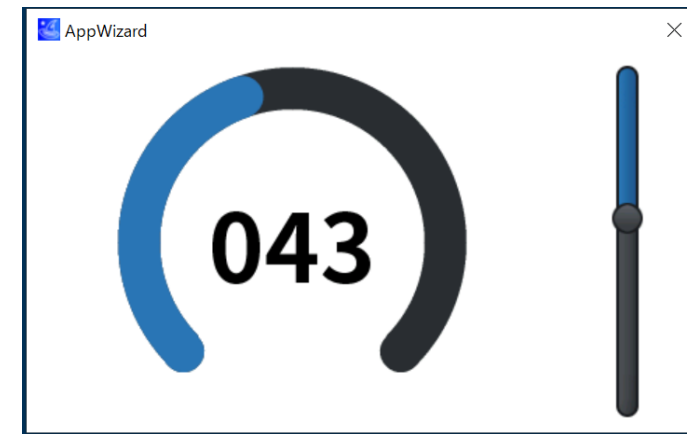


# emWin for RX GUI開発トレーニング (実践1)

---

Using AppWizard  
RX72N Envision 評価ボード用



Embedded Software / Hardware Solutions

試作から量産ツールまで

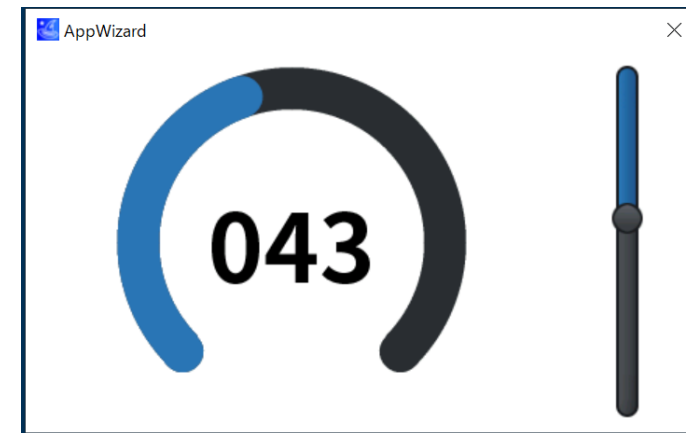
## 「emWin for RX」について

RXグループMCUに対応するSEGGER社製「**emWin for RX**」GUIソフトウェアは、**RXマイコングループ（RXv1、RXv2、RXv3）** システム環境で無償で商用利用し、開発することが可能です。1999年から販売しており、emWinはワールドワイドで産業制御、スマートホーム・省エネ対応IoT機器、テレコム、自動車産業、家電、医療器械等、航空電子機器様々なアプリケーションで使用されています。emWin GUIはRXマイコンデバイスの内蔵RAMだけで使用出来ますので、外付けRAMが不要です。emWin GUIは使用中のOS（RX用embOS、μITRON等）で対応可、OS無しも使用可能です。**ロイヤリティー費用、量産制限等はありません。**

### GUI開発トレーニング（実践1）

emWin AppWizardツールから画面UIをデザインして必要なリソース及びアプリケーションコードを作成します。

**本資料内では、その手順を解説します。**



## 実行手順

1

### 事前準備

開発環境 e2Studio、QE for Display [RX] をインストールします。

QE for Display [RX] のSmart Configuratorを使用してRX72N Envisionボード用BSPパッケージを作成します。

2

### 画面作成

emWin AppWizardツールから画面UIをデザインして必要なリソース及びアプリケーションコードを作成します

3

### ビルド & 動作確認

e2Studioプロジェクト環境でBSP+UIデザインアプリケーションコードをビルドして、RX72N Envisionボード環境で動作を確認します。

## TASK-1

# 事前準備

---

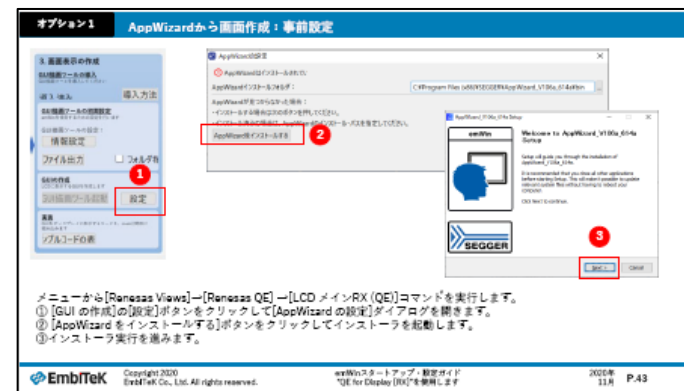
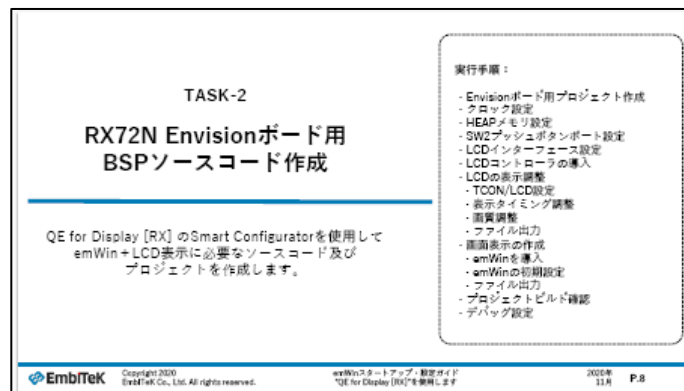
- ① 開発環境 e2Studio、QE for Display [RX] をインストールします。
- ② QE for Display [RX] のSmart Configuratorを使用してRX72N Envisionボード用BSPパッケージを作成します。
- ③ アプリケーションで使用するフォントをパソコンにインストールします。

本セッションでは以下の開発環境を使用します。

- Host PC: Windows10 / Windows11
- 統合開発環境 e2 studio 2023-10
- ディスプレイ対応開発支援ツール パッケージ QE for Display V3.1.1

# 事前準備：BSP+ベースプロジェクトの作成

開発環境 e2Studio、QE for Display [RX] のインストール及びSmart Configuratorを使用してRX72N Envisionボード用BSPパッケージの作成手順は別資料「**emWin for RXスタートアップ・設定ガイド**」（ファイル名：emWin\_for\_RX72N\_qedisp\_startup.pdf）をご参照ください。  
[https://www.embitek.co.jp/technote/emwin/emWin\\_for\\_RX72N\\_qedisp\\_startup.pdf](https://www.embitek.co.jp/technote/emwin/emWin_for_RX72N_qedisp_startup.pdf)



## 関連ドキュメント・参考資料：

- RXファミリ QE for Display [RX] アプリケーションノート

<https://www.renesas.com/jp/ja/document/apn/rx-family-qe-display-rx-application-note>

本セッションではグーグル社無償提供の「BIZ UDPGothic」  
モリサワのBIZ UDゴシックフォントを使用します。

**BIZ UDPGothic** (Google)

<https://fonts.google.com/specimen/BIZ+UDPGothic>

他、商用ライセンスフォントの  
ご利用が可能です。

① 以下URLから「BIZ UDPGothic」フォントパッケージ  
をダウンロードします。

<https://fonts.google.com/download?family=BIZ+UDPGothic>

② ZIPパッケージの (\*.ttf) フォントをマウス右クリック  
メニューからパソコンにインストールします。



BIZUDPGothic-Bold.ttf  
BIZUDPGothic-Regular.ttf  
OFL.txt

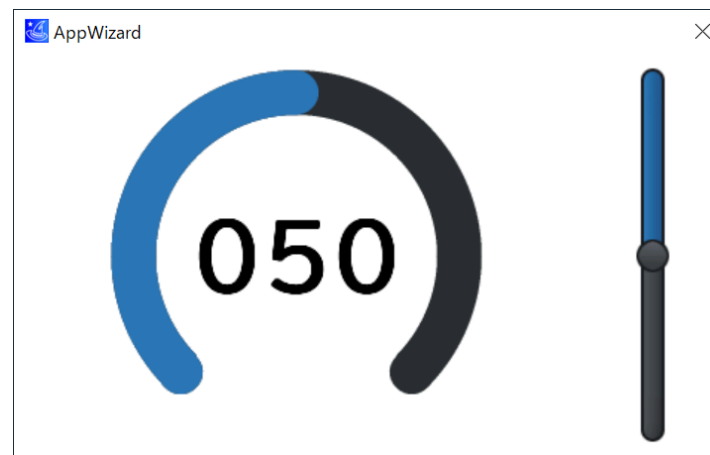
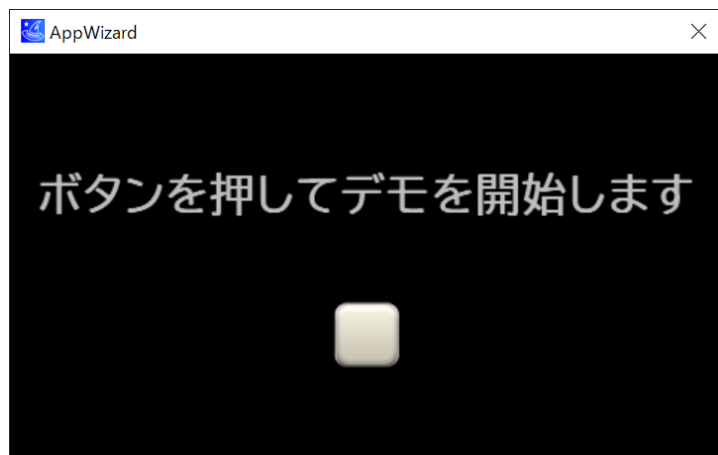
上記フォントの御社製品利用については、  
各ライセンスの利用条件をご確認の上、ご利用ください。

## TASK-2

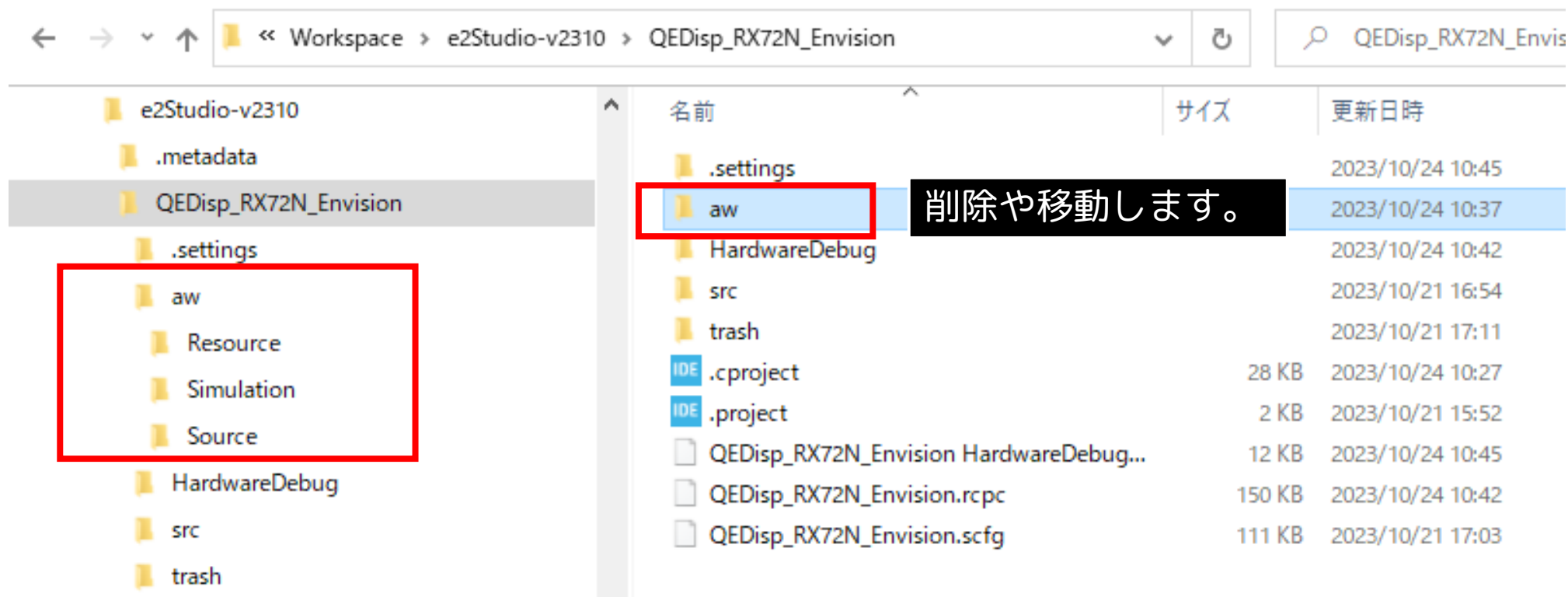
# 画面作成（UIデザイン）

---

emWin AppWizardツールから画面UIをデザインして必要なリソース  
及びアプリケーションコードを作成します



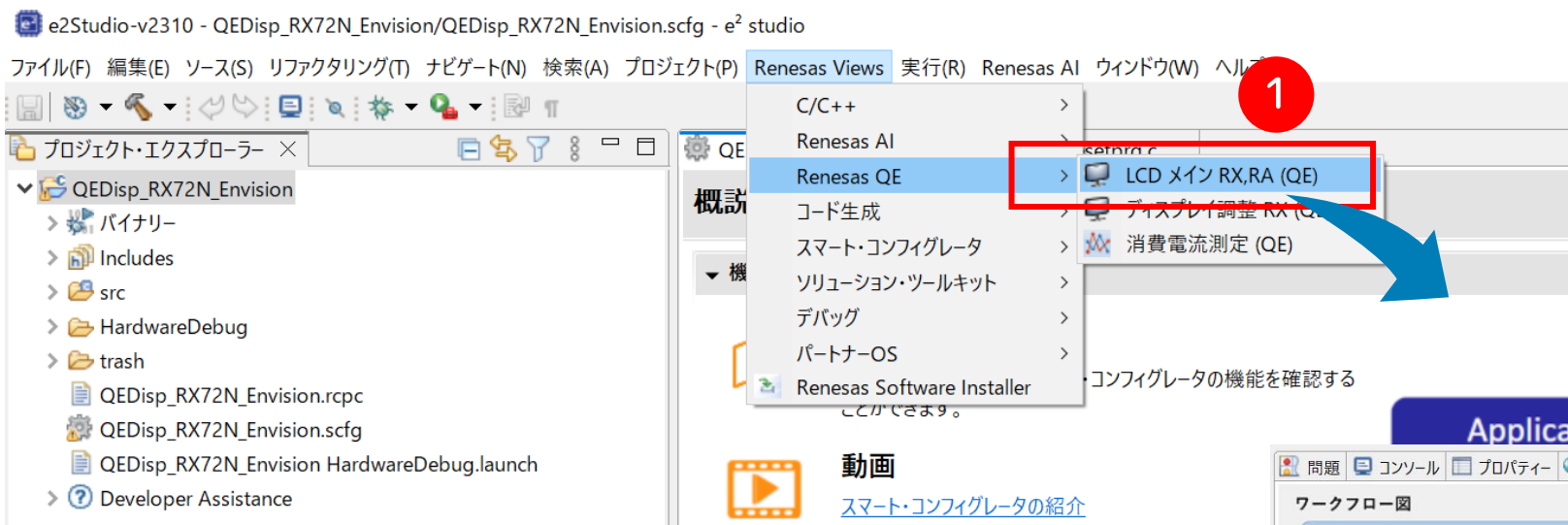
e2Studio QE環境でのデフォルト「aw」AppWizardプロジェクトを使用しますので、  
e2Studioワークスペースの既存の「aw」フォルダを削除するか別のローカルディレクトリに移動してください。



既存の「aw」フォルダを削除や移動してからe2Studioツールを起動します。



# 画面作成：AppWizardツールを起動します



- ① e2StudioからRX72N Envisionボード用プロジェクトを起動し、メニューから[Renesas Views]→[Renesas QE] → [LCD メインRX (QE)]コマンドを実行します。
- ② [GUI 描画ツールの起動]ボタンをクリックしてAppWizardツールを起動します。

# 画面作成：スクリーン作成

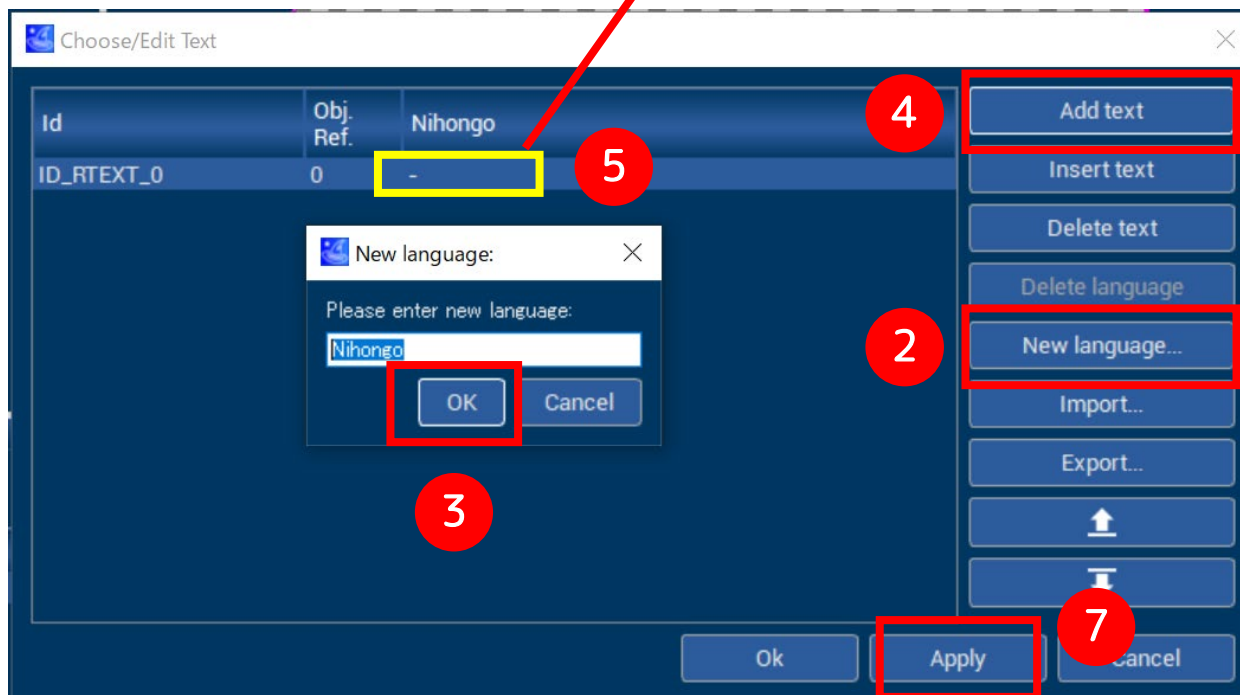
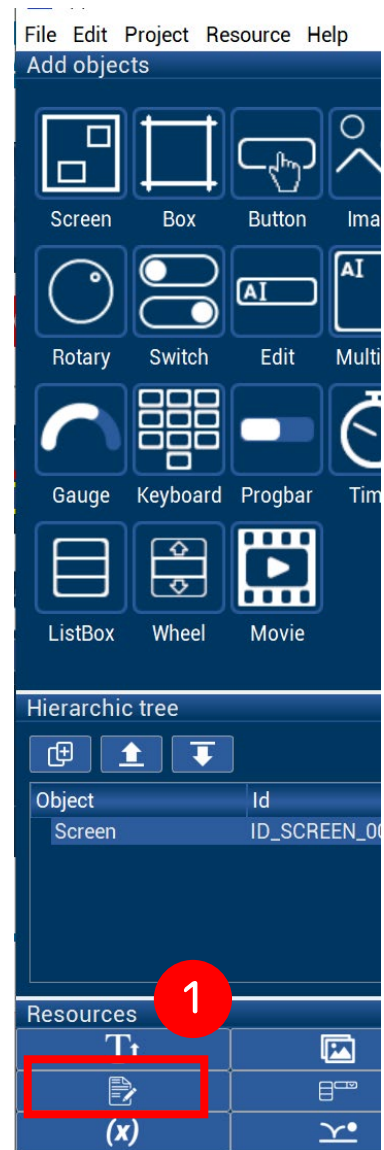
The screenshot displays the AppWizard V1.36a\_6.32a - \*aw application window. The interface is divided into several panels:

- Add objects:** A grid of icons for various UI elements. The 'Screen object' icon (a square with a smaller square inside) is highlighted with a red box and a red circle containing the number '1'. A yellow arrow points from this icon to the 'Screen' entry in the Hierarchic tree.
- Hierarchic tree:** A list of objects in the project. The 'Screen' object with ID 'ID\_SCREEN\_00' is highlighted with a yellow box.
- Editor:** A central workspace showing a checkered pattern representing the screen. A yellow arrow points from the 'Screen object' icon to this workspace.
- Properties:** A panel on the right showing the properties of the selected 'Screen' object. It includes fields for Id (ID\_SCREEN\_00), Position (0, 0), Size (480, 272), and various motion and interaction settings.
- Interactions:** A panel at the bottom showing a table for defining interactions between objects.

A text box at the bottom of the Editor panel contains the following text:

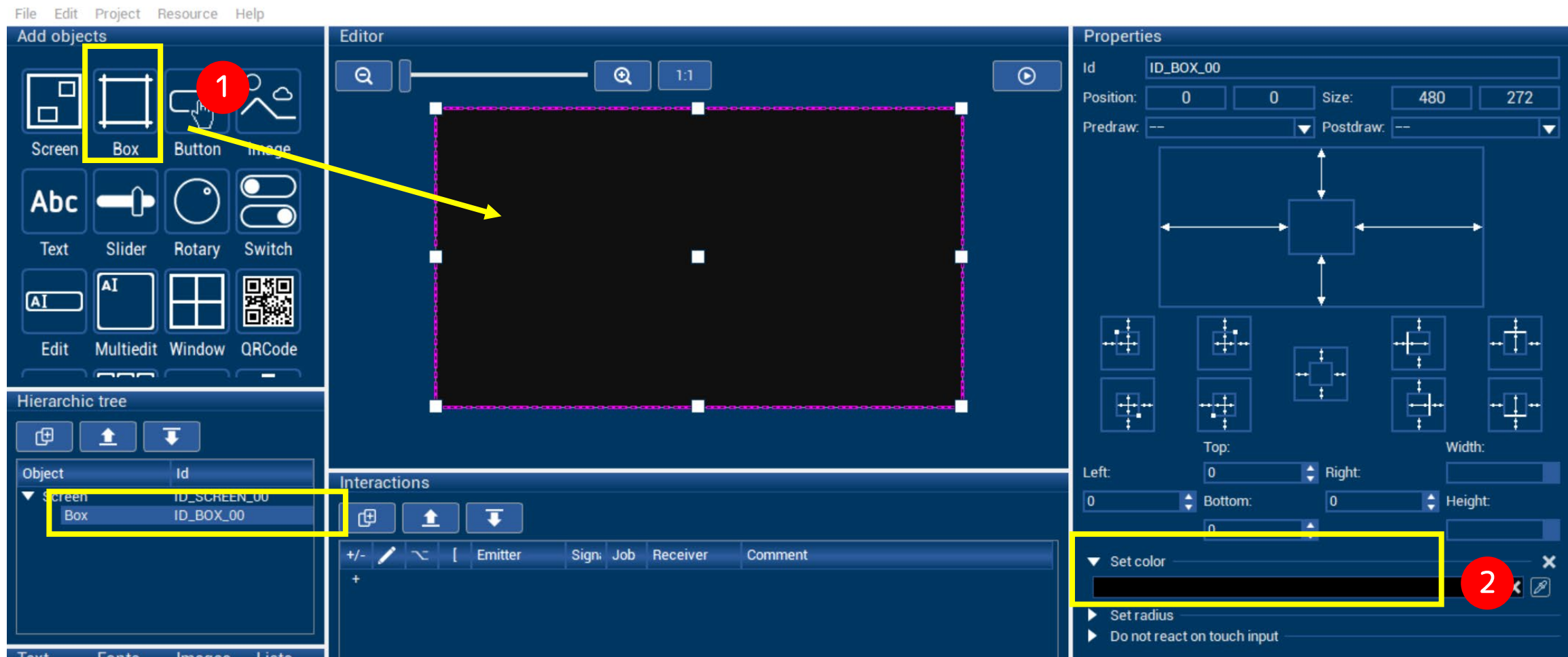
Screenボタンを押して画面にスクリーンオブジェクトを追加します

## 画面作成：日本語テキストリソースを追加



- ① Textリソースボタンを押して、
- ② Textダイアログから「New Language」ボタンを押して、
- ③ 言語名「Nihongo」を追加します。
- ④ 「Add text」ボタンを押して ID\_RTEXT\_0オブジェクトを追加します。
- ⑤ TextダイアログのID\_RTEXT\_0オブジェクトの「Nihongo」言語項目下の「…」をマウスダブルクリックして
- ⑥ 以下の文字列を設定します。  
**「ボタンを押してデモを開始します」**
- ⑦ 設定完了後に「Ok」ボタンを押して Textダイアログを閉じます。

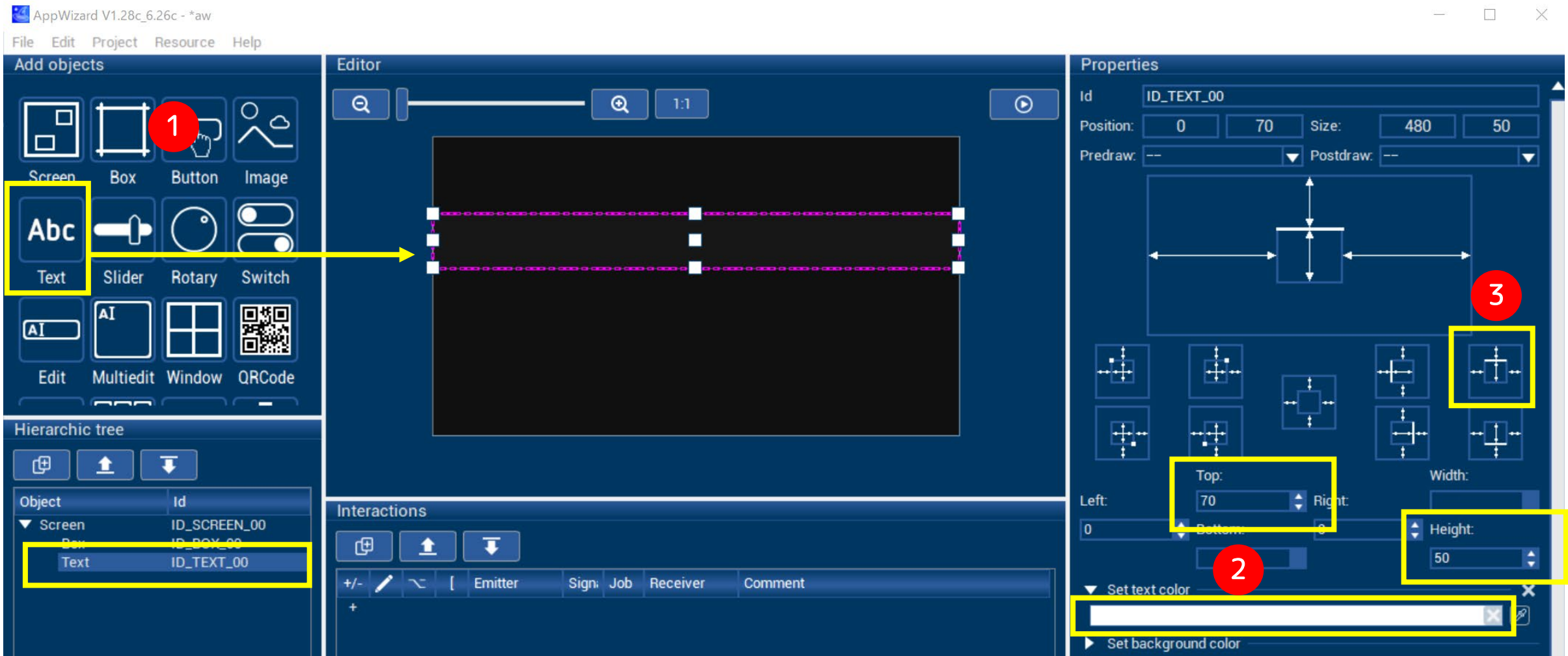
## 画面作成：黒色バックグラウンドを設定



- ① Boxボタンを押して画面にバックグラウンドオブジェクトを追加します。
- ② Colorを黒色に設定します。



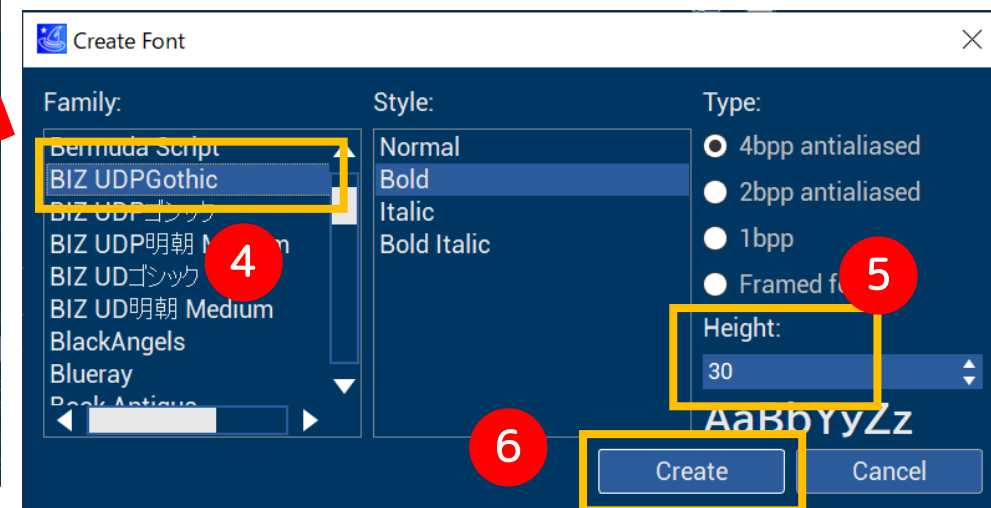
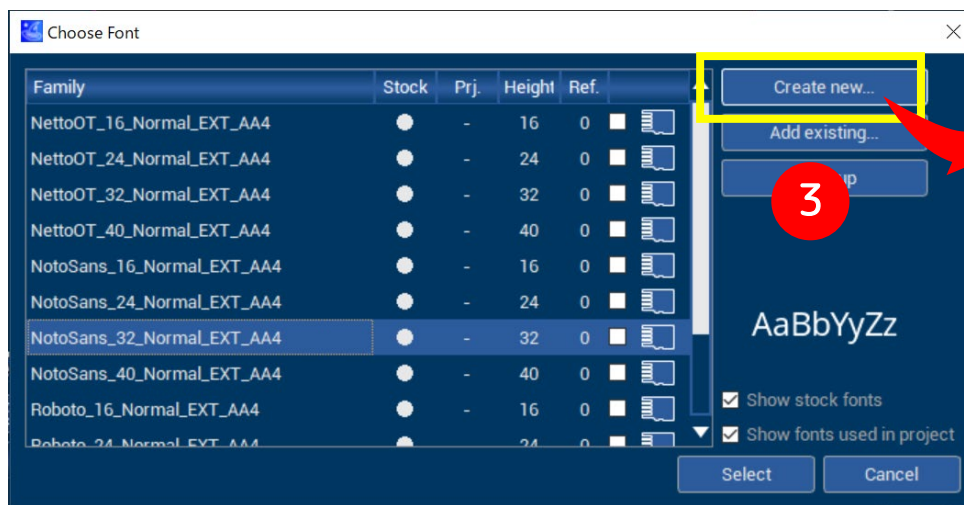
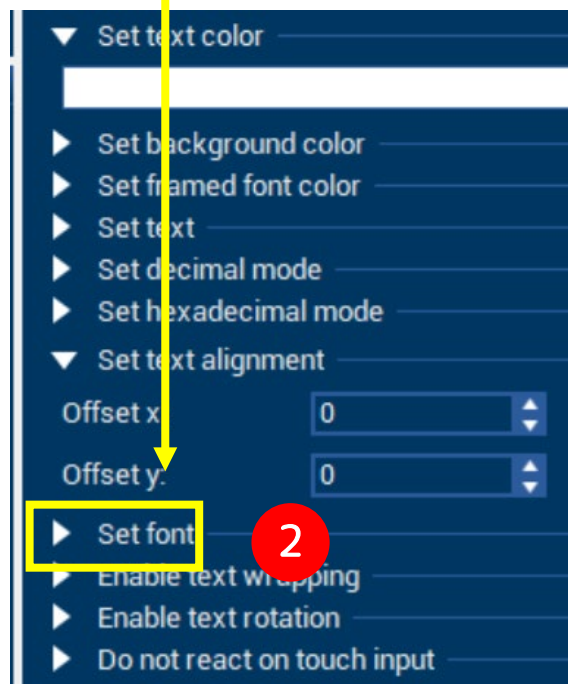
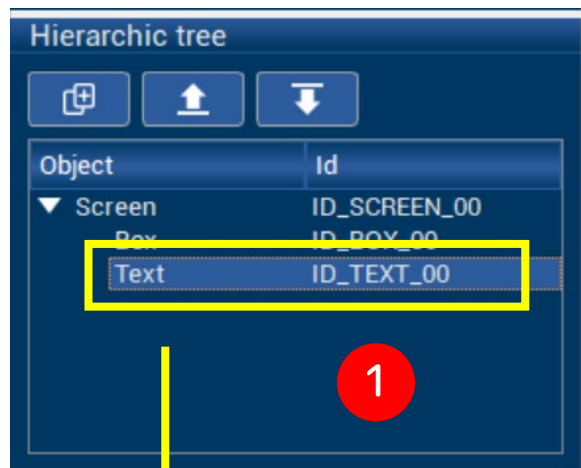
## 画面作成：テキストオブジェクトの追加



- ① **Text**ボタンを押して画面にテキストオブジェクトを追加します。
  - ② **Color**を白色に設定します。
  - ③ **アラインメント**を設定します。  
TOP = 70, Height = 50, Right = 0

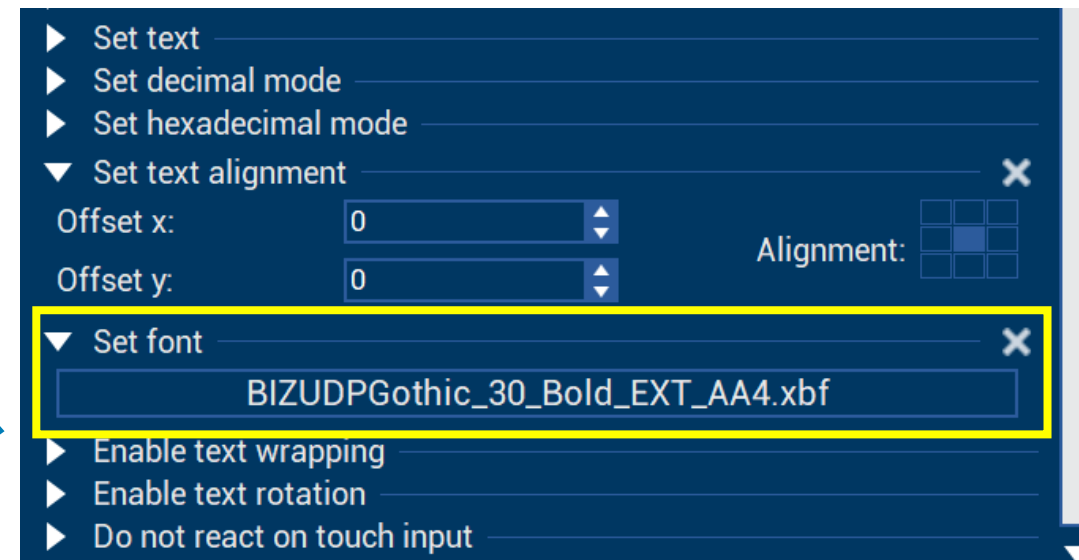
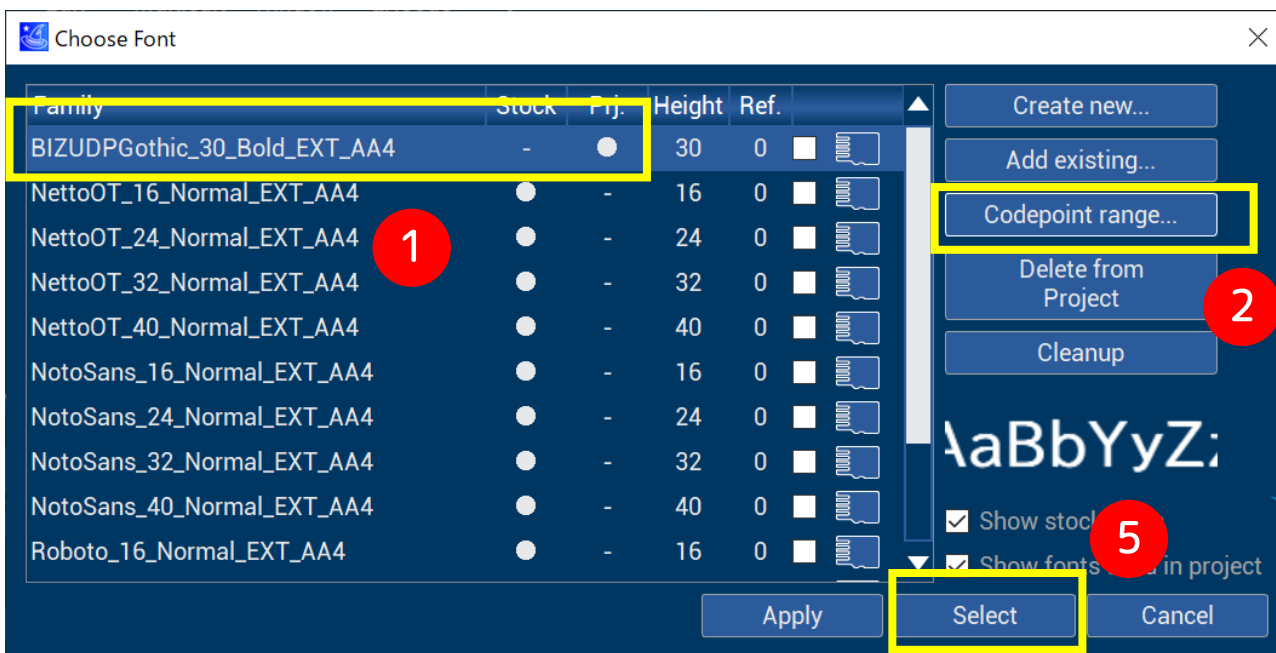
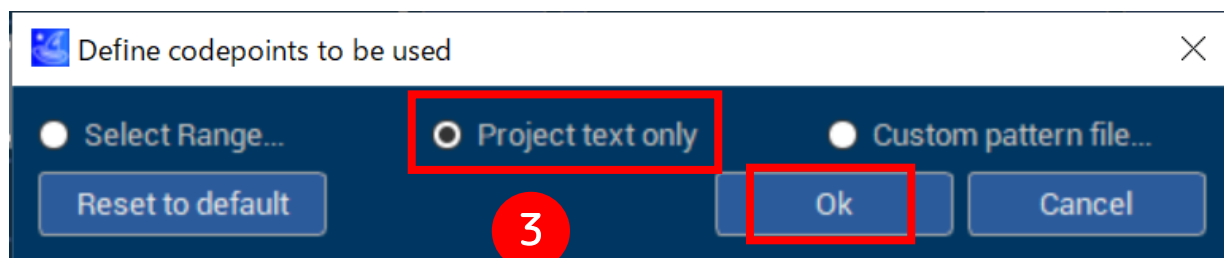
## 画面作成：テキストオブジェクト（ID\_TEXT\_00）の日本語フォントを作成

- ① 「ID\_TEXT\_00」を選択します。
- ② Propertyウィンドウから「Set font」アイコンをクリックしてフォント設定ダイアログを開きます。
- ③ 「Create New」ボタンを押してフォント作成ダイアログを開きます。
- ④ 「BIZ UDPGothic / Bold」フォントを選択
- ⑤ フォントHeight=30を設定します。
- ⑥ 「Create」ボタンを押してダイアログを閉じます。

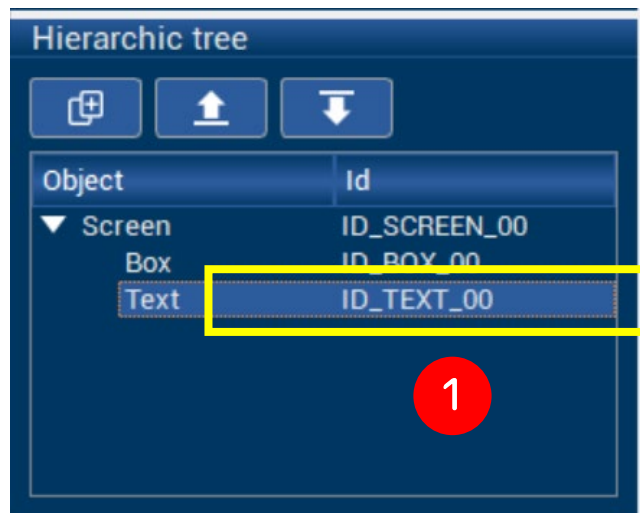


## 画面作成：テキストオブジェクト（ID\_TEXT\_00）の日本語フォント設定

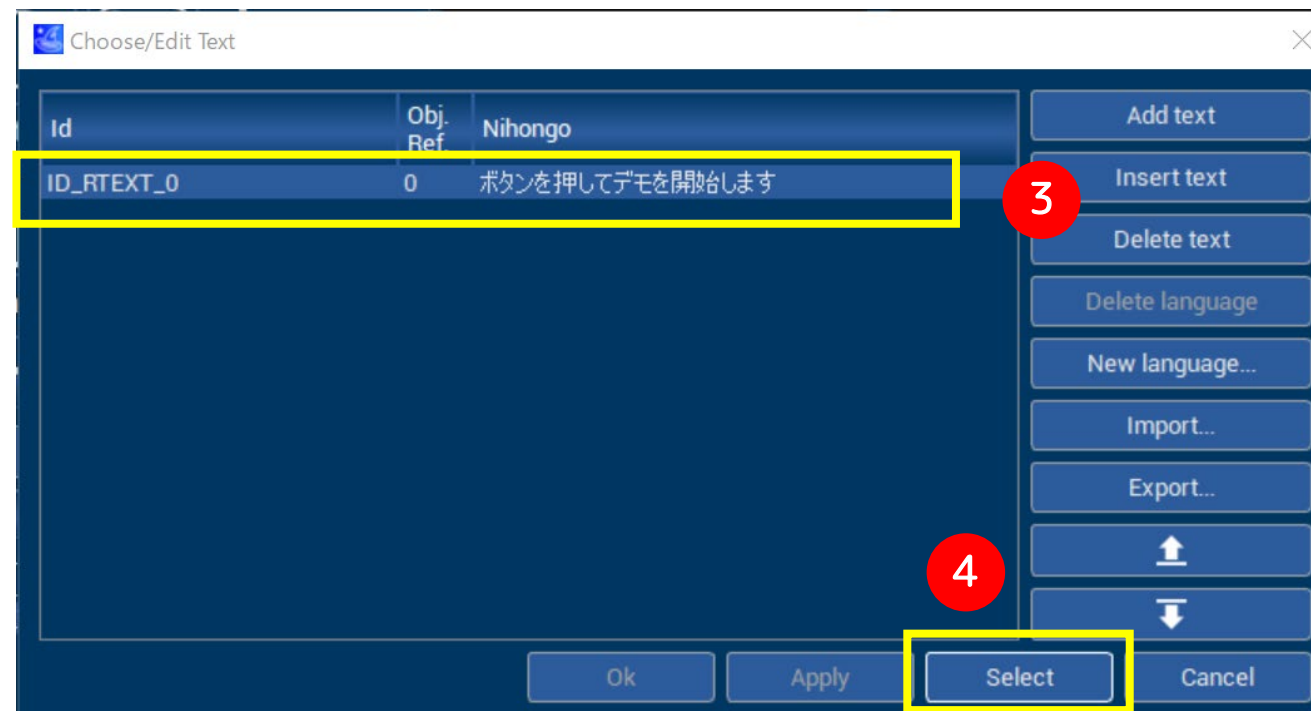
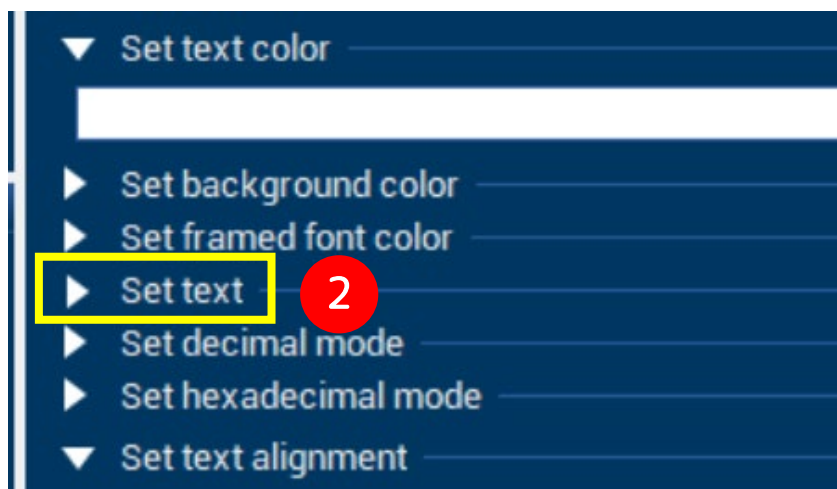
- ① 「BIZUDPGothic\_30\_Bold\_…」フォントを選択します。
- ② 「Codepoint range」ボタンを押して、
- ③ 「Project text only」オプションを選択してプロジェクトで使用する文字分のみフォントデータを作成するように設定します。
- ④ 「OK」ボタンを押して設定を保存
- ⑤ 「Select」ボタンを押してダイアログを閉じます。



## 画面作成：テキストオブジェクト（ID\_TEXT\_00）に文字列を設定

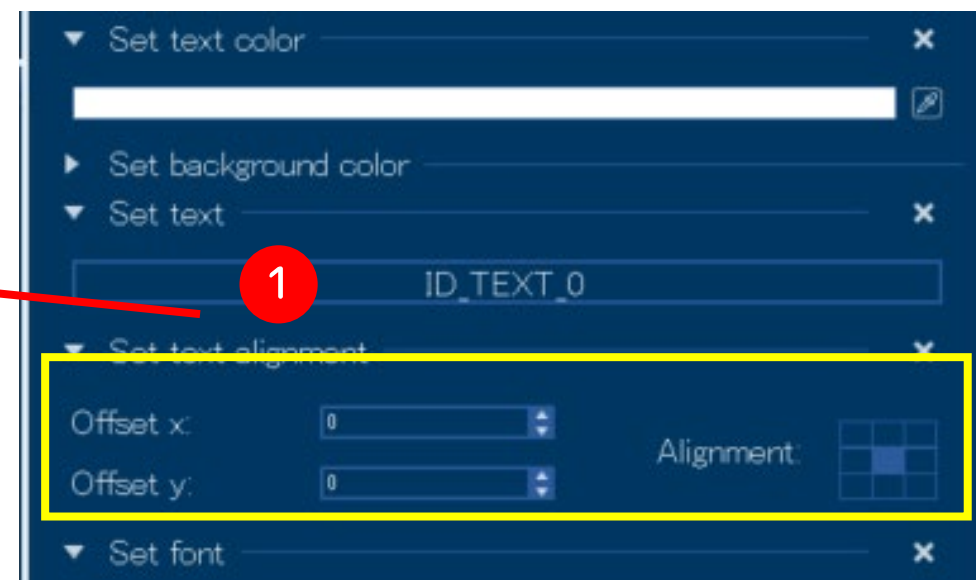
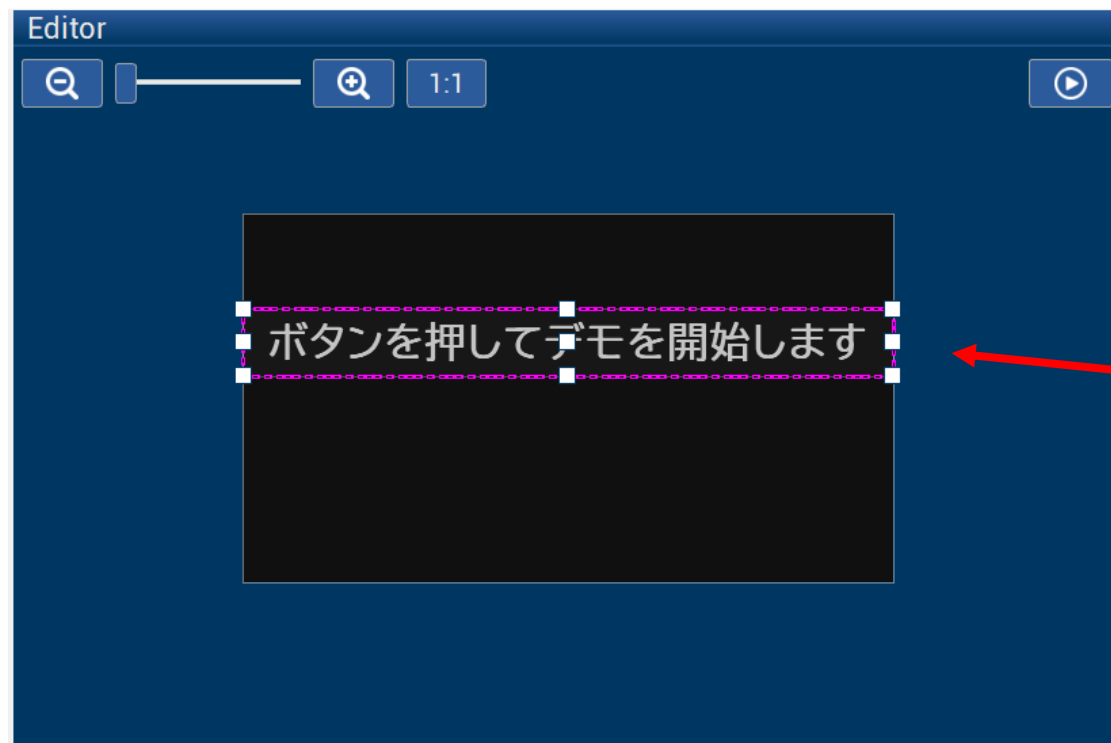


- ① 「ID\_TEXT\_00」を選択します。
- ② Propertyウィンドウから「Set Text」アイコンをクリックText設定ダイアログを開きます。
- ③ 「ボタンを押してデモを開始します」文字列を選択し
- ④ 「Select」ボタンを押してダイアログを閉じます。



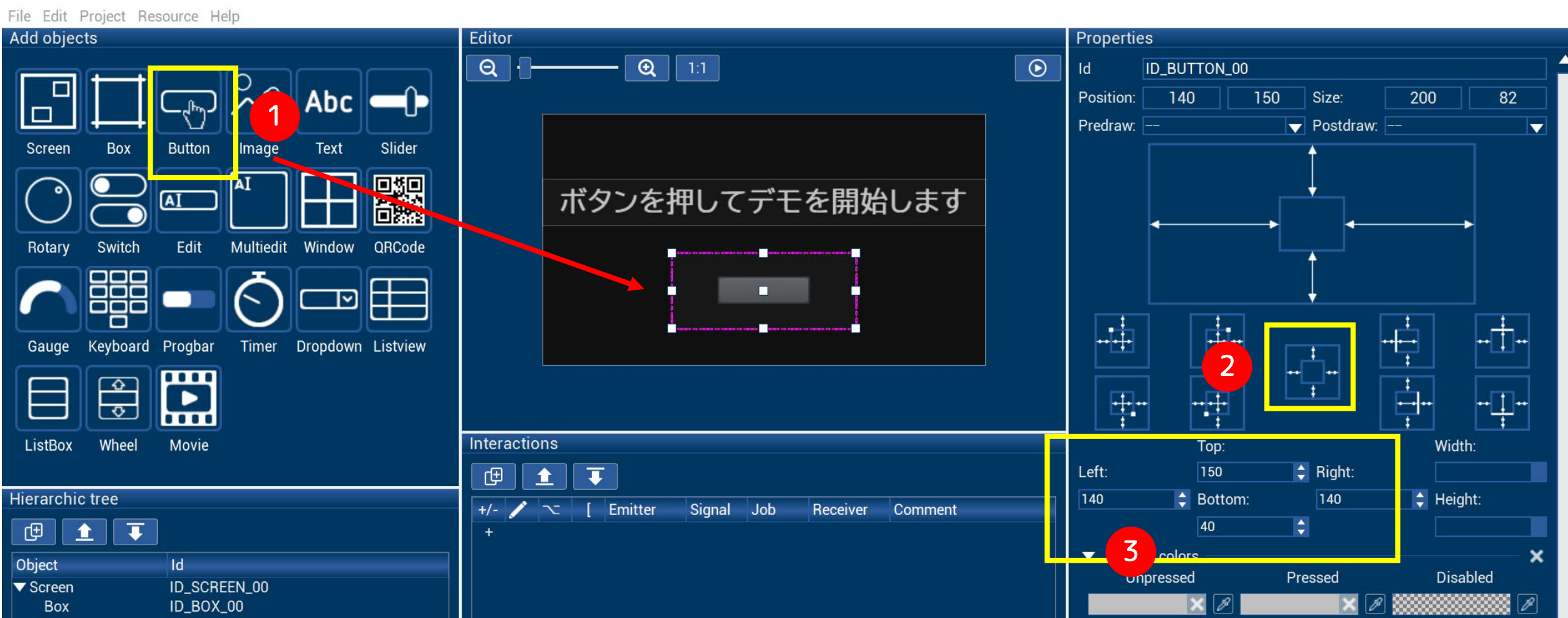


## 画面作成：テキストオブジェクト（ID\_TEXT\_00）のアラインメント設定



テキストを真ん中アラインメントに設定します。

# 画面作成：ボタンオブジェクトの追加



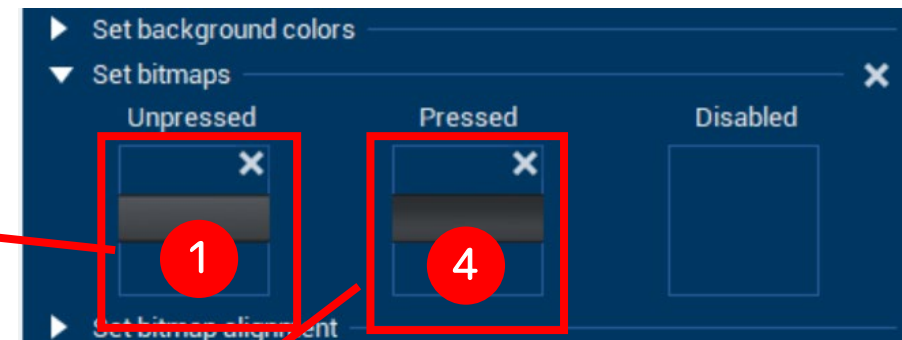
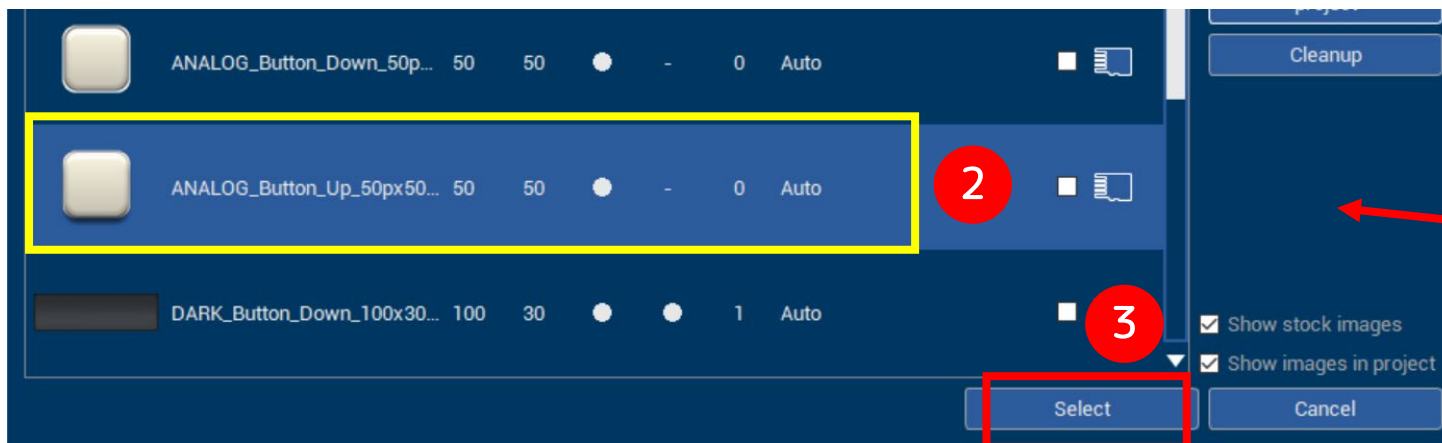
① Buttonボタンを押して画面にボタンオブジェクトを追加します。

② 画面の真ん中アラインメントに設定します。

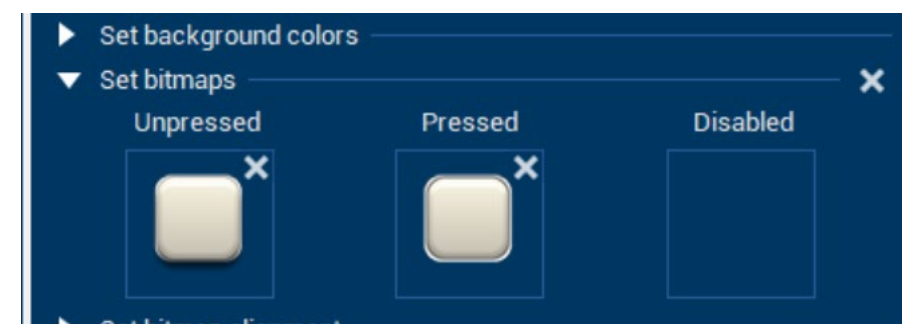
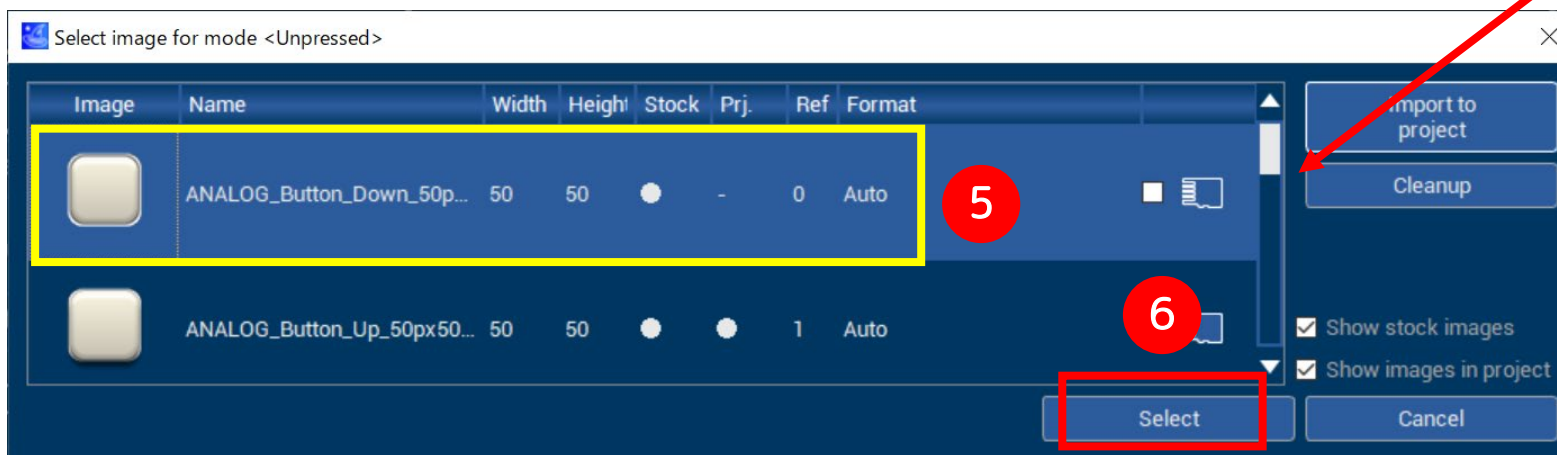
③ アラインメントを設定します。

TOP = 150, Bottom = 40,  
LEFT = 140, RIGHT = 140

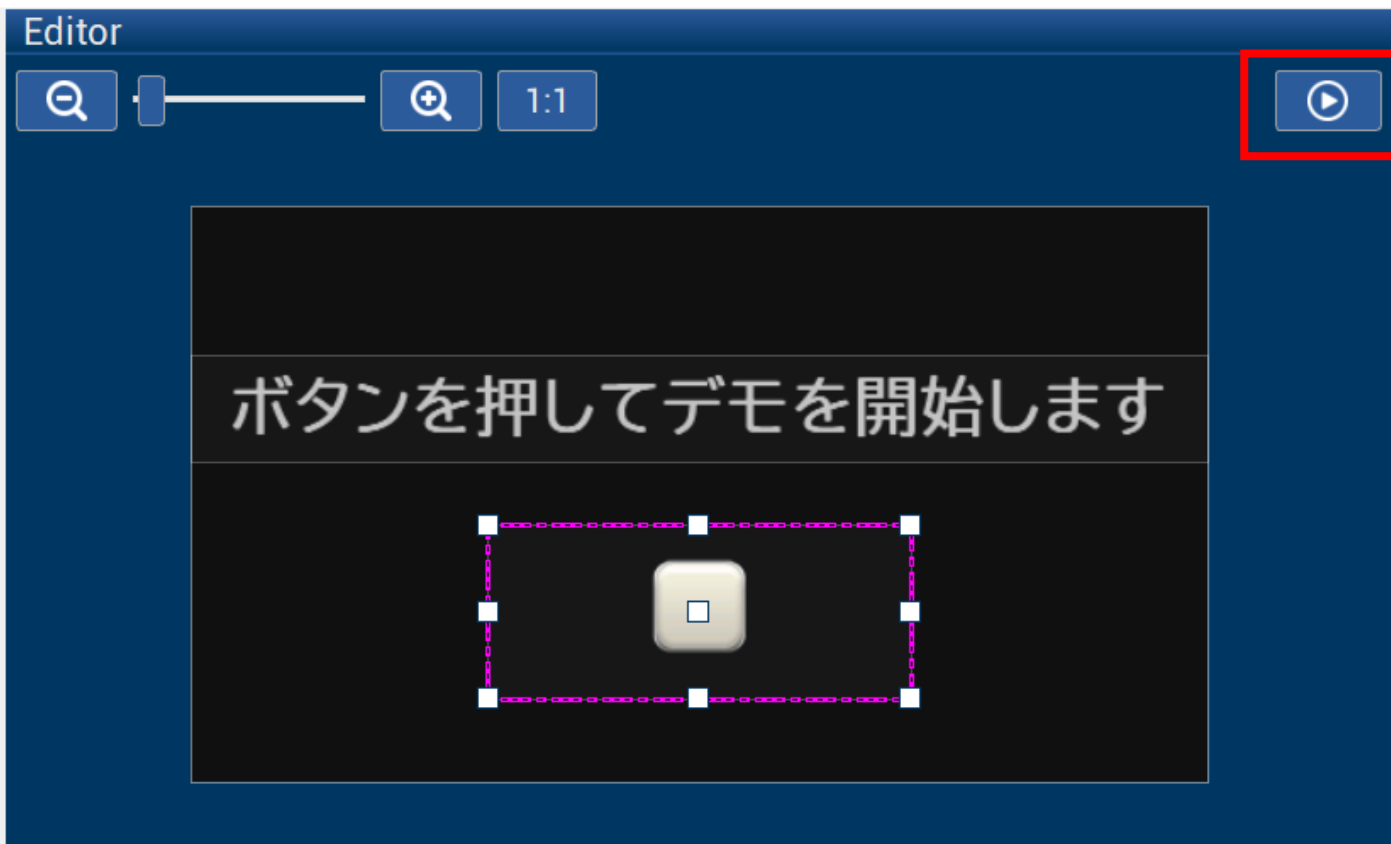
## 画面作成：ボタンオブジェクトの画像設定



Unpressed状態のビットマップボタン（50px x 50px）のイメージを選択します



Pressed状態のビットマップボタン（50px x 50px）のイメージを選択します



- ① AppWizardのPlayモードシミュレーションでボタン操作の動作確認を行います。
- ② 確認後にPlayモードを終了します。

# 画面作成：新しいスクリーン（Screen2）を追加

File Edit Project Resource Help

Add objects

Screen Box Button Image

1

Text Slider Rotary Switch

Edit Multiedit Window QRCode

Hierarchic tree

Object	Id
Screen	ID_SCREEN_00
Box	ID_BOX_00
Text	ID_TEXT_00
Button	ID_BUTTON_00
Screen	ID_SCREEN_01

Editor

Interactions

+	-	Emitter	Signal	Job	Receiver	Comment

Properties

Id: ID\_SCREEN\_01

Position: 0 0 Size: 480 272

Predraw: --- Postdraw: ---

Top: 0 Right: 0

Left: 0 Bottom: 0 Height: 0

Set horizontal motion

Set vertical motion

Set persistent



## 画面作成：白色バックグラウンドを設定

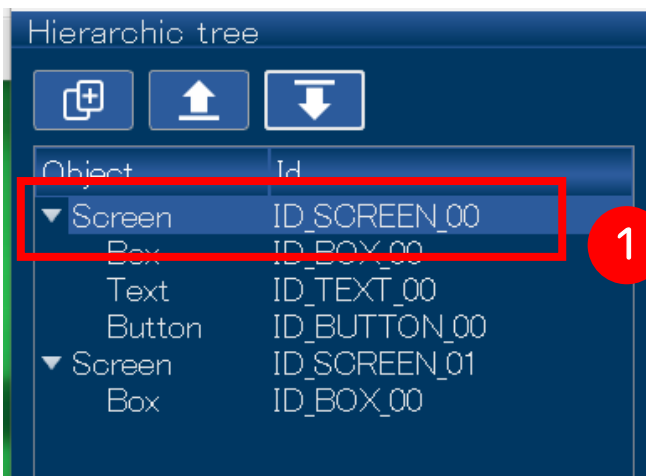
The screenshot displays the emWin GUI development software interface, which is divided into several panels:

- Add objects:** A grid of icons for various GUI elements. The **Box** icon is highlighted with a yellow box and a red circle with the number 1. A yellow arrow points from this icon to the Editor panel.
- Editor:** The central workspace showing a white rectangular box with a dashed border and handles, representing the background object.
- Properties:** A panel on the right showing the properties of the selected object (ID\_BOX\_00). The **Set color** option is highlighted with a yellow box and a red circle with the number 2.
- Hierarchic tree:** A panel on the bottom left showing the object hierarchy. The **Box** object under the selected screen is highlighted with a yellow box.
- Interactions:** A panel at the bottom showing the interaction table.

- ① Boxボタンを押して画面にバックグラウンドオブジェクトを追加します。
- ② Propertyウィンドウから「Set color」アイコンをクリックして白色を設定します。

## 画面作成：スクリーン1への画面移動ジョブの設定

インタラクション（Interactions）・ジョブ機能でScreen0→Screen1への画面移動を設定します。



① オブジェクトウィンドウから「ID\_SCREEN\_00」を選択します。

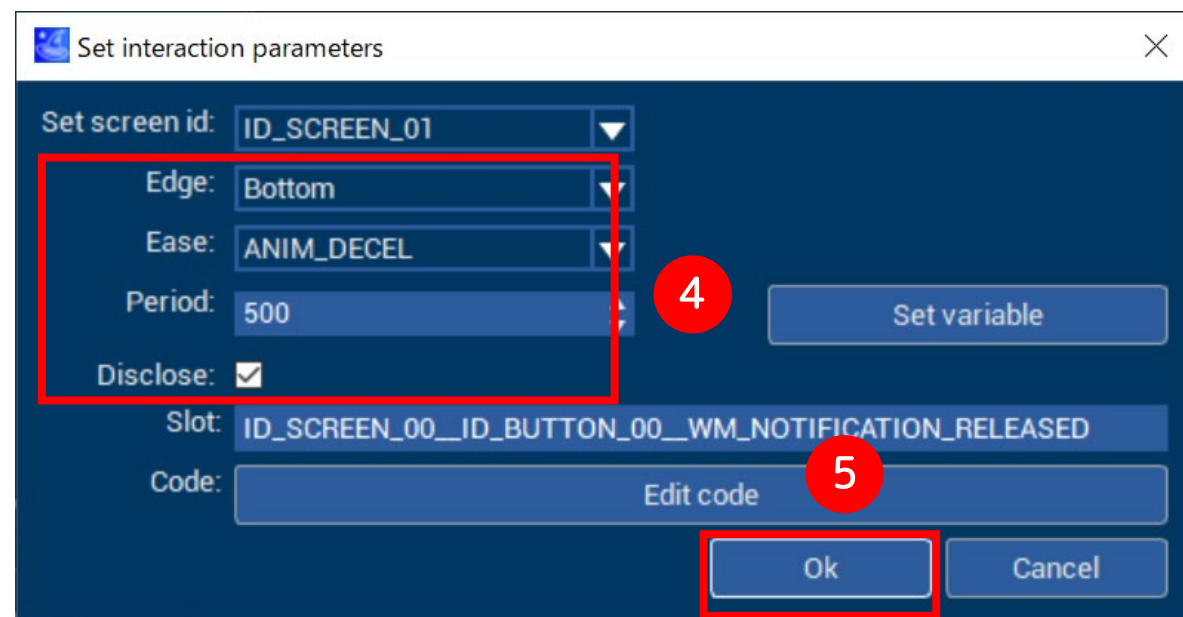
② 「Interactions」ウィンドウから「+」アイコンボタンをクリックして新しいジョブイベントを追加します。

③ ④ 画面の通りにSHIFTSCREENインタラクションコマンドを設定します。

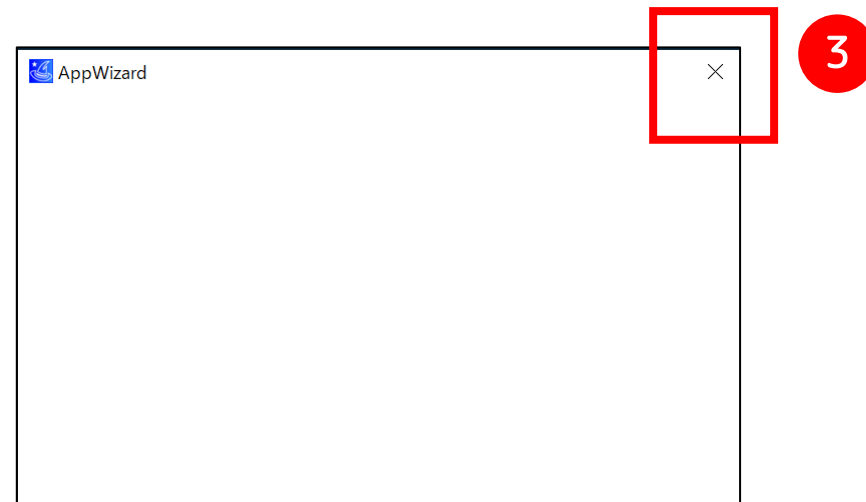
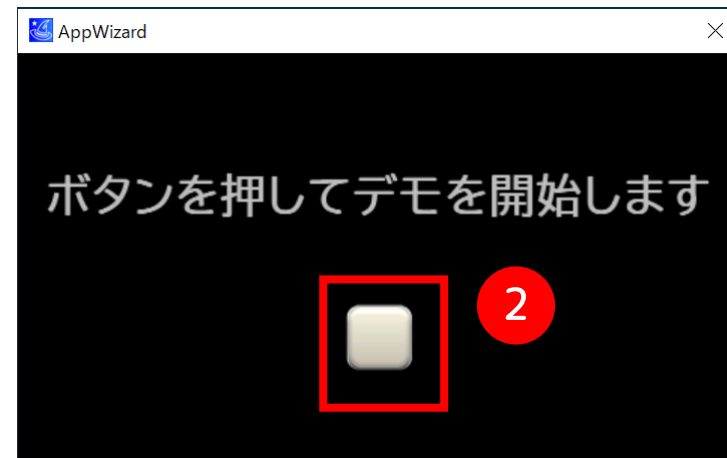
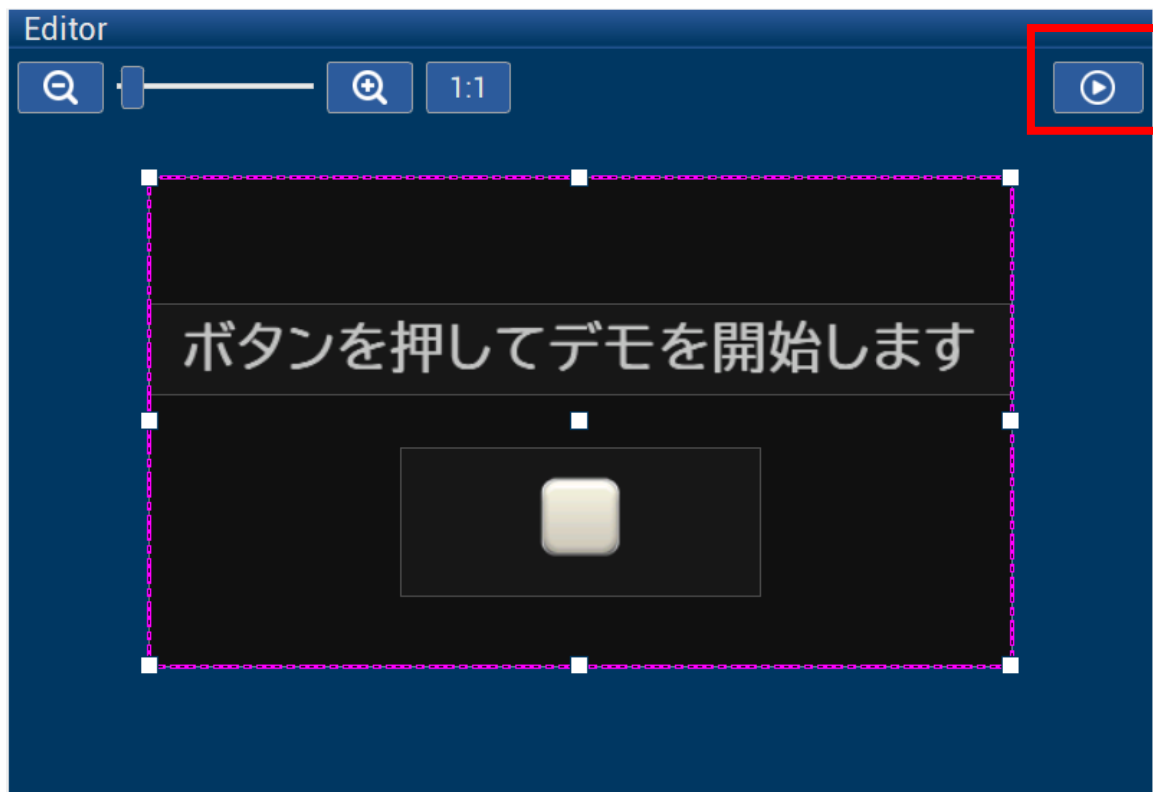
Edge=Bottom, Ease=ANIM\_DECEL, Period=500

Disclose: 有効 (ON)

⑤ 「OK」ボタンを押して設定を保存



## 画面作成：スクリーン1への画面移動ジョブの設定（動作確認）



- ① AppWizardのPlayモードシミュレーションでボタン操作での画面移動を確認します。
- ②③ 確認後にPlayモードを終了します。



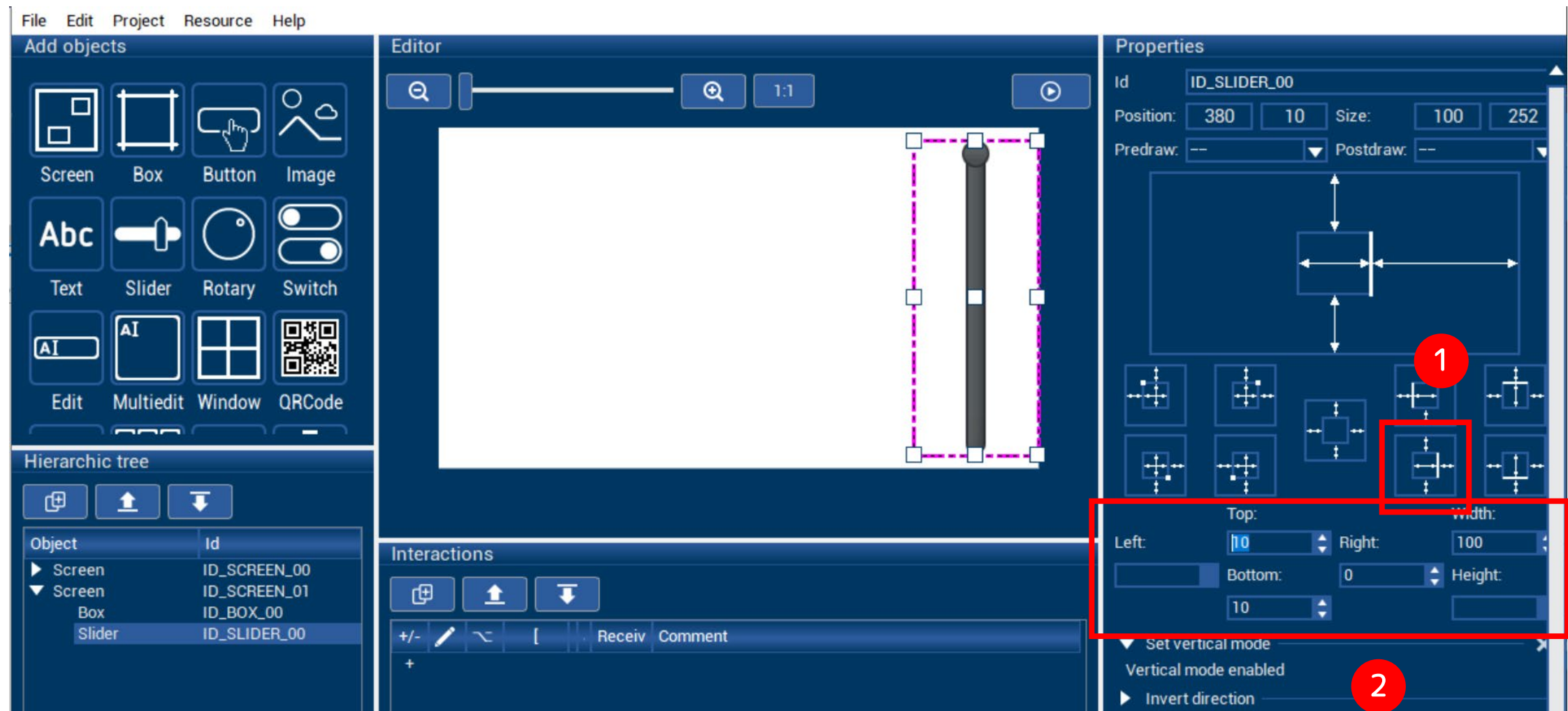
## 画面作成：Sliderオブジェクトの追加

The screenshot displays the emWin GUI development environment with three main panels: 'Add objects', 'Editor', and 'Properties'.

- Add objects panel:** Contains icons for various GUI elements. The 'Slider' icon is highlighted with a red box and a red circle with the number '2'. A red arrow points from this icon to the 'Editor' panel.
- Editor panel:** Shows a central workspace where a vertical slider has been added to the left edge of the screen. A red circle with the number '1' is placed near the 'Hierarchic tree' panel.
- Hierarchic tree panel:** A tree view showing the GUI hierarchy. The 'Slider' object under 'Screen' is highlighted with a red box.
- Properties panel:** Displays settings for the selected 'ID\_SLIDER\_00' object. The 'Set vertical mode' option is checked and highlighted with a red box and a red circle with the number '3'.

**① オブジェクトウィンドウから「ID\_SCREEN\_01」を選択します。**  
**② Sliderを追加します。**  
**③ Verticalモードに設定します。**

# 画面作成：Sliderオブジェクトの設定



- ① Sliderオブジェクトの配置を（画面右側に）設定します。
- ② Width = 100, Top = 10, Bottom = 10, Right = 0

## 画面作成：Gaugeオブジェクトの追加

The screenshot displays the AppWizard GUI development environment with four main panels: Add objects, Editor, Properties, and Hierarchic tree. The 'Add objects' panel on the left contains various UI components, with the 'Gauge' icon highlighted by a red box and a red arrow pointing to the 'Editor' panel. The 'Editor' panel shows a canvas with a semi-circular gauge and a vertical slider. The 'Properties' panel on the right shows the configuration for the selected 'ID\_GAUGE\_00' object, with its position and size settings visible. The 'Hierarchic tree' panel at the bottom left shows a list of objects, with 'ID\_SCREEN\_01' selected and highlighted by a red box. The 'Interactions' panel at the bottom is partially visible.


① オブジェクトウィンドウから「ID\_SCREEN\_01」を選択します。

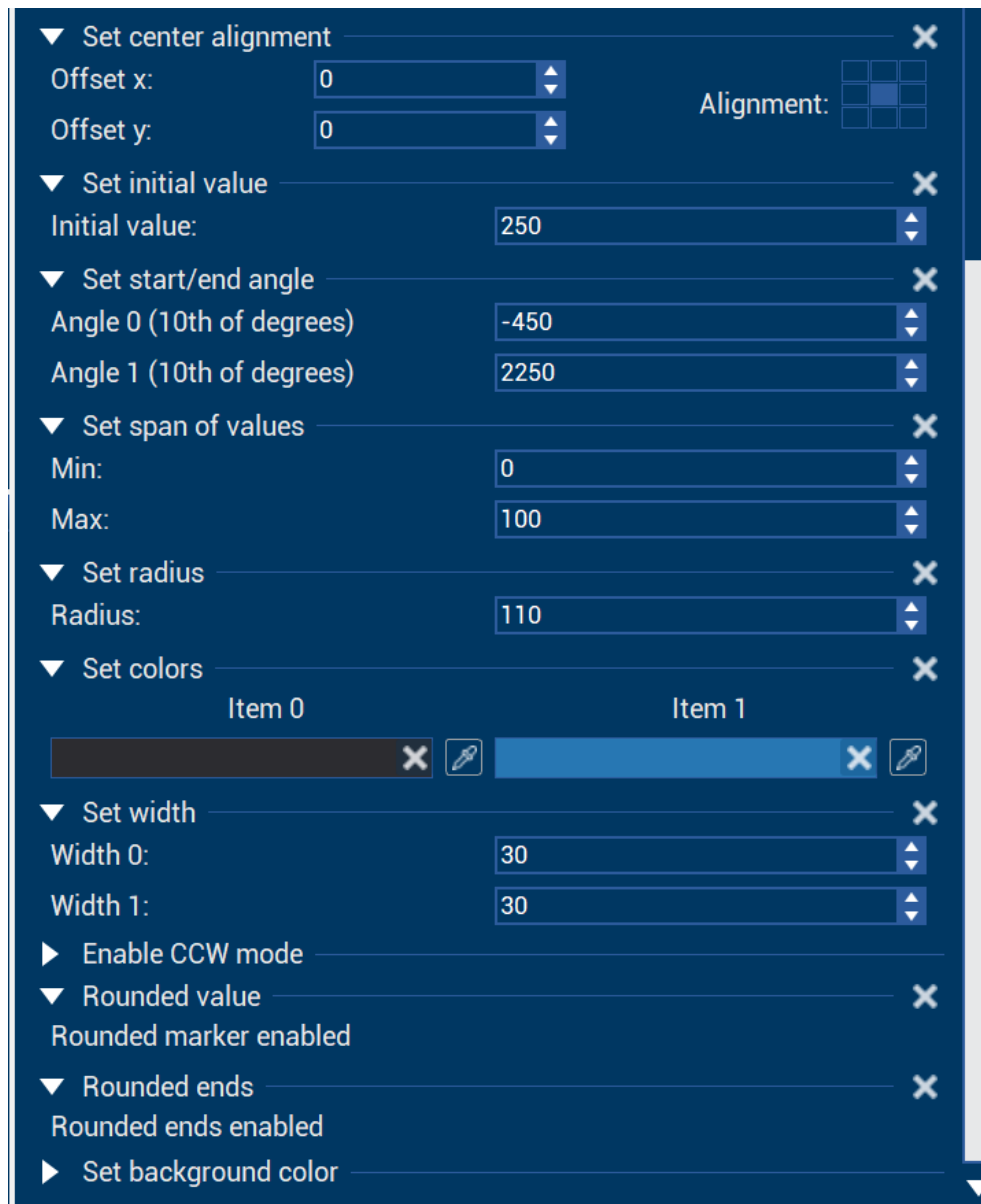
② Gaugeを追加します。

③ Gaugeオブジェクトの配置を（画面左側に）設定します。

④ Right = 100, Top = 0, Bottom = 0, Left = 0

以下の通りにGaugeオブジェクトのPropertyを設定します。


Offset x	0
Offset y	0
Alignment	Center 
Angle0 (Start Angle)	-450
Angle1 (End Angle)	2250
Min value	0
Max value	100
Radius	110
Width0 (Background)	30
Width1 (Foreground)	30
Rounded marker	enabled
Rounded ends	enabled



▼ Set center alignment

Offset x: 0

Offset y: 0

Alignment: 

▼ Set initial value

Initial value: 250

▼ Set start/end angle

Angle 0 (10th of degrees) -450

Angle 1 (10th of degrees) 2250

▼ Set span of values

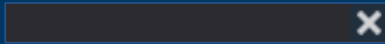

Min: 0



Max: 100

▼ Set radius

Radius: 110

▼ Set colors

Item 0  

Item 1  

▼ Set width

Width 0: 30

Width 1: 30

▶ Enable CCW mode

▼ Rounded value

Rounded marker enabled

▼ Rounded ends

Rounded ends enabled

▶ Set background color



## 画面作成：カウンタ用テキストオブジェクトの追加

The screenshot displays the emWin GUI development environment. On the left, the 'Hierarchic tree' shows a project structure with 'ID\_SCREEN\_01' selected. The 'Add objects' panel on the top left contains various UI components, with the 'Text' object highlighted. The central 'Editor' window shows a gauge with a text object being placed in its center. The 'Properties' panel on the right shows the configuration for the selected text object, with the 'Set text color' field highlighted.

① オブジェクトウィンドウから「ID\_SCREEN\_01」を選択します。

② Textオブジェクトを追加します。

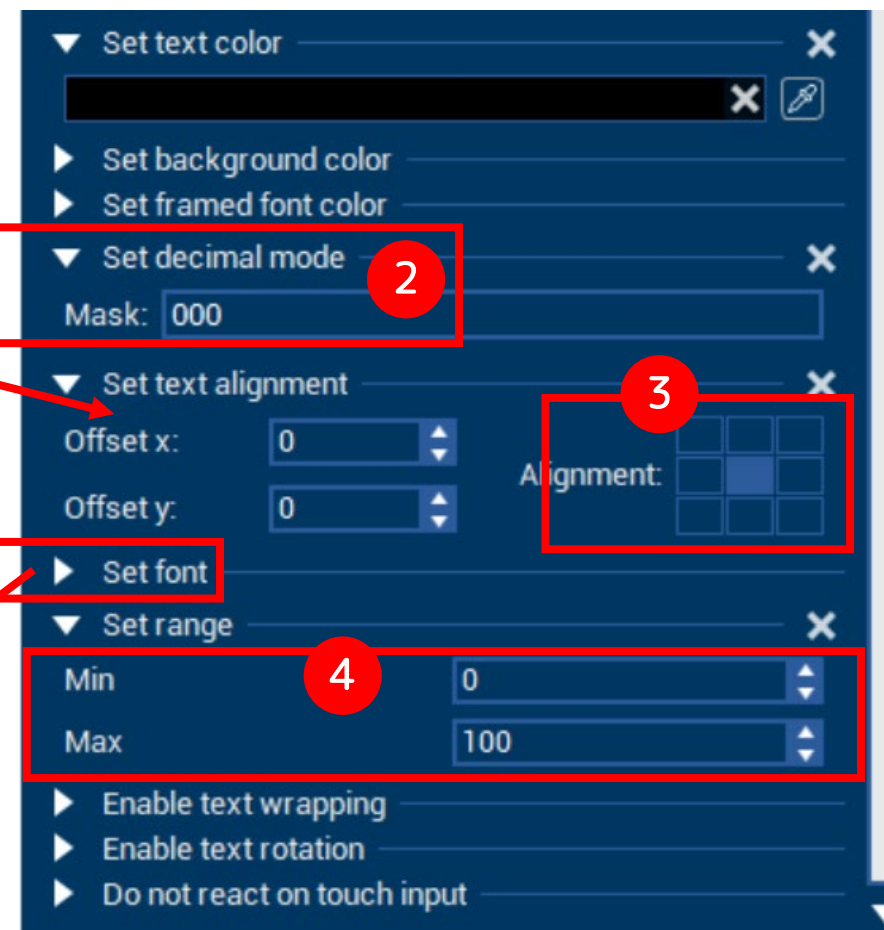
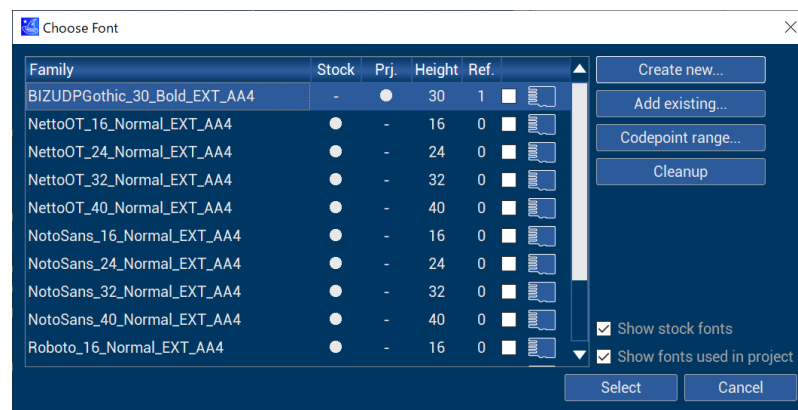
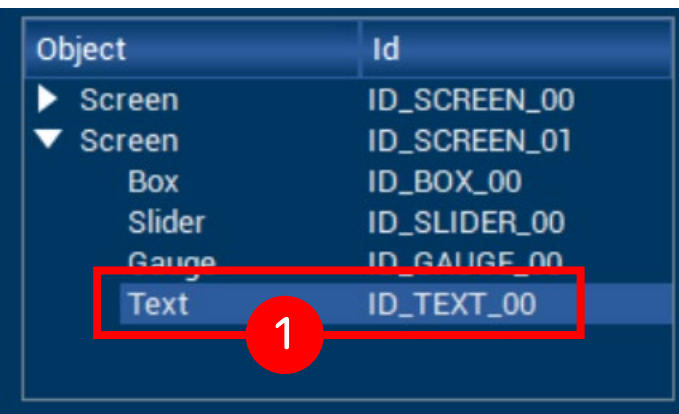
③ Textオブジェクトの配置を（Gauge画面の真ん中に）設定します。

Right = 220, Top = 90, Bottom = 90, Left = 120

④ テキスト色はドフォルト設定値の黒色に設定します。

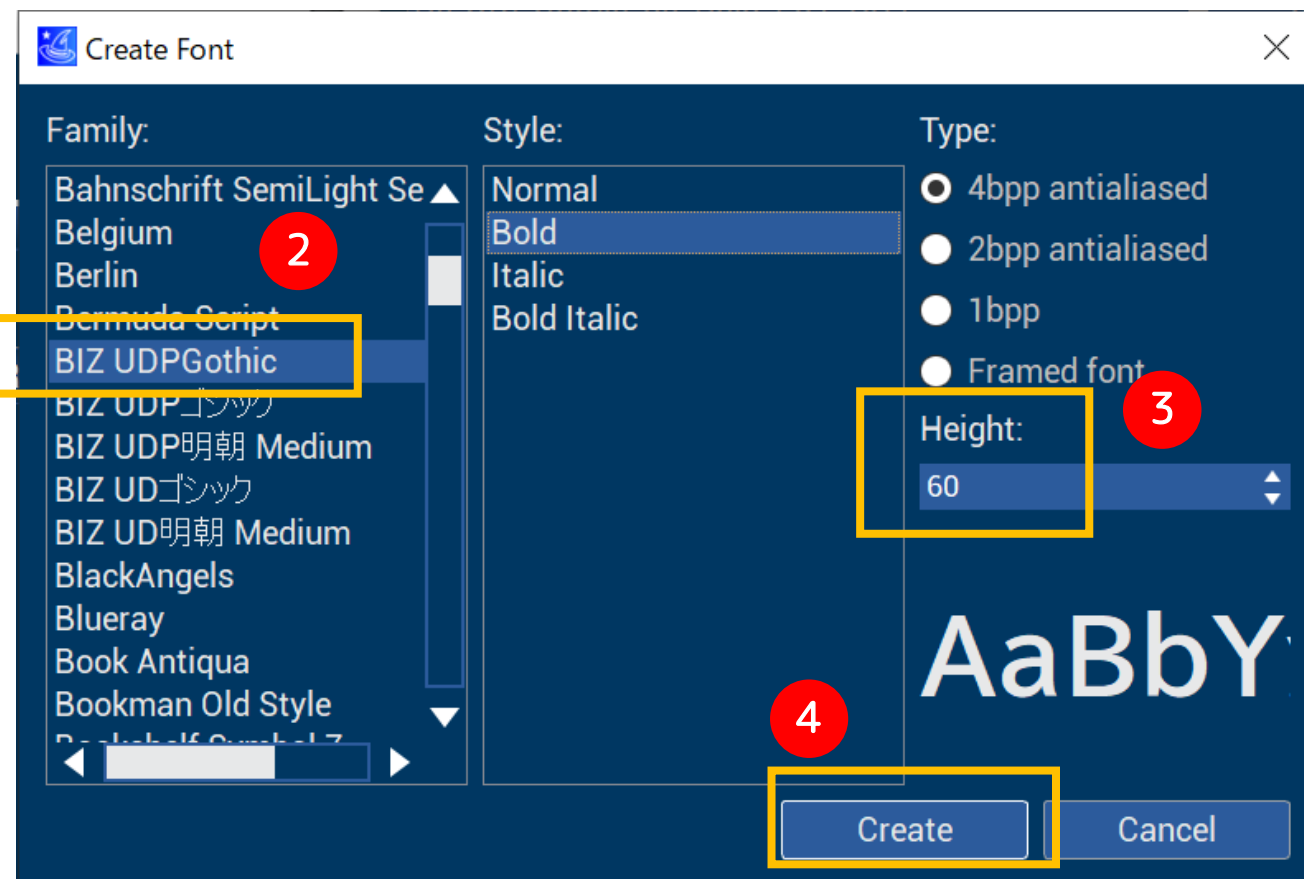
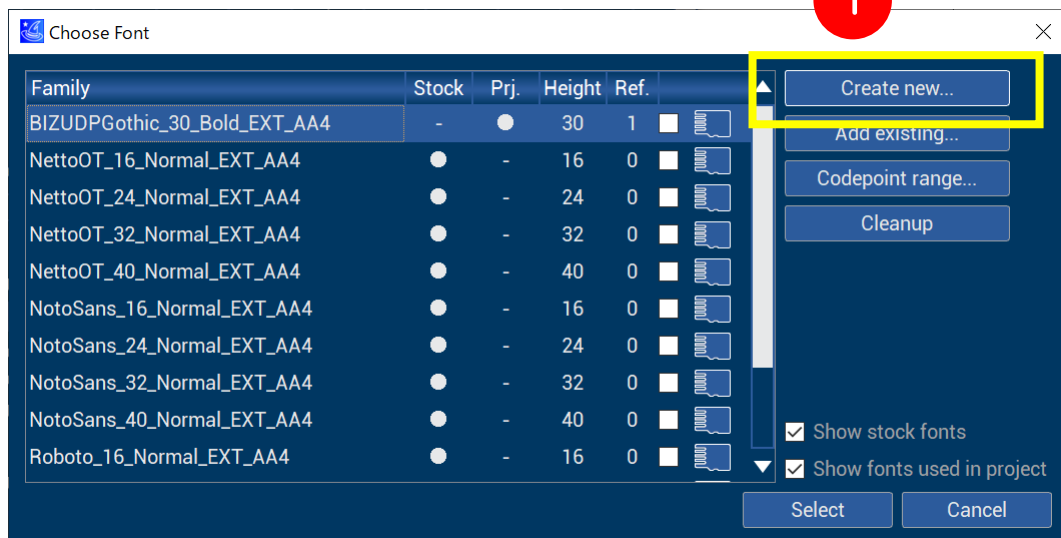
## 画面作成：カウンタ用テキストオブジェクトの設定

- ① [ID\_SCREEN\_01]ウィンドウのオブジェクトウィンドウから「ID\_TEXT\_00」を選択します。  
TextオブジェクトのProperty設定を変更します。
- ② Decimalモード（数字だけ表示）を設定します。マスク設定値：000
- ③ Textオブジェクトの配置を（Gauge画面の真ん中に）設定します。
- ④ 「Set range」のアイコンをクリックして設定値の範囲（0～100）を設定します。Min = 0, Max = 100
- ⑤ 「Set Font」のアイコンをクリックしてフォント設定ダイアログを開きます。



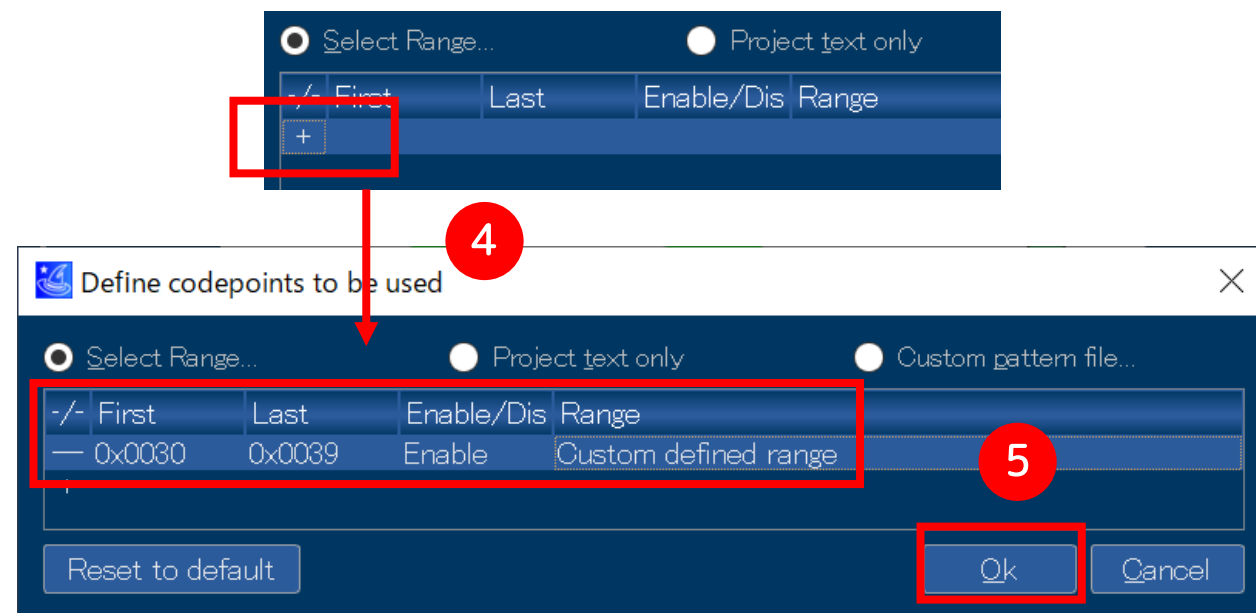
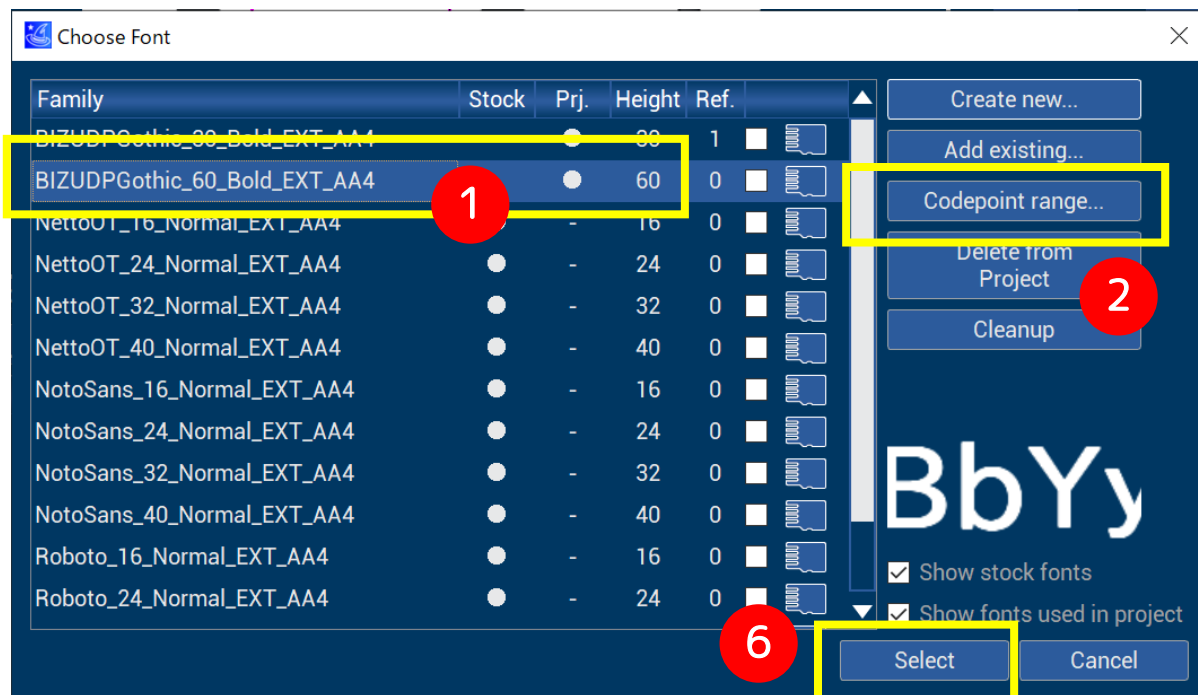
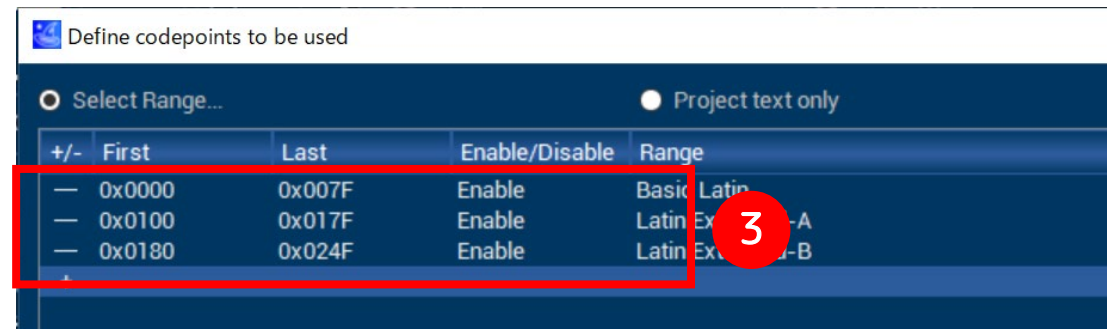
## 画面作成：カウンタ用テキストオブジェクトのフォント設定

- ① フォント設定ダイアログ「Create New」ボタンをクリックしてフォント作成ダイアログを開きます。
- ② 「BIZ UDPGothic / Bold」フォントを選択
- ③ フォントHeight=80 を設定します。
- ④ 「Create」ボタンを押してダイアログを閉じます。



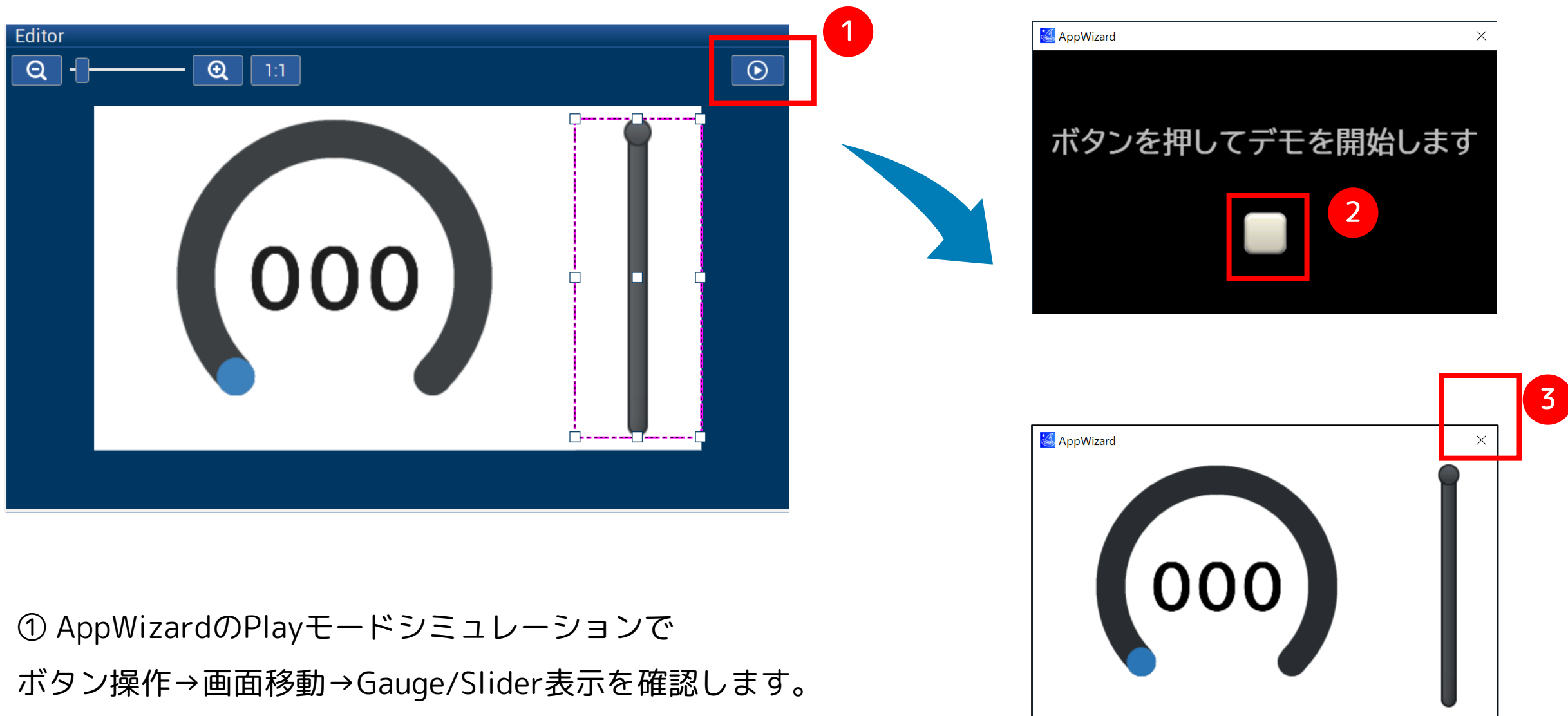
## 画面作成：カウンタ用テキストオブジェクトのフォント設定

- ① 「BIZUDPGothic\_60\_Bold…」フォントを選択します。
- ② 「Codepoint range」ボタンを押して、
- ③ 「Select Range」オプションを選択して左側の「-」アイコンをクリックして現在のすべてのエントリを削除します。
- ④ 「+」アイコンをクリックし、「custom range」を選択して以下の設定で（0～9）文字のASCIIエンコードのみ有効に設定します。  
First=30, Last=39, Enable/Disable=Enable
- ⑤ 「OK」ボタンを押して設定を保存
- ⑥ 「Select」ボタンを押してダイアログを閉じます。





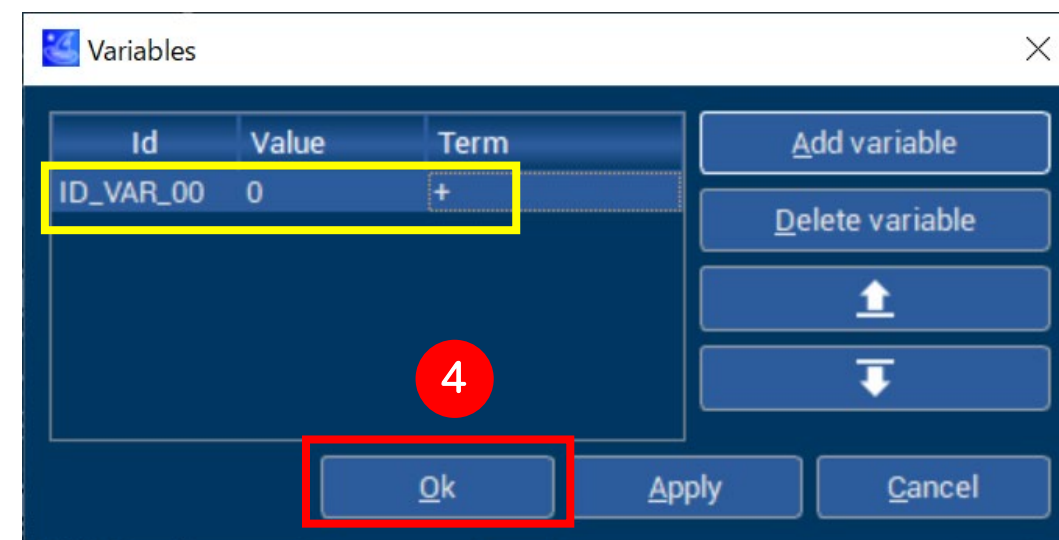
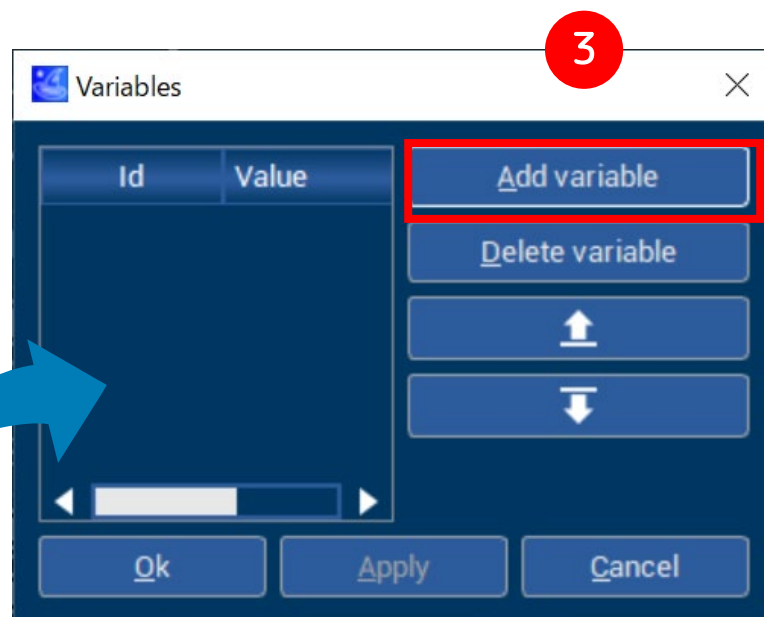
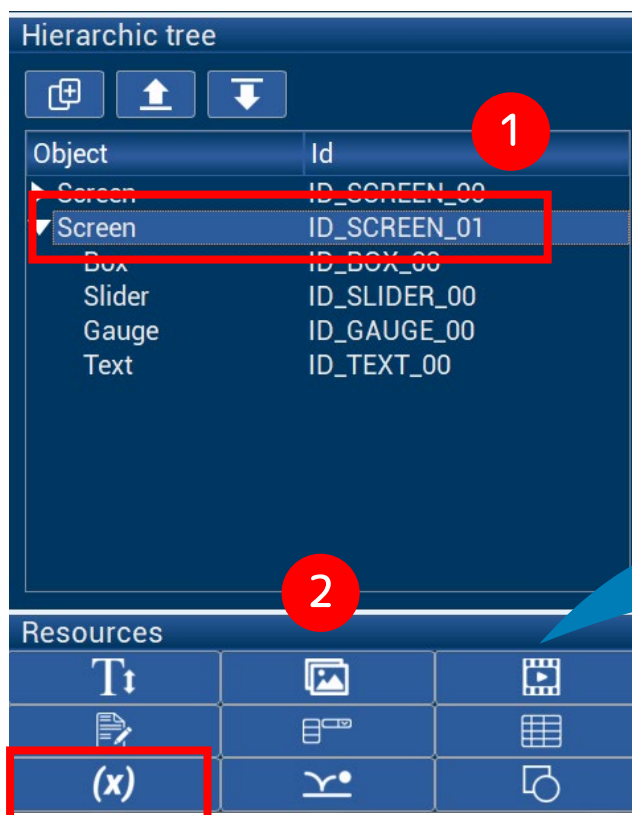
## 画面作成：Gauge、Sliderオブジェクトの動作確認



## 画面作成：データ共有のためVariableを作成

GaugeとSliderオブジェクトの設定値を共有するためにAppWizardの変数を作成します。

- ① オブジェクトウィンドウから「ID\_SCREEN\_01」を選択します。
- ② Variables Resource 設定の「(x)」ボタンをクリックします。
- ③ Variablesダイアログ画面から「Add variables」ボタンをクリックして、
- ④ Id項目に「ID\_VAR\_00」変数の表示を確認して「Ok」ボタンをクリックします。

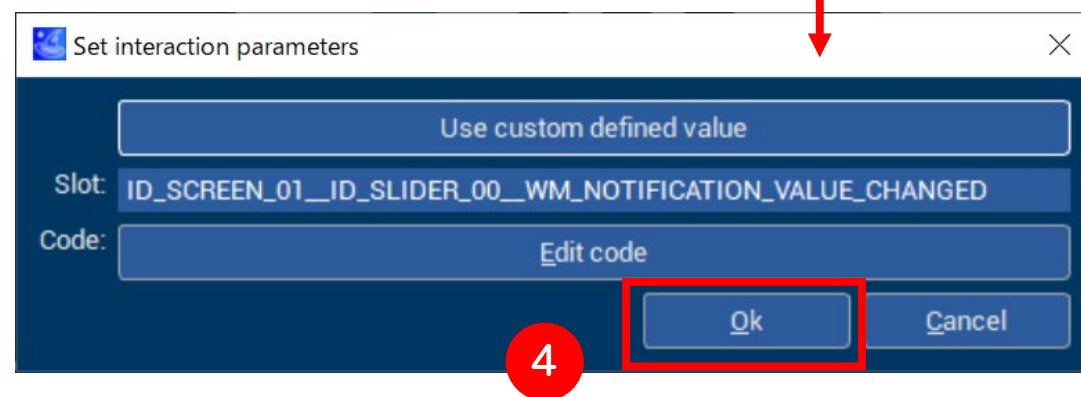


## 画面作成：「ID\_VAR\_00」変数の設定

Sliderオブジェクトの設定値が変わったら「ID\_VAR\_00」変数も更新するようにインタラクションジョブを設定します。

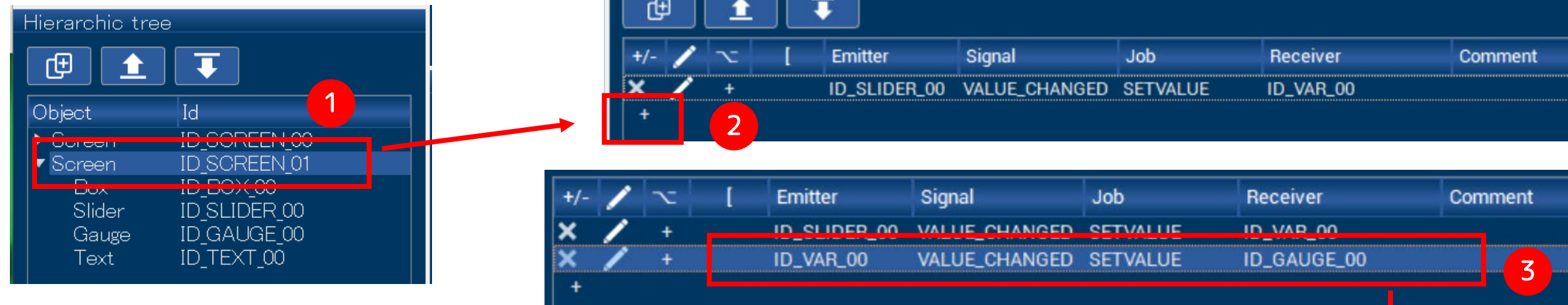


- ① オブジェクトウィンドウから「ID\_SCREEN\_01」を選択します。
- ② 「Interactions」ウィンドウから「+」アイコンボタンをクリックして新しいジョブイベントを追加します。
- ③ 画面の通りにSETVALUEインタラクションコマンドを設定します。  
Emitter=ID\_SLIDER\_00, Signal=VALUE\_CHANGED,  
Job=SETVALUE, Receiver = ID\_VAR\_00
- ④ 「OK」ボタンを押して設定を保存

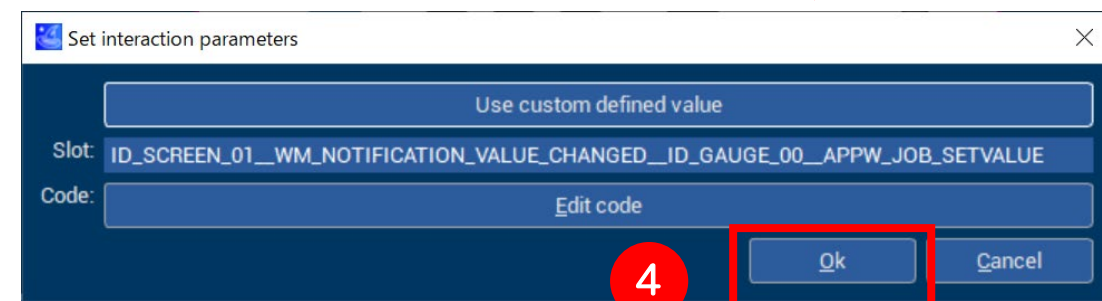


## 画面作成：「ID\_VAR\_00」変数の設定

「ID\_VAR\_00」変数の値が変わりましたらGaugeオブジェクトのステータス表示も更新するようにインタラクションジョブを設定します。



- ① オブジェクトウィンドウから「ID\_SCREEN\_01」を選択します。
- ② 「Interactions」ウィンドウから「+」アイコンボタンをクリックして新しいジョブイベントを追加します。
- ③ 画面の通りにSETVALUEインタラクションコマンドを設定します。  
Emitter= ID\_VAR\_00, Signal=VALUE\_CHANGED,  
Job=SETVALUE, Receiver = ID\_GAUGE\_00
- ④ 「OK」ボタンを押して設定を保存



## 画面作成：「ID\_VAR\_00」変数の設定

「ID\_VAR\_00」変数の値が変わりましたらカウンタテキストオブジェクトのステータス表示も更新するようにインタラクションジョブを設定します。

The screenshot shows the GUI development environment. On the left, the 'Hierarchic tree' panel displays a list of objects: Screen (ID\_SCREEN\_00), Screen (ID\_SCREEN\_01), Box (ID\_BOX\_00), Slider (ID\_SLIDER\_00), Gauge (ID\_GAUGE\_00), and Text (ID\_TEXT\_00). A red box highlights the 'Screen (ID\_SCREEN\_01)' entry, with a red circle '1' next to it. A red arrow points from this box to the 'Interactions' table. The 'Interactions' table has columns: Emitter, Signal, Job, Receiver, and Comment. A red box highlights the '+' icon in the table header, with a red circle '2' next to it. Below the table, a red box highlights the row where Emitter is 'ID\_VAR\_00', Signal is 'VALUE\_CHANGED', Job is 'SETVALUE', and Receiver is 'ID\_TEXT\_00', with a red circle '3' next to it. A red arrow points from this row to the 'Set interaction parameters' dialog box.

Emitter	Signal	Job	Receiver	Comment
ID_SLIDER_00	VALUE_CHA...	SETVALUE	ID_VAR_00	
ID_VAR_00	VALUE_CHA...	SETVALUE	ID_GAUGE_00	
ID_SLIDER_00	VALUE_CHANGED	SETVALUE	ID_VAR_00	
ID_VAR_00	VALUE_CHANGED	SETVALUE	ID_GAUGE_00	
ID_VAR_00	VALUE_CHANGED	SETVALUE	ID_TEXT_00	

- ① オブジェクトウィンドウから「ID\_SCREEN\_01」を選択します。
- ② 「Interactions」ウィンドウから「+」アイコンボタンをクリックして新しいジョブイベントを追加します。
- ③ 画面の通りにSETVALUEインタラクションコマンドを設定します。  
Emitter= ID\_VAR\_00, Signal=VALUE\_CHANGED,  
Job=SETVALUE, Receiver = ID\_TEXT\_00

- ④ 「OK」ボタンを押して設定を保存

The 'Set interaction parameters' dialog box is shown. It has a title bar with a close button. Inside, there is a section 'Use custom defined value' with a slot field containing 'ID\_SCREEN\_01\_\_WM\_NOTIFICATION\_VALUE\_CHANGED\_ID\_TEXT\_00\_\_APPW\_JOB\_SETVALUE' and a code field with 'Edit code'. At the bottom, there are 'Ok' and 'Cancel' buttons. A red box highlights the 'Ok' button, with a red circle '4' next to it.



## 画面作成：「ID\_VAR\_00」変数の初期化設定

スクリーン作成の時に「ID\_VAR\_00」変数の値を初期化するようにインタラクションジョブを設定します。

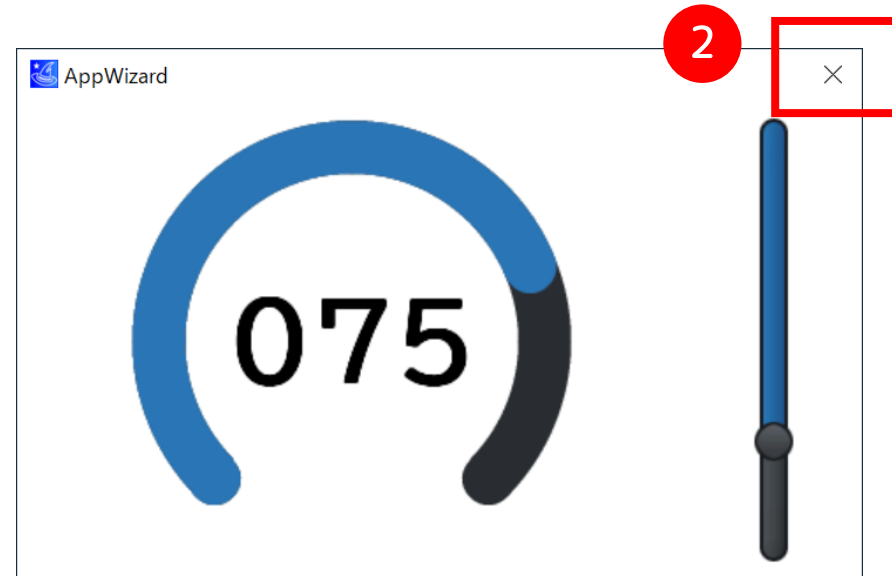
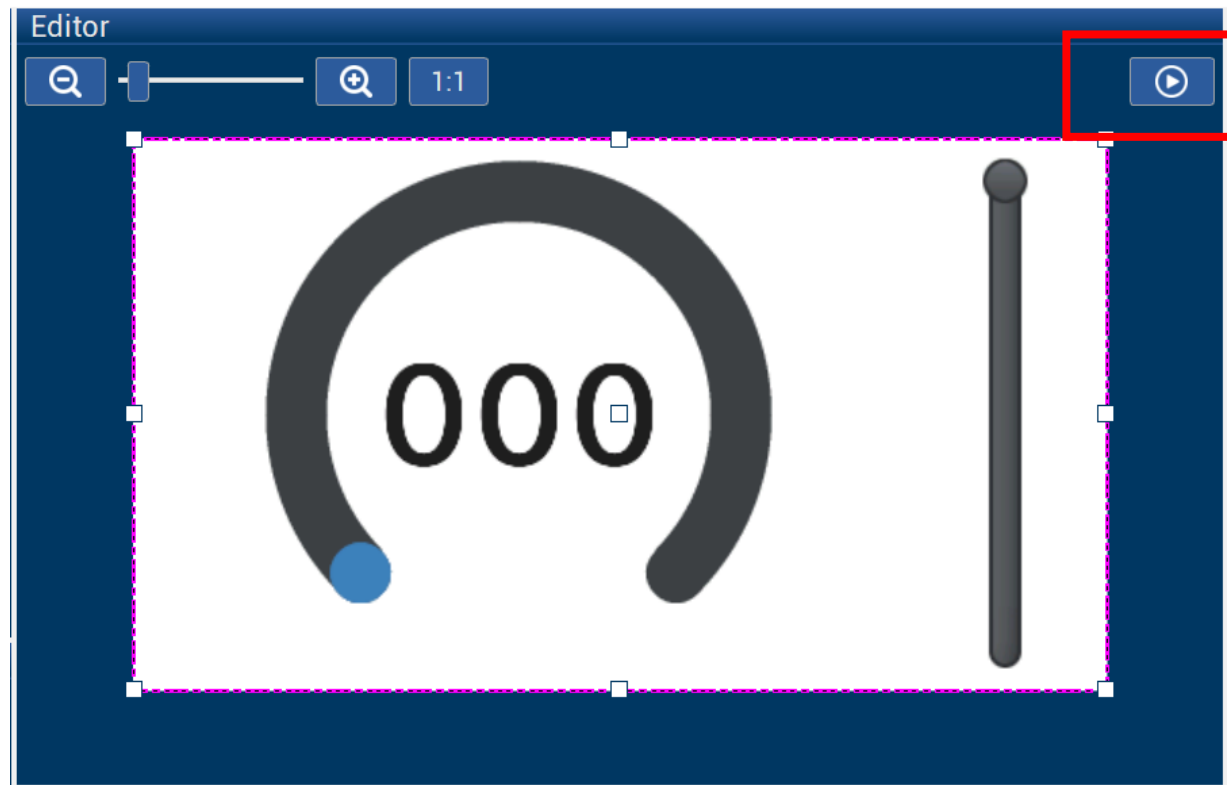
The screenshot shows the AppWizard interface. On the left, the 'Hierarchic tree' displays a list of objects: Screen (ID\_SCREEN\_00), Screen (ID\_SCREEN\_01), Box (ID\_BOX\_00), Slider (ID\_SLIDER\_00), Gauge (ID\_GAUGE\_00), and Text (ID\_TEXT\_00). A red box labeled '1' highlights 'ID\_SCREEN\_01'. On the right, the 'Interactions' window shows a table of interactions. A red box labeled '2' highlights the '+' icon to add a new interaction. A red box labeled '3' highlights the new interaction row being added with Emitter 'ID\_SCREEN\_01', Signal 'INITDIALOG', Job 'SETVALUE', and Receiver 'ID\_VAR\_00'.

-/-	Ed	Emitter	Signal	Job	Receiver	Comment
x		ID_SLIDER_00	VALUE_CHANGED	SETVALUE	ID_VAR_00	
x		ID_VAR_00	VALUE_CHANGED	SETVALUE	ID_GAUGE_00	
x		ID_VAR_00	VALUE_CHANGED	SETVALUE	ID_TEXT_00	
x		ID_SCREEN_01	INITDIALOG	SETVALUE	ID_VAR_00	

- ① オブジェクトウィンドウから「ID\_SCREEN\_01」を選択します。
- ② 「Interactions」ウィンドウから「+」アイコンボタンをクリックして新しいジョブイベントを追加します。
- ③ 画面の通りにSETVALUEインタラクションコマンドを設定します。  
Emitter= ID\_SCREEN\_01, Signal=INITDIALOG,  
Job=SETVALUE, Receiver = ID\_VAR\_00, Set value = 0
- ④ 「OK」ボタンを押して設定を保存

The screenshot shows the 'Set interaction parameters' dialog box. It has fields for 'Set value' (p), 'Slot' (ID\_SCREEN\_01\_APPW\_NOTIFICATION\_INITDIALOG), and 'Code' (Edit code). There are buttons for 'Set variable', 'Ok', and 'Cancel'. A red box labeled '4' highlights the 'Ok' button.

## 画面作成：Gauge、Sliderオブジェクトのインタラクション設定の動作確認

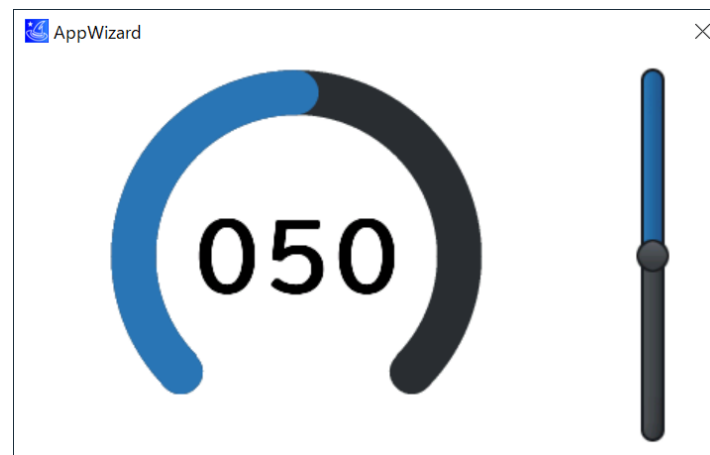
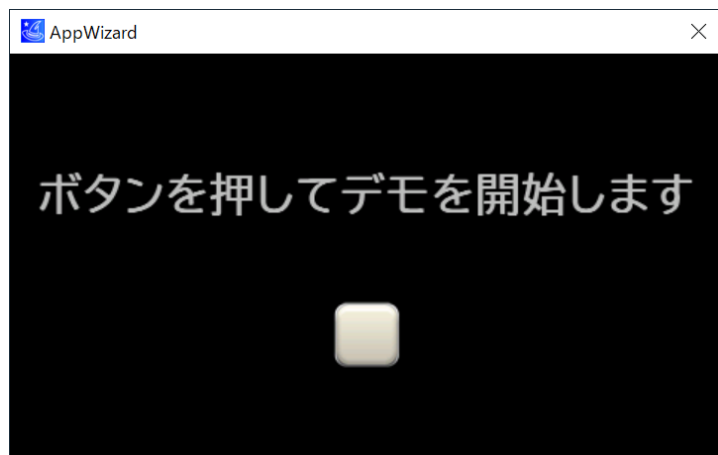


- ① AppWizardのPlayモードシミュレーションで  
ボタン操作→画面移動→Gauge/Slider/カウンタ動作を確認します。
- ② 確認後にPlayモードを終了します。

## TASK-3

# ビルド & 動作確認

e2Studioプロジェクト環境でBSP+UIデザインアプリケーションコードをビルドして、RX72N Envisionボード環境で動作を確認します。





# ビルド & 動作確認：画面デザインのソースコードを作成

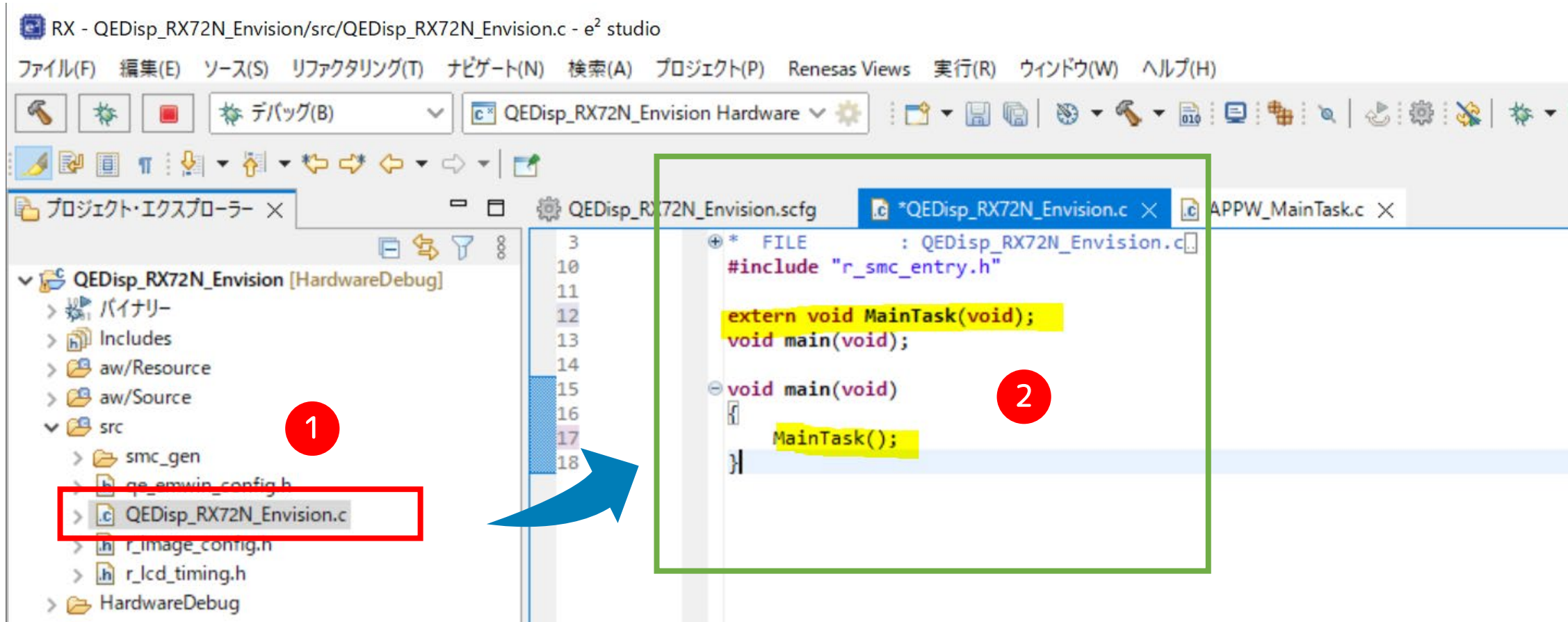
The screenshot shows the e2Studio IDE interface. On the left, the 'File' menu is open, and the 'Export & Save' option is highlighted with a red box. A red circle with the number '1' is next to it. A blue arrow points from this menu item to the 'aw' folder in the project explorer on the right. The project explorer shows the 'aw' folder expanded, revealing subfolders like 'Resource', 'Simulation', 'Source', and 'Generated'. The 'Source' folder is further expanded, showing files like 'APPWConf.c', 'ID\_SCREEN\_00.c', 'ID\_SCREEN\_00.h', 'ID\_SCREEN\_01.c', 'ID\_SCREEN\_01.h', 'Resource.c', 'Resource.h', and 'APPW\_MainTask.c'. The 'APPW\_MainTask.c' file is selected. The main editor window displays the source code of 'APPW\_MainTask.c'. A green box highlights the code for the 'MainTask' function, which is annotated with 'emWinスタートアップ 起動処理'.

```
#include "Generated/Resource.h"

/* MainTask */
void MainTask(void) {
    //
    // Setup configuration dependent pointers
    APPW_X_Setup();
    //
    // Initialize AppWizard
    APPW_Init(APPW_PROJECT_PATH);
    //
    // Create all persistent screens except initial screen
    APPW_CreatePersistentScreens();
    //
    // Create initial screen...
    APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN);
    //
    // ...and keep it alive
    while (1) {
        while (GUI_Exec1()) {
            APPW_Exec();
        }
        APPW_Exec();
        GUI_X_Delay(5);
    }
}
```

① AppWizardの「File → Export & Save」メニューコマンドでGUIコード（リソース、アプリケーション）を作成します。e2Studio側に「aw」フォルダ下にコードが表示されます。

# ビルド & 動作確認：スタートアップコードを作成



- ① AppWizardのプロジェクトエクスプローラーウィンドウから「QEDisp\_RX72N\_Envision.c」スタートアップエントリーファイルを開きます。
- ② 「QEDisp\_RX72N\_Envision.c」ファイルにemWinスタートアップエントリー関数 MainTask() を追加します。

# ビルド & 動作確認：プロジェクトのビルド

①プロジェクトを選択して、  
②ビルドコマンドを実行します。

```
CDT ビルド・コンソール [QEDisp_RX72N_Envision]
Scanning and building file: ../src/smc_gen/r_fw20_rx/src/rx_dave_memory_rx.c
Invoking: Library Generator
Library Generator Completed
Finished building:

Invoking: Linker
Building target:

Renesas Optimizing Linker Completed
Finished building target:

Loading input file QEDisp_RX72N_Envision.abs
Parsing the ELF input file....
49 segments required LMA fixes
Converting the DWARF information....
Constructing the output ELF image....
Saving the ELF output file QEDisp_RX72N_Envision.elf
Build complete.

18:36:56 Build Finished. 0 errors, 0 warnings. (took 9s.323ms)
```

# ビルド & 動作確認：評価ボード環境でデバッグ及び動作確認

e2Studio-v2310 - QEDisp\_RX72N\_Envision/src/smc\_gen/r\_bsp/mcu/all/resetprg.c - e2 studio

ファイル(F) 編集(E) ソース(S) リファクタリング(T) ナビゲート(N) 検索(A) プロジェクト(P) Renesas Views 実行(R) Renesas AI ウィンドウ(W) ヘルプ(H)

デバッグ ×

1 QEDisp\_RX72N\_Envision HardwareDebug

APPW\_MainTask.c QEDisp\_RX72N\_Envision.c

QEDisp\_RX72N\_Envision HardwareDebug [Renesas GