JavaScript によるモデルの操作

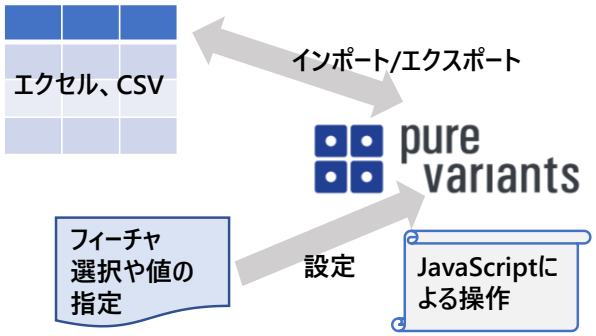
目次

■ はじめに

pure::variants には API による JavaScript インターフェースが提供されており、JavaScript を使用して、外部に定義した一連のフィーチャを pure::variants のフィーチャモデル（`*.xfm`）としてインポートすることや、バリアント定義モデル（`*.vdm`）上の選択内容を変更することができます。

この資料ではバリアント定義モデルの設定を、外部のファイルに定義したフィーチャの選択情報や属性値で変更する例を通じて、pure::variants の API による JavaScript インターフェースについて紹介します。外部のファイルには、JavaScript との相性の良さから JSON 形式を想定しています。また JSON 形式であれば、様々なツールで定義された情報から変換することも容易です。

JavaScript インターフェースの詳細については、「pure::variants JavaScript Extensibility Guide」をご覧ください。このドキュメントは pure::variants SDK をインストールしている場合、メニューの「ヘルプ」から見ることができます。



● **pure::variants SDK のインストール**

SDK は pure::variants インストール時に選択してインストールできますが、インストールされていない場合は、pure::variants のメニューから ヘルプ > 新規ソフトウェアのインストール で更新サイト（http://www.pure-systems.com/pvce-update/）を指定し、SDK を選択することでインストールできます[1]。

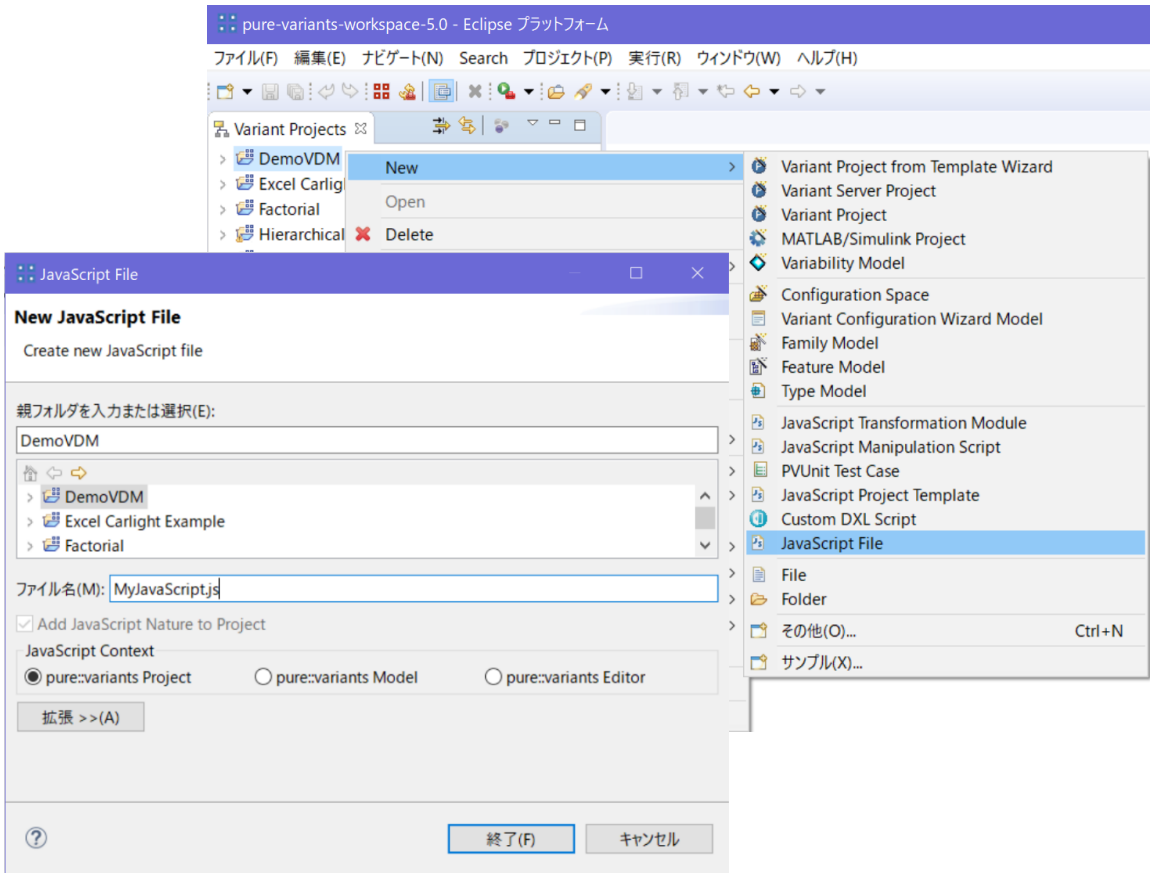
● **サンプルプロジェクトのダウンロード**

https://www.fuji-setsu.co.jp/files/DemoVDM.zip

■ モデル操作の手順

1．JavaScript の作成

pure::variantsのプロジェクト内に JavaScript のスクリプトを作成するには、プロジェクトツリーから New > JavaScript File をクリックします。New JavaScript File ダイアログボックスの ファイル名 欄で JavaScriptファイルの名称を入れて終了し、ファイルの内容を入力できるようになります。



作成した MyJavaScript.js をエディタで開いて、スクリプトを入力できます。

[1] 別法として、Eclipse のメニューからファイル > 新規 > サンプル > Variant Management SDK を選択します。このカテゴリには「Extensibility Example Plugins」と「Extensibility Example Projects」がありますので、それぞれ選択してインストールできます。

スクリプトの例

```
/** Do some work on the model. E.g. get root element.
 * @param {IPVModel} model The pure::variants model.
 * @returns {IPElement} Any element
 */
function work(model) {
  // auto completion works for model.
  var element = model.getElementWithID(model.getElementsRootID());
  return element;
}

model = new PVModel();
var element = work(model);

// auto completion works for element.
var name = element.getName();
```

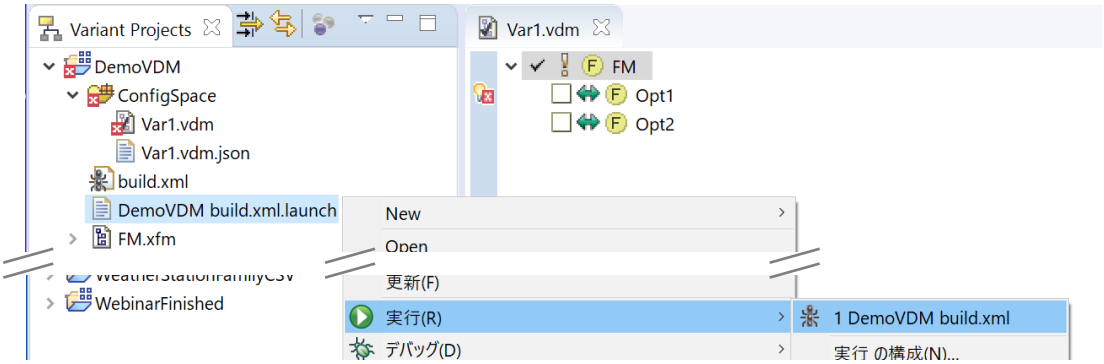
2．外部ファイルによるフィーチャの指定

JavaScript API による操作対象のフィーチャを JSON 形式で外部のファイルに記述して指定します。非常に簡単な例として二つのフィーチャ「Opt1」と「Opt2」を指定する場合は以下のようなものになります。

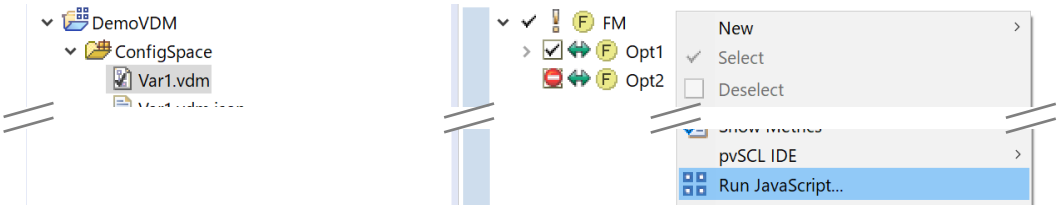
```
[ "Opt1", "Opt2" ]
```

3．JavaScript の実行

作成したJavaScriptスクリプトは、プロジェクトから ～.launch を右クリックし、実行 > ～ build.xml を選択することで実行できます。



また、プロジェクトからVDMエディタ内で右クリックし、Run JavaScript としても実行できます。



■ 外部ファイルに記述した選択指定の実行例

以下の例で、簡単なプロジェクトを用いて、JavaScript API と Ant によって外部のデータファイルを読み込んで、バリアント定義モデルの設定変更を自動化する方法を紹介します。

