



Elysium  
**3DxSUITE™**

エリジオン 3DxSUITE  
*Data Package Studio* マニュアル

Version EX9.1 2022

株式会社エリジオン

# 目次

|  |    |
|--|----|
| 1. 概要  | 1  |
| 2. 変更履歴  | 1  |
| 3. Data Package Studio とは                        | 7  |
| 4. インストール  | 8  |
| 4.1. 推奨動作環境                                      | 8  |
| 4.2. 3DxSUITE のインストールおよびバージョンアップ                 | 8  |
| 4.3. Data Package Studio のインストールおよびバージョンアップ      | 8  |
| 4.4. Data Package Studio の起動                     | 9  |
| 5. TDP の作成                                       | 9  |
| 6. 操作ガイド   | 11 |
| 6.1. 基本機能  | 11 |
| 6.1.1. データソースの追加                                 | 12 |
| 6.1.2. 3D View Widgetの追加およびカスタマイズ                | 14 |
| 6.1.3. View Carousel Widgetの追加およびカスタマイズ          | 15 |
| 6.1.4. Button Widget の追加                         | 16 |
| 6.1.5. Text Field Widgetの追加                      | 17 |
| 6.1.6. PMI Table Widgetの追加                       | 18 |
| 6.1.7. BoM Table Widgetの追加                       | 19 |
| 6.1.8. Custom Table Widgetの追加                    | 20 |
| 6.1.9. User Property Table Widgetの追加             | 21 |
| 6.1.10. Table Column Filter Widgetの追加            | 23 |
| 6.1.11. Widgetをカスタマイズ                            | 24 |
| 6.1.12. Trigger と Action の追加                     | 25 |
| 6.2. 使用事例  | 26 |
| 6.2.1. TDP のテンプレートの作成                            | 26 |
| 6.2.2. CAD Validator レポートのカスタマイズ                 | 29 |
| 6.2.3. Table および View Carousel のコンテンツのフィルタリング    | 34 |
| 6.2.4. Bill of Characteristics Tableの追加          | 37 |
| 6.2.5. エリジオン TDP JavaScript API を用いたテーブルデータの保存   | 42 |
| 6.2.6. 作業指示 TDP の作成                              | 45 |
| 6.2.7. ビューの印刷の有効化                                | 53 |
| 6.2.8. エリジオン TDP JavaScript API を用いた 3D ツールバーの追加 | 54 |

|  |    |
|--|----|
| 6.3. TDP の作成   | 59 |
| 6.3.1. 3DxSUITE SmartLauncher を用いた TDP の作成                   | 59 |
| 6.3.2. 3DxSUITE SmartController を用いた TDP の作成                 | 60 |
| 6.3.3. 3DxSUITE TransServer を用いた TDP の作成                     | 61 |
| 7. 予約済みの PDF フィールド名  | 62 |
| 8. FAQ   | 63 |
| 8.1. 用語  | 63 |
| 8.1.1. DPS   | 63 |
| 8.1.2. Formula   | 63 |
| 8.1.3. Data Source (データソース)                                  | 63 |
| 8.1.4. Widget  | 64 |
| 8.1.5. Trigger   | 65 |
| 8.1.6. Action  | 65 |
| 8.1.7. Trigger と Action の組合せ                                 | 66 |
| 8.1.8. Data Package Studio が 3種類のファイルを保存する理由                 | 69 |
| 8.1.9. 事前定義プロパティとは   | 69 |
| 8.1.10. ユーザ定義プロパティとは   | 69 |
| 8.1.11. 事前定義プロパティとユーザ定義プロパティの違いとは                            | 69 |
| 8.2. 機能の拡張   | 69 |
| 8.2.1. Data Package Studio の機能拡張                             | 69 |
| 8.2.2. Data Package Studio の API ドキュメント                      | 70 |
| 8.3. フォーマット変更とカスタマイズ   | 70 |
| 8.3.1. 添付ファイルの TDP への一括追加                                    | 70 |
| 8.3.2. 表の背景色の変更  | 70 |
| 8.3.3. CAD Validator レポートに含まれる elements/widgets<br>のフォーマット変更 | 70 |
| 8.3.4. TDP に含まれる表の列/行のサイズ変更                                  | 70 |
| 8.3.5. CAD 比較レポートに含まれる表の列の削除                                 | 70 |
| 8.3.6. レポートに含まれる表のスクロール                                      | 71 |
| 8.3.7. CAD 比較レポートの全体フィットと選択解除について                            | 71 |
| 8.3.8. 3D ビューの背景色が設定した色と異なる点について                             | 71 |
| 8.3.9. View Carousel のサムネイル画像が表示されない現象について                   | 71 |
| 8.3.10. 添付ファイル名の自動読み込みについて                                   | 71 |
| 8.4. インストールとライセンス  | 71 |

|   |    |
|---|----|
| 8.4.1. Data Package Studio 使用時のライセンスについて                | 71 |
| 8.4.2. Adobe Acrobat の plug_ins フォルダの場所                 | 71 |
| 8.5. PDF  | 72 |
| 8.5.1. ファイルの添付  | 72 |
| 8.5.2. 日付フィールドの追加                                       | 72 |
| 8.5.3. 不足しているフォントを PDF に追加                              | 72 |
| 8.5.4. ページの追加およびページの整理                                  | 72 |
| 8.5.5. タブ及びボタンの追加                                       | 73 |
| 8.5.6. 透かしの追加   | 73 |
| 8.5.7. 編集可能な PDF フィールドの作成及び変更                           | 73 |
| 8.5.8. テンプレートの作成  | 73 |
| 8.5.9. PDF 内テキストの編集                                     | 73 |
| 8.5.10. デジタル署名の追加                                       | 73 |
| 8.5.11. 3D コンテンツの有効化                                    | 74 |
| 8.5.12. パスワードによる保護                                      | 74 |
| 8.5.13. 添付ファイルの確認                                       | 74 |
| 8.6. ユーザインタフェース   | 74 |
| 8.6.1. ユーザインタフェースに関するその他の情報について                         | 74 |
| 9. テンプレートおよびデータソースの例                                    | 74 |
| 9.1. サンプルファイルの設定  | 75 |
| 9.1.1. サンプルファイルをコピー                                     | 75 |
| 9.1.2. サンプルシナリオをコピー                                     | 75 |
| 9.1.3. サンプルシナリオを実行するために 3DxSUITE SmartLauncher<br>を構成する | 75 |
| 9.2. サンプルファイル   | 76 |
| 9.2.1. データソース   | 76 |
| 9.2.2. Formulas   | 77 |
| 9.2.3. 出力   | 78 |
| 9.2.4. シナリオ   | 78 |
| 9.2.5. スクリプト  | 79 |
| 9.2.6. テンプレート   | 80 |
| 付録 A: 3DxSUITE パラメーター                                   | 81 |
| A.1. 3D PDF Component で必要となるパラメーター                      | 81 |
| A.2. 3D PDF Component のオプションパラメーター                      | 81 |



|   |     |
|---|-----|
| A.3. CAD Validator Component で必要となるパラメーター | 82  |
| A.4. CAD Validator Component のオプションパラメーター | 83  |
| A.5. 3DxSUITE のオプションパラメーター (CAD変換関連)      | 84  |
| 付録 B: 事前定義モデルプロパティ                        | 85  |
| 付録 C: 事前定義 PMI プロパティ                      | 91  |
| C.1. 共通プロパティ                              | 91  |
| C.2. Datum プロパティ                          | 92  |
| C.3. Datum Target プロパティ                   | 92  |
| C.4. Dimension プロパティ                      | 93  |
| C.5. Geometric Tolerance プロパティ            | 95  |
| C.6. Line Weld プロパティ                      | 96  |
| C.7. Locator プロパティ                        | 98  |
| C.8. Roughness プロパティ                      | 98  |
| C.9. Spot Weld プロパティ                      | 99  |
| C.10. Bill of Characteristic (BoC) プロパティ  | 100 |
| 付録 D: Widget の最小サイズ                       | 101 |
| 付録 E: エリジオン TDP JavaScript API            | 106 |
| E.1. クラス: Ely3dpdf                        | 106 |
| E.1.1. 関数: getApi                         | 106 |
| E.2. クラス: ElyTdpApi                       | 107 |
| E.2.1. 関数: getField                       | 107 |
| E.2.2. 関数: getWidget                      | 107 |
| E.3. クラス: ElyTdpWidget                    | 108 |
| E.3.1. 関数: getType                        | 108 |
| E.4. クラス: ElyTdp3dViewWidget              | 108 |
| E.4.1. 関数: reset                          | 109 |
| E.4.2. 関数: fitAll                         | 109 |
| E.4.3. 関数: fitSelection                   | 109 |
| E.4.4. 関数: clearSelection                 | 110 |
| E.4.5. 関数: getRenderModes                 | 110 |
| E.4.6. 関数: setRenderMode                  | 111 |
| E.4.7. 関数: resetRenderMode                | 111 |
| E.4.8. 関数: isValidRenderMode              | 112 |
| E.4.9. 関数: getViews                       | 112 |

|  |     |
|--|-----|
| E.4.10. 関数: setView .....                                | 113 |
| E.4.11. 関数: resetView .....                              | 113 |
| E.4.12. 関数: isValidView .....                            | 114 |
| E.4.13. 関数: getShowPmi .....                             | 114 |
| E.4.14. 関数: setShowPmi .....                             | 115 |
| E.5. クラス: ElyTdpTableWidget .....                        | 115 |
| E.5.1. 関数: clearFiltering .....                          | 115 |
| E.5.2. 関数: clearSelection .....                          | 115 |
| E.5.3. 関数: getData .....                                 | 116 |
| E.5.4. 関数: getHeader .....                               | 117 |
| E.5.5. 関数: reset .....                                   | 117 |
| E.5.6. 関数: resetSorting .....                            | 118 |
| E.6. Class: ElyTdpTextWidget .....                       | 118 |
| E.6.1. Function: getReadOnly .....                       | 118 |
| E.6.2. Function: setReadOnly .....                       | 119 |
| E.6.3. Function: toggleReadOnly .....                    | 119 |
| 付録 F:  |     |
| CAD Validator レポートテンプレートの新しいフォーマットへのマイグレーション ..          | 120 |
| F.1. CAD Validator Widget タイプ .....                      | 121 |
| F.1.1. 3D View (CAD Validator) .....                     | 121 |
| F.1.2. Assembly Tree (CAD Validator) .....               | 122 |
| F.1.3. CAD Information (CAD Validator) .....             | 122 |
| F.1.4. Color Bar (CAD Validator) .....                   | 122 |
| F.1.5. Component Summary (CAD Validator) .....           | 123 |
| F.1.6. Detail List (CAD Validator) .....                 | 123 |
| F.1.7. Difference Summary (CAD Validator) .....          | 125 |
| F.1.8. Execution Property (CAD Validator) .....          | 125 |
| F.1.9. Property Table (CAD Validator) .....              | 126 |
| F.1.10. Type Summary (CAD Validator) .....               | 126 |
| F.1.11. Validation Setting (CAD Validator) .....         | 127 |
| F.2. CAD Validator 3D View Widget に関連するプレースホルダーの結合 ..... | 127 |
| F.3. CAD Validator Widget プレースホルダーの名称変更 .....            | 128 |
| F.4. Data Package Studio で Formula を作成する .....           | 129 |
| 付録 G: TDP 前処理スクリプトの例 .....                               | 130 |

|                   |     |
|-------------------|-----|
| 付録 H: 既知の問題 ..... | 132 |
|-------------------|-----|

# 1. 概要

このドキュメントは、Technical Data Package<sup>1</sup> (TDP) をカスタマイズする場合や次の様な場合に役立ちます。

- はじめて Data Package Studio (DPS) を使用する場合
- DPS の多くの機能について詳しく知りたい場合

3DxSUITE の設定方法について知りたいシステム管理者の方は、3DxSUITEのドキュメントを参照してください。

(例: C:\Program Files\Elysium\3DxSUITE\doc)

1: Technical Data Package には、学習、エンジニアリング、ロジスティクス、生産などに適した関連製品情報が含まれています。



最新の更新については、オンラインマニュアル (<https://doc.elysium-global.com/>) にアクセスしてください。(ログイン資格情報については管理者にお問い合わせください。)

## 2. 変更履歴

### Data Package Studio EX9.1 での更新内容

EX9.1 は、さまざまな機能の拡張を行ったメジャーリリース版になります。更新内容は以下の通りです。

- Adobe Acrobat 64bit
  - Data Package Studio を Adobe Acrobat 64bit で実行できるようにしました。64bit 版 インストーラーを使用してください。
  - 生成された 3D PDF ファイルを Adobe Acrobat 64bit で開いた際に正しく動作するようにしました。



Adobe Acrobat 64bit 版では、PMI に関連する形状のハイライト表示と要素のカラー表示が想定通りに動作しない場合があります。詳細は [既知の問題](#) を参照してください。

- CAD Validator
  - 起動時に 3D ビューで PMI を表示/非表示にするオプションを追加しました。PMI が画面を埋め尽くしてしまう現象を抑制するためです。
  - Property Table から URL を開くことができるようにしました。(Diff Property で提供される場合)
    - CAD Validator の比較内容をカスタマイズする必要があります。
    - レポートのレビュー作業を単純化する目的で要素タイプに関連するドキュメントを提供したい場合などに便利です。

- 1つのビューを含むグループを選択した場合、ビューの方向に揃えるようにしました。
- グループをソートできるようにしました。
- グループ化されていない (Ungrouped) グループを選択した場合には、画面のフィットと差分のハイライトを行わないようにしました。
- 作業指示 (以下に記載した Table に関する機能拡張により実現しました)
  - Custom Table に作業指示の手順を表示できるようにしました。
  - 作業指示の手順に関する情報を以下の部分に表示できるようにしました。
    - Text Field (例: 手順の詳細)
    - Custom Table (例: 手順に関連するツール、パーツ、安全上の注意事項など)
  - 作業指示の手順に関連付けられたビューを表示できるようにしました。

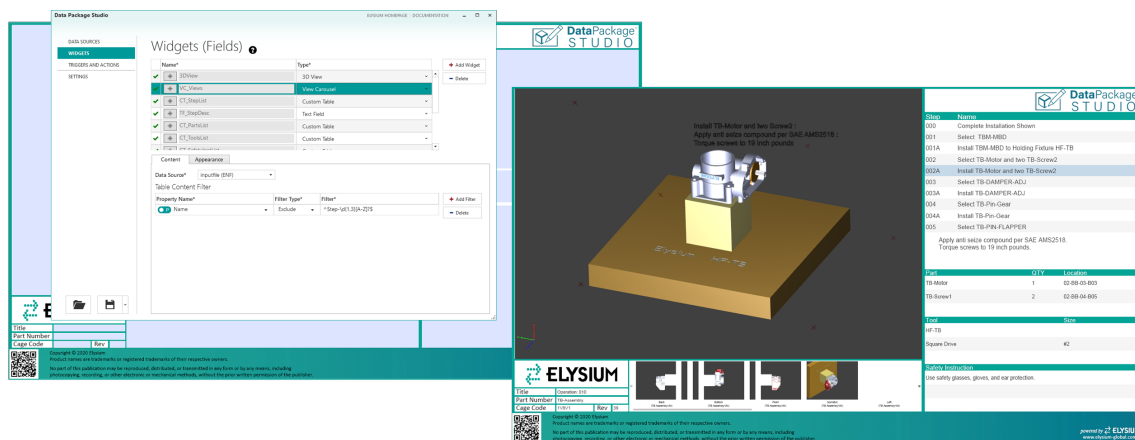


図 1. 例: 作業指示 TDP (右) と手順に関するビューのフィルタリング方法 (左) - "Step-" + 1~3 桁の数字 + 大文字 (オプション) に一致するすべての手順

- Table
  - フィルタリング機能を強化しました。
    - すべてのテーブルのフィルターで正規表現を使用できるようにしました。
    - コンテンツのフィルタリングを可能にしました。除外/包含フィルターを指定して、テーブルのコンテンツを制御することができます。
    - フィルタリングで "ドリルダウン" ができるようにしました。
      - 表の列にフィルターを適用します。ヘッダーをクリックしてフィルターオプションを選択します。
      - 複数の列にフィルターを適用すると、すべてのフィルターに一致するコンテンツのみが表示 (ドリルダウン) されます。
    - 別の Custom Table で選択された項目をもとに Custom Table をフィルタリングできるようにしました。
      - 2 番目のテーブル (トリガーテーブル) のプロパティで指定されたフィルターを適用します。
      - 下に記載した "空の Custom Table" の機能拡張と組み合わせることで、1:n の関係に

ある情報を表示する際に役立ちます。

- 複数の列でフィルタリングができるようにしました。指定された 1 つ以上の列内で フィルターに一致する項目が表示されます。
- 並べ替え機能を強化しました。ヘッダーをクリックして並べ替え順序 "A to Z" または "Z to A" を選択します。
- Custom Table で選択した項目をもとに 3D ビューを選択できるようになりました。
  - Custom Table のプロパティで指定されたビュー名を 3D ビューに適用します。
  - Custom Table の項目と特定のビューを関連付けたい場合に便利です。
- 選択された Custom Table 項目のプロパティを Text Field に表示できるようにしました。
- Custom Table を空の状態に表示し、適用されたフィルターに一致する項目のみを表示できるようにしました。
  - 指定した場合、起動時およびフィルタリングされていない状態のテーブルは空になります。
  - 上に記載した "Custom Table のフィルタリング" 機能拡張と組み合わせて使用すると便利です。
- BoM Table に物理的プロパティ (質量や体積など) を表示できるようにしました。
- Text データソースについて文字エンコーディングのサポートを拡張しました。(CSV データソースと同様のエンコーディングをサポート)
- JavaScript API
  - 編集可能なテキストフィールドのロック/ロック解除を可能にしました。

ドキュメントについては以下の内容を更新しています。

表 1. Data Package Studio EX9.1 で追加した内容

| 新たに追加した章                                      | 概要  |
|---|---|
| <b>Table および View Carousel のコンテンツのフィルタリング</b> | Table および View Carousel widget のコンテンツをフィルタリングする方法について説明を追加しました。 |
| <b>作業指示 TDP の作成</b>                           | 作業指示 TDP の作成方法に関する説明を追加しました。                                    |
| <b>Widget の最小サイズ</b>                          | widget の最小の幅と高さに関する説明を追加しました。                                   |

表 2. Data Package Studio EX9.1 で更新した内容

| 更新した章   | 概要  |
|---|---|
| <a href="#">CAD Validator レポートのカスタマイズ</a>                 | 新しいオプション "Show PMI (at startup)" に関する説明を追記しました。また、CAD Validator コンポーネントで TDP を作成するためのパラメーターを指定する際、有用となる情報を追記しました。       |
| <a href="#">事前定義モデルプロパティ</a>                              | 3DEXPERIENCE / V6 のプロパティが DPS のどのプロパティにマッピングされるかを追記しました。  |
| <a href="#">Class: ElyTdpTextWidget</a>                   | JavaScript API を用いて Text Field に編集ロックを設定する方法および編集ロックを解除する方法を追記しました。   |
| <a href="#">インストール</a>                                    | Data Package Studio の更新内容を反映しました。<br>バージョンに関連する情報の更新、テーブルコンテンツフィルターに関する説明の追加、シナリオパラメーターセットの使用に関するサンプルファイルの更新などを実施しています。 |
| <a href="#">View Carousel Widgetの追加およびカスタマイズ</a>          |   |
| <a href="#">PMI Table Widgetの追加</a>                       |   |
| <a href="#">BoM Table Widgetの追加</a>                       |   |
| <a href="#">Custom Table Widgetの追加</a>                    |   |
| <a href="#">User Property Table Widgetの追加</a>             |   |
| <a href="#">Table Column Filter Widgetの追加</a>             |   |
| <a href="#">Bill of Characteristics Tableの追加</a>          |   |
| <a href="#">エリジオン TDP JavaScript API を用いたテーブルデータの保存</a>   |   |
| <a href="#">ビューの印刷の有効化</a>                                |   |
| <a href="#">エリジオン TDP JavaScript API を用いた 3D ツールバーの追加</a> |   |
| <a href="#">3DxSUITE SmartLauncher を用いた TDP の作成</a>       |   |
| <a href="#">3DxSUITE SmartController を用いた TDP の作成</a>     |   |
| <a href="#">テンプレートおよびデータソースの例</a>                         |   |
| <a href="#">3DxSUITE パラメーター</a>                           |   |
| <a href="#">既知の問題</a>                                     |   |

# Data Package Studio EX9.0.5 での更新内容



本リリースでは Data Package Studio のバックエンドで動作する 3DxSUITE 3D PDF adapter のみをアップデートしており、Data Package Studio 自体は更新されません。このため、本リリースに Data Package Studio のインストーラーは含まれていません。



3DxSUITE をインストール/更新する方法および過去のバージョンから Data Package Studio EX9.0 へ更新する方法についての詳細は、[インストール](#) の章を参照してください。

EX9.0.5 では、以下のような機能強化を行いました。

- CAD Validator 入力ファイルとして設定されたモデルを CAD Validator 以外の widget でデータソースとして容易に利用できるようにしました。
- アセンブリ構造におけるインスタンスの表示状態の処理を改善しました (例: ビュー内での、より正確な表示状態のため)。

以下の問題も修正しました。

- ビューを適用した際に、要素が部分的もしくは全体的に消えてしまう問題。
- 分解ビュー内で PMI 項目が予期しない表示状態となってしまう問題。

ドキュメントについては以下の内容を更新しています。

表 3. Data Package Studio EX9.0.5 で追加した内容

| 新たに追加した章                       | 概要  |
|--------------------------------|---|
| <a href="#">ENF データソースの命名則</a> | ENF データソースの一般的な命名規則および CAD Validator 入力ファイルを利用するための具体的な方法について説明を追加しました。 |

表 4. Data Package Studio EX9.0.5 で更新した内容

| 更新した章                                     | 概要   |
|---|--|
| <a href="#">CAD Validator レポートのカスタマイズ</a> | Data Package Studio で、データソースとして CAD Validator 入力ファイルを利用するための方法を追記しました。 |
| <a href="#">インストール</a>                    | Data Package Studio の最新の変更を反映しました。                                     |
| <a href="#">既知の問題</a>                     |  |



# Data Package Studio EX9.0 での更新内容

Data Package Studio EX9.0 では、以下のような機能強化を行いました。

- CAD Validator widget カスタマイズ (表示/非表示):
  - 下図の例または [CAD Validator レポートのカスタマイズ](#) を参照してください。
- 3D View widget:
  - 「分解」ビューをサポートしました。
  - Default view を指定するオプションを追加しました。
  - Cross sections の表示状態 (表示/非表示) を指定するオプションを追加しました。
  - Cross sections のキャッピングを制御するパラメーター (SectionCapping) を追加しました。詳細は [3DxSUITE パラメーター](#) を参照してください。
  - 画面に平行な PMI の変換を制御するパラメーター (XConvertFlatToScreenPMI) を追加しました。詳細は [3DxSUITE パラメーター](#) を参照してください。
- Text Field widget:
  - Text Field を編集するためのオプションを追加しました。
- 色が指定されていない面のボリューム色の代わりに、モデルのデフォルト色が使用される問題を修正しました。
- パラメーター 「Script」 の設定値 「BasicTDP」 を廃止しました。



「Script=BasicTDP」が設定されているシナリオは依然実行可能です。ただし、実際には「Script=Simple」が指定されているものとして処理は実行され、パラメーターの値「BasicTDP」がサポートされなくなったという警告をログに記載します。



「BasicTDP」スクリプトの代わりに自作のテンプレートやフォーミュラを使用するようにシナリオを変更することをおすすめします。

ドキュメントについては以下の内容を更新しています。

表 5. Data Package Studio EX9.0 で追加した内容

| 新たに追加した章  | 概要   |
|---|--|
| <b>CAD Validator</b> レポートテンプレートの新しいフォーマットへのマイグレーション | CAD Validator レポートテンプレートを新しいフォーマットへマイグレーションする方法を追記しました (widget をカスタマイズする場合のみ必要です)。 |

表 6. Data Package Studio EX9.0 で更新した内容

| 更新した章                              | 概要   |
|------------------------------------|--|
| <b>CAD Validator</b> レポートのカスタマイズ   | CAD Validator widget をカスタマイズする Data Package Studio の機能に合わせて更新しました。 |
| <b>3D View Widget</b> の追加およびカスタマイズ | Cutting planes の default view および表示状態 (表示/非表示) を指定する方法を更新しました。     |
| <b>Text Field Widget</b> の追加       | Text Field widget の編集方法を更新しました。                                    |
| <b>3DxSUITE</b> パラメーター             | Data Package Studio の最新の変更を反映しました。                                 |
| インストール                             |  |
| 既知の問題                              |  |

### 3. Data Package Studio とは

Data Package Studio は、3DxSUITE が出力する TDP を、テンプレートと選択されたデータソースに基づいてカスタマイズするためのツールです。これはカスタマイズされた TDP を作成するために必要な 3つのステップのうちの一つです。各ステップについては以下の図を参照してください。

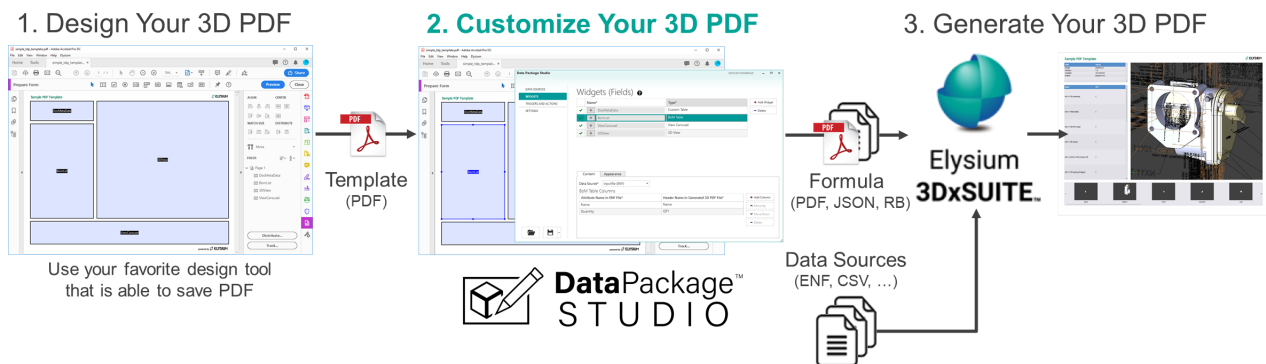


図 2. Data Package Studio を用いて 3DxSUITE が出力する TDP をカスタマイズする

Data Package Studio で以下の内容を指定することにより、TDP を作成する際に用いる formula をカスタマイズすることができます。

1. テンプレート内においてどのデータソースをどのプレースホルダー(テキストフィールド)に関連付けるか
2. 各プレースホルダーがデータをどのように表示するか (例: BoMリスト、3D view)
3. ユーザが TDP を操作した際に各プレースホルダーがどのような動作をするか (例: BoMリストで選択した部分を 3D ビュー上でハイライトする)

データソースとなるファイルと定義した formula を 3DxSUITE に入力することで、カスタマイズされた TDP が出力されます。

## 4. インストール

### 4.1. 推奨動作環境

表 7. Data Package Studio をインストールするための動作要件

|                            |   |
|----------------------------|---|
| OS                         | バージョン: <ul style="list-style-type: none"><li>Windows 10 (32bit/64bit)</li><li>Windows 11 (32bit/64bit)</li></ul>  |
| Adobe Acrobat <sup>1</sup> | <ul style="list-style-type: none"><li>Adobe Acrobat Pro DC Continuous Release</li><li>Adobe Acrobat Pro 2020 リリース (Classic)</li><li>Adobe Acrobat Pro 2017 リリース (Classic)</li></ul> |

1 : Adobe Acrobat はアドビ社の登録商標です。

### 4.2. 3DxSUITE のインストールおよびバージョンアップ

TDP を作成する際に Data Package Studio の最新機能を利用するためには、3DxSUITE EX9.1 が必要です。



3DxSUITE をインストールするためには **管理者権限** が必要です。

更新もしくはインストールする場合:

- 3DxSUITE インストールフォルダーから "**Elysium\_3DxSUITE\_EX\*.\*\_Components.exe**"<sup>1</sup> を起動します。
- インストーラの指示に従ってセットアップを完了します。

1. アスタリスク2個(\*)が意味するのは、3DxSUITE のバージョンです。例: 9.1

### 4.3. Data Package Studio のインストールおよびバージョンアップ

Data Package Studio は以下の手順でインストールしてください。

- ("Utility\_Tool" フォルダー) から起動:
  - Acrobat Pro 32bit の場合:  
"**Elysium\_3DxSUITE\_EX\*.\*\_Data\_Package\_Studio\_for\_Acrobat\_32bit.exe**"<sup>1</sup>
  - Acrobat Pro 64bit の場合:  
"**Elysium\_3DxSUITE\_EX\*.\*\_Data\_Package\_Studio\_for\_Acrobat\_64bit.exe**"<sup>1</sup>
- インストーラの指示に従ってインストール作業を進めます。

1. アスタリスク 2 個 (\*) が意味するのは、3DxSUITE のバージョンです。例：9.1



1. Data Package Studio をインストールする場合、**管理者権限**が必要です。
2. Data Package Studio は [Adobe Acrobat](#) の "plug\_ins" フォルダにインストールしてください。

## 4.4. Data Package Studio の起動

以下の手順で Data Package Studio を起動してください。

1. Adobe Acrobat を起動します。
2. メニューから [Elysium] > [Start Data Package Studio] を選択します。

以上で [TDP の作成](#) に進む準備が整いました。

## 5. TDP の作成

本節では、最初のTDPを作成する手順について説明します。手順の実施にはおよそ 10 分間かかります。

### 前提条件:

- Data Package Studio (DPS) がセットアップされていること
- 3DxSUITE SmartLauncher (SLS) がセットアップされていること

### サンプルファイル:

本ドキュメントの並びにある "examples" フォルダ内の以下のファイルをご使用ください。

| ファイル   | 内容     |
|--|--------|
| "templates\basic_how_to_guide_template.pdf"  | テンプレート |
| "data_sources\example_throttle_body_r01.enf" | 入力モデル  |

以下の動画では、次の手順について説明しています。

### 手順:

*Data Package Studio* を使用して TDP テンプレートをカスタマイズします:

1. Adobe Acrobatで [basic\\_how\\_to\\_guide\\_template.pdf](#) (ご使用のテンプレート) を開きます。
2. DPS を起動します。(Elysium メニューから [Start Data Package Studio] を選択します。)
3. [Data Sources] タブ/ページでは、ENF タイプのデータソースを追加して、名前に **inputfile** を指定します。
4. [Widgets] タブ/ページでは:

- a. **BoM Table** widget を追加します。(データソースとして **inputfile (ENF)** を選択し、**Part Quantity**、**Part Number**、**Part Material** 列を追加してください。)
- b. **3D View** widget を追加します。(データソースとして **inputfile (ENF)** を選択してください。)
- c. **View Carousel** widget を追加します。(データソースとして **inputfile (ENF)** を選択してください。)

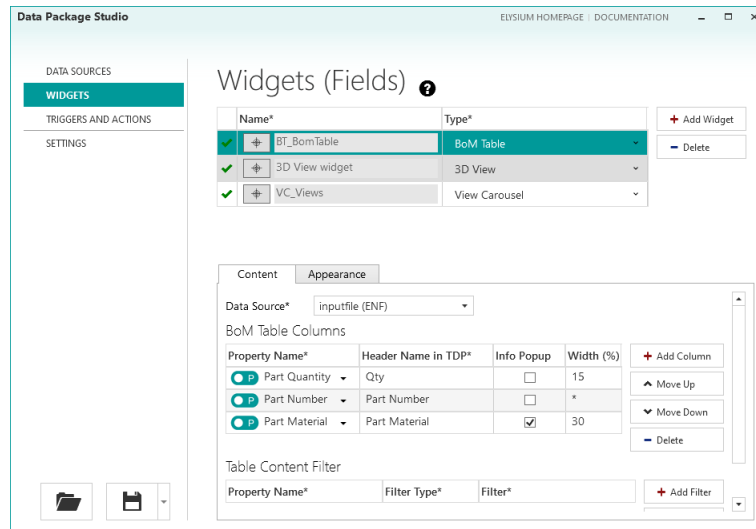


図 3. カスタマイズ後の widgets の例

5. [Triggers and Actions] タブ/ページでは、BoM Table widget (Trigger) と 3D View (Target) に接続する Trigger/Action を追加します。これにより、BoM table 内の項目をクリックすると接続された 3D ビュー内で項目が拡大表示されるようになります。

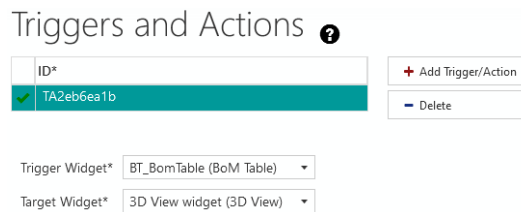


図 4. Trigger/Action を追加した際の例

6. カスタマイズした TDP の formula を任意の場所に保存します。(例: "C:\dps\my\_formula.rb")

SLS で TDP を作成します:

1. SLS で **example\_throttle\_body\_r01.enf** を開きます。(ファイルを右クリックして、Elysium メニューから [変換] を選択します。)
2. 保存するファイルの種類として **3D PDF** を選択します。
3. 3D PDF オプションを構成します ([変換設定...] をクリックします):
  - a. '実行スクリプト (Script)' を **Custom** に設定します。
  - b. 'スクリプトファイル(ファイルパス)(CustomizedScript)' をご使用の formula の場所に設定します。(例: "C:\dps\my\_formula.rb")

- c. 必要に応じて他のパラメーターを設定します。(詳細については、[3DxSUITE パラメーター](#)を参照してください。)
  - d. オプションを確認して保存します。
4. 3D PDF を任意の場所に保存します。(例: "C:\dps\my\_tdp.pdf")

▶ <https://vimeo.com/341267962/be3f3fb882> (Vimeo video)

## 6. 操作ガイド

本節では、より高度な formula の作成と TDP の作成を行うための操作方法をご紹介します。ガイドの作業内容は、完了するのに5分もかかりません。



ガイドで使用するテンプレートファイルとサンプルファイルは、Data Package Studio と併せて提供したドキュメントの並びにある "examples" フォルダーにあります。詳細は[サンプルファイルの設定](#)を参照してください。

前提として、Data Package Studio と 3DxSUITE をインストールして設定してください。これ以外の準備が必要な場合には、それぞれのガイドに記載しています。

本手順の前に、[TDP の作成](#) ガイドを完了しておくことをお勧めします。Data Package Studio (DPS) の一般的なワークフローと、TDP を作成して結果を確認する方法を理解するのに役立ちます。

操作ガイドは以下の 3 つの内容で構成されます:

- [基本機能](#) - 基本操作について (例: [BoM Table Widgetの追加](#))
- [使用事例](#) - 一般的な使用例について (例: [CAD Validator レポートのカスタマイズ](#))
- [TDP の作成](#) - 作成について (例: [3DxSUITE SmartController を用いた TDP の作成](#))

### 6.1. 基本機能

本節では、Data Package Studio (DPS) の基本機能について説明します。以下のサンプルファイルを使用して、様々な DPS のカスタマイズ方法をお試しくささい。



様々なカスタマイズ方法を試す際には "[basic\\_how\\_to\\_guide\\_formula.rb](#)" を修正してください。

#### サンプルファイル:

| ファイル名  | 内容              |
|--|-----------------|
| " <a href="#">templates\basic_how_to_guide_template.pdf</a> "  | テンプレートファイル      |
| " <a href="#">formulas\basic_how_to_guide_formula.rb</a> "     | カスタマイズした入力スクリプト |
| " <a href="#">data_sources\example_throttle_body_r01.enf</a> " | 入力モデル           |

| ファイル名                                       | 内容  |
|---|---|
| "data_sources\example_revision_history.csv" | リビジョン履歴の入力                                  |
| "scenarios\dps_scenario.esa"                | How-To Guide Formula パラメーター<br>セットのシナリオファイル |

### 6.1.1. データソースの追加

本節では、Data Package Studio (DPS) を使用してカスタマイズした formula にデータソースを追加する方法について説明します。データ ソースは TDP の内容となる情報です。これは **widget** の入力や添付ファイルとして使用されます。

2つの **ENF** ファイルと1つの **CSV** ファイルを追加する手順を説明します。CSV ファイルを TDP に添付ファイルとしても追加します。

ENF ファイルには widget で表示するパートおよび 3D データが含まれています。(widget の例: **BoM Table** widget、**3D View** widget、**PMI Table** widget) また事前定義プロパティ (例: パーツ番号) や **Text Field** widget に表示するユーザ定義のプロパティも含まれています。CSV ファイルには、**Custom Table** widget に追加する情報が含まれている場合があります。

#### 手順:

1. Adobe Acrobat で TDP テンプレートを開き、DPS を起動します。
2. [Data Sources] tab/page をクリックします。
3. 最初の ENF データソースを追加し、そのID を **inputfile** に設定します。
  - a. [+ Add Data Source] ボタンをクリックします。
  - b. Type が **ENF** であり、ID が **inputfile** であることを確認します。
4. 2番目の ENF データソースを追加し、任意の名称を設定します。(例: **enf2**)
  - a. [+ Add Data Source] ボタンをクリックします。
  - b. Type が **ENF** になっていることを確認します。
  - c. ID を **enf2** に変更します。
5. CSV データソースを追加します。(例: **revision\_history**)
  - a. [+ Add Data Source] ボタンをクリックします。
  - b. Type を **CSV** に変更します。
  - c. ID を **revision\_history** に変更します。
  - d. CSV のエンコード方式に応じて Encoding の値を **UTF-8If you are** などに変更します。



## Data Sources ?

| ID*                | Type* | Encoding* | Attachment                          | Description |
|--------------------|-------|-----------|-------------------------------------|-------------|
| ✓ inputfile        | ENF   | -         | <input type="checkbox"/>            |             |
| ✓ enf2             | ENF   | -         | <input type="checkbox"/>            |             |
| ✓ revision_history | CSV   | UTF-8     | <input checked="" type="checkbox"/> |             |

図 5. 追加された 3 種類のデータソース

6. **revision\_history** データソースを添付ファイルとして追加します。

- データソースを添付ファイルとして追加するため、「Attachment」チェックボックスをオンにします。

## Data Sources ?

| ID*                | Type* | Encoding* | Attachment                          | Description |
|--------------------|-------|-----------|-------------------------------------|-------------|
| ✓ inputfile        | ENF   | -         | <input type="checkbox"/>            |             |
| ✓ enf2             | ENF   | -         | <input type="checkbox"/>            |             |
| ✓ revision_history | CSV   | UTF-8     | <input checked="" type="checkbox"/> |             |

図 6. *custom\_data* データソースは、添付ファイルとして追加されます

以上で TDP をカスタマイズするために使用する 3 種類のデータソースが作成されました。

## ENF データソースの命名則

DPS では ENF データソースに対して以下のような命名則を適用しています。

CAD Validator (CV) report をカスタマイズして CV 入力ファイルを DPS のデータソースとして使用したい場合、**inputfile1** (CV source) または **inputfile2** (CV target) と名付けます。下図をご参照ください。

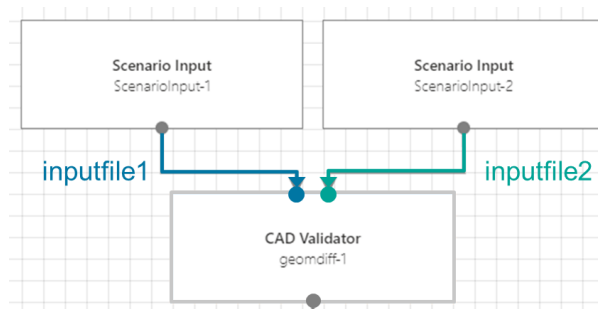


図 7. CAD Validator のソースおよびターゲットファイルは、それぞれ **inputfile1** および **inputfile2** と定義されます

CV 以外の widget で ENF を使用する場合、CV 入力ファイルをデータソースとして明示的に追加する必要があります。すべての CV widget は **CAD Validator Result** データソースを入力として使用します。CV report をカスタマイズして両方の CV 入力ファイルを DPS のデータソースとして使用したい場合、下図のようなデータソースの登録が必要となります。

## Data Sources ?

| ID*                    | Type*                | Encoding* | Attachment                          | Description |
|------------------------|----------------------|-----------|-------------------------------------|-------------|
| ✓ ely_cv_result_source | CAD Validator Result | -         | <input checked="" type="checkbox"/> |             |
| ✓ inputfile1           | ENF                  | -         | <input type="checkbox"/>            |             |
| ✓ inputfile2           | ENF                  | -         | <input type="checkbox"/>            |             |

図 8. DPS で両方の CV 入力ファイルを ENF データソースとして使用する例





CAD Validator レポートのカスタマイズ も参照してください。

上記以外の場合には、ENF データソースのうち1つは **inputfile** と名付ける必要があります。その他の ENF データソースは、独自のものであれば任意の名前を付けることができます。つまり、入力用に2つの ENF ファイルがある場合、例えば下図のようになります。

Data Sources

|   | ID*       | Type* | Encoding* | Attachment               | Description |
|---|-----------|-------|-----------|--------------------------|-------------|
| ✓ | inputfile | ENF ▾ | -         | <input type="checkbox"/> |             |
| ✓ | enf2      | ENF ▾ | -         | <input type="checkbox"/> |             |

図 9. DPS で2つの ENF データソースを使用する例

### 6.1.2. 3D View Widgetの追加およびカスタマイズ

本節では、3D View widget を追加およびカスタマイズするための手順を説明します。3D View widget は **ENF** を入力として受け取り、その内容を3D形状、アノテーションなどとして表示します。TDPを使用すると、3D で表示された内容を回転したりズームしたりすることができます。

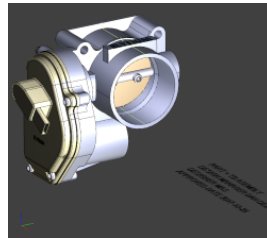


図 10. 3D viewの例

3D View widget を追加する場合、3D View widget 自身 (Action) と同じデータソースに接続されている **View Carousel** widget (Trigger) の間の Trigger/Action ペアも併せて追加されます。Trigger/Action ペアにより、TDP を閲覧する際3Dビューに表示するビューを選択することができるようになります。



本手順を実施する前に **データソースの追加** をご覧ください。

#### 手順:

1. Adobe Acrobat で TDP テンプレートを開き、DPS を起動します。
2. ENF データソースを追加し、**inputfile** という名前を付けます。
3. 3D View widget を追加します。
  - a. [+ Add Widget] ボタンをクリックします。
  - b. 新しい widget のタイプとして **3D View** を選択します。
  - c. 新しい 3D View widget を TDP テンプレートのプレースホルダーに接続します。

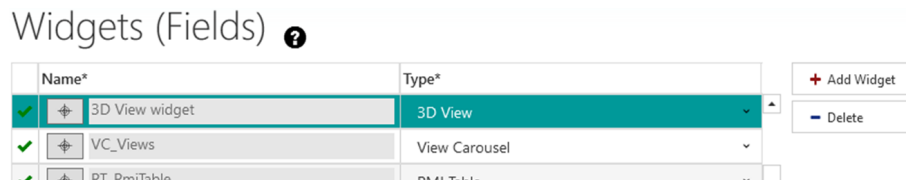


図 11. 3D View widget を追加します

#### 4. 3D View widget の内容を指定します。

- a. データソースとして「inputfile (ENF)」を選択します。
- b. Default view を指定します。(例: TDP が開いているときにビューが表示されます。)<sup>1</sup>
  - i. **First View (ordered by name)** は、英数字順に名前です。
  - ii. **CAD Default View** は、元のCADシステムでデフォルトとして指定されているビューです。
  - iii. **Named View** は、特定の名前と一致したときの最初のビューです。

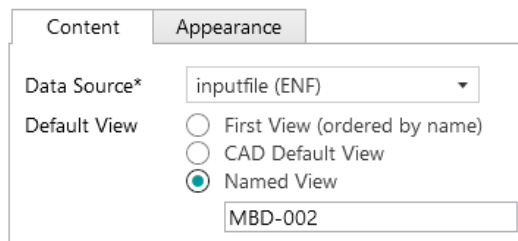


図 12. 3D View widget の内容に関する指定

1) ビューが表示されない場合は、フォールバックにより chronological order の最初のビューを使用、それでも表示されない場合は ISO view を生成します。

#### 5. 3D View widget の外観をカスタマイズします。

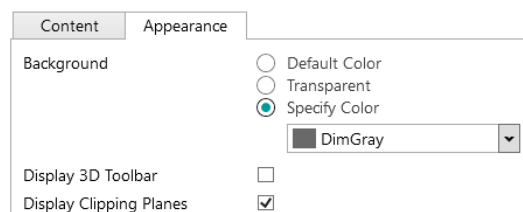


図 13. 3D View widget の外観に関する指定

### 6.1.3. View Carousel Widgetの追加およびカスタマイズ

本節では、View Carousel widget を追加およびカスタマイズする手順を説明します。View Carousel widget は、3D モデルの保存済みビューをグリッドレイアウトのサムネイルとして表示します。ENF データソースタイプを入力として受け取ります。

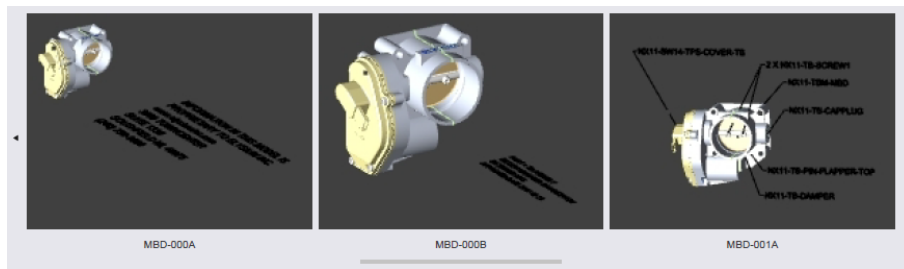


図 14. View carousel の例

View Carousel widgetを使用して、3D View widget に表示する保存済みビューを操作することができます。これは、View Carousel widget、3D View widget、および BoM Table widget が同じデータソースに接続され、BoM Table widget (Trigger) と 3D View widget (Action) の間に Trigger/Action 接続がある場合に自動的に有効になります。

以下の動画では、次の手順について説明しています。

#### 手順:

1. Adobe Acrobat で TDP テンプレートを開き、DPS を起動します。
2. ENF データソースを追加し、**inputfile** という名前を付けます。
3. View Carousel widget を追加します。
  - a. [+ Add Widget] ボタンをクリックします。
  - b. 新しい widget のタイプとして **View Carousel** を選択します。
  - c. 新しい View Carousel widget を TDP テンプレートのプレースホルダーに接続します。
4. View Carousel widget の内容を指定します。（データソースとして **inputfile (ENF)** を選択します）
5. 必要な場合には、コンテンツをフィルタリングするために **フィルター** を追加します。
6. View Carousel widget の外観をカスタマイズします。

▶ <https://vimeo.com/373805643/7c36189138> (Vimeo video)

### 6.1.4. Button Widget の追加

Button Widget の追加について詳しく説明します。Button は、**ビューの印刷の有効化** で説明しているように、ビューの印刷などのアクションを起動するために使用することができます。データソースタイプを入力として必要としません。

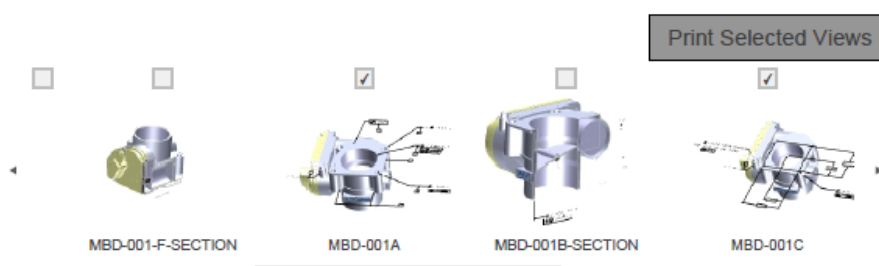


図 15. 選択したビューを印刷する Button の例

## 手順:

1. Adobe Acrobat で TDP テンプレートを開き、DPS を起動します。
2. Button widget を追加します。
  - a. [+ Add Widget] をクリックします。
  - b. 新しい widget のタイプとして **Button** を選択します。
  - c. 新しい Button widget を TDP テンプレートのプレースホルダーに接続します。

### Widgets (Fields) ?

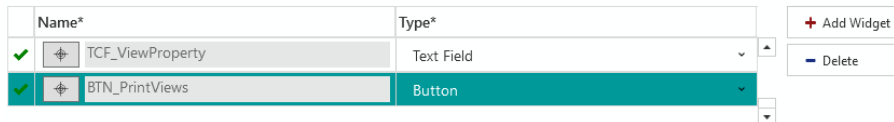


図 16. Button widget の追加

3. Button widget のラベルを指定します。(例: **選択したビューを印刷**)

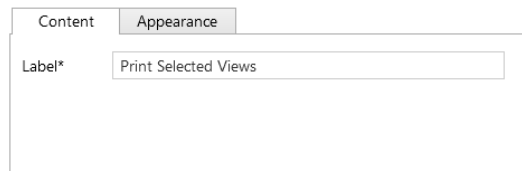


図 17. Button widget ラベルの指定

4. Button widget の外観をカスタマイズします。(widget の外観 を参照)
5. Trigger/Action を追加します。(詳細は [Trigger/Action 接続の追加] を参照)

## 6.1.5. Text Field Widgetの追加

本節では、Text Field widget を追加する方法について説明します。Text Field widget は、データソースのコンテンツを TDP のプレーンテキストとして表示します。ENF データソースタイプと表示するプロパティの名前を入力として受け取ります。

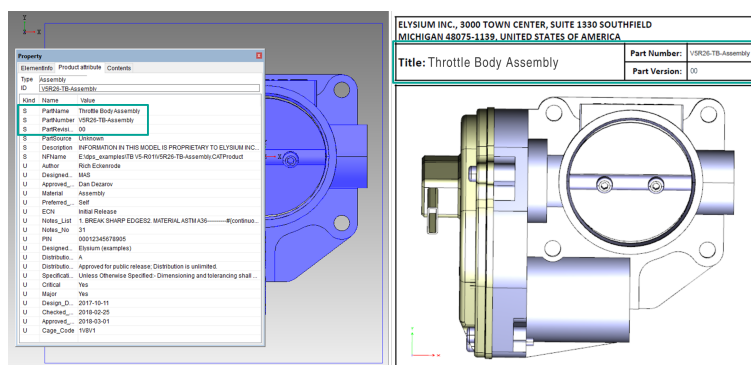


図 18. モデルプロパティの例(左)と Text Field に表示された文字列(右)

また、Text Field widget は **text file** データソースタイプを入力として受け取ることもできます。これは例えば EAR/ITAR 注記を表示するのに役立ちます。このような場合、プロパティ名を指定する必要はありません。(ファイル全体の内容が表示されます)

Text Field widget のデータソースに **by Trigger/Action** を指定することもできます。この場合 Trigger widget で使用可能なプロパティとイベントに応じて、Text Field に何をいつ表示するかを制御することができます。(Trigger と Action の追加を参照ください。)

#### 手順:

1. Adobe Acrobat で TDP テンプレートを開き、DPS を起動します。
2. ENF データソースを追加し、**inputfile** という名前を付けます。
3. Text Field widget を追加します。
  - a. [+ Add Widget] ボタンをクリックします。
  - b. 新しい widget のタイプとして **Text Field** を選択します。
  - c. 新しい Text Field widget を TDP テンプレートのプレースホルダーに接続します。
4. Text Field widget の内容を指定します。
  - a. データソースとして **inputfile (ENF)** を選択します。
  - b. Text Field widget に表示するプロパティを指定します。(例: **Part Number**)
  - c. オプションとして、Text Field widget が編集可能になるよう指定するため「Text is Editable」チェックボックスをオンにします。(データソースタイプが **ENF** または **Text** のときに限り有効です。)
5. Text Field **widget の外観** をカスタマイズします。

### 6.1.6. PMI Table Widgetの追加

本節では、PMI Table widget を TDP に追加する方法について説明します。PMI Table widget は、PMI の項目 (例: データム、寸法、幾何公差) と指定されたプロパティを表形式で一覧表示します。ENF データソースタイプを入力として受け取ります。

| Dimension / Geom. Tol. - Unit | Type                      |
|-------------------------------|---------------------------|
| 0.005   A   H   D             | inch gdt                  |
| 0.005   A   B                 | inch gdt                  |
| 0.005   A   G   D-E           | inch gdt                  |
| 0.005   A   G   D-E           | inch gdt                  |
| 0.005   F                     | inch gdt                  |
| 0.01   A                      | inch gdt                  |
| 0.01   A   B   D-E            | inch gdt                  |
| 0.01   A   B   D-E            | inch gdt                  |
| 0.01   A   G   D-E            | inch gdt                  |
| 0.02   J                      | inch gdt                  |
| 1.000                         | inch dimension / distance |
| 1.000 -0.005/-0.005           | inch dimension / distance |

図 19. PMI table の例



使用可能な PMI タイプとプロパティについては **事前定義 PMI プロパティ** を参照してください。

以下の動画では、次の手順について説明しています。

**手順:**

1. Adobe Acrobat で TDP テンプレートを開き、DPS を起動します。
2. ENF データソースを追加し、**inputfile** という名前を付けます。
3. PMI Table widget を追加します。
  - a. [+ Add Widget] ボタンをクリックします。
  - b. 新しい widget のタイプとして **PMI Table** を選択します。
  - c. 新しい PMI Table widget を TDP テンプレートのプレースホルダーに接続します。
4. PMI Table widget の内容を指定します。
  - a. データソースとして **inputfile (ENF)** を選択します。
  - b. PMI Table widget に含める PMI タイプを指定します。
  - c. PMI Table widget に列を追加するには、[+ Add Column] ボタンをクリックします。
  - d. 列に表示するプロパティを指定します。(例: **Text Rep.**)
  - e. 各プロパティを追加するには、手順の 4.c と 4.d を繰り返し行います。
5. 必要な場合には、コンテンツをフィルタリングするために **フィルター** を追加します。
6. PMI Table **widget の見た目** をカスタマイズします。

▶ <https://vimeo.com/373801203/71d5d891e9> (Vimeo video)

### 6.1.7. BoM Table Widgetの追加

本節では、BoM Table widget の追加方法について説明します。BoM Table には、部品と指定した プロパティが表形式で一覧表示されます。**ENF** データ ソースタイプを入力として受け取ります。

| Name             | QTY |
|------------------|-----|
| NX11-TB Assembly | 1   |
| NX11-TBM-MBD     | 1   |
| NX11-TB-Pin-Gear | 1   |

図 20. BoM tableの例

**手順:**

1. Adobe Acrobat で TDP テンプレートを開き、DPS を起動します。
2. ENF データソースを追加し、**inputfile** という名前を付けます。
3. BoM Table widget を追加します。
  - a. [+ Add Widget] ボタンをクリックします。
  - b. 新しい widget のタイプとして **BoM Table** を選択します。
  - c. 新しい BoM Table widget を TDP テンプレートのプレースホルダーに接続します。

## Widgets (Fields) ?

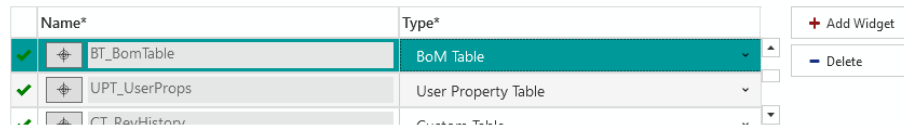


図 21. BoM Table widget を追加します

4. BoM Table widget の内容を指定します。
  - a. データソースとして **inputfile (ENF)** を選択します。
  - b. [+ Add Column] ボタンをクリックして、BoM Table widget に列を追加します。
  - c. 列に表示するプロパティを指定します。 (**Part Quantity** 等)
  - d. 追加するプロパティごとに、手順 4.b と 4.c を繰り返します。

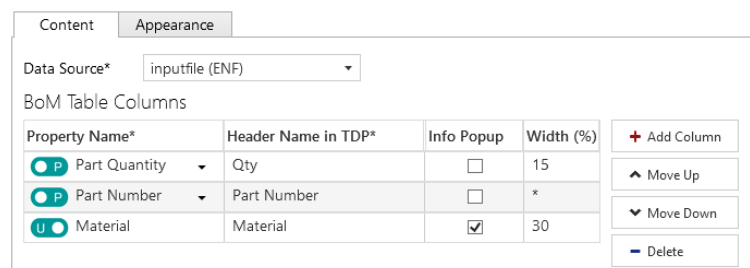


図 22. BoM Table widget の列と、それらにリンクするプロパティを指定します



スイッチを **P** から **U** に切り替えることで、事前定義されたプロパティの代わりにユーザ定義のプロパティを追加することもできます。

5. 必要な場合には、コンテンツをフィルタリングするために **フィルター** を追加します。
6. BoM Table widget の **見た目** をカスタマイズします。

### 6.1.8. Custom Table Widgetの追加

本節では、Custom Table widget を追加する方法について説明します。Custom Table widget は、**CSV** ファイルを入力として受け取り、その内容をテーブルに表示します。

|   | A        | B                            | C          | D       |
|---|----------|------------------------------|------------|---------|
| 1 | Revision | Description                  | Date       | ECN     |
| 2 | Rev_000  | Initial release.             | 13/03/2018 | -       |
| 3 | Rev_001  | The cap pin location change. | 15/03/2018 | ECN-001 |

| Revision | Description                  | Date       | ECN     |
|----------|------------------------------|------------|---------|
| Rev_000  | Initial release.             | 13/03/2018 | -       |
| Rev_001  | The cap pin location change. | 15/03/2018 | ECN-001 |

図 23. CSV のリビジョン履歴データ (上) を Custom table に表示した例 (下)



詳細は [example\\_revision\\_history.csv ファイル](#) を参照してください。



CSV ファイルの最初の行は、Data Package Studio がヘッダとして使用します。



**手順:**

1. Adobe Acrobat で TDP テンプレートを開き、DPS を起動します。
2. CSV データソースを追加し、**revision\_history** という名前を付けます。
3. Custom Table widget を追加します。
  - a. [+ Add Widget] ボタンをクリックします。
  - b. 新しい widget のタイプとして **Custom Table** を選択します。
  - c. 新しい Custom Table widget を TDP テンプレートのプレースホルダーに接続します。

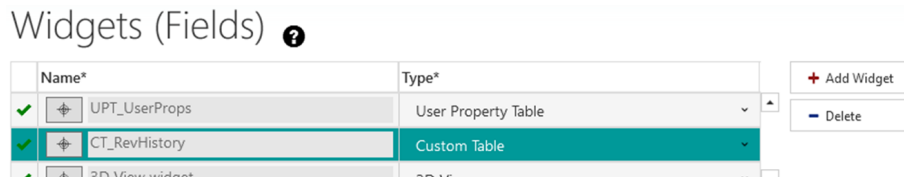


図 24. Custom Table widget を追加します

4. Custom Table widget の内容を指定します。
  - a. **revision\_history (CSV)** をデータソースとして選択します。
  - b. [+ Add Column] ボタンをクリックして Custom Table widget に列を追加します。
  - c. 列に表示する CSV 列名を指定します。(例: **Revision**)
  - d. 追加するプロパティごとに手順 4.b と 4.c を繰り返します。

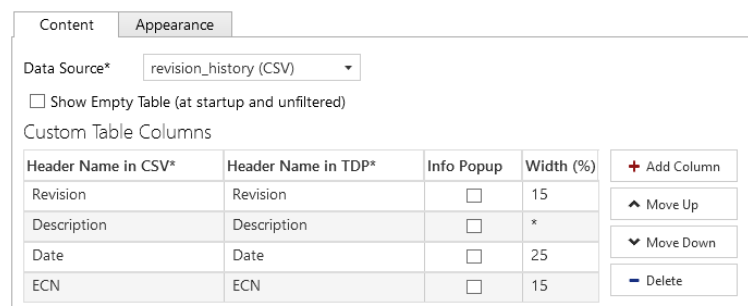


図 25. Custom Table widget の列と、それらにリンクする CSV 列を指定します

5. 必要な場合には、コンテンツをフィルタリングするために **フィルター** を追加します。
6. Custom Table widget の見た目をカスタマイズします。

### 6.1.9. User Property Table Widgetの追加

本節では、User Property Table widget を追加する方法について説明します。User Property Table widget には、モデルのパート/アセンブリに関連するユーザ定義プロパティが、テーブル内のキー/値のペアとして一覧表示されます。



| Property    |                   |   | Property          | Value                            |
|-------------|-------------------|---|-------------------|----------------------------------|
| ElementInfo | Product attribute | Contents                                  |                   |                                  |
| Type        | Assembly          |   | Author            | Lynn Garcia                      |
| ID          | TB-Assembly       |   | Designed_By       | Michael Reynolds                 |
| Kind        | Name              | Value                                     | Approved_By       | James Payne                      |
| U           | Author            | Lynn Garcia                               | Material          | Assembly                         |
| U           | Designed_By       | Michael Reynolds                          | Preferred_Vendor  | Self                             |
| U           | Approved_By       | James Payne                               | ECN               | Initial Release                  |
| U           | Material          | Assembly                                  | Notes_List        | 1. BREAK SHARP EDGES 2. MATERIAL |
| U           | Preferred_Vendor  | Self                                      | Notes_No          | 33                               |
| U           | ECN               | Initial Release                           | Cage_Code         | 1V8V1                            |
| U           | Notes_List        | 1. BREAK SHARP EDGES 2. MATERIAL ASTM ... | PIN               | 00012345678905                   |
| U           | Notes_No          | 33  | Designed_For      | Elysium (examples)               |
| U           | Cage_Code         | 1V8V1                                     | Distribution_Code | A                                |
| U           | PIN               | 00012345678905                            |                   |                                  |
| U           | Designed_For      | Elysium (examples)                        |                   |                                  |
| U           | Distribution_Code | A   |                   |                                  |

図 26. ユーザ定義プロパティの例（左）と User Property Table の表示例（右）

他の widget とは対照的に、User Property Table widget はデータソースを必要としません。代わりに [BoM Table widget](#) から [Trigger/Action](#) への接続が必要です。これにより、接続されている BoM Table widget で選択されている部品/アセンブリのすべてのユーザー定義プロパティを表示することができます。

#### 手順:

1. Adobe Acrobat で TDP テンプレートを開き、DPS を起動します。
2. ENF データソースを追加し、inputfile という名前を付けます。
3. BoM Table widget を追加し、inputfile データソースに接続します。（詳細は [BoM Table Widget の追加](#) を参照してください。）
4. User Property Table widget を追加します。
  - a. [+ Add Widget] をクリックします。
  - b. 新しいウィジェットのタイプとして「User Property Table」を選択します。
  - c. 新しい User Property Table widget を TDP テンプレートのプレースホルダーに接続します。

#### Widgets (Fields) ?

| Name*         | Type*               |  |
|---------------|---------------------|--|
| BT_BomTable   | BoM Table           | <a href="#">+ Add Widget</a><br><a href="#">- Delete</a> |
| UPT_UserProps | User Property Table |  |
| CT_BomHistory | Custom Table        |  |

図 27. User Property Table widget を追加します

5. User Property Table widget の内容を指定します。
  - a. プロパティ名のヘッダー名を指定します。（例: Property）
  - b. プロパティ値のヘッダー名を指定します。（例: Value）

|   |            |
|---|------------|
| Content                                 | Appearance |
| Header Name of Property Name (in TDP)*  | Property   |
| Header Name of Property Value (in TDP)* | Value      |

図 28. User Property Table widget 列のヘッダー名を指定します

6. 必要な場合には、コンテンツをフィルタリングするために [フィルター](#) を追加します。
7. Trigger/Action を追加します。（詳細は [Trigger と Action の追加](#) を参照してください。）
  - a. Trigger Widget として BoM Table widget を選択します。

- b. Target Widget として User Property Table widget を選択します。

### 6.1.10. Table Column Filter Widgetの追加

本節では、Table Column Filter widget を追加する方法について説明します。これにより、接続された表をフィルタリングすることができるようになります。 [User Property Table widget](#) と同様に、データソースを入力として受け取りませんが、フィルタリングする表への [Trigger/Action](#) 接続が必要です。

| Name             | QTY |
|------------------|-----|
| NX11-TB Assembly | 1   |
| NX11-TBM-MBD     | 1   |
| NX11-TB-Pin-Gear | 1   |

| Name         | QTY |
|--------------|-----|
| NX11-TBM-MBD | 1   |

図 29. 例: table column filter (左)と表の "Name" 列にフィルタを適用した場合の例(右)

#### 手順:

1. Adobe Acrobat で TDP テンプレートを開き、DPS を起動します。
2. ENF データソースを追加し、**inputfile** という名前を付けます。
3. Table Column Filter widget を追加します。
  - a. [+ Add Widget] ボタンをクリックします。
  - b. 新しい widget のタイプとして **Table Column Filter** を選択します。
  - c. 新しい Table Column Filter widget を TDP テンプレートのプレースホルダーに接続します。
  - d. Table Column Filter [widget の外観](#) をカスタマイズします。

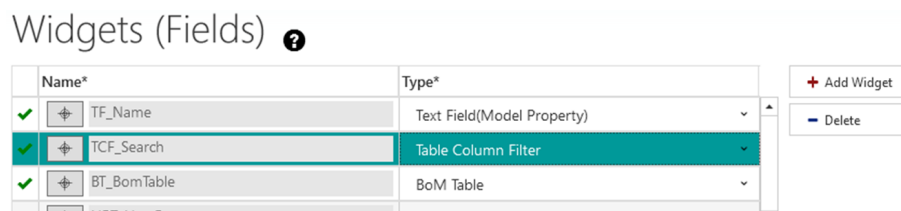


図 30. Table Column Filter widgetの追加

4. BoM Table widget を追加します (詳細は [BoM Table Widgetの追加](#) を参照してください)。
5. Trigger/Action を追加します。(詳細は [Trigger と Action の追加](#) を参照してください。)
  - a. Trigger Widget として Table Column Filter widget を選択します。
  - b. Target Widget として BoM Table widget を選択します。
  - c. BoM Table widget でフィルターする列を選択します。

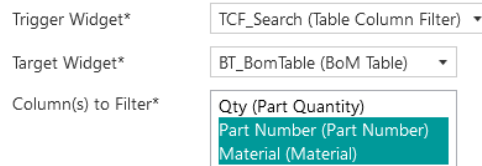


図 31. Table Column Filter と BoM Table widget の間への Trigger/Action 接続の追加

### 6.1.11. Widgetをカスタマイズ

本節では、BoM Table widget を使用する例を通じて、widget の外観をカスタマイズする方法について説明します。表には独自の外観オプションがいくつかありますが、ここで説明した方法で他のウィジェットにも簡単に適用することができます。

最初に出てくるいくつかの手順については、[BoM Table Widgetの追加](#) ガイドにより詳細な説明がありますので、事前にそちらを参照することをお勧めします。

以下の動画では、次の手順について説明しています。

1. Adobe Acrobat で TDP のテンプレートを開き、DPS を起動します。
2. データソースを追加します。(例: **inputfile** という名前の ENF ファイル)
3. BoM Table widget を追加して内容をカスタマイズします:
  - a. データソースを追加します。(例: **inputfile (ENF)**)
  - b. 列を追加します。(例: **Part Quantity**、**Part Number**と**Material**)
  - c. 各列の幅をカスタマイズします。(例: **15%**、**\*1**、**30%**)
    - i. 当該列においてセル内容のポップアップ表示を許可するかを指定します。

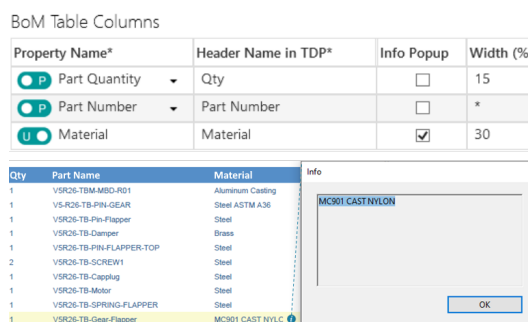


図 32. セル内容のポップアップ表示を有効化

1. Widgets / Appearance タブでは、次の手順で表の外観をカスタマイズします。
  - a. 表示する行の数 (例: **14**) - 表の項目数が表示する行よりも多い場合、スクロールが有効になります。
  - b. スクロールバーのサイズ (例: **15**)
  - c. 表の見出しの外観をカスタマイズします。
    - i. フォントサイズ (例: **11**)
    - ii. テキストの色 (例: **White**)

- iii. テキストの配置 (例: **Center**)
  - iv. 大文字/小文字 (例: **UPPERCASE**)
  - v. 背景色 (例: **#00A395**)
- d. 表本体の外観をカスタマイズします。
- i. フォントサイズ (例: **8**)
  - ii. テキストの色 (例: **#00A395**)
  - iii. テキストの配置 (例: **Center**)
  - iv. 大文字/小文字 (例: **lowercase**)
  - v. 枠線の太さ (例: **1**) - デフォルトは「0」です。
  - vi. 枠線の色 (例: **#00A395**) - 厚みが「0」以外に適用されます。
  - vii. 奇数行の背景色 (例: **White**) - 1行目、3行目など奇数行に適用されます。
  - viii. 偶数行の背景色 (例: **White**) - 2行目、4行目など偶数行に適用されます。
  - ix. ハイライト色 (例: **LightGoldenrodYellow**) - 選択した行に適用されます。

1: \* とは、"distribute" と results で使用可能な幅 (100% から明示的に指定された幅) がそのような列に均等に分配されます。



ゼブラ表示の表の奇数行と偶数行には異なる背景色を設定してください。

▶ <https://vimeo.com/373805771/6d8e50285c> (Vimeo video)

## 6.1.12. Trigger と Action の追加

本節では、**trigger** と **action** を **widget** の間に追加する方法について説明します。

以下の動画では、次の手順について説明しています。

手順:

1. Adobe Acrobat で TDP テンプレートを開き、DPS を起動します。
2. ENF データソースを追加し、「inputfile」という名前を付けます。
3. BoM Table widget を追加します。(BoM Table Widgetの追加を参照)
4. 3D View widgetを追加します。(3D View Widgetの追加およびカスタマイズを参照)
5. BoM Table widget と 3D View widget の間に Trigger/Action 接続を追加します。
  - a. [Triggers and Actions] タブ > [ + Add Trigger/Action ] を選択します。
  - b. Triggerとして BoM Table widget を選択します。
  - c. ターゲットとして 3D View widget を選択します。



手順 5 では、一部の widget の組合せについては [ + Add Return Trigger/Action ] ボタンが表示されます。クリックすると他の方向への trigger/action 接続も追加することができます。

▶ <https://vimeo.com/389388993/60be240065> (Vimeo video)

## 6.2. 使用事例

### 6.2.1. TDP のテンプレートの作成

本節では、カスタマイズした TDP テンプレートを作成する方法について説明します。テンプレートを最初からデザインする場合、完了までにおよそ1時間15分かかります。

#### サンプルファイル:

| ファイル   | 内容          |
|--|-------------|
| "templates\tdp_example_template_design.pptx" | テンプレートのデザイン |

### TDP の作成におけるテンプレートの役割

テンプレートのデザインについて説明する前に、まずはテンプレート、Data Package Studio および 3DxSUITE が TDP の作成で果たす様々な役割について説明します。役割を理解することで、より良いテンプレートをより迅速に作成できるようになります。

- テンプレートは以下のような役割の PDF です。
  - TDP の静的コンテンツ (例: ロゴタイプやテキストなど) を定義します。
  - 動的コンテンツのプレースホルダーを用意します。これは TDP 作成時に利用されます。
- Data Package Studio は [formula](#) を提供します。
  - 動的コンテンツ (データソース) を、[widget](#) を介してプレースホルダーに接続します。
  - 動的コンテンツの表示方法 (widgetタイプと外観) を指定します。
- 3DxSUITE は、指定されたテンプレート、formula およびデータソースに基づいて TDP を作成します。

Data Package Studioの観点から見ると、テンプレートの最も重要な要素は widget プレースホルダーです。Widget プレースホルダーが無い場合、Data Package Studio には TDP で表示する内容、場所、方法を指定することができません。

テンプレートで widget プレースホルダーを定義します。以下の図のように、テキストタイプの PDF フォームフィールドを追加するだけです。

The image shows a portion of a PDF form template. At the top, there is a header block with the text: "ELYSIUM INC., 3000 TOWN CENTER, SUITE 1330 SOUTHFIELD MICHIGAN 48075-1139, UNITED STATES OF AMERICA". Below this, there are several form fields. One field is labeled "Title:" and contains the text "Title". Another field is labeled "Part Number:" and contains the text "PartNumber". A third field is labeled "Distribution Co" and contains the text "Field Name:". A yellow highlight is placed over the "Field Name:" field, and a dropdown menu is open, showing "PartNumber" as the selected option. Below the dropdown, there is a checkbox labeled "Required field" and a link labeled "All Properties".

