

CSS GUI Tutorial (KEK 版) ver.1.0 について

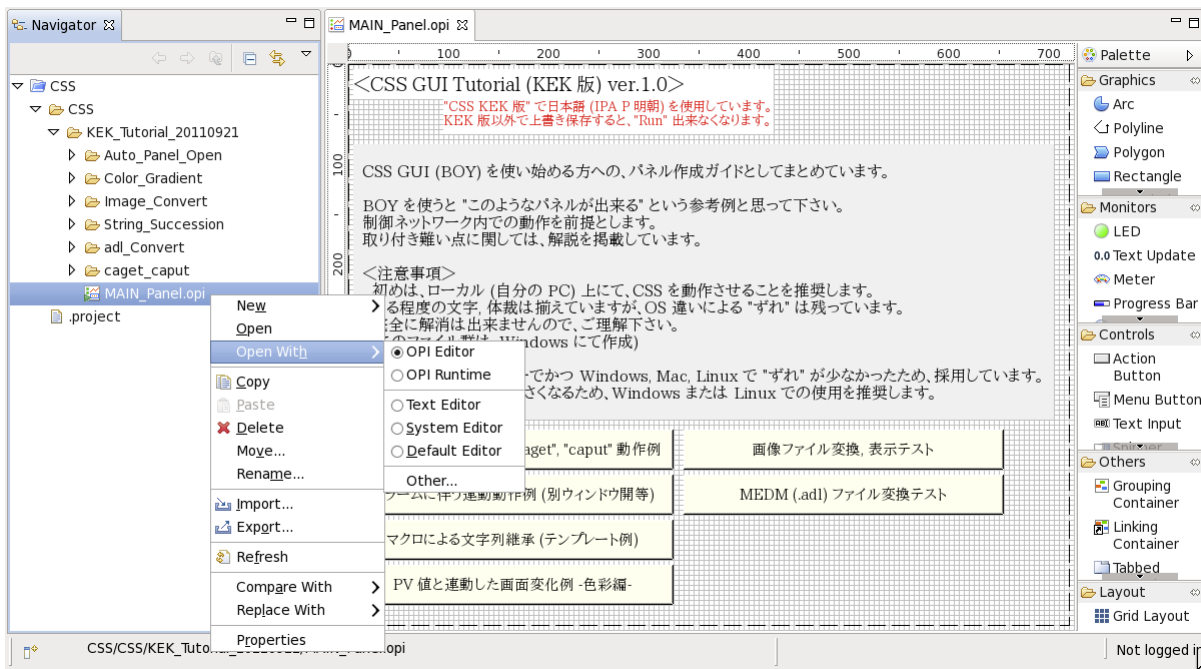
任意のワークスペースに導入して、使用して下さい。

使用自体は、Windows, Mac, Linux および x86, x64 どれでも可能です。
注意として、CSS (KEK 版) 以外で“編集保存”すると、使用出来なくなります。

下図は“CSS”オブジェクト > “CSS”フォルダに配置した例になります。
“Main_Panel.opi”がトップページです。

配置直後にダブルクリック起動させると、“Runtime Mode”になってしまうので
右クリックから“OPI Editor”を指定する事を推奨します。
(次回からは、最後に使用した状態が記憶されます)

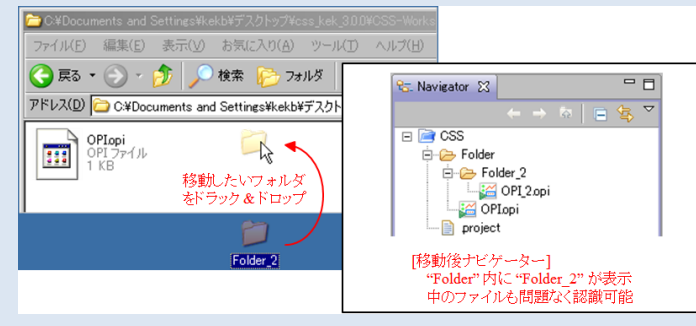
開いたら、メニューバーにある再生ボタン ▶ を押せば、別ウィンドウにて実行されます。



<使用するには>

解凍したファイル群を、そのままインポートする。
または、下図のように、任意のフォルダに移動させるだけです。

※ CSS 起動中の場合は、ナビゲーターをリフレッシュすること



実行中は、実行画面“Runtime”しか使用しません。

内部の設定が見たい場合には、“Editor”の画面から参照して下さい。

<CSS GUI Tutorial (KEK 版) ver.1.0>

"CSS KEK 版" で日本語 (IPA P 明朝) を使用しています。
KEK 版以外で上書き保存すると、"Run" 出来なくなります。

CSS GUI (BOY) を使い始める方への、パネル作成ガイドとしてまとめています。

BOY を使うと "このようなパネルが出来る" という参考例と思って下さい。
制御ネットワーク内での動作を前提とします。
取り付き難い点に関しては、解説を掲載しています。

<注意事項>

初めは、ローカル (自分の PC) 上にて、CSS を動作させることを推奨します。
ある程度の文字、体裁は揃えていますが、OS 違いによる "ずれ" は残っています。
完全に解消は出来ませんので、ご理解下さい。
(このファイル群は、Windows にて作成)

※ IPA 日本語がフリーでかつ Windows, Mac, Linux で "ずれ" が少なかったため、採用しています。
※ Mac では文字が小さくなるため、Windows または Linux での使用を推奨します。

簡易スクリプトを用いた "caget", "caput" 動作例	画像ファイル変換, 表示テスト
アラームに伴う連動動作例 (別ウィンドウ開等)	MEDM (.adl) ファイル変換テスト
マクロによる文字列継承 (テンプレート例)	
PV 値と連動した画面変化例 - 色彩編 -	

<文字列 (マクロ) 継承試験 ver.1.0>

"CSS KEK 版" で日本語 (IPA P 明朝) を使用しています。
KEK 版以外で上書き保存すると、"Run" 出来なくなります。

- 構成が同じパネルを、共用して開きます。
- ここでは、AR エリアモニターを例として用います。

ボタンを押すと、対応するパネルが開きます。

"子".opi ファイル自体は一つです。
"PV Name" を置き換えることで各エリアとします。

YEL-509 (ARE)

YEL-510 (ARNW)

YEL-511 (ARNE)

YEL-512 (ARS)

文字列継承
(マクロ) 解説
※ 画面大

<"caget", "caput" 簡易スクリプト動作 ver.1.0>

"CSS KEK 版" で日本語 (IPA P 明朝) を使用しています。
KEK 版以外で上書き保存すると、"Run" 出来なくなります。

- ボタンを押すと、対応するレコードから PV 値を取得します。
- ここでは、例として外気温を示すレコードを使っています。

--- ボタンエリア ---

--- PV 表示ラベルエリア ---

特別な注意はありません。
"Name" はターゲットとなるので、区別出来るように、
"text" が最初の表示になります。

<準備として>

- ソフトレコード "loc://test_flag_01" を作成
"PV Name" にレコード名を入れるだけで良い

- "Actions" に "Write PV" を追加して、"1" をレコードに書き込む設定
- "Scripts" に対応ファイルを追加
[右カラム "Input PVs"]
 - "PV Name" にレコードを追加記入 (順番重要)
 - ここでは "Trigger" はソフトレコードのみ
 - スクリプトでは、記述順に (上から) pvs[0] pvs[1] になる

<自動アラーム表示, オープン動作 ver.1.0>

"CSS KEK 版" で日本語 (IPA P 明朝) を使用しています。
KEK 版以外で上書き保存すると、"Run" 出来なくなります。

- 開閉ボタンで "1", "0" を出力して、動作させます。
- ここでは、全てソフトレコードを使っています。
- 地図、場所名は、フィクションとして下さい。

<信号出力>

このエリアは、仮想的な操作パネルと考えて下さい。
以下のように仮定します。

"0" → 正常: 安全保持
"1" → 異常: インターロック動作

SOR 北ドアを操作	ふじ棟 北西ドアを操作
0.0 開 閉	0.0 開 閉
SOR 東ドアを操作	ふじ棟 南東ドアを操作
0.0 開 閉	0.0 開 閉
SOR 西ドアを操作	I/L Reset
0.0 開 閉	

--- 解説 1 ---

スクリプトは、登録の分散を避けるため、表示パネルエリアの地下だけに登録しています。
個々に "Rules" を追加しても可能ですが、分散するのでやりません。

<準備として>

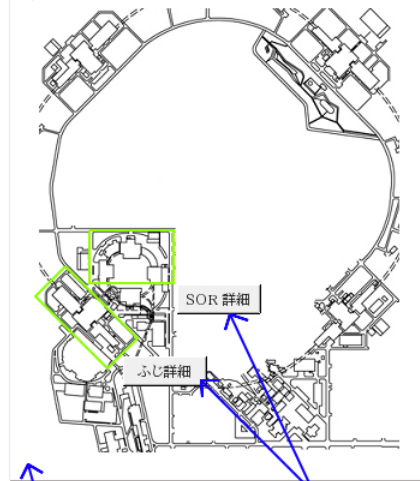
- ソフトレコード "loc://test_flag_01~05" をドア開閉に作成
(各 "Text Update" の "PV Name" にレコード名を入れるだけ)
- 開: "Actions" に "Write PV" を追加し、"1" を書き込む設定
- 閉: "Actions" に "Write PV" を追加し、"0" を書き込む設定

"開" すると、別タブにて拡大図が開きます。
同時に、右側の該当エリアがアラーム (赤色点滅) を出します。
スクリプトにて、"アンドライク" 動作をさせます。

一度 "閉" すると、所属エリア内のアラームは反応なくなります。
つまり、最初の一回のみ、別タブに強制移動します。

I/L リセットは "アンドライク" を確認し、該当エリア内が全て "0" なら
引きアラームが消えて緑色に戻ります。

<表示パネルエリア>



--- 解説 2 ---

別タブで開くものと、同じ画面です。

- "Actions" で "Open OPI" を追加、"File Path" を指定する
- "Replace" は "No" に設定

--- エリア地下 ---

- ソフトレコード "loc://test_flag_06" をデータ保持管理用に作成
("PV Name" にレコード名を入れるだけ)

- "Scripts" に対応ファイルを追加
[右カラム "Input PVs"]
 - "PV Name" にレコードを追加記入 (順番重要)
 - ここでは "Trigger" を全てに (アンドライク)
 - スクリプトでは、記述順に pvs[0] ~ pvs[5] になる
- "flag_06" は I/L リセット時に "1", 実行が終わったら "0" に戻る
- pvs[6] は、シミュレーションレコード

JavaScript
について

<CSS GUI Tutorial (KEK 版) ver.1.0>

"CSS KEK 版" で日本語 (IPA P 明朝) を使用しています。
KEK 版以外で上書き保存すると、"Run" 出来なくなります。

CSS GUI (BOY) を使い始める方への、パネル作成ガイドとしてまとめています。

BOY を使うと "このようなパネルが出来る" という参考例と思って下さい。
制御ネットワーク内での動作を前提とします。
取り付き難い点に関しては、解説を掲載しています。

<注意事項>

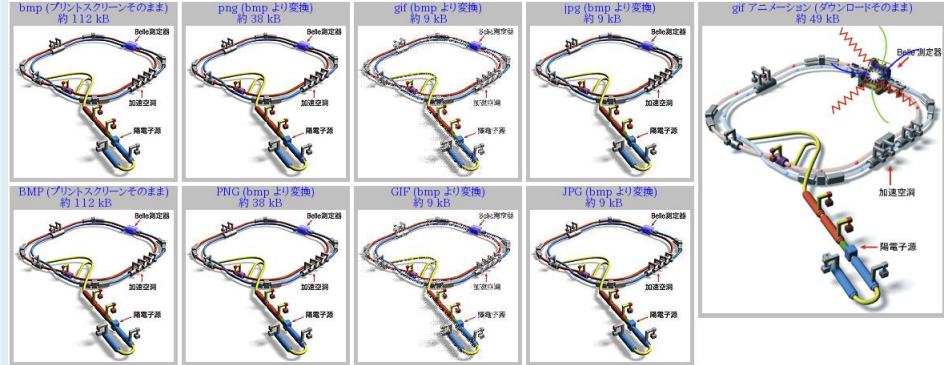
初めは、ローカル (自分の PC) 上にて、CSS を動作させることを推奨します。
ある程度の文字、体裁は揃えています、OS 違いによる "ずれ" は残っています。
完全に解消は出来ませんので、ご理解下さい。
(このファイル群は、Windows にて作成)

※ IPA 日本語がフリーでかつ Windows, Mac, Linux で "ずれ" が少なかったため、採用しています。
※ Mac では文字が小さくなるため、Windows または Linux での使用を推奨します。

簡易スクリプトを用いた "caget", "caput" 動作例	画像ファイル変換, 表示テスト
アラームに伴う連動動作例 (別ウィンドウ開等)	MEDM (.adl) ファイル変換テスト
マクロによる文字列継承 (テンプレート例)	
PV 値と連動した画面変化例 - 色彩編 -	

<画像ファイル変換, 表示確認 ver.1.0>

Windows 標準ペイントにて、単に変換したものです。
複雑な処理は一切していません。
(※透過確認のため下地は濃色)



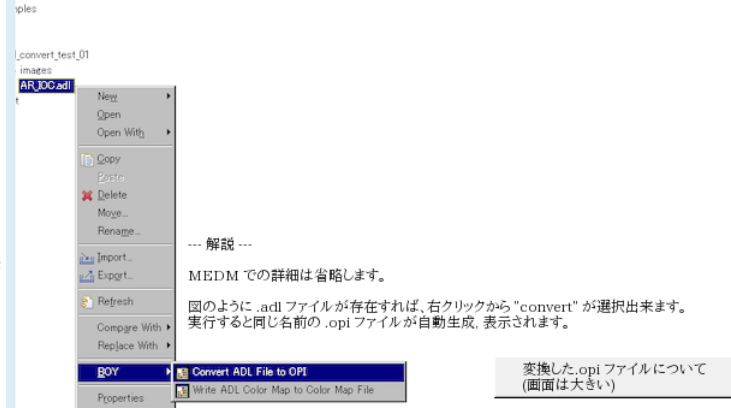
--- "Image" ウィジェット使用に関する補足 ---

- ・パスとディレクトリ構造に気を付けること
 - ・拡張子が大文字の場合、ファイル選択時にフォルダ内に表示されない (Navigator で見えていけば、"Resource Path" に正確記述すれば良い)
- 所見から、拡張子は小文字にし、相対パスで指定する方が良いと思います。

<".adl" → ".opi" 変換試験 ver.1.0>

"CSS KEK 版" で日本語 (IPA P 明朝) を使用しています。
KEK 版以外で上書き保存すると、"Run" 出来なくなります。

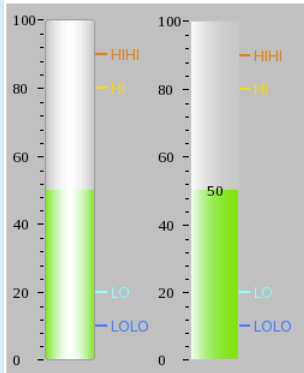
- ・ ".adl" ファイルを簡易変換した場合の取り扱いについて、記述します。
- ・ここでは、例として "AR_IOC.adl" を用います。



<色相変化動作 ver.1.0>

"CSS KEK 版" で日本語 (IPA P 明朝) を使用しています。
KEK 版以外で上書き保存すると、"Run" 出来なくなります。

- ・このパネルでは、ソフトレコードを使っています。
- ・値に応じて、"Fill" の色を連続的に変化させます。



比較までに、二つ用意します。

- (左) "Tank" ウィジェット
- (右) "Progress Bar" ウィジェット

--- 実例 ---

実際に、運転に流用できそうなパネルを作成しました。
BT および PF-AR 地区 Area Monitor が題材です。
元々は MEDM パネルだったものを、"PF-AR Only" という形でコンパクトにまとめ直したのになります。
(値変化率が小さいため、色相変化は分かり難いです)

BT & PF-AR
Area Monitor

"Rules" では出来ないの "Scripts" を各ウィジェットに組み込んでいます。
※ スクリプトファイルは一つですが、"Trigger" は対応させること!

"pv0" → 0.1 秒毎に 1 ずつ増加する (0 から 100 まで) ソフトレコード

```
importPackage(Packages.org.csstudio.opibuilder.scriptUtil);
var pv0=PVUtil.getDouble(pvs[0]);
```

"HSB" 関数化
のための項目

```
var hue_theta;
var theta1=0;
var theta2=240;
var val1=100;
var val2=0;
hue_theta=theta1+(((theta2-theta1)/(val2-val1))*(pv0-val1));
var FillColor=ColorFontUtil.getColorFromHSB(hue_theta,0.9,0.9);
```

簡単な関数を使用しています。
単に割合を出して、現在値を当てはめるだけです。

```
widget.setPropertyValue("fill_color",FillColor);
```

この記述で、"HSB" 表記から色を指定できます。

[HSB (HSV) についての補足]
色相 (Hue), 彩度 (Saturation, Chroma), 明度 (Brightness, Lightness, Value)
今回のように色相を関数化する場合、"RGB" より楽になります。

※ RGB や HSB を正確に指定するなら、ベクトル計算して下さい。