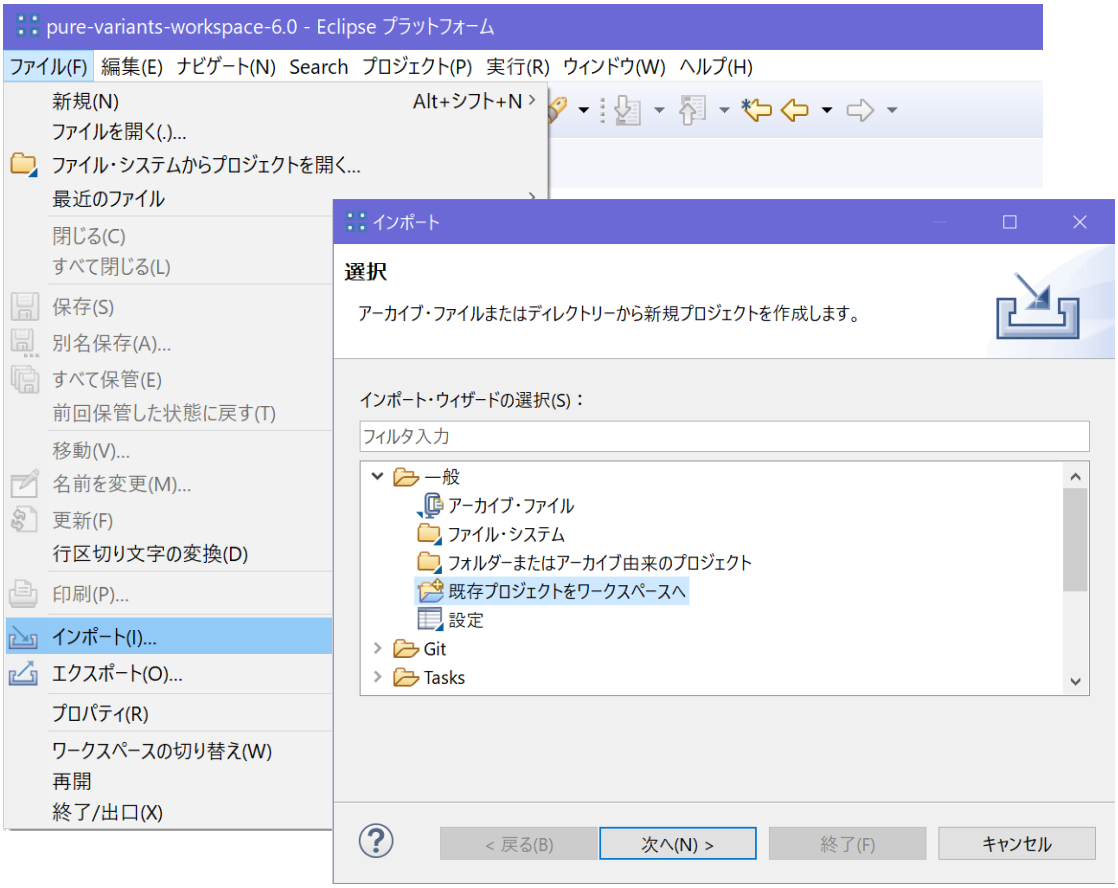
● プロジェクトの構成

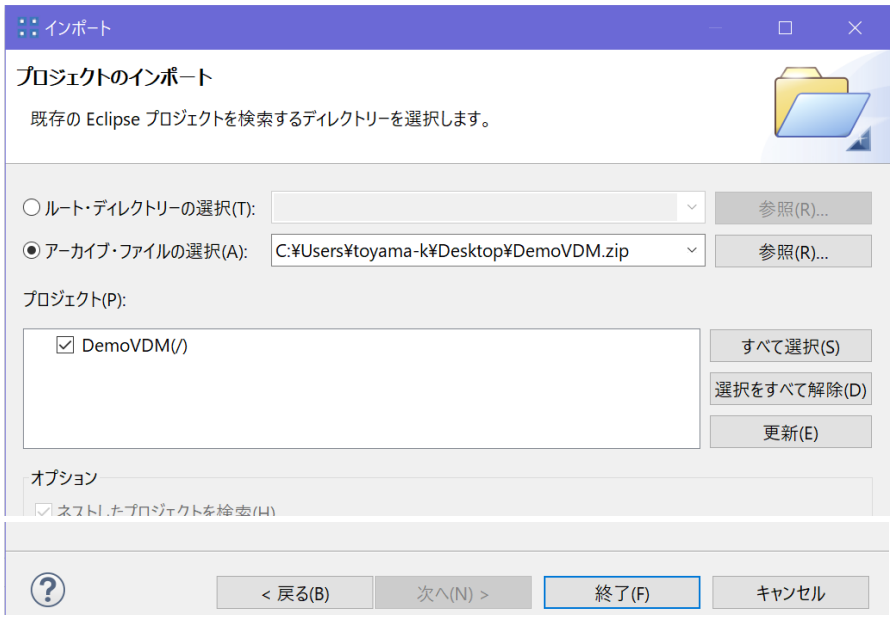
- `FM.xfm` … 簡単なフィーチャモデル
- `ConfigSpace`
  - └ `Var1.vdm` … 評価用に使用する VDM
  - └ `Var1.vdm.json` … 選択情報を定義した外部データファイル
- `build.xml` … Ant 自動化スクリプトであり、以下を行います
  - まず、すべてのセットアップを行う（`pv.import`）
  - 次に、`Var1.vdm` 上で JavaScript を実行
  - 最後に、結果の VDM コンフィグレーションを評価
- `SetSelection.js` … 単純な JavaScript で、上記外部データによって選択操作を行う

● プロジェクトの使用法

1．プロジェクトのインポート

例題プロジェクトファイルのアーカイブ（`DemoVDM.zip`）を適当な場所に置き、このプロジェクトをワークスペースにインポートします。





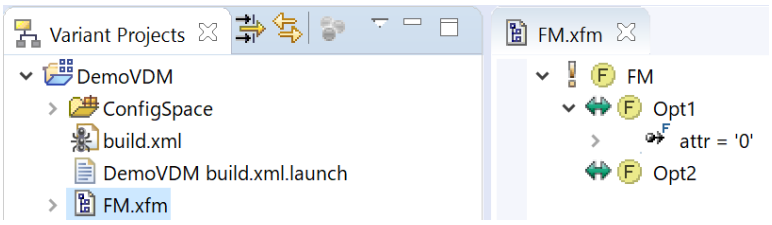
`build.xml` 内の `"pv.import path"` や `"pv.javascript script"` のパスをインポートしてきたワークスペースのものに変更します。

build.xml

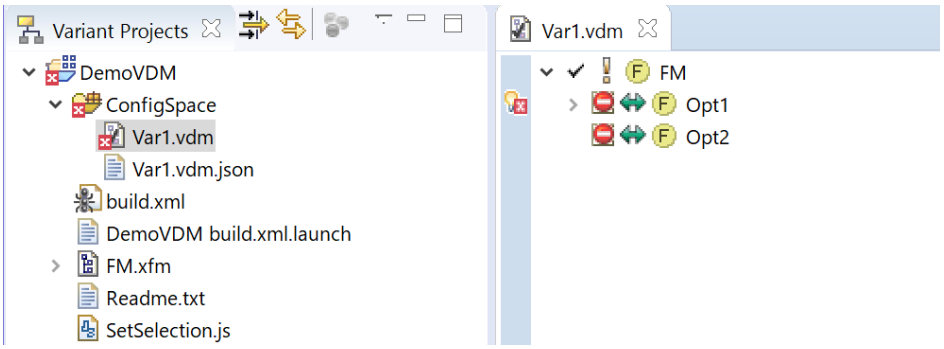
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE xml>

