

PSD Importer 連携仕様概要

2023.11 MxEサービスドキュメント

目次

- [文書の目的](#)
- [PSD Importer起動方法](#)
- [インポート基本処理](#)
- [各要素の取り込みルール](#)
 - [レイヤーグループの取り込み](#)
 - [レイヤー効果について](#)
 - [レイヤー名タグ（制御コード）](#)
 - [レイヤーカンプの取り込み](#)
- [GUI Object向け仕様](#)

文書の目的

- IoT用HMIデザインツール「MatrixEngine（略称「MxE」）SDK」では、デザインから動作まで、コンテンツ制作をスムーズに行うための機能を多数ご提供しています。 (*[次頁参照](#))
- 「PSD インポーター」は、デザイナーが制作したPhotoshopデザインデータを特定の画像ファイル形式への変換を挟むことなく、そのまま開発環境（MxE SDK）へと取り込む機能です。
- UIフレームワーク「GUIObject」の仕様に沿ったレイヤー構成でPSDファイルを制作すれば、UI部品の動作や機能を予め設定することもできます。 PSDファイルに設定した動作や機能は、インポート時に開発環境（MxE SDK）へ自動的に反映されます。
- MxEが推奨するPSDインポーターとGUI Object連携を活用したGUI制作では、デザインと機能のスマートな結びつけが可能です。
- 本書では、MxE SDKへの取り込みに適用されるPSDファイル内のデータ設定と、その連携仕様について概説します。
 - 上記目的以外のPhotoshop操作や用語については、Photoshopのヘルプを参照ください。

HMIデザインを容易にするMatrixEngineツール群

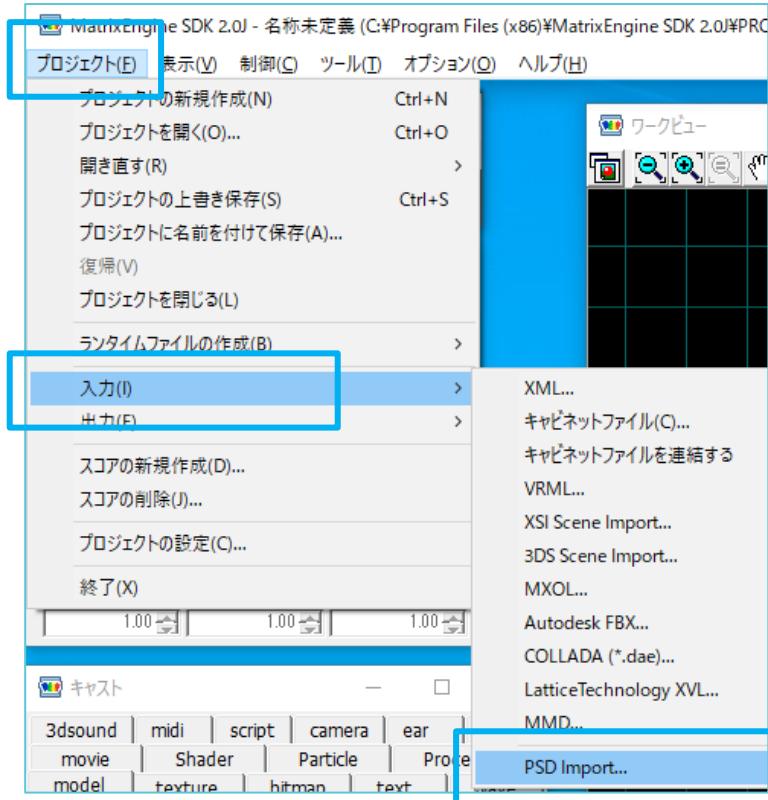
- MatrixEngine SDK
 - GUIコンテンツオーサリングツール本体です。
- PSD Importer
 - MatrixEngine SDK向けのプラグインです。
 - Photoshopで作成した素材（PSDファイル）をMatrixEngine SDKにインポートできます。
 - Photoshop向けプラグインと合わせてご利用いただくと、デザイン制作との協働効率がより向上します。
- Photoshop Plugin GUIObject Constructor
 - Photoshop向けのプラグインです。
 - MatrixEngine SDKに取り込むためのPSDファイル（レイヤー構造）を簡単・効率的に作成できます。
- Outline Builder
 - MatrixEngine SDK向けのプラグインです。
 - 一定パターンでコードを自動生成できる便利ツールです。
- GUIObject
 - MatrixEngineスクリプトで記述されたライブラリ群（コントロールモジュールとフレームワーク）です。

PSD Importer起動方法

プラグイン設定 & 使用方法アウトライン

PSDImporter起動方法

- MxE SDK用プラグイン PSDImporterの起動方法を説明します。



- メニューからプロジェクト > 入力 > PSD Import… を選択します。
- 取り込み用Photoshopデータ（psdファイル）を指定するファイル選択ダイアログが表示されます。

