



# AUTODESK Platform Services

## Design Automation API for Revit

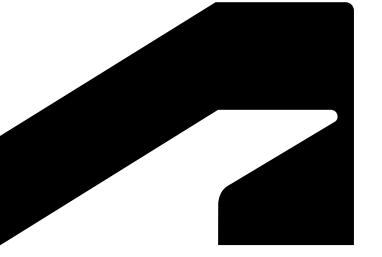
### アップデート詳細

小笠原 龍司

Developer Advocacy & Support、Autodesk Developer Network

# アジェンダ

- Design Automation API 概要
- Direct-to-S3 対応
- 複数ファイルを ZIP 化して出力する
- サポートフォントの追加
- PDF 生成のサポート
- 外部ネットワークとの接続
- Revit Cloud Model サポート
- WorkItem の処理時間

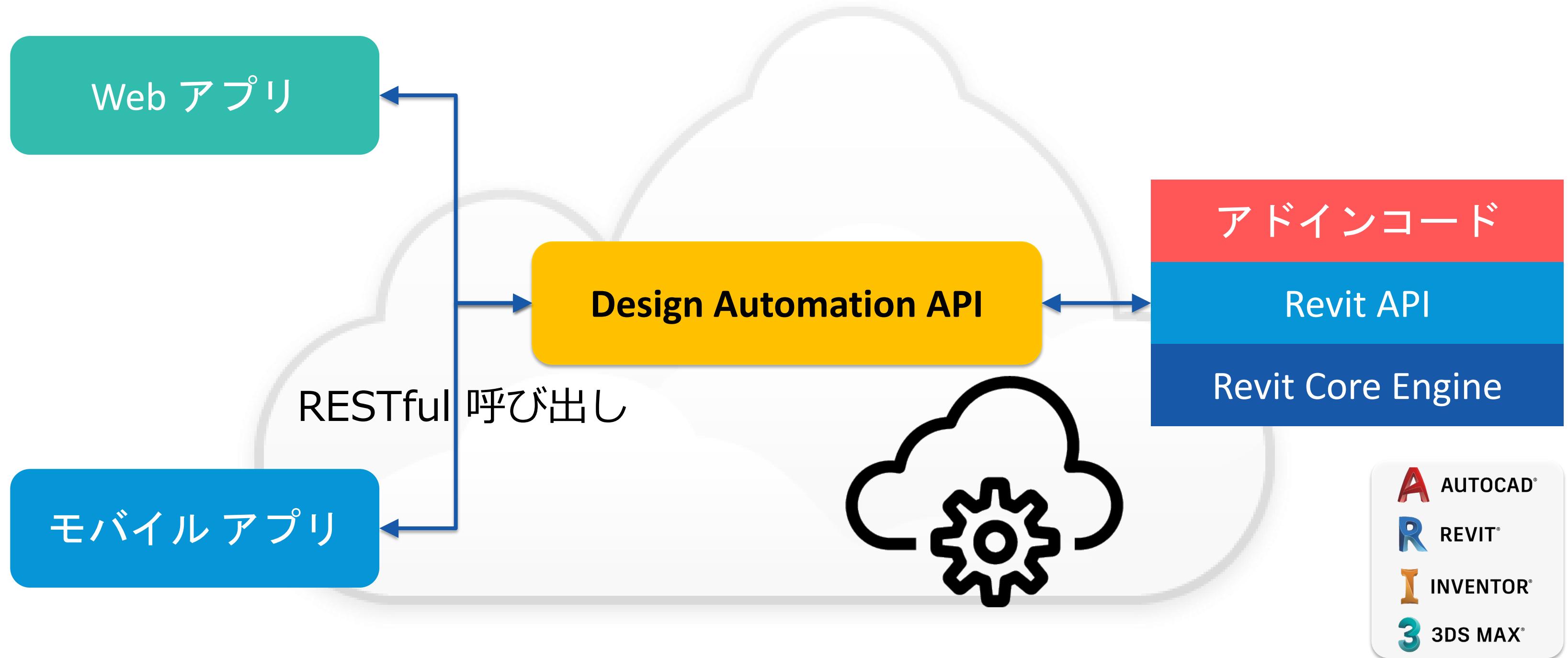


# Design Automation API

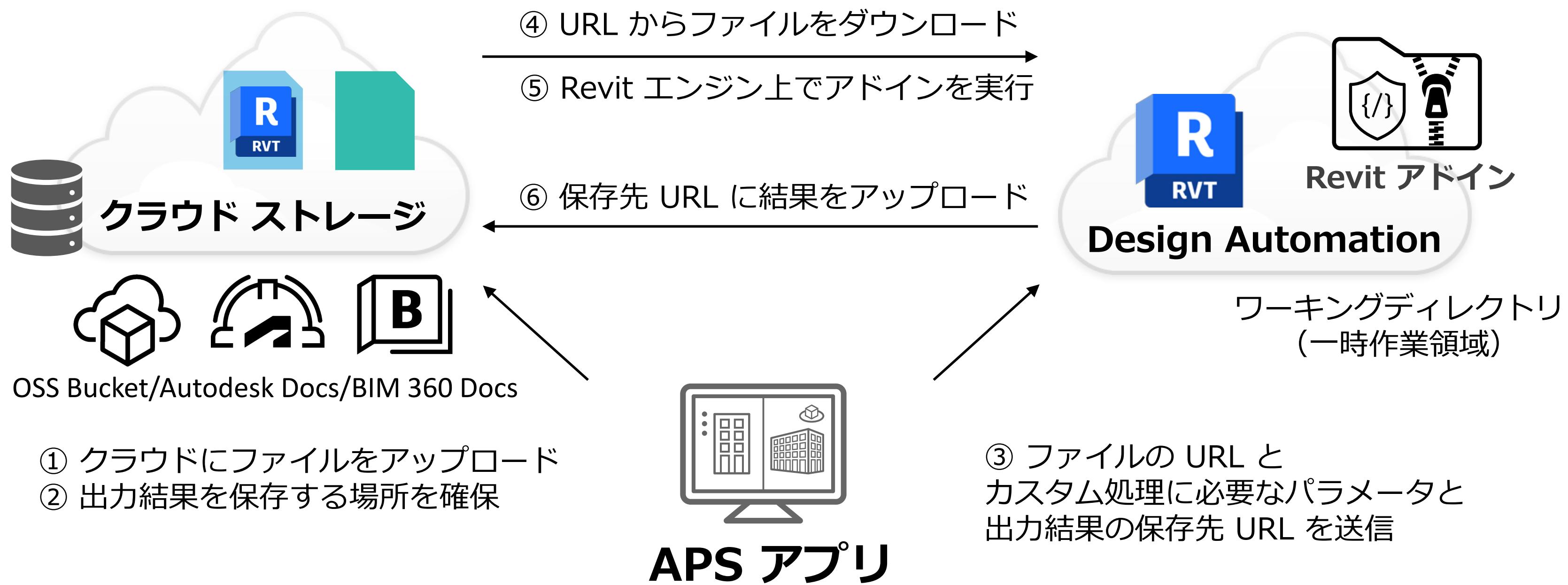
## 概要



# Revit アドインをクラウドで実行するための API

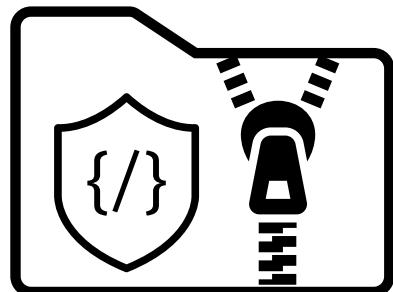


# Design Automation for Revit のワークフロー



# 3つのキーワード

## AppBundle

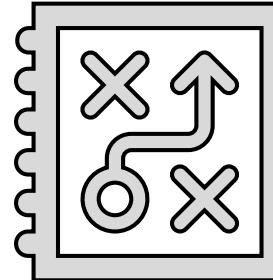


Revit アドインのパッケージ

```
Id : CustomApp
Engine : 2023
Description: Test custom app
Package: Storage URL
```

Revit の .NET API で作成したアセンブリや関連ファイルを ZIP 圧縮してアップロードし、AppBundle として登録する。

## Activity

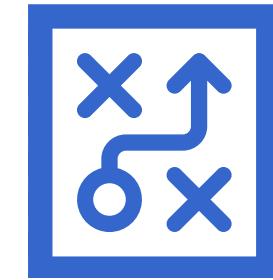


実行されるアクションの定義

```
Id: UpdateParamActivity
Input Parameter A : RVT, TXT
Output Parameter B: RVT
AppBundle: CustomApp
```

カスタム処理の雛型を定義する。  
.NET アセンブリ内でどんなデータを入力して、どんなデータを出力するか定義する。

## WorkItem



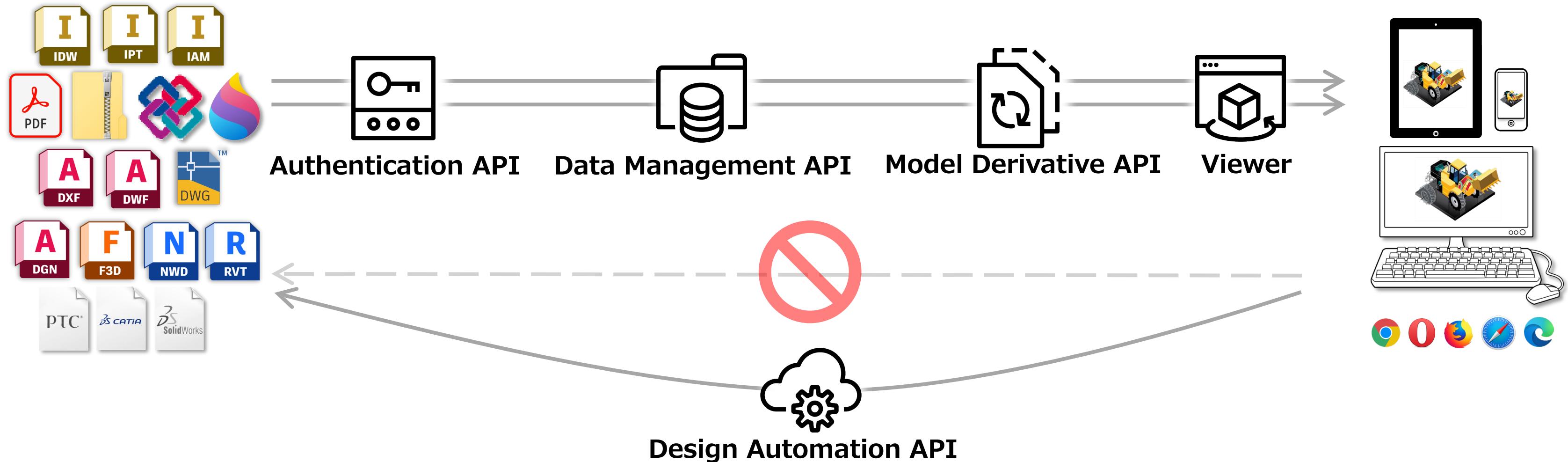
指定のアクションを呼び出すジョブ

```
Id: 返却される文字列
Activity : UpdateParamActivity
Input Parameter A : File URL
Output Parameter B: Storage URL
```