<project name="DemoVDM" default="dist" basedir=".">
  <description>
    simple example build file
  </description>
  <!-- set global properties for this build -->
  <property name="src" location="src"/>
  <property name="build" location="build"/>
  <property name="dist" location="dist"/>

  <target name="dist">
    <!-- Create the time stamp -->
    <tstamp/>
    <!-- Create the build directory structure used by compile -->
    <pv.import path="C:\Users\toyama-k\pure-variants-workspace-6.0\DemoVDM"/>
    <pv.javascript script="C:\Users\toyama-k\pure-variants-workspace-6.0\DemoVDM\SetSelection.js" prc
    <pv.evaluate vdm="DemoVDM\ConfigSpace\Var1.vdm"/>
  </target>
</project>
```

フィーチャの初期状態

プロジェクト起動時に、フィーチャモデルは下図のとおりであり、`Opt1` と `Opt2` の 2 フィーチャがあって、Opt1 には Attributes 属性 attr があり、値は `0` です。



Var1.vdm は以下のとおりで、Opt1 と Opt2 は選択がなされていない状態です。これらは Alternative なフィーチャなので、いずれかを選択する必要がありますが、どちらも選択されていないため、先頭に警告（赤色）のアイコン が出ています。



2．JavaScript でフィーチャの選択を行う

- 選択情報指定ファイル： Var1.vdm.json