図 33. PDF Form フィールド (テキスト) を追加して widget プレースホルダーを作成した例

## テンプレートを作成

目標は、TDP（右下）を作成するためのテンプレート（左下）をデザインすることです。

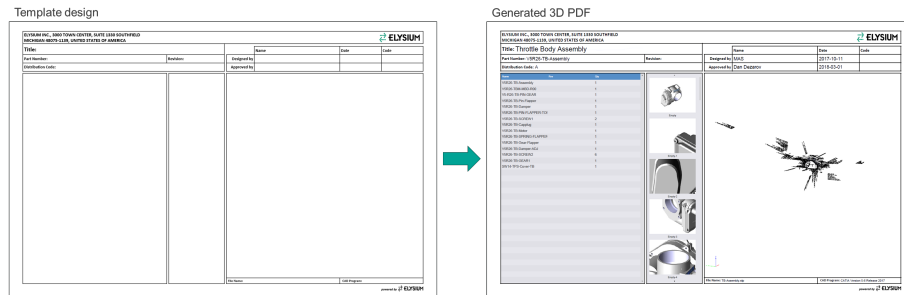


図 34. テンプレートのデザイン

テンプレートを作成する手順は以下の通りです。ここでは、PowerPoint と Adobe Acrobat Pro を使用することを想定しています。PowerPoint を使用してテンプレートの複雑なグリッド レイアウトをデザインし、Adobe Acrobat を使用してテンプレートの作成とプレースホルダー の調整を行います。



PowerPoint の代わりに、PDF を保存できる任意のアプリケーションを使用 することができます。

### 手順:

1. PowerPoint でテンプレートをデザインします。
  - a. 新たに PowerPoint ファイルを作成します。
  - b. 表を挿入します。(例: size = 1x1)
  - c. 表の幅と位置、背景色、罫線の幅と色を適宜調整します。
  - d. セルの分割機能や結合機能を用いて以下の図のように表を調整します。
  - e. 以下の図にあるようにテキストを静的ラベルとして追加します。
  - f. 以下の図にあるようにロゴを追加します。

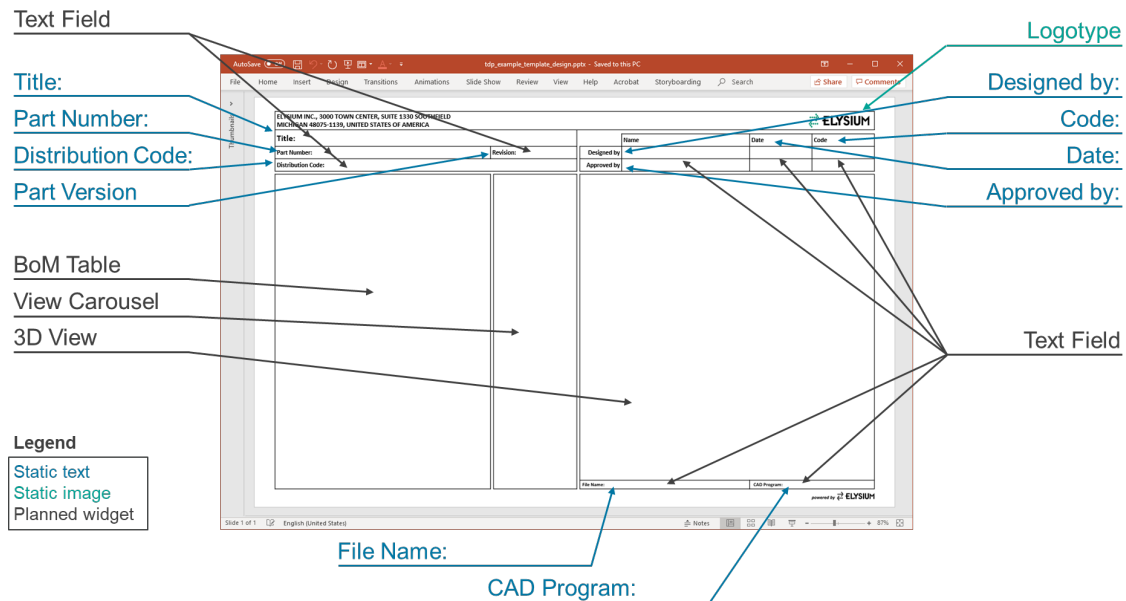


図 35. PowerPoint を用いたテンプレートのデザイン (Widget の配置予定位置を含む)



上記の手順はサンプルテンプレートの「tdp\_example\_template\_design.pptx」を使用すれば、作業時間を大幅に短縮することができます。

## 2. PowerPoint から TDP テンプレートを作成します。

- PowerPoint のホームリボンで [Adobe PDF の作成および共有] をクリックします。
- 作成した PDF を保存します。

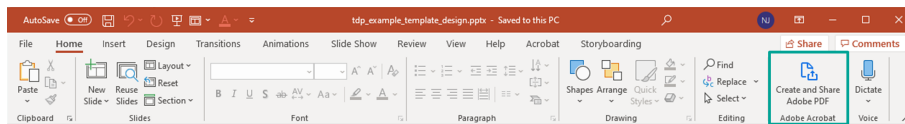


図 36. PowerPoint デザインからの TDP テンプレート作成

## 3. Adobe Acrobat Pro で PDF フィールドを作成します。

- Adobe Acrobat Pro で作成した PDF を開きます。ツールから [フォームを準備] を選択します。
- テンプレートファイルを指定します。（“文書をスキャン” を選択しないでください）[開始] を選択して PDF フィールドを作成します。

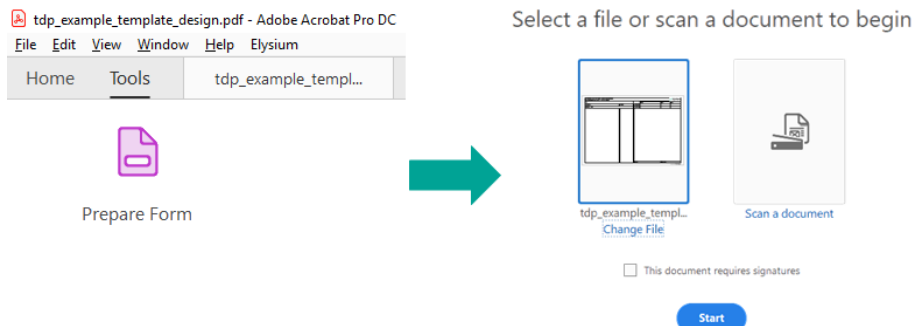


図 37. Adobe Acrobat Pro による PDF フィールドの作成

## 4. 必要に応じて Adobe Acrobat Pro の機能でテンプレートの調整を行います。（作成した PDF フィ



ールドの名前やサイズの調整など)

- a. 必要に応じて PDF フィールドの名前を変更します。
- b. 作成された PDF フィールドのサイズと位置を適宜調整します。
- c. 作成されなかった PDF フィールドがあれば追加します。(サンプルの PowerPoint デザインテンプレートを使用した場合、"Distribution Code" フィールドが欠落している可能性があります。)
- d. 不要な PDF フィールドがあれば削除します。(サンプルの PowerPoint デザインテンプレートを使用する場合、フィールドを削除する必要はありません。)

5. 調整したテンプレートを保存します。

以上で Data Package Studio で使用できるテンプレートが完成しました。

## 6.2.2. CAD Validator レポートのカスタマイズ

本節では、CAD Validator レポートのカスタマイズを作成する手順について説明します。TDP のカスタマイズ実施には、3DxSUITE SmartLauncher (SLS) をご使用ください。



CAD Validator レポートのウィジェットやテンプレートの基本操作に関する 導入は、[CAD Validator レポートテンプレートの新しいフォーマットへのマイグレーション](#) の付録を参照してください。

サンプルファイル:

| ファイル名  | 内容   |
|--|--|
| "data_sources\example_throttle_body_r01.enf" | 1番目の入力モデル (変更後)                                  |
| "data_sources\example_throttle_body_r00.enf" | 2番目の入力モデル (変更前)                                  |
| "templates\cv_example_template.pdf"          | CAD Validator レポートのテンプレート                        |
| "formulas\cv_formula.rb"                     | カスタマイズした CAD Validator レポートを作成する Formula およびシナリオ |
| "scenarios\cv_scenario.esa"                  |  |

この後の節では、以下の手順を説明します。

1. [CAD Validator レポートテンプレートの作成](#)
2. [CAD Validator レポート Formula の作成](#)
3. [CAD Validator レポートシナリオ の作成](#)
4. [SLS による CAD Validator レポートの生成](#)



## CAD Validator レポートテンプレートの作成

CAD Validator レポートテンプレートの作成には、レポートに含めたい各widgetのプレースホルダー (PDF Form fields) を持つPDFを作成する必要があります。テンプレートの作成方法および使用可能な CAD Validator widgets のリストの [CAD Validator Widget タイプ](#) については、[TDP のテンプレートの作成](#) を参照してください。



カスタマイズされたデータのプレースホルダー (PDF fields) に、「ELY\_」で始まる名前を付けないでください。(予約済みの PDF フィールド名を参照してください。)

以上で、以下の図にあるようなテンプレートが完成します。

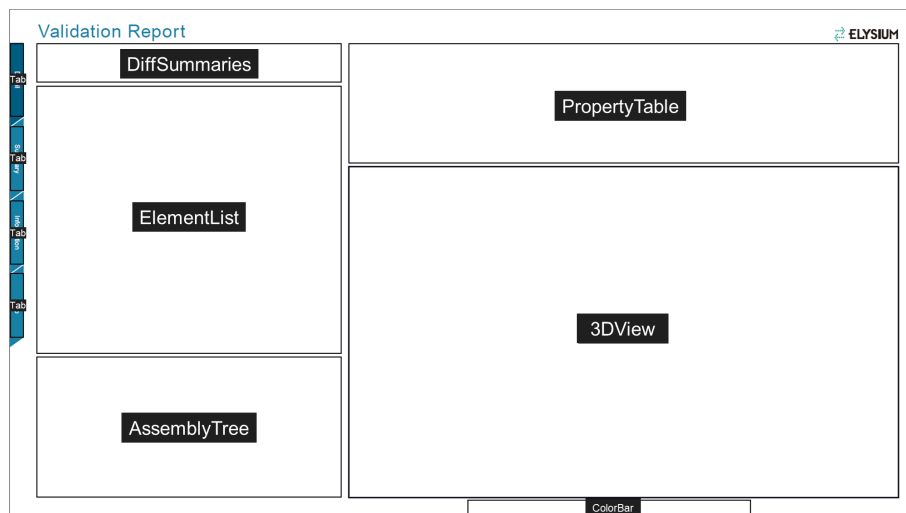


図 38. 例: CAD Validator レポートテンプレート

## CAD Validator レポート Formula の作成

本節では、カスタマイズされた CAD Validator レポートテンプレートの formula 作成手順について説明します。すべての CAD Validator widgets ではなく、[Difference Summary](#), [Detail List](#), [3D View](#) widgets など最も重要なもののみです。

手順:

1. Adobe Acrobat Pro でテンプレートを開きます。
2. Data Package Studio (DPS) を起動します。(Elysium ▶ **Start Data Package Studio**を選択します。)
3. **[Data Sources]** タブ/ページでは、CAD Validator 結果タイプのデータソースを追加して、名前に **cv-result** を指定します。



CAD Validator 結果データソースを追加する際、すべての CAD Validator widgets が自動的に **[Widgets]** タブ/ページに追加されます。

4. **[Widgets]** タブ/ページでは:
  - a. 3D View (CAD Validator) widget を選択します。
  - b. レポートを開いたときにすべての PMI 項目を非表示にする場合は "Show PMI (at startup)" の

チェックをオフにします。

- c. レポートに含めたくないツールバー項目のチェックをオフにします。(例: Render Mode (Selector), Action Menu / Show Unselected Parts (Checkbox), Refresh (Button))

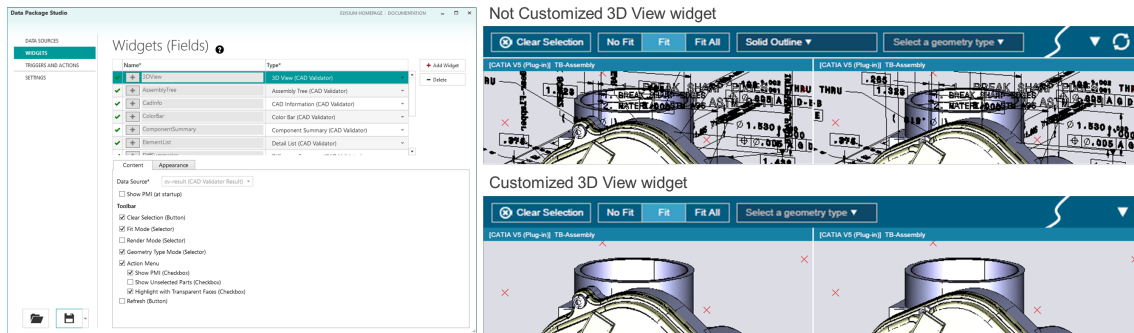


図 39. 3D View widget カスタマイズの例: DPS オプション (左)、カスタマイズなしのレポート (右上)、カスタマイズなしのレポート (右下)

## 5. [Widgets] タブ/ページでは:

- Detail List (CAD Validator) widget を選択します。
- レポートに含めたくないアイテムのチェックを外します。(例: PMI (Checkbox), Group List / Action Menu, そして Element List / Action Menu / Copy to Group)

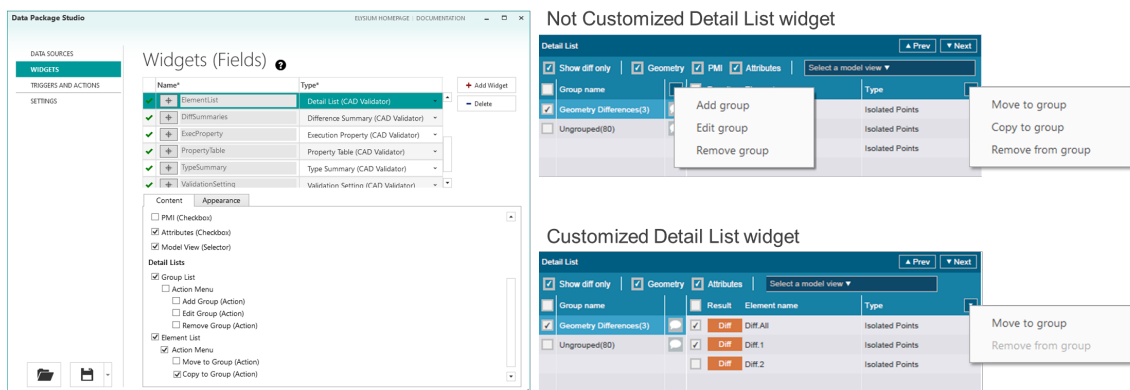


図 40. Detail List widget カスタマイズの例: DPS オプション (左)、カスタマイズなしのレポート (右上)、カスタマイズありのレポート (右下)

## 6. [Widgets] タブ/ページでは:

- Difference Summary (CAD Validator) widget を選択します。
- レポートに含めたくないアイテムのチェックを外します。(例: Passed/Failed (Assessment), PMI (Differences)).

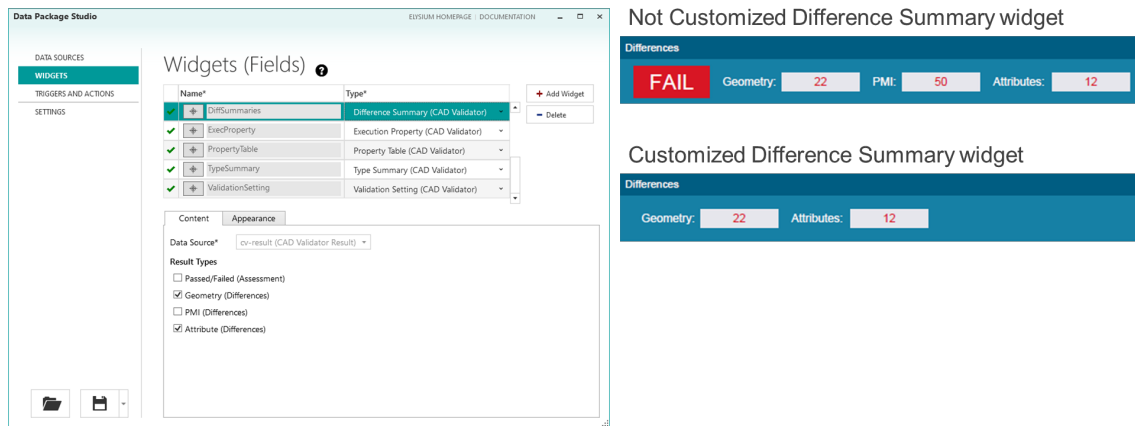


図 41. Difference Summary widget カスタマイズの例: DPS オプション (左)、カスタマイズなしのレポート (右上)、カスタマイズありのレポート (右下)

## CAD Validator レポートシナリオ の作成

カスタマイズされたレポートを生成する前に、以下のパラメーターを設定する必要があります。  
( [CAD Validator Component](#) で必要となるパラメーター を参照してください。 )

- `Create3DPdfReport=1`
- `3DPdfReportType=Custom`
- `Customized3DPdfReportScript=<PATH_TO_FORMULA>`

3DxSUITE ScenarioEditorを使用したシナリオを作成します。 カスタマイズされた CAD Validator レポートを生成する前に、3DxSUITE SmartLauncher (SLS) シナリオフォルダにシナリオを保存 します。

### 手順:

- 3DxSUITE ScenarioEditor を開きます。
- CAD Validator component を追加します。
  - 左手のメニューから Optimizer item を展開します。
  - CAD Validator component item を選択します。(1番目の Scenario Input と Scenario Output boxes<sup>1</sup> の間に追加されることに注意してください。)
- 2番目の Scenario Input を追加し、CAD Validator component に接続します。
  - 左手のメニューから Scenario I/O item を展開します。
  - Scenario Input item を選択します。(2番目の Scenario Input が追加されますが、 接続がないことに注意してください。)
  - CAD Validator component に2番目の Scenario Input を接続します。
    - 2番目の Scenario Input box の出力ポート (下部の灰色の点) をクリックします。(そのままマウスのボタンをホールドします。)
    - 出力ポートを CAD Validator component box の2番目の input port (右上の灰色の点) に「ドラッグ」します。(ドラッグし始めたときに現れる矢印に注意してください。)

- iii. CAD Validator component box の2番目の input port に、矢印を「ドロップ」します。  
(これにより2つのアイテムが接続されたことに注意してください。)
4. カスタマイズされたレポートを生成するために必要な CAD Validator パラメーターを設定します。
  - a. CAD Validator box を選択します。
  - b. 3D PDF Report | Script File Type for Report テンプレートのパラメーター (**3DPdfReportType**) を **Custom** に設定します。
  - c. Export Validation Result | 3D PDF Report パラメーター (**Create3DPdfReport**) を **1** に設定します。
  - d. 3D PDF Report | Script File for Customized Template パラメーターを (**Customized3DPdfReportScript**) に設定します。
    - i. External Reference (または Embed File) オプションを選択します。
    - ii. Formula スクリプトを選択します。(例: **C:\dps\examples\formulas\cv\_formula.rb**)
5. シナリオファイルを保存します。
  - a. 3DxSUITE シナリオフォルダを選択します。(例: **C:\Users\Public Documents\Elysium\3DxSUITE\Scenarios**)
  - b. シナリオに名前を付けます。(例: **cv\_scenario**)
  - c. シナリオを保存します。

1 Box(es)については、3DxSUITE ScenarioEditorの visual scenario flow editor の中央にあるボックスをご参照ください。

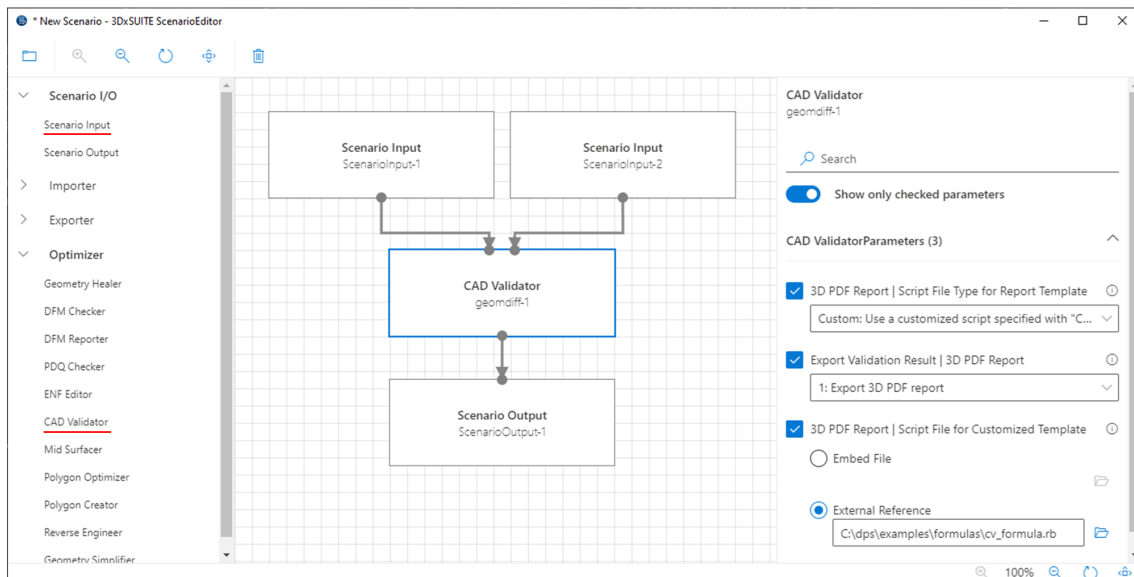


図 42. 3DxSUITE ScenarioEditor で作成されたシナリオの例



EX9.0 から、上記シナリオ内の CAD Validator (CV) コンポーネントと、CV および CV widgetを含むテンプレート / formula を編集するだけで CV レポートをカスタマイズすることができるようになりました。



CV input として使用する2つのENFファイルは、**inputfile1** (CV source) および **inputfile2** (CV target) と名付けられたデータソースを追加することによって

Data Package Studio にアクセスできるようになります。[ENF データソースの命名則](#) も参照してください。



CV コンポーネントで 3D PDF プロパティを設定する場合、**\$ENF23DPDF\_** プレフィックスを使用してカスタムパラメーターとして追加します。例えば、3D ビューの背景色を赤に変更する場合、**\$ENF23DPDF\_BackgroundColor=(255,0,0)** を追加します。

以上で、カスタマイズされた CAD Validator レポートを生成する準備が整いました。

## SLS による CAD Validator レポートの生成

最後の手順では、カスタマイズされた CAD Validator レポートを生成します。残っている作業は、比較する2つのモデルを選択すること、生成されたレポートの保存場所を選択することで、レポートを生成することができます。

### 手順:

1. 1番目のモデルで SLS を開きます。(例: `example_throttle_body_r00.enf` で右クリック、そして **Elysium > Run Scenario...** を選択します。)
2. シナリオを設定します。(例: **CAD Validator Customization Scenario (cv\_scenario.esa)**)
3. シナリオの入力ファイル数を **2** に設定します。
4. 2番目の入力ファイルを選択します。(例: `example_throttle_body_r01.enf`)
5. 出力フォルダーを指定します。(例: `C:\dps\output`)
6. カスタマイズされた CAD Validator レポートを生成するには、**[OK]** をクリックします。

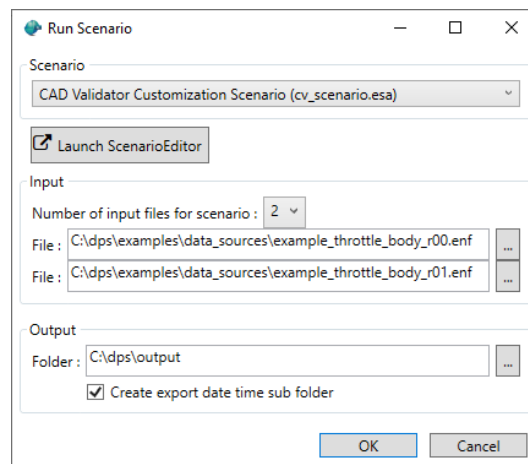


図 43. 3DxSUITE SmartLauncher によるシナリオ実行の例

## 6.2.3. Table および View Carousel のコンテンツのフィルタリング

本節では、Table widget と View Carousel widget でコンテンツをフィルタリングする方法について説明します。ここでは BoM Table widget を例として説明しますが、すべての Table と View Carousel で同じ方法を利用できます。

テーブルコンテンツフィルターの追加方法を説明する前に、まずはコンテンツフィルターの仕組みに

ついて説明します。コンテンツフィルターは、以下の 3 つの要素から構成されています。

1. Property Name - フィルタリングの対象となるプロパティを指定します。
2. Filter Type - コンテンツフィルターの種類です。フィルタリングの条件に一致した項目を含める (**Include**) か除外する (**Exclude**) かを指定します。
3. Filter - フィルタリングの条件となる単純な文字列または正規表現を指定します。

**Include** を指定した場合、フィルタリングの条件に一致しないすべての項目が除外されます。 **Exclude** を指定した場合、フィルタリングの条件に一致するすべての項目が除外されます。 フィルタリングの条件がすべて適用された後に残っている項目のみが、Table widget または View Carousel widget のコンテンツとして使用されます。

複数のコンテンツフィルターを設定した場合、フィルタリングの処理は以下のように実行されます。

1. 最初に、1 つ以上の **Exclude** を用いて定義された条件に一致するアイテムを除外します。
2. 次に、残りの項目から 1 つ以上の **Include** を用いて定義された条件に一致するものだけを保持します。

テーブルコンテンツフィルターの内部処理としては、フィルタリングを実行する際に正規表現を用いています。(例: `property =~ /filter/`) フィルタリングの条件を指定するために正規表現を理解している必要はなく、単純な文字列を指定するのみで十分です。ただし、単純な文字列を指定するとより多くの項目と一致する可能性があるため、正規表現を用いる場合より明示的な条件を追加する必要があります。

#### 例:

以下の表には、左側にフィルタリングされる対象となる BoM コンテンツ、右側に実現したい BoM table のコンテンツが示されています。この例では BoM コンテンツを Part Number プロパティのみでフィルタリングすることとするため、表には Part Number のみを示しています。

表 8. 元の BoM コンテンツ(左)と実現したい BoM table コンテンツ(右)

| Original           | Wanted             |
|--------------------|--------------------|
| TB-Assembly        | TB-Assembly        |
| TB-Pin-Flapper     | -                  |
| TB-Pin-Flapper-Top | TB-Pin-Flapper-Top |
| TB-Pin-Gear        | TB-Pin-Gear        |
| TBM-Body           | -                  |

まずはフィルタリングの条件として、単純な文字列 **Flapper** を適用します。この条件は **Flapper** を含むすべての項目に一致します。つまり、すべての **Flapper** を含む項目は Filter Type に指定されている内容に応じてフィルタリングされることになります。しかし、この条件は含めたい項目と除外したい項目の両方に一致するため、この条件を使用して目的のコンテンツを実現することはできません。

以下の文字列を用いた 3 つの **Include** 条件を追加することで、目的のコンテンツを実現することができます。

- **Assembly**
- **Pin-Flapper-top**
- **Gear**

しかしこの条件は非常に明示的であり、モデルが変更されることに対して堅牢性がありません。

正規表現を使用すると、**Flapper** で終わるすべての項目を除外できます。それだけでなく、**TB-** で始まり、途中に 0 文字以上の文字があり、**Flapper** で終わるというようにさらに細かく指定することも可能です。具体的には、**Exclude** 条件の Filter に正規表現で **^TB-.\*-Flapper\$** を指定します。**^** はテキストの先頭に一致します。**.** は改行以外のすべての文字に一致します。

**\*** は直前の文字 0 個以上に一致します。(ここでは任意の文字への一致を意味します)

**\$** はテキストの末尾に一致します。



インターネット上には、正規表現を作成および調整する際に役立つオンラインツールなど、正規表現を扱う上で有用な様々なリソースが公開されています。

以下の手順では、上で説明したフィルタリングの条件に加えて **TB-** を含む項目に関する フィルタリングの条件を追加します。

#### 手順:

1. Data Package Studio に Part Name、Number、および Quantity 列を含む BoM Table を追加します。(詳細は [BoM Table Widgetの追加](#) を参照してください)
2. Table Content Filter セクションにある [+ Add Filter] をクリックします。
  - a. フィルタリングしたいプロパティを指定します。(例: **Part Number**)
  - b. コンテンツフィルターの種類を指定します。例えば、一致する項目を除外するには **Exclude** を指定します。
  - c. フィルタリングの条件を指定します。例えば、**^TB-.\*-Flapper\$** のように **TB-** で始まり、**-Flapper** で終わる項目を一致させます。
3. **Part Number** の値に **TB-** を含む項目を含めるようにするコンテンツフィルターをもう一つ追加します。



BoM Table Columns

| Property Name*    | Header Name in TDP* | Info Popup                          | Width (%) |
|-------------------|---------------------|-------------------------------------|-----------|
| P Part Name ▼     | Part Name           | <input type="checkbox"/>            | *         |
| P Part Number ▼   | Part Number         | <input checked="" type="checkbox"/> | 35        |
| P Part Quantity ▼ | QTY                 | <input type="checkbox"/>            | 15        |

Table Content Filter

| Property Name*  | Filter Type* | Filter*          |
|-----------------|--------------|------------------|
| P Part Number ▼ | Exclude ▼    | ^TB-.*-Flapper\$ |
| P Part Number ▼ | Include ▼    | TB-              |

図 44. 上記の手順で作成したコンテンツフィルターの例

作成したコンテンツフィルターを最初の例と同じモデルに適用すると、下の表の右側にある BoM コンテンツが表示されます。"TB-Pin-Flapper" は **Exclude** フィルターに一致するため除外されます。"TBM-Body" は **Include** フィルターに一致しないため除外されます。

表 9. 上記の手順で作成したコンテンツフィルターの適用例

| Original           | Exclude →        | Output             | Include → | Final Output       |
|--------------------|------------------|--------------------|-----------|--------------------|
| TB-Assembly        | ^Tb-.*-Flapper\$ | TB-Assembly        | ^Tb-      | TB-Assembly        |
| TB-Pin-Flapper     |                  | -                  |           | -                  |
| TB-Pin-Flapper-Top |                  | TB-Pin-Flapper-Top |           | TB-Pin-Flapper-Top |
| TB-Pin-Gear        |                  | TB-Pin-Gear        |           | TB-Pin-Gear        |
| TBM-Body           |                  | TBM-Body           |           | -                  |

## 6.2.4. Bill of Characteristics Tableの追加

本節では、**PMI Table widget** に表示することのできる Bill of Characteristics (BoC) データを準備するため、3Dモデルを前処理する手順について説明します。

3Dモデルから BoC データを自動抽出し、それを TDP に含めた上で検査部門に送ることにより、設計部門と検査部門間のコミュニケーションを円滑化するというユースケースが考えられます。

ここでは、以下の内容について説明します：

- 前処理スクリプトの一般的な構造
- 独自のBoC前処理スクリプトを作成する方法
- TDPを作成する際、前処理スクリプトを実行するように3DxSUITEを構成する方法



## サンプルファイル:

| ファイル                               | 内容                               |
|------------------------------------|----------------------------------|
| "scripts\pre_process_template.rb"  | 前処理スクリプトのテンプレート                  |
| "scripts\pre_process_boc.rb"       | BoC 前処理スクリプトの例                   |
| "scenarios\dps_scenario.esa"       | BoC Formula パラメーターセットを含むシナリオファイル |
| "data_sources\TBM-MBD-R00_BoC.enf" | BoC balloon を使用した 3D モデル         |

## 前処理スクリプトの構造

前処理スクリプトの構造はシンプルです。このスクリプトは Ruby で記述されており、3DxSUITE が前処理スクリプトを識別して実行するために必要な 3 種類の要素が含まれています。(以下のテンプレートスクリプトを参照)

### 1. TDP 前処理スクリプトのテンプレート

```
# -*- encoding: UTF-8 -*-
module PdfEditor                                ①
  class CustomizedEnfPreProcess                 ②
    logger.info "Custom File (#{__FILE__}): "

    def on_run_before(ee_session)               ③
    end

  end
end
```

① **module PdfEditor** は、3DxSUITE 内のどこにスクリプトが存在するかを定義します。

② **class CustomizedEnfPreProcess** は、PdfEditor モジュールに含まれており、前処理スクリプトを実装するクラスを定義します。

③ **def on\_run\_before(ee\_session)** は、ENF Editor Session を入力とし受け取る CustomizedEnfPreProcess クラスのメソッドを定義します。前処理ロジックもここに実装する必要があります。

プログラミングやRubyに慣れていない場合でもご安心ください。以下の 2 点だけを覚えておけば実装することができます。

1. 前処理スクリプトは、上記の構造と名前に従う必要があります。
2. 前処理ロジックは**on\_run\_before(ee\_session)**で実行する必要があります。

これらの手順に従えば 3DxSUITE は前処理スクリプトを識別して実行することができます。

この方法で、ほぼすべての種類の前処理を実行することができます。例えば、外部JSONファイルによって提供される属性情報をモデルにプロパティとして追加したり、次の節で説明するように PMI Table widget のための BoC データを準備したりすることができます。

## カスタマイズしたBoC前処理スクリプトの作成

以上で、PMI Table widget に表示される BoC データを準備するために必要となる、カスタマイズされた前処理スクリプトを作成する準備が整いました。

スクリプトは次の3つのステップでモデルを処理します。

1. BoC の balloon に関連する製品製造情報 (PMI) を識別します。
2. BoC の balloon を表す注釈を識別します。
3. BoC の balloon を関連するそれぞれの PMI 項目に接続します。

3つの手順はすべてモデル内の注釈をループする必要があります。まず ENF Editor API を例として、どのように行うかを説明します。



元モデルは前処理を実施する前に [ENF](#) に変換されるため、ENF Editor API を使用してモデルデータを処理することができます。

Listing 1. モデル内のすべての注釈をループする方法

```
# ...

def on_run_before(ee_session)
  ee_session.model.components.each do |compo|      ①
    compo.annotations.each do |anno|                ②
      # ...
    end
  end
end
# ...

end
end

# ...
```

- ① ENF Editor Session オブジェクト (ee\_session) を使って、モデル内の各コンポーネントをループします。
- ② コンポーネントに含まれる注釈それぞれをループします。

次に BoC の balloon に関連付けられている注釈 (PMI 項目) を識別します。サンプルモデルでは、PMI 項目名に `1_EREf(2;5)` のようなテキストが含まれます。ここで、`EREf` は単語を識別するキーになります。括弧内の整数 `2;5` は、それぞれ `2` と `5` という識別子を持つ balloon への参照です。



PMI 項目名で複数の数字を指定する場合、区切り文字にコンマ (,) を使用しないでください。コンマを使用した場合、BoC の balloon やそれに関連する PMI のハイライトが正しく動作しません。

この内容を踏まえ、BoC の balloon に関連付けられた PMI 項目を識別する方法を説明します。

Listing 2. BoC の balloon に関連する PMI 項目を識別する

```
# ...
ee_session.model.components.each do |compo|
  # Create mapping from BoC ID to referenced PMI
  id_to_ref = Hash.new
  compo.annotations.each do |anno|
    md = anno.name.match(/EREF\(([^\d;]+)\)/) ①
    next if md.nil? || md.size != 2 ②
    md[1].split(";").each do |id| ③
      id_to_ref[id] = anno
    end
  end
end
# ...
end
# ...
```

- ① 正規表現を用いて EREF という名前パターンで BoC 関連の PMI 項目を抽出します。
- ② 正規表現が一致しない場合は、スキップして次の注釈に移動します。
- ③ それぞれの参照識別子を PMI 項目にマップします。(後でそれぞれの balloon に接続するため、Ruby ハッシュに格納されます。)

以上で BoC 識別子を PMI 項目を表す ENF オブジェクトにマップする Ruby ハッシュ ("id\_to\_ref") ができました。次に、BoC の balloon にフラグを立て、ハッシュを使用してそれぞれの PMI 項目に接続します。ただし、最初に BoC balloon を識別する必要があります。

サンプルモデルに含まれる BoC の balloon の項目名は、9\_EBOC2 のようなテキストを含み、EBOC は、注釈を BoC の balloon 項目として識別するキーワードです。最後の整数 2 は、PMI 項目（上記参照）を対応する balloon に関連付けるために使用する一意の識別子です。

以上を踏まえ、以下では実際の使用方法を説明します。

Listing 3. BoC の balloon を識別する

```
# ...
# Create mapping from BoC ID to referenced PMI
# ...

# Flag BoC and set referenced PMI
compo.annotations.each do |anno|
  md = anno.name.match(/EBOC(\d+)/) ①
  next if md.nil? || md.size != 2 ②
  # ...
end
# ...
```

- ① 正規表現を用いて **EBOC** という名前パターンで BoC の balloon を抽出します。
- ② 正規表現が一致しない場合は、スキップして次の注釈に移動します。

以上の結果として、以下の様な状態になります。

- BoC 識別子を、関連する PMI 項目の ENF オブジェクトにマッピングする Ruby ハッシュが 出来 上がりました。
- BoC の balloon 項目とその BoC 識別子 (**anno** オブジェクトと **md** 配列) を識別しました。

これらにより、識別された注釈に BoC の balloon であることを示すフラグを設定し、それに 関連付 けられた PMI 項目に接続することができます。

Listing 4. BoC の balloon にフラグを設定して関連する BoC 項目に接続する

```
# ...
# Create mapping from BoC ID to referenced PMI
# ...

# Flag BoC and set referenced PMI
compo.annotations.each do |anno|
  md = anno.name.match(/EBOC(\d+)/)
  next if md.nil? || md.size != 2
  ref_anno = id_to_ref[md[1]]      ①
  next if ref_anno.nil?          ②
  anno.boc_tag = true             ③
  anno.boc_ref = ref_anno        ④
end
# ...
```

- ① BoC の balloon に関連付けられた PMI 項目を取得します。
- ② PMI 項目に関連付けられていない場合、スキップして次の BoC の balloon に移動します。
- ③ 注釈に BoC の balloon としてタグを付けます。
- ④ BoC の balloon を、それに関連する PMI 項目に接続します。

完全な前処理スクリプトのサンプルについては、[TDP 前処理スクリプトの例](#) を参照してください。

## BoC 前処理スクリプトを実行するように 3DxSUITE を構成する

BoC の前処理スクリプトやその他の前処理スクリプトを使用する際には、最後にもう 1 つ作業を 実施 する必要があります。TDP を作成するときに、スクリプトを実行するように 3DxSUITE を構成して ください。

手順は次の通りです。

'CustomizedEnfPreProcessScript' パラメーターを使用して、前処理スクリプトがどこに あるかを指 定します。3DxSUITE SmartLauncher を使用している場合は、以下の図のように ユーザインターフ ェースを介してこのパラメーターを 設定します。

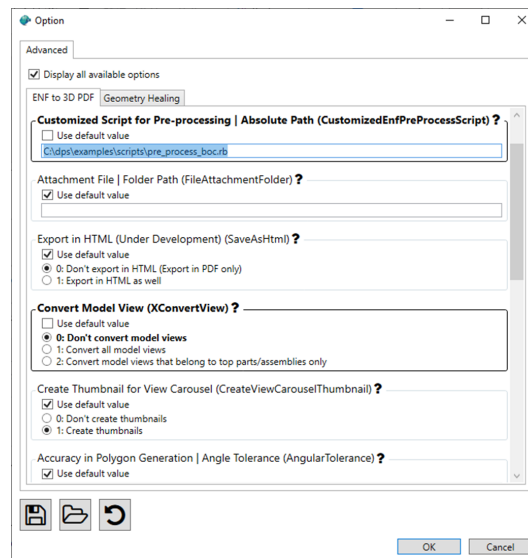


図 45. 3DxSUITE SmartLauncher で 'CustomizedEnfPreProcessScript' パラメーターを設定



シナリオを使用する場合は、3DxSUITE ScenarioEditor でシナリオ作成/編集する際に 'CustomizedEnfPreProcessScript' パラメーターを設定 してください。

TDP を作成する際に BoC 前処理スクリプトを実行できるようになりました。この機能を試する 場合には、`dps_scenario.esa` とそのパラメーターセットである `BoC Formula`、およびサンプル モデル `TBM-MBD-R00_BoC.enf` を使ってください。シナリオパラメーター 'CustomizedEnfPreProcessScript' を変更して独自の BoC 前処理スクリプトを指すようにすることもできますし、当該パラメーターを 変更せずサンプルスクリプトをそのまま使用することもできます。

## 6.2.5. エリジオン TDP JavaScript API を用いたテーブルデータの保存

このドキュメントは、エリジオンの TDP JavaScript API (TDP JS API) を使用してテーブルデータを 保存するボタンを作成する手順について説明します。テーブル用の JS API の詳細については [クラス: ElyTdpTableWidget](#) を参照してください。

