



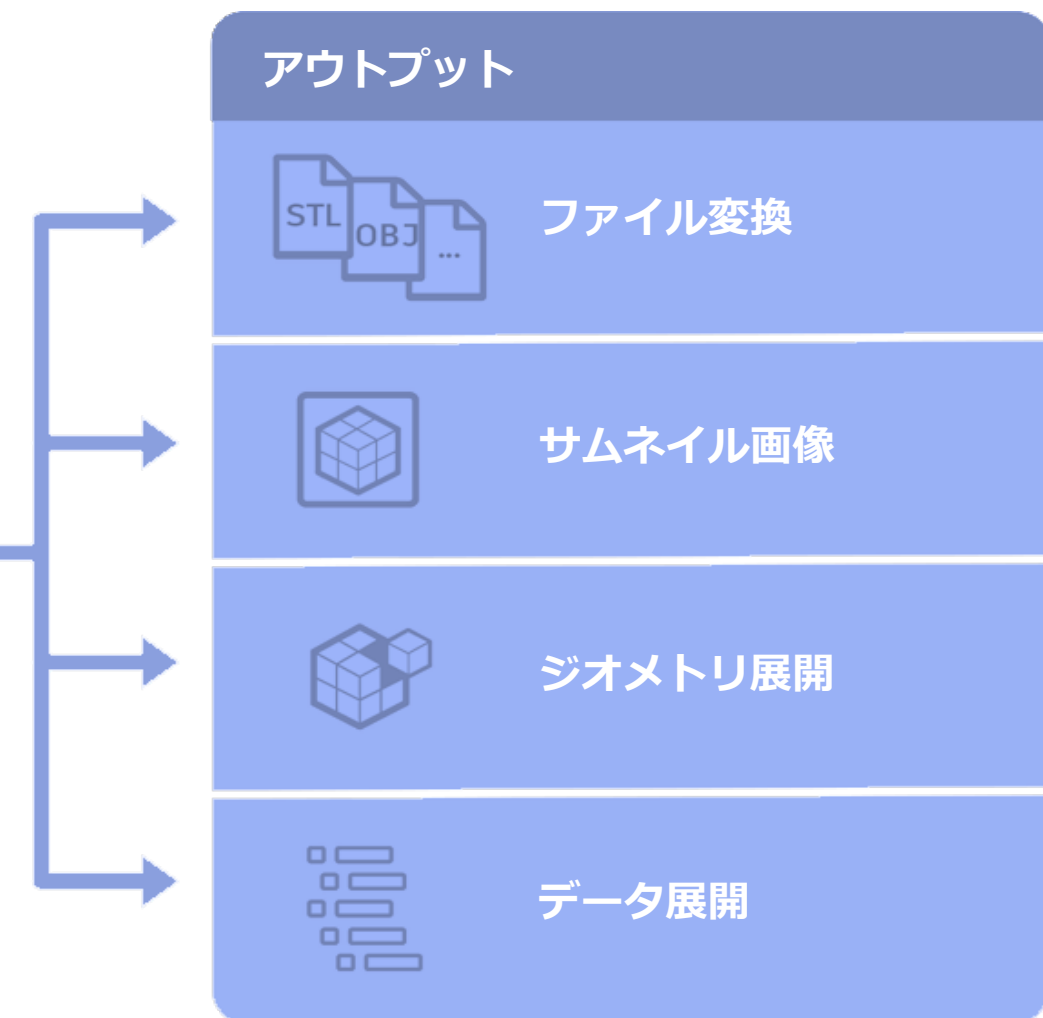
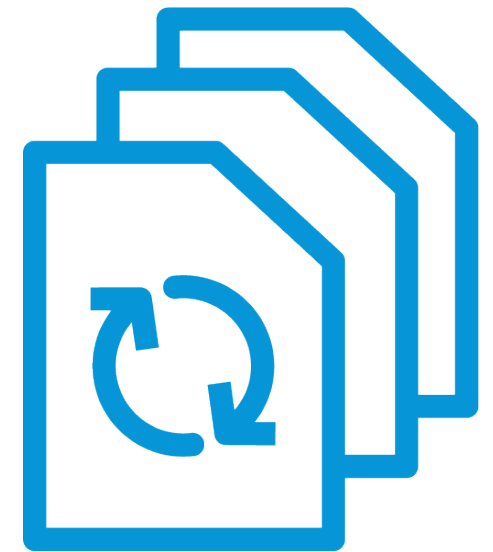
# Model Derivative API アップデート Advanced オプション