REST API でリクエストするジョブ。  
対象のモデルやテキストデータ、出力先の URL と、実行する Activity を指定する。

# Viewer と Design Automation API の連携

- Design Automation API の連携は可能
- AutoCAD、Revit、Inventor、3ds Max に対応



# Design Automation API を正しく理解しましょう

## 対話的な表示/編集機能は ありません

- Revit Web 版 (Revit UI 画面) のようなものではありません

## ビューア機能は ありません

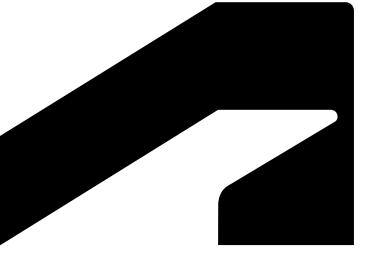
- 必要に応じて Forge Viewer の利用を検討出来ます

## エンドユーザ向けのサービスでは ありません

- 開発者向けのサービスです

## サーバー モジュールでは ありません

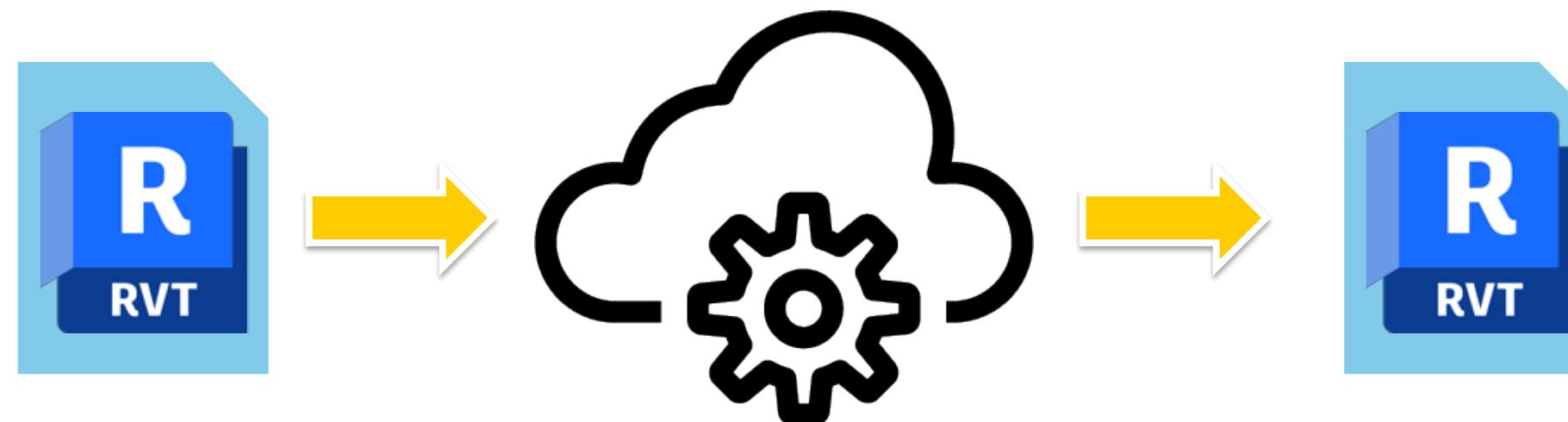
- オンプレミス(プライベート)サーバー版はありません



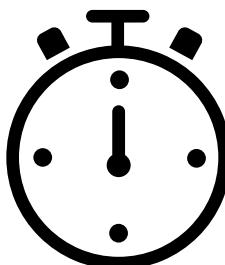
**Direct-to-S3 対応**

# Design Automation for Revit での対応

- WorkItem の入出力ファイルに OSS Bucket、Autodesk Docs、BIM 360 Docs を指定している場合、新しい Direct-to-S3 アプローチをお使いいただけるようになりました。



AWS S3 Bucket と Object に直接アクセス  
大容量ファイルのアップロード/ダウンロード速度が向上



# 変更点

- WorkItem リクエスト時に入力・出力ファイルのパスを指定する際、署名付き S3 URLを生成する必要がなくなりました。
- **objectId** と **アクセストークン** のみで OK

```
{  
  "activityId": "{{dasNickName}}.{{activityName}}+{{activityAlias}}",  
  "arguments": {  
    "inputRvtFile": {  
      "verb": "get",  
      "url": "urn:adsk.objects:os.object:test_bucket/Sample_Revit_Original.rvt",  
      "headers": {  
        "Authorization": "Bearer {{access_token}}"  
      }  
    },  
    "resultRvtFile": {  
      "verb": "put",  
      "url": "urn:adsk.objects:os.object:test_bucket/Sample_Revit_Updated.rvt ",  
      "headers": {  
        "Authorization": "Bearer {{access_token}}"  
      }  
    }  
  }  
}
```



# Object Storage Service (OSS) Bucket の場合

- 入力ファイル
  - Bucket に保存されている Object の一覧を取得します。
    - bucketKey をキーに GET Objects エンドポイントを呼び出します。

```
{  
  "bucketKey": "test_bucket",  
  "objectKey": "サンプル意匠.rvt",  
  "objectId": "urn:adsk.objects:os.object:test_bucket/%E3%82%B5%E3%83%B3%E3%83%97%E3%83%AB%E6%84%8F%E5%8C%A0.rvt",  
  "sha1": "6be5ee13c56179c6a1bb8342a2a8d2a671910c22",  
  "size": 66318336,  
  "location": "https://developer.api.autodesk.com/oss/v2/buckets/test_bucket/objects/%E3%82%B5%E3%83%B3%E3%83%97%E3%83%  
AB%E6%84%8F%E5%8C%A0.rvt"  
}
```

- 出力ファイル
  - “**urn:adsk.objects:os.object:<BucketKey>/<ObjectKey>**” の形式で直接パスを作成します。

# Autodesk Docs / BIM 360 Docs の場合

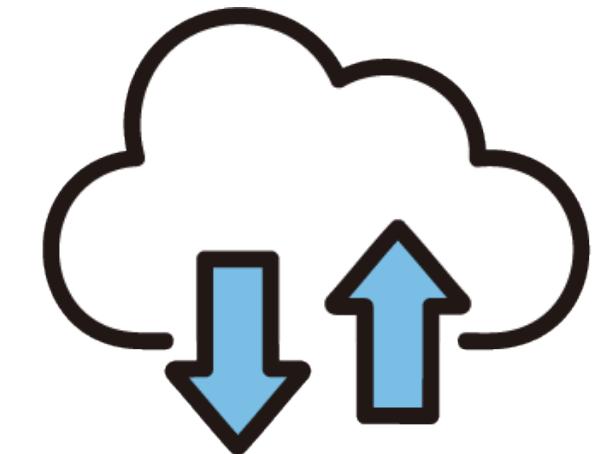
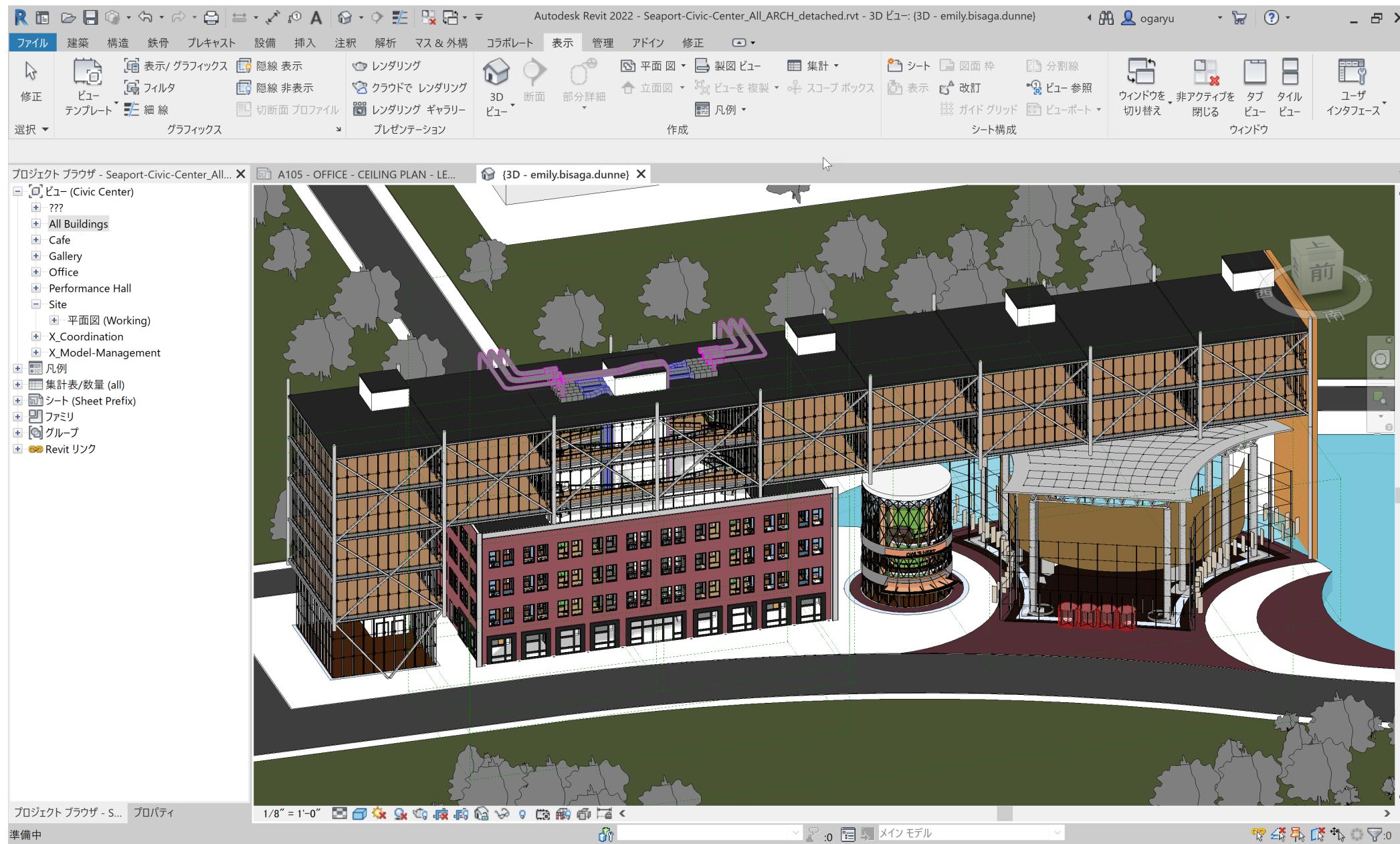
- 入力ファイル
  - ファイルのバージョンは GET Version エンドポイントで取得し、objectId は response.data.relationships.storage.data.id で確認します。

```
"storage": {  
    "data": {  
        "type": "objects",  
        "id": "urn:adsk.objects:os.object:wip.dm.prod/ecc3802e-d16a-46c8-afdc-aa6e482a93c2.rvt"  
    },  
    ~ 省略 ~  
}
```

