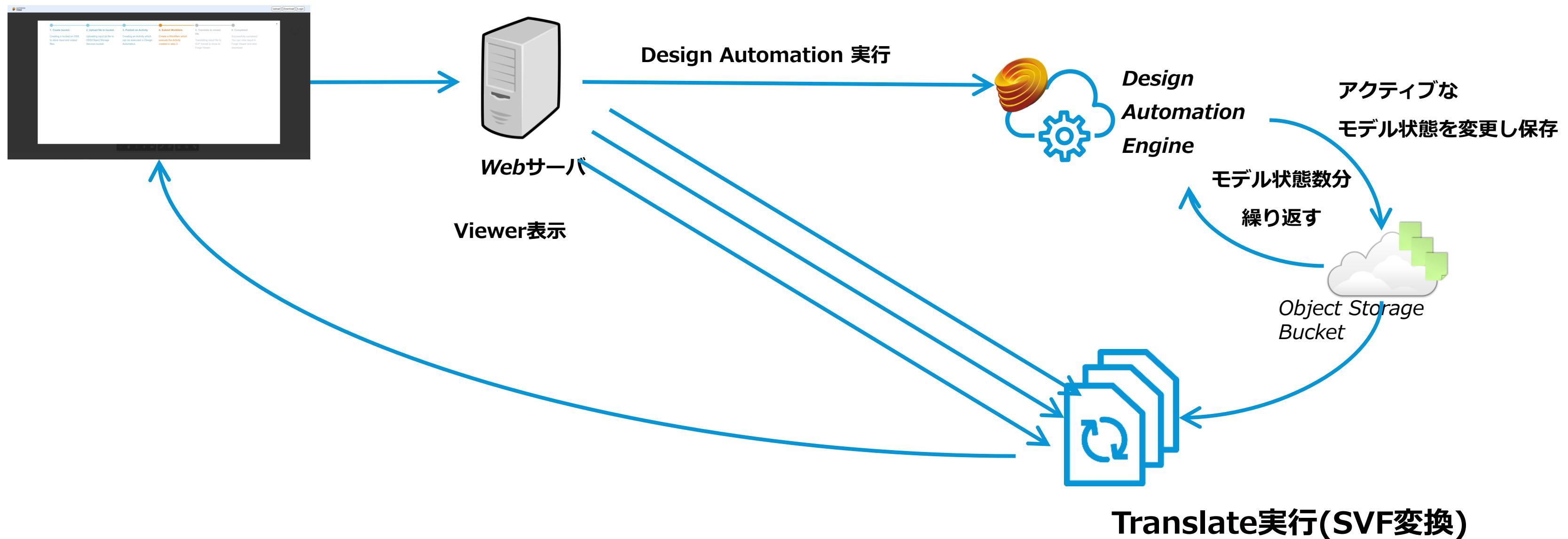


# Inventor モデル状態とForge Model Derivative API

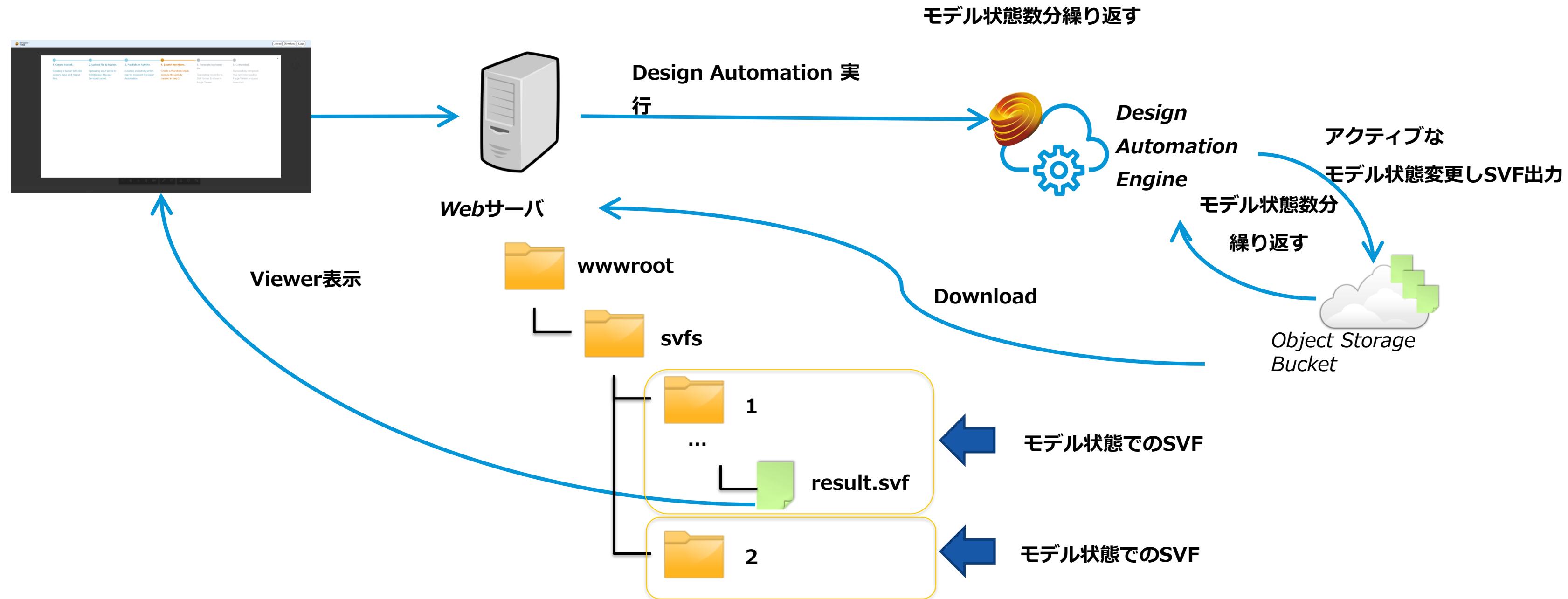
- Forge APIでは、ドキュメントが保存された時にアクティブなモデル状態に対して処理が行われる。
  - Viewableファイルの出力
  - プロパティ情報の出力等
- Forge Viewer (SVF、 SVF2) は汎用Viewerであることもあり、モデル状態を表示する機能はない

⇒Model Derivative APIを使用して全モデル状態をForge Viewerで参照するにはアクティブなモデル状態を切り替えて、モデル状態毎にModel Derivative APIを実行するような処理が必要になる