MxE SDKにPSDImporterが設定済みであることを前提とします。
設定については、別資料「インストールおよび環境構築ガイド」をご参照ください。

インポート基本処理

プラグイン設定 & 使用方法アウトライン

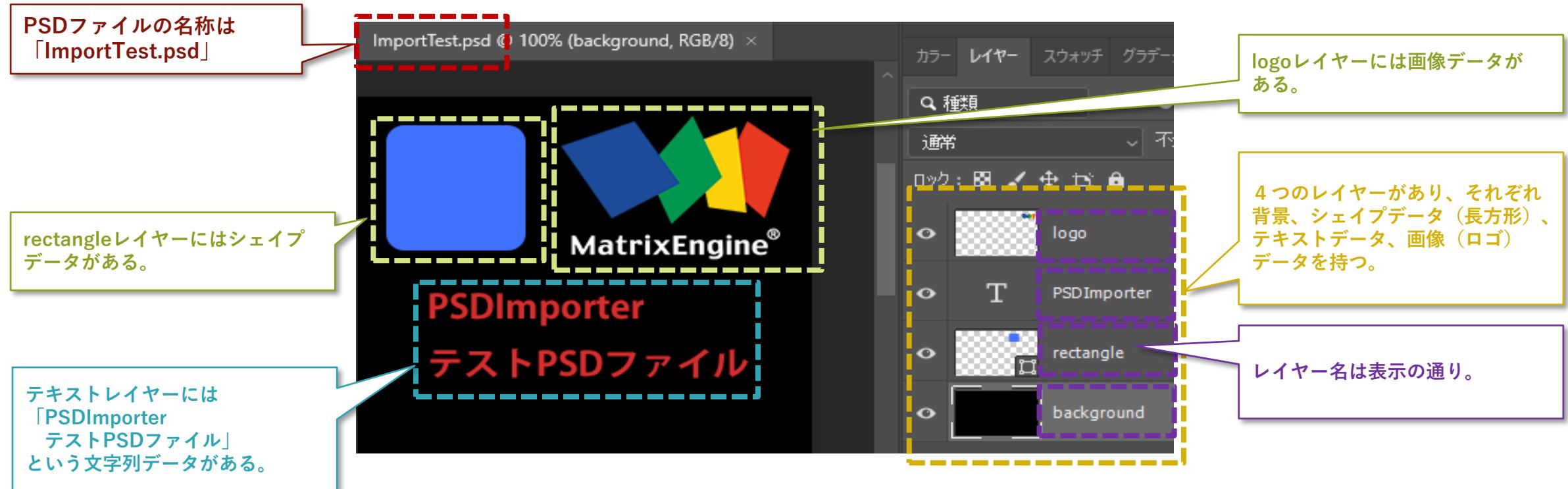
インポート基本処理：対応表

- PSDファイル内のデータ（レイヤー、画像、テキストなど）はMxE SDKに取り込まれると、基本的に以下の表のようになります。

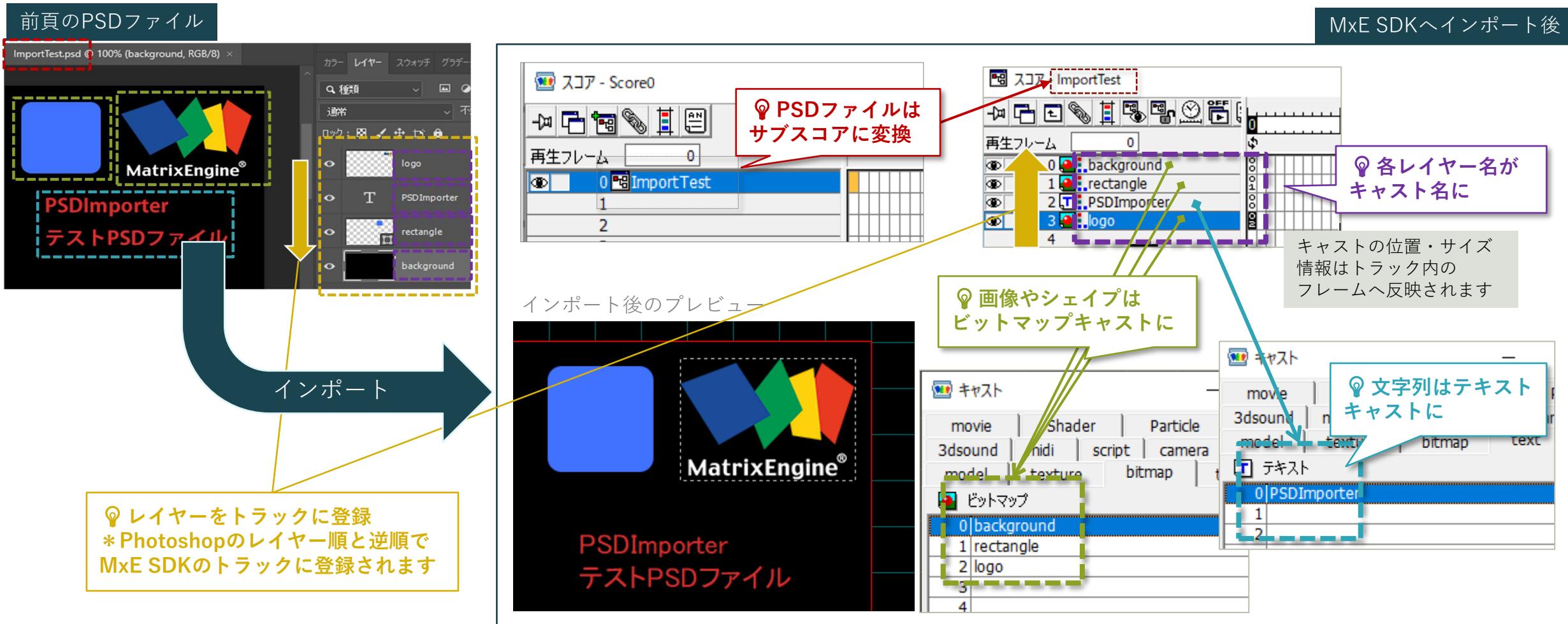
Photoshop		MatrixEngine SDK
PSDファイル全体	→	サブスコア
レイヤー	→	(サブスコア内) トランク
レイヤー名	→	(ビットマップ/テキスト)キャスト名
テキストレイヤー：文字列データ	→	テキストキャスト
シェイプレイヤー：シェイプデータ	→	ビットマップキャスト (ラスタライズ後)
画像レイヤー：画像データ	→	ビットマップキャスト