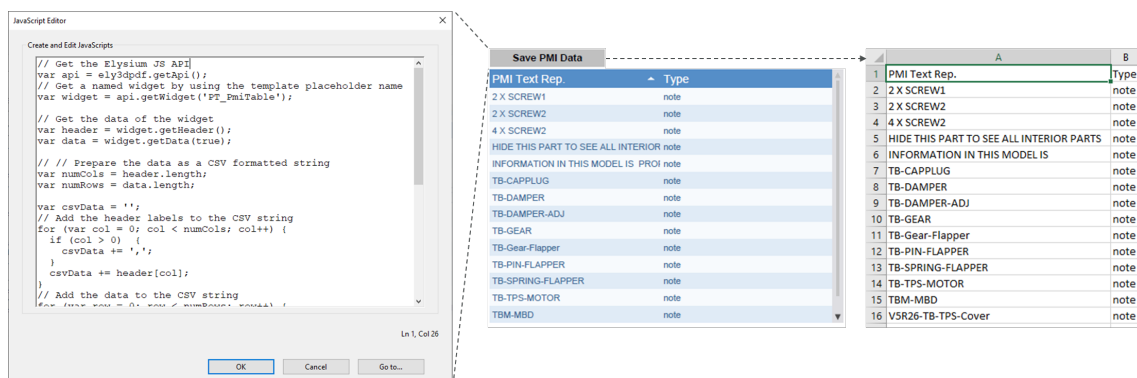


図 46. TDP JS API でテーブルデータを保存する例

本手順の前に、[エリジオン TDP JavaScript API を用いた 3D ツールバーの追加](#) ガイドを完了しておく ことをお勧めします。最初の手順について詳しく説明してあります。

**サンプルファイル:**

| ファイル   | 内容                                      |
|--|---|
| "formulas\basic_how_to_guide_template.pdf/json/rb" | Formula                                 |
| "scenarios\dps_scenario.esa"                       | How-To Guide Formula パラメーター<br>セットのシナリオ |
| "data_sources\example_throttle_body_r01.enf"       | 入力モデル                                   |

**手順:**

1. ボタンを追加して名前を付けます。(例: btn\_save\_pmi)
2. ボタンをダブルクリックしてボタンのプロパティを開きます。
3. [Options] タブで以下の通り入力します。
  - a. レイアウトとして "Label only" を選択します。
  - b. ラベルとして "Save PMI Data" を入力します。
4. [Actions] タブでは以下の通りに入力します。
  - a. Trigger として "Mouse Up" を選択します。
  - b. Action として "Run a JavaScript" を選択します。
  - c. [Add...] ボタンをクリックします。
5. 開いた JavaScript エディタでは以下の作業を実施します。
  - a. PMI の情報を保存するためのコードを追加します。後述のコードを参照してください。
  - b. コードを保存して、エディタを閉じます。(組み込みのエディタを使用する場合、[OK] をクリックします)
6. ボタンのプロパティウインドウで [Close] をクリックします。

Listing 5. テーブルデータを CSV ファイルとして保存するコードの例

```

var api = ely3dpdf.getApi();
var widget = api.getWidget('PT_PmiTable');

var header = widget.getHeader(); ①
var data = widget.getData(true); ②

var numCols = header.length;
var numRows = data.length;

var csvDelimiter = ','; ③
var csvData = '';
for (var col = 0; col < numCols; col++) { ④
    if (col > 0) {

```

```

        csvData += csvDelimiter;
    }
    csvData += header[col];
}
for (var row = 0; row < numRows; row++) { ⑤
    csvData += '\n';
    for (var col = 0; col < numCols; col++) {
        if (col > 0) {
            csvData += csvDelimiter;
        }
        var cellData = data[row][col];
        if (cellData != '') {
            cellData = '"' + cellData + '"'; ⑥
        }
        csvData += cellData;
    }
}
try {
    var dataObjectName = 'PmiCsvFile.csv';
    createDataObject(dataObjectName, csvData); ⑦
    exportDataObject(dataObjectName); ⑧
    removeDataObject(dataObjectName); ⑨
} catch (err) {
    app.alert({
        cTitle: 'Save CSV File Error',
        cMsg: 'Saving requires Adobe Acrobat Pro.'
    });
}

```

- ① テーブルのヘッダーラベル (文字列の 配列) を取得します。
- ② フィルタリングしたテーブルデータを取得します (文字列の 配列 が格納された 配列)。フィルタしていないデータを取得したい場合には、入力値を指定しないか **false** を指定します。
- ③ CSV の区切り文字を指定します。ここではコンマ (,) を指定します。
- ④ csvData にヘッダーラベルを追加します。
- ⑤ テーブルデータを csvData に追加します。
- ⑥ 選択した CSV の区切り文字がテーブルデータに含まれている場合は、セルデータの前後にダブルクォーテーションを付けないと CSV 構造が壊れます。
- ⑦ CSV ファイルを作成し、TDP に添付します
- ⑧ ファイルを保存するためのプロンプトを開いて、添付された CSV ファイルを保存します。
- ⑨ TDP から添付された CSV ファイルを削除します。



手順 7～9 では、Adobe JS API を使用して CSV ファイルを保存します。この際には Adobe Acrobat Pro<sup>1</sup> が必要となりますが、この条件は今後変更となる可能性があります。

1: 手順 7 で `createDataObject` 関数を呼び出すと、Adobe Acrobat Reader により JS エラーが発生します。

## 6.2.6. 作業指示 TDP の作成

本節では、作業指示 TDP の作成方法について説明します。最初に **作業指示** の意味を説明し、続いて **Data Package Studio** でどのように **作業指示** を作成するかについて説明します。最後に、**サンプルの作業指示用 formula** を作成する手順を段階的に説明します。

**サンプルファイル:**

| ファイル  | 内容   |
|---|--|
| "scenarios\dps_scenario.esa"                        | Work Instructions Formula パラメーターセットを含むシナリオファイル |
| "data_sources\work_instructions\TB-Assembly-WI.enf" | 入力モデル  |
| "data_sources\work_instructions\wi_steps.csv"       | 作業指示の手順の情報                                     |
| "data_sources\work_instructions\wi_parts.csv"       | 手順に関連する部品の情報                                   |
| "data_sources\work_instructions\wi_tools.csv"       | 手順に関連するツールの情報                                  |
| "data_sources\work_instructions\wi_safety.csv"      | 手順に関連する安全指示の情報                                 |

## 定義

作業指示とは、ある作業の実行方法を詳細に記述したガイドもしくは標準的な操作手順を意味しています。求められる仕様に正確に適合するように、製品を製造し、組み立て、梱包し、出荷するのに役立ちます。

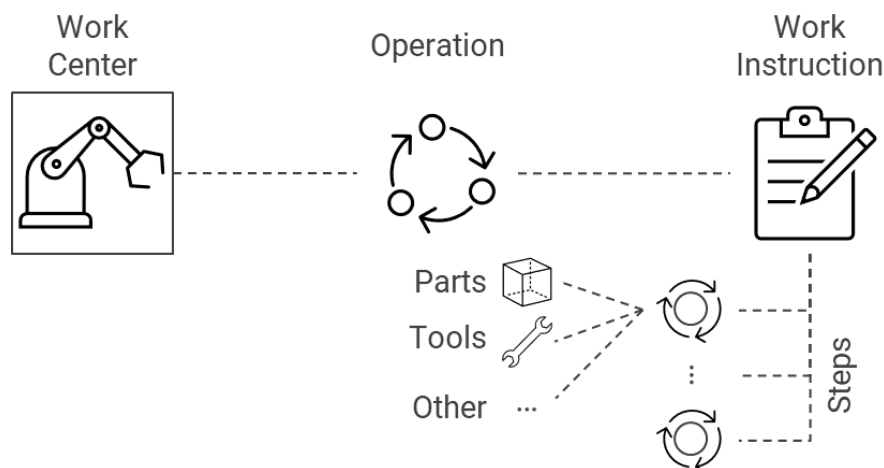


図 47. 作業指示が製造に関連する他の要素とどのように関連するかの例



## 概念

作業指示を作成できるようにするにあたり、弊社では専用のデータソース、widget、Trigger/Actionを用意するのではなく、一般的なデータソース、widget、Trigger/Action を用いて作業指示を実現できるようにしました。これにより、作業指示 TDP をユーザー独自のニーズに適合させることができます。

**サンプル** では、**図 48** のような基本的な作業指示 TDP を作成します。作業指示の手順が手順リスト (右上) で選択されている場合、その手順は 3D ビュー (左上) で視覚化され、その手順の詳細は手順リストの下に表示されます。手順に関連する部品、ツール、および安全指示は、手順の詳細の下にあるそれぞれのリストに表示されます。

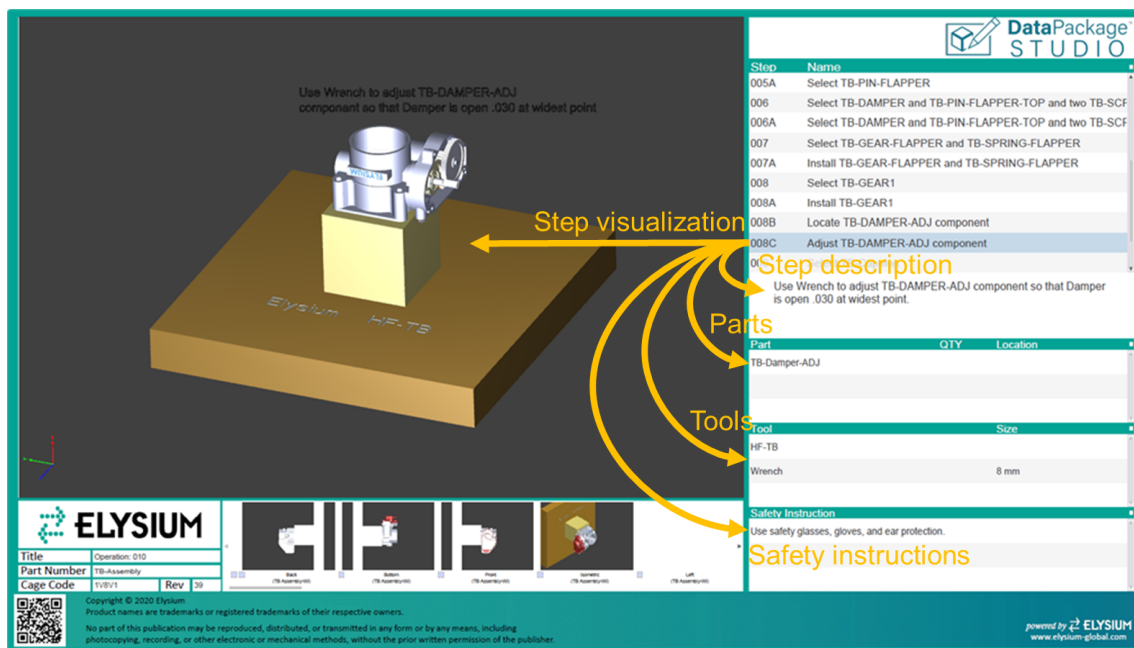


図 48. 作業指示 TDP の例

図 48 の作業指示 TDP のデータは、図 49 で示すように CSV データソースによって提供されます。widget を接続するために必要なデータさえ含まれていれば、作業指示を表現したり説明したりするために必要となるデータはどんなものでも使用することができます。

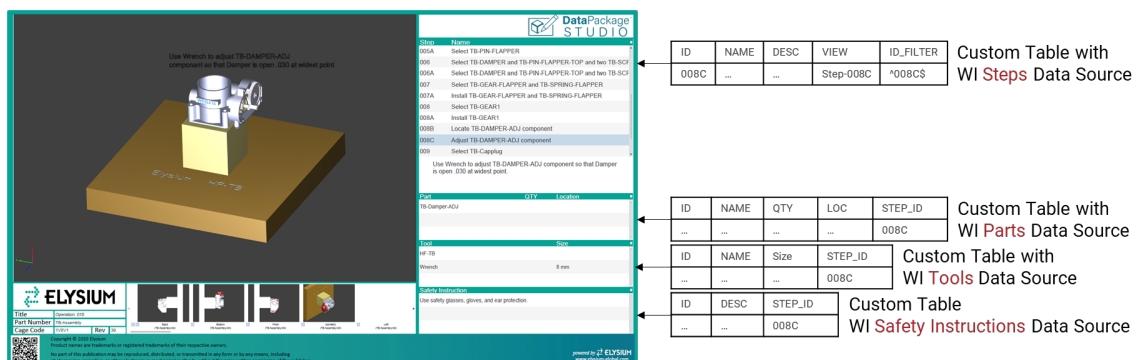


図 49. CSV データソースとして提供される作業指示のデータ

最初に、手順リストから手順を選択した際、選択した手順に応じて対応するビューを 3D View widget に適用するために必要なデータについて説明します。ビューを適用する上で必要な情報はビューの完全名です。これはビューの名前と、ビューが関連付けられているパート名を括弧で囲んで連結したも

のになります。(例: **Step-008C (TB Assembly-WI)**) 作業指示の手順のデータソース (**wi\_steps.csv**) の **STEP\_VIEW** 列を参照してください。

**サンプル** で説明する 3D View widget と手順リスト (Custom Table widget) の間に作成する Trigger/Action は、選択した項目の **STEP\_VIEW** の値に基づいてビューを適用します。

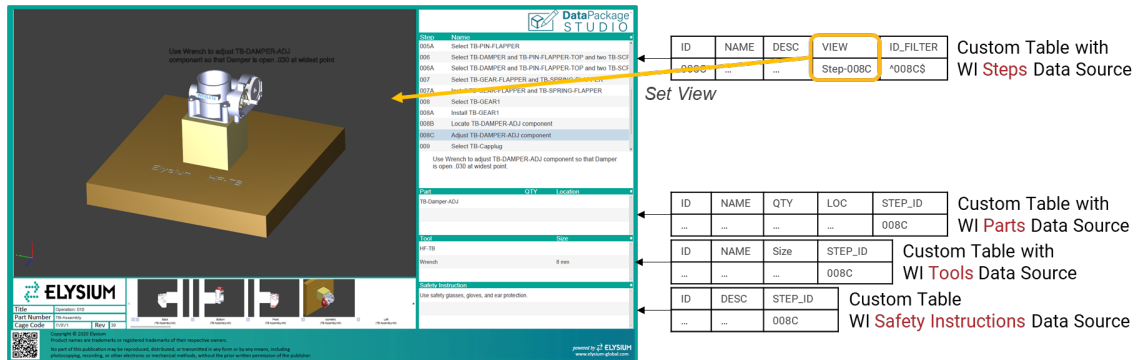


図 50. 手順のデータソースにはビューを適用するのに必要なビューの名前が含まれます

次に手順の詳細について説明します。これは手順リストから手順を選択した際に Text Field widget に表示される内容になります。作業指示の手順のデータソースにある **STEP\_DESC** 列に各手順の詳細が含まれています。

**サンプル** で説明する Text Field widget と手順リスト (Custom Table widget) の間に作成する Trigger/Action は、選択した項目の **STEP\_DESC** の値を表示します。

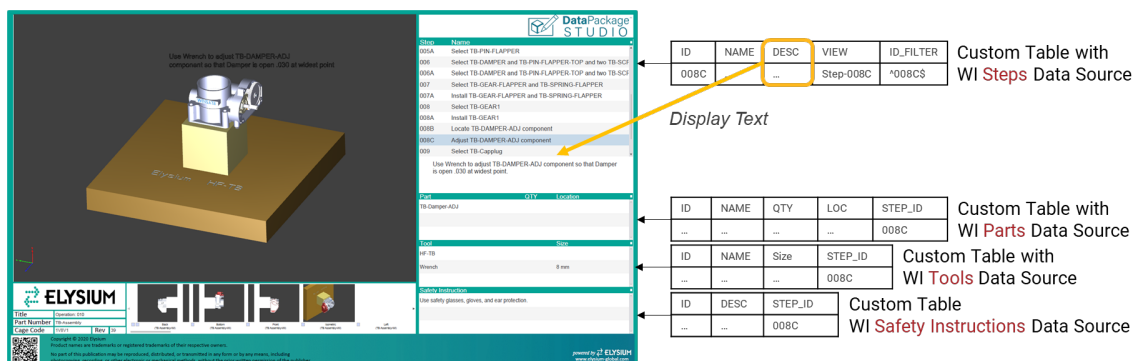


図 51. 手順のデータソースには手順の詳細が含まれます

最後に、手順リストから手順を選択した際に、関連付けられている部品、ツール、および安全指示がどのように表示されるかを説明します。これは、手順が選択されたときに、部品、ツール、および安全指示のテーブルをフィルタリングすることで実現しています。

作業指示の手順のデータソースにおいて、適用するフィルターが ID を用いて指定されているとします。(例: **^008C\$**) 部品、ツール、および安全指示のデータソースに含まれる各項目は、関連付けられている手順の ID を含んでいます。(例: **008C**)

正規表現を用いて条件に一致する ID を指定します。例えば、テキストの先頭 (^) + ID + テキストの末尾 (008C\$) といったように指定します。作業指示の手順のデータソース (**wi\_steps.csv**) にある **STEP\_ID\_FILTER** 列や他の CSV データソースの **STEP\_ID** 列を参照してください。

**サンプル** で説明する部品、ツール、および安全指示リスト (Custom Table widgets) と手順のリスト

(Custom Table widget) の間に作成する Trigger/Action は、STEP\_ID\_FILTER の値と、接続されたテーブルの STEP\_ID 列を用いてフィルタリングを行います。

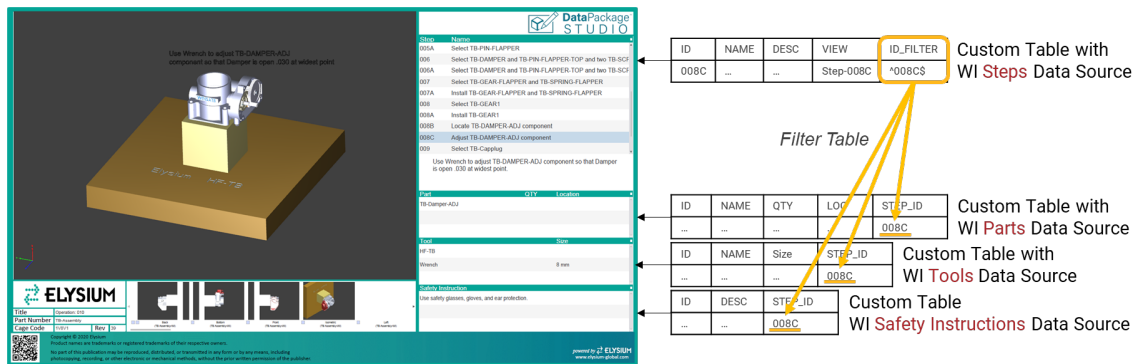


図 52. 手順のデータソースには、部品、ツール、および安全指示データソースに含まれる項目を関連付けるためのフィルターが含まれています

ここまでは Data Package Studio を用いて実現する作業指示の概要を説明してきました。次は目的に沿った CSV データソースを準備します。前述したように、必要な情報さえ含まれていれば、目的に合わせてどのようなデータもデータソースに含めることができます。

図 53 で示すように、場合によっては MES やその他のシステムから必要な情報を抽出することが必要になる場合があります。作業指示を作成するためのデータを抽出する手順 (グレー表示) については 3DxSUITE 製品の対象外になりますので注意してください。

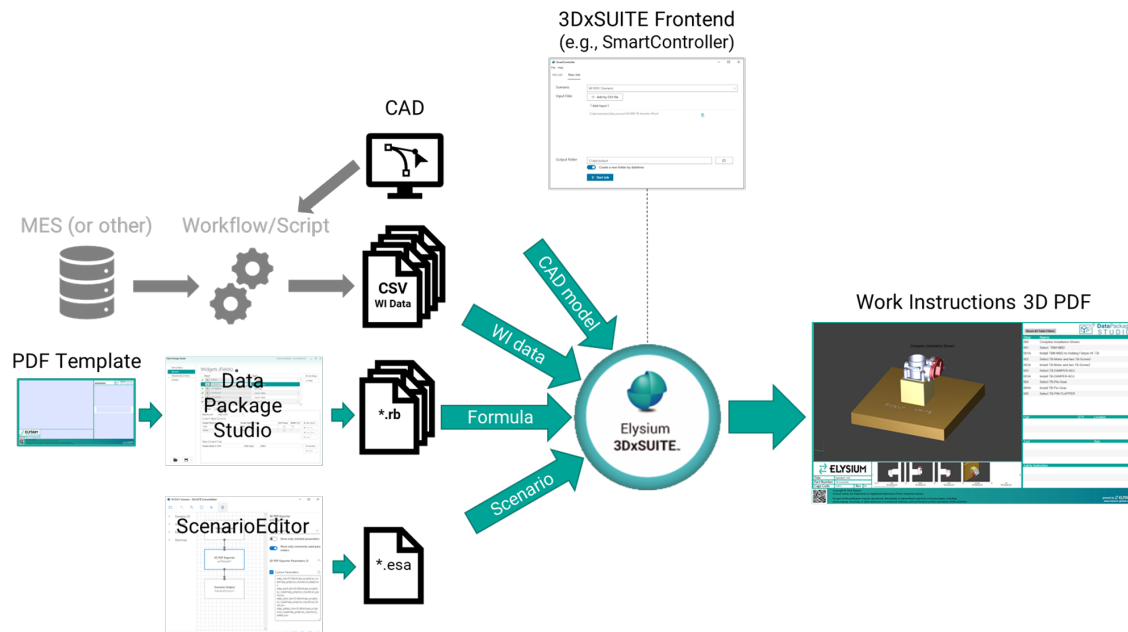


図 53. 作業指示 TDP 作成ワークフローの例

## サンプル

この節では、図 48 のような基本的な作業指示 TDP を作成します。

### 手順:

1. Data Package Studio で **wi\_formula.pdf** を開きます。

## 2. 必要なデータソース (図 54 を参照) を作成します:

- a. **inputfile** という名称の ENF データソース
- b. **step\_list (UTF-8 encoded)** という名称の CSV データソース
- c. **step\_part\_list (UTF-8 encoded)** という名称の CSV データソース
- d. **step\_tool\_list (UTF-8 encoded)** という名称の CSV データソース
- e. **step\_safety\_list (UTF-8 encoded)** という名称の CSV データソース

Data Sources ?

| ID*              | Type* | Encoding*       | Attachment                          | Description |
|------------------|-------|-----------------|-------------------------------------|-------------|
| inputfile        | ENF   | -               | <input checked="" type="checkbox"/> |             |
| step_list        | CSV   | Unicode (UTF-8) | <input type="checkbox"/>            |             |
| step_part_list   | CSV   | Unicode (UTF-8) | <input type="checkbox"/>            |             |
| step_tool_list   | CSV   | Unicode (UTF-8) | <input type="checkbox"/>            |             |
| step_safety_list | CSV   | Unicode (UTF-8) | <input type="checkbox"/>            |             |

図 54. 作業指示 TDP のデータソース

## 3. widget を追加:

- a. **CT\_StepList** のプレースホルダーとして Custom Table widget を追加します。
  - i. データソースとして **step\_list (CSV)** を指定します。
  - ii. 次の内容を追加します。 CSV 側のヘッダー名: **STEP\_ID**, TDP 側のヘッダー名: **Step**, 幅: 15
  - iii. 次の内容を追加します。 CSV 側のヘッダー名: **STEP\_NAME**, TDP 側のヘッダー名: **Name**, 幅: \*

Data Source\* step\_list (CSV)

☐ Show Empty Table (at startup and unfiltered)

Custom Table Columns

| Header Name in CSV* | Header Name in TDP* | Info Popup               | Width (%) |
|---------------------|---------------------|--------------------------|-----------|
| STEP_ID             | Step                | <input type="checkbox"/> | 15        |
| STEP_NAME           | Name                | <input type="checkbox"/> | *         |

図 55. 手順リストのデータソースと列 (Custom Table widget)

- b. **TF\_StepDesc** のプレースホルダーとして Text Field widget を追加します。
  - i. データソースとして **By Trigger/Action** を指定します。(選択した手順からデータを取得するため)
- c. **CT\_PartsList** のプレースホルダーとして Custom Table widget を追加します。
  - i. データソースとして **step\_part\_list (CSV)** を指定します。
  - ii. 手順が選択されるまでテーブル内の項目を非表示にしておくため、"Show Empty Table" チェックボックスをオンにします。
  - iii. 次の内容を追加します。 CSV 側のヘッダー名: **PART\_NAME**, TDP 側のヘッダー名: **Part**, 幅: 50
  - iv. 次の内容を追加します。 CSV 側のヘッダー名: **PART\_QTY**, TDP 側のヘッダー名: **QTY**, 幅: \*
  - v. 次の内容を追加します。 CSV 側のヘッダー名: **PART\_LOC**, TDP 側のヘッダー名: **Location**, 幅: 35

Data Source\*

☒ Show Empty Table (at startup and unfiltered)

Custom Table Columns

| Header Name in CSV* | Header Name in TDP* | Info Popup                          | Width (%) |
|---------------------|---------------------|-------------------------------------|-----------|
| PART_NAME           | Part                | <input checked="" type="checkbox"/> | 50        |
| PART_QTY            | QTY                 | <input type="checkbox"/>            | *         |
| PART_LOC            | Location            | <input checked="" type="checkbox"/> | 35        |

図 56. 部品リストのデータソースと列 (Custom Table widget)

- d. **CT\_ToolsList** のプレースホルダーとして Custom Table widget を追加します。
- データソースとして **step\_tool\_list (CSV)** を指定します。
  - 手順が選択されるまでテーブル内の項目を非表示にしておくため、"Show Empty Table" チェックボックスをオンにします。
  - 次の内容を追加します。CSV 側のヘッダー名: **TOOL\_NAME**, TDP 側のヘッダー名: **Tool**, 幅: \*
  - 次の内容を追加します。CSV 側のヘッダー名: **TOOL=SIZE**, TDP 側のヘッダー名: **Size**, 幅: 35
- e. **CT\_SafetyList** のプレースホルダーとして Custom Table widget を追加します。
- データソースとして **step\_safety\_list (CSV)** を指定します。
  - 手順が選択されるまでテーブル内の項目を非表示にしておくため、"Show Empty Table" チェックボックスをオンにします。
  - 次の内容を追加します。CSV 側のヘッダー名: **TOOL\_NAME**, TDP 側のヘッダー名: **Tool**, 幅: \*
  - 次の内容を追加します。CSV 側のヘッダー名: **TOOL=SIZE**, TDP側のヘッダー名: **Size**, 幅: 35
- f. **VC\_Views** のプレースホルダーとして View Carousel widget を追加します。
- データソースとして **inputfile (ENF)** を指定します。
  - 正規表現を用いた **Exclude** フィルター (**^Step-\d{1,3}[A-Z]?\$**) を追加して、全ての手順のビューを除外します。(詳細は [Table](#) および [View Carousel のコンテンツのフィルタリング](#) を参照してください。)

Data Source\*

Table Content Filter


| Property Name*   | Filter Type* | Filter*               |
|--|--------------|-----------------------|
|  Name | Exclude      | ^Step-\d{1,3}[A-Z]?\$ |

図 57. View Carousel の手順ビューを除外するためのデータソースとコンテンツフィルター

- g. **3DView** のプレースホルダーとして 3D View widget を追加し、データソースとして **inputfile (ENF)** を指定します。
- h. **TF\_Title** のプレースホルダーとして Text Field widget を追加します。
- データソースとして **inputfile (ENF)** を指定します。
  - プロパティとして **Enter property name...** を指定します。
  - プロパティ名として **Operation** を指定します。
- i. **TF\_PartNumber** のプレースホルダーとして Text Field widget を追加します。

- i. データソースとして **inputfile (ENF)** を指定します。
    - ii. プロパティとして **Enter property name...** を指定します。
    - iii. プロパティ名として **PartNumber** を指定します。
  - j. **TF\_CageCode** のプレースホルダーとして Text Field widget を追加します。
    - i. データソースとして **inputfile (ENF)** を指定します。
    - ii. プロパティとして **Enter property name...** を指定します。
    - iii. プロパティ名として **CageCode** を指定します。
  - k. **TF\_Revision** のプレースホルダーとして Text Field widget を追加します。
    - i. データソースとして **inputfile (ENF)** を指定します。
    - ii. プロパティとして **Part Version** を指定します。
4. Trigger/Action を追加:
- a. Trigger/Action を追加して、3D ビュー上で手順を確認できるようにします。
    - i. Trigger Widget として **CT\_StepList (Custom Table)** を選択します。
    - ii. Target Widget として **3DView (3D View)** を選択します。
    - iii. Trigger Property に **STEP\_VIEW** を指定します。

|                   |                              |
|-------------------|------------------------------|
| Trigger Widget*   | CT_StepList (Custom Table) ▼ |
| Target Widget*    | 3DView (3D View) ▼           |
| Trigger Property* | STEP_VIEW                    |

図 58. 手順リストを 3D View に接続

- b. Trigger/Action を追加して、手順の説明を表示します。
  - i. Trigger Widget として **CT\_StepList (Custom Table)** を選択します。
  - ii. Target Widget として **TF\_StepDesc (Text Field)** を選択します。
  - iii. Trigger Property に **STEP\_DESC** を指定します。

|                   |                              |
|-------------------|------------------------------|
| Trigger Widget*   | CT_StepList (Custom Table) ▼ |
| Target Widget*    | TF_StepDesc (Text Field) ▼   |
| Trigger Property* | STEP_DESC                    |

図 59. 手順リストを Text Field の説明に接続

- c. Trigger/Action を追加して、手順に関連付けられた Part を表示します。
  - i. Trigger Widget として **CT\_StepList (Custom Table)** を選択します。
  - ii. Target Widget として **CT\_PartsList (Custom Table)** を選択します。
  - iii. Trigger Property に **STEP\_ID\_FILTER** を指定します。
  - iv. Target Property に **STEP\_ID** を指定します。



|                   |                               |
|-------------------|-------------------------------|
| Trigger Widget*   | CT_StepList (Custom Table) ▼  |
| Target Widget*    | CT_PartsList (Custom Table) ▼ |
| Trigger Property* | STEP_ID_FILTER                |
| Target Property*  | STEP_ID                       |

図 60. 手順リストを部品リストに接続

d. Trigger/Action を追加して、手順に関連付けられた Tool を表示します。

- i. Trigger Widget として CT\_StepList (Custom Table) を選択します。
- ii. Target Widget として CT\_ToolsList (Custom Table) を選択します。
- iii. Trigger Property に STEP\_ID\_FILTER を指定します。
- iv. Target Property に STEP\_ID を指定します。

|                   |                               |
|-------------------|-------------------------------|
| Trigger Widget*   | CT_StepList (Custom Table) ▼  |
| Target Widget*    | CT_ToolsList (Custom Table) ▼ |
| Trigger Property* | STEP_ID_FILTER                |
| Target Property*  | STEP_ID                       |

図 61. 手順リストをツールリストに接続

e. Trigger/Action を追加して、手順に関連付けられた Safety Instruction を表示します。

- i. Trigger Widget として CT\_StepList (Custom Table) を選択します。
- ii. Target Widget として CT\_SafetyList (Custom Table) を選択します。
- iii. Trigger Property に STEP\_ID\_FILTER を指定します。
- iv. Target Property に STEP\_ID を指定します。

|                   |                                |
|-------------------|--------------------------------|
| Trigger Widget*   | CT_StepList (Custom Table) ▼   |
| Target Widget*    | CT_SafetyList (Custom Table) ▼ |
| Trigger Property* | STEP_ID_FILTER                 |
| Target Property*  | STEP_ID                        |

図 62. 手順リストを安全指示リストに接続

f. View Carousel から 3D View に Trigger/Action を追加してビューの変更を有効にします。

5. Formula を保存します。

6. シナリオを作成 (または設定) します。(詳細は、3DxSUITE ScenarioEditor マニュアルを参照してください)

- a. CSV データソースを 3D PDF Exporter カスタムパラメーターとして追加します。(図 63 を参照してください)
  - **step\_list** - step\_list データソースの場所を指定します。
  - **step\_part\_list** - step\_part\_list データソースの場所を指定します。
  - **step\_tool\_list** - step\_tool\_list データソースの場所を指定します。
  - **step\_safety\_list** - step\_safety\_list データソースの場所を指定します。

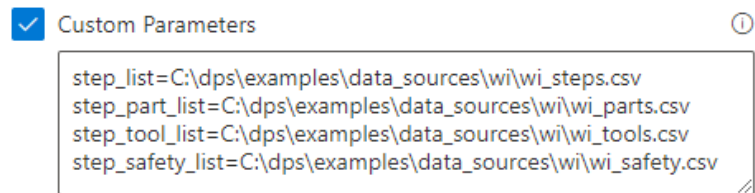


図 63. CSV データソースの場所を指定するカスタムパラメーター

7. 3DxSUITE SmartController を使用してシナリオを実行します。(詳細は [3DxSUITE SmartController を用いた TDP の作成](#) を参照してください。)

  - a. シナリオを選択します。(例: [DPS How-To Guide Scenario](#))
  - b. パラメーターセットを選択します。(例: [Work Instructions Formula](#))
  - c. 入力ファイルを選択します。(例: [C:\dps\examples\data\\_sources\work\\_instructions\TB-Assembly-WI.enf](#))
  - d. ジョブを開始します。

### 6.2.7. ビューの印刷の有効化

本節では、ビューの印刷を有効にする方法を説明します。本手順を実施する前に [TDP の作成](#) をご覧ください。

ファイル例:

| ファイル   | 内容  |
|--|---|
| "templates\basic_how_to_guide_template.pdf"  | TDP テンプレート  |
| "data_sources\example_throttle_body_r01.enf" | 入力モデル   |
| "scenarios\dps_scenario.esa"                 | <a href="#">How-To Guide Formula</a> パラメーターが設定されたシナリオファイル |

手順:

1. Adobe Acrobat で TDP テンプレートを開き、DPS を起動します。
2. 3D View widget、View Carousel widget、Button widget (例: [Print Selected Views](#) 等) を追加します。
3. [widget の外観](#) をカスタマイズします。
4. ビューを印刷するための Trigger/Action を追加します。
  - a. [+ Add Trigger/Action] ボタンをクリックします。
  - b. Trigger widget になるボタンを選択します。(例: [BTN\\_PrintViews \(ボタン\)](#))
  - c. Target widget として [Document](#) を選択します。
  - d. Action として [Print Saved Views](#) を選択します。



- e. 印刷用のビューに切り替えるモードを選択します。(例: **Illustration**)
- f. 印刷用のビューの表示に使用する 3D View widget を選択します。(例: **3D View widget (3D View)**)
- g. 印刷用のビューの選択に使用する View Carousel widget を選択します。(例: **VC\_VIEWS (View Carousel)**)

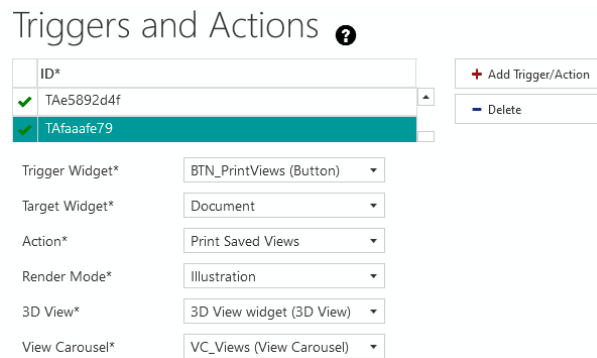


図 64. Button と Print Saved Views の間に Trigger/Action を設定する

7. Formula を保存します。(例: "C:\dps\my\_formula.rb")
8. カスタマイズした TDP を作成します。(詳細は [TDP の作成](#) を参照してください。)

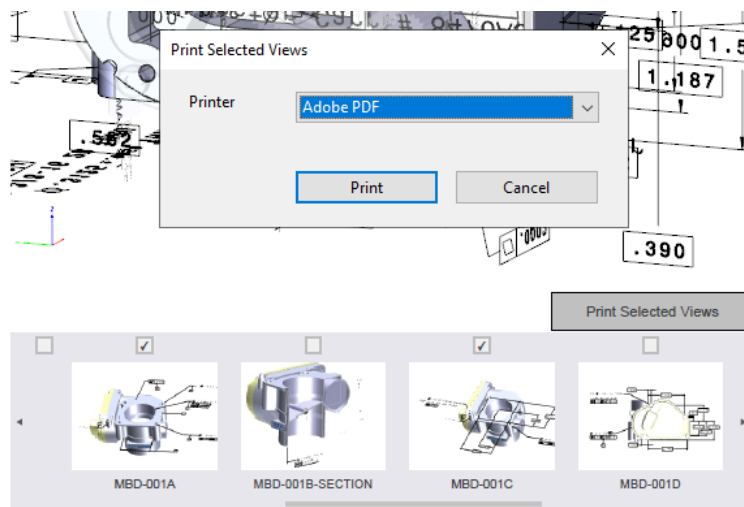


図 65. ビューの印刷を有効化した場合の例

## 6.2.8. エリジオン TDP JavaScript API を用いた 3D ツールバーの追加

本節では、[エリジオン TDP JavaScript API](#) (TDP JS API) を使用して 3D ツールバーを作成する手順について説明します。Data Package Studio (DPS) でカスタマイズ可能な widget の関数のうち、予め選択された一部の関数に対して TDP JS API を用いてアクセスすることで、TDP にカスタマイズした動作を追加することができます。

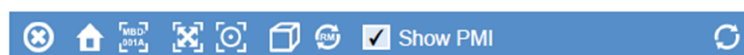


図 66. TDP JS API で作成した 3D ツールバーの例

ここでは以下の内容について説明します:

- API を介してカスタマイズした関数を追加する際の一般的な考え方
- API を取得し、それを使って名前付き widget を取得および操作する方法
- 上記を踏まえた 3D ツールバーの簡易的な作成方法

#### サンプルファイル:

| ファイル   | 内容  |
|--|---|
| "formulas\basic_how_to_guide_template.pdf/json/rb" | Formula                                   |
| "data_sources\example_throttle_body_r01.enf"       | 入力モデル                                     |
| "scenarios\dps_scenario.esa"                       | How-To Guide Formula パラメーターセットを含むシナリオファイル |

### TDP JS API を使ってカスタマイズした関数を追加する際の一般的な考え方

TDP に JavaScript コードを追加することで、TDP JS API を使用してカスタマイズした関数を追加することができます。

コーディングを始める前に、以下の点について検討してください。

1. コードを追加するタイミング: TDPを生成する前か後か
2. コードを追加する方法: コードを埋め込むか、フィールドのアクションに追加するか、またはその2種類の組合せにするか
3. コードの名前付け: どのように名前の競合を回避するか

上記の点に対してどのようなアプローチをとるかは、実現したい内容によって異なります。以下に、それぞれの点についての簡単なガイドラインを示します。

#### コードを追加するタイミング:

- 生成されたすべての TDP で同じ動作をさせたい場合には、TDP を生成する前にコードを追加します。
- 生成された TDP に固有の動作をさせたい場合には、TDP を生成した後にコードを追加します。

#### コードを追加する方法:

- 以下のような場合にはコードを埋め込みます。
  - イベントを受けて関数を実行したい (例: ドキュメントを閉じる前)
  - コードを DRY<sup>1</sup> に保つために関数を再利用したい
- ユーザのマウスイベントを関数のトリガーにしたい場合には、フィールドのアクションにコードを追加します。

- フィールド間で関数を再利用したい場合は、上記を組合せます。

1: DRY = Don't Repeat Yourself - コーディングのベストプラクティスです。例えば、メンテナンスコストの削減につながります。

## コードの名前付け

エリジオン TDP JavaScript API ドキュメントでは、**Ely** を TDP JS API クラスの接頭辞として使用しました (例: **ElyTdpApi**)。しかし、TDP に埋め込まれたクラス、グローバル関数、グローバル変数/定数のすべてがこの接頭辞を持っているわけではありません。従って名前の競合を回避するためには、独自の接頭辞を追加するなどの命名規則を定めることをお勧めします。

## TDP JS APIを使用

アプローチが決まったらコーディングを開始します。最初に TDP JS API を取得します。**ely3dpdf** から取得できる **getApi** 関数を呼び出すことで API を取得することができます。

```
var api = ely3dpdf.getApi();
```

次に、widget を取得してその関数を呼び出します。言い換えると、widget を動作させます。このためには、widget の名前を知っている必要があります。widget 名は、DPS Widgets / Widgets (Fields) 表の Name 列にあります。

### Widgets (Fields) ?

| Name*              | Type*         |
|--------------------|---------------|
| ✓ 3D View widget   | 3D View       |
| ✓ VC_VIEWS         | View Carousel |
| ✓ PT_PmiTable      | PMI Table     |
| ✓ TCF_ViewProperty | Text Field    |
| ✓ BTN_PrintViews   | Button        |

図 67. DPS Widgets / Widgets (Fields) 表の Name 列

次に、widget 名を使って **getWidget** 関数を呼び出し、widget を取得します。

```
var api = ely3dpdf.getApi();
var widget = api.getWidget('3D View widget');
```

これら 2 つの手順により、動作させることのできる widget を取得することができました。以下は、widget のタイプを確認し、それが 3D View widget であった場合にはリセットするコードの例です。

```
// ...
var type = widget.getType();
if (type === 'ElyTdp3dViewWidget') {
  widget.reset();
}
```

次に、これらを使って簡単な 3D ツールバーを実装します。

## TDP JS API を使用して 3D ツールバーを作成

本節では、"[basic\\_how\\_to\\_guide\\_formula.pdf](#)" にある 3D ツールバーの項目の一部を作成した際の基本的な手順を説明します。これらの手順では、ツールバーの項目を作成し、それらの項目に JS コード (関数) を追加しますが、書式設定は行いません。

作成するツールバー項目:

1. Fit All (parts) ボタン
2. PMI 表示/非表示のチェックボックス
3. (3D View) の Reset ボタン

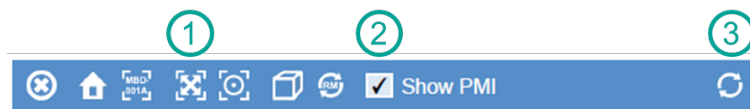


図 68. TDP JS API を使って作成する 3D ツールバー項目の例

最初に、Adobe Acrobat Pro で 3D View widget を含むテンプレートを開き、[Prepare Form] を起動します。Adobe Acrobat Pro ツールバーには、いくつかの新しい項目が表示されていることが分かります。例えば、Text Field、Checkbox、ボタンなどです。



"[basic\\_how\\_to\\_guide\\_formula.pdf](#)" をコピーし、[Prepare Form] を使って既存の 3D ツールバーを削除してください。

### Fit All ボタンの作成

以下の手順で Fit All ボタンを作成します:

1. ボタンを追加して名前を付けます。例: FIT\_ALL\_BTN

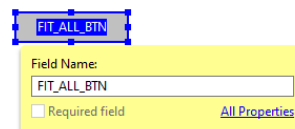


図 69. Fit All ボタンの追加

2. ボタンをダブルクリックしてボタンのプロパティを開きます。
3. [Options] タブでは以下の通りに入力します。
  - a. レイアウトとして "Label only" を選択します。
  - b. ラベルとして "Fit All" を入力します。

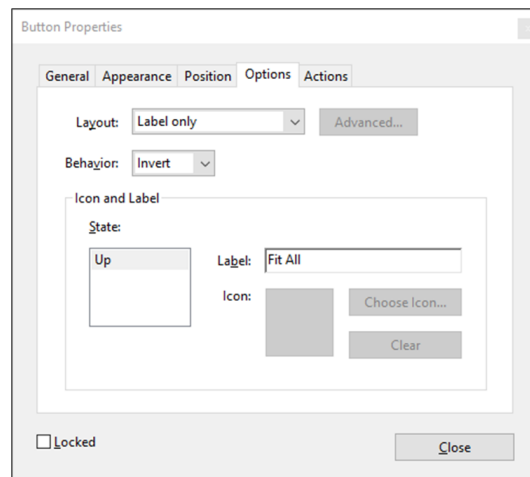


図 70. Fit All ボタンのラベルを設定

4. [Actions] タブでは以下の通りに入力します。
  - a. Trigger として "Mouse Up" を選択します。
  - b. Action として "Run a JavaScript" を選択します。
  - c. [Add...] ボタンをクリックします。

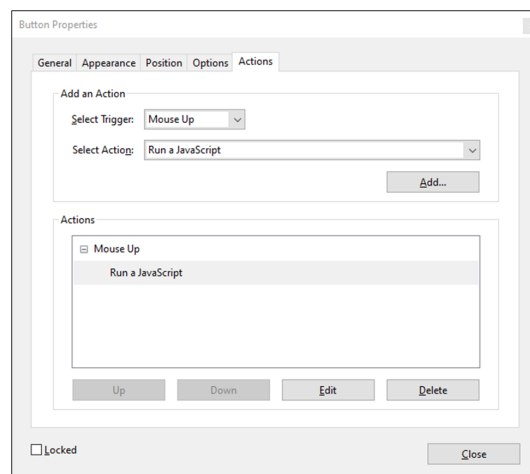


図 71. Fit All ボタンに JavaScript アクションを追加

5. 開いた JavaScript エディタでは以下の作業を実施します。
  - a. Fit All コードを追加します。後述のコードを参照してください。
  - b. コードを保存してエディタを閉じます。(組み込みのエディタを使用する場合、[OK] をクリックします)
6. ボタンプロパティのウィンドウで [Close] をクリックします。

Listing 6. すべてのパートを 3D View にフィットさせるコード

```
var api = ely3dpdf.getApi();
var widget = api.getWidget('3D View widget');
if (widget != null) {
  widget.fitAll();
}
```

}

## PMI の表示/非表示用を切り替えるチェックボックスの作成

PMI の表示/非表示を切り替えるチェックボックスを作成するには、[Fit All] ボタンを作成したときと同じ手順を実行します。新しいチェックボックスに別の名前を付け (例: SHOW\_PMI\_CB)、以下に含まれるコードを JavaScript アクションに追加します。チェックボックスのラベルが必要な場合は、チェックボックスの横に「Text」項目を追加し、「Show PMI」等の名称を設定します。

Listing 7. チェックボックスの値に基づいて PMI の表示/非表示用を切り替えるコード

```
var api = ely3dpdf.getApi();
var widget = api.getWidget('3D View widget');
var checkbox = api.getField('SHOW_PMI_CB');

if (widget != null && checkbox != null) {
    var showPmi = checkbox.isBoxChecked(0);
    widget.setShowPmi(showPmi);
}
```



現状、TDP を開いた際には常に PMI が表示されるようになっているため、デフォルトでチェックボックスにチェックが入っているようにしてください。

## Reset ボタンの作成

3D View widget をリセットするボタンを作成するには、Fit All ボタンを作成した時と同じ手順を実行します。新しいボタンに別の名前を付け (例: RESET\_ALL\_BTN)、ラベルには "Reset 3D" 等の名称を設定します。そして、以下のコードを JavaScript アクションに追加します。

Listing 8. 3D View widget をリセットするコード

```
var api = ely3dpdf.getApi();
var widget = api.getWidget('3D View widget');
if (widget != null) {
    widget.reset();
}
```

## 6.3. TDP の作成

### 6.3.1. 3DxSUITE SmartLauncher を用いた TDP の作成

本節では、3DxSUITE SmartLauncher (SLS) を使用して TDP を作成する方法について説明します。

**サンプルファイル:**

| ファイル   | 内容   |
|--|--|
| "scenarios\dps_scenario.esa"                 | Basic Product Definition Formula<br>パラメーターセットのシナリオ |
| "data_sources\example_throttle_body_r01.enf" | 入力モデル  |

以下の動画では、次の手順について説明しています。

**手順:**

1. Data Package Studioで TDP テンプレートをカスタマイズします。(詳細は [TDP の作成](#) を参照してください。)
2. 3DxSUITE SmartLauncherを用いて TDPを作成します。
  - a. 入力モデルファイルを開きます。(右クリックして **Convert > Change File Type...** をElysiumメニューから選択します。)
  - b. シナリオとして ("DPS How-To Guide Scenario") を選択します。
  - c. パラメーターセットとして ("Basic Product Definition Formula") を選択します。
  - d. 出力フォルダーを指定します。
  - e. ジョブを開始します。

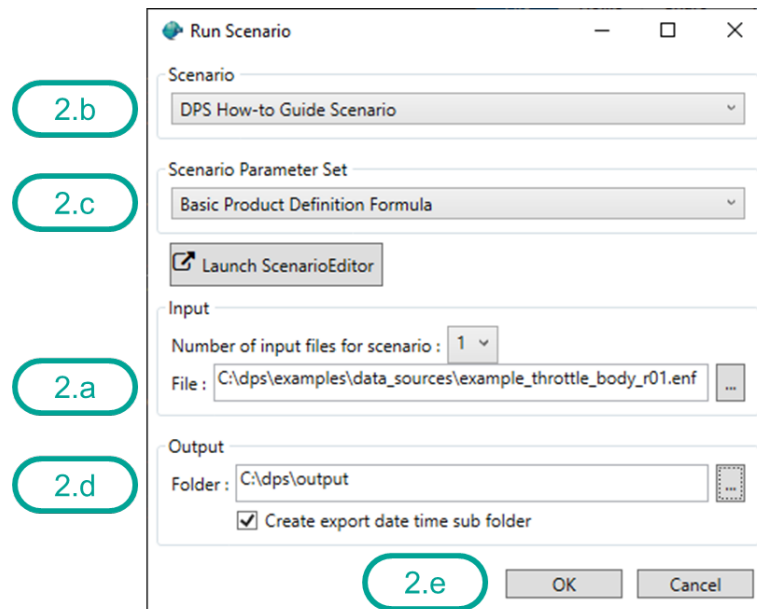


図 72. 3DxSUITE SmartLauncher で TDP を生成する例

### 6.3.2. 3DxSUITE SmartController を用いた TDP の作成

本節では、3DxSUITE SmartController (SC) を使用して TDP を作成する方法について説明します。

**サンプルファイル:**

| ファイル   | 内容   |
|--|--|
| "scenarios\dps_scenario.esa"                 | Basic Product Definition Formula<br>パラメーターが設定されたシナリオ |
| "data_sources\example_throttle_body_r01.enf" | 入力モデル  |

**手順:**

1. Data Package Studio を使用して TDP をカスタマイズします。(TDP の作成 を参照してください)
2. SC を使用して TDP を作成します。
  - a. SC を開きます。
  - b. 新しいジョブタブで以下の手順を実行します。
    - i. シナリオとして ("DPS How-To Guide Scenario") を選択します。
    - ii. パラメーターセットとして ("Basic Product Definition Formula") を選択します。
    - iii. ファイルとして ("example\_throttle\_body\_r01.enf") を選択します。
    - iv. 出力フォルダーを指定します。
    - v. ジョブを開始します。

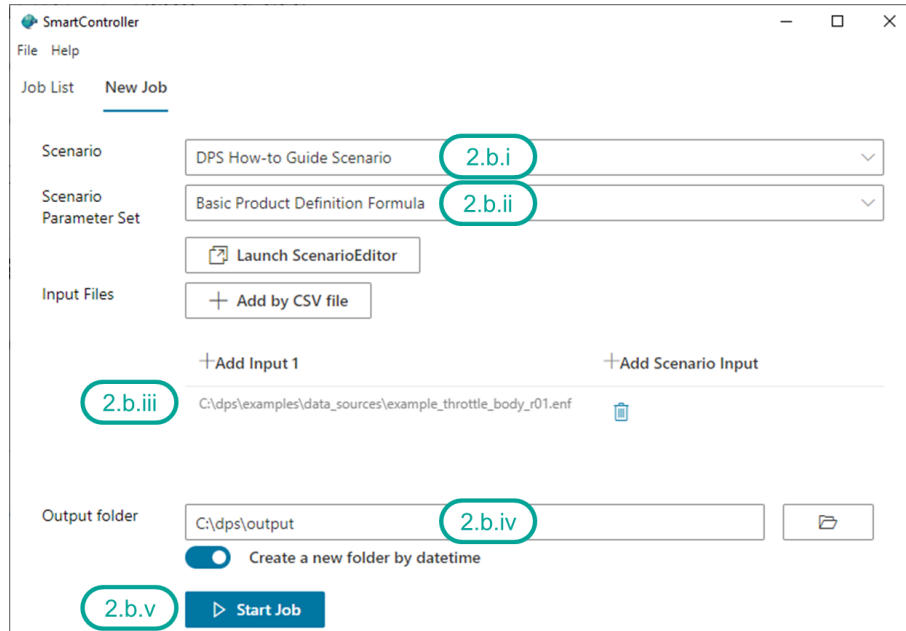


図 73. 3DxSUITE SmartController で TDP を作成する例

**6.3.3. 3DxSUITE TransServer を用いた TDP の作成**

本節では、3DxSUITE TransServerを用いて TDPを作成する方法を説明します。



前提条件:

- サンプルシナリオ "3DPDF" がインポートされた 3DxSUITE TransServer へアクセスします。



“3DPDF” シナリオが見つからない場合は、3DxSUITE TransServer の管理者に「examples」フォルダーに含まれる「example\_ats\_scenario.json」をインポートするよう依頼してください。

サンプルファイル:

| ファイル   | 内容                            |
|--|-------------------------------|
| "example_ats_scenario.json"                  | 3DxSUITE TransServer シナリオファイル |
| "formulas\basic_how_to_guide_formula.rb"     | カスタマイズした入力スクリプト               |
| "data_sources\example_throttle_body_r01.enf" | 入力モデル                         |
| "data_sources\example_revision_history.csv"  | モデルのリビジョン履歴                   |

以下の動画では、次の手順について説明しています。

手順:

1. Data Package Studioで TDP をカスタマイズします。(詳細は [TDP の作成](#) を参照してください。)
2. 3DxSUITE TransServer で TDP を作成します。
  - i. 新しいジョブを作成します。
  - ii. 新しいジョブを設定します。
  - iii. 設定したジョブを実行します。

▶ <https://vimeo.com/373806155/2e7b02a09c> (Vimeo video)

## 7. 予約済みの PDF フィールド名



「ELY\_」で始まる名前の PDF フィールドを作成しないでください。

TDP テンプレートを作成する場合、フィールドに「ELY\_」で始まる名前を付けしないでください。(例: 「ELY\_BoMList」) Elysium が提供するテンプレートに含まれる事前定義フィールドの名前の先頭に「ELY\_」が付いており、3DxSUITE で TDP を正しく作成するには固有のフィールド名を指定する必要があります。

## 8. FAQ

### 8.1. 用語

#### 8.1.1. DPS

DPS とは Data Package Studio の省略名です。

#### 8.1.2. Formula

formula とは、Data Package Studio が出力する 3 種類のファイルの総称です。

- JSON - 定義したデータソース、widget、Action が格納されているファイル
- PDF - TDP テンプレートのコピー
- Ruby - JSON と TDP テンプレートを結合するためのスクリプト



TDP を作成するには、formula となる 3 種類のファイルすべてが必要です。

#### 8.1.3. Data Source (データソース)

データソースとは、TDP を作成する際に 3DxSUITE への入力となるデータを格納したファイルのことです。(例: ENFファイル、CSVファイル) データソースIDを 3DxSUITE にパラメーターとして渡して入力ファイルを指定することで有効になります。



SC<sup>1</sup>、SLS<sup>2</sup>、または TS<sup>3</sup> を使用して TDP を作成する場合、いずれか一つの ENF データソースに "inputfile" を ID として指定する必要があります。

1: SC = 3DxSUITE SmartController; 2: SLS = 3DxSUITE SmartLauncher; 3: TS = 3DxSUITE TransServer.

また、データソースがどのように表示されるかについては [Widget](#) を参照してください。(例: BoM Tableとして) [データソースの追加](#) も併せて参照してください。

Data Package Studio は以下のデータソースをサポートしています。

- [ENF \(Elysium 中間ファイル\)](#)
- [CSV \(csv 形式\)](#)
- [テキストファイル \(txt 形式\)](#)
- [その他](#)

#### ENF (Elysium 中間ファイル)

Elysium 独自のフォーマットです。Elysium 製品の内部処理において中間ファイルとして使用します。

## CSV (csv 形式)

コンマ(",") で区切って値を列挙したテキストファイルです。

## テキストファイル (txt 形式)

Text Field widget の入力として使用できるプレーンテキストファイルです。

## その他

上記以外のファイル形式です。(例: JSON ファイル、XML ファイル) これらのファイル形式は [添付ファイル](#) として TDP に追加することができますが、[Widget](#) への入力として使用することはできません。

### 8.1.4. Widget

widget を使用することで、TDP テンプレート内のフィールドを BoM Table や 3D モデルなどのよりリッチなコンテンツに置き換えて、製品モデルデータを表示したり操作したりすることができます。方法は以下の通りです。

1. それぞれの PDF フィールドをデータソースに接続します。
2. データソースに含まれるデータを PDF フィールドにどのように表示するかを指定します。(例: BoM Table)
3. [Trigger](#) と [Action](#) ペアの接続ポイントを指定します。

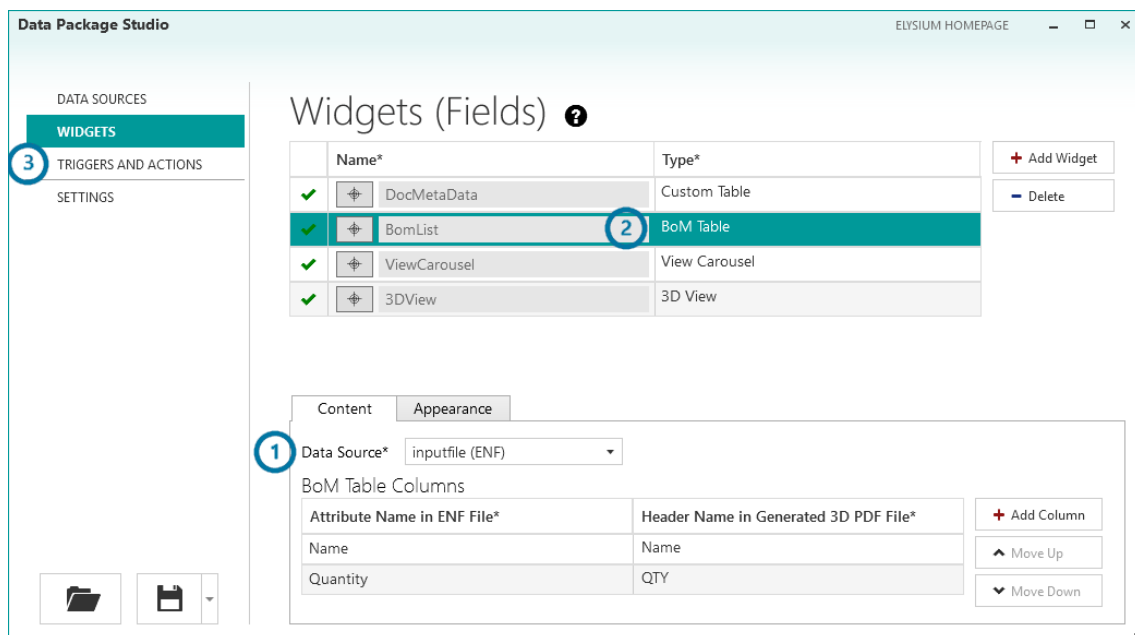


図 74. Widget の設定例

以下の widget を使用することができます。

- [3D View](#)
- [Button](#)

- [BoM Table](#)
- [Custom Table](#)
- [PMI Table](#)
- [Table Column Filter](#)
- [Text Field](#)
- [User Property Table](#)
- [View Carousel](#)

### 8.1.5. Trigger

Trigger は、TDP の同一ページ内にある 2つの Widget 間で定義された応答動作の入力になる部分です。例えば BoM table をクリックするという "Trigger" により、3D View 内で対応するパートがハイライト表示されます。

Trigger は [Action](#) の実行を開始します。

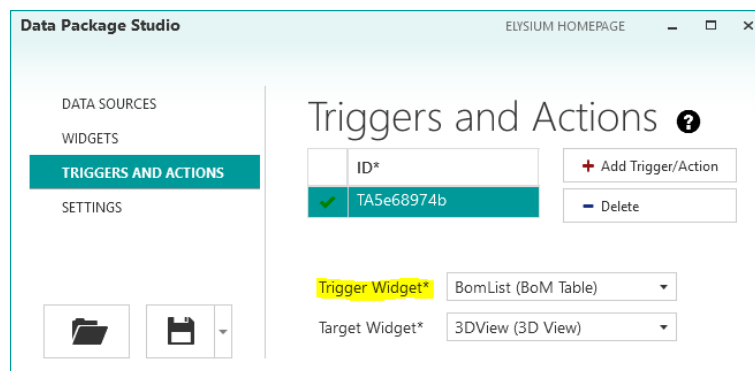


図 75. Trigger の設定例

詳細は「[Trigger と Action の組合せ](#)」や「[Trigger と Action の追加](#)」を参照してください。

### 8.1.6. Action

Action は、TDP の同一ページ内にある 2つの widget 間で定義された応答動作の動作になる部分です。例えば BoM Table の行をクリックしたとき、3D View 内で対応するパートがハイライト表示されるという "Action" が実行されます。

Action が開始されるためにはペアになっている [Trigger](#) が実行される必要があります。

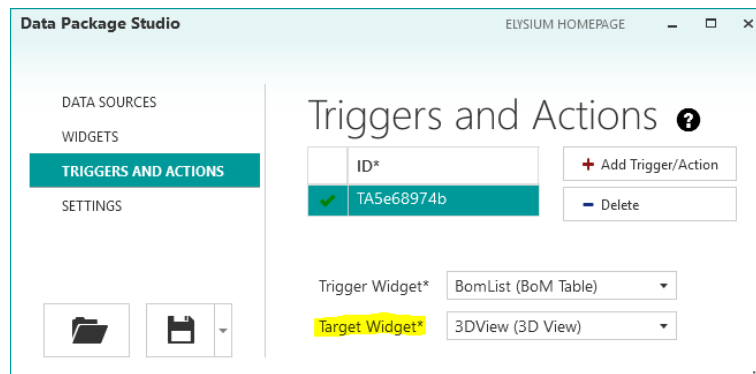


図 76. Action の設定例

詳細は「[Trigger と Action の組合せ](#)」や「[Trigger と Action の追加](#)」を参照してください。

### 8.1.7. Trigger と Action の組合せ

同じページ内にある場合に Trigger/Action ペアを用いて接続することができる widget の組合せは以下の表のとおりです。



Data Package Studio は自動的にいくつかの Trigger/Action ペアを追加します。これらのペアは "Triggers and Actions" ページの一覧には表示されません。

表 10. Trigger/Actionペアを介して接続できる widgetの組合せ

| Trigger   | Action    | 内容   |                 |
|-----------|-----------|--|-----------------|
| BoM table | 3D View   | イベント   | パートを選択する        |
|           |           | 結果   | パートがズームアップされる   |
|           |           | 選択したパートが 3D View 内でズームアップされます。(その他の要素は透明になります)   |                 |
|           | PMI Table | イベント   | パートを選択する        |
|           |           | 結果   | PMI 項目の一覧が表示される |
|           |           | 選択したパートに関連する PMI 項目 (geometric tolerance、データム、およびメモなど。PMI Table widget のカスタマイズ内容によりまず) だけがフィルタリング表示されます。 <ul style="list-style-type: none"> <li>パートが選択されていない場合は、PMI Table widget のカスタマイズ内容に合致するモデルに含まれるすべての PMI 項目が一覧表示されます。</li> <li>パートに PMI 項目がない場合、一覧には何も表示されません。</li> </ul> |                 |

| Trigger      | Action              | 内容  |                          |
|--------------|---------------------|---|--------------------------|
| BoM table    | User Property Table | イベント  | パートを選択する                 |
|              |                     | 結果  | パートのユーザ属性一覧が表示される        |
|              |                     | <p>選択したパートのユーザ属性は、User Property Table にキーと値のペアとして表示されます。</p> <p>1 列目に属性名、2 列目に属性値が表示されます。</p>   |                          |
|              | View Carousel       | イベント  | パートを選択する                 |
|              |                     | 結果  | 選択したパートに対応するビューの一覧が表示される |
|              |                     | <p>View Carousel に選択したパートに対応するビューが表示されます。</p> <ul style="list-style-type: none"> <li>パートが選択されていない場合、そのモデルの全てのビューが表示されます。</li> <li>パートにビューが無い場合、View Carousel は空です。</li> </ul> <p>同じデータソースを共有する <b>BoM Table</b> widget と <b>3D View</b> widget の間に Trigger/Action ペアを追加すると、Trigger/Action ペアを自動的に追加します。</p> |                          |
| Button       | Document            | イベント  | クリックする                   |
|              |                     | 結果  | 印刷ダイアログが表示される            |
|              |                     | 印刷ダイアログが表示され、選択したビューの印刷が可能になります。  |                          |
| Custom Table | Custom Table        | イベント  | テーブルから項目を選択する            |
|              |                     | 結果  | フィルタリングされたテーブルが表示される     |
|              |                     | <p>テーブルから選択した項目の指定されたプロパティが、接続されたテーブルのフィルタリングに使用されます。プロパティの指定には正規表現を利用できます。</p>   |                          |
| Custom Table | 3D View             | イベント  | テーブルから項目を選択する            |
|              |                     | 結果  | ビューが選択される                |
|              |                     | <p>テーブルから選択した項目の指定されたプロパティが、保存されたビューを選択するためのビュー名として使用されます。</p>  |                          |

| Trigger                              | Action     | 内容  |                      |
|--------------------------------------|------------|---|----------------------|
| CV <sup>1</sup><br>Property<br>Table | Document   | イベント  | テーブルから項目を選択する        |
|                                      |            | 結果  | URL で指定されたサイトが開く     |
|                                      |            | <p>選択した Property Table の項目が有効な URL である場合、その URL で指定されるサイトが開きます。</p> <p>1: CAD Validator.</p>  |                      |
| PMI Table                            | 3D View    | イベント  | PMI 項目を選択する          |
|                                      |            | 結果  | PMI 項目がハイライト表示される    |
|                                      |            | <p>選択した PMI 項目が 3D ビューでハイライト表示されます。また、選択した項目から参照する他の PMI 項目もハイライト表示されます。</p>  |                      |
| Table                                | Text Field | イベント  | テーブルから項目を選択する        |
|                                      |            | 結果  | 指定したプロパティが表示される      |
|                                      |            | <p>テーブルから選択した項目の指定されたプロパティが Text Field に表示されます。</p>   |                      |
| Table<br>Column<br>フィルタ              | Table      | イベント  | ユーザーがテキストを入力する       |
|                                      |            | 結果  | フィルタリングされたテーブルが表示される |
|                                      |            | <p>指定された 1 つ以上の列の内容がユーザーの入力した内容と一致するかが判定され、その結果に基づいてテーブルがフィルタリングされます。</p>   |                      |
| View<br>Carousel                     | 3D View    | イベント  | ビューを選択する             |
|                                      |            | 結果  | ビューが表示される            |
|                                      |            | <p>選択したビューが 3D View に表示されます。</p> <p>同じデータソースを共有する View Carousel widget と <a href="#">3D View</a> widget を追加すると、自動的に Trigger/Action ペアが追加されます。</p> |                      |
|                                      | Text Field | イベント  | ビューを選択する             |
|                                      |            | 結果  | 指定したビューのプロパティが表示される  |
|                                      |            | <p>選択したビューのプロパティは、Trigger/Action 接続で指定されたとおり、テキストフィールドに表示されます。</p>  |                      |

| Trigger | Action    | 内容   |                                 |
|---------|-----------|--|---------------------------------|
| 3D View | PMI Table | イベント   | PMI のクリック (マウスボタンを離れた時)         |
|         |           | 結果   | PMI テーブル内の当該 PMI が選択され、ハイライトされる |
|         |           | 3D ビュー上でクリックした PMI は、PMI テーブルで選択・ハイライト表示されます。（フィルタリングされている場合を除く） |                                 |

### 8.1.8. Data Package Studio が 3種類のファイルを保存する理由

Data Package Studio が保存する 3 種類のファイルは、TDP を作成するための formula を構成します。formula や formula ファイルの詳細については「[Formula](#)」を参照してください。

### 8.1.9. 事前定義プロパティとは

事前定義プロパティは、システム（CADシステムやData Package Studioなど）によって定義されており、システムを使用する際に変更することはできません。

事前定義プロパティの例:

- [ENF](#) のPart Name
- PMI 項目のText Rep.



詳細は [事前定義モデルプロパティ](#) と [事前定義 PMI プロパティ](#) を参照してください。

### 8.1.10. ユーザ定義プロパティとは

ユーザ定義プロパティとは、CAD システムなどでユーザが作成したプロパティです。この種類のプロパティは、任意の方法で追加、削除、名前変更をすることができます。

### 8.1.11. 事前定義プロパティとユーザ定義プロパティの違いとは

詳細は [事前定義プロパティとは](#) と [ユーザ定義プロパティとは](#) を参照してください。

## 8.2. 機能の拡張

### 8.2.1. Data Package Studio の機能拡張

現時点において、Data Package Studio の機能を拡張することは製品としてサポートしていません。今後リリースするバージョンでサポートする可能性があります。



## 8.2.2. Data Package Studio の API ドキュメント

現時点において Data Package Studio の API は未公開です。詳細は「[Data Package Studio の機能拡張](#)」を参照してください。このため API に関するドキュメントも提供していません。

## 8.3. フォーマット変更とカスタマイズ

### 8.3.1. 添付ファイルの TDP への一括追加

3DxSUITEで '[FileAttachmentFolder](#)' パラメーターを使用して、添付するファイルの場所を指定することができます。



指定したフォルダ内のファイルはすべて添付されます。ただし、サブフォルダ内のファイルは添付されません。

### 8.3.2. 表の背景色の変更

Data Package Studio EX8.2 以降では表の背景色を変更することができます。詳細は [Widgetをカスタマイズ](#) を参照してください。

### 8.3.3. CAD Validator レポートに含まれる elements/widgets のフォーマット変更

CAD Validator レポートに追加された [non-CAD Validator widgets](#) のフォーマットを変更することができます。[CAD Validator widget](#) で、何を表示するかカスタマイズすることもできます。（例: Difference Summary widget で Passed/Failed assessment のみを表示）



詳細は [CAD Validator レポートのカスタマイズ](#) を参照してください。

ただし、現時点において CAD Validator widgets のフォーマット変更 (例: フォント/背景色) は製品としてサポートしていません。今後リリースするバージョンでサポートする可能性があります。

### 8.3.4. TDP に含まれる表の列/行のサイズ変更

Data Package Studio EX8.2 以降では、列の幅を変更することができます。ただし、現時点では行の高さを明示的に設定することはできません。表に表示する行数を指定することで行の高さを暗黙的に設定することはできます。

詳細は [Widgetをカスタマイズ](#) を参照してください。

### 8.3.5. CAD 比較レポートに含まれる表の列の削除

現時点において CAD 比較レポートに含まれる表の列を削除することは製品としてサポートしていません。今後リリースするバージョンでサポートする可能性があります。

### 8.3.6. レポートに含まれる表のスクロール

表の右側にある上下の矢印を使用して表をスクロールすることができます。

### 8.3.7. CAD 比較レポートの全体フィットと選択解除について

Data Package Studio の 3D View widget は、そのままでは「全体フィット」および「選択解除」の機能を使えません。しかし、エリジオン TDP JavaScript API を使って [Fit All] ボタンと [Clear Selection] ボタンを追加すれば可能です。詳細は[エリジオン TDP JavaScript API を用いた 3D ツールバーの追加](#)を参照してください。

### 8.3.8. 3D ビューの背景色が設定した色と異なる点について

3DxSUITE の enf23dpdf コンポーネントに '**BackgroundColor**' パラメーターが設定されていないか確認してください。設定されている場合には Data Package Studio で行ったカスタマイズ内容が上書きされます。

### 8.3.9. View Carousel のサムネイル画像が表示されない現象について

サムネイルの生成が有効になっていることを確認するには、3DxSUITE enf23dpdf の '**CreateViewCarouselThumbnail**' パラメーターに '**0**' が設定されていないことを確認してください。'**0**' が設定されている場合、サムネイルは作成されません。(デフォルトの '**1**' を設定することで有効になります。)

サムネイルの作成を有効にしてもサムネイルが表示されない場合は、次の理由が考えられます:

1. 入力モデル (ENF) には、ビューが含まれていない可能性があります。元モデルを ENF に変換する際に、'**XConvertView**' パラメーターに '**0**' が設定されていないことを確認してください。
2. 接続された **BoM Table** widget で選択されたパートにはビューがない可能性があります。詳細は [View Carousel Widget の追加およびカスタマイズ](#) を参照してください。

### 8.3.10. 添付ファイル名の自動読み込みについて

添付ファイルのファイル名は自動的に読み込まれます。TDP の添付ファイルリストで添付ファイルの名前として使用されます。ファイルパスはインポートされません。

## 8.4. インストールとライセンス

### 8.4.1. Data Package Studio 使用時のライセンスについて

3DxSUITE EX9.0 では Data Package Studio 使用時にライセンスを必要としません。

### 8.4.2. Adobe Acrobat の plug\_ins フォルダの場所

Adobe Acrobat plug\_ins フォルダは Adobe Acrobat アプリケーションフォルダにあります。通常のパスは以下の通りです。

「C:\Program Files\Adobe\Acrobat <version>\Acrobat\plug\_ins」

または

「C:\Program Files(x86)\Adobe\Acrobat <version>\Acrobat\plug\_ins」

## 8.5. PDF

### 8.5.1. ファイルの添付

手順については使用しているアプリケーションのマニュアルを参照してください。Adobe 社の Adobe Acrobat を使用している場合には以下のリンクから検索してください。

[https://helpx.adobe.com/jp/acrobat/using/links-attachments-pdfs.html#open\\_save\\_or\\_delete\\_an\\_attachment](https://helpx.adobe.com/jp/acrobat/using/links-attachments-pdfs.html#open_save_or_delete_an_attachment)

### 8.5.2. 日付フィールドの追加

手順については使用しているアプリケーションのマニュアルを参照してください。Adobe 社の Adobe Acrobat を使用している場合には以下のリンクから検索してください。

<https://helpx.adobe.com/jp/acrobat/kb/support-for-image-and-date-fields-in-acroforms.html>



テンプレートに日付フィールドを追加すると、TDP を作成するときに問題が発生する可能性があります。(詳細は [既知の問題](#) を参照してください)

### 8.5.3. 不足しているフォントを PDF に追加

PDF が思ったように表示されない場合、または PDF のコンテンツを適切に操作できない場合 (例: コメントの追加)、PDF で指定されているフォントが使用できない状態である可能性があります。(当該フォントが PDF に埋め込まれていない可能性や、OS に含まれていない可能性が考えられます。)

Adobe Acrobat を使用している場合は、システムに Adobe Reader フォントパックをインストールすることで解決することができます。Adobe Reader フォントパックは、下記のページの下部にある「ADD-ONS」からダウンロードすることができます。

Windows: <https://supportdownloads.adobe.com/product.jsp?product=10&platform=Windows>

Mac: <https://supportdownloads.adobe.com/product.jsp?product=10&platform=Mac>

Adobe Acrobat を使用していない場合、または Adobe Reader フォントパックをインストールしても問題が解決しない場合は、不足しているフォントをインストールする必要があります。OS のマニュアルを参照してください。

あなたが PDF の作成者である場合には、以下のページを参考にして、使用したフォントが PDF に埋め込まれているかを確認することもできます。

<https://helpx.adobe.com/jp/acrobat/using/analyzing-documents-preflight-tool-acrobat.html>

### 8.5.4. ページの追加およびページの整理

手順については使用しているアプリケーションのマニュアルを参照してください。Adobe 社の Adobe Acrobat を使用している場合には以下のリンクから検索してください。

<https://helpx.adobe.com/jp/acrobat/how-to/organize-insert-pages-pdf.html>

### 8.5.5. タブ及びボタンの追加

エリジオンの CAD 比較レポートで使用しているタブは、それぞれ 3種類の要素で作成されています。

1. タブページとして機能するページ
2. タブのように見える要素
3. タブのように見える要素の上にある見えないボタン (正確にはタブページへのリンクのようなもの)

手順については使用しているアプリケーションのマニュアルを参照してください。Adobe 社の Adobe Acrobat を使用している場合には以下のリンクから検索してください。

<https://helpx.adobe.com/jp/acrobat/using/setting-action-buttons-pdf-forms.html>

### 8.5.6. 透かしの追加

手順については使用しているアプリケーションのマニュアルを参照してください。Adobe 社の Adobe Acrobat を使用している場合には以下のリンクから検索してください。

<https://helpx.adobe.com/jp/acrobat/using/add-watermarks-pdfs.html>

### 8.5.7. 編集可能な PDF フィールドの作成及び変更

手順については使用しているアプリケーションのマニュアルを参照してください。Adobe 社の Adobe Acrobat を使用している場合には以下のリンクから検索してください。

<https://helpx.adobe.com/jp/acrobat/user-guide.html?topic=/jp/ja/acrobat/morehelp/forms.ug.js>

### 8.5.8. テンプレートの作成

TDP テンプレートの作成 を参照してください。

### 8.5.9. PDF 内テキストの編集

手順については使用しているアプリケーションのマニュアルを参照してください。Adobe 社の Adobe Acrobat を使用している場合には以下のリンクから検索してください。

<https://helpx.adobe.com/jp/acrobat/using/edit-text-pdfs.html>

### 8.5.10. デジタル署名の追加

手順については使用しているアプリケーションのマニュアルを参照してください。Adobe 社の Adobe Acrobat を使用している場合には以下のリンクから検索してください。

<https://helpx.adobe.com/jp/acrobat/how-to/fill-and-sign-forms-anywhere.html>

<https://helpx.adobe.com/jp/acrobat/using/certificate-based-signatures.html>



デジタル署名は TDP を作成した後に追加してください。(詳細は [既知の問題](#) を参照してください。)



証明書ベースの署名を使用すると、3D コンテンツが期待通りに動作しない可能性があります。3D コンテンツを有効にするには TDP への変更が必要ですが、証明書ベースの署名を行っている場合には変更が許可されないためです。

### 8.5.11. 3D コンテンツの有効化

手順については使用しているアプリケーションのマニュアルを参照してください。Adobe 社の Adobe Acrobat を使用している場合には以下のリンクから検索してください。

<https://helpx.adobe.com/jp/acrobat/using/enable-3d-content-pdf.html>

### 8.5.12. パスワードによる保護

手順については使用しているアプリケーションのマニュアルを参照してください。Adobe 社の Adobe Acrobat を使用している場合には以下のリンクから検索してください。

[https://helpx.adobe.com/jp/acrobat/using/securing-pdfs-passwords.html#id\\_48581](https://helpx.adobe.com/jp/acrobat/using/securing-pdfs-passwords.html#id_48581)

### 8.5.13. 添付ファイルの確認

手順については使用しているアプリケーションのマニュアルを参照してください。Adobe 社の Adobe Acrobat を使用している場合には以下のリンクから検索してください。

[https://helpx.adobe.com/jp/acrobat/using/links-attachments-pdfs.html#open\\_save\\_or\\_delete\\_an\\_attachment](https://helpx.adobe.com/jp/acrobat/using/links-attachments-pdfs.html#open_save_or_delete_an_attachment)

## 8.6. ユーザインタフェース

### 8.6.1. ユーザインタフェースに関するその他の情報について

Data Package Studio のユーザインタフェースに関するヘルプを参照したい場合には、【?】ボタンをクリックしてください。

## 9. テンプレートおよびデータソースの例

本ドキュメントを読まれる方の理解を助けるため、いくつかのサンプルファイル(テンプレート、データソース、formula およびシナリオ)を用意しました。(詳しい内容については、[サンプルファイル](#)を参照してください) Data Package Studio と併せて提供したドキュメントの並びにある "examples" フォルダー内にあります。

サンプルのテンプレート、formula、およびシナリオをそのまま使用するか、変更したバージョンを使用して、無制限に使用/配布できる独自のTDPを作成することができます。

ご注意いただきたい点:



1. 修正テンプレートから「Elysium」ロゴタイプを削除してください。
2. 修正テンプレートの「Empowered by Elysium」ロゴタイプはそのままにしてください。

詳細は「Data Package Studio End User License Agreement」を参照してください。

## 9.1. サンプルファイルの設定

サンプル formula とサンプルシナリオをそのまま使用できるように、次の手順を実行することをお勧めします:

1. "examples" フォルダを "C:\dps" へコピーします。
2. サンプルシナリオを "scenarios" フォルダへコピーします。
3. インポートされたサンプルシナリオを実行するように 3DxSUITE SmartLauncher を構成します。

### 9.1.1. サンプルファイルをコピー

"examples" フォルダを "C:\dps" へコピーすると、フォルダ構造は以下の通りになります。

```
c:
|-dps
  |-examples
    |-data_sources
    |-formulas
    |-output
    |-scenarios
    |-templates
```

### 9.1.2. サンプルシナリオをコピー

"examples\scenarios" フォルダーにあるファイルを任意の "scenarios" フォルダーにコピーすることをお勧めします。(例: "C:\Users\Public Documents\Elysium\3DxSUITE\Scenarios") これにより、サンプルシナリオを使用して 3DxSUITE SmartLauncher から繰り返し TDP 作成タスクを実行することができます。

### 9.1.3. サンプルシナリオを実行するために 3DxSUITE SmartLauncher を構成する

次に、サンプルシナリオが配置されているフォルダをシナリオフォルダとして使用するよう 3DxSUITE SmartLauncher を構成します。

3DxSUITE SmartLauncherを構成:

1. 3DxSUITE SmartLauncher でのシナリオ実行機能を可能にします。
  - a. 任意のファイルで右クリックし、**Elysium > Preferences** を選択します。
  - b. **[Yes]** ボタンをクリックします。(デバイスに対する変更の可否を尋ねられた場合)
  - c. **[Feature]** タブ/ボタンをクリックします。



- d. 「シナリオ実行機能を使用する」のチェックボックスにチェックが入っていることを確認します。
- e. **【OK】** をクリックします。

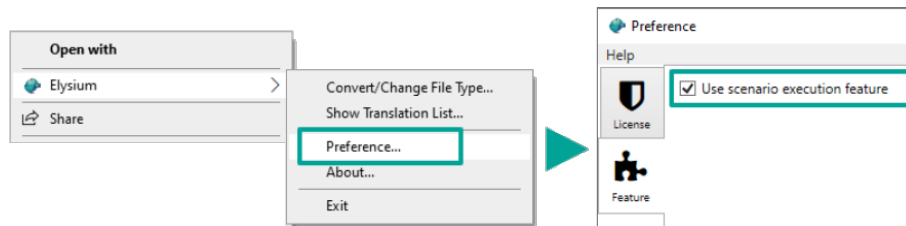


図 77. 3DxSUITE SmartLauncher でのシナリオ実行を可能にします

2. 3DxSUITE 共通設定 でシナリオファイルの格納場所を指定します。
  - a. Windows スタートメニューから、**エリジオン 3DxSUITE** > **3DxSUITE 共通設定** を選択します。
  - b. **【Yes】** ボタンをクリックします。(デバイスに対する変更の可否を尋ねられた場合)
  - c. **【Feature】** タブ/ボタンをクリックします。
  - d. サンプルシナリオファイルを格納した「シナリオフォルダ」を指定します。
  - e. **【OK】** をクリックします。

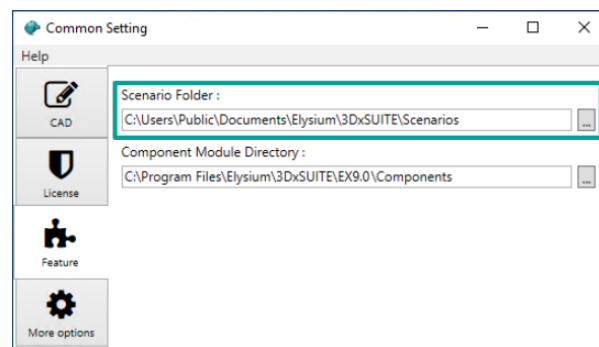


図 78. シナリオファイルの格納場所を指定します

## 9.2. サンプルファイル

### 9.2.1. データソース

3D モデルのデータソース (\*.enf および \*.stp) は内部的な検証にのみ使用してください。

| ファイル                          | 内容   |
|-------------------------------|--|
| example_revision_history.csv  | 本マニュアルで使用するリビジョン履歴の例                         |
| example_throttle_body_r00.enf | 本マニュアルで使用する3Dモデル (以下のENFファイル (V2) とは若干異なります) |

| ファイル                                 | 内容   |
|--------------------------------------|--|
| example_throttle_body_r01.enf        | 本マニュアルで使用する3Dモデル (上記のENFファイル (V1) とは若干異なります) |
| TBM-MBD-R00_BoC.enf                  | Boc "balloons" を持つ3Dモデルの例                    |
| work_instructions\TB-Assembly-WI.enf | 作業指示の手順に対応するビューが含まれている 3D モデル                |
| work_instructions\wi_steps.csv       | 作業指示の手順の詳細 (対応するビューの名前を含む)                   |
| work_instructions\wi_parts.csv       | 作業指示の手順に関連する部品データ                            |
| work_instructions\wi_tools.csv       | 作業指示の手順に関連するツールデータ                           |
| work_instructions\wi_safety.csv      | 作業指示の手順に関連する安全指示                             |

## 9.2.2. Formulas

| ファイル                            | 内容  |
|---------------------------------|---|
| basic_how_to_guide_formula.json | basic_how_to_guide_template.pdf に基づいて作成したものです。 <a href="#">基本機能</a> で使用します。                             |
| basic_how_to_guide_formula.pdf  |   |
| basic_how_to_guide_formula.rb   |   |
| basic_product_definition.json   | basic_product_definition_template.pdf に基づいて作成したものです。単一ページの製品定義 TDP が作成されます。                             |
| basic_product_definition.pdf    |   |
| basic_product_definition.rb     |   |
| boc_formula.json                | boc_example_template.pdf に基づいて作成したものです。 <a href="#">Bill of Characteristics Tableの追加</a> で使用します。        |
| boc_formula.pdf                 |   |
| boc_formula.rb                  |   |
| cv_example_formula.json         | cv_example_template.pdf に基づいて作成したものです。Data Package Studio で実施した カスタマイズ内容を反映した CAD Validatorレポート を作成します。 |
| cv_example_formula.pdf          |   |
| cv_example_formula.rb           |   |



| ファイル            | 内容  |
|-----------------|---|
| wi_formula.json | wi_example_template.pdf に基づいて作成したものです。 <a href="#">作業指示 TDP の作成</a> で使用します。 |
| wi_formula.pdf  |   |
| wi_formula.rb   |   |

### 9.2.3. 出力

サンプルの formula を用いて作成した TDP ファイルです。参考としてお使いください。

| ファイル                                   | 内容  |
|--|---|
| basic_how_to_guide_scenario_output.pdf | dps_scenario.esa シナリオおよび (Basic How-To Guide Formulaのパラメーターセット) example_throttle_body_r01.enf revision_history.csv を用いて作成した ファイルです。 |
| basic_product_definition_output.pdf    | dps_scenario.esa シナリオおよび (Basic Product Definition Formulaのパラメーターセット) example_throttle_body_r01.enf を用いて作成した ファイルです。                |
| boc_output.pdf                         | dps_scenario.esa シナリオおよび (BoC Formula のパラメーターセット) TBM-MBD-R00_BoC.enf を用いて作成したファイル です。  |
| cv_output.pdf                          | cv_scenario.esa シナリオおよび example_throttle_body_r00.enf example_throttle_body_r01.enf を用いて作成したファイル です。                                |
| wi_output.pdf                          | dps_scenario.esa シナリオおよび (Work Instructions Formulaのパラメーターセット) TB-Assembly-WI.enf を用いて作成したファイル です。                                  |

### 9.2.4. シナリオ

シナリオは、3DxSUITE がどのコンポーネント (Adapter と Optimizer) をどのような順序で実行するかを指定します。また、コンポーネントの実行時に使用されるパラメーターセットも指定します。

| ファイル            | 内容   |
|-----------------|--|
| cv_scenario.esa | Data Package Studio で実施したカスタマイズ内容を反映したCAD Validator レポートを生成するシナリオです。 |

| ファイル                      | 内容  |
|---------------------------|---|
| dps_scenario.esa          | <p>以降で説明する手順の大半で使用するシナリオです。このシナリオは以下のパラメーターセットを含んでいます。いずれもすべての PMI (flat-to-screen PMI 含む) を変換するように設定されています。</p> <ul style="list-style-type: none"> <li>• <b>Basic How-To Guide Formula</b> <ul style="list-style-type: none"> <li>◦ Formula: basic_how_to_guide_formula.rb</li> <li>◦ データソース: example_revision_history.csv</li> <li>◦ すべてのモデルビューを変換する</li> </ul> </li> <li>• <b>Basic Product Definition Formula</b> <ul style="list-style-type: none"> <li>◦ Formula: basic_product_definition.rb</li> <li>◦ すべてのモデルビューを変換する</li> </ul> </li> <li>• <b>BoC Formula</b> <ul style="list-style-type: none"> <li>◦ Formula: boc_formula.rb</li> <li>◦ モデルビューを変換しない</li> </ul> </li> <li>• <b>Work Instructions Formula</b> <ul style="list-style-type: none"> <li>◦ Formula: wi_formula.rb</li> <li>◦ データソース: <ul style="list-style-type: none"> <li>▪ wi_steps.csv</li> <li>▪ wi_parts.csv</li> <li>▪ wi_tools.csv</li> <li>▪ wi_safety.csv</li> </ul> </li> <li>◦ トップのモデルビューを変換する</li> </ul> </li> </ul> |
| example_ats_scenario.json | 3DxSUITE TransServer を用いた TDP の作成 で使用される 3DxSUITE TransServer シナリオ  |

### 9.2.5. スクリプト

BoC (Bill of Characteristics) が PMI Table widget で適切に表示されるようにするため、ENF 内で BoC データを構成するのに用いる前処理スクリプトのサンプルです。詳細は [Bill of Characteristics Tableの追加](#) を参照してください。

| ファイル               | 内容   |
|--------------------|--|
| pre_process_boc.rb | TDP 前処理の一部として、エリジオン で作成した BoC "balloons" を準備するスクリプトです。 |

| ファイル                    | 内容                               |
|-------------------------|----------------------------------|
| pre_process_template.rb | TDP 前処理の一部として実行するスクリプトのテンプレートです。 |

## 9.2.6. テンプレート

| ファイル                                  | 内容  |
|---------------------------------------|---|
| basic_how_to_guide_template.pdf       | 単一ページの TDP が作成されます。 <a href="#">TDP の作成</a> と <a href="#">基本機能</a> で使います。 |
| basic_product_definition_template.pdf | 単一ページの製品定義 TDP が作成されます。   |
| pd_example_template.pdf               | 「カバーシート」、「製品定義」、および「ヘルプ」ページを含む TDP が作成されます。                               |
| tdp_example_template.pdf              | 単一ページの製品定義 TDP が作成されます。   |
| tdp_example_template_design.pptx      | tdp_example_template.pdf テンプレートデザインの PowerPoint ファイルです。                   |
| wi_example_template.pdf               | 単一ページの作業指示書 TDP が作成されます。  |

# 付録 A: 3DxSUITE パラメーター

## A.1. 3D PDF Component で必要となるパラメーター

### 'CustomizedScript' / スクリプトファイル (ファイルパス)

TDP を作成する場合、使用するカスタマイズ済み formula (Ruby スクリプト) への絶対パスを指定します。(例えば、Data Package Studio で作成したformula)

### 'Script' / 実行スクリプト

値に 'Custom' を設定してください。設定されていない場合、'CustomizedScript' のパラメーターが無視されます。

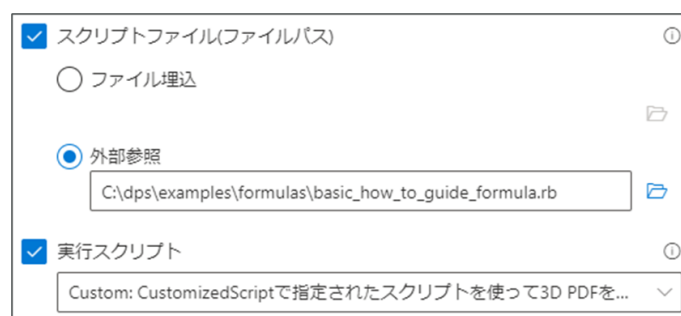


図 79. 3D PDF Component で必要となるパラメーターの使用例

## A.2. 3D PDF Component のオプションパラメーター

### 'CreateViewCarouselThumbnail' / ビューカーセルのサムネイル

View Carousels のサムネイルを表示する ('1') か表示しない ('0'、デフォルト) かを指定します。

### 'CustomizedEnfPreProcessScript' / プレ処理用カスタマイズスクリプト (ファイルパス)

TDP を作成する際に事前プロセスで使用するカスタマイズ済みスクリプトへの絶対パスを指定します。

### 'SaveAsHtml' / HTML変換 (開発中機能) [非推奨]

PDF に加えて TDP を HTML として保存する ('1') か保存しない ('0'、デフォルト) かを指定します。HTML ファイルは作業フォルダーに保存されます。本機能は Python 3.4 以降が必要であることに注意してください。(64 ビットバージョンを推奨します。) こちらは現時点で実験的な機能となりますのでご注意ください。実際の運用には使用せず、検証等を目的として使用して下さい。

警告: [非推奨] であるとは、'SaveAsHtml' を引き続き使用できるものの、今後のリリースで予告なしにサポートが終了する予定であることを意味しています。

### 'SectionCapping' / セクションキャップ

Cross Section をキャップありで表示する ('1', デフォルト) かキャップなしで表示する ('0') かを指定します。

**'XConvertFlatToScreenPMI' / 画面に平行なPMIの変換**

Flat-to-screen PMI を、画面に平行なPMIとして変換する ('1', デフォルト) か通常のPMIとして変換する ('0') かを指定します。パラメーターは 'XConvertPMI' が '1' に設定された場合のみ有効となります。

**'XConvertPMI' / PMI変換**

PMI を変換する ('1', デフォルト) か変換しない ('0') かを指定します。

**'<CustomDataSource>' / 任意パラメーター**

Data Package Studio に追加されたカスタムデータソースは、パラメーターとして使用することができます。(例: 'revision\_history') データソースに関連付けたいファイルへの絶対パスを指定してください。

図 80. 3D PDF component のオプションパラメーター使用例

## A.3. CAD Validator Component で必要となるパラメーター

**'3DPdfReportType' / スクリプトの種類 (3D PDF レポート)**

値として 'Custom' を設定してください。設定しない場合、'Customized3DPdfReportScript' で指定した内容が無視されます。

**'Create3DPdfReport' / 比較レポートファイル (3D PDF形式) 作成**

CAD比較レポートを 3D PDF 形式で作成したい場合に '1' を指定します。

**'Customized3DPdfReportScript' / カスタマイズスクリプト (3D PDF レポート)**

レポートのカスタマイズに使用する formula file にパラメーターを設定します。

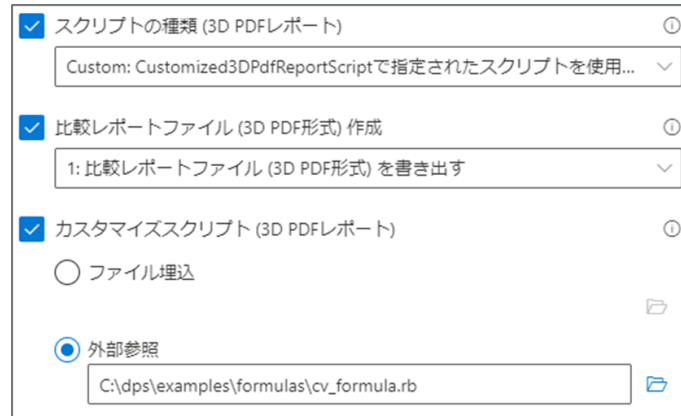


図 81. CAD Validator component のオプションパラメーター使用例

## A.4. CAD Validator Component のオプションパラメーター

**'PdfTableMode' / 閲覧パフォーマンス (3D PDF レポート)**

パラメーターを '1' に設定すると、パフォーマンスを向上させるために簡略化した比較表を使用してレポートが作成されます。(通常の表のデフォルトは '0' です。)

**'ReportLanguage' / 表示言語 (3D PDF レポート/3D HTML レポート)**

英語版 CAD 比較テンプレートを使用する場合、値に 'en' を指定します。日本語版のテンプレートを使用する場合は 'ja' を指定します。言語コード ISO 639-1 を使用してください。

**'Custom Parameters'**

"KEY=VALUE" のように KEY と VALUE のペアを等号で区切って追加することで、任意のカスタムパラメーターを指定することができます。1 行に 1 組の KEY/VALUE ペアを追加します。3D PDF プロパティを設定する場合、KEY のプレフィックスに \$ENF23DPDF\_ を付けた KEY/VALUE ペアを追加します。例えば 3D ビューの背景色を赤に設定する場合、\$ENF23DPDF\_BackgroundColor=(255,0,0) を追加します。

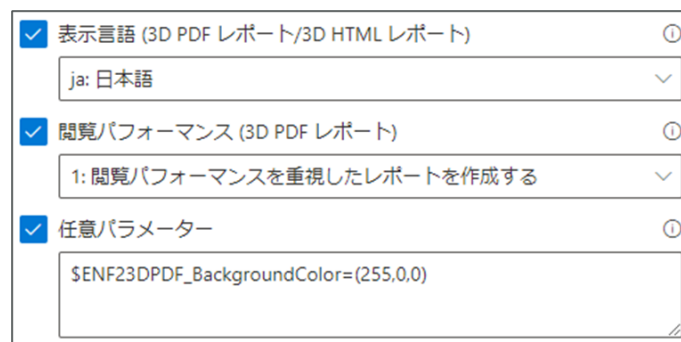


図 82. CAD Validator component のオプションパラメーター使用例

## A.5. 3DxSUITE のオプションパラメーター (CAD変換関連)

### 'XConvertDimension' / Dimension変換 (PMI)

値に '1' を設定してください。設定しない場合 ENF に寸法の情報が含まれません。

### 'XConvertGDT' / GDT変換 (PMI)

値に '1' を設定してください。設定しない場合 ENF に GDT の情報が含まれません。

### 'XConvertMaterial' / マテリアル属性変換 (Attribute)

値に '1' を設定してください。設定しない場合 ENF にマテリアルの情報が含まれません。

### 'XConvertNote' / Note変換 (PMI)

値に '1' を設定してください。設定しない場合 ENF に注記が含まれません。

### 'XConvertSystemProperty' / システム属性変換 (Attribute)

値に '1' を設定してください。設定しない場合 ENF に定義済みプロパティが含まれません。

### 'XConvertUserProperty' / ユーザ属性変換 (Attribute)

値に '1' を設定してください。設定しない場合 ENF にユーザプロパティが含まれません。

### 'XConvertView' / ビュー変換

値に '1' を設定してください。設定しない場合 ENF にビューの情報が含まれません。

|                                     |                      |   |
|-------------------------------------|----------------------|---|
| <input checked="" type="checkbox"/> | Dimension変換(PMI)     | ① |
|                                     | 1: 変換する              | ▼ |
| <input checked="" type="checkbox"/> | GDT変換(PMI)           | ① |
|                                     | 1: 変換する              | ▼ |
| <input checked="" type="checkbox"/> | マテリアル属性変換(Attribute) | ① |
|                                     | 1: 変換する              | ▼ |
| <input checked="" type="checkbox"/> | Note変換(PMI)          | ① |
|                                     | 1: 変換する              | ▼ |
| <input checked="" type="checkbox"/> | システム属性変換(Attribute)  | ① |
|                                     | 1: 変換する              | ▼ |
| <input checked="" type="checkbox"/> | ユーザー属性変換(Attribute)  | ① |
|                                     | 1: 変換する              | ▼ |
| <input checked="" type="checkbox"/> | ビュー変換                | ① |
|                                     | 1: 変換する              | ▼ |

図 83. CAD translation に有効な一部のパラメーター使用例

## 付録 B: 事前定義モデルプロパティ

以下は Data Package Studio で利用可能なすべての事前定義プロパティと、それに対応する [ENF](#) のプロパティの一覧です。いくつかの主要な CAD システム/フォーマットについて、CAD から ENF へ変換する際に併せて変換されるプロパティも記載しています。(プロパティを変換するための設定については [3DxSUITE のオプションパラメーター \(CAD変換関連\)](#) を参照してください。)

| Data Package Studio でのプロパティ     | ENF でのプロパティ        | CAD | CAD でのプロパティ | 説明                                   |
|---------------------------------|--------------------|-----|-------------|--------------------------------------|
| Executed CAD Version            | ExecutedCadVersion | -   | -           | CAD ファイルを ENF に変換する際に利用した CAD のバージョン |
| Native <sup>1</sup> CAD Version | NativeCadVersion   | -   | -           | CAD ファイルを作成する際に使用した CAD の製品名およびバージョン |
| Native File Base Name           | NFName             | -   | -           | モデルのファイル名 (ファイル拡張子を含む。パスは含まない。)      |



| Data Package Studio でのプロパティ      | ENF でのプロパティ         | CAD                     | CAD でのプロパティ                                       | 説明                           |
|----------------------------------|---------------------|-------------------------|---|------------------------------|
| Native File Modified Date        | NativeFileLastWrite | 3DEXPERIENCE / CATIA V6 | Last modification                                 | ネイティブ CAD ファイルの最終更新日時        |
|                                  |                     | NX                      | 入力ファイルの最終更新時刻                                     |                              |
|                                  |                     | STEP                    | 入力ファイルの最終更新時刻                                     |                              |
|                                  |                     | STEP BOM                | PartVersion / DateTimeAssignment (Role is update) |                              |
|                                  |                     | XPDMXML                 | TimeModified (Product)                            |                              |
| Native File Name                 | NFName              | -                       | -   | モデルファイルのパスとファイル名(ファイル拡張子を含む) |
| Native File Size                 | NativeFileSize      | -                       | -   | CADファイルの サイズ                 |
| Part <sup>2</sup> Change History | ChangeHistory       | NX I-deas               | Change History                                    | 注記の変更履歴                      |
| Part Comment                     | PartComment         | Inventor                | Description                                       | パートに付随する注記・コメント              |
|                                  |                     | STEP                    | PRODUCT_DEFINITION_FORMATION.description          |                              |
| Part Definition                  | PartDefinition      | CATIA V5                | Definition  | パートの定義                       |
| Part Density                     | Density             | 算出された値                  |   | パートの密度                       |

| Data Package Studio でのプロパティ | ENF でのプロパティ | CAD                     | CAD でのプロパティ                              | 説明             |
|-----------------------------|-------------|-------------------------|--|----------------|
| Part Description            | Description | 3DEXPERIENCE / CATIA V6 | Description                              | パートの説明         |
|                             |             | CATIA V5                | Description                              |                |
|                             |             | NX I-deas               | Description                              |                |
|                             |             | Inventor                | Comments                                 |                |
|                             |             | MDT                     | Description                              |                |
|                             |             | STEP                    | PRODUCT. description                     |                |
| Part Identifier             | Name        | CATIA V4                | Work Space Name                          | パートを識別するための文字列 |
|                             |             | CATIA V5                | PartNumber                               |                |
|                             |             | CFIO                    | Object Name                              |                |
|                             |             | NX I-deas               | Name                                     |                |
|                             |             | Inventor                | Name                                     |                |
|                             |             | NX                      | Part Name                                |                |
|                             |             | ProE                    | Name                                     |                |
|                             |             | SolidWorks              | Component Name                           |                |
|                             |             | STEP                    | PRODUCT.id or<br>PRODUCT.name (if no id) |                |

| Data Package Studio でのプロパティ | ENF でのプロパティ | CAD                     | CAD でのプロパティ                      | 説明           |
|-----------------------------|-------------|-------------------------|----------------------------------|--------------|
| Part Layer                  | PartLayer   | CATIA V5                | Part Layer                       | パートの所属するレイヤー |
| Part Mass                   | Mass        | 算出された値                  |                                  | パートの質量       |
| Part Material               | Material    | CATIA V5                | Material Name                    | パートの素材       |
|                             |             | NX I-deas               | Material Name                    |              |
|                             |             | Inventor                | Material                         |              |
|                             |             | MDT                     | Material                         |              |
|                             |             | ProE                    | Material                         |              |
|                             |             | SolidWorks              | Material                         |              |
| Part Name                   | PartName    | 3DEXPERIENCE / CATIA V6 | Name                             | パート名         |
|                             |             | CATIA V5                | Nomenclature                     |              |
|                             |             | NX I-deas               | Name                             |              |
|                             |             | STEP                    | PRODUCT.id                       |              |
|                             |             | STEP BOM                | Part / Name /<br>CharacterString |              |

| Data Package Studio でのプロパティ | ENF でのプロパティ  | CAD                     | CAD でのプロパティ                                 | 説明                      |
|-----------------------------|--------------|-------------------------|---|-------------------------|
| Part Number                 | PartNumber   | CATIA V5                | PartNumber                                  | パートの識別番号                |
|                             |              | NX I-deas               | Part Number                                 |                         |
|                             |              | Inventor                | Part Number                                 |                         |
|                             |              | STEP                    | PRODUCT.id                                  |                         |
|                             |              | STEP BOM                | Part / Id / Identifier.id                   |                         |
| Part Revision               | PartRevision | 3DEXPERIENCE / CATIA V6 | Revision                                    | パートの リビジョン              |
|                             |              | CATIA V5                | Revision                                    |                         |
|                             |              | NX I-deas               | Revision                                    |                         |
|                             |              | Inventor                | Revision Number                             |                         |
|                             |              | ProE                    | Revision                                    |                         |
|                             |              | STEP BOM                | PartVersion / Id / Identifier.id            |                         |
| Part Source                 | PartSource   | CATIA V5                | Source                                      | パートのソース (例: build, buy) |
|                             |              | STEP                    | PRODUCT_ DEFINITION_ FORMATION. make_or_buy |                         |
| Part Surface Area           | SurfaceArea  | 算出された値                  |   | パートの表面積                 |

| Data Package Studio でのプロパティ | ENF でのプロパティ | CAD       | CAD でのプロパティ                     | 説明        |
|-----------------------------|-------------|-----------|---------------------------------|-----------|
| Part Version                | PartVersion | NX I-deas | Version                         | パートのバージョン |
|                             |             | STEP      | PRODUCT_DEFINITION_FORMATION.id |           |
| Part Volume                 | Volume      | 算出された値    |                                 | パートの体積    |

1 : Nativeとは、ENF に変換する前のファイルを意味しています。

2 : ここでの Part とは、パートもしくはアセンブリです。

## 付録 C: 事前定義 PMI プロパティ

以下は、Data Package Studio（DPS）で利用可能なすべての事前定義 PMI プロパティの一覧です。一覧には最初に共通のプロパティを、次に PMI の種類ごとに DPS UI と同じ順序でプロパティを示しています。

- [Common](#)
- [Datum](#)
- [Datum Target](#)
- [Dimension](#)
- [Geometric Tolerance](#)
- [Line Weld](#)
- [Locator](#)
- Note (すべての PMI タイプに共通のプロパティのみが含まれているため、一覧に表示されません)
- [Roughness](#)
- [Spot Weld](#)
- [Bill of Characteristic](#)
- Other (すべての PMI タイプに共通のプロパティのみが含まれているため、一覧に表示されません)

### C.1. 共通プロパティ

| プロパティ | 内容        |
|-------|-----------|
| Name  | PMI 項目の名前 |

| プロパティ     | 内容  |
|-----------|---|
| Text Rep. | PMI 項目のテキスト表現<br>(例: 1.100   A   B   D-E) |
| Type      | PMI 項目のタイプ (例: note, dimension/distance)  |

## C.2. Datum プロパティ

| プロパティ       | 内容              |
|-------------|-----------------|
| Datum Label | データムのラベル (例: A) |

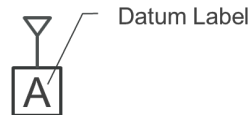


図 84. データムのデータムラベルプロパティの例

## C.3. Datum Target プロパティ

| プロパティ             | 内容  |
|-------------------|---|
| 下の文字列 (1番目)       | 文字とターゲット番号を識別するデータム (例: A1)               |
| Unit System (1番目) | Datum Target の単位 (例: millimeter または inch) |
| 上の文字列 (1番目)       | ターゲット領域のサイズ (例: 5x5)                      |



図 85. Datum Target プロパティの例

## C.4. Dimension プロパティ

| プロパティ                 | 内容  |
|-----------------------|---|
| 下の文字列 (1番目)           | メインとなる Dimension 要素の下のテキスト<br>(例: Tapped .875-14 UNF-3B).       |
| 下の文字列 (2番目)           | Dimension の値/トレランスの下のテキスト                                       |
| 下の文字列 (3番目)           | Dimension の値/トレランス(副寸法)より下のテキスト                                 |
| Prefix (1番目)          | メインとなる Dimension 要素の前のテキスト (例: 4X)                              |
| Prefix (2番目)          | Dimension の値/トレランスの接頭辞  |
| Prefix (3番目)          | Dimension の値/トレランス(副寸法)の接頭辞                                     |
| Subtype               | Dimension のサブタイプ (例: diameter と distance)                       |
| Suffix (1番目)          | メインとなる Dimension 要素の後のテキスト (例: THRU)                            |
| Suffix (2番目)          | Dimension の値/トレランスの接尾辞  |
| Suffix (3番目)          | Dimension の値/トレランス(副寸法)の接尾辞                                     |
| Tolerance Class (1番目) | Dimension のトレランスクラス<br>(例: H7 または h7 - hole fit または shaft fit). |



| プロパティ                 | 内容  |
|-----------------------|---|
| Tolerance Class (2番目) | Dimension のトレランスクラス(副寸法)<br>(例： H7 または h7 - hole fit または shaft fit) |
| Tolerance (1番目)       | Dimension のトレランス (例: -.100 / +.100)                                 |
| Tolerance (2番目)       | Dimension のトレランス (副寸法)<br>(例： -2.540 / +2.540)                      |
| Unit System (1番目)     | Dimension の主単位 (例: inch)  |
| Unit System (2番目)     | Dimension の副寸法 (例: millimeter)                                      |
| Value (1番目)           | Dimension 値 (例: 2.500)  |
| Value (2番目)           | Dimension 値 (副寸法) (例: 63.500)                                       |

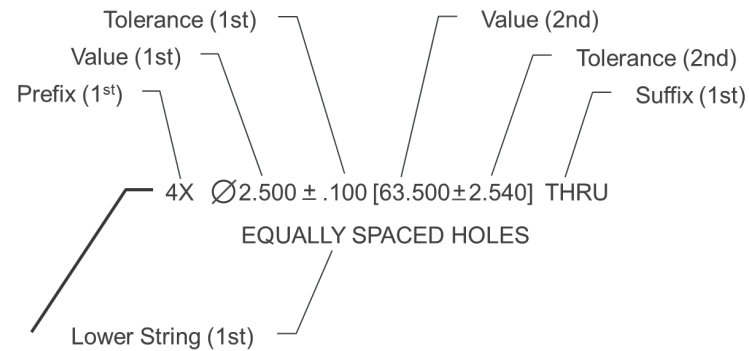


図 86. *Dimension* プロパティの例

## C.5. Geometric Tolerance プロパティ

| プロパティ                                   | 内容   |
|---|--|
| All Around                              | Geometric Tolerance を「all around」適用するか (true) 適用しないか (false) を指定 |
| Datum (1st/2nd/3rd, Cell 1)             | Geometric Tolerance の1行目、2行目、3行目の1次データム参照 (例: A)                 |
| Datum (1st/2nd/3rd, Cell 2)             | Geometric Tolerance の1行目、2行目、3行目の2次データム参照 (例: B)                 |
| Datum (1st/2nd/3rd, Cell 3)             | Geometric Tolerance の1行目、2行目、3行目の3次データム参照 (例: C)                 |
| Geom. Tol. Characteristic (1st/2nd/3rd) | Geometric Tolerance の1行目、2行目、3行目の特性<br>(例: position)             |
| Lower String (1st)                      | Geometric Tolerance の下のテキスト                                      |
| Prefix (1st/2nd/3rd)                    | Geometric Tolerance の1行目、2行目、3行目の前のテキスト                          |
| Projected Value (1st/2nd/3rd)           | Geometric Tolerance の1行目、2行目、3行目の予測値                             |
| Suffix (1st/2nd/3rd)                    | Geometric Tolerance の1行目、2行目、3行目の後のテキスト                          |
| Tolerance (1st/2nd/3rd)                 | Geometric Tolerance の1行目、2行目、3行目の許容値<br>(例: 1.100)               |
| Unit System                             | Geometric Tolerance の単位 (例: ミリメートル または インチ)                      |
| Upper String (1st)                      | Geometric Tolerance の上のテキスト                                      |

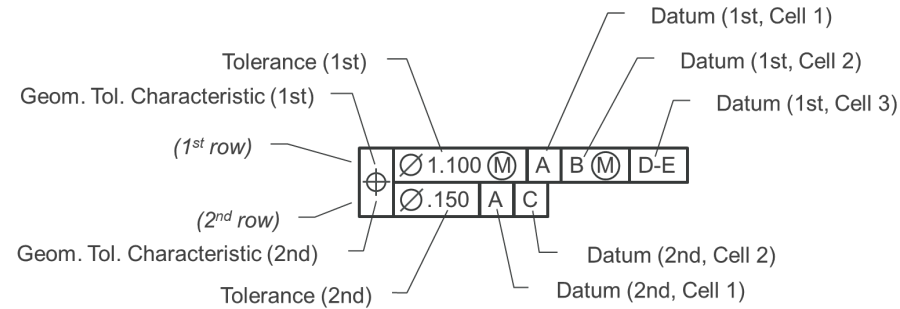


図 87. Geometric Tolerance プロパティの例

## C.6. Line Weld プロパティ

| プロパティ                           | 内容   |
|---------------------------------|--|
| All Around                      | Line Weldが「all around」に適用されるか（true） 否か（false）を指定 |
| Coordinated Entity              | 非連想ラベルを指定します。例えば、溶接される部品を識別するといった目的に使用することができます。 |
| Field Symbol                    | フィールドシンボルが表示されるか（true） 表示されないか（false）を指定         |
| Groove Angle (Lower/Upper)      | Line Weld の下部/上部溝角度                              |
| Longitudinal Size (Lower/Upper) | Line Weld の縦方向のサイズ                               |
| Reference Flag                  | Line Weld が参照として使用されるか（true） 否か（false）を指定        |
| Root Radius (Lower/Upper)       | Line Weld のルート半径                                 |
| Root Spacing (Lower/Upper)      | Line Weld のルート間隔                                 |

| プロパティ                           | 内容  |
|---------------------------------|---|
| Special Directive               | Line Weld に関する特別な指示を指定                      |
| Staggered Flag                  | Line Weld がずれているか (true) そうでないか (false) を指定 |
| Staggered Size (Lower/Upper)    | Line Weld の上下の互い違いのサイズ                      |
| Surface Condition (Lower/Upper) | Line Weld の下面/上面の状態 (例: convex と flat)      |
| Surface Finish (Lower/Upper)    | Line Weld の下面/上面仕上げ (例: F と G)              |
| Test Method                     | Line Weld のテスト方法 (例: mt と pt_f)             |
| Total Size (Lower/Upper)        | Line Weld の合計サイズ                            |
| Weld Size (Lower/Upper)         | Line Weld の溶接サイズ                            |
| Weld Type (Lower/Upper)         | Line Weld の溶接タイプ                            |