- 出力ファイル
  - POST projects/:project\_id/storage エンドポイントを使用して、ファイルをアップロードするためのストレージを作成します。この際、objectId は response.data.id で取得することができます。

# テスト結果

- ZIP 圧縮したモデル 約200MB

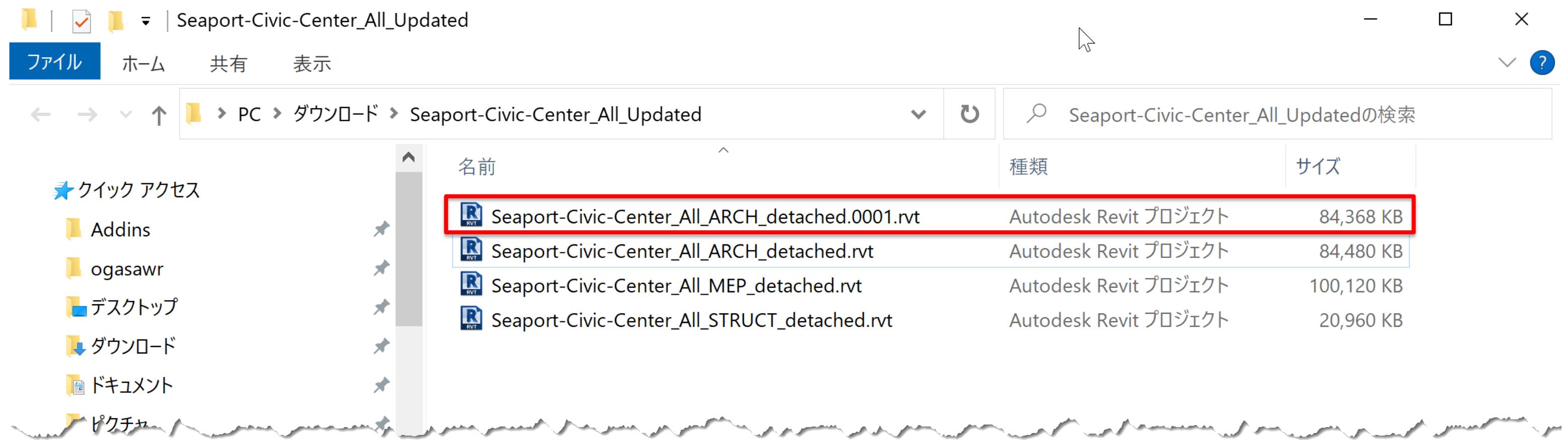


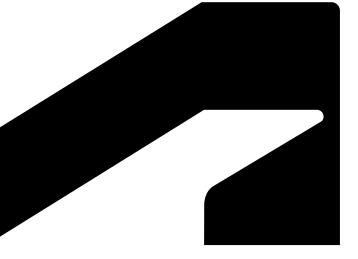
旧方式  
ダウンロード：5秒  
アップロード：27秒

新方式  
ダウンロード：6秒  
アップロード：17秒

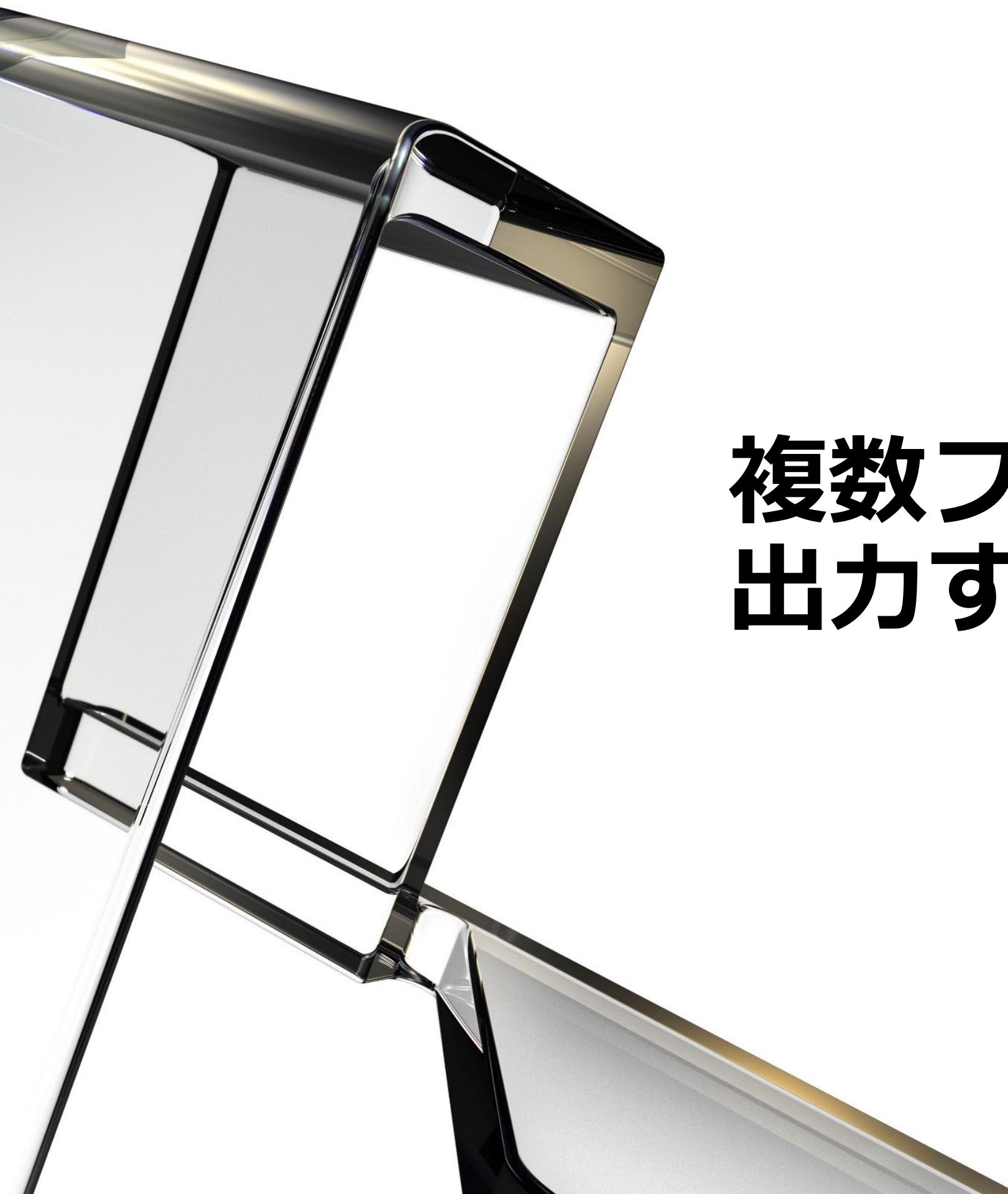
# ZIP 圧縮して出力する場合の注意点

- バックアップファイルが自動的に作成されてしまうため、サイズが元のサイズより大きくなります。



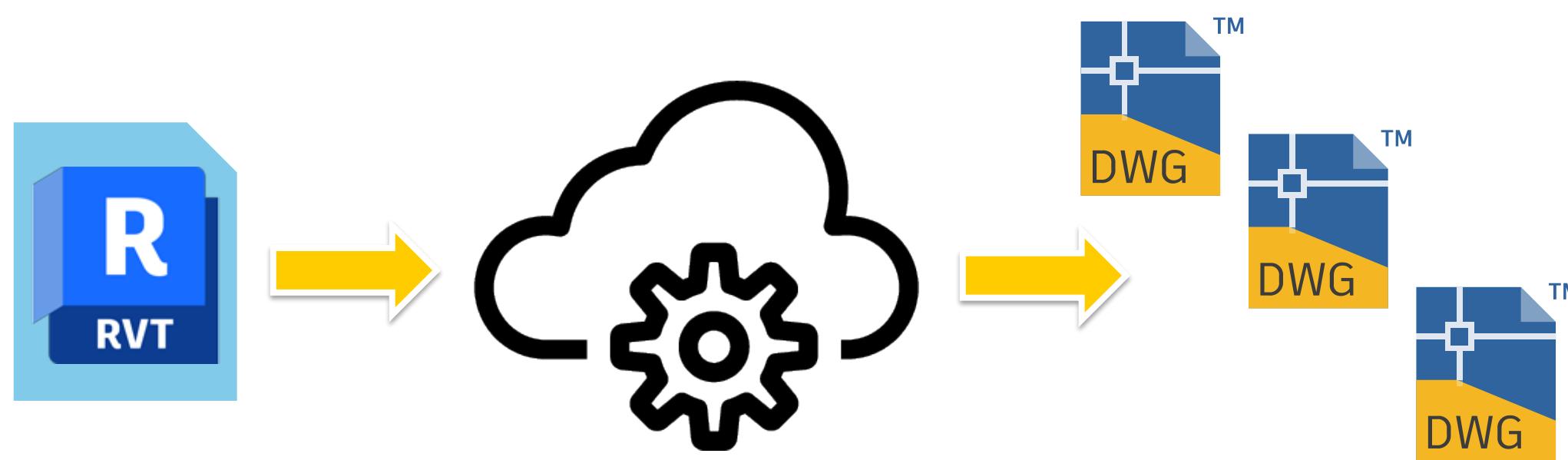


複数ファイルを ZIP 化して  
出力する

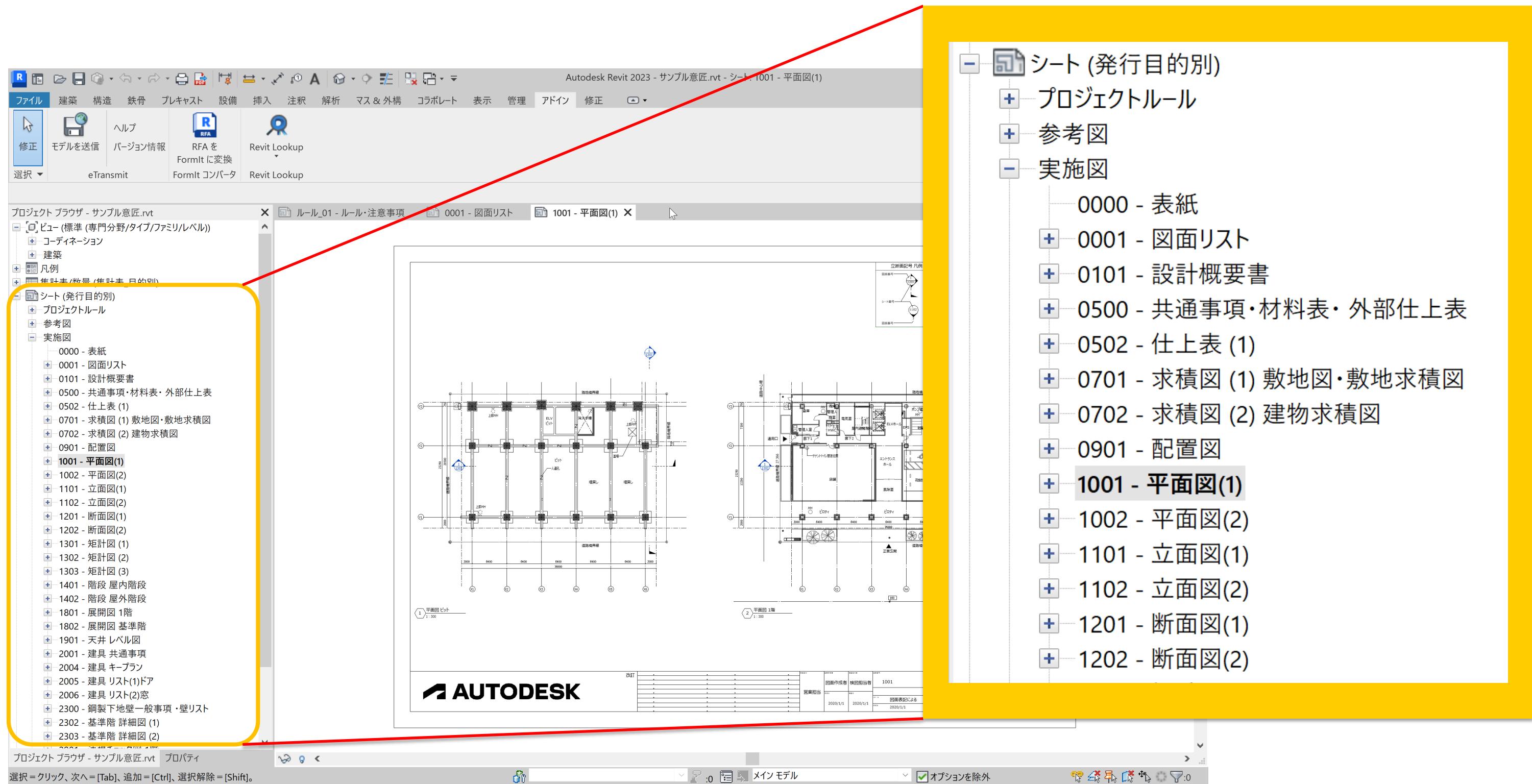


# 動的に複数ファイルを出力する場合のアップロード方法

- 出力したいファイルが複数あり、且つその数が WorkItem のリクエスト時に確定できない場合
  - シートの書き出し
  - 抽出したデータを他のファイル形式で書き出し
  - 複数ファミリの作成

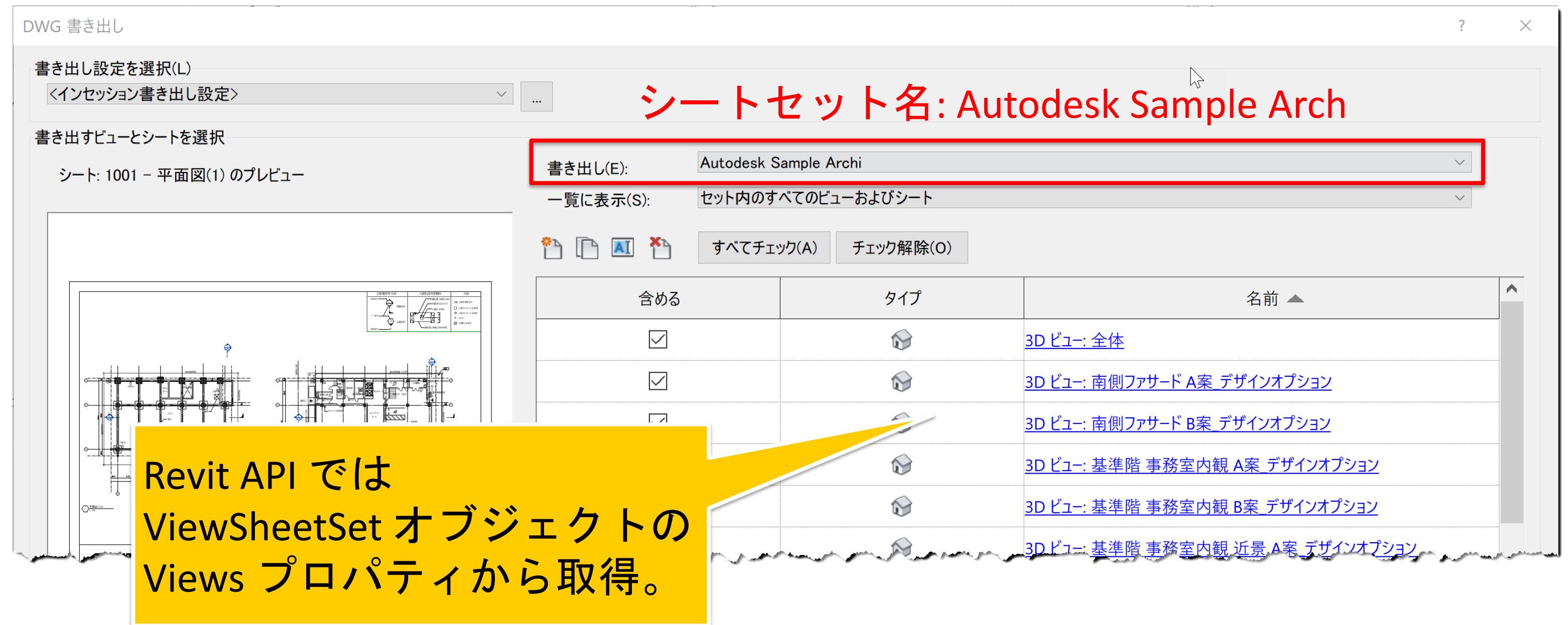


# Revit サンプルモデルのシート



# シートセットの複数シートを一括で書き出す場合

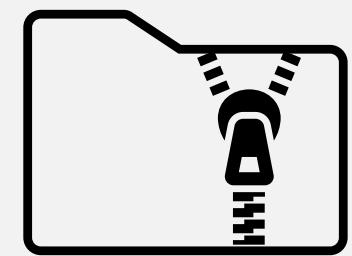
- シートセットに登録されているシートを取得。



# Activity 定義

```
{
  "id": "{$activityName}",
  "commandLine": [ "$(engine.path)¥¥¥¥revitcoreconsole.exe /i ¥$(args[inputRvtFile].path)¥" /al ¥$(appbundles[{$AppBundleName}].path)¥" /I JPN" ],
  "parameters": {
    "inputRVTFile": {
      "zip": false,
      "ondemand": false,
      "verb": "get",
      "description": "Input Revit model",
      "required": true
    },
    "resultDWGFile": {
      "zip": true,
      "ondemand": false,
      "verb": "put",
      "description": "Results",
      "required": true,
      "localName": "exportedDwgs"
    }
  },
  "engine": "Autodesk.Revit+2023",
  "appbundles": [ "{$dasNickName}.{$AppBundleName}+{$AppBundleAlias}" ],
  "description": "."
}
```

- “verb”: “put” → DA4R から出力するという意味
- “zip”: “true” → ZIP 圧縮を指定
- “localName”: “exportedDwgs”  
→ 指定の名前のサブフォルダを圧縮する