伊勢崎俊明

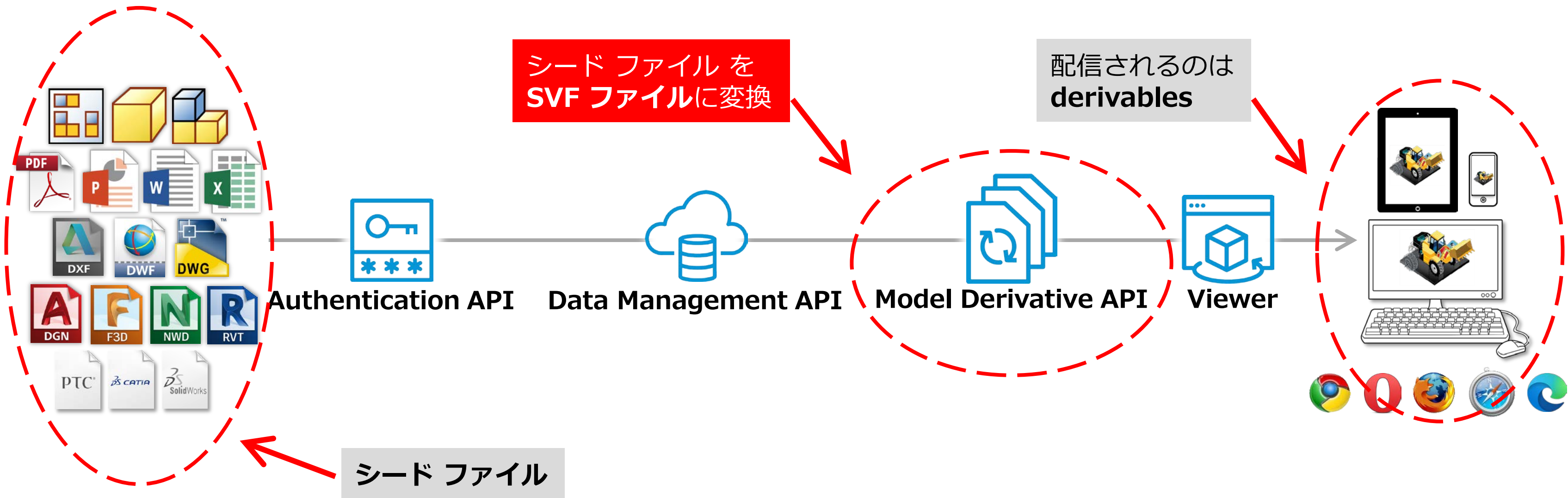
オートデスク 株式会社

# Model Derivative API の役割

- デザイン ファイルを変換
  - 他のデザイン ファイル形式に変換
  - Viewer 用に SVF 形式に変換してブラウザで表示
  - ジオメトリ データやモデル階層の展開
  - サムネイル画像の生成
  - RESTful API



# Forge Viewer ソリューションの流れ



# SVF 変換時の変換オプション



Model Derivative API v2

> Developer's Guide

> Step-by-Step Tutorials

> Code Samples

▼ API Reference

▼ HTTP Specification

▼ Derivatives

GET formats

POST job

POST references

GET :urn/thumbnail

GET :urn/manifest

DELETE :urn/manifest

GET :urn/manifest/:derivativeurn

HEAD :urn/manifest/:derivativeurn

GET :urn/metadata

## Attributes by Output Type

Case 3: Input file type is Navisworks

## Navisworks Timeliner 情報の出力指定

### timelinerProperties

*bool*

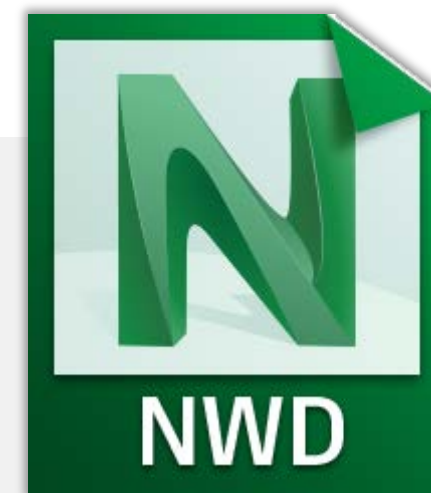
An option to be specified when the input file type is Navisworks.

**true** : Extract timeliner properties.

**false** : **(Default)** Do not extract timeliner properties.

# リクエスト パラメータ timelinerProperties

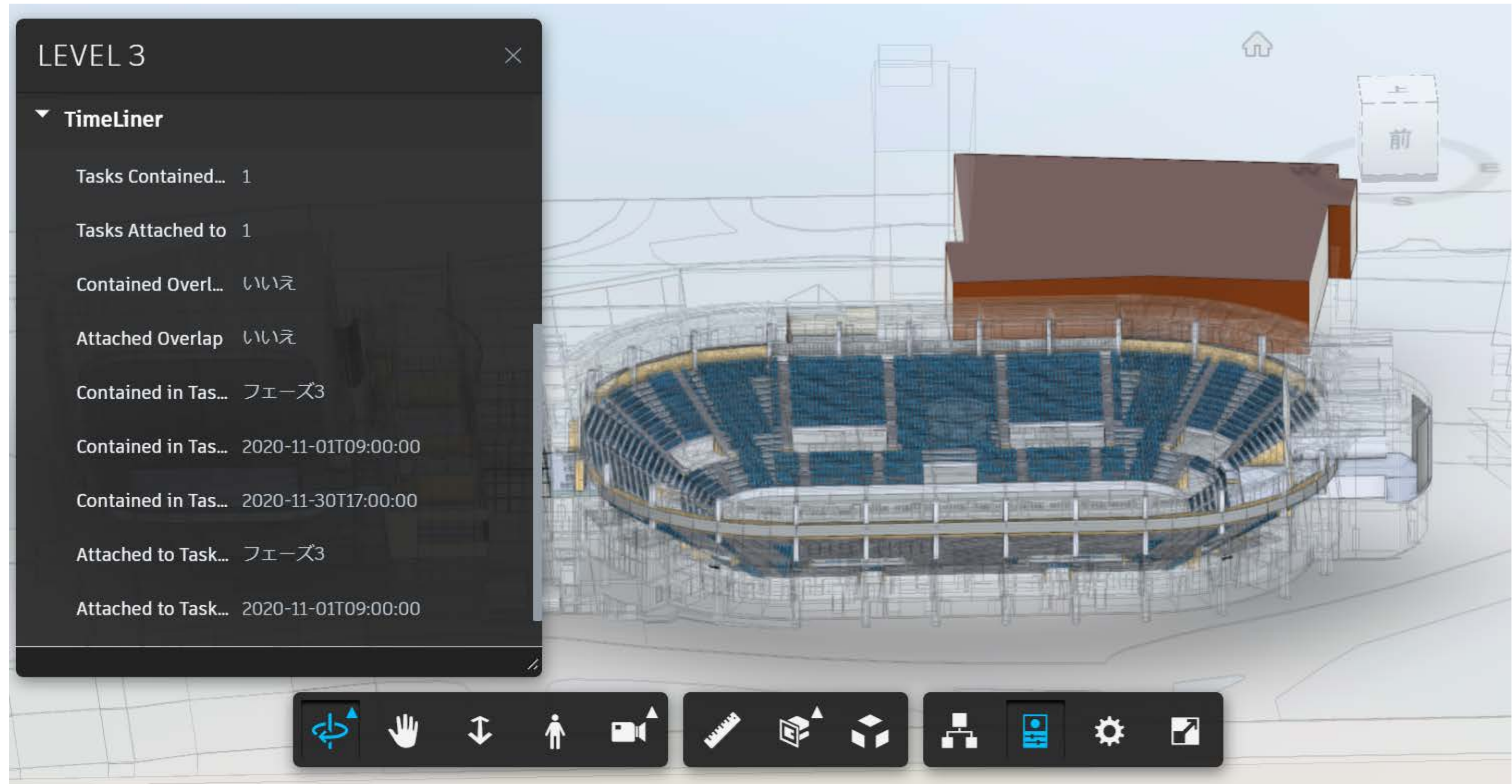
- POST job のリクエスト ボディ



```
{
  "input": {
    "urn": <urn>
  },
  "output": {
    "formats": [
      {
        "type": "svf",
        "views": [
          "2d",
          "3d"
        ],
        "advanced": {
          "timelinerProperties": "true"
        }
      }
    ]
  }
}
```