```
[ "Opt1", "Opt2" ]
```

これは、フィーチャとして Opt1 と Opt2 があることを示しています。

- 設定用 JavaScript： SetSelection.js

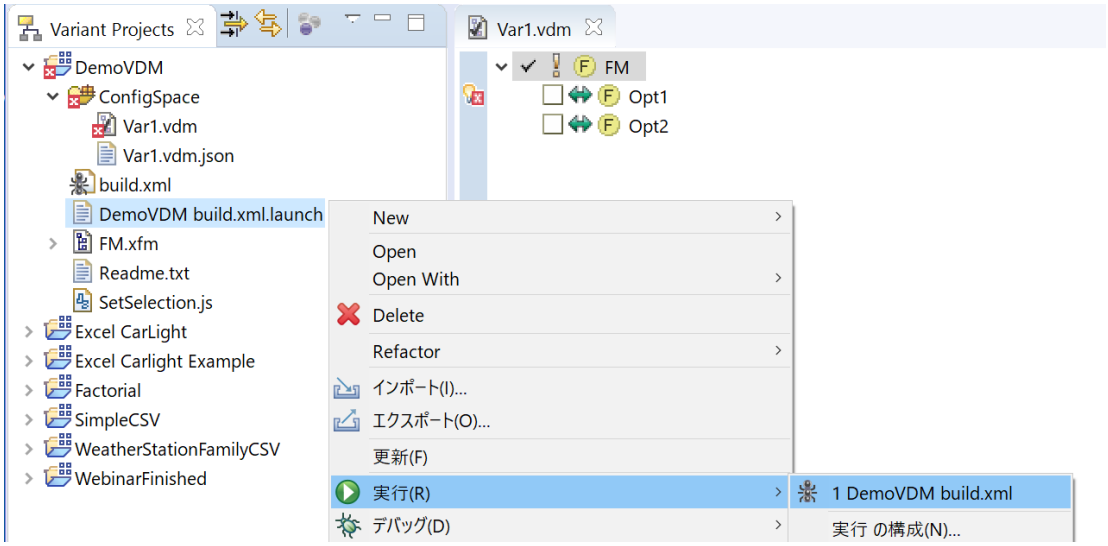
```
・・・
var config = JSON.parse(data);
・・・
   e = fm.getElementWithName(config[0]);
・・・
          if (vdm.getState(e).getSelection() != VariantElementState().SELECTION)
          {
              op.select(e);
              console().println(e.getName() + " selected");
          }
          else
          {
              op.exclude(e);
              console().println(e.getName() + " excluded");
          }
```

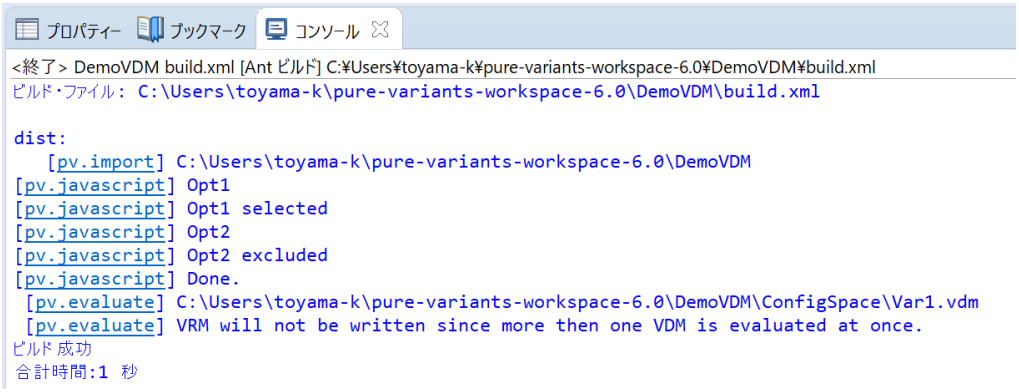
このスクリプトでは、選択情報指定ファイルから読み込んだフィーチャ Opt1 に対して、それが選択されていない（!= VariantElementState().SELECTION）のであれば「選択」（op.select(e)）し、そうでない（選択済み）なら「非選択」（op.exclude(e)）と設定しています[2]。

[2] したがって、Opt1 が選択済みで、Opt2 が非選択の場合などに実行すると、両者が非選択となり、エラーとなります（「ビルド失敗」となります）。

3．pure::variants から JavaScript を実行する

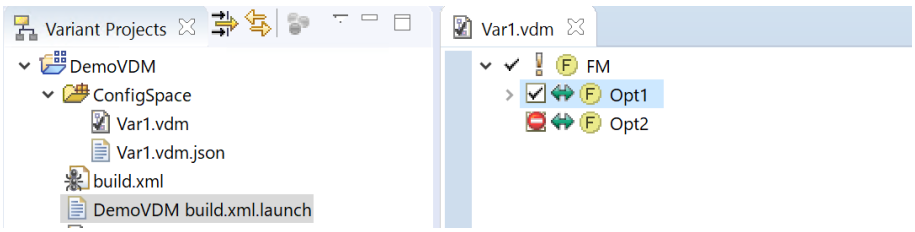
DemoVDM build.xml.launch を右クリックし「実行」を選択します。





選択設定されたフィーチャの状態

Var1.vdm は以下のとおり Opt1 が選択されている状態となります。



JavaScript スクリプトに以下の記述を行うことで、Opt1 に属性値を設定できます。