インポート基本処理（1）

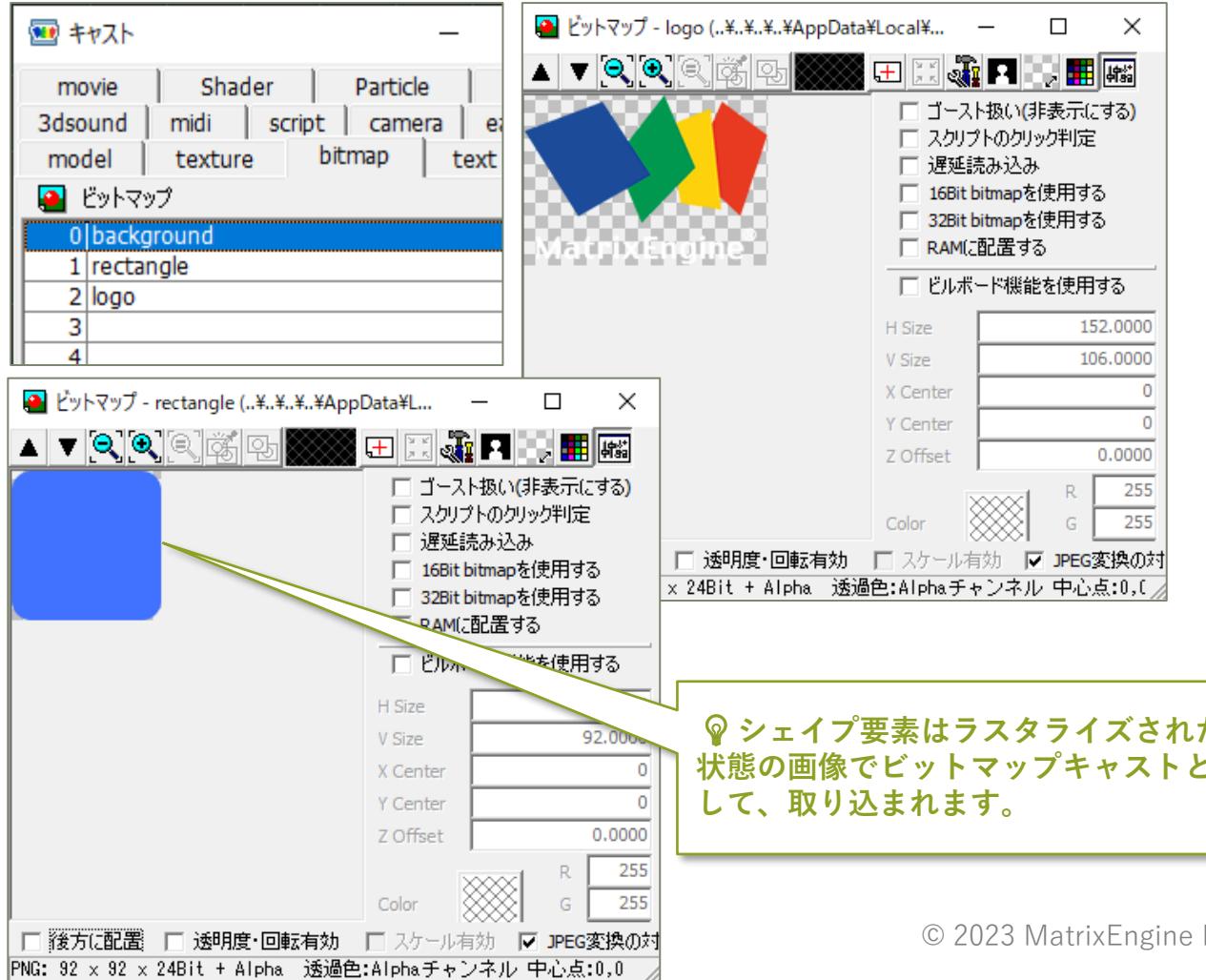
次のような構成の「ImportTest」というPSDファイルがある場合を例に、インポート処理前後の対応をご紹介します。