# Timeliner (NWD ファイルのみ)



# SVF 変換時の変換オプション



Model Derivative API v2

> Developer's Guide

> Step-by-Step Tutorials

> Code Samples

▼ API Reference

▼ HTTP Specification

▼ Derivatives

GET formats

POST job

POST references

GET :urn/thumbnail

GET :urn/manifest

DELETE :urn/manifest

GET :urn/manifest/:derivativeurn

HEAD :urn/manifest/:derivativeurn

GET :urn/metadata

GET :urn/metadata/:guid

Attributes by Output Type

SVF Output

## Revit 「部屋」情報の出力指定

**generateMasterViews**

*bool*

An option to be specified when the input file type is Revit.

Generates master views when translating from the Revit input format to SVF. This option is ignored for all other input formats. This attribute defaults to `false`.

Master views are 3D views that are generated for each phase of the Revit model. A master view contains all elements (including “room” elements) present in the host model for that phase. The display name of a master view defaults to the name of the phase it is generated from. However, if a view with that name already exists, the Model Derivative service appends a suffix to the default display name.

### Notes:

1. Master views do not contain elements from linked models.
2. Enabling this option can increase the time it takes to translate the model.

# リクエスト パラメータ generateMasterViews

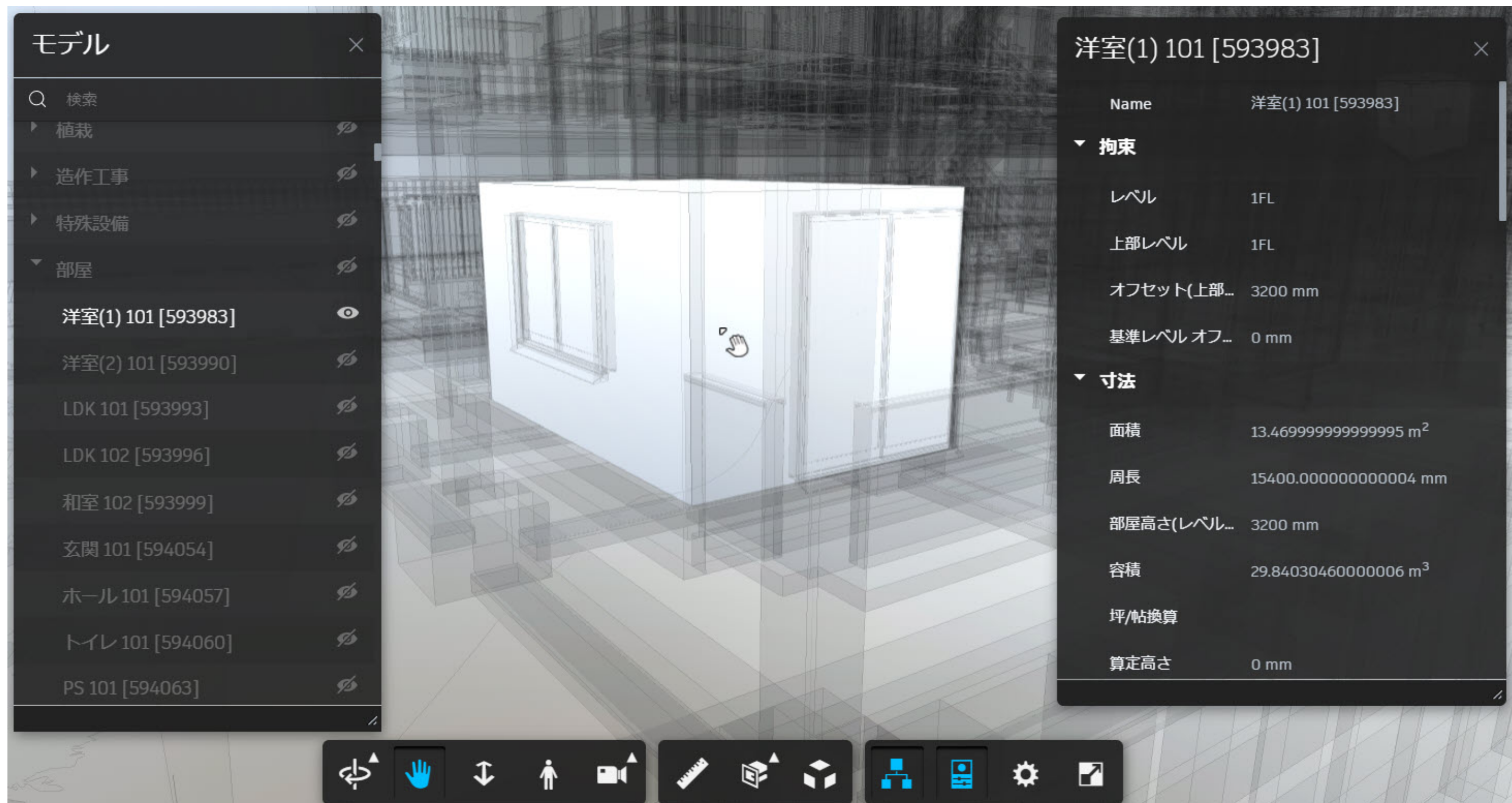
- POST job のリクエスト ボディ



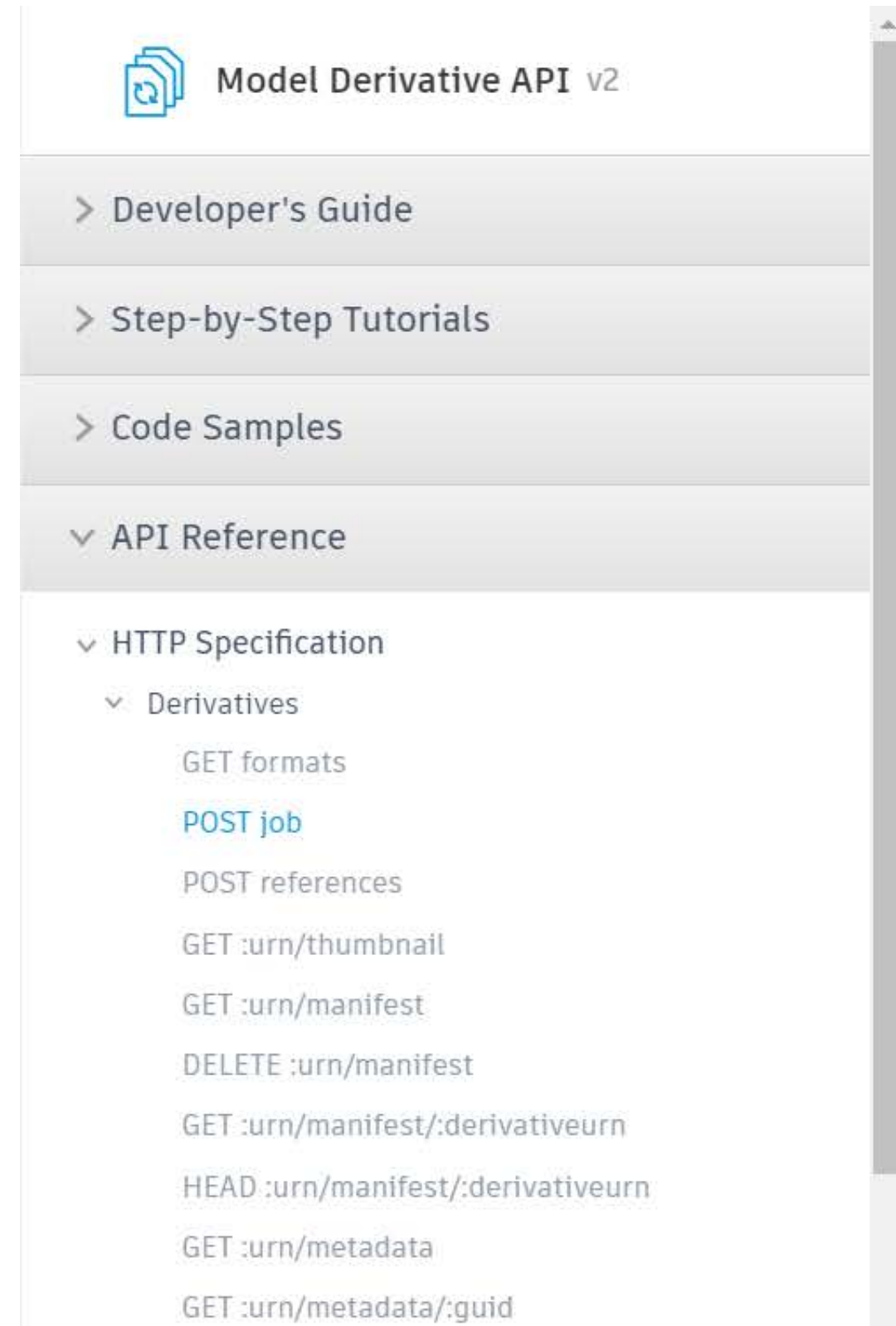
```
{
  "input": {
    "urn": <urn>
  },
  "output": {
    "formats": [
      {
        "type": "svf",
        "views": [
          "2d",
          "3d"
        ],
        "advanced": {
          "generateMasterViews": "true"
        }
      }
    ]
  }
}
```



# 部屋とスペース (RVT ファイルのみ)



# SVF 変換時の変換オプション



## Attributes by Output Type

### SVF Output

## IFC 変換時の使用ロジック指定 (Navisworks方式かRevit方式か)

### 旧オプション：非推奨

#### switchLoader

bool

**Deprecated** Switches the IFC loader from the Navisworks IFC loader to the new Revit IFC loader, when translating from the IFC input format to SVF. This attribute defaults to `false`.