```
var prop = e.getPropertyWithName("attr");
・・・
    var vel = vdm.getSelectionOfReference(e);
    vel = op.change(vel);
    VariantOperations().setPropertyValue(vel, prop,"123");
・・・
```

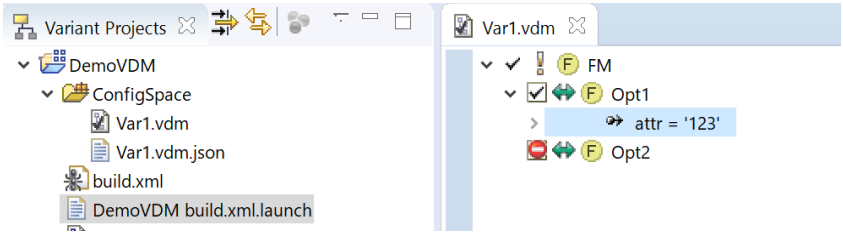
これは、名前 attr の属性（Attributes）に 123 を設定するものです。プロジェクト内で提供する SetSelection.js ではコメントアウトされていますので、有効にして実行してください。

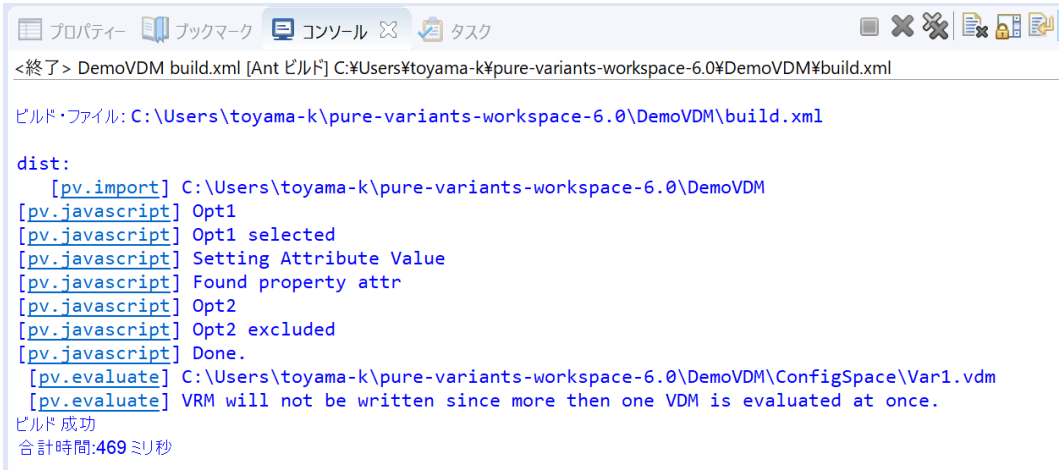
※ 実行前に、この vdm で Opt1 と Opt2 の選択をどちらも非選択と、初期状態に戻してください。

SetSelection.js

```
}

//console().println("Setting Attribute Value");
//var prop = e.getPropertyWithName("attr");
//if(prop != null)
//{
//  console().println("Found property " + prop.getName());
//  var vel = vdm.getSelectionOfReference(e);
//  vel = op.change(vel);
//  VariantOperations().setPropertyValue(vel, prop,"123");
//}