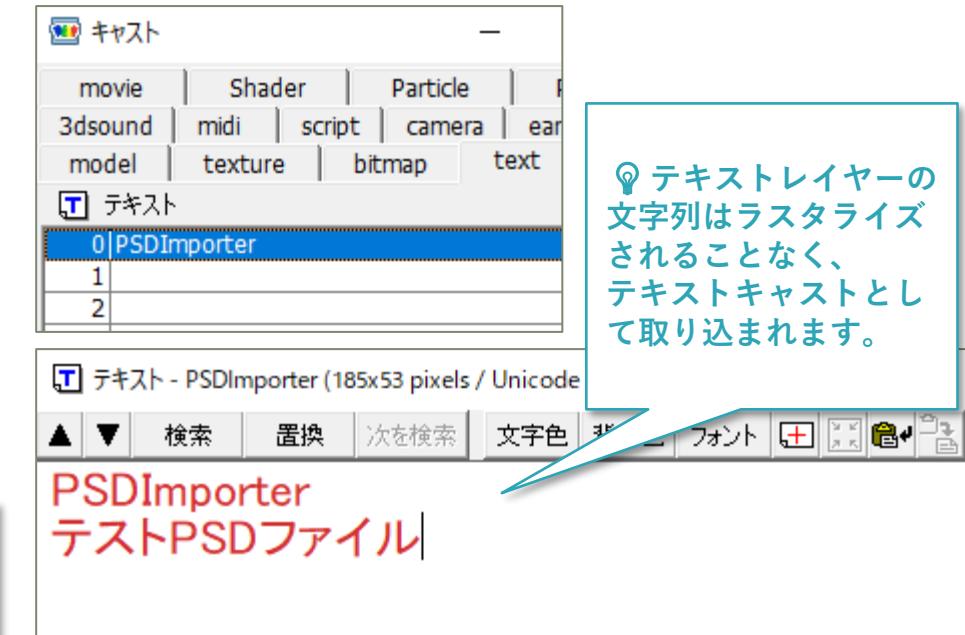
インポート基本処理（2）



インポート基本処理（3）



💡 シェイプ要素はラスタライズされた状態の画像でビットマップキャストとして、取り込まれます。



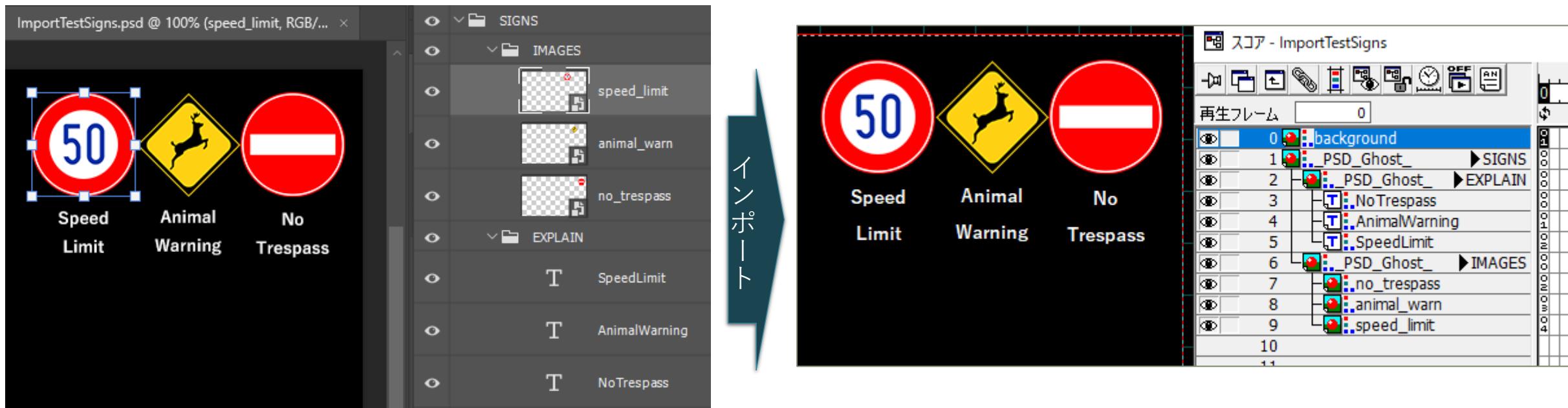
各要素の取り込みルール

プラグイン設定 & 使用方法アウトライン

レイヤーグループの取り込み

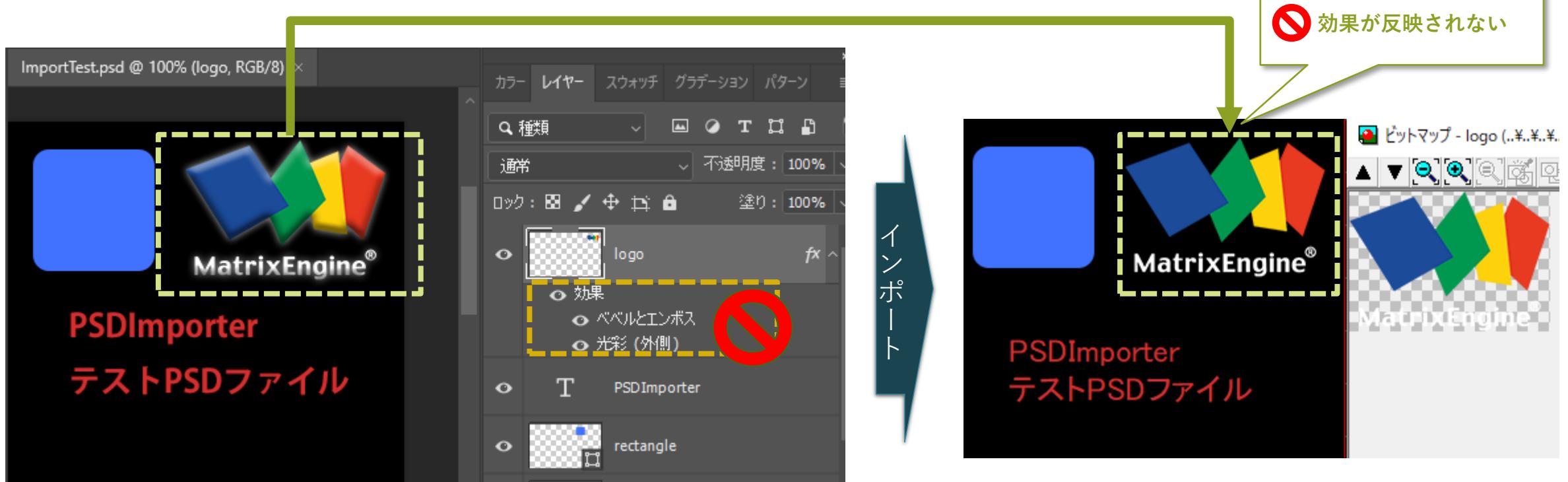
- PSDファイル内のレイヤーグループは、MxE SDKに取り込まれると”_PSD_Ghost_”という名前のゴースト属性(*)のビットマップトラックに置き換えられます。
- そのレイヤーグループに含まれる子レイヤーは、取り込み後、”_PSD_Ghost_”の子トラックとして配置されます。