# WorkItem のリクエスト (OSS Bucket の場合)

- “urn:adsk.objects:os.object:<BucketKey>/<ObjectKey>” の形式でパスを作成。ストレージの事前確保は不要。

```
{  
  "activityId": "{{dasNickName}}.{{activityName}}+{{activityAlias}}",  
  "arguments": {  
    "inputRvtFile": {  
      "verb": "get",  
      "url": "urn:adsk.objects:os.object:test_bucket/Sample_Revit_2023.rvt",  
      "headers": {  
        "Authorization": "Bearer {{access_token}}"  
      }  
    },  
    "resultDWGFile": {  
      "verb": "put",  
      "url": "urn:adsk.objects:os.object:test_bucket/exportedDwgs.zip", objectID を設定  
      "headers": {  
        "Authorization": "Bearer {{access_token}}"  
      }  
    }  
  }  
}
```



objectId を設定

# WorkItem のリクエスト (Autodesk Docs の場合)

- POST projects/:project\_id/storage エンドポイントを使用して、ファイルをアップロードするストレージを作成します。

```
curl --location --request POST 'https://developer.api.autodesk.com/data/v1/projects/b.******/storage' ¥  
--header 'Authorization: Bearer *****' ¥  
--header 'Content-Type: application/json' ¥  
--header 'Cookie: PF=3WPKykFgS6bYB59Fzylv1D' ¥  
--data-raw '{  
  "jsonapi": { "version": "1.0" },  
  "data": {  
    "type": "objects",  
    "attributes": {  
      "name": "exportedDwgs.zip" } ファイル名 : exportedDwgs.zip  
    },  
    "relationships": {  
      "target": {  
        "data": { "type": "folders", "id": "urn:adsk.wipprod:fs.folder:*****" } } フォルダ ID  
      }  
    }  
  }  
}'
```

# WorkItem のリクエスト (Autodesk Docs の場合)

```
{  
  "jsonapi": {  
    "version": "1.0"  
  },  
  "data": {  
    "type": "objects",  
    "id": "urn:adsk.objects:os.object:wip.d  
          m.prod/fdf3dcbe-7d03-42fe-ad93-0ab6766a872a.zip",  
    "relationships": {  
      "target": {  
        "data": {  
          "type": "folders",  
          "id": "urn:adsk.wipprod:fs.folder:co.5QtsXDrvR4uVgtvlyoKpaw"  
        },  
      },  
      "links": {  
        "related": {  
          "href": "https://developer.api.autodesk.com/data/v1/projects/b.******/folders/urn:adsk.wipprod:fs.folder:*****"  
        }  
      }  
    }  
  }  
}
```

objectId を取得

# Design Automation for Revit での DWG 出力結果

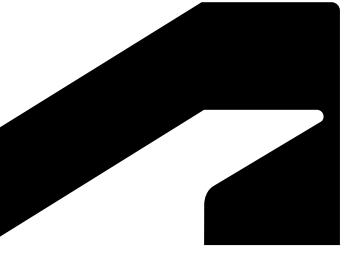
# exportedDwgs.zip フォルダ内

- サンプル意匠\_シート-3D ビュー - 基準階 事務室内観 A案\_デザインオプション.dwg
- サンプル意匠\_シート-3D ビュー - 基準階 事務室内観 A案\_デザインオプション.pcp
- サンプル意匠\_シート-3D ビュー - 基準階 事務室内観 B案\_デザインオプション.dwg
- サンプル意匠\_シート-3D ビュー - 基準階 事務室内観 B案\_デザインオプション.pcp
- サンプル意匠\_シート-3D ビュー - 基準階 事務室内観 近景 A案\_デザインオプション.dwg
- サンプル意匠\_シート-3D ビュー - 基準階 事務室内観 近景 A案\_デザインオプション.pcp
- サンプル意匠\_シート-3D ビュー - 基準階 事務室内観 近景 B案\_デザインオプション.dwg
- サンプル意匠\_シート-3D ビュー - 基準階 事務室内観 近景 B案\_デザインオプション.pcp
- サンプル意匠\_シート-3D ビュー - 全体.dwg
- サンプル意匠\_シート-3D ビュー - 全体.pcp
- サンプル意匠\_シート-3D ビュー - 南側ファサード A案\_デザインオプション.dwg
- サンプル意匠\_シート-3D ビュー - 南側ファサード A案\_デザインオプション.pcp
- サンプル意匠\_シート-3D ビュー - 南側ファサード B案\_デザインオプション.dwg
- サンプル意匠\_シート-3D ビュー - 南側ファサード B案\_デザインオプション.pcp
- サンプル意匠\_シート-シート - 0000 - 表紙.dwg
- サンプル意匠\_シート-シート - 0000 - 表紙.pcp
- サンプル意匠\_シート-シート - 0001 - 図面リスト.dwg
- サンプル意匠\_シート-シート - 0001 - 図面リスト.pcp
- サンプル意匠\_シート-シート - 0101 - 設計概要書.dwg
- サンプル意匠\_シート-シート - 0101 - 設計概要書.pcp
- サンプル意匠\_シート-シート - 0500 - 共通事項・材料表・外部仕上表.dwg
- サンプル意匠\_シート-シート - 0500 - 共通事項・材料表・外部仕上表.pcp
- サンプル意匠\_シート-シート - 0502 - 仕上表 (1).dwg
- サンプル意匠\_シート-シート - 0502 - 仕上表 (1).pcp
- サンプル意匠\_シート-シート - 0701 - 求積図 (1) 敷地図・敷地求積図.dwg
- サンプル意匠\_シート-シート - 0701 - 求積図 (1) 敷地図・敷地求積図.pcp
- サンプル意匠\_シート-シート - 0702 - 求積図 (2) 建物求積図.dwg
- サンプル意匠\_シート-シート - 0702 - 求積図 (2) 建物求積図.pcp
- サンプル意匠\_シート-シート - 0901 - 配置図.dwg

- サンプル意匠\_シート-シート - 0901 - 配置図.pcp
- サンプル意匠\_シート-シート - 1001 - 平面図(1).dwg
- サンプル意匠\_シート-シート - 1001 - 平面図(1).pcp
- サンプル意匠\_シート-シート - 1002 - 平面図(2).dwg
- サンプル意匠\_シート-シート - 1002 - 平面図(2).pcp
- サンプル意匠\_シート-シート - 1101 - 立面図(1).dwg
- サンプル意匠\_シート-シート - 1101 - 立面図(1).pcp
- サンプル意匠\_シート-シート - 1102 - 立面図(2).dwg
- サンプル意匠\_シート-シート - 1102 - 立面図(2).pcp
- サンプル意匠\_シート-シート - 1201 - 断面図(1).dwg
- サンプル意匠\_シート-シート - 1201 - 断面図(1).pcp
- サンプル意匠\_シート-シート - 1202 - 断面図(2).dwg
- サンプル意匠\_シート-シート - 1202 - 断面図(2).pcp
- サンプル意匠\_シート-シート - 1301 - 矩計図 (1).dwg
- サンプル意匠\_シート-シート - 1301 - 矩計図 (1).pcp
- サンプル意匠\_シート-シート - 1401 - 階段 屋内階段.dwg
- サンプル意匠\_シート-シート - 1401 - 階段 屋内階段.pcp
- サンプル意匠\_シート-シート - 1402 - 階段 屋外階段.dwg
- サンプル意匠\_シート-シート - 1402 - 階段 屋外階段.pcp
- サンプル意匠\_シート-シート - 1801 - 展開図 1階.dwg
- サンプル意匠\_シート-シート - 1801 - 展開図 1階.pcp
- サンプル意匠\_シート-シート - 1802 - 展開図 基準階.dwg
- サンプル意匠\_シート-シート - 1802 - 展開図 基準階.pcp
- サンプル意匠\_シート-シート - 1901 - 天井 レベル図.dwg
- サンプル意匠\_シート-シート - 1901 - 天井 レベル図.pcp
- サンプル意匠\_シート-シート - 2001 - 建具 共通事項.dwg
- サンプル意匠\_シート-シート - 2001 - 建具 共通事項.pcp
- サンプル意匠\_シート-シート - 2004 - 建具 キープラン.dwg
- サンプル意匠\_シート-シート - 2004 - 建具 キープラン.pcp
- サンプル意匠\_シート-シート - 2005 - 建具 リスト(1)ドア.dwg

- サンプル意匠\_シート-シート - 2005 - 建具 リスト(1)ドア.pcp
- サンプル意匠\_シート-シート - 2006 - 建具 リスト(2)窓.dwg
- サンプル意匠\_シート-シート - 2006 - 建具 リスト(2)窓.pcp
- サンプル意匠\_シート-シート - 2300 - 鋼製下地壁一般事項・壁リスト.dwg
- サンプル意匠\_シート-シート - 2300 - 鋼製下地壁一般事項・壁リスト.pcp
- サンプル意匠\_シート-シート - 2302 - 基準階 詳細図 (1).dwg
- サンプル意匠\_シート-シート - 2302 - 基準階 詳細図 (1).pcp
- サンプル意匠\_シート-シート - 2303 - 基準階 詳細図 (2).dwg
- サンプル意匠\_シート-シート - 2303 - 基準階 詳細図 (2).pcp
- サンプル意匠\_シート-シート - 2901 - 法規チェック図 1階.dwg
- サンプル意匠\_シート-シート - 2901 - 法規チェック図 1階.pcp
- サンプル意匠\_シート-シート - 2902 - 法規チェック図 基準階.dwg
- サンプル意匠\_シート-シート - 2902 - 法規チェック図 基準階.pcp

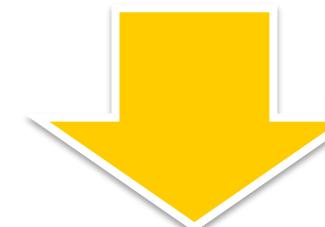




**サポートフォントの追加**

# 日本語フォントの問題

- Revit を起動する OS は **Windows Server 2019 英語版**
- Worker インスタンスで使用できないフォントは自動的に置換。