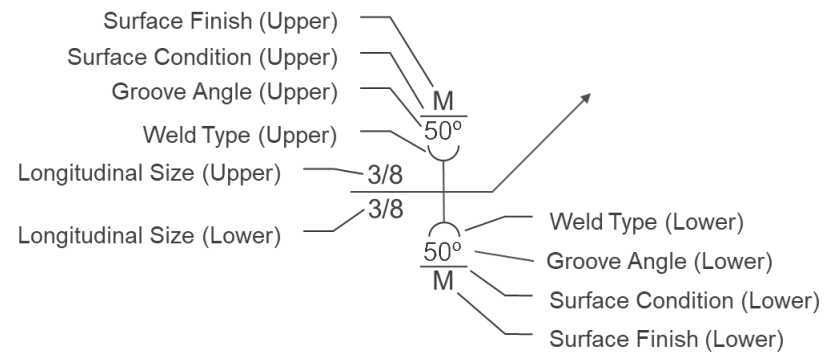


図 88. Line Weld プロパティの例

## C.7. Locator プロパティ

| プロパティ                  | 内容   |
|------------------------|--|
| Coordinated Entity     | 非連想ラベルを指定します。例えば Locator のコンテキストを記録するために使用することができます。 |
| Locator Note           | Locator のメモ  |
| Locator Part Number    | Locator が参照するパート番号                                   |
| Locator Reference Flag | Locator が参照対象であるか (true) 否か (false) を示すフラグ           |
| Locator Symbol Type    | Locator のシンボルタイプ (例: edge と rectangle).              |
| Locator Type           | Locator のタイプ (例: fixing と molds).                    |

## C.8. Roughness プロパティ

| プロパティ      | 内容   |
|------------|--|
| All Around | 粗さを「all around」適用するか (true) 適用しないか (false) を指定 |
| Field 5    | Field 5 のテキスト/値                                |
| Field 6    | Field 6 のテキスト/値                                |
| Field 7    | Field 7 のテキスト/値                                |
| Field 8    | Field 8 のテキスト/値                                |

| プロパティ    | 内容               |
|----------|------------------|
| Field 9  | Field 9 のテキスト/値  |
| Field 10 | Field 10 のテキスト/値 |

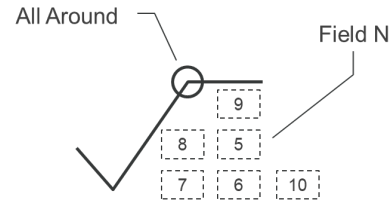


図 89. All Around Roughness プロパティの例

## C.9. Spot Weld プロパティ

| プロパティ              | 内容  |
|--------------------|---|
| Coordinated Entity | 非連想ラベルを指定します。例えば位置を合わせるために部品または嵌合パネル上の点を指定するといった目的に使用することができます。 |
| Critical           | Spot Weld がクリティカルか (yes) 否か (no) または不明かを指定                      |
| Diameter           | Spot Weld の直径   |
| Geometry Flag      | Spot Weld は形状か (true) 否か (false) を指定                            |
| Group ID           | Spot Weld が属するグループの ID  |
| Inspection Flag    | Spot Weld を検査するか (true) 否か (false) を指定                          |
| Joint ID           | Spot Weld が属する結合の ID  |

| プロパティ               | 内容  |
|---------------------|---|
| Life-Cycle Standard | Spot Weld のライフサイクルの標準                             |
| Manufacturing Code  | Spot Weld の製造コード                                  |
| Process             | Spot Weld のプロセス                                   |
| Shape Type          | Spot Weld の形状タイプ (例: hemisphere と shank)          |
| Subtype             | Spot Weld のサブタイプ (例: resistance_2 と projection_3) |
| Weld Type           | Spot Weld の溶接タイプ (例: edge と seam)                 |

## C.10. Bill of Characteristic (BoC) プロパティ

| プロパティ         | 内容                                     |
|---------------|--|
| Ref. PMI Type | BoC が参照する PMI タイプ (例: note、 dimension) |

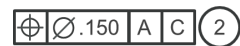


図 90. PMI の形状トレランスタイプに関する *Bill of Characteristics balloon* の例

## 付録 D: Widget の最小サイズ

プレースホルダーのサイズが [表 11](#) に記載している最小サイズ (幅または高さ) よりも小さい Widget は作成に失敗する場合があります。



Widget の最小サイズは、そのサイズの Widget が作成可能であることを示しているに過ぎません。Widget が意図した用途に使用できることは保証していません。

[表 11](#) に記載している Widget の最小サイズは次のように算出しています:

- Widget のすべてのオプション項目を含む (例: チェックボックス)
- ボーダーの太さを最小値 **0** に設定する (ボーダーなし)
- スクロールバーのサイズを最小値 **5** に設定する (Tables のみ)

従って、widget のボーダーやスクロールバーのサイズを変更した場合、それに応じてテンプレート内の Widget プレースホルダーのサイズも調整する必要があります。さらに Table の場合、表示する列と行の数はそれぞれ最小幅と高さに影響するため、考慮する必要があります。View Carousel の場合、表示されるサムネイル (表示可能なビュー) の数を考慮する必要があります。



記載しているサイズはすべて数値を切り上げているため、サイズが完全に一致しない場合があります。 (例:  $2.118 \text{ mm} = 0.083385827 \text{ inch}$  となりますが  $2.2 \text{ mm} \neq 0.084 \text{ inch}$  となります)

Data Package Studio (DPS) 内のサイズは point 単位で指定します。1 point は、およそ  $0.353 \text{ mm}$  または  $0.014 \text{ inch}$  に相当します。その結果、ボーダーの太さを **2** に設定した場合、widget の最小サイズに  $2 \times 0.353 \approx 2.2 \text{ mm}$  を追加する必要があります。

最小サイズの調整に関する他の例については、[Example 1](#) から [Example 6](#) を参照してください。

表 11. Widget プレースホルダーの最小幅 (W) と高さ (H)

| Widget                      | W/H | サイズ mm (inch) | 備考 |
|-----------------------------|-----|---------------|----|
| CAD Validator 向けではない widget |     |               |    |
| 3D View                     | W   | 1.0 (0.040)   | -  |
|                             | H   | 1.0 (0.040)   |    |
| Button                      | W   | 1.0 (0.040)   | -  |
|                             | H   | 1.0 (0.040)   |    |
| Table Column Filter         | W   | 7.5 (0.296)   | -  |
|                             | H   | 2.5 (0.099)   |    |



| Widget                          | W/H | サイズ mm (inch)                          | 備考   |
|---------------------------------|-----|--|--|
| Tables                          | W   | 7.5/col (0.296/col)                    | 列の数によります。 <a href="#">Example 1</a> を参照してください。                 |
|                                 | H   | 5.0 + 1.0/row<br>(0.200 + 0.040/row)   | 行の数によります。 <a href="#">Example 1</a> を参照してください。                 |
| Text Field                      | W   | 2.5 (0.099)                            | -  |
|                                 | H   | 2.5 (0.099)                            |  |
| View Carousel<br>(印刷なし)         | W   | 3.6 + 5.5/view<br>(0.138 + 0.217/view) | スクロールバーのサイズと可視ビューの数によります。 <a href="#">Example 2</a> を参照してください。 |
|                                 | H   | 16.8 (0.659)                           | スクロールバーのサイズによります。 <a href="#">Example 2</a> を参照してください。         |
| View Carousel<br>(印刷する)         | W   | 7.6 + 7.5/view<br>(0.296 + 0.296/view) | スクロールバーのサイズと可視ビューの数によります。 <a href="#">Example 3</a> を参照してください。 |
|                                 | H   | 16.8 (0.659)                           | スクロールバーのサイズによります。 <a href="#">Example 3</a> を参照してください。         |
| <b>CAD Validator 向けの Widget</b> |     |  |  |
| 3D View                         | W   | 20.0 - 131.5 (0.788 = 5.186)           | カスタマイズの内容によります。 <a href="#">Example 4</a> を参照してください。           |
|                                 | H   | 20.0 (0.788)                           | -  |
| Assembly Tree                   | W   | 30.0 (1.182)                           | -  |
|                                 | H   | 10.0 (0.394)                           | -  |
| CAD Information                 | W   | 20.0 (0.788)                           | -  |
|                                 | H   | 17.5 (0.689)                           | -  |
| Color Bar                       | W   | 98.3 (3.871)                           | -  |
|                                 | H   | 6.3 (0.249)                            | -  |
| Component Summary               | W   | 140.0 (5.512)                          | -  |
|                                 | H   | 23.0 (0.906)                           | -  |

| Widget             | W/H | サイズ mm (inch)                | 備考  |
|--------------------|-----|------------------------------|---|
| Detail List        | W   | 12.1 - 113.5 (0.477 - 4.469) | カスタマイズの内容によります。<br><a href="#">Example 5</a> を参照してください。 |
|                    | H   | 27.0 (1.063)                 | -   |
| Difference Summary | W   | 23.0 - 114.0 (0.906 - 4.490) | カスタマイズの内容によります。<br><a href="#">Example 6</a> を参照してください。 |
|                    | H   | 14.0 (0.552)                 | -   |
| Execution Property | W   | 20.0 (0.788)                 | -   |
|                    | H   | 12.5 (0.493)                 | -   |
| Property Table     | W   | 180.0 (7.087)                | -   |
|                    | H   | 12.5 (0.493)                 | -   |
| Type Summary       | W   | 80.0 (3.150)                 | -   |
|                    | H   | 23.0 (0.906)                 | -   |
| Validation Setting | W   | 40.0 (1.575)                 | -   |
|                    | H   | 12.5 (0.493)                 | -   |

Example 1. 3 列 6 行の Table の最小サイズ

**Width:**

$$22.5 \text{ mm} = 3^A \times 7.5^B$$

$$0.888 \text{ inch} = 3^A \times 0.296^B$$

**Height:**

$$11.0 \text{ mm} = 5.0^C + 6 \times 1.0^D$$

$$0.437 \text{ inch} = 0.197^C + 6 \times 0.040^D$$

A: 列の数 B: 各列の幅 C: ヘッダーの高さ(固定) D: 各行の高さ

Example 2. 9 point のスクロールバーと 5 つのビューを持つ View Carousel の最小サイズ

**Width:**

$$30.7 \text{ mm} = 9^A \times 0.353^B + 5^C \times 5.5^D$$

$$1.211 \text{ inch} = 9^A \times 0.014^E + 5^C \times 0.217^D$$

**Height:**

$$19.9 \text{ mm} = 9^A \times 0.353^B + 15.0^F$$

$$0.716 \text{ inch} = 9^A \times 0.014^E + 0.590^F$$

A: point のスクロールバー B: point から mm への変換率 C: ビューサムネイルの表示数 D: ビューサムネイルの幅 E: point から inch への変換率 F: サイズ固定(ビュー、ビュー名)

*Example 3. 9 point のスクロールバーと 5 つのビューを持つ印刷ありの View Carousel の最小サイズ*

#### Width:

$$34.7 \text{ mm} = 4.0^A + 9^B \times 0.353^C + 5^D \times 5.5^E$$

$$1.349 \text{ inch} = 0.138^A + 9^B \times 0.014^F + 5^D \times 0.217^E$$

#### Height:

$$19.9 \text{ mm} = 9^B \times 0.353^C + 15.0^G$$

$$0.716 \text{ inch} = 9^B \times 0.014^F + 0.590^G$$

A: 「Check All」 のチェックボックス B: point のスクロールバー C: point から mm への変換率 D: ビューサムネイルの表示数 E: ビューサムネイルの幅 F: point から inch への変換率 G: サイズ固定(ビュー、ビュー名、とチェックボックス)

*Example 4. すべてのオプションを含む CAD Validator 3D ビューのサイズ*

#### Width:

$$131.5 \text{ mm} = 2 \times 1.8^A + 4 \times 1.5^B + 21.2^C + 28.3^D + 30.0^E + 30.0^F + 5.7^G + 5.7^H + 1.0^I$$

$$5.186 \text{ inch} = 2 \times 0.071^A + 3 \times 0.060^B + 0.835^C + 1.115^D + 0.182^E + 1.182^F + 0.225^G + 0.225^H + 0.040^I$$

A: Widget のパディング(左/右) B: フィールド間の余白 (Action と Refresh を除く) C: Clear 選択ボタン D: Fit オプションのセレクトター E: Renderer モードのセレクトター F: Geometry タイプのセレクトター G: Action メニュー H: Refresh ボタン I: 3D ビューの幅

*Example 5. すべてのオプションを含む CAD Validator 詳細リストのサイズ*

#### Width:

*Title/Navigation 領域の最小幅:*

$$20.0 \text{ mm} = 9.6^A + 9.6^B$$

$$0.756 \text{ inch} = 0.378^A + 0.378^B$$

*Toolbar 領域の最小幅:*

$$113.5 \text{ mm} = 2 \times 1.1^C + 17.6^D + 5.3^E + 15.2^F + 9.9^G + 13.4^H + 5.8^E + 44.1^I$$

$$4.473 \text{ inch} = 2 \times 0.044^C + 0.693^D + 0.209^E + 0.599^F + 0.390^G + 0.528^H + 0.229^E + 1.737^I$$

*リスト領域の最小幅:*

$$45.0 \text{ mm} = 15.0^J + 30.0^K$$

$$1.773 \text{ inch} = 0.591^J + 1.182^K$$

*最小幅:*

$$113.5 \text{ mm} = \max(20.0^L, 113.5^M, 45.0^N)$$

$$4.473 \text{ inch} = \max(0.756^L, 4.473^M, 1.773^N)$$

A: Previous ボタン B: Next ボタン C: ツールバーのパディング(左/右) D: Show Diff Only チェックボックス E: フィールド間の余白 F: Geometry チェックボックス  
G: PMI チェックボックス H: Attributes チェックボックス I: Model View セレクター J: Group リスト K: Element リスト L: Title/Navigation の領域 M: Toolbar の領域 N: List の領域

*Example 6. すべてのオプションを含む CAD Validator 差異のサマリーのサイズ*

**Width:**

$$114.0 \text{ mm} = 2 \times 3.2^A + 3 \times 2.2^B + 16.6^C + 30.2^D + 24.0^E + 30.2^F$$

$$4.490 \text{ inch} = 2 \times 0.126^A + 3 \times 0.087^B + 0.654^C + 1.189^D + 0.945^E + 1.189^F$$

A: Widget のパディング(左/右) B: フィールド間の余白 C: 評価フィールド D: 形状の差異のサマリー E: PMI の差異のサマリー F: 属性の差異のサマリー

# 付録 E: エリジオン TDP JavaScript API

3D ツールバーの追加など、より高度な TDP のカスタマイズを可能にするため、エリジオン TDP JavaScript API (略して TDP JS API) を Data Package Studio バージョン EX8.3 で導入しました。  
`ely3dpdf` という名前のグローバルオブジェクトとその関数 `getApi` を介して API を利用することができます。関数: `getApi` を参照してください。

API はいくつかの JavaScript クラスに依存し、かつ実装されています。これらのクラスとそれぞれの関数については以下の節で説明します。

以下の用語を使用して、クラスと関数を説明します:

- **内容** - クラス/関数の説明
- **追加されたバージョン** - クラス/関数が API に導入されたときのバージョン
- **入力** - 関数の名前付き入力パラメーターとタイプの簡単な説明
- **戻り値** - 関数の戻り値と型の説明
- **例** - 関数を呼び出すサンプルコード

NOTE: サンプルコードでは、Adobe Acrobat の JavaScript エンジンとの互換性を保つために、`const` と `let` の代わりに `var` を使用しています。

## E.1. クラス: Ely3dpdf

|            |   |
|------------|---|
| 内容         | グローバルオブジェクト <code>ely3dpdf</code> が TDP JS API にアクセスするためのクラスです。 |
| 追加されたバージョン | EX8.3   |

### E.1.1. 関数: `getApi`

|            |   |
|------------|---|
| 内容         | TDP JS API を返します。                                   |
| 追加されたバージョン | EX8.3   |
| 戻り値        | TDP JS API ( <code>ElyTdpApi</code> ) のインスタンスを返します。 |

例

```
var api = ely3dpdf.getApi();
```

## E.2. クラス: ElyTdpApi

|            |                        |
|------------|------------------------|
| 内容         | TDP 汎用 API を実装するクラスです。 |
| 追加されたバージョン | EX8.3                  |

### E.2.1. 関数: getField

|            |   |      |  |
|------------|---|------|--|
| 内容         | テンプレートの名前付きフィールド (widget のプレースホルダー) を返します。  |      |  |
| 追加されたバージョン | EX8.3   |      |  |
| 入力         | <table><tr><td>name</td><td>テンプレート内のフィールドの名前を指定する <b>string</b>。(テンプレート widget のプレースホルダー名と同じ)</td></tr></table> | name | テンプレート内のフィールドの名前を指定する <b>string</b> 。(テンプレート widget のプレースホルダー名と同じ) |
| name       | テンプレート内のフィールドの名前を指定する <b>string</b> 。(テンプレート widget のプレースホルダー名と同じ)  |      |  |
| 戻り値        | 入力名に一致するフィールドオブジェクトを返します。名前に一致するフィールドがない場合は <b>null</b> を返します。  |      |  |

#### 例

```
var api = ely3dpdf.getApi();
var field = api.getField('3D View widget'); ①

console.println('The ' + field.name + ' field is located on Page ' field.page);
```

① 返された **field** オブジェクトとそのプロパティについては、[Adobe Acrobat SDK ドキュメント](#)を参照してください。

### E.2.2. 関数: getWidget

|            |  |      |   |
|------------|--|------|---|
| 内容         | 名前付き widget を返します。   |      |   |
| 追加されたバージョン | EX8.3  |      |   |
| 入力         | <table><tr><td>name</td><td>テンプレート内の widget の名前を指定する <b>string</b>。(テンプレート widget のプレースホルダー名と同じ)</td></tr></table> | name | テンプレート内の widget の名前を指定する <b>string</b> 。(テンプレート widget のプレースホルダー名と同じ) |
| name       | テンプレート内の widget の名前を指定する <b>string</b> 。(テンプレート widget のプレースホルダー名と同じ)  |      |   |
| 戻り値        | 入力名に一致する widget ( <b>ElyTdpWidget</b> )を返します。名前に一致する widget がない場合は <b>null</b> を返します。                              |      |   |

## 例

```
var api = ely3dpdf.getApi();
var widget = api.getWidget('3D View widget');
```

## E.3. クラス: ElyTdpWidget

|            |                             |
|------------|-----------------------------|
| 内容         | ウィジェット間で共通の API を実装するクラスです。 |
| 追加されたバージョン | EX8.3                       |

### E.3.1. 関数: getType

|            |  |
|------------|--|
| 内容         | widget のタイプを返します。                          |
| 追加されたバージョン | EX8.3                                      |
| 戻り値        | widget のタイプ名を <code>string</code> として返します。 |

## 例

```
var api = ely3dpdf.getApi();
var widget = api.getWidget('3D View widget');
var type = widget.getType();

if (type === 'ElyTdp3dViewWidget') {
    // Do something with a 3D View widget
}
```

## E.4. クラス: ElyTdp3dViewWidget

NOTE: ElyTdp3dViewWidget のインスタンスが正しく動作するためには、対応する 3D ビューがアクティブである必要があります。

|            |   |
|------------|---|
| 内容         | 3D View widget API を実装するクラスです。 <a href="#">ElyTdpWidget</a> を拡張します。 |
| 追加されたバージョン | EX8.3   |

### E.4.1. 関数: reset

|            |  |
|------------|--|
| 内容         | 3D View widget をリセットします。<br>(part, pmi, view, render mode) |
| 追加されたバージョン | EX8.3  |

#### 例

```
var api = ely3dpdf.getApi();
var widget = api.getWidget('3D View widget');
// ...
widget.reset();
```

### E.4.2. 関数: fitAll

|            |  |
|------------|--|
| 内容         | 3D View widget のビューポートにすべてのパートをフィットします。<br>(選択した保存済みビューに表示しているすべてのパート) |
| 追加されたバージョン | EX8.3  |

#### 例

```
var api = ely3dpdf.getApi();
var widget = api.getWidget('3D View widget');
// ...
widget.fitAll();
```

### E.4.3. 関数: fitSelection

|            |   |
|------------|---|
| 内容         | 3D View widget のビューポートに選択したパートおよび PMI をフィットします。 |
| 追加されたバージョン | EX8.3   |

#### 例

```
var api = ely3dpdf.getApi();
var widget = api.getWidget('3D View widget');
// ...
widget.fitSelection();
```



### E.4.4. 関数: clearSelection

|            |   |
|------------|---|
| 内容         | 3D View widget でのパートおよび PMI の選択状態をクリアします。 |
| 追加されたバージョン | EX8.3                                     |

#### 例

```
var api = ely3dpdf.getApi();
var widget = api.getWidget('3D View widget');
// ...
widget.clearSelection();
```

### E.4.5. 関数: getRenderModes

|            |  |
|------------|--|
| 内容         | 3D View widget の有効なレンダリングモードを取得します。                  |
| 追加されたバージョン | EX8.3  |
| 戻り値        | 有効なレンダリングモードの一覧を <b>string</b> の <b>array</b> で返します。 |

#### 例

```
var api = ely3dpdf.getApi();
var widget = api.getWidget('3D View widget');
// ...
var modes = widget.getRenderModes(); ①
modes.forEach(mode => {
  console.println(mode);
});
```

① 有効なレンダリングモードについては、[関数: setRenderMode](#)以下を参照してください。

## E.4.6. 関数: setRenderMode

|            |      |   |
|------------|------|---|
| 内容         |      | 3D View widget のレンダリングモードを設定します。  |
| 追加されたバージョン |      | EX8.3   |
| 入力         | name | <p>レンダリングモード名を指定する <b>string</b> は有効な名前にしてください。有効でない場合、レンダリングモードは変更されません。</p> <p>有効なレンダリングモードの一覧については、<a href="#">Adobe Acrobat SDK ドキュメント</a>の Scene property にある <b>renderMode</b> を参照してください。</p> |

### 例

```
var api = ely3dpdf.getApi();
var widget = api.getWidget('3D View widget');
// ...
widget.setRenderMode('illustration');
```

## E.4.7. 関数: resetRenderMode

|            |  |  |
|------------|--|--|
| 内容         |  | 3D View widget のレンダリングモードを初期値にリセットします。 |
| 追加されたバージョン |  | EX8.3                                  |

### 例

```
var api = ely3dpdf.getApi();
var widget = api.getWidget('3D View widget');
// ...
widget.resetRenderMode();
```

### E.4.8. 関数: isValidRenderMode

|            |   |                                 |
|------------|---|---------------------------------|
| 内容         | 入力された名前が有効なレンダリングモード名かどうかを確認します。                                  |                                 |
| 追加されたバージョン | EX8.3   |                                 |
| 入力         | name  | レンダリングモード名を指定する <b>string</b> 。 |
| 戻り値        | 入力名が有効なレンダリングモードの場合は <b>true</b> を返し、それ以外の場合は <b>false</b> を返します。 |                                 |

#### 例

```
var api = ely3dpdf.getApi();
var widget = api.getWidget('3D View widget');
// ...
if (widget.isValidRenderMode('bounding box')) {
    // Do something
};
```

### E.4.9. 関数: getViews

|            |  |  |
|------------|--|--|
| 内容         | 3D View widget のビュー名一覧を返します。                   |  |
| 追加されたバージョン | EX8.3  |  |
| 戻り値        | ビュー名の一覧を <b>string</b> の <b>array</b> として返します。 |  |

#### 例

```
var api = ely3dpdf.getApi();
var widget = api.getWidget('3D View widget');
// ...
var views = widget.getViews();
views.forEach(view => {
    console.println(view);
});
```

## E.4.10. 関数: setView

|            |         |   |
|------------|---------|---|
| 内容         |         | 3D View widget のビューを設定します。  |
| 追加されたバージョン |         | EX8.3   |
| 入力         | name    | ビュー名を指定する <b>string</b> は有効な名前にしてください。有効でない場合、3D ビューは変更されません。   |
|            | animate | 3D View widget で新しいビューへの遷移をアニメーション化するか ( <b>true</b> ) 否か ( <b>false</b> 、デフォルト) を指定する <b>Boolean</b> 。 |

### 例

```
var api = ely3dpdf.getApi();
var widget = api.getWidget('3D View widget');
// ...
widget.setView('MBD-001D', true);
```

## E.4.11. 関数: resetView

|            |         |   |
|------------|---------|---|
| 内容         |         | 3D View widget のビューを初期ビューにリセットします。  |
| 追加されたバージョン |         | EX8.3   |
| 入力         | animate | 3D View widget が初期ビューへの遷移をアニメーション化する必要があるか ( <b>true</b> ) 否か ( <b>false</b> 、デフォルト) を指定する <b>Boolean</b> 。 |

### 例

```
var api = ely3dpdf.getApi();
var widget = api.getWidget('3D View widget');
// ...
widget.resetView();
```

### E.4.12. 関数: isValidView

|            |  |                           |
|------------|--|---------------------------|
| 内容         | 名前が有効なビュー名かどうかを確認します。  |                           |
| 追加されたバージョン | EX8.3  |                           |
| 入力         | name   | ビュー名を指定する <b>string</b> 。 |
| 戻り値        | 入力された名前が有効なビュー名であれば <b>true</b> を、それ以外の場合は <b>false</b> を返します。 |                           |

#### 例

```
var api = ely3dpdf.getApi();
var widget = api.getWidget('3D View widget');
// ...
if (widget.isValidView('MBD-001D')) {
    // Do something
};
```

### E.4.13. 関数: getShowPmi

|            |  |  |
|------------|--|--|
| 内容         | 3D View widget の PMI の表示/非表示を示す値を取得します。  |  |
| 追加されたバージョン | EX8.3  |  |
| 戻り値        | 3D View widget の現在の設定が PMI を表示している場合は <b>true</b> を、そうでない場合は <b>false</b> を返します。 |  |

#### 例

```
var api = ely3dpdf.getApi();
var widget = api.getWidget('3D View widget');
// ...
if (widget.getShowPmi()) {
    console.println('PMIs are shown!');
}
```

### E.4.14. 関数: setShowPmi

|            |                    |  |
|------------|--------------------|--|
| 内容         | PMI の表示/非表示を設定します。 |  |
| 追加されたバージョン | EX8.3              |  |
| 入力         | value              | PMI を表示するか ( <b>true</b> )、非表示にするか ( <b>false</b> ) を指定する <b>Boolean</b> 。 |

#### 例

```
var api = ely3dpdf.getApi();
var widget = api.getWidget('3D View widget');
// ...
widget.setShowPmi(false)
```

## E.5. クラス: ElyTdpTableWidget

|            |   |
|------------|---|
| 内容         | Table widget API を実装するクラス。 <a href="#">ElyTdpWidget</a> を拡張します。 |
| 追加されたバージョン | EX8.3.2   |

### E.5.1. 関数: clearFiltering

|            |                              |
|------------|------------------------------|
| 内容         | Table widget のフィルタリングを解除します。 |
| 追加されたバージョン | EX8.3.2                      |

#### 例

```
var api = ely3dpdf.getApi();
var widget = api.getWidget('PT_PmiTable');
// ...
widget.clearFiltering();
```

### E.5.2. 関数: clearSelection

|            |                           |
|------------|---------------------------|
| 内容         | Table widget の項目選択を解除します。 |
| 追加されたバージョン | EX8.3.2                   |

## 例

```
var api = ely3dpdf.getApi();
var widget = api.getWidget('PT_PmiTable');
// ...
widget.clearSelection();
```

## E.5.3. 関数: getData

|            |          |  |
|------------|----------|--|
| 内容         |          | Table widget のデータ(内容)を返します。関数: <a href="#">getHeader</a> を参照してください。  |
| 追加されたバージョン |          | EX8.3.2  |
| 入力         | filtered | オプションの <b>Boolean</b> でデータをフィルタリングして返すか ( <b>true</b> )、フィルタリングしないで返すか ( <b>false</b> がデフォルト) を指定します。<br>テーブルをフィルタリングしていない場合、filtered を <b>true</b> に設定しても、フィルタリングしていないデータを返します。 |
| 戻り値        |          | <b>文字列の配列</b> を格納した <b>配列</b> を返します。各内部配列はテーブルの行を表し、内部配列の要素はその行のセルの内容を表します。  |

## 例

```
var api = ely3dpdf.getApi();
var widget = api.getWidget('BT_BomTable');
// ...
var data = widget.getData(); ①
var numRows = data.length;
for (var row = 0; row < numRows; row++) {
    var rowData = data[row];
    var numCols = rowData.length;
    for (var col = 0; col < numCols; col++) {
        var cellData = rowData[col];
        // ...
    }
}
// ...
```

① データをフィルタリングするには、「true」を入力します。

### E.5.4. 関数: getHeader

|            |   |
|------------|---|
| 内容         | Table widget のヘッダーラベルを返します。関数: <a href="#">getData</a> を参照してください。 |
| 追加されたバージョン | EX8.3.2   |
| 戻り値        | ヘッダーラベルの文字列の配列を返します。(例: TDP の表のヘッダーに表示するもの)                       |

#### 例

```
var api = ely3dpdf.getApi();
var widget = api.getWidget('BT_BomTable');
// ...
var header = widget.getHeader();
var numCols = header.length;
for (var col = 0; col < numCols; col++) {
    var label = header[col];
    // ...
}
// ...
```

### E.5.5. 関数: reset

|            |  |
|------------|--|
| 内容         | Table widget のフィルタリング、選択、ソートの状態をリセットします。 |
| 追加されたバージョン | EX8.3.2                                  |

#### 例

```
var api = ely3dpdf.getApi();
var widget = api.getWidget('BT_BomTable');
// ...
widget.reset();
```



### E.5.6. 関数: resetSorting

|            |                                   |
|------------|-----------------------------------|
| 内容         | Table widget のソートを元のソート順にリセットします。 |
| 追加されたバージョン | EX8.3.2                           |

#### 例

```
var api = ely3dpdf.getApi();
var widget = api.getWidget('PT_PmiTable');
// ...
widget.resetSorting();
```

## E.6. Class: ElyTdpTextWidget

|            |  |
|------------|--|
| 内容         | Text widget API を実装するクラス。 <a href="#">ElyTdpWidget</a> を拡張します。 |
| 追加されたバージョン | EX9.1.0  |

### E.6.1. Function: getReadOnly

|            |   |
|------------|---|
| 内容         | Text widget が読み取り専用か書き込み可能かを返します。<br><a href="#">Function: setReadOnly</a> も参照してください。 |
| 追加されたバージョン | EX9.1.0   |
| 戻り値        | Text widget が読み取り専用の場合は "true" を返します。それ以外の場合 (書き込み可能) は <b>false</b> を返します。           |

#### 例

```
var api = ely3dpdf.getApi();
var widget = api.getWidget('TF_PartName');
// ...
widget.setReadOnly(false);
assert(widget.getReadOnly() === false);
// ...
```

## E.6.2. Function: setReadOnly

|            |  |  |
|------------|--|--|
| 内容         | Text widget が読み取り専用か書き込み可能かを設定します。<br><a href="#">Function: getReadOnly</a> も参照してください。 |  |
| 追加されたバージョン | EX9.1.0  |  |
| 入力         | 値  | Text widget が読み取り専用 (true) か書き込み可能 (false) かを示す <b>Boolean</b> を指定します。 |

### 例

```
var api = ely3dpdf.getApi();
var widget = api.getWidget('TF_PartName');
// ...
widget.setReadOnly(true);
assert(widget.getReadOnly() === true);
// ...
```

## E.6.3. Function: toggleReadOnly

|            |  |  |
|------------|--|--|
| 内容         | Text widget が読み取り専用か書き込み可能かを切り替えて新しい値を返します。読み取り専用であった場合は書き込み可能に、書き込み可能であった場合は読み取り専用になります。 <a href="#">Function: getReadOnly</a> と <a href="#">Function: setReadOnly</a> を参照してください。 |  |
| 追加されたバージョン | EX9.1.0  |  |
| 戻り値        | 切り替えた後で Text widget が読み取り専用になった場合は "true" を返します。書き込み可能になった場合は <b>false</b> を返します。  |  |

### 例

```
var api = ely3dpdf.getApi();
var widget = api.getWidget('TF_PartName');
// ...
assert(widget.getReadOnly() === !widget.toggleReadOnly());
// ...
```

## 付録 F: CAD Validator レポートテンプレートの新しいフォーマットへのマイグレーション

本付録では、既存のカスタマイズされた CAD Validator レポートテンプレートから Data Package Studio (DPS) EX9.0 で導入された新しいフォーマットへのマイグレーション方法について説明します。Difference Summary table 内で Failed/Passed assessment を非表示にするなど、CAD Validator widget をカスタマイズする DPS の機能を活用したい場合は、新しいフォーマットへのマイグレーションが必要となります（下図参照）。

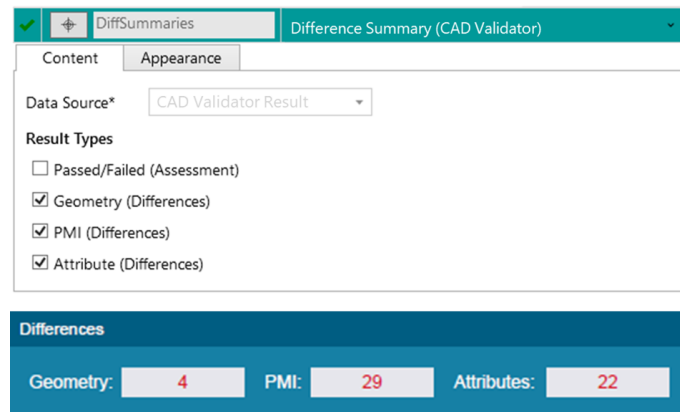


図 91. Difference Summary table 内で Failed/Passed assessment を非表示にした例（DPS option（上））、Failed/Passed assessment なしで生成されたテーブル（下）

ただし、widget のカスタマイズをせずに TDP を生成するのみの場合は、新しいフォーマットへのマイグレーションは必要ありません。3DxSUITE EX9.0 でも、EX8.3以前のDPSで作成したテンプレートをそのまま使用して TDP を生成することができます。

本付録では、新しいフォーマットで CAD Validator レポートテンプレートを作成する方法もご紹介します。



**Property Table** など、いくつかの widget はカスタマイズできません。ただし、テンプレート/formula から削除することによって widget を除外することができます。

最初に、マイグレーション手順を説明する際の前提となる **CAD Validator widget タイプ** について説明します。これは EX9.0 で Data Package Studio が新たにサポートした widget です。この説明の後、CAD Validator レポートテンプレートを新しいフォーマットへマイグレーションする手順を説明します。

1. [CAD Validator 3D View Widget に関連するプレースホルダーの結合](#)
2. [CAD Validator Widget プレースホルダーの名称変更](#)
3. [Data Package Studio で Formula を作成する](#)

## F.1. CAD Validator Widget タイプ

CAD Validator widget は11種類あります。以下に示す旧タイプの CAD Validator レポートテンプレートにおいて、これらの widget に対応するプレースホルダーについても簡単に説明します。(元のテンプレートにあるページについての説明も含まれます。)

### F.1.1. 3D View (CAD Validator)

CAD Validator の 3D View widget は DPS 標準の 3D View widget とは異なります。1つのモデルの形状の代わりに、2つのモデルの形状およびその差異を可視化します。

下図のように、旧タイプの CAD Validator レポートテンプレートは1つの widget に対して4つのプレースホルダーを持っています。(すべてページ1にあります。)

- ELY\_CV\_3DController
- ELY\_CV\_3DHeader
- ELY\_CV\_ViewSource
- ELY\_CV\_ViewTarget

しかし、DPS EX9.0 で起動するのに必要なプレースホルダーは1つのみです。これは [マイグレーションの手順1](#) の理由でもあり、4つのプレースホルダーを1つに統合する必要があります。

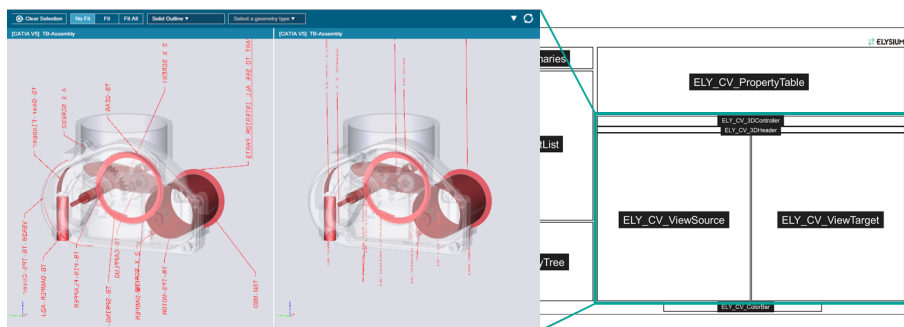


図 92. 3D View widget とプレースホルダーの対応関係 (テンプレート内で対応)

ツールバーに何を表示するか (表示/非表示) をカスタマイズすることができます (左から右へ)。

- Clear Selection button
- Fit Mode selector
- Render Mode selector
- Geometry Type Mode selector
- Action Menu (右端の下向き矢印)
  - Show PMI checkbox
  - Show Unselected Parts checkbox
  - Highlight Transparent Faces checkbox
- Refresh button

デフォルトではすべて表示されます。

### F.1.2. Assembly Tree (CAD Validator)

Assembly Tree は、"ELY\_CV\_AssemblyTree" という、ページ1 のプレースホルダーに対応しています（下図参照）。比較対象となる2つのモデルに関するアセンブリーツリーを示すもので、現在のところカスタマイズ可能ではありません。

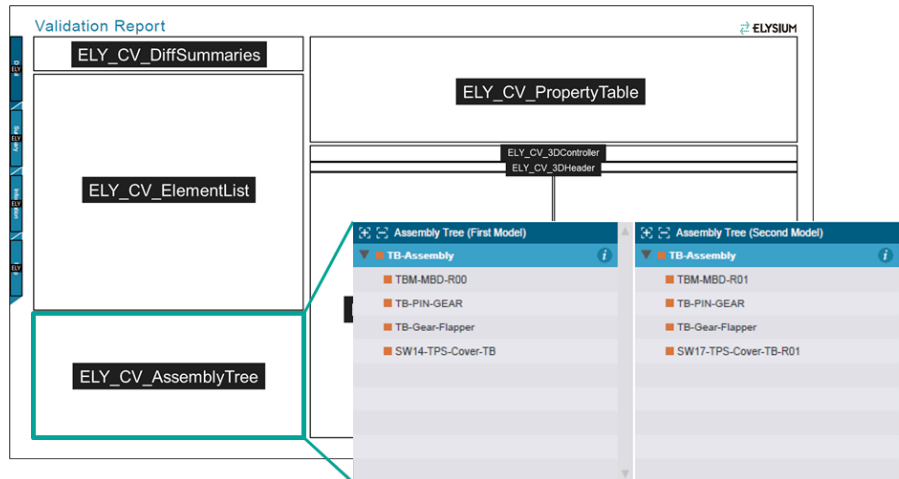


図 93. Assembly Tree widget とプレースホルダーの対応関係（テンプレート内で対応）

### F.1.3. CAD Information (CAD Validator)

CAD Information は "ELY\_CV\_CadInfo" という、ページ3 のプレースホルダーに対応しています（下図参照）。各モデルの CAD version など、比較対象となる2つのモデルに関する CAD information をリスト化するものです。現在のところカスタマイズ可能ではありません。

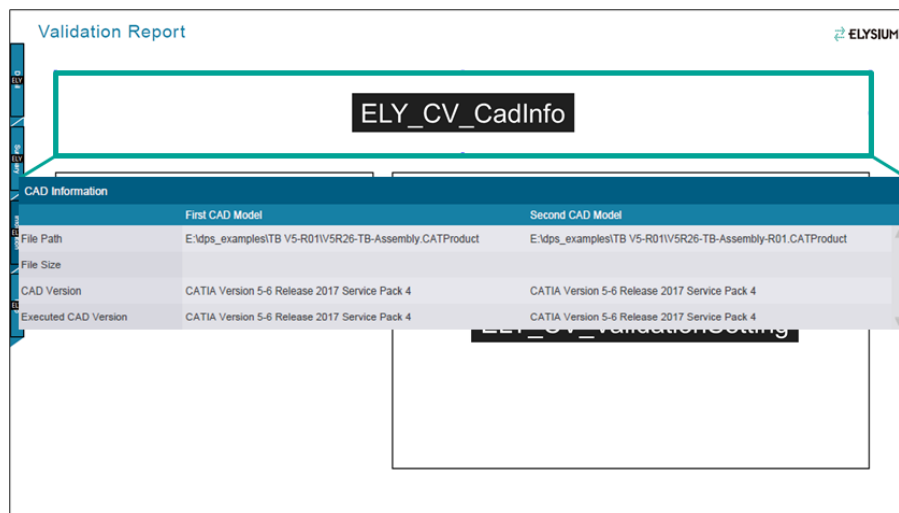


図 94. CAD Information widget とプレースホルダーの対応関係（テンプレート内で対応）

### F.1.4. Color Bar (CAD Validator)

Color Bar は "ELY\_CV\_ColorBar" という、ページ1 のプレースホルダーに対応しています（下図参照）。3D View widget で差異を可視化する際に、color coding scale を示します。DPS を介しては現

在のところカスタマイズ可能ではありませんが、オフ(0) / オン(1, デフォルト)を設定するために CAD Validator parameter '**PdfEmbedColorMap**' を使用することは可能です。

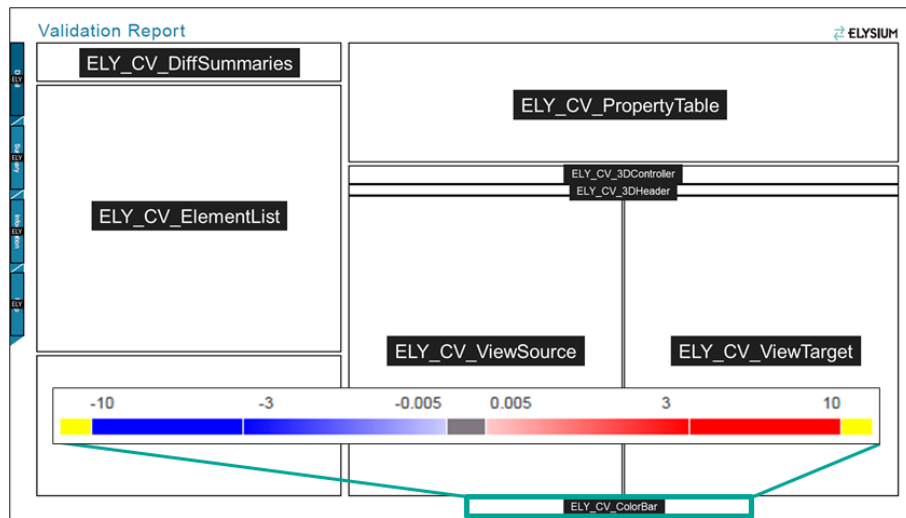


図 95. Color Bar widget とプレースホルダーの対応関係（テンプレート内で対応）

### F.1.5. Component Summary (CAD Validator)

Component Summary は "ELY\_CV\_ComponentSummary" という、ページ2 のプレースホルダーに対応しています（下図参照）。1つの component に対して2つのモデルを比較した結果の要約です。

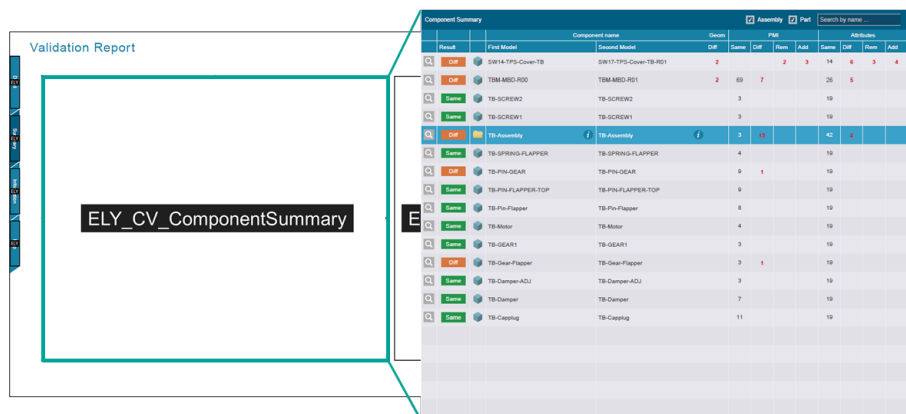


図 96. Component Summary widget とプレースホルダーの対応関係（テンプレート内で対応）

タイトルバーに何を表示するか（表示/非表示）をカスタマイズすることができます（左から右へ）。

- Assembly checkbox
- Part checkbox
- Search field

デフォルトではすべて表示されます。

### F.1.6. Detail List (CAD Validator)

Detail List は "ELY\_CV\_ElementList" という、ページ1 のプレースホルダーに対応しています（下図参照）。2つの観点から2つのモデルを比較した結果を示します。

1. Group List: 手動または自動で作成される結果のグループ
2. Element List: 要素およびそれぞれの結果

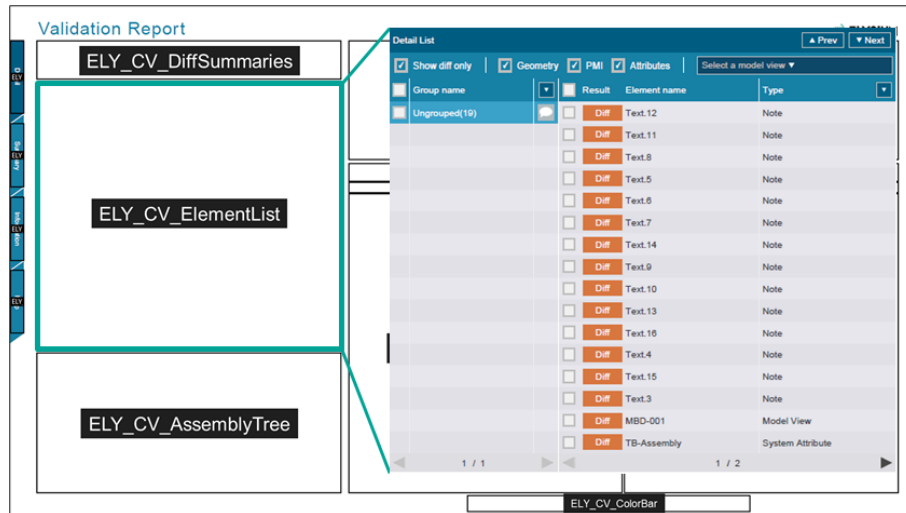


図 97. Detail List widget とプレースホルダーの対応関係（テンプレート内で対応）

タイトルバー・ツールバー・詳細リストに何を表示するか（表示/非表示）をカスタマイズすることができます（左から右へ）。

- Title bar:
  - Prev button
  - Next button
- Toolbar:
  - Show Diff Only checkbox
  - Geometry checkbox
  - PMI checkbox
  - Attributes checkbox
  - Model View selector
- Detail Lists:
  - Group List
    - Action Menu
      - Add Group action
      - Edit Group action
      - Remove Group action
  - Element List
    - Action Menu
      - Move to Group action
      - Copy to Group action

- Remove from Group action

デフォルトではすべて表示されます。

### F.1.7. Difference Summary (CAD Validator)

Difference Summary は "ELY\_CV\_DiffSummaries" という、ページ1 のプレースホルダーに対応しています（下図参照）。Failed/Passed の観点から2つのモデルを比較した結果の要約と、各カテゴリー (Geometry, PMI, Attributes) での差異の数を示します。

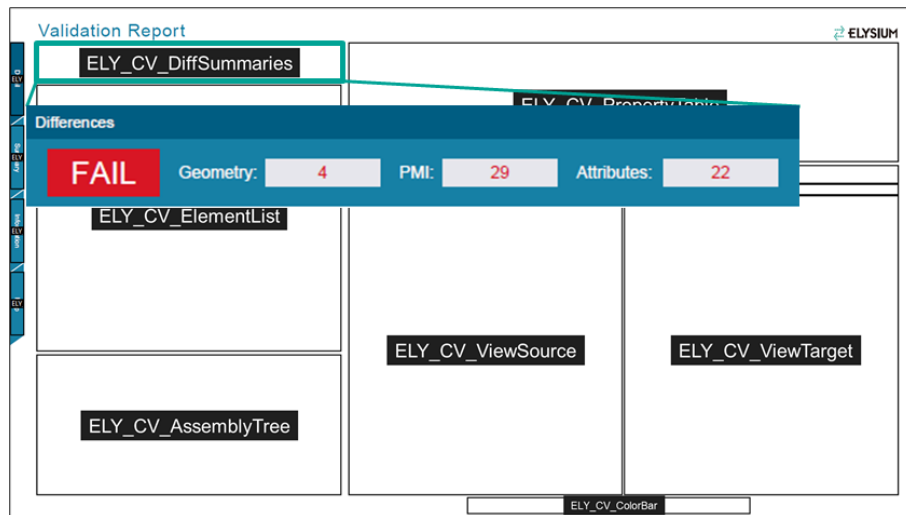


図 98. Difference Summary widget とプレースホルダーの対応関係（テンプレート内で対応）

Widget に何を表示するか（表示/非表示）をカスタマイズすることができます（左から右へ）。

- Passed/Failed assessment
- Geometry differences
- PMI differences
- Attribute differences

デフォルトではすべて表示されます。

### F.1.8. Execution Property (CAD Validator)

Execution Property は "ELY\_CV\_ExecProperty" という、ページ3 のプレースホルダーに対応しています（下図参照）。CAD Validator component (例: processed date/time, processing time) の実行に関連する property をリスト化するものです。現在のところカスタマイズ可能ではありません。



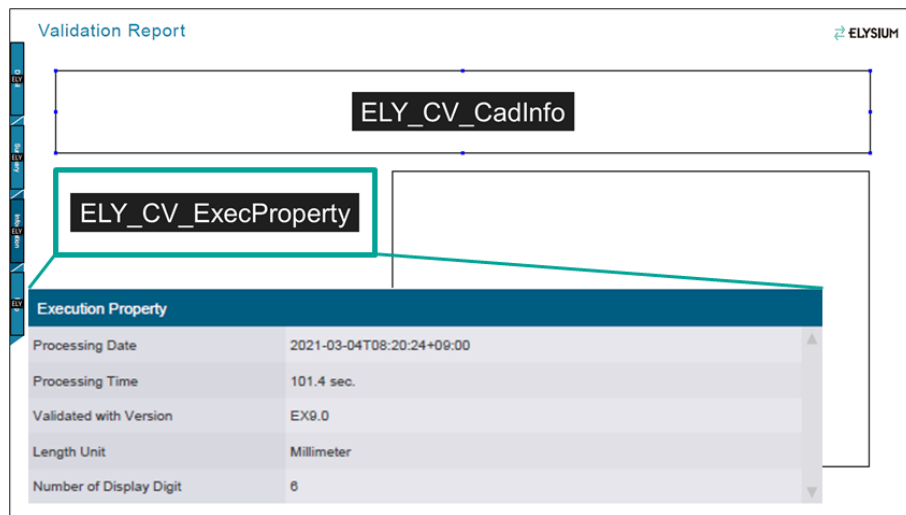


図 99. Execution Property widget とプレースホルダーの対応関係（テンプレート内で対応）

### F.1.9. Property Table (CAD Validator)

Property Table は "ELY\_CV\_PropertyTable" という、ページ1 のプレースホルダーに対応しています（下図参照）。Detail List 内で選択された差異に関連する Property をリスト化するもので、現在のところカスタマイズ可能ではありません。

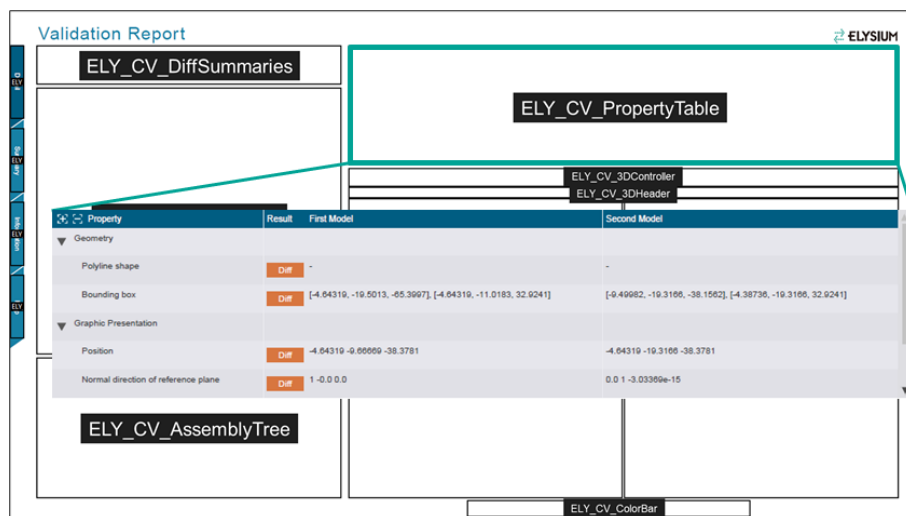


図 100. Property Table widget とプレースホルダーの対応関係（テンプレート内で対応）

### F.1.10. Type Summary (CAD Validator)

Type Summary は "ELY\_CV\_TypeSummary" という、ページ2 のプレースホルダーに対応しています（下図参照）。Assembly Tree widget で選択された component に対して、要素のタイプごとに差異を要約したものです。

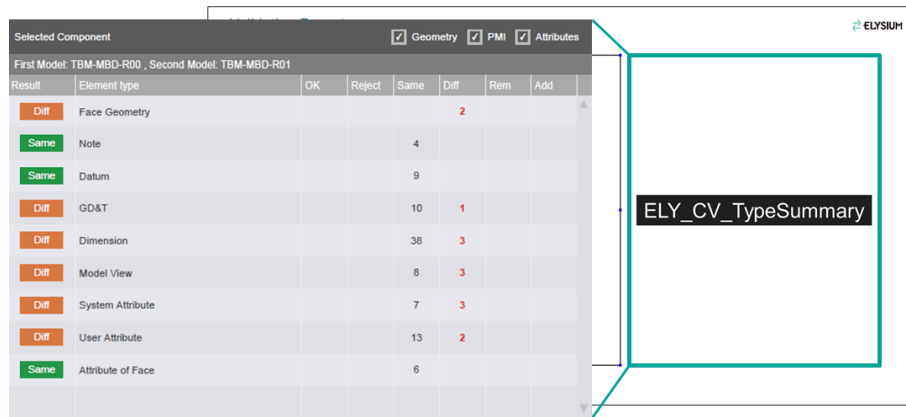


図 101. Type Summary widget とプレースホルダーの対応関係（テンプレート内で対応）

タイトルバーに何を表示するか（表示/非表示）をカスタマイズすることができます（左から右へ）。

- Geometry checkbox
- PMI checkbox
- Attributes checkbox

デフォルトではすべて表示されます。

### F.1.11. Validation Setting (CAD Validator)

Validation Setting は "ELY\_CV\_ValidationSetting" という、ページ3 のプレースホルダーに対応しています（下図参照）。2つのモデルを比較する際に使用した設定をリスト化するもので、現在のところカスタマイズ可能ではありません。

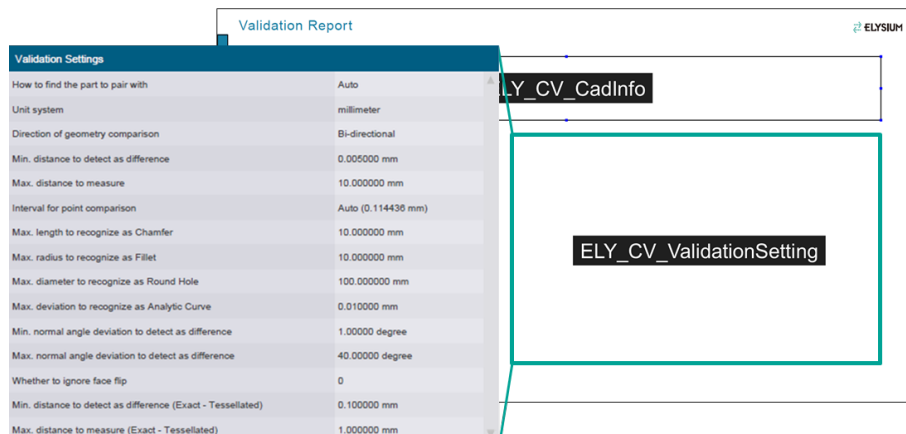


図 102. Validation Setting widget とプレースホルダーの対応関係（テンプレート内で対応）

## F.2. CAD Validator 3D View Widget に関連するプレースホルダーの結合

各種の CAD Validator widget がわかったところで、マイグレーションの工程を始めましょう。まず、**3D View (CAD Validator)** の4つのプレースホルダーを1つに結合します。

お好みのエディターでテンプレートを開き、編集可能な状態にします (Adobe Acrobat Pro の

"Prepare Form")。3つのプレースホルダーを削除します(任意の3つを削除、または4つ削除し1つ追加するなど)。最終的に、使用すると決めた1つのプレースホルダーを移動およびサイズ調整します(下図を参照してください)。

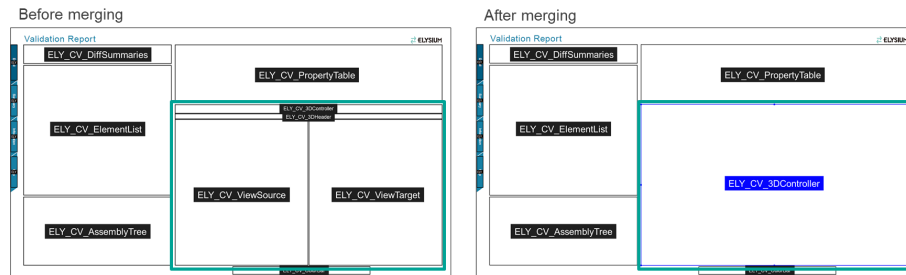


図 103. 3D View に関連する4つのプレースホルダーを1つに結合する例

テンプレートを保存します(元のテンプレートをバックアップとして保持したい場合は、新しい名前を付けてください)。

### F.3. CAD Validator Widget プレースホルダーの名称変更

マイグレーション工程の次のステップは、名前の競合リスクを排除するためにプレースホルダーの名称を変更することです。'ELY\_' で始まるプレースホルダーすべての名称変更が必要です。エリジオンではこれを、作成/生成した要素の接頭辞に使用しているためです。(予約済みの PDF フィールド名を参照してください)



プレースホルダーには、「'ELY\_'」で始まるものでなければ自由に名前を付けられません。

名称変更が必要なプレースホルダーには、少なくとも元のテンプレートが残っているプレースホルダーすべてが含まれます。カスタマイズしたテンプレートからボタンを削除していない限り、ナビゲーションペインの非表示ボタンの名称も変更する必要があります。名称変更後は、下図の例のような状態になるはずです。

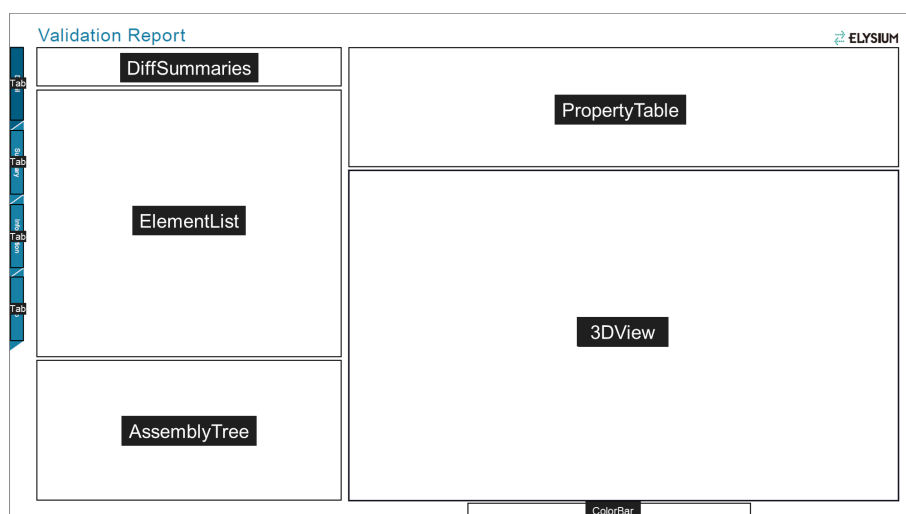


図 104. テンプレート内でプレースホルダーの名称変更をした例

テンプレートを保存します。

## F.4. Data Package Studio で Formula を作成する

マイグレーション工程の最後のステップは、formula を作成することです。CAD Validator レポートのカスタマイズ方法については、[CAD Validator レポートのカスタマイズ](#) を参照してください。

## 付録 G: TDP 前処理スクリプトの例

Listing 9. TDP 前処理スクリプトを使用して、ENF で *Bill of Characteristics* データを準備する例

```
# -*- encoding: UTF-8 -*-
module PdfEditor ①
  class CustomizedEnfPreProcess ②
    logger.info "Custom File ({__FILE__}): "

    def on_run_before(ee_session) ③
      ee_session.model.components.each do |compo| ④
        # Create mapping from BoC ID to referenced PMI
        id_to_ref = Hash.new
        compo.annotations.each do |anno| ⑤
          md = anno.name.match(/EREF\(([[:\d;]+])\)/) ⑥
          next if md.nil? || md.size != 2
          md[1].split(";").each do |id|
            id_to_ref[id] = anno ⑦
          end
        end

        # Flag BoC and set referenced PMI
        compo.annotations.each do |anno| ⑧
          md = anno.name.match(/EBOC([[:\d;]+])/) ⑨
          next if md.nil? || md.size != 2
          ref_anno = id_to_ref[md[1]] ⑩
          next if ref_anno.nil?
          anno.boc_tag = true ⑪
          anno.boc_ref = ref_anno ⑫
        end
      end
    end
  end
end
```

- ① スクリプトは、PdfEditor モジュール内で「存続」します。
- ② CustomizedEnfPreProcess クラスでもスクリプトは「存続」します。
- ③ 「on\_run\_before」メソッドは、スクリプトが定義されている場所です。
- ④ 各コンポーネント/パートについては、手順 5～12 を繰り返します。
- ⑤ コンポーネント/パートのアノテーションごとについては、手順 6～7 を繰り返します。
- ⑥ EREF(<number>[:<number>;...]) を含む名前を持つすべての PMI を検索します。ここで、BoC ID は number を参照します。
- ⑦ 発見した PMI に対応する BoC の ID をキーとしてハッシュマップに格納します。(次の検索を有効にします。)

- ⑧ コンポーネント/パートのアノテーションごとについては、手順 9～12 を繰り返します。
- ⑨ EBOC<number> を含む名前を持つすべてのアノテーション (BoC) を検索します。ここで、number は BoC の一意の ID です。
- ⑩ 手順 7 で格納しておいた対応する PMI を取得します。
- ⑪ boc\_tag プロパティを **true** に設定して、アノテーション (BoC) にタグを付けます。(これが「通常の」のアノテーションを BoC に変えるものです。)
- ⑫ boc\_ref プロパティによって BoC を その PMI に接続します。(これにより、PMI 表で BoC が選択されている場合、3D ビューで PMI のハイライト/ズームも有効になります。)

## 付録 H: 既知の問題

表 12. EX9.1 で新たに報告された問題

| 事象   | 内容  |
|--|---|
| Adobe Acrobat (ソフトウェアレンダリング) における 3D PDF レポート (CAD Validator と PDQ を含む) のパフォーマンス | <p>ソフトウェアレンダリングは、DirectX レンダリングに比較すると応答が大幅に遅くなることがあります。場合によってはレポートが応答しなくなることもあります。</p> <p><u>回避策</u>：<br/>Acrobat 32bit: DirectX レンダリングを使用してください。<br/>Acrobat 64bit: 回避策はありません。</p>   |
| Adobe Acrobat (ソフトウェアレンダリング) の要素 (面など) の色が正しくない                                  | <p>要素の色を変更するには DirectX レンダリングが必要です。そのためソフトウェアレンダリングを使用した場合、PMI 関連要素のハイライト表示やビュー内の要素の色変更などは想定通りに描画されません。要素はハイライトや指定された色ではなく、元の色で描画されます。</p> <p><u>回避策</u>：<br/>Acrobat 32bit: DirectX レンダリングを使用してください。<br/>Acrobat 64bit: 回避策はありません。</p> |
| 生成された TDP を開くと Adobe Acrobat が不正終了する   | <p>直近のバージョンでは、生成された TDP を開くと Adobe Acrobat が不正終了することがあります。</p> <p><u>回避策</u>：<br/>EX9.1 で修正しました。EX9.1 より前に生成されたレポートを Adobe Acrobat で開いたら、次に開いた時に不正終了が起きることを防止するため、TDP を保存してください。</p>  |
| アイコンの背景色が正しくない   | <p>アイコンの背景色が透明ではない色になります。アイコンの背景色は、デフォルトの CAD Validator レポートテンプレートの配色と適合するように一時的に変更されます。</p> <p><u>回避策</u>：回避策はありません。</p>   |
| PDQ レポートに形状が含まれていない  | <p>一部の例外的なケースにおいて、PDQ レポートのエラーにより元モデルの形状の一部が欠落している場合があります。</p> <p><u>回避策</u>：回避策はありません。</p>   |
| ビューの表示が正しくない   | <p>モデルが非常に複雑な場合、一部の要素の可視性が想定通りにレンダリングされないことがあります。</p> <p><u>回避策</u>：回避策はありません。</p>  |

| 事象                              | 内容  |
|---------------------------------|---|
| テセレーション精度が低下する                  | <p>一部の場合において、テセレーションの精度が低くなる場合があります。</p> <p><u>回避策</u>：回避策はありません。</p>   |
| メモリが不足する                        | <p>モデルのサイズが非常に大きく何千ものインスタンスがある場合、Adobe Acrobat が TDP を開くときにメモリを使いきってしまうため、一部もしくはすべての widget が描画されない可能性があります。</p> <p><u>回避策</u>：回避策はありません。</p>   |
| 要素の色が正しくない                      | <p>一部の例外的なケースにおいて、一部の要素の色が想定通りに描画されないことがあります。</p> <p><u>回避策</u>：回避策はありません。</p>  |
| テキストフィールドの文字が正しく表示されない          | <p>テキストファイルのコンテンツをテキストフィールドに表示する場合、一部の文字が「□」と表示されることがあります。</p> <p><u>回避策</u>：<br/>テンプレートに正しいフォントが埋め込まれていること、またはレポートを開くコンピューターに正しいフォントがインストールされていることを確認してください。</p>   |
| 他言語版 OS で TDP を開くと正しく表示されない     | <p>生成するときに使用した OS (およびレポート言語) 以外の言語で TDP を開いた場合、TDP が正しく表示されないことがあります。例えばドイツ語 OS で日本語の CAD Validator レポートを開くとこの問題が発生することがあります。</p> <p><u>回避策</u>：<br/>レポートが使用される言語以外の OS とレポート言語の組み合わせでは TDP を生成しないでください。</p> |
| HTMLレポートが正しく表示されない (SaveAsHtml) | <p>HTMLレポート (SaveAsHtml によって生成) を読み込む時に拡大・縮小を行うと、レポートが正しく表示されない場合があります。</p> <p><u>回避策</u>：<br/>HTMLの読み込み時に拡大・縮小を行わないでください。問題が起きた場合には、HTML レポートを再読み込みまたは更新してください。</p>  |



| 事象   | 内容   |
|--|--|
| CAD Validator の HTMLレポートの 3Dビューが動作しない (SaveAsHtml) | <p>特定の状況下、例えばテーブル内のある項目と3Dビュー内のある要素を選択した場合に、3Dビューが動作しなくなりレポートの上部にある赤いバナーに "TypeError" と表示されることがあります。</p> <p><u>回避策</u>：<br/>HTMLレポートを再読み込みまたは更新してください。</p> |

表 13. Data Package Studio の既知の問題

| 事象                                     | 内容   |
|--|--|
| 入力ファイルに ENF ファイルが含まれていない場合、TDP が生成されない | <p>3DxSUITEの Front-End において、少なくとも1つの ENF 入力ファイルを指定する必要がある、それが無い場合は TDP が生成されません。</p> <p><u>回避策</u>：<br/>フォーミュラで使用しない場合でも、ENFファイルを「ダミー」として指定します。</p>  |
| 不足しているフォントの有無を Adobe Acrobat が確認しない    | <p>最近のバージョンの Adobe Acrobat では不足しているフォントの確認、および不足している場合の警告の表示が行われなくなりました。このため、不足しているフォントがあっても当該フォントをダウンロードするか尋ねられることはありません。</p> <p><u>回避策</u>：不足しているフォントを手動で確認してダウンロードしてください。<a href="#">不足しているフォントを PDF に追加</a> を参照してください。</p> |
| TDP を保存するとファイルサイズが増加する                 | <p>Adobe Acrobat の既知の問題で、保存するたびに TDP のファイルサイズが増加します。Adobe Acrobat が変更を追跡するためのデータを追加していることに起因しています。</p> <p><u>回避策</u>：<br/>メニューから [File] &gt; [名前を付けて保存...] を選択して、TDP を保存してください。</p>  |

| 事象   | 内容   |
|--|--|
| PMI を選択する方法によってハイライトのされ方が異なる                     | <p>PMI Table widget から 3D View widget への Trigger/Action 接続が存在するが、逆の方向には存在しない場合に問題が発生します。</p> <p>3D ビューから PMI を選択した場合と PMI テーブルから PMI を選択した場合でハイライトのされ方が異なります。</p> <p><u>回避策</u>：<br/>3D View widget から PMI Table widget への Trigger/Action 接続を追加します。</p>        |
| アジア言語の文字を含むテキストのフォーマットが正しく表示されない                 | <p>TDP で使用されているフォントが原因です。この影響でアジア言語の文字を含むテキストが望ましい位置よりわずかに下に表示されます。</p> <p><u>回避策</u>：回避策はありません。</p>   |
| formula のファイルパスに ASCII 以外の文字が含まれる場合、処理が正しく実行されない | <p>3DxSUITE コンポーネントでエラーが発生し、TDP を作成することができません。このエラーは formula のファイル名もしくはパスに含まれるフォルダ名に ASCII 以外の文字がある場合に発生します。</p> <p>パスの区切り文字 (例えば '\ ' や '¥') については ASCII 以外の文字であってもエラーになることはありません。</p> <p><u>回避策</u>：<br/>フォルダーおよび formula ファイルの名前には ASCII 文字のみを使用してください。</p> |
| ビュー内で PMI が適切にレンダリングされない                         | <p>クロスセクション(断面)と PMI の基準面が同一平面上にあるなどの例外的なケースでは、ビュー内で PMI が正しく描画されないことがあります。</p> <p><u>回避策</u>：<br/>クロスセクション(断面)と PMI の基準面をずらします。</p>   |
| ページ回転が正しく処理されない場合がある                             | <p>コンテンツが正しく表示されなかったり欠落したりします。</p> <p><u>回避策</u>：ページを回転させないでください。</p>  |
| 複数の 3D ビューがある場合に 3D 背景色が正しく処理されない                | <p>複数の 3D ビューを使用すると、3D 背景色が想定通りに表示されません。formula (JSON) の最後に指定されている 3D ビューの背景色がすべての 3D ビューで使用されます。</p> <p><u>回避策</u>：回避策はありません。</p>   |

| 事象   | 内容  |
|--|---|
| 特定の形式の CSV ファイルが正しく処理されない                  | <p>CSV ファイルの内容が正しく表示されなかったり、欠落したりします。</p> <p><u>回避策</u>：以下の形式の CSV ファイルを使用します。</p> <ul style="list-style-type: none"> <li>区切り文字：コンマ (,)</li> <li>ヘッダー行：必須 (CSV の最初の行は見出し行として解釈されます。)</li> </ul>   |
| 一部の場合において画面に対して注釈を同じ向きに保つ機能が動作しない          | <ol style="list-style-type: none"> <li>注釈がテキスト以外の文字を含んでいると判断される場合には通常の注釈として表示されます。<br/>(例: 注釈に引き出し線が含まれている場合)</li> <li>View Carousel のサムネイル内では、注釈が表示されません。</li> </ol> <p><u>回避策</u>：回避策はありません。</p>  |
| PDF フィールドのフォーマットを変更するとコンテンツが正しく表示されない      | <p>コンテンツが正しく表示されなかったり欠落したりします。</p> <p><u>回避策</u>：<br/>テンプレートのフィールドのフォーマットを変更せず、既定の "None" フォーマットのままにしてください。 "Date" フィールドには "date" フォーマットが指定されるため、本制限事項に該当することに注意してください。</p>   |
| サインフィールドおよびバーコードフィールドが正しく表示されない            | <p>コンテンツが正しく表示されなかったり欠落したりします。</p> <p><u>回避策</u>：以下の方法でサインフィールドおよびバーコードフィールドを追加してください。</p> <ul style="list-style-type: none"> <li>TDP を作成する前の時点で TDP テンプレートに追加する</li> <li>TDP を作成した後に追加する</li> </ul> <p>いずれの場合も作成された TDP を開いてからフィールドを含めて保存した上で、Adobe Acrobat を再起動する必要があります。</p> |
| 一部の widget が Trigger もしくは Action として使用できない | <p>Trigger もしくは Action として選択できない widget があります。それらは trigger widget コンボボックスや action widget コンボボックスに表示されません。</p> <p>設定可能な widget、Trigger、Action の組合せについては「<a href="#">Trigger と Action の組合せ</a>」を参照してください。</p> <p><u>回避策</u>：回避策はありません。</p>  |

| 事象                                       | 内容   |
|--|--|
| CAD Validatorにおいて、グループを選択したとき、応答に時間がかかる  | <p>デフォルトのグループ「グループ未登録」など差異の多いグループを選択すると、グループ内の差異数によっては応答がとても遅くなる場合があります。</p> <p><u>回避策</u>：<br/>大きなグループを複数の小さなグループに分割します。<br/>CAD Validator の事前処理スクリプトでグループ作成ロジックを適用することにより実行できます。</p> |
| 一部のビューが正しくフィットしません                       | <p>一部の例外的なビューでは、正しくフィットしない場合があります。</p> <p><u>回避策</u>：回避策はありません。</p>  |
| クロスセクション (断面) の色がビューサムネイルに表示されません        | <p>ビューカラーセルのサムネイルでは、クロスセクション (断面) の色が表示されません。</p> <p><u>回避策</u>：回避策はありません。</p>   |
| 「分解」ビューのサムネイルが正しく表示されない                  | <p>「分解」ビューのサムネイルがビューカラーセルに正しく表示されない場合があります。</p> <p><u>回避策</u>：回避策はありません。</p>   |
| プレースホルダーの名前にピリオド「.」を含む場合、widget は作成されません | <p>widget のプレースホルダーの名前にピリオドを含む場合、widget は作成されず、生成された TDP に表示されません。<br/>例：「Text.1」</p> <p><u>回避策</u>：widgetのプレースホルダー名にピリオドを含めないように変更します。例：「Text_1」</p>                                  |

本コンテンツに関わる著作権は株式会社エリジオンもしくは原権利者に帰属しています。著作権者の承諾なしに無断で改変、複製、転載、再配布、転送、公衆送信、販売、貸与などの行為をすることは禁じられています。