*：表示画像の存在しない（見えない）キャスト



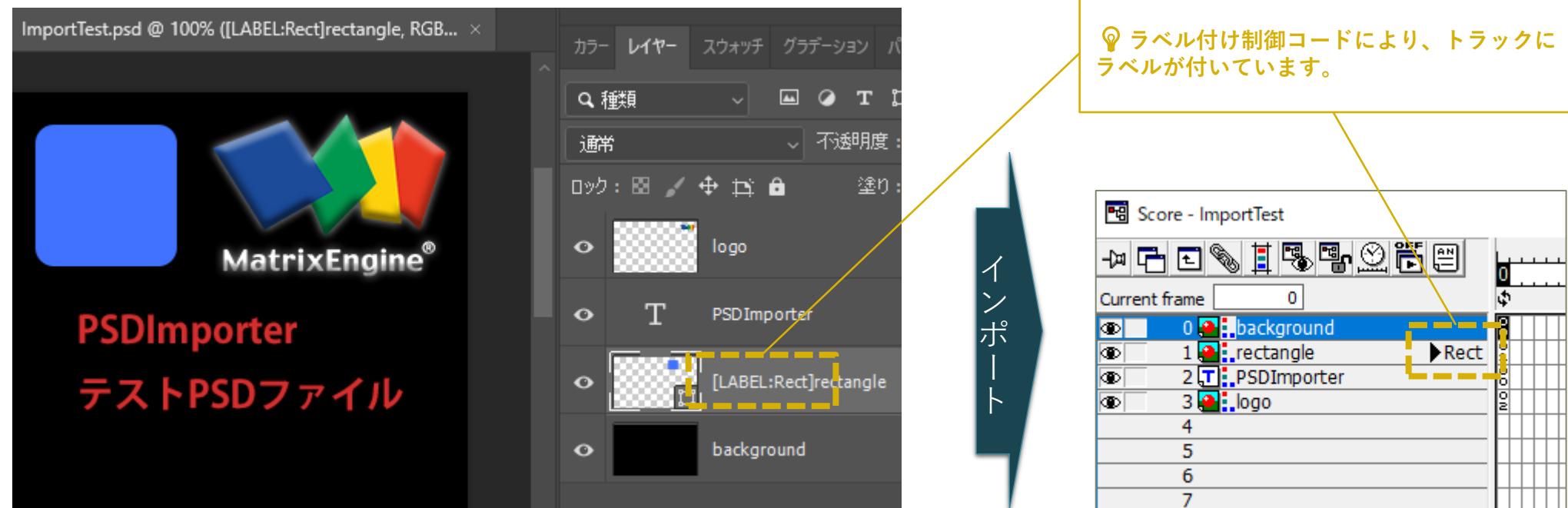
レイヤー効果について

- いずれのレイヤー効果も取り込まれないため、表示に反映させたい場合にはレイヤースタイルのラスタライズが必要です。



レイヤー名タグ（制御コード）

- レイヤー名の冒頭に[LABEL:…]のような制御コード（タグ）を記述することで、PSDファイルを取り込む際の様々な挙動の指定が可能です。



レイヤー名タグ（制御コード）の例

- 以下に、よく使用される制御コードと、その用途を列挙します。

Control Code	Summary
[NA]	レイヤーが取り込み対象から除外されます。
[NT]	レイヤーはトラックには登録されず、キャスト（のリスト）にのみ登録されます。
[ORIGIN]	レイヤーグループの原点として指定されます。
[LABEL:ラベル名]	取り込み後のトラックのラベル名を指定します。
[SCORE:スコア名]	レイヤーグループの内容が名前で指定したサブスコアへ取り込まれます。
[SCRIPT:スクリプトキャスト名]	同名のスクリプトキャストがあれば、トラックに配置され、先頭フレームにキーフレームが設定されます。
[CLICKABLE:ON/OFF]	ビットマップ／テキストキャストのクリック可否を指定します。
[SCALING:ON/OFF]	ビットマップ／テキストキャストの（キーフレーム毎の）スケール変更の可否を指定します。
[OPACITY:ON/OFF]	ビットマップ／テキストキャストの（キーフレーム毎の）透明度・回転角度変更の可否を指定します。
[ANSI] / [UTF]	テキストキャストの文字列のエンコーディング形式を指定します。

レイヤーカンプの取り込み

- レイヤーカンプは、Photoshopデータにおける各レイヤーの状態を記録した複数のスナップショットのことです。
- MxE SDKではPSDImporterによる取り込みの際にレイヤーカンプ毎に、その位置情報をキーフレームのデータとして、またレイヤーカンプ名をキーフレームのラベルとして、割り付けます。