#### conversionMethod

string

An option to be specified when the input file type is IFC. Specifies what IFC loader to use during translation. Available options are:

- `legacy` - Use the Navisworks IFC loader.
- `modern` - Use the newer Revit IFC loader.

If both `switchLoader` and `conversionMethod` are specified, Model Derivative uses the `conversionMethod` parameter. If `conversionMethod` is not specified, Model Derivative uses the `switchLoader` parameter.

### 新オプション：推奨

# リクエスト パラメータ conversionMethod

- POST job のリクエスト ボディ



```
{
  "input": {
    "urn": <urn>
  },
  "output": {
    "formats": [
      {
        "type": "svf",
        "views": [
          "2d",
          "3d"
        ],
        "advanced": {
          "conversionMethod": "modern"
        }
      }
    ]
  }
}
```

"modern":Revit 方式

"legacy":Navisworks 方式




# Model Derivative API アップデート メタデータと差分検出

伊勢崎俊明

オートデスク 株式会社



# SVF 変換で得られるメタデータの利用

 Model Derivative API v2

▼ Developer's Guide

Overview

▼ API Basics

About this API

Translate Models

Extract Metadata

Extract Geometry

Extract Thumbnails

GDPR Compliance

Webhooks

Field Guide

Supported Translations

> Rate Limits

> Step-by-Step Tutorials

> Code Samples

## Extract Metadata

When you translate a model into the SVF format, the Model Derivative service saves information about the derivatives (metadata) in the manifest. When the source model is large, the manifest can become difficult to parse. As such, the Model Derivative API provides endpoints to specifically query metadata.

[GET :urn/metadata](#) lets you extract information about the 3D Views and 2D sheets/views referenced in the manifest. These derivatives are the Viewables that you can typically display in a browser using the Forge Viewer SDK. See the tutorial on [Extract Metadata from a Source Model](#) to see how you can extract the names of Viewables and their metadata GUIDs (Global Unique Identifier).

Source models from applications such as Autodesk Inventor and Fusion 360 produce only one Viewable per model. However, source models from applications such as Autodesk Revit can contain multiple Viewables.

Once you obtain the GUID of a Viewable, you can use [GET :urn/metadata/:guid](#) to obtain the object/component hierarchy of the model. In addition to the hierarchy, the list provides the `objectId` of each object. See the tutorial on [Extract Geometry from a Source File](#) for a demonstration on how the object hierarchy and objectsids are used to uniquely identify geometry and thereafter extracted as OBJ files.

Using [GET :urn/metadata/:guid/properties](#) you can obtain a flat list of objects in that Viewable. It also returns the properties of each object. Using a query parameter, you can filter the results to provide the properties of one specific object.

The following image shows the object hierarchy and the properties of a selected object, as displayed in a browser using the Forge Viewer SDK.





# 派生データの取得 – その1

- ビュー上のオブジェクト階層メタデータの GUID を取得

**GET** :urn/metadata

Returns a list of model view (metadata) IDs for a design model. The metadata ID enables end users to select an object tree and properties for a specific model view.

Although most design apps (e.g., Fusion and Inventor) only allow a single model view (object tree and set of properties), some apps (e.g., Revit) allow users to design models with multiple model views (e.g., HVAC, architecture, perspective).

Note that you can only retrieve metadata from an input file that has been translated into an SVF file.

# 派生データの取得 – その2

- GUID からツリー構造やオブジェクト毎の情報を取得

**GET** :urn/metadata/:guid

Returns an object tree, i.e., a hierarchical list of objects for a model view.

To call this endpoint you first need to call the [GET :urn/metadata](#) endpoint, to determine which model view (object tree and set of properties) to use.

**GET** :urn/metadata/:guid/properties

Returns a list of [properties](#) for each object in an object tree. Properties are returned according to object ID and do not follow a hierarchical structure.

# Revit 派生データでのバージョン比較例

\* VS Code Forge Tools エクステンションを利用

The screenshot displays the VS Code interface with the VS Code Forge Tools extension. It compares two Revit files, 'Untitled-1' and 'Untitled-2', side-by-side. The left pane shows the JSON data for objectid 8692, and the right pane shows the data for objectid 8683. Both objects have the same name '鋼製\_一般枠\_引違い-二枚 [563046]' and the same externalId 'd9b3f277-b695-4a6f-a1a5-8d30ae36ba4e-00089766'. The properties are also identical, including 'IFC パラメータ' and '開勝手'.