# Model Derivative API を使用した場合のWorkFlow



# Inventor プラグインでSVFを出力 +ローカルサーバにSVFを保存



# デモアプリケーション

## Upload

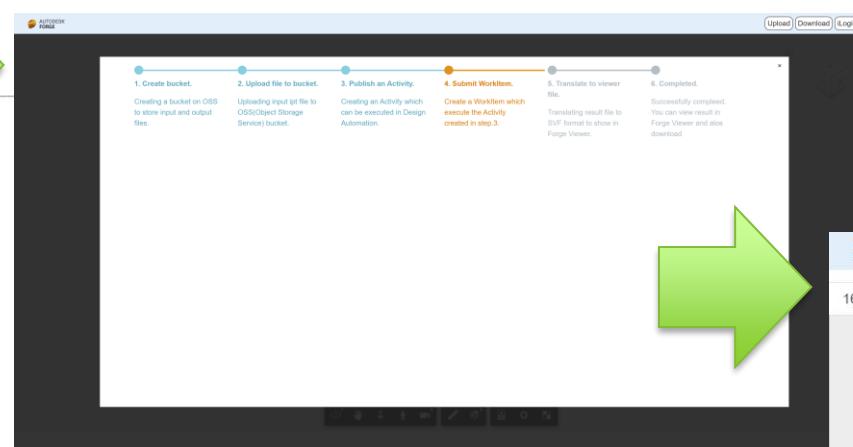


ここにファイルをドロップしてください  
または

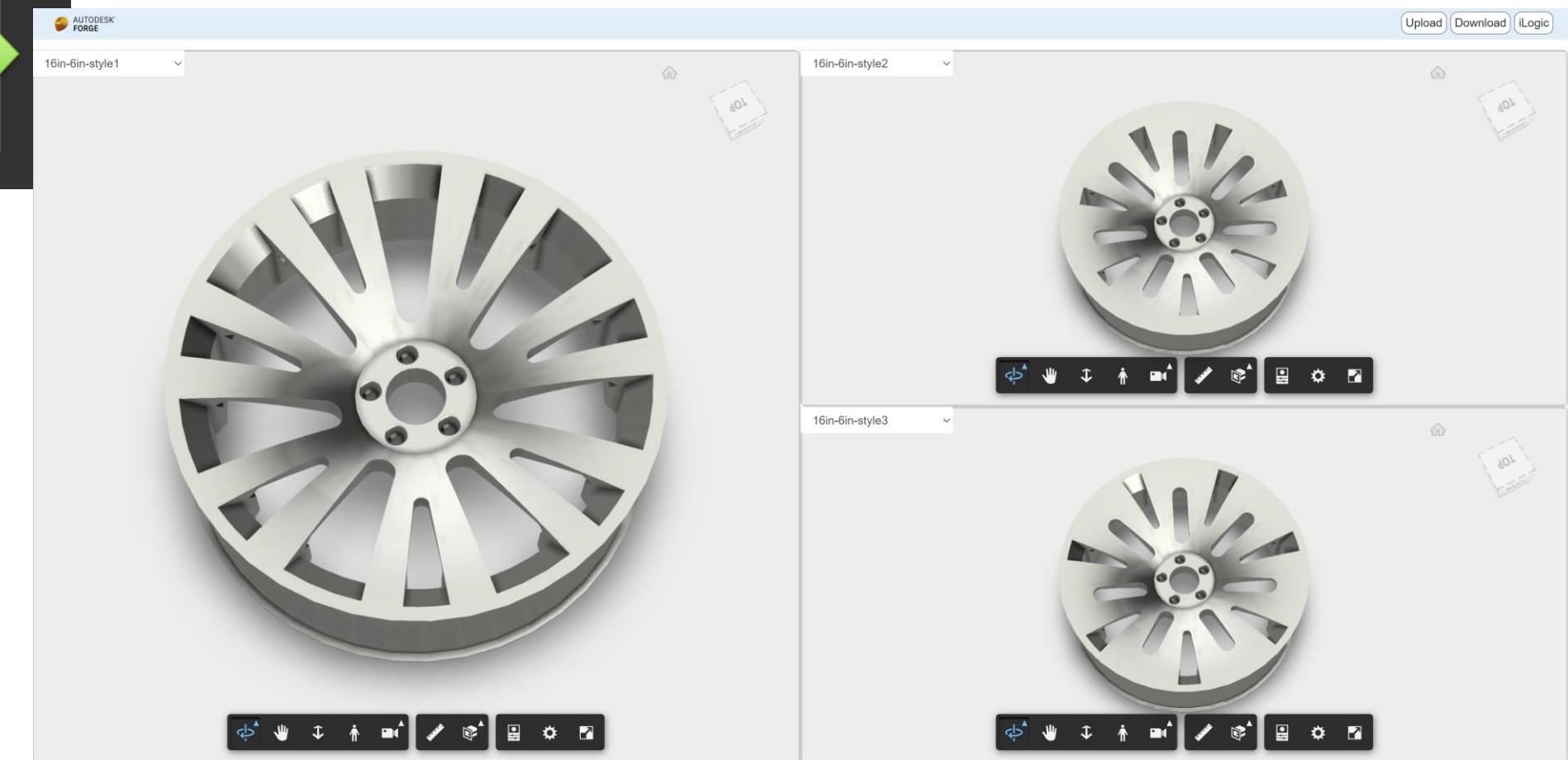
ファイルを選択する



## Design Automation実行



## Viewer表示



# 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

# iLogicファイルの指定方法



## 場所



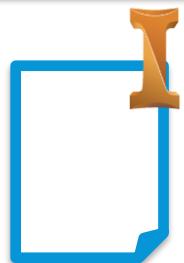
Cloud Storage



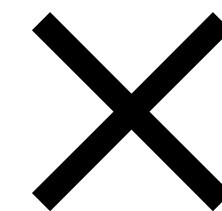
AppBundle

Parameter/  
Argument

```
Parameter("height") = "2  
in"nTrace.Information("Updating  
document")nInven  
torVB.D ...
```



