

## 目次

E.Builder 操作説明書 .....	1	レイアウト方法 .....	11
セットアップ .....	2	レイアウトの確認 .....	11
E. Builder とは .....	2	アイテムのバリューをストリンガーに表示 .....	11
対応 OS / ブラウザーと画像フォーマット .....	2	アイテムのバインド .....	11
セットアップ方法 .....	2	フレームとページ の概念 .....	11
resource.js の作成と使いかた .....	2	フレームとページ の紐付け .....	11
アドレス・マップの作成 .....	2	1 つのページに複数のページを表示する .....	12
各部の名称と機能 .....	3	ページの切り替え（ページ遷移） .....	12
アイテムの機能と使いかた .....	4	ポップアップ画面の作り方 .....	13
プロパティ一覧 .....	7	レンジ・スライダーの作り方 .....	13
レイアウト .....	9	ライン（線）の作り方 .....	14
アイテムの配置、サイズ変更、デザイン .....	9	ローカライズ .....	14
アイテムの配置 .....	9	パート・パラメーターの作り方 .....	15
アイテム・サイズの変更 .....	9	パラメーターの設定 .....	15
複数アイテムの選択 .....	9	キー・パラメーターの作り方 .....	16
アイテムのロック .....	9	MFX パラメーターの割り当て .....	17
スナップと整列 .....	9	プロジェクトの再現と納品 .....	18
重ね順 .....	9	補足 .....	18
CSS とアイテムのプロパティ .....	10	カスタム UI の作成 .....	18
CSS の変更と追加 .....	10	カスタム js ファイルの読み込み .....	18
Reload .....	10	E.Builder が起動しない場合 .....	18

# セットアップ

## E. Builder とは

E.Builder は、製品エディターのレイアウトとレイアウトしたパーツに MIDI のパラメーターを割り当てることができるソフトウェアです。

※ E.Builder では、レイアウトするパーツのことを『アイテム』とよびます。

## 対応 OS / ブラウザーと画像フォーマット

対応 OS と使用可能な画像フォーマットは、以下のとおりです。

	PC	mac
OS	Windows7 SP1 以上	macOS 10.11 以上
ブラウザー	Chrome (*)	safari
画像フォーマット	png (推奨)、jpg、bmp 形式	

### 注意

エクスポートしたレイアウトを確認するには引数の設定が必要です (P.11)。

## セットアップ方法

E.Builder を使用するには、以下のファイルを準備し、設定をします。

- Quattro(Win).zip
- ELF (Git リポジトリ)

Quattro(Win).zip と ELF (Git リポジトリ) は、開発部・技術部サーバーからダウンロード／クローンします。

ダウンロード／クローンには、メンバー登録者が必要です。未登録の方は管理者に問い合わせてください。

## 設定

1. Quattro(Win).zip を解凍します。
2. 解凍した Quattro(Win) フォルダー内にある、「ebuilder.exe」と「quattro(win).exe」のショートカットを作成します。
3. 作成したショートカットのプロパティを表示し、リンク先に URL 形式のパスを追加します。

### ebuilder.exe ショートカット

C:\ (quattro(win) フォルダーのパス) \ebuilder.exe file:/// (ELF フォルダーのパス) /ebuilder.html

### quattro(win).exe ショートカット

C:\ (quattro(win) フォルダーのパス) \quattro(win).exe file:/// (ELF フォルダーのパス) /elf.html

※ パスに半角スペースが含まれると正しく起動できません。半角スペースが入らないように注意してください。

## ebuilder.exe と quattro(win).exe について

ebuilder.exe は、エディターの画面レイアウトとレイアウトしたアイテムに MIDI のパラメーターを割り当てることのできるアプリケーションです。

ebuilder.exe で作成した html は、quattro(win).exe に読み込むことで、製品と MIDI で通信できるようになります。

## resource.js の作成と使いかた

resource.js ファイルは、「Group button」や「Select panel」など、resource-id プロパティを持つアイテムに、パラメーター名や画像を割り当てるために作成します。

E.Builder でのレイアウトを始める前に、テキスト・エディターなどで作成／編集をしてください。resource.js ファイルの編集方法は、以下のとおりです。

1. ELF-js-config-resource.js ファイルを開きます。
2. 以下の書き方で、パラメーター名や画像を追加します。

### ●パラメーター名の場合

```
{ text: "パラメーター名 1, パラメーター名 2, ..." },
```

### ●画像の場合

```
{ icon: "images/ ファイル名 .png, images/ ファイル名 .png, ..." },
```

3. 画像を追加した場合は、ELF-images フォルダー内にファイルをコピーします。
4. resource.js を保存し、「Reload」ボタンを押します。
5. resource-id プロパティを持つアイテムを配置し、アイテムの resource-id プロパティで、リソースを選びます。  
選択した内容が、アイテムに反映されます。

※ 画像を読み込む場合は、読み込んだ画像のサイズによって、アイテムのプロパティや CSS での調整が必要になります。

## アドレス・マップの作成

E.Builder でアイテムに MIDI のパラメーターを割り当てるには、製品のアドレス・マップ・ファイル (address\_map.js) が必要です。レイアウトと合わせて、MIDI パラメーターの割り当てをする場合は、address\_map.js を作成してください。