The 'Snoop Objects' window at the bottom shows a 'FamilyInstance' with the dimensions '1800x1200 563046'. It also displays a table of fields and values:

Field	Value
UniqueId	d9b3f277-b695-4a6f-a1a5-8d30ae36ba4e-00089766
VersionGuid	940b4090-a921-4f99-a97e-14b064df1b30
ViewSpecific	False
WorksetId	< WorksetId >
--- Methods ---	
IsReadOnly	True

参考ブログ記事：[https://adndevblog.typepad.com/technology\\_perspective/2020/11/utilizeing-meta-data.html](https://adndevblog.typepad.com/technology_perspective/2020/11/utilizeing-meta-data.html)

# 参考 : model-compare 差分検出ツール

## ■ Go 言語実装の応用例

```

→ ~ model-compare
Check the difference between two models, identifying
  - changes in component hierarchy;
  - changes in property of a component;
  - changes in geometry of a component.

Usage:
  model-compare [command]

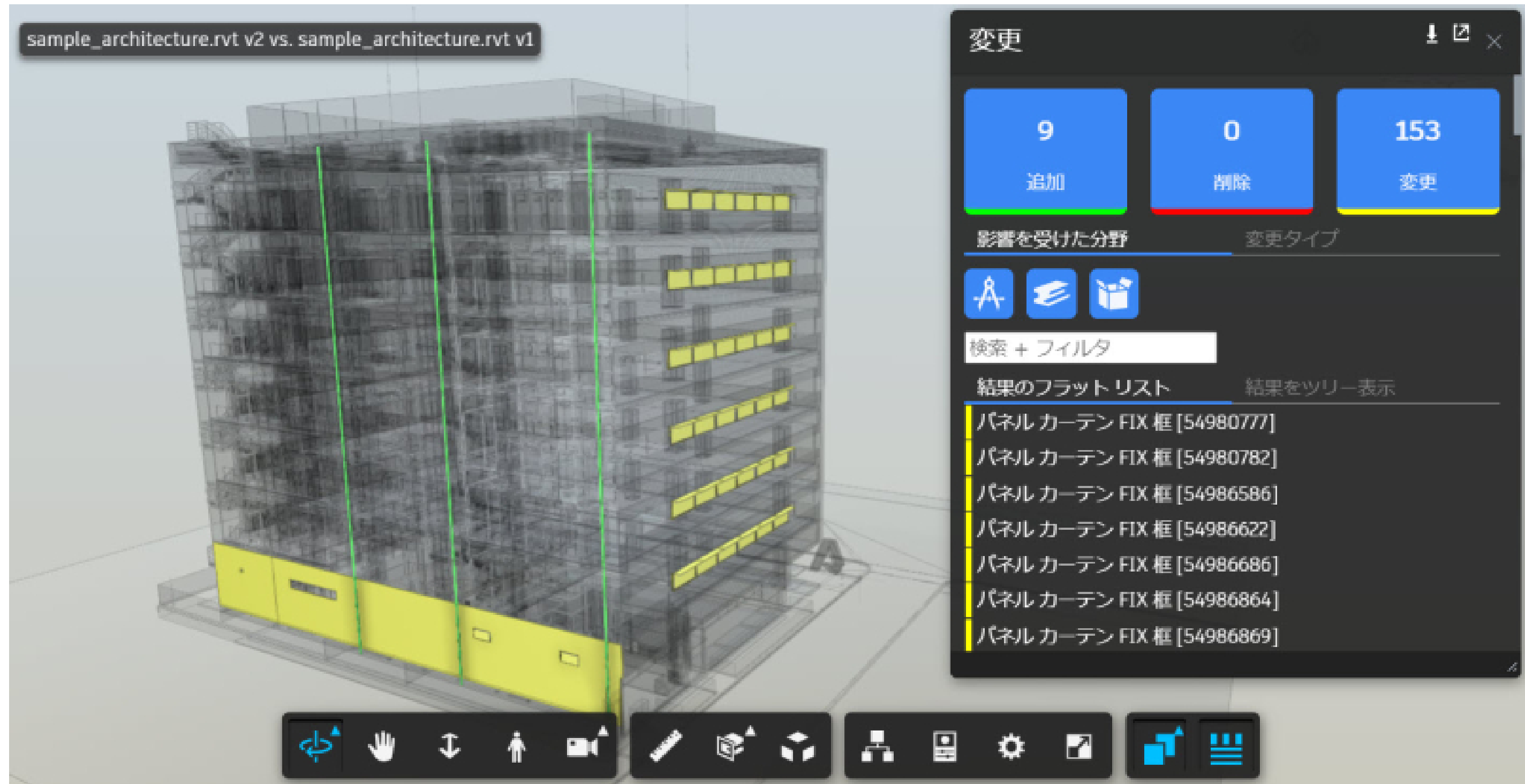
Available Commands:
  file      Compares translated models by providing the local files
  help      Help about any command
  urn       Compares translated models by providing their urns

Flags:
  -h, --help  help for model-compare

Use "model-compare [command] --help" for more information about a command.
→ ~ model-compare urn -h
Compare two translated files by providing their urns as follows:
  -> model-compare urn -t dXJuOmF...mplY3Q dXJuOmF...6YXU
will compare the model trees of the translated files and provide the following result:
=====
@ ["data","objects",0,"objects",0,"objects",1,"objects",2]
+ {"name":"RightSide2","objectid":10}
=====
meaning that compared with first node, the second one has an additional node

```

# 参考：difftool エクステンション







# Model Derivative API アップデート SVF2 による大規模モデル対応

伊勢崎俊明

オートデスク 株式会社

# 2021年1月現在 Public Beta : SVF2 形式

- Streaming Vector Format の正常進化バージョン
- 当初 OTG (Oscar The Grouch's) で紹介
- BIM 360 Docs 上で超大規模モデル表示用途で利用
- ジオメトリの共有利用、キャッシュ再利用による効果
- 少メモリ、Socket 通信による高速ロード、etc
- Model Derivative API で SVF2 を指定して変換が必要
- Forge Viewer 7.25 以上のバージョンで表示可能

# SVF2 (Public Beta) 利用に必要な処理

## 1. Model Derivative API での SVF2 変換

- デザイン ファイル変換時に SVF2 を指定が必要
  - デザイン ファイル ⇒ SVF2
  - 既存の変換済 SVF ⇒ SVF2

## 2. Viewer 実装の変更

- 初期化オプション api と env 属性の変更
- api: 'derivativeV2' ⇒ api: '**D3S**'
- env: 'AutodeskProduction' ⇒ env: '**MD20ProdUS**'

# SVF2 変換時のリクエスト ボディ

```
{
  "input": {
    "urn": "<Your Encoded URN>"
  },
  "output": {
    "formats": [
      {
        "type": "svf2",
        "views": [
          "2d",
          "3d"
        ]
      }
    ]
  }
}
```

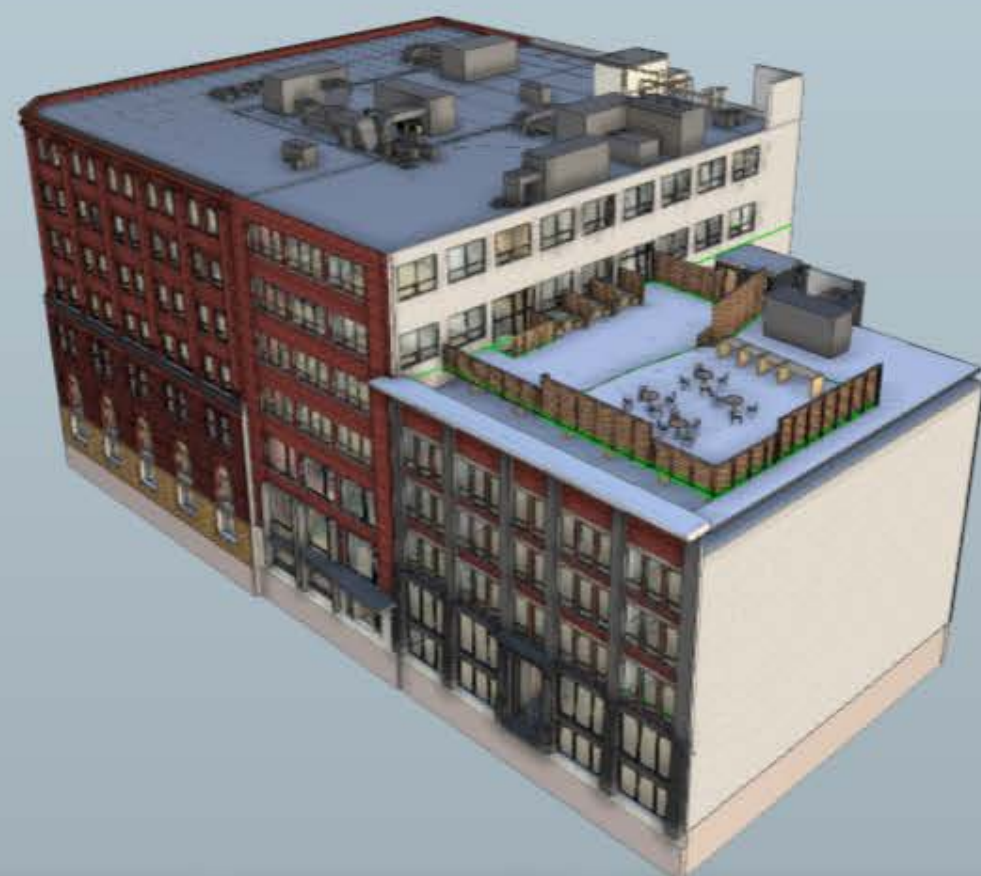
SVF : isSVF2()=undefined

Total geometry size: 108.47375679016113 MB

Number of meshes: 27796

Num Meshes on GPU: 10000

Net GPU geom memory used: 53333904



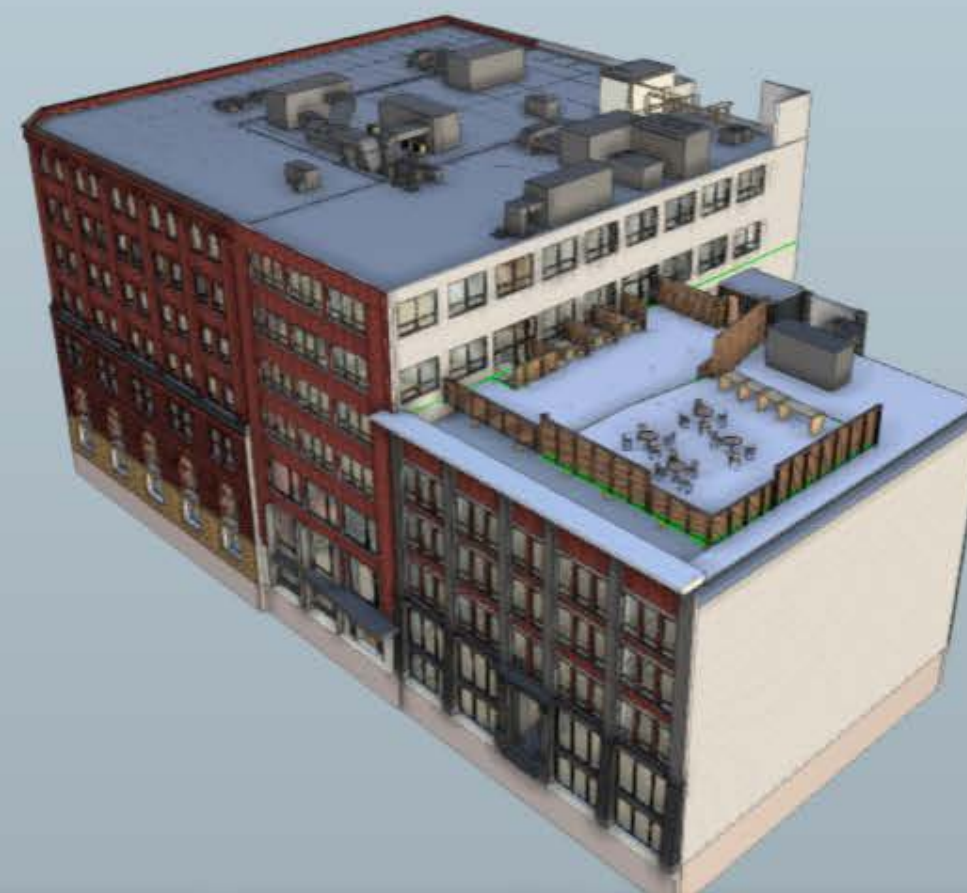
SVF2 : isSVF2()=true

Total geometry size: 16.585140228271484 MB

Number of meshes: 3040

Num Meshes on GPU: 3040

Net GPU geom memory used: 16369340





# SVF2 変換時のマニフェスト：デザイン ⇒ SVF2

```
{
  "urn": "<Your Encoded URN>",
  "derivatives": [
    {
      "hasThumbnail": "true",
      "children": [
        :
        "progress": "complete",
        "outputType": "svf2",
        "status": "success"
      ],
      :
    }
  ],
  "hasThumbnail": "true",
  "progress": "complete",
  "type": "manifest",
  "region": "US",
  "version": "1.0",
  "status": "success"
}
```

# SVF2 変換時のマニフェスト : SVF $\Rightarrow$ SVF2

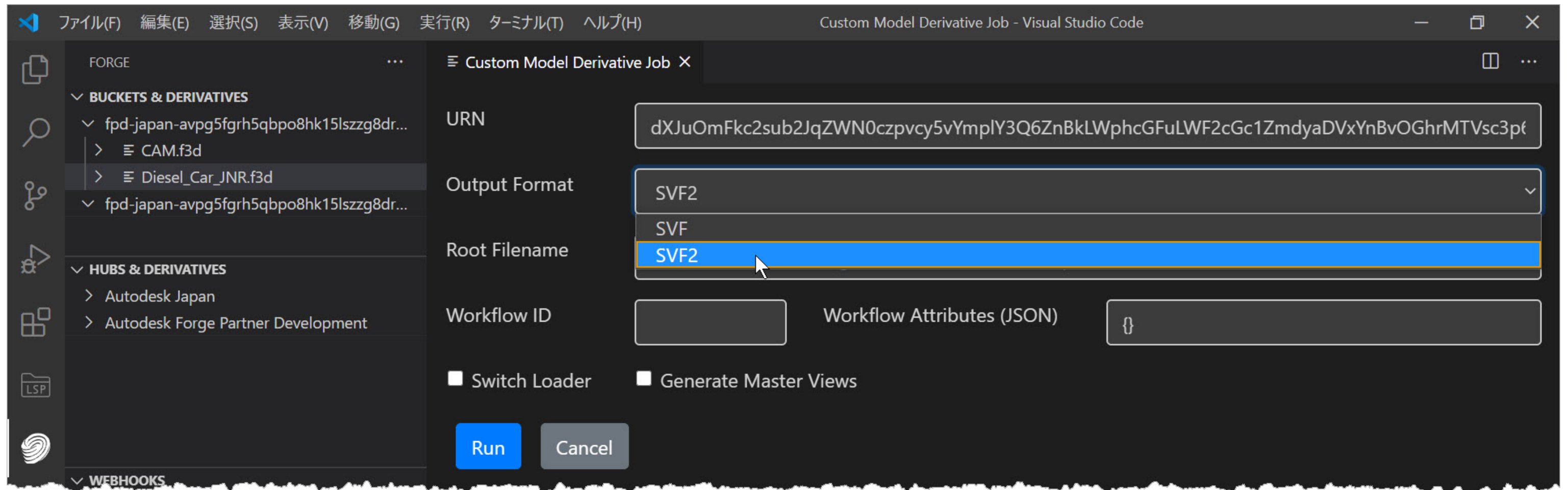
```
{
  "urn": "<Your Encoded URN>",
  "derivatives": [
    {
      "hasThumbnail": "true",
      "overrideOutputType": "svf2",
      "children": [
        :
        {
          "progress": "complete",
          "outputType": "svf",
          "status": "success"
        },
        :
      ]
    },
    :
  ],
  "hasThumbnail": "true",
  "progress": "complete",
  "type": "manifest",
  "region": "US",
  "version": "1.0",
  "status": "success"
}
```

# SVF2 利用時の Viewer 実装 (Public Beta)

```
var options = {  
  env: 'MD20ProdUS',  
  api: 'D3S',  
  getAccessToken: getCredentials  
};  
  
Autodesk.Viewing.Initializer(options, function () {  
  _viewer = new Autodesk.Viewing.GuiViewer3D(document.getElementById('viewer3d-1'));  
  var startedCode = _viewer.start();  
  if (startedCode > 0) {  
    console.error('Failed to create a 3D Viewer: WebGL not supported.');    return;  
  }  
  
  var documentId = 'urn:' + urn_svf;  
  Autodesk.Viewing.Document.load(documentId, onDocumentLoadSuccess, onDocumentLoadFailure);  
});
```

# VS Code Forge Tools での SVF2 対応

- バージョン 2.2.0 以降で SVF2 変換と表示をサポート





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 and services offerings, and specifications and pricing at any time without notice, and is not responsible for typographical or graphical errors that may appear in this document.

© 2018 Autodesk. All rights reserved.