- DWG などの書き出し時に英語名のフォントに置き換わる。



Desktop Revit  
MS Pゴシック\_B



Design Automation Revit  
MS PGothic\_B



# 追加のフォントサポート (2022年3月)

Design Automation for Revit で一般的に利用されるフォント  
を追加でプリインストール

- ARIALN
- CenturyGothic
- GOST Common Italic
- GOST Common
- ISOCPEUR
- Meiryo**
- Meiryo-Bold
- Meiryo-BoldItalic
- Meiryo-Italic
- MeiryoUI**
- MeiryoUI-Bold
- MeiryoUI-BoldItalic
- MeiryoUI-Italic
- MSGothic\_M
- MSGothicJap**
- MS-Mincho\_M
- MSMinchoJap**
- MSPGothicJap**
- MSPMinchoJap**
- NanumGothic-Bold
- NanumGothic-ExtraBold
- NanumGothic-Regular
- NLCS-ISO
- NLCS-RW
- TitilliumWeb-Black
- TitilliumWeb-Bold
- TitilliumWeb-BoldItalic
- TitilliumWeb-ExtraLight
- ~省略~

# Design Automation for Revit での DWG 出力結果

文字

内容	
文字スタイル	MS Pゴシック_B
異尺度対応	いいえ
位置合わせ	左上
方向	水平
文字の高さ	5.5570
回転角度	0
行間隔の尺度	1.0000
行間隔の距離	9.2617
行間隔のスタイル	固定
背景マスク	いいえ
定義幅	109.5714
定義高さ	0.0000
段組み	なし
文字枠	いいえ

ジオメトリ

位置 X	3215.3418
位置 Y	5508.4087

/ モデル Layout1 Layout2 / + /

MS Pゴシック\_B



注：カスタムフォントを別途アップロードして  
インストールすることはできません。

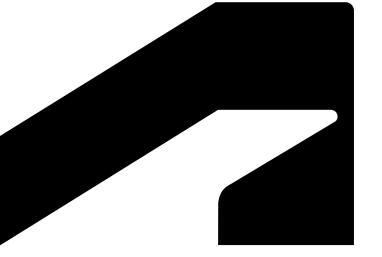
# Design Automation for Revit での DWG 出力結果

Yu Gothic UI



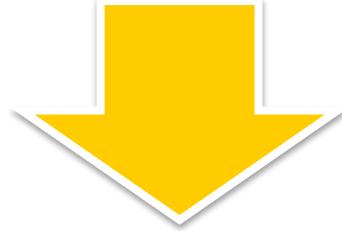
游ゴシック





**PDF 生成のサポート**

# Revit 2022 - PDF 書き出し

- Revit 2022 から 2D ビューとシートをPDF ファイルに書き出すことができるようになりました。
  - 単一ページを書き出したり、複数のビュー やシートを PDF ファイルに結合することもできます。
  - Design Automation API for Revit でも PDF を書き出すことができるようになりました。
- 



# Revit 2022 で書き出した MS P ゴシック

## ■ 設計概要書

■工事名 フリガナ（仮称）サンプル意匠工事

■建築主 住所 電話

フリガナ

(会社名等) 氏名 代表者名

■敷地概要 地名地番

(住居表示)

都市計画区域の  
内外の別等 ●都市計画区域内(●市街化区域 ○市街化調整区域 ○区域区分非設定)

○準都市計画区域内 ○都市計画区域及び準都市計画区域外

防火地域 ●防火地域 ○準防火地域 ○指定なし

その他の区域、地域、地区又は街区(○法第 22 条区域)

高度地区 なし

日影規制(隣地の規制) なし

道路(m) 幅員 22m

## ■※床面積表 (単位:m<sup>2</sup>)

階	延床面積			
	容積対象面積		容積対象外面	
	A:専有面積	B:共有面積	C:駐車場	D:EV
7F	640.55 m <sup>2</sup>	159.85 m <sup>2</sup>		17
6F	640.55 m <sup>2</sup>	159.85 m <sup>2</sup>		17
5F	640.55 m <sup>2</sup>	159.85 m <sup>2</sup>		17
4F	640.55 m <sup>2</sup>	159.85 m <sup>2</sup>		17
3F	640.55 m <sup>2</sup>	159.85 m <sup>2</sup>		17
2F	640.55 m <sup>2</sup>	159.85 m <sup>2</sup>		17
1F	164.83 m <sup>2</sup>	336.16 m <sup>2</sup>	208.14 m <sup>2</sup>	17
合計	4008.13 m <sup>2</sup>	1295.26 m <sup>2</sup>	208.14 m <sup>2</sup>	120

# Design Automation で書き出した MS P ゴシック

## ■ 設計概要書

■工事名 フリガナ（仮称）サンプル意匠工事

■建築主 住所 電話

フリガナ

(会社名等) 氏名 代表者名

■敷地概要 地名地番

(住居表示)

都市計画区域の  
内外の別等 ●都市計画区域内(●市街化区域 ○市街化調整区域 ○区域区分非設定)

○準都市計画区域内 ○都市計画区域及び準都市計画区域外

防火地域 ●防火地域 ○準防火地域 ○指定なし

その他の区域、地域、地区又は街区(○法第22条区域)

高度地区 なし

日影規制(隣地の規制) なし

道路(m) 幅員

22m

■※床面積表 (単位:m<sup>2</sup>)

階	延床面積			
	容積対象面積		容積対象外面	
	A:専有面積	B:共有面積	C:駐車場	D:EV
7F	640.55m <sup>2</sup>	159.85m <sup>2</sup>		17
6F	640.55m <sup>2</sup>	159.85m <sup>2</sup>		17
5F	640.55m <sup>2</sup>	159.85m <sup>2</sup>		17
4F	640.55m <sup>2</sup>	159.85m <sup>2</sup>		17
3F	640.55m <sup>2</sup>	159.85m <sup>2</sup>		17
2F	640.55m <sup>2</sup>	159.85m <sup>2</sup>		17
1F	164.83m <sup>2</sup>	336.16m <sup>2</sup>	208.14m <sup>2</sup>	17
合計	4008.13m <sup>2</sup>	1295.26m <sup>2</sup>	208.14m <sup>2</sup>	120

# Revit 2022 で書き出した Meiryo UI

■ 設計概要書				
■ 工事名 フリガナ (仮称) サンプル意匠工事				
■ 建築主 住所 電話				
フリガナ				
(会社名等) 氏名		代表者名		
■ 敷地概要 地名地番				
(住居表示)				
都市計画区域の 内外の別等	<input checked="" type="radio"/> 都市計画区域内 ( <input checked="" type="radio"/> 市街化区域 <input type="radio"/> 市街化調整区域 <input type="radio"/> 区域区分非設定) <input type="radio"/> 準都市計画区域内 <input type="radio"/> 都市計画区域及び準都市計画区域外			
防火地域	<input checked="" type="radio"/> 防火地域 <input type="radio"/> 準防火地域 <input type="radio"/> 指定なし			
その他の区域、地域、地区又は街区 ( <input type="radio"/> 法第22条区域)				
高度地区	なし			
日影規制 (隣地の規制)	なし			
道路(m)	幅員	22m		

■ ※床面積表 (単位:m<sup>2</sup>)

階	延床面積			
	容積対象面積		容積対象外面積	
	A:専有面積	B:共有面積	C:駐車場	D:その他
7F	640.55m <sup>2</sup>	159.85m <sup>2</sup>		17
6F	640.55m <sup>2</sup>	159.85m <sup>2</sup>		17
5F	640.55m <sup>2</sup>	159.85m <sup>2</sup>		17
4F	640.55m <sup>2</sup>	159.85m <sup>2</sup>		17
3F	640.55m <sup>2</sup>	159.85m <sup>2</sup>		17
2F	640.55m <sup>2</sup>	159.85m <sup>2</sup>		17
1F	164.83m <sup>2</sup>	336.16m <sup>2</sup>	208.14m <sup>2</sup>	17
合計	4008.13m <sup>2</sup>	1295.26m <sup>2</sup>	208.14m <sup>2</sup>	120

# Design Automation で書き出した Meiryo UI

## ■ 設計概要書

■工事名 フリガナ (仮称) サンプル意匠工事

■建築主 住所 電話

フリガナ

(会社名等) 氏名 代表者名

■敷地概要 地名地番

(住居表示)

都市計画区域の  
内外の別等 ●都市計画区域内 (●市街化区域 ○市街化調整区域 ○区域区分非設定)

○準都市計画区域内 ○都市計画区域及び準都市計画区域外

防火地域 ●防火地域 ○準防火地域 ○指定なし

その他の区域、地域、地区又は街区 (○法第22条区域)