※ 参考用の address\_map.js は、開発部・技術部サーバーからダウンロードできます。

作成したアドレスマップは、ELF- [js] - [config] にコピーします。

# 各部の名称と機能

## ● ファイル・エリア

プロジェクト (.elf) の読み込みや保存、html ファイルのエクスポートをします。

## ● Items / Pages ボタン

アイテム／ページ・エリアの表示を切り替えます。

## ● スナップ／整列エリア

アイテムの整列をします。

## ● 重ね順ボタン

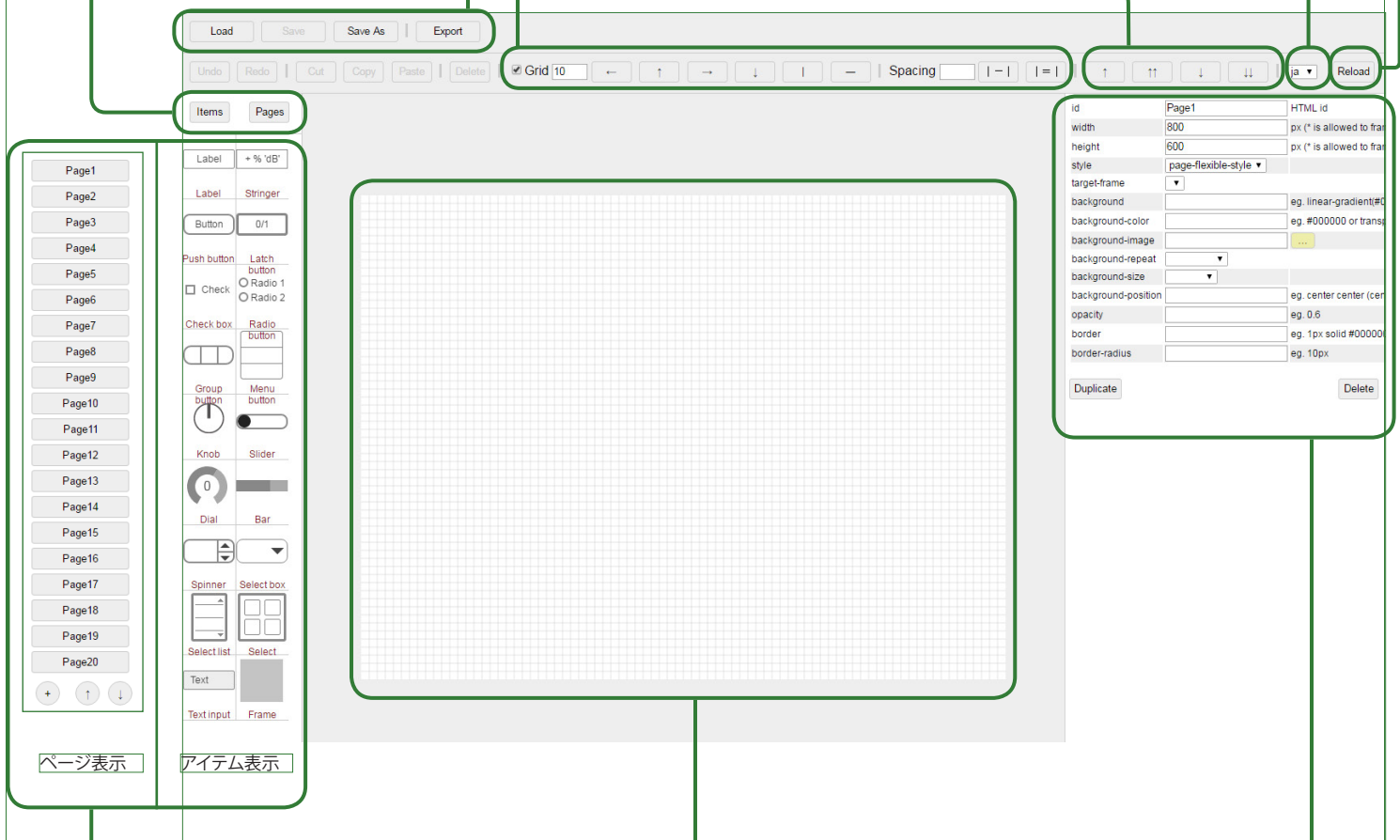
アイテムの重ね順を変更します。

## ● Reload ボタン

CSS の変更や追加した画像を反映して、ページを再表示します。

## ● 言語切替

言語設定を変更し、ローカライズ・テキストの確認をします。



## ● アイテム／ページ・エリア

Items ボタンを押すと、レイアウト用の部品が表示されます。配置されたアイテムには、固有の id が割り当てられます。

Pages ボタンを押すと、作成中のページ一覧が表示され、ページ追加や削除、順番の変更ができます。

## ● エディット・エリア

ページごとにアイテムを配置します。

## ● プロパティ・エリア

アイテムのデザインや動作に関する設定をします。MIDI パラメーターの割り当てやページ遷移の設定、色の変更など、アイテム固有の設定をします。「プロパティ一覧」(P.7)

## アイテムの機能と使いかた

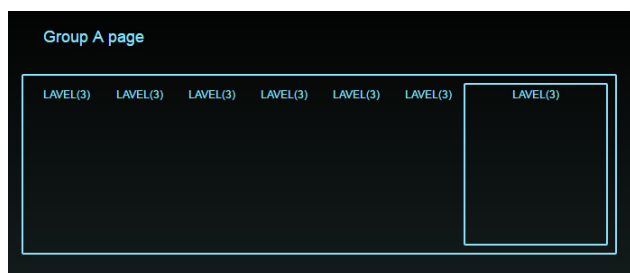
各アイテムの機能と使いかたは以下のとおりです。

### Label



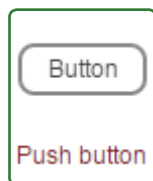
パラメーター名やタイトルなどのテキストを配置するアイテムです。

テキストを入力せず、仕切り枠として使うことや画像を読み込み、配置することも可能です。



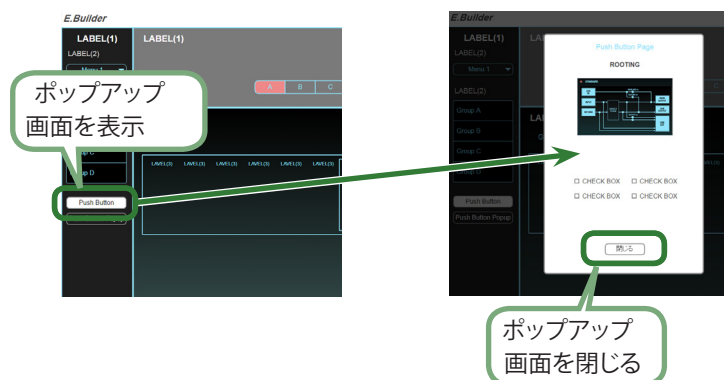
(青色の文字、仕切り枠が Label)

### Push button

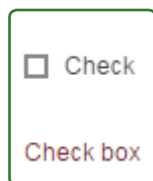


ブッシュ型のボタンです。

ポップアップ画面やマニュアルの表示、画面を閉じるトリガーとしても使用できます。

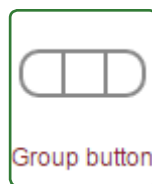


### Check box



チェックボックス型の切り替えアイテムです。  
2 値の切り替えができます。

### Group button

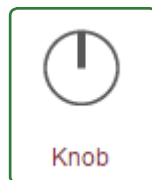


複数のパラメーター（グループ）の中から、1 つを選ぶボタンです。

ボタン選択時に Frame のページの遷移やパートの切り替えセレクターとしても使用します。



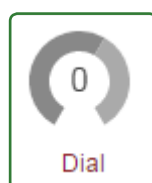
### Knob



ノブ型のアイテムです。

オリジナルの画像を作成し、ノブの形状を変更することも可能です。

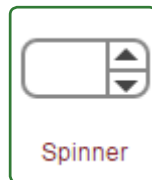
### Dial



帯型のメーター・アイテムです。  
値がアイテムの中央に表示されます。

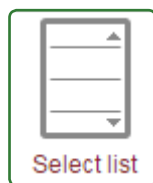
値は、value-hidden プロパティを有効にすると非表示にすることが可能です。

### Spinner



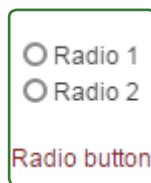
右側の▲/▼ボタンで値を変更することができます。また、左側の値欄は数値の直接入力も可能です。

## Select list



アイテムをクリックすると、セレクト・リストがポップアップします。

## Radio button

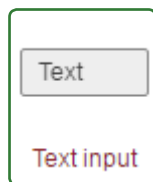


ラジオ・ボタン型のアイテムです。

設定したパラメーターの数だけ、ラジオ・ボタンが作成されます。

パラメーターの設定は、「resource-id」プロパティでします。

## Text input



テキスト入力に対応したアイテムです。  
Labelとは異なり、ユーザーが任意の文字列を入力することができます。

入力前

User Patch-name

入力中

あいうえお abcdef

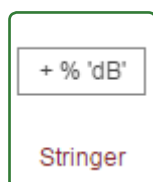
## Menu button



縦型のメニュー・アイテムです。

Group buttonと同様に、複数のパラメーターの中から、1つを選びます

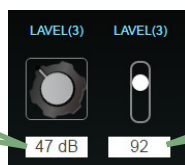
## Stringer



Slider や Knob などの値を表示させることができます。また、単位などの文字列を追加したり、計算式を設定して結果を表示することもできます。

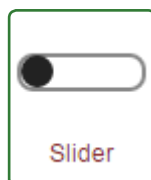
参照: 「アイテムのバリューをストリンガーに表示」(P.11)