Inventor File



## 設定



Activity



WorkItem

# Parameter/Argumentによる指定



## Activity

```
{
  "id": "{ActivityId}",
  "commandLine": [
    "$(engine.path)¥¥InventorCoreConsole.exe /i ¥"$(args[InventorDoc].path)¥" /s $(args[iLogicName].path)"
  ...
}
```



## WorkItem

```
{
  "activityId": "{specify activity id + alias}",
  "arguments": {
    "InventorDoc": {
      "url": "{specify input file url}"
    },
    "iLogicName": {
      "value": "iLogicVb.RunRule(¥"MyRule¥")"
    },
  ...
}
```

```
[01/27/2021 06:37:20] Starting work item 06d287cc9e7248259f08e6050499a75f
[01/27/2021 06:37:20] Start download phase.
[01/27/2021 06:37:20] Start downloading input: verb - 'GET', url -
'https://developer.api.autodesk.com/oss/v2/signedresources/f46ee514-9332-4e96-8c9a-53a55dd71fce?region=US'
[01/27/2021 06:37:20] '84480' bytes have been written to
T:¥Aces¥Jobs¥06d287cc9e7248259f08e6050499a75f¥Input.ipt.
[01/27/2021 06:37:20] End downloading file 'https://developer.api.autodesk.com/oss/v2/signedresources/f46ee514-
9332-4e96-8c9a-53a55dd71fce?region=US'.
[01/27/2021 06:37:20] End download phase successfully.
[01/27/2021 06:37:21] Start preparing script and command line parameters.
[01/27/2021 06:37:21] Command line: [ /i "T:¥Aces¥Jobs¥06d287cc9e7248259f08e6050499a75f¥Input.ipt" /s
"T:¥Aces¥Jobs¥06d287cc9e7248259f08e6050499a75f¥arg_iLogic.dat" ]
[01/27/2021 06:37:21] End preparing script and command line parameters.
```

WorkItem実行時に、指定したvalue値が記載された  
ファイルとしてCoreエンジンに渡され、実行される

# Inventor SVF アドインによるSVFの出力

```
Sub ExportEachModelStateSVFs()
    Dim svfAddInGUID As String = "{C200B99B-B7DD-4114-A5E9-6557AB5ED8EC}"
    Dim svfAddin As Inventor.ApplicationAddIn = Nothing
    Try
        Trace.TraceInformation("Start to SVF Addin.")
        svfAddin = ThisServer.ApplicationAddIns.ItemById(svfAddInGUID)
        Trace.TraceInformation("End to load SVF Addin.")
    End Try
```

SVFアドイン取得

```
Dim oCompDef As ComponentDefinition = ThisDoc.Document.ComponentDefinition
Dim oModelStates As ModelStates = oCompDef.ModelStates
```

```
Dim modelStates = New System.Collections.Generic.List(Of String)()
```

```
For i As Integer = 1 To oModelStates.Count
    oModelStates(i).Activate
    modelStates.Add(String.Format("{{\"Index\":\"{0}\",\"Description\":\"{1}\",\"Content\":{{\"Content\":[\"{2}\"]}}}}", CStr(i),
    oModelStates(i).Name))
    ExportSVF(CStr(i), svfAddin)
```

```
Next
```

```
Dim docDir As String = System.IO.Path.GetDirectoryName(ThisDoc.Document.FullName)
System.IO.Compression.ZipFile.CreateFromDirectory(System.IO.Path.Combine(docDir, "svfs"),
System.IO.Path.Combine(docDir, "svfs.zip"))
```

```
Dim text As String = String.Format("{{\"modelStates\":[{0}]}}", Join(modelStates.ToArray, ","))
Dim filePath As String = System.IO.Path.Combine(docDir, "modelstate.json")
```

```
If System.IO.File.Exists(filePath) Then
    Trace.TraceInformation("Delete existing Json file file {0}.", filePath)
    System.IO.File.Delete(filePath)
End If
```

```
Trace.TraceInformation("Save Json file file to {0}.", filePath)
System.IO.File.WriteAllText(filePath, text)
```

```
Catch ex As Exception
    Trace.TraceError("Unable to load SVF Add In {0}", ex.Message)
End Try
End Sub
```