レイヤーカンプの取り込み

PSDImporter テストPSDファイル

PSDImporter テストPSDファイル

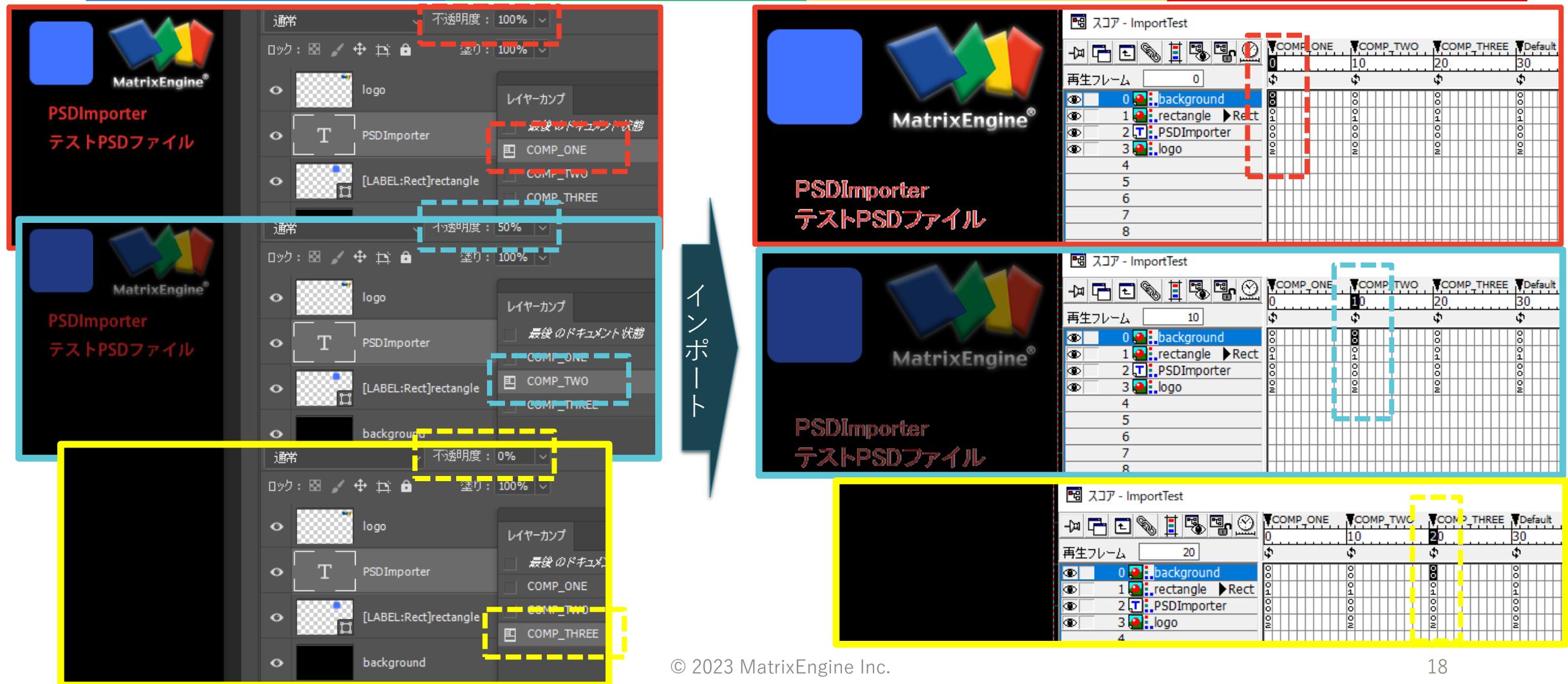
PSDImporter テストPSDファイル

インポート

PSDImporter テストPSDファイル

PSDImporter テストPSDファイル

PSDImporter テストPSDファイル



フレーム	COMP_ONE	COMP_TWO	COMP_THREE	Default
0	ON	ON	ON	OFF
10	ON	ON	ON	OFF
20	ON	ON	OFF	OFF

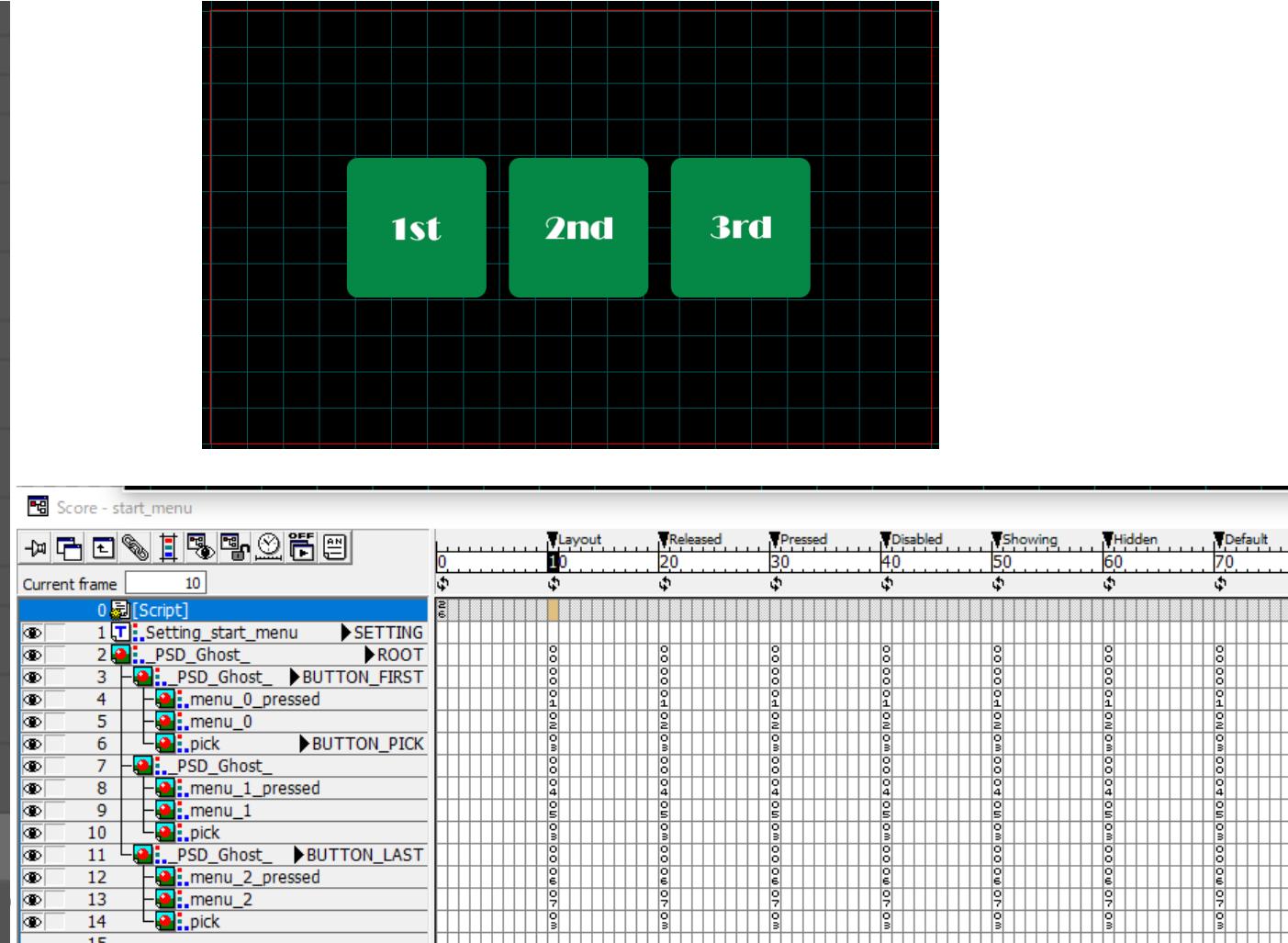
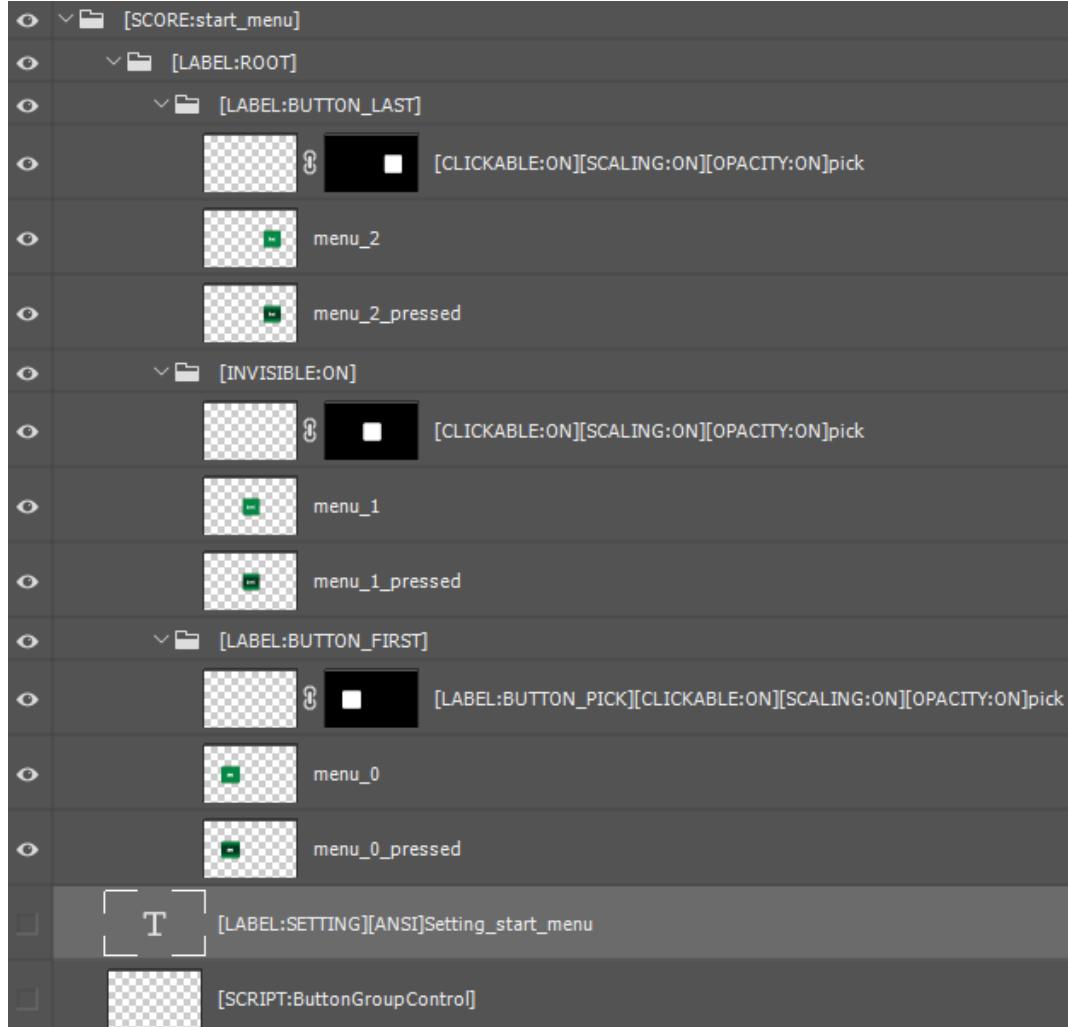
GUI Object向け仕様

プラグイン設定 & 使用方法アウトライン

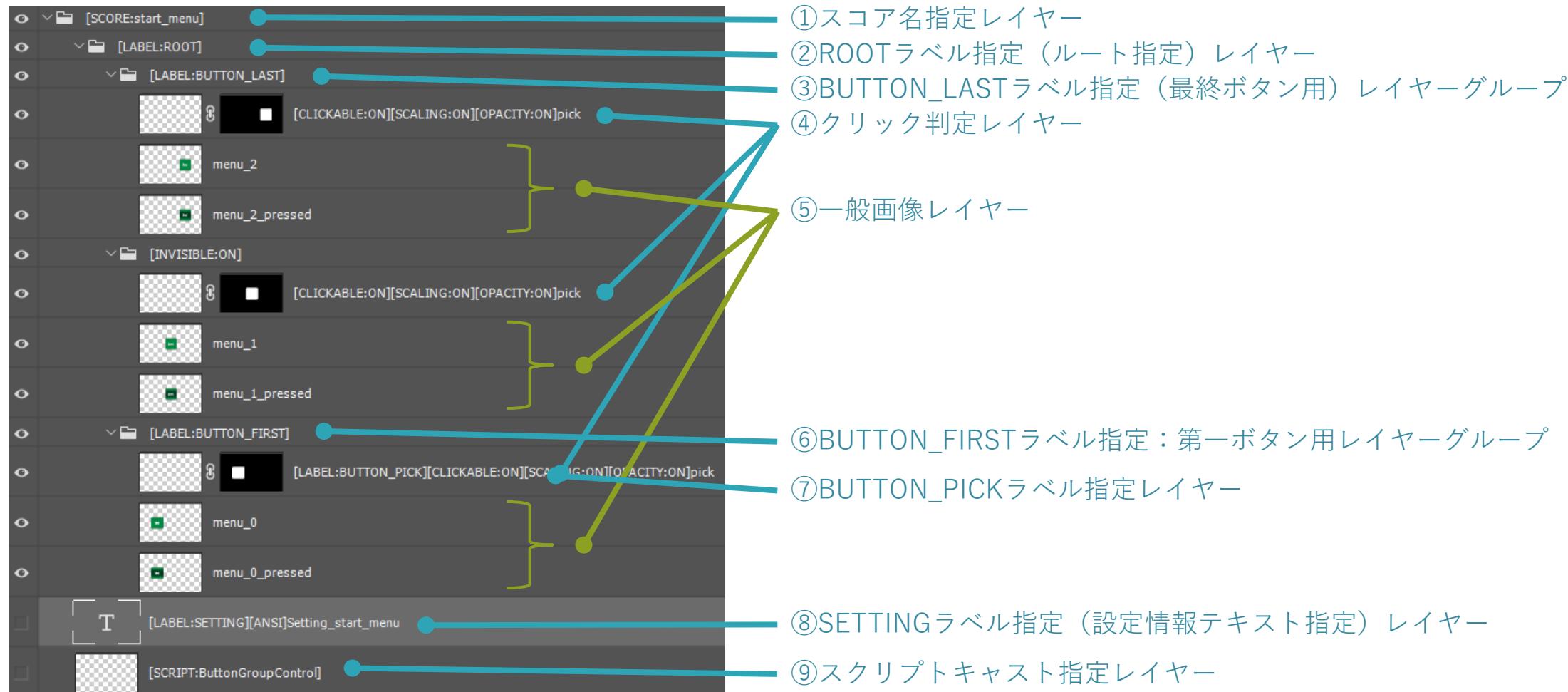
GUIObject向け仕様

- 以降は、GUIObjectを利用する際に使用される頻度の高い、取り込み仕様について説明します。
- 提供するアーカイブの中にあるPhotoshopプラグイン「**GUIObject Constructor**」を利用すると取り込み仕様に沿ったレイヤー構成を生成できるので、ご活用ください。
- GUIObjectとの連携におけるPSDImporterでの取り込み方の詳しい仕様については、GUIObjectアーカイブのdocumentフォルダ下にある「PSDインポーター連携ガイド.docx」を参照してください。
- 以下はButtonGroupコントロールにおけるPSDファイルのレイヤー構成とMxE SDKへの取り込み後のサブスコア・トラック構成の一例です。

レイヤー構成



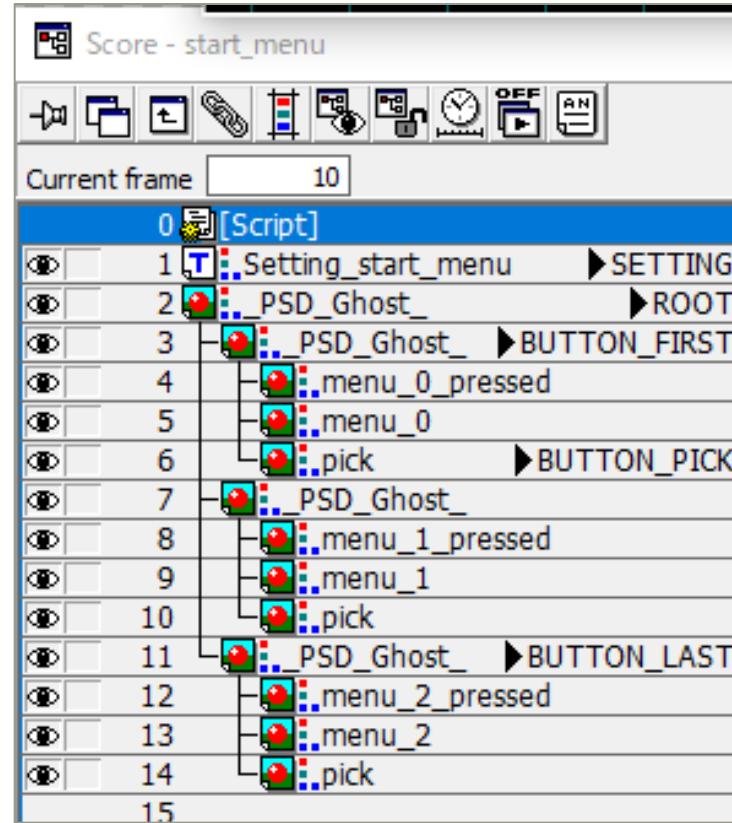
レイヤー構成



レイヤー構成

Photoshop	MatrixEngine SDK (GUIObject) サブスコアトラック変換後処理
①スコア名指定レイヤー	(サブ) スコアの名前を指定します。
②ROOTラベル指定レイヤー	GUIObjectではROOTラベルの付いたトラックを親として、その配下に表示要素を配置する前提となっているので、その仕様に合わせてROOTラベル付きトラックへの変換を指定します。
③BUTTON_LASTラベル指定レイヤーグループ	ButtonGroupコントロールにおいて、BUTTON_FIRSTラベルとBUTTON_PICKラベル付きトラックとのオフセットから、ボタン数を算出するための指標となります。
④クリック判定レイヤー	各ボタンの（レイヤーマスクによって指定される透明画像の位置とサイズで）クリック・タップ判定領域を指定します。
⑤一般画像レイヤー	1つのボタンが持つ（各状態の）画像を配置します。
⑥BUTTON_FIRSTラベル指定レイヤーグループ	ButtonGroupコントロールにおいて、BUTTON_FIRSTラベルとBUTTON_PICKラベル付きトラックのオフセットから、各ボタンが画像用に保持するトラック数の算出指標となります。
⑦BUTTON_PICKラベル指定レイヤー	ButtonGroupコントロールにおいて、BUTTON_FIRSTラベルとBUTTON_PICKラベル付きトラックのオフセットから、各ボタンが画像用に保持するトラック数の算出指標となります。
⑧SETTINGラベル指定レイヤー	SETTINGラベルの付いたテキストキャストは、コントロールのコンフィグ設定として動作します。
⑨スクリプトキャスト指定レイヤー	使用するコントロールに合わせて、GUIObject各コントロールの処理を司るスクリプトキャストを指定します。（図例ではGUIObjectフレームワークに含まれるButtonGroupコントロール用スクリプトを指定しています）

レイヤー構成(オフセットによる算出の例)



💡 (レイヤーグループ内) レイヤー順はトラックに変換される際、逆順になります。【BUTTON_PICK付きトラック番号 – BUTTON_FIRST付きトラック番号 + 1】により、各ボタンのトラック数が算出 (Ghost、画像 2 つ、pick) できます。

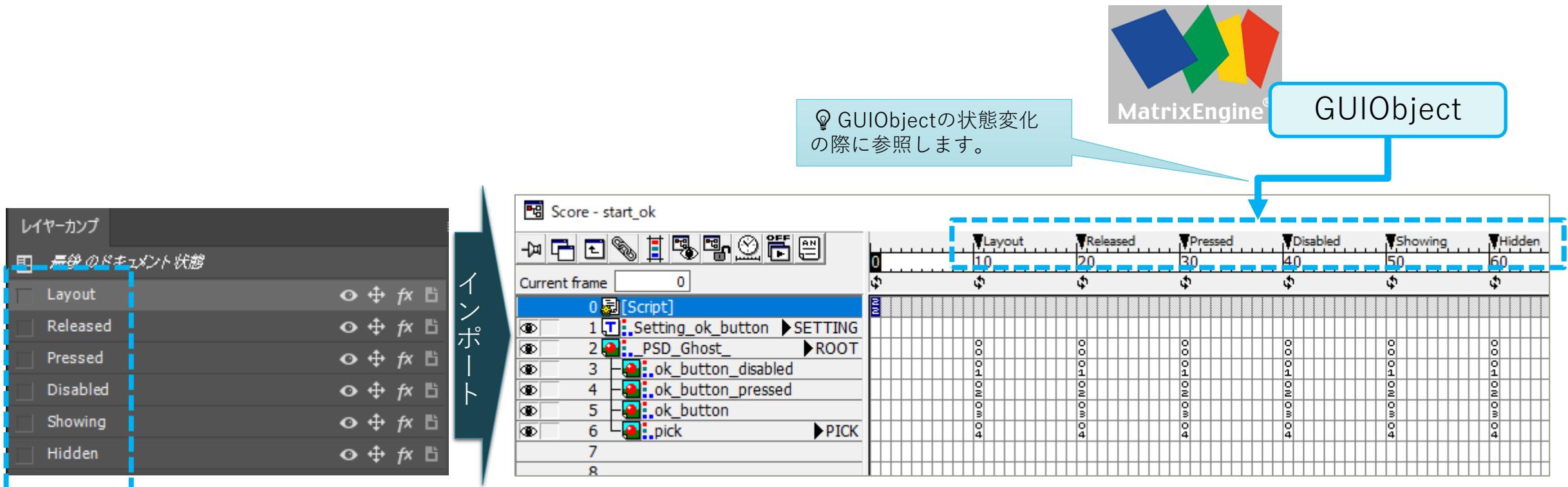
💡 BUTTON_LAST付きトラック番号(L)、BUTTON_FIRST付きトラック番号(F)と各ボタンのトラック数(T)により、 $[((L - F) \div T) + 1]$ で、この ButtonGroup コントロールのボタンの数が算出できます。

レイヤー構成

- 各コントロールでのトラックラベルの種類、機能についてはドキュメントの[コントロール名]コントロールマニュアル.docxを参照してください。

レイヤーカンプ構成

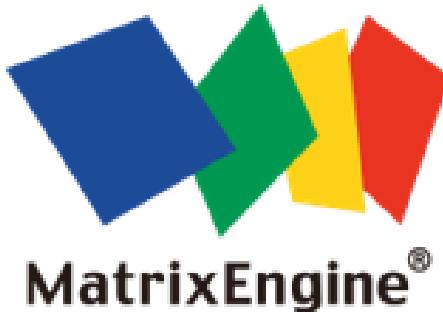
- GUIObjectには使用を想定したキーフレームラベルがあり、従ってGUIObject用にデフォルトとなるレイヤーカンプの構成が存在します。（下図参照）



お問い合わせ

- 担当営業、もしくは以下よりお問い合わせください。
 - <https://www.matrixengine.jp/contact>
 - info@matrixengine.jp

『IoTデザインの未来を創る』



<https://www.matrixengine.jp>