Stringer  
Value + 単位



Stringer  
Value のみ

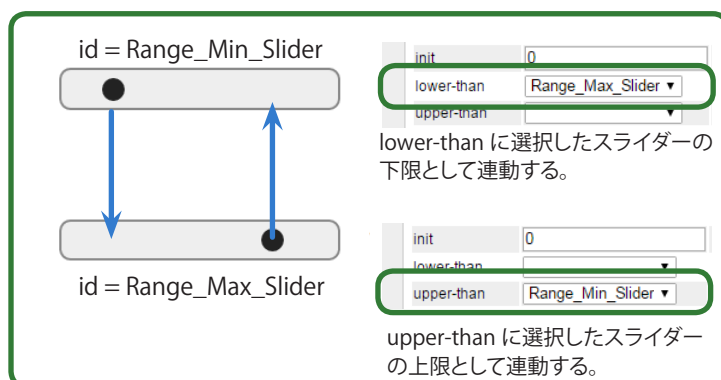
## Slider



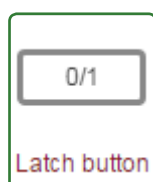
スライダー型のアイテムです。

アイテムの width が height よりも大きい場合は横型スライダーになり、小さい場合は縦型スライダーになります。

また、Slider を 2 つ使用し、数値範囲（上限／下限）を設定することも可能です。数値範囲を設定する場合は、「lower-than」、「upper-than」プロパティを使用します。

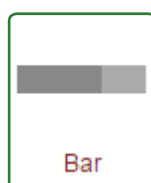


## Latch button



ボタン型の切り替えアイテムです。  
オン／オフなどの 2 値の切り替えができます。

## Bar



バー型のアイテムです。

Slider とは異なり、ノブがなく、値がアイテム中央に表示されます。

また、Bar を 2 つ使用し、レンジ・スライダーを作ることができます。

→ (「レンジ・スライダーの作り方」) (P.13)

値は、value-hidden プロパティを有効にすると非表示にすることが可能です。

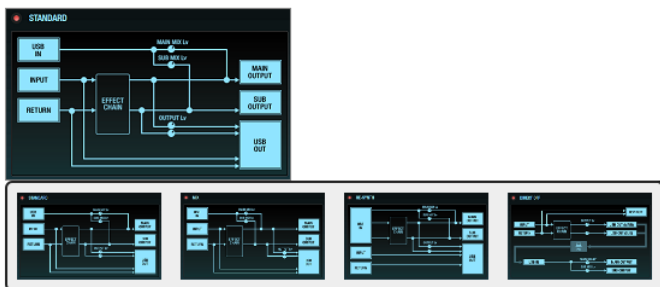
アイテムの width が height よりも大きい場合は横型スライダーになり、小さい場合は縦型スライダーになります。

## Select panel



任意の画像を割り当て、画像をクリックして選択させるアイテムです。

参照：「resource.js の作成と使いかた」(P.2)

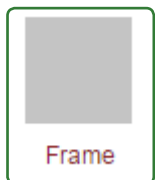


## Select box



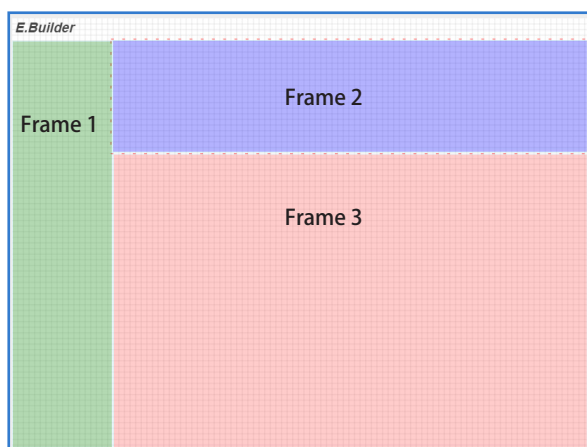
アイテムをクリックすると、プルダウン型のリストを表示します。

## Frame



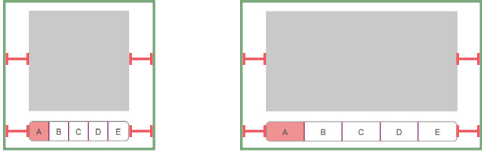
ページを表示する位置と領域を指定するアイテムです。紐付けしたページを Frame 内に表示します。

また Frame には複数のページを紐付けることができ、Group ボタンや Menu button などで表示するページを切り替えることができます。



## プロパティ一覧

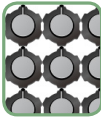

E.Builder のアイテムには、以下のプロパティがあります。  
プロパティを変更することで、アイテムにパラメーターや機能を割り当てることや、デザインを変更することができます。

名称	機能
id	<p>ページ／アイテムを識別するための符号です。 id は、ページ／アイテム配置した際、自動的に割り振られます。割り振られた id は、重複しない番号や名称に変更することが可能です。</p> <p><b>注意</b></p> <p>id は、半角英数とアンダースコアのみ使用できます。また id の先頭は、半角のアルファベットで始まるようにしてください。</p>
left	ページの左端を基準として、アイテムの左位置を指定します。
top	ページの上端を基準として、アイテムの上位置を指定します。
right	<p>ページの右端を基準として、アイテムの右位置を指定します。</p> <p>left と right を設定するとページ・サイズが変わっても左右の間隔を保持する。</p> 
bottom	ページの下端を基準として、アイテムの下位置を指定します。
width	アイテムの幅を指定します。
height	アイテムの高さを指定します。
class	アイテムが参照する CSS のクラスを指定します。指定したクラスの設定に従い、アイテムのデザインが変わります。

名称	機能
style	<p>ページの表示方法を指定します。</p> <p><b>page-flexible-style :</b> target-frame で指定したフレームの左上隅を基準にページが表示されます。</p> <p><b>page-left-style :</b> target-frame で指定したフレームの左側を基準にページが表示されます。</p> <p><b>page-center-style :</b> target-frame で指定したフレームの水平方向センターを基準にページが表示されます。</p> <p><b>page-right-style :</b> target-frame で指定したフレームの右側を基準にページが表示されます。</p> <p><b>page-middle-style :</b> target-frame で指定したフレームの水平／垂直方向センター（フレームの中心）を基準にページが表示されます。</p> <p><b>page-popup-style :</b> ページをポップアップ表示できるようにします。</p>
target-frame	ページを表示するフレームを指定します。
max-height	Select box のプルダウン・リストの高さを指定します。
max-width	Select panel の選択パネルの幅を指定します。
image-width	Select panel の選択パネル内に表示される画像の幅を指定します。
thumb-width	スライダー・ノブの幅を指定します。
thumb-height	スライダー・ノブの高さを指定します。
min	アイテムの変化幅の最小値を指定します。 * MIDI の最小値ではなく、UI 上の最小値
max	アイテムの変化幅の最大値を指定します。 * MIDI の最小値ではなく、UI 上の最大値
Center	設定した値の位置でスライダーの値を示す色が切り替わります。色の指定は、css の color で変更します。
lower-than	選択したスライダーやノブ、ダイヤルの lower（下限）として連動するようになります。
upper-than	選択したスライダーの upper（上限）として連動するようになります。
linking	lower-than、upper-than を設定した、2 つの Bar アイテムをリンクさせ、レンジの変更が可能なスライダー（レンジ・スライダー）にします。
value-hidden	Dial、Bar アイテムの値を非表示にします。
touch-handler	アイテムにタッチした際、touch-handler で指定した Javascript 関数を呼び出します。独自のタッチ操作や操作子でコントロールする際、使用します。