e = fm.getElementWithName(config[1]);
if (e != null)
```

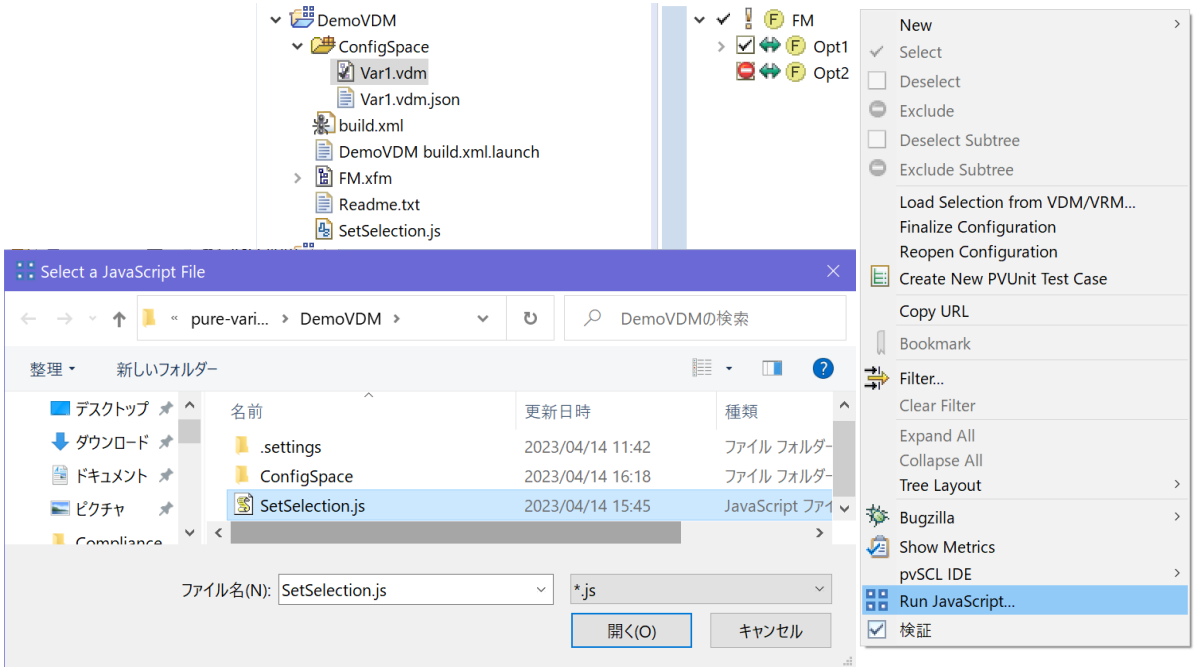
先と同様に DemoVDM build.xml.launch を右クリックして実行すると、Var1.vdm で Opt1 の attr が 123 に設定されます。



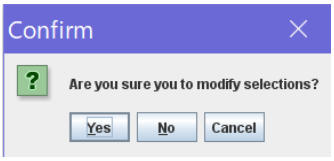


JavaScript 実行の別法

p.3 にありますように、SetSelections.js は VDM エディタの中、すなわちこの ConfigSpace のどの VDM からも実行できます。Var1.vdmを開いてVDMエディタ内で右クリックし、Run JavaScript から Select a JavaScript File ダイアログで JavaScript ファイルを指定して実行します。

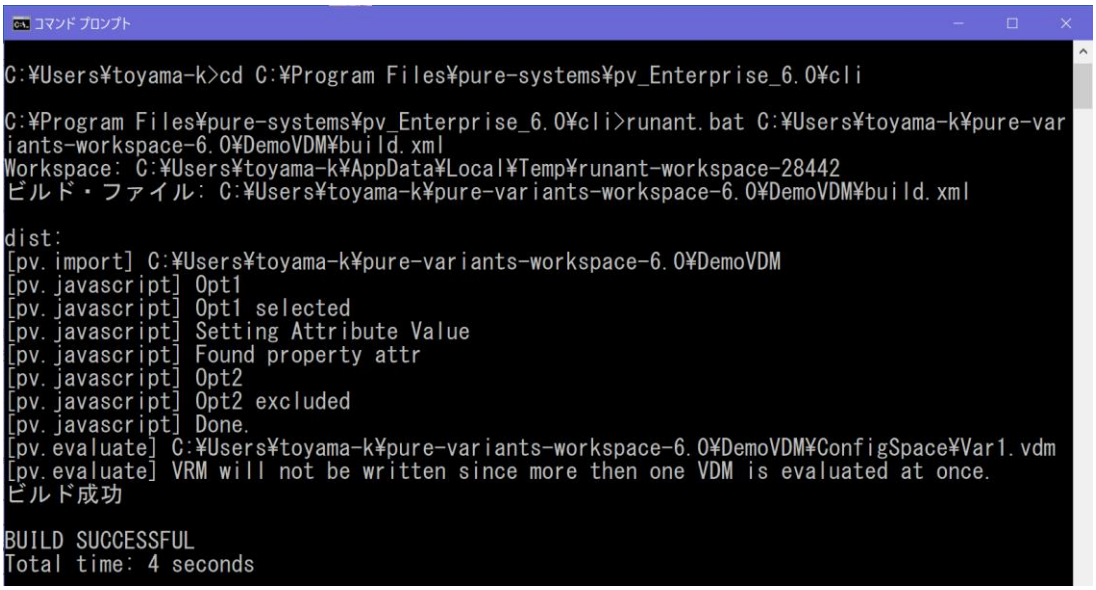


確認ダイアログが出ますので、Yes で進めます。



コマンドラインからの JavaScript バッチ実行

この JavaScript による実行は、コマンドラインからも行えます。pure::variants のインストールディレクトリ ¥cli¥から runant.bat で、build.xml を引数に指定して実行してください。GUI での操作を行わず、コマンドラインから選択指定や値の設定ができます。



FUJI SETSUBI
富士設備工業株式会社 電子機器事業部
〒591-8025 大阪府堺市北区長曽根町1928-1
Tel: 072-252-2128 www.fuji-setsu.co.jp