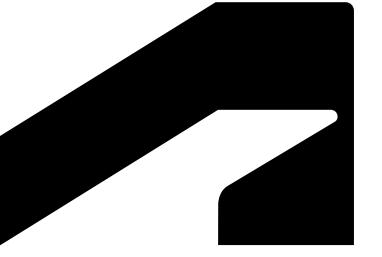
高度地区 なし

日影規制 (隣地の規制) なし

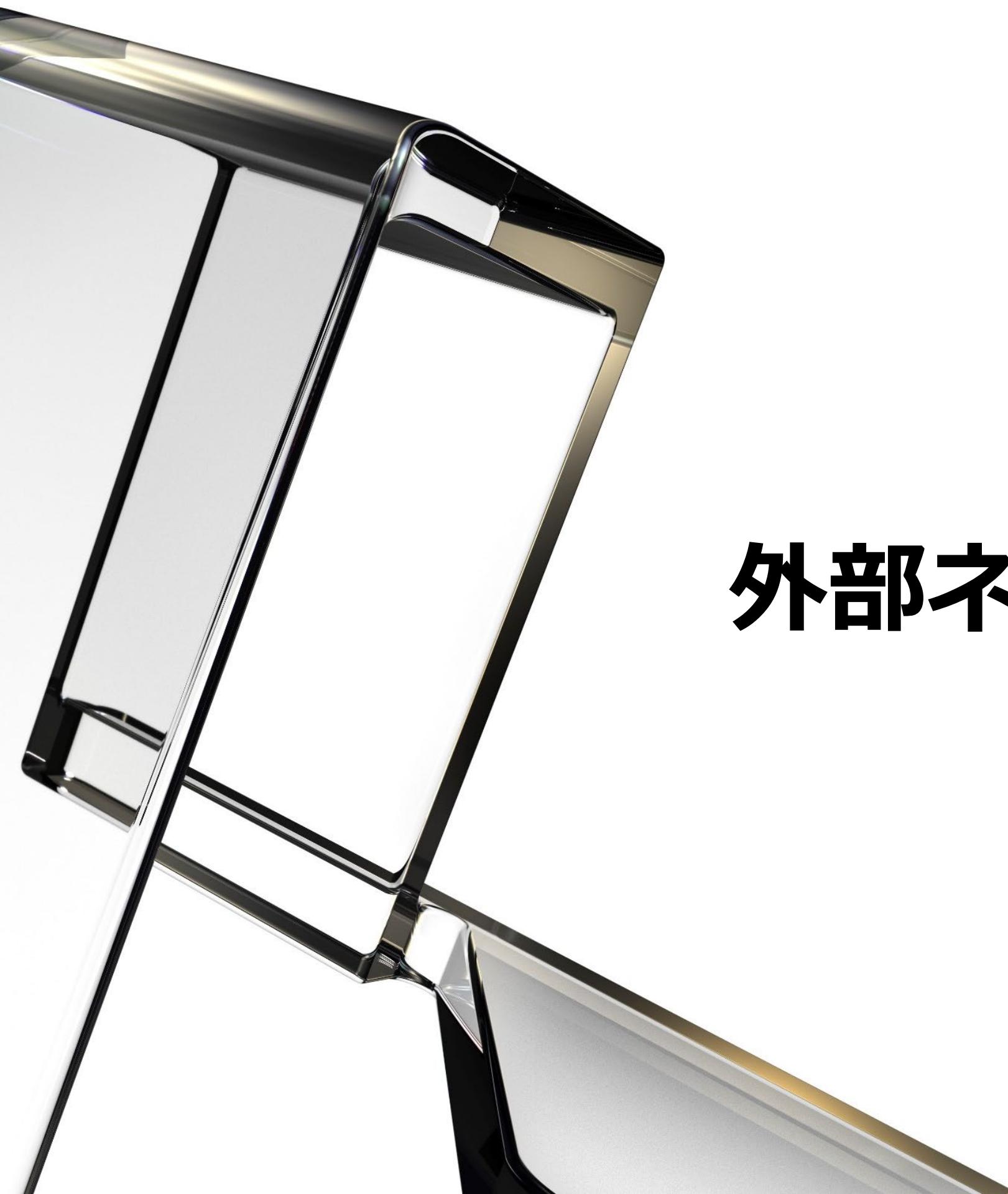
道路(m) 幅員 22m

■※床面積表 (単位:m<sup>2</sup>)

階	延床面積			
	容積対象面積		容積対象外面積	
	A:専有面積	B:共有面積	C:駐車場	D:E
7F	640.55m <sup>2</sup>	159.85m <sup>2</sup>		17
6F	640.55m <sup>2</sup>	159.85m <sup>2</sup>		17
5F	640.55m <sup>2</sup>	159.85m <sup>2</sup>		17
4F	640.55m <sup>2</sup>	159.85m <sup>2</sup>		17
3F	640.55m <sup>2</sup>	159.85m <sup>2</sup>		17
2F	640.55m <sup>2</sup>	159.85m <sup>2</sup>		17
1F	164.83m <sup>2</sup>	336.16m <sup>2</sup>	208.14m <sup>2</sup>	17
合計	4008.13m <sup>2</sup>	1295.26m <sup>2</sup>	208.14m <sup>2</sup>	120

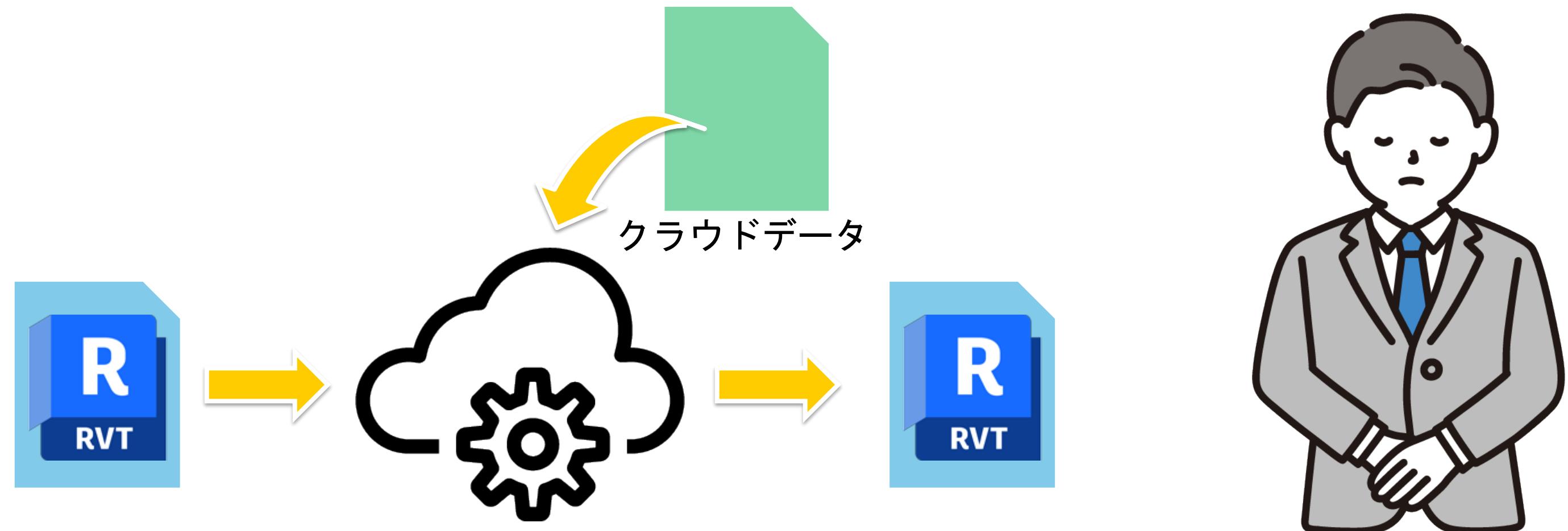


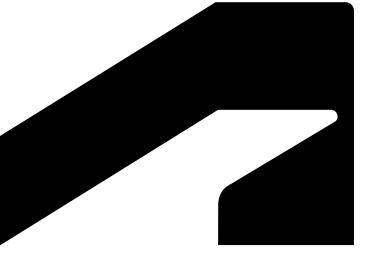
**外部ネットワークとの接続**



# Open Networkについて

- WorkItem の実行中に外部ネットワークにアクセスする方法。
- 現時点では公式ドキュメントはありません。
- お客様の想定されているワークフローをご提示いただき、本社開発チームに確認を取らせていただきます。

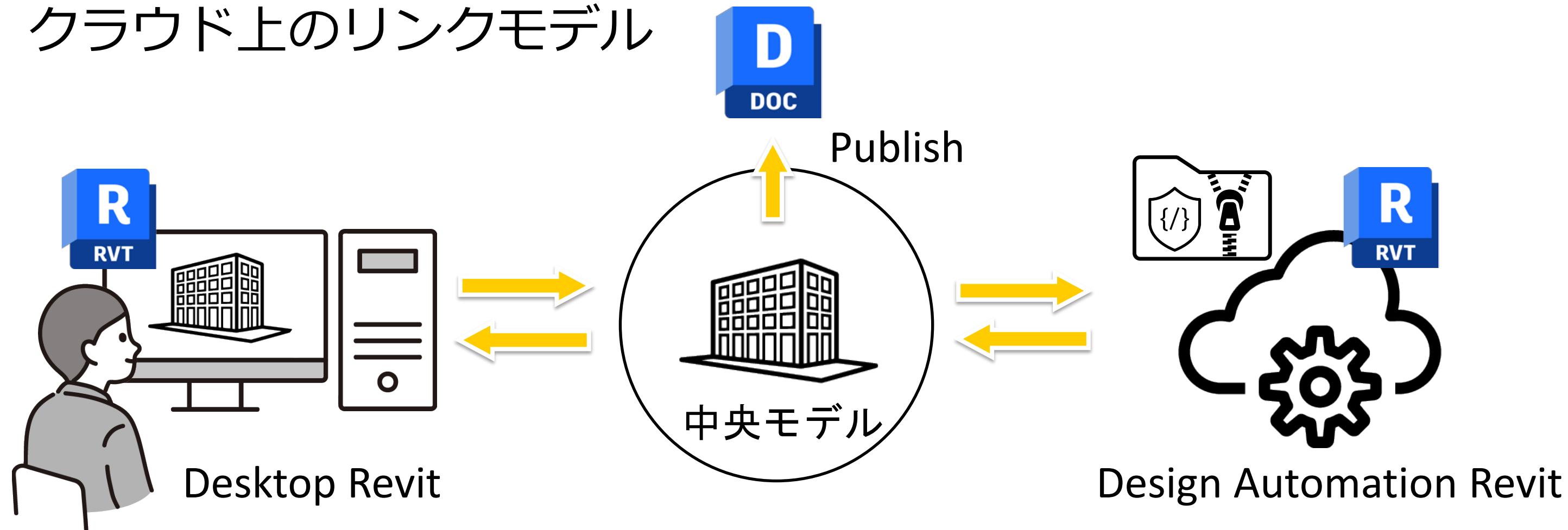




**Revit Cloud Model サポート**

# Revit Cloud Model サポート

- ワークシェアされていないクラウドモデルと、ワークシェアされているクラウドモデルのどちらもサポート。
- Revit 2022, 2023 対象
- ファイルを開く、保存、編集、チェックアウト、中央モデルと同期
- クラウド上のリンクモデル



# Revit Cloud Model を開く

- AppBundle のアドインからクラウドモデルを直接開くことで、モデルへのアクセス処理が大幅に簡素化されます。
- WorkItem リクエスト時の入力モデルの URL 設定は不要に。
- Region、ProjectGuid、ModelGuid の情報を渡します。

```
var cloudModelPath = ModelPathUtils.ConvertCloudGUIDsToCloudPath(  
    inputParams.Region,  
    inputParams.ProjectGuid,  
    inputParams.ModelGuid);  
  
Document doc = rvtApp.OpenDocumentFile(cloudModelPath, new OpenOptions());
```

# Revit Cloud Model を保存する/同期する

- 出力する Revit モデルをストレージの URL を指定してアップロードする必要はありません。
- 以下のコードを使用して、編集内容を直接クラウドモデルに保存することができます。

```
if (doc.IsWorkshared) // work-shared/C4R model
{
    SynchronizeWithCentralOptions swc = new SynchronizeWithCentralOptions();
    swc.SetRelinquishOptions(new RelinquishOptions(true));
    doc.SynchronizeWithCentral(new TransactWithCentralOptions(), swc);
}
else
{
    // Single user cloud model
    doc.SaveCloudModel();
}
```

# 中央モデルと同期後のパブリッシュ

- 中央モデルに変更内容を同期した後、モデルをパブリッシュする必要がある場合
  - onComplete() コールバックから Data Management API の Publish Model Command を使ってパブリッシュすることができます。
  - "commands:autodesk.bim360:C4RModelPublish"

**POST** [https://developer.api.autodesk.com/data/v1/projects/:project\\_id/commands](https://developer.api.autodesk.com/data/v1/projects/:project_id/commands)

<https://forge.autodesk.com/en/docs/data/v2/reference/http/PublishModel/>

# Revit Cloud Model を作成する

- AppBundle (Revit アドイン) から直接、新規にワークシェアリングモデルを作成することができます。
- モデルのセットアップ作業を自動化したい場合に非常に便利です。

```
Document newDoc = data.RevitApp.NewProjectDocument(UnitSystem.Imperial);

var cloudModelLocation = CloudModelLocation.Parse("CloudModelLocation.json");

newDoc.EnableWorksharing("Shared Levels and Grids", "Workset1");

newDoc.SaveAsCloudModel(
    cloudModelLocation.AccountId,
    cloudModelLocation.ProjectId,
    cloudModelLocation.FolderId,
    "newRCWModel.rvt");
```

# Revit Cloud Model API 利用の条件

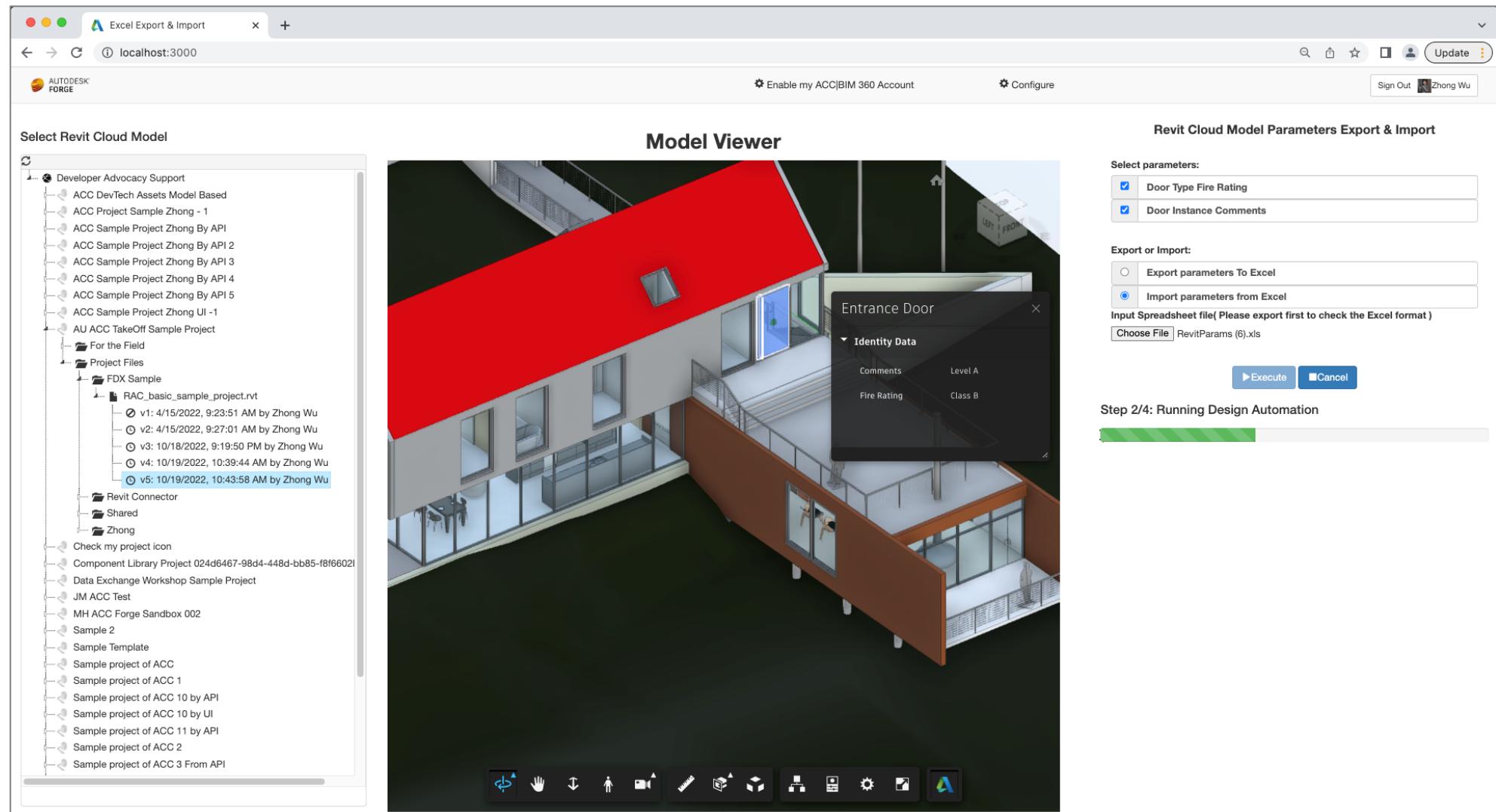
- WorkItem リクエスト時に、scope=code:all を指定して取得した 3-legged アクセストークンを adsk3LeggedToken プロパティで渡します。
  - これにより Design Automation Revit でクラウドモデルへのアクセスが有効になります。

```
{  
    inputXls: {  
        url: "XXXXXXXXXXXX"  
    },  
    onComplete: {  
        verb: "post",  
        url: designAutomation.webhook_url  
    },  
    adsk3LeggedToken: access_token  
}
```



# サンプル

- design.automation-nodejs-revit.rcw.parameters.xlsx

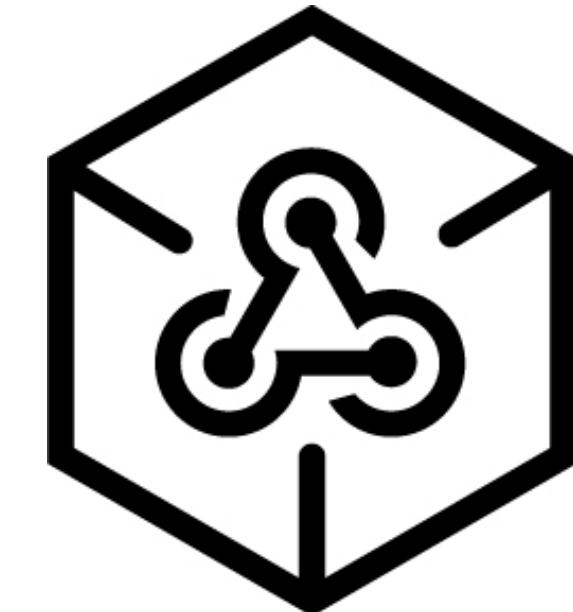


Design Automation API Supports Revit Cloud Model

New sample to support access Revit Cloud Model from Revit Design Automation Engine

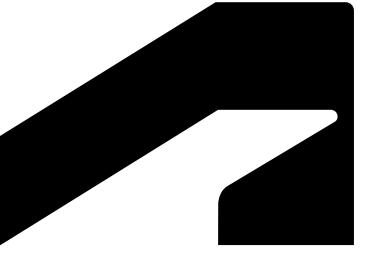
# 補足 : WebHooks API アップデート

- ACC/BIM 360 にモデルをアップロードした際、自動的に Model Derivative サービスが実行されて SVF/SVF2 変換が実行されます。
- この処理の完了通知を受け取れるようになりました。
- WorkItem 完了後に Viewer で表示するアプリを作成する場合に便利です。
- SVF 変換後、SVF2 変換後の 2 回呼び出されます。
- event (監視対象のイベントの種類を指定する文字列)
  - extraction.finished : 変換処理終了後に通知
  - extraction.updated : 変換処理過程を通知



[Webhook now supports translation progress of BIM360 or ACC files](#)

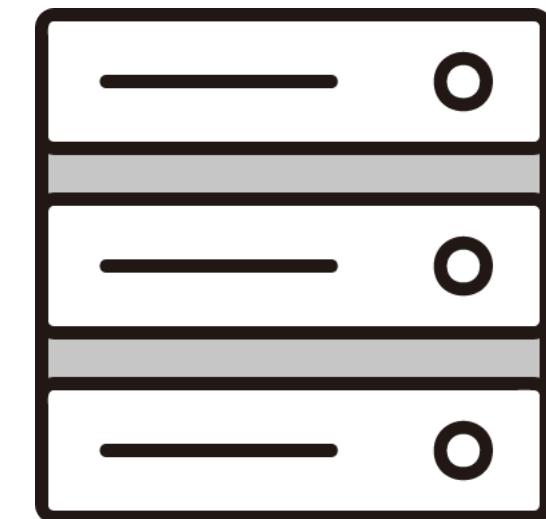
[Forge Online - WebHooks : イベント通知と活用](#)



# WorkItem の処理時間

# Revit エンジンの動作環境

- OS は **Windows Server 2019 英語版**
- メモリは 最大 **32GB**
- CPU 処理能力はデスクトップ PC と比較するとやや劣ります。



# Design Automation API の課金に含まれる処理

- Design Automation API の課金とコスト算出について
  - 課金時間は、WorkItem 処理の成功時の処理時間（秒数）に基づいて算出されています。
  - WorkItem が実行時に使用する入力ファイルのダウンロードと、出力ファイルのアップロードにかかる時間も対象です。
  - ネットワークのトラフィック等の影響も。





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.