名称	機能
nth-power	Spinner アイテムの値に 10 の n 乗を掛けて表示します。
ascii-only	Text input アイテムで ASCII コードの文字列のみ表示されます。コード外の文字列は、「?」で表示されます。
text	アイテムの表示名を入力します。
icon	スライダー・ノブの画像を指定します。 * 画像は、ELF フォルダの中にある images フォルダにコピーしてください。
resource-id	resource.js 参照し、アイテムにテキストや画像を割り当てます。
init	アイテムの初期値を指定します。 * MIDI の初期値ではなく、UI 上の初期値
parameter-id	address_map.js を参照し、アイテムに MIDI パラメーターを割り当てます。
parameter-size	パラメーターのバイトサイズを指定します。 アドレス・マップから取得します。 ※ 手動での設定はしません。
parameter-vofs	オフセット値を指定します。 指定するとアイテムの値とオフセット値を足し合わせた値を MIDI 出力します。 ※ 手動での設定はしません。
option	MFX パラメーターを設定する際に使用します。
option-addr	詳しくは、「MFX パラメーターの割り当て」(P.17) を参照してください。
option-type	
trigger-changed	アイテムの値やパラメーターが変化すると、javascript のトリガー・イベントを出すようになります。別途スクリプトを記述することで、追加のアクションをさせることが可能になります。
replace-brock	パートの切り替えに使用します。 詳しくは、「パート・パラメーターの作り方」(P.15) を参照してください。
replace-frame	replace-brock の対象となるフレームを指定します。詳しくは、「パート・パラメーターの作り方」(P.15) を参照してください。
page-selector	指定したフレームのページ遷移を発生させるアイテムになります。
page-order	指定したフレームに紐付くページを任意に指定し、ページ遷移するようにします。
popup	Push button を押した際、ポップ・アップ表示させるページを指定します。「--close」を選択すると、ポップ・アップ画面を閉じるボタンになります。 * popup には、style が「page-popup-style」になっているページのみ表示されます。

名称	機能
inc-value	Push button を押した際、指定したアイテムの値をインクリメントまたはデクリメントします。
dec-value	
stringer	アイテムの値を指定した Stringer アイテムに表示させます。
format	Stringer の表示を設定します。アイテムの値だけでなく、単位などの文字列、値を元にした計算式なども設定することができます。
color	テキストの色を指定します。
font-size	テキストのフォント・サイズを指定します。
font-weight	テキストをレギュラーまたはボールドにします。
font-style	テキストをレギュラー、oblique (斜字体)、italic (イタリック体) にします。
text-align	テキストの位置を指定します。 <b>left</b> : 左寄せ <b>center</b> : 中央寄せ <b>right</b> : 右寄せ
background	アイテムの背景色やグラデーションなど、背景全般に関する設定ができます。
background-color	背景色を設定をします。
background-image	背景に画像を貼り付けます。 * 背景に使用する画像は、ELF-images フォルダ内にファイルをコピーしてください。
background-repeat	画像を繰り返し貼り付けます。 <b>repeat</b> :  <b>no-repeat</b> : 
background-size	背景画像のサイズを指定します。 <b>auto</b> : 自動的に算出します。 <b>contain</b> : 画像の縦横比を保持しつつ、ページ領域内で最大のサイズになります。 <b>cover</b> : 画像の縦横比を保持しつつ、ページ領域を完全に覆う最小サイズになります。
background-position	背景画像を配置する基準位置を指定します。 center、left、right、top、bottom などの指定が可能。
opacity	透過度を指定します。
border	枠線の太さ、色、形を指定します。
border-radius	枠線の形を指定します。



# レイアウト

## アイテムの配置、サイズ変更、デザイン

レイアウトの基本的な操作について説明します。

### アイテムの配置

アイテムは、エディット・エリアにドラッグして配置します。  
配置したアイテムは、プロパティ・エリアの「left」、「top」に値を入力すると、指定した位置に移動します。

### アイテム・サイズの変更

アイテムのサイズは、マウスの右ボタンを押しながらドラッグ (mac の場合は Command + ドラッグ)、またはプロパティ・エリアの「width」、「height」に値を入力して、変更します。

### 複数アイテムの選択

Ctrl キーを押しながらアイテムをクリックすると、複数同時に選択することができます。

### アイテムのロック

Alt キーを押しながらアイテムをクリックすると、アイテムのロック / ロック解除をすることができます。アイテムをロックすると、位置が固定され、また選択や編集をできなくすることができます。

## スナップと整列

### グリッド

グリッド設定にあるチェック・ボックスを有効にすると、指定した間隔でグリッド（格子線）が表示されます。

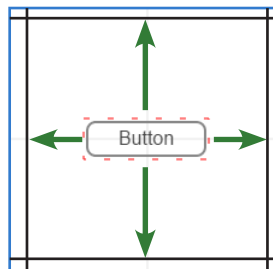
グリッドの色は、ebuilder.css-#e-canvas-color で変更することができます。

また、グリッドの色が見にくい場合は、ebuilder.css-#e-canvas-background でエディット・エリアの背景色を変更し、調整してください。



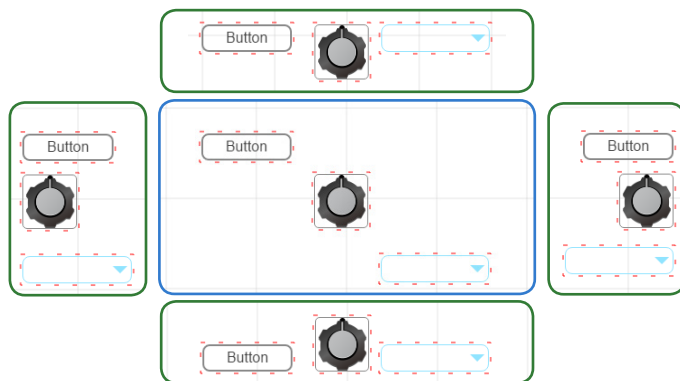
### スナップ

アイテムを1つ選択して、スナップ、整列エリアの「←」「↑」「→」「↓」ボタンを押すと最寄りのグリッドの位置にアイテムがスナップ（位置合わせ）します。



### 整列

アイテムを複数選択して、スナップ、整列エリアの「←」「↑」「→」「↓」ボタンを押すと、選択したアイテムの最も左、上、右、下の位置に整列します。



また、アイテムを1つ選択して、スナップ、整列エリアの「|」、「—」を押すと、ページの垂直方向、水平方向のセンターにアイテムが移動し、アイテムを複数選択して「|」、「—」を押すと、アイテムが垂直方向、水平方向に整列します。

### 間隔を指定して整列

アイテムを複数選択し「| — |」を押すと、Spacing で指定した間隔で、アイテムが水平方向に整列します。

### 均等配置

アイテムを複数選択し「| = |」を押すと、アイテムが水平方向に均等配置されます。

### 重ね順

アイテムの重ね順は、「重ね順ボタン」で変更することができます。

ボタン	動作
↑	選択したアイテムを前面に移動させます。
↑↑	選択したアイテムを最前面に移動させます。
↓	選択したアイテムを背面に移動させます。
↓↓	選択したアイテムを最背面に移動させます。