モデル状態を切り替えながら繰り返しSVFを出力

出力結果をZip圧縮

# Inventor SVF アドインによるSVFの出力

```

Sub ExportSVF(modelState As String, svfAddin As Inventor.ApplicationAddIn)
    Trace.TraceInformation("Start extract SVF.")

    Dim oContext As TranslationContext =
    ThisServer.TransientObjects.CreateTranslationContext()
    oContext.Type = IOMechanismEnum.kFileBrowseIOMechanism

    Dim oOptions As NameValueMap =
    ThisServer.TransientObjects.CreateNameValueMap()
    Dim oData As DataMedium = ThisServer.TransientObjects.CreateDataMedium()

    Dim docDir As String =
    System.IO.Path.GetDirectoryName(ThisDoc.Document.FullName)
    Dim outputFolderPath As String = System.IO.Path.Combine(docDir, "svfs",
modelState)

    If System.IO.Directory.Exists(outputFolderPath) Then
        System.IO.Directory.Delete(outputFolderPath, True)
    End If
    System.IO.Directory.CreateDirectory(outputFolderPath)

    oData.FileName = System.IO.Path.Combine(outputFolderPath,
"result.collaboration")

    If (svfAddin.HasSaveCopyAsOptions(ThisDoc.Document, oContext, oOptions)) Then

        oOptions.Value("EnableExpressTranslation") = False
        oOptions.Value("SVFFileOutputDir") = outputFolderPath
        oOptions.Value("ExportFileProperties") = True
        oOptions.Value("ObfuscateLabels") = False
    End If

    svfAddin.SaveCopyAs( ThisDoc.Document, oContext, oOptions, oData)

    Trace.TraceInformation("End extract SVF.")
End Sub

```



SVFに（現在のモデル状態を）出力

# 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。

# Viewstateの同期

```
var sfilter = {
  viewport: true
};
```

```
viewer3d1.addEventListener(Autodesk.Viewing.CAMERA_CHANGE_EVENT, function () {
  var v1state = viewer3d1.getState(sfilter);
  var v2state = viewer3d2.getState(sfilter);
  var v3state = viewer3d3.getState(sfilter);
```

```
v1state['seedURN'] = "";
v2state['seedURN'] = "";
v3state['seedURN'] = "";
```

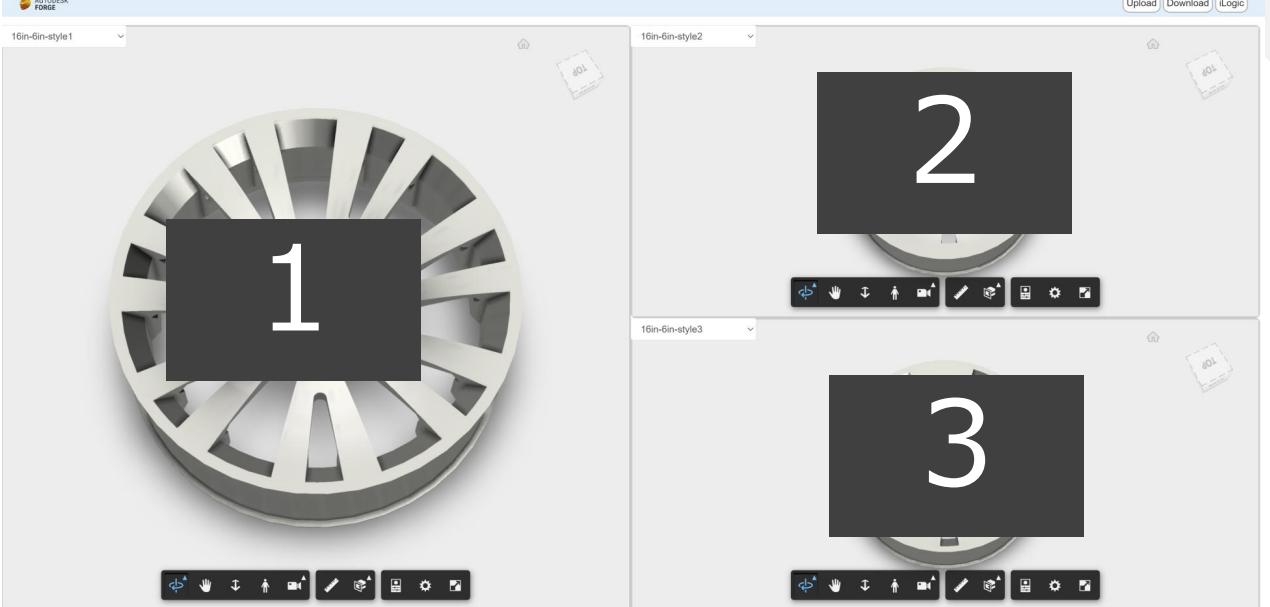
```
if (!viewer3d1.viewerState.areEqual(v1state, v2state)) {
  viewer3d2.restoreState(viewer3d1.getState(sfilter), sfilter, true);
}
```

```
if (!viewer3d1.viewerState.areEqual(v1state, v3state)) {
  viewer3d3.restoreState(viewer3d1.getState(sfilter), sfilter, true);
}
```

```
});
```

```
...
```

同様のEventリスナーを2と3にも設定



# まとめ

- **Inventor 2022の新機能 モデル状態により、1つのドキュメント内に、パートまたはアセンブリの複数の表現を作成することが可能**  
→Forge APIでは、ドキュメントが保存された時にアクティブなモデル状態に対して処理が行われる
- **Inventorのプラグインと、Design Automationで、モデル状態を切り替えて、モデル状態毎のSVFを出力可能**  
→対象はパートファイルとアセンブリ（図面は対象外）  
→今のところ、アドインからのSVF2の出力はサポートされていない

# 参考ブログ記事

- Inventor 2022 新機能～その2

[https://adndevblog.typepad.com/technology\\_perspective/2021/04/inventor-2022-whats-new-part2.html](https://adndevblog.typepad.com/technology_perspective/2021/04/inventor-2022-whats-new-part2.html)

- Inventor 2022でのAPI Update

[https://adndevblog.typepad.com/technology\\_perspective/2021/05/inventor-2022-api-update.html](https://adndevblog.typepad.com/technology_perspective/2021/05/inventor-2022-api-update.html)

- Forge Online : Design Automation : Inventor iLogicの活用

[https://adndevblog.typepad.com/technology\\_perspective/2021/01/forge-onlinedesign-automationinventor-ilogic.html](https://adndevblog.typepad.com/technology_perspective/2021/01/forge-onlinedesign-automationinventor-ilogic.html)

- Design Automation API for Inventor - Inventor iLogic利用時の注意点

[https://adndevblog.typepad.com/technology\\_perspective/2020/12/design-automation-api-for-inventor-note-writing-an-ilogic-for-da.html](https://adndevblog.typepad.com/technology_perspective/2020/12/design-automation-api-for-inventor-note-writing-an-ilogic-for-da.html)

- Run iLogic Rule without AppBundle

<https://forge.autodesk.com/blog/run-ilogic-rule-without-appbundle>

- Different ways to run iLogic code

<https://forge.autodesk.com/ja/node/1801>

- Prepare iLogic Rules for Design Automation

<https://forge.autodesk.com/blog/prepare-ilogic-rules-design-automation>

- Speed up viewable generation when using Design Automation for Inventor

<https://forge.autodesk.com/blog/speed-viewable-generation-when-using-design-automation-inventor>



# AUTODESK®

Make anything.