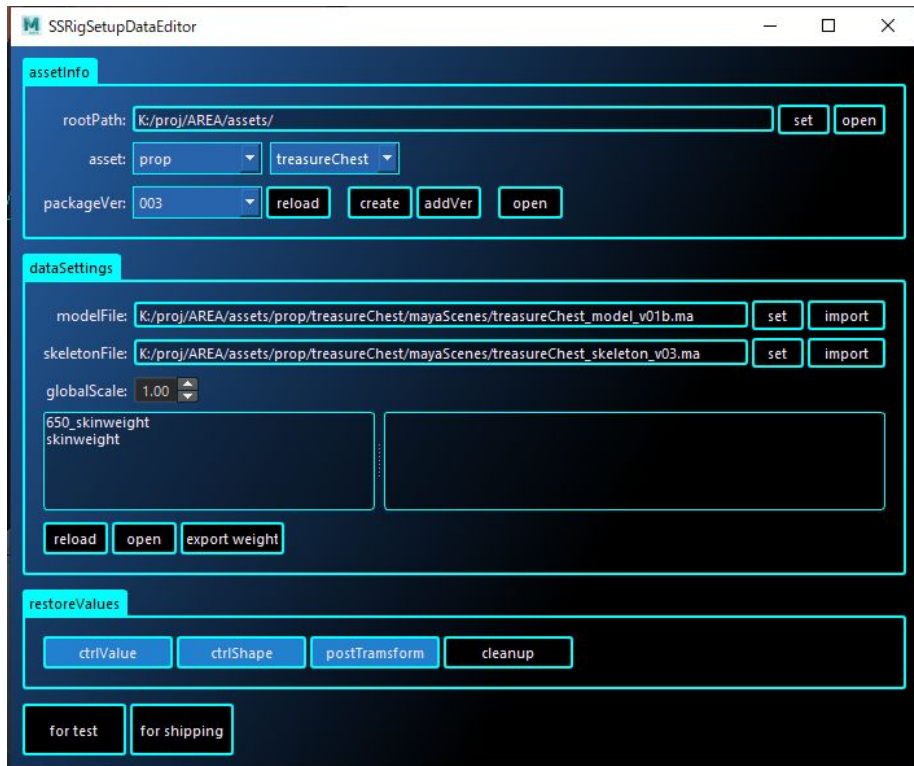


# SSRigSetupDataEditor



## 説明

リグのセットアップデータを管理するためのツールです。  
階層ルールなどは、次項を参照してください。

## 起動方法

SSRigフォルダをmayaのPYTHONPATHが通っている階層へ格納してください。

例)

maya標準でPYTHONPATHが通っているフォルダに格納する場合

C:/Users/kubo/Documents/maya/2025/scripts/

-> C:/Users/kubo/Documents/maya/2025/scripts/SSRig

下記のスクリプトを実行してください。

```
##SSRigSetupDataEditor-----  
from SSRig.gui import setupDataEditor  
setupDataEditor.callWindow()
```

# SSRigSetupDataEditor

rootPath

環境によって変わる

assetType

prop / character などアセットの分類毎

assetName

アセット毎のフォルダ

mayaScenes

mayaのデータ置き場 (ma / mb)

setupPackages

セットアップ情報・データ置き場

setupData\_dev\_v001

セットアップデータ置き場 ( weightデータ等)

weight

postTransform.json

allCtrlValue.json

allCtrlShape.json

setupInfo\_dev\_v001.json

セットアップ情報格納ファイル

ここから下は固定

# SSRigSetupDataEditor

setupPackages

setupData\_dev\_v001

weight

postTransform.json

allCtrlValue.json

allCtrlShape.json

setupData\_dev\_v002

setupInfo\_dev\_v001.json

setupInfo\_dev\_v002.json

## データフォルダ

-セットアップ再現に必要なデータ (weight/数値) を格納しておく

-バージョンに関しては jsonファイルと合わせておく。

-バージョンを上げる際には、  
現行のバージョンの中身を複製する

## jsonファイル

-使用している model/skeleton のパス情報と  
その他保持しておきたい情報を格納しておく。

-rootPathが環境によって変わっても問題ないように、  
rootPath以下のパス情報に変換しておく。

# SSRigSetupDataEditor

SSRigSetupDataEditor

**assetInfo**

rootPath:

asset:

packageVer:

**dataSettings**

modelFile:

skeletonFile:

globalScale:

650\_skinweight  
skinweight

**restoreValues**

## rootPath

環境に合わせてアセット管理階層を指定してください。

## asset

rootPath以下にある階層をリストします。  
対象となる assetType,assetNameを選択してください。

{rootPath}/{assetType}/{assetName}/

## packageVer

選択した assetType,assetName以下に格納されている  
パッケージ verをリストします。

{rootPath}/{assetType}/{assetName}/setupPackages/

## create

新規パッケージを作成します。

## addVer

現在選択している verを元にして、最新の verでパッケージを作成します。

## open

現在選択しているパッケージ verのフォルダを開きます。

# SSRigSetupDataEditor

SSRigSetupDataEditor

**assetInfo**

rootPath:

asset:

packageVer:

**dataSettings**

modelFile:

skeletonFile:

globalScale:

650\_skinweight  
skinweight

**restoreValues**

## modelFile / skeletonFile

使用するモデル・骨格データを登録してください。(drag&drop対応)  
import で現在登録しているファイルをシーンに importします。

## globalScale

セットアップの基本スケール値を設定できます。

## fileList

現在選択しているパッケージフォルダ以下に格納されているフォルダ (左)と  
その中のファイル (右)をリストします。

## open

現在選択しているフォルダを開きます

## export weight

選択しているメッシュの skinClusterをXMLに書き出します。

# SSRigSetupDataEditor

SSRigSetupDataEditor

**assetInfo**

rootPath:

asset:

packageVer:

**dataSettings**

modelFile:

skeletonFile:

globalScale:

650\_skinweight  
skinweight

**restoreValues**

## restoreValues

**for test** 実行時に読み込む情報・処理を一時的に変更できます

## ctrlValue

コントローラーの初期値

右クリックメニューで export できます。

## ctrlShape

コントローラーの形状

右クリックメニューで export できます。

## postTransform

全体移動・回転・スケール値

右クリックメニューで export できます。

## cleanup

ノードのロック等 cleanup 処理を実施します。

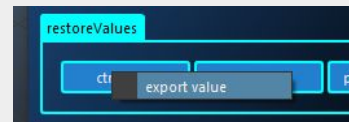
デフォルト OFF

## for test

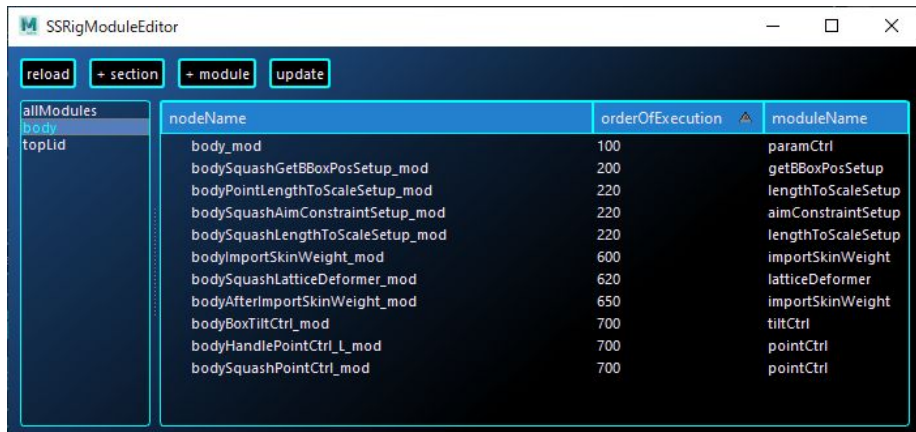
編集向けデータのビルドを行います。

## for shipping

出荷向けデータのビルドを行います。



# SSRigModuleEditor



## 説明

リグモジュールノードの作成と管理を行うツールです。

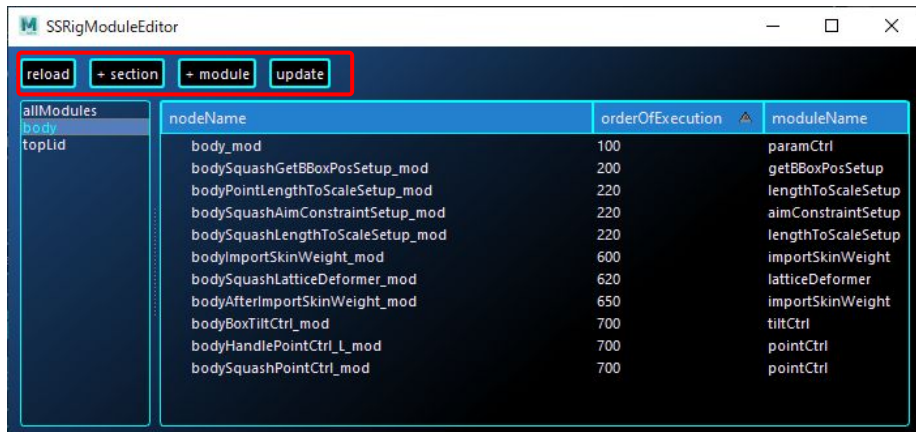
## 起動方法

SSRigフォルダをmayaのPYTHONPATHが通っている階層へ格納してください。

下記のスクリプトを実行してください。

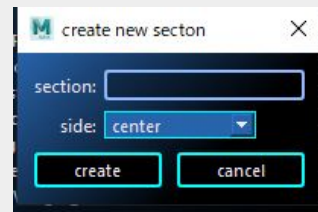
```
##SSRigModuleEditor-----  
from SSRig.gui import moduleEditor  
moduleEditor.callWindow()
```

# SSRigModuleEditor



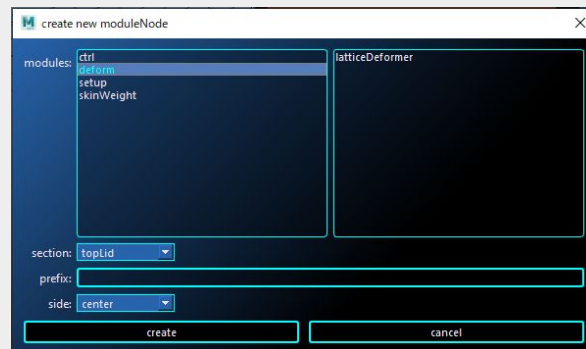
## + section

セクションノードを新規作成します。  
section名・sideを指定してください。



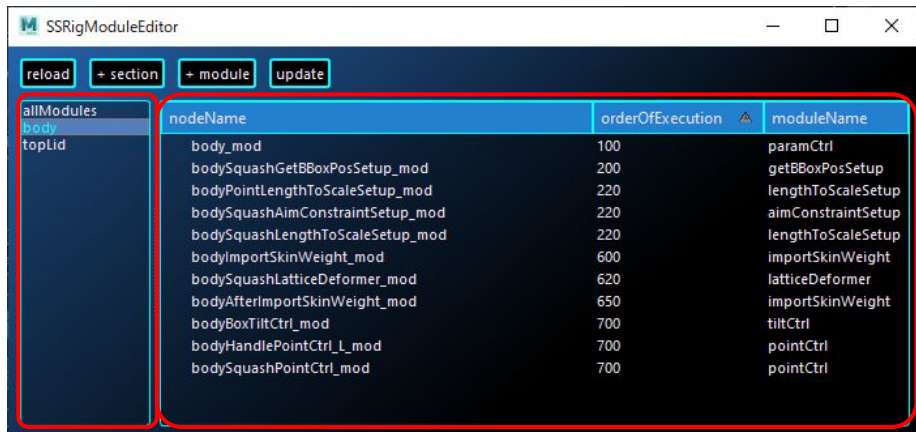
## + module

リグモジュールノードを新規作成します。  
リグモジュール・section・prefix・sideを指定してください。





# SSRigModuleEditor



## sectionList

**rigModule\_gp** 以下に格納されているsectionをリストします。  
allModules を選択すると、  
全section下に格納されている全リグモジュールノードを表示します。

## moduleList

sectionListの選択にあわせてリグモジュールノードをリストします。