## CSS とアイテムのプロパティ

アイテムのデザインは、CSS とアイテムのプロパティで変更することができます。

### ● CSS

CSS を変更した場合は、変更した CSS を参照するアイテムのデザインが一括変更されます。

### ●アイテムのプロパティ

アイテムのプロパティでの変更は、CSS の設定よりも優先されるため、アイテムごとに個別のデザインすることができます。

各アイテムのプロパティについては、「プロパティ一覧」(P.7)を参照してください。

## CSS の変更と追加

アイテムの CSS は、ELF-CSS フォルダーに格納されています。変更したいアイテムの CSS を開き、色や形、画像の指定をします。

### ● 画像変更例 (Knob.css)

目的の CSS を開き、ファイルの参照を書き換えます。

```

26 .item-knob-style:focus {
27   background-color: #e88;
28 }
29 .item-knob-style div {
30   width: 100%; height: 100%;
31   background-image: url("../images/img_knob.png");
32   background-repeat: no-repeat;
33   background-position: center;
34   transform: rotate(0deg);
35   -moz-transform: rotate(0deg);
36   -webkit-transform: rotate(0deg);
37

```

画像を変更する場合は、images フォルダーに画像をコピーし、ファイル名を指定します。

### ● CSS の追加例 (Label.css)

デザインの異なるアイテムを使い分けたい (バリエーションを作りたい) 場合は、新たに CSS を作り、アイテム・プロパティの class で CSS を切り替えます。

1. ELF-CSS フォルダーを開き、目的のアイテムの css ファイルを複製します。
2. 複製した CSS ファイルを開き、すべてのクラス名を変更します。

元の CSS のクラス名

```

.item-label-style {
  position: absolute;
}
.item-label-style p {
  display: table-cell;
}

```

複製した CSS のクラス名

```

.item-label-style2 {
  position: absolute;
}
.item-label-style2 p {
  display: table-cell;
}

```

### 3. CSS の内容を変更します。

```

.item-label-style {
  position: absolute;
  clear: both;
  display: table;
  margin: 0;
  padding: 0;
  text-align: center;
  color: #444;
  /* background-color: #eee; */
  /* border: 1px solid #888; */
  border-radius: 2px;
}

```

color: #444;

```

.item-label-style2 {
  position: absolute;
  clear: both;
  display: table;
  margin: 0;
  padding: 0;
  text-align: center;
  color: #90E4FF;
  font-size: 30px;
  border-radius: 2px;
}

```

color: #90E4FF;  
font-size: 30px;

色とフォントサイズを変更したバリエーション

### 4. CSS を保存し、E.Builder を起動します。

※ 起動中の場合は、Reload ボタンを押します。

### 5. Label アイテムを配置します。

### 6. 配置したアイテムの「class」プロパティに、手順 2 で変更したクラス名を選びます。

id	id157d19e78ce2c0
left	275
top	63
width	100
height	20
class	item-label-style2
text	Label
color	

設定が反映され、ラベルのバリエーションが使えるようになります。

## Reload

Reload ボタンを押すと、アイテムやページ、パラメーターなどの編集状態を保持したまま、変更した CSS や resource.js、images フォルダーに追加した画像ファイル読み込み、即座に反映します。

そのため、E.Builder の再起動やプロジェクトの開きなおしをすることなく、編集を続けることができます。

## レイアウト方法

基本的なレイアウト方法を説明します。

## レイアウトの確認

レイアウト位置やアイテムの動き、ページ遷移の確認は、「Export」ボタンを押すと作成される、export.html ですることができます。export.html は、ELF フォルダの中に作られます。

Chrome でレイアウト (export.html) の確認をする場合は、Chrome.exe に以下の引数を加えて起動させてください。

**\\(chrome のパス )\chrome.exe --allow-file-access-from-files**

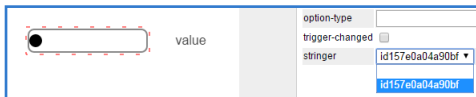
※ export.html からは、製品と MIDI の通信はできません (P.2 参照)。

## アイテムのバリューをストリンガーに表示

スライダーやノブなどの値を表示するには、Stringer アイテムを使用します。

ここでは、スライダーの値をストリンガーに表示する方法を例に説明します。

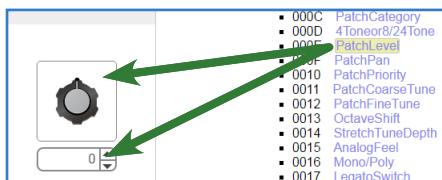
1. Slider と Stringer を配置します。
2. スライダーの「stringer」プロパティに追加した stringer の id を設定します。



3. export.html を表示し、レイアウトを確認します。

## アイテムのバインド

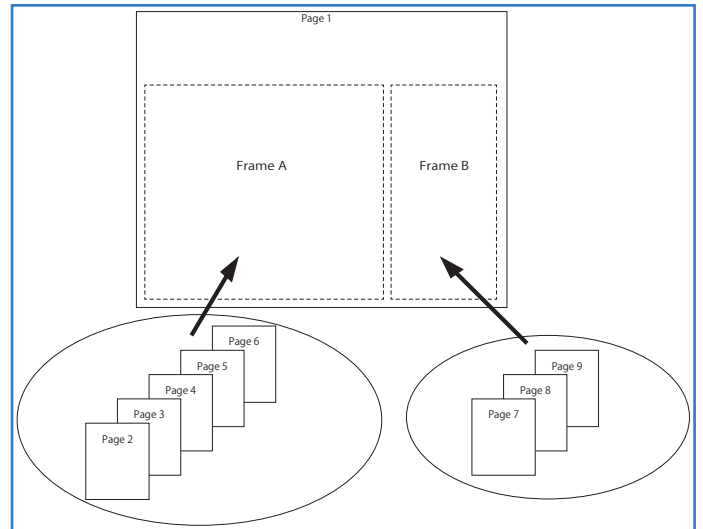
スピナーとスライダーの組み合わせなど、複数のアイテムで同じパラメーターをコントロールしたい場合は、それぞれのアイテムの「parameter-id」プロパティに同じパラメーターを設定します。



## フレームとページの概念

E.Builder では、アイテムを配置したものを『ページ』と呼び、ページを表示する位置と領域を指定するものを『フレーム』と呼びます。

フレームには、複数のページを紐付けることができ、Group ボタンや Menu button などのアイテムを使いページ遷移をコントロールします。



## フレームとページの紐付け

フレームとページを紐付け、フレーム内にページを表示させます。

1. Page1 に Frame を配置します。  
※ Frame の id は、管理しやすいように変更することができます。
2. [Pages] ボタンを押して、ページ・エリアを表示し、+ ボタンでページを追加します。
3. 追加したページをクリックし、アイテムを配置します。
4. 「target-frame」プロパティに、紐付けるフレームの id を設定します。  
選択した id のフレームとページが紐付きます。
5. export.html を表示し、レイアウトを確認します。

### Tips

デフォルトでは、Frame の位置がわかりやすいように「background-color (背景色)」と「opacity (透過度)」が設定されています。

それぞれの欄を空白にすると、フレームが透明になり、紐付いたページの色合いがそのまま表示されます。

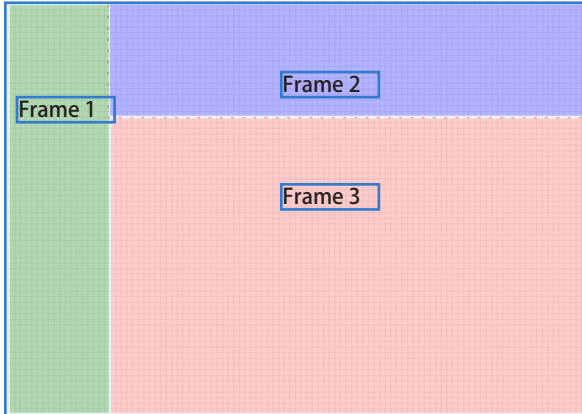
## 1つのページに複数のページを表示する

複数のフレームを配置し、それぞれページを紐付けすることで、1つのページの中に複数のページを表示させることができます。

ここでは、1つのページに3つの異なるページを表示する方法を例に説明します。

1. E.Builder を起動し、下図のように Page1 にフレームを3つ配置します。

※ 図のフレームは、色を個別に設定してあります。



2. 配置したフレームの「id」プロパティをそれぞれ、「F1」、「F2」、「F3」に変更します。

※ id は、フレームの管理をしやすいよう、変更しています。

3. [Pages] ボタンを押して、ページ・エリアを表示し、+ ボタンでページを3つ (Page2 ~ 4) 追加します。
4. Page2 をクリックし、任意のアイテムを配置します。
5. Page2 の「target-frame」プロパティに「F1」を選択します。

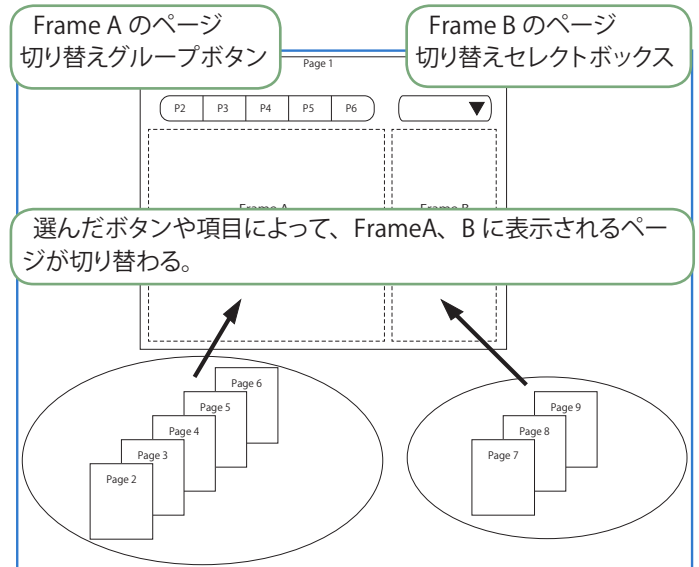
id	Page2
width	800
height	600
style	page-flexible-style ▼
target-frame	F1 ▼
background	F1
background-color	F2
background-image	F3

Page2 が F1 のフレームに紐付き、F1 の領域内に表示されるようになります。

6. Page3、Page4 にもアイテムを配置し、Page3 の「target-frame」プロパティを「F2」、Page4 の「target-frame」プロパティを「F3」に設定します。
7. export.html を表示し、レイアウトを確認します。

## ページの切り替え (ページ遷移)

アイテム、フレーム、ページを紐付けることで、任意のページに切り替えることができます。



## アイテム、フレーム、ページの紐付け

ページの切り替えをするには、遷移を発生させるためのアイテムとフレーム、フレームとページを紐付けます。

1. Page1 に Frame を配置します。
2. Group button など「page-selector」プロパティを持つアイテムを配置します。
3. 「page-selector」プロパティを持つアイテムの「resource-id」を設定します。

アイテムに resource-id の内容が反映されます。



※ resource-id で設定した要素の数だけ、ページを切り替えることができます。要素が4つある resource-id の場合は、4つのページの切り替えるが可能です。

4. 「page-selector」プロパティでフレームの id を選択します。



※ アイテムと選択した id のフレームが紐付きます。

5. [Pages] ボタンを押して、ページ・エリアを表示し、+ ボタンでページを作成します。

6. 作成したページの「target-frame」に、紐付けるフレームの id を設定します。

※ ページと選択した id のフレームが紐付きます。

7. 手順 5、6 を繰り返して、ページを作成と紐付けをします。

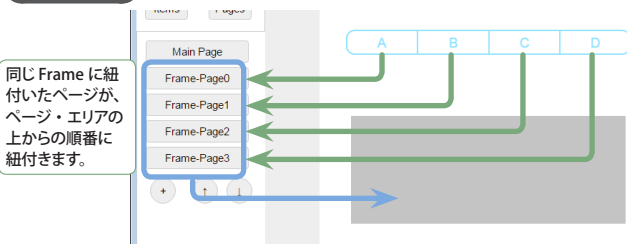
8. export.html を表示し、レイアウトとページの遷移を確認します。

## 切り替わる（遷移する）ページを変更する

ページをフレームに紐付けると、ページ・エリアでの表示が上側にあるページから順にフレームと紐付きます。

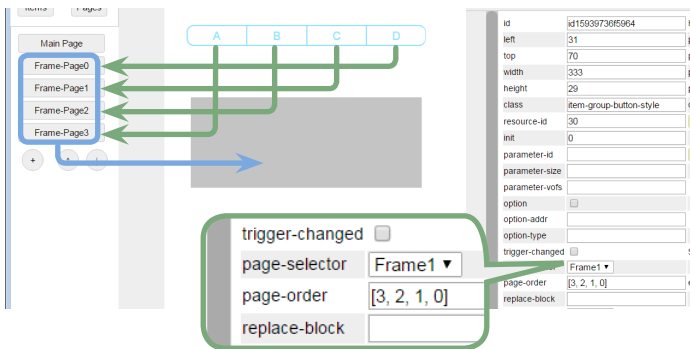
また「page-selector」プロパティを持つアイテムは、デフォルトでは左側（または上側）から順に紐付いたページを切り替えます。

### デフォルト



切り替わるページを変更したい場合は、アイテムの「page-order」プロパティで、ページを指定します。

### 変更例



## ポップアップ画面の作り方

1. 「Push button」を配置します。

※ 必要に応じて、ボタン名を変更します。

2. [Pages] ボタンを押して、ページ・エリアを表示し、+ ボタンでポップ・アップ用のページを追加します。

3. 追加したページにアイテムを配置します。

4. 「style」プロパティに「page-popup-style」を設定します。

5. ポップ・アップ用ページに、ポップ・アップを閉じるための「Push button」を配置します。

6. 配置した Push button の「popup」プロパティに「--close」を設定します。

7. 手順 1 で配置した Push button の「popup」プロパティにポップ・アップ用のページを設定します。

8. export.html を表示し、レイアウトとページの遷移を確認します。

## レンジ・スライダーの作り方

キー・レンジのように、任意に上限と下限を変更し範囲を指定するレンジ・スライダーは、Bar アイテムを 2 つ使って作成します。

33

111

1. Bar アイテムを 2 つ配置します。

2. 配置したアイテムの大きさや位置をを調整し、1 本のスライダーに見えるように並べます。



3. 左側（縦型の場合は下側）のアイテムを選択し、「parameter-id」を設定します。

4. 「lower-than」プロパティに右側（縦型の場合は上側）のアイテムの id を設定します。

init	0
lower-than	RangeHigh ▼
upper-than	▼
linking	<input checked="" type="checkbox"/>
value-hidden	<input type="checkbox"/>

5. 「linking」プロパティにチェックを入れます。

6. 右側（縦型の場合は上側）のアイテムを選択し、「parameter-id」を設定します。

7. 「upper-than」プロパティに左側（縦型の場合は下側）のアイテムの id を設定します。

init	0
lower-than	▼
upper-than	RangeLow ▼
linking	<input checked="" type="checkbox"/>
value-hidden	<input type="checkbox"/>

8. 「linking」プロパティにチェックを入れます。

9. 必要に応じて「Value-hidden」プロパティにチェックを入れ、値の表示を消したり、Stringer アイテム（「アイテムのバリューをストリンガーに表示」（P.11）や Spinner を配置します。

10. export.html を表示し、レイアウトと動作を確認します。

## ライン（線）の作り方

ラインは、Frame や Label アイテムの border を利用して作成することができます。

## ローカライズ

あらかじめローカライズ・テキストを用意しておく、ラベル表示やボタン名、ダイアログ・メッセージなどのテキストを OS の言語設定を参照して切り替えることができます。

### ローカライズ・テキストの作成

ローカライズ・テキストは、ELF-config フォルダーにある、message.js ファイルに記述します。

#### message.js 編集時の注意

以下の記述例を参考に、英語と日本語をセットで記述する。

```
Message_en.prototype.IDM_AAA = " 英語テキスト "
Message_ja.prototype.IDM_AAA = " 日本語テキスト "
```

※ AAA の部分は、和文、英文とも同じにします。異なると和英のテキストが対にならず、言語設定に従って切り替わりません。

### ローカライズ・テキストの設定方法

1. Label や Push button など「text」プロパティを持つアイテムを配置します。
2. 「text」プロパティ横のをボタンをクリックし、一覧からテキストを設定します。  
アイテムに設定したテキストが反映されます。

text プロパティに直接入力する場合は、「@IDM\_AAA」のように入れます。

class	item-label-style
text	@IDM_MANUAL
color	

3. 言語切り替えボックスで言語表示を切り替え、日本語テキストと英語テキストが、正しく切り替わるか確認します。



# パラメーターの設定

## パート・パラメーターの作り方

パート・ボリュームやパート・パンなどのパート・パラメーターは、以下の表のように送信するチャンネルが異なるだけで、パラメーターの構成は同じです。

パート (送信 ch)	パラメーター				
1	Level	Pan	Fine	Coarse	etc...
2	Level	Pan	Fine	Coarse	etc...
3	Level	Pan	Fine	Coarse	etc...
・	・	・	・	・	・
・	・	・	・	・	・
・	・	・	・	・	・
16	Level	Pan	Fine	Coarse	etc...

ここでは、パート数のだけページを作るのではなく、パートの切り替えにアイテムを追従させる方法を説明します。

### 重要

E.Builder は、アドレス・マップを参照し「TemporaryTone(1)」のように「()」の付いたアドレス以下を、パート・パラメーターとして認識します。

サンプルの `address_map.js` を参考に、あらかじめアドレス・マップを作成してください。

1. 「replace-block」と「replace-frame」プロパティを持つアイテムを配置します。

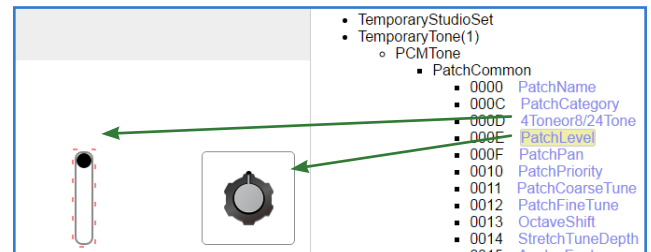
※ 以下、Group button を例に説明します。他のアイテムを配置した場合は、読み替えてください。

2. 配置したアイテムの「resource-id」プロパティを設定します。
3. パート・パラメーターを表示するための Frame を配置します。



4. [Pages] ボタンを押して、ページ・エリアを表示し、+ ボタンでパート・パラメーター用のページを追加します。
5. 追加したページの「target-frame」に、手順 3 で作成したフレームを設定します。

6. 追加したページにアイテムを配置し、「parameter-id」プロパティでパート・パラメーターを設定します。



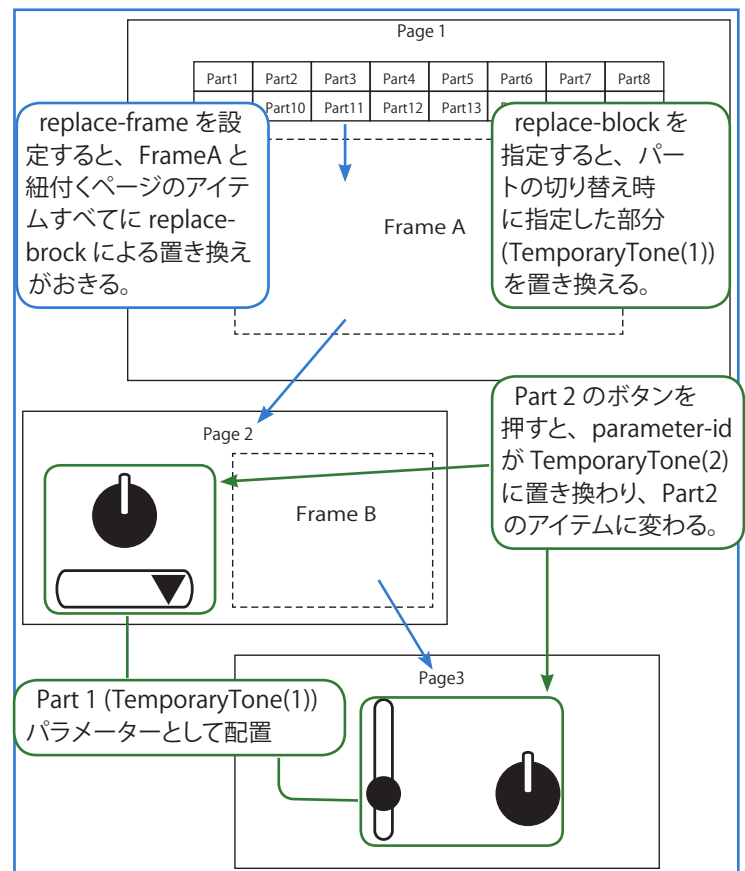
図では、TemporaryTone(1) 以下のパラメーターを選んでいるため、スライダーとノブはパート 1 の PatchLevel と PatchPan のコントローラーになっています。

7. Group button のプロパティを以下のように設定します。

replace-block	TemporaryTone(1)
replace-frame	手順 3 で指定した (=パート・パラメーター用のページを紐付けた) フレーム

replace-block を指定すると、指定したアドレス・ブロックを持つアイテムの「parameter-id」プロパティを、「TemporaryTone(Group button で選択したパート番号)」に置き換えるようになります。

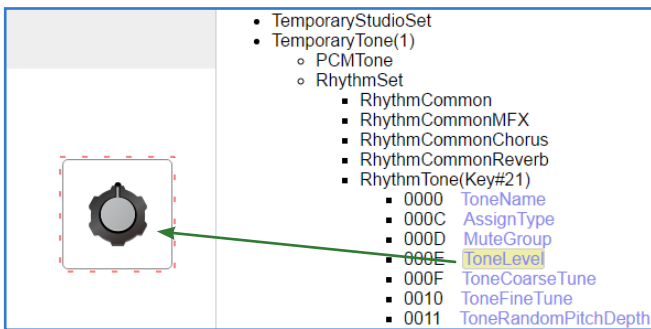
また、replace-frame を指定することで、指定したフレームに紐付くページの全アイテムを置き換えの対象にすることができます。



## キー・パラメーターの作り方

キー・パラメーターも「パート・パラメーターの作り方」(P.15)と同様の手順で作成します。

1. 「replace-block」と「replace-frame」プロパティを持つアイテムを配置します。  
※ 以下、Select box を例に説明します。他のアイテムを配置した場合は、読み替えてください。
2. 配置したアイテムの「resource-id」プロパティを設定します。
3. パート・パラメーターを表示するための Frame を配置します。
4. [Pages] ボタンを押して、ページ・エリアを表示し、+ ボタンでキー・パラメーター用のページを追加します。
5. 追加したページの「target-frame」に、手順 3 で作成したフレームを設定します。
6. 配置したアイテムの「parameter-id」プロパティでリズム・トーンパラメーターを設定します。



7. 「replace-block」と「replace-frame」プロパティを持つアイテムを以下のように設定します。

replace-block	RhythmTone(Key#21) * キー・パラメーターの最初のキーを指定します。
replace-frame	手順 3 で指定した (=キー・パラメーター用のページを紐付けた) フレーム

## MFX パラメーターの割り当て

ここでは、エフェクトの種類やタイプに応じて、レンジやパラメーターが変化する、MFX ページとパラメーターの割り当て方法を説明します。

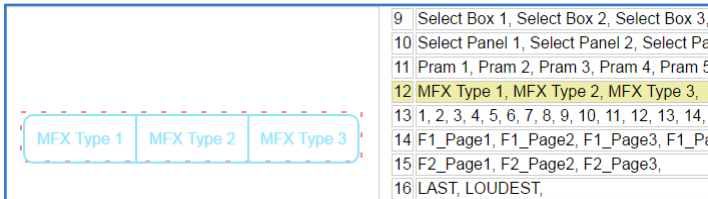
※ MFX パラメーターの割り当てには、アドレス・マップが必要になります。E.Builder 所定の書き方で、アドレス・マップを準備してください。

### ● アドレス・マップ例

```
var PatchCommonMFX = [
  {addr: 0x0000, size: INTEGER1x7, ofs: 0, min: 0, max: 127, init: 0, name: "MFXType"},
  {addr: 0x0011, size: INTEGER4x4, ofs: 32768, min: -20000, max: 20000, init: 0, name: "MFXParameter1"},
  {addr: 0x0015, size: INTEGER4x4, ofs: 32768, min: -20000, max: 20000, init: 0, name: "MFXParameter2"},
  {addr: 0x0019, size: INTEGER4x4, ofs: 32768, min: -20000, max: 20000, init: 0, name: "MFXParameter3"},
  {addr: 0x001D, size: INTEGER4x4, ofs: 32768, min: -20000, max: 20000, init: 0, name: "MFXParameter4"},
]
```

## MFX 用ページの作成

1. [Pages] ボタンを押して、ページ・エリアを表示し、+ ボタンでページを追加します。
2. 追加したページに、Select box（MFX Type を切り替えるためのセレクター）を配置します。
3. Select box の resource-id で、MFX Type のリソースを設定します。



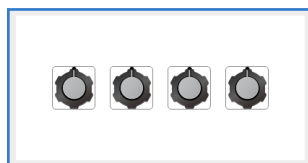
4. Frame を配置し、select box の「page-selector」プロパティに配置したフレームの id を設定します。
5. [Pages] ボタンを押して、ページ・エリアを表示し、+ ボタンで 1 つ目の MFX 用ページを追加します。
6. 追加したページの「target-frame」プロパティに手順 4 で作成したフレームを設定します。

## MFX パラメーターの割り当て

「MFX 用ページの作成」手順 5 で作成したページに、アイテムを配置し、MFX パラメーターの割り当てをします。

1. MFX Type 1 のパラメーターをコントロールするアイテムを配置します。

※ アイテムの数や種類は、実際の MFX に合わせて配置してください。



2. 配置したアイテムの「parameter - id」、「option」、「option-addr」、「option-type」プロパティを設定します。

	Knob 1 (左)	Knob 2	Knob 3	Knob 4 (右)
parameter - id	MFX Parameter1	MFX Parameter2	MFX Parameter3	MFX Parameter4
option	有効 (チェック)	有効 (チェック)	有効 (チェック)	有効 (チェック)
option-addr	0	0	0	0
option-type	0	0	0	0

### ● option-addr :

MFX Type、Chorus Type、Reverb Type などの、タイプを示すパラメーターのアドレス（ブロック先頭からのオフセット）を指定します。

### ● option-type :

option-addr で指定したパラメーターのタイプを指定します。

例：MFX Type の値 0 = Flanger、値 1 = Overdrive など

3. 「MFX 用ページの作成」手順 5 ～ 6 の方法で、2 つ目の MFX 用ページを作成します。
4. 手順 3 で作成したページに、MFX Type 2 のパラメーターをコントロールするアイテムを配置します。
5. 配置したアイテムの「parameter - id」、「option」、「option-addr」、「option-type」プロパティを設定します。

	アイテム 1	アイテム 2	アイテム 3	アイテム 4
parameter - id	MFX Parameter1	MFX Parameter2	MFX Parameter3	MFX Parameter4
option	有効 (チェック)	有効 (チェック)	有効 (チェック)	有効 (チェック)
option-addr	0	0	0	0
option-type	1	1	1	1

※ タイプを示すアドレスは MFXType のままなので、option-addr は「0」、パラメーターのタイプ（エフェクトの種類）は「1」になるので、option-type は「1」に設定します。

6. MFX Type の数だけ手順 3 ～ 5 を繰り返し、ページを作成します。
7. export.html を表示し、レイアウトを確認します。

※ 実機と通信し、パラメーターの送受信を確認する場合は、「ebuilder.exe と quattro(win).exe について」(P.2) を参照してください。

# プロジェクトの再現と納品

E.Builder のプロジェクトを再現するには、以下のファイルとフォルダーを必要とします。

- E.Builder プロジェクト・ファイル (.elf)
- ELF フォルダー一式  
フォルダー：CSS、ebuilder、export、images、js  
ファイル：export.html、ebuilder.html、import.js

社外のデザイナーにデザインの発注をした場合は、上記のファイルとフォルダーを納品してもらいます。

また、次のような場合も左記のファイルとフォルダーのコピーをとってください。

別のプロジェクトを新規作成する場合  
派生のデザイン（バリエーション）を作る場合  
別の環境でプロジェクトを再現する場合  
完了したプロジェクトを保管する場合

## 補足

### カスタム UI の作成

E.Builder では作ることができないカスタム UI が必要な場合は、カスタム UI を配置する領域にフレームを作成し、作成したフレームの id に対して個別の処理、ソースコードを記述します。

### カスタム .js ファイルの読み込み

ELF フォルダーの直下に additional ファイルを作成し、取り込むカスタム .js ファイルのパスを記述しておく、エクスポートする際 (export.html) に、ファイルが読み込みます。

1. ELF フォルダーの直下に additional ファイルを作成し、拡張子を削除します。
2. additional ファイルをテキストエディターで開き、カスタム .js ファイルのパスを記述します。

記述例)

```
js/custom/abc.js  
js/custom/xyz.js
```

3. export.html をエクスポートします。  
additional ファイルに記述したスクリプトが読み込みされます。

### E.Builder が起動しない場合

resource.js、address\_map.js、message.js など、ファイル編集後に、E.Builder が起動しなくなった場合は、ブラウザ (chrome) の開発者向けツールを利用することで、問題のファイルを探し出すことができます。

1. E.Builder を起動します。
2. chrome を起動し、アドレス欄に「localhost:8081」と入力します。
3. 表示されたリンクをクリックします。  
開発者向けツールが起動し、プロジェクトを構成するフォルダーやファイルが表示されます。
4. 「Console」画面などでエラー内容の確認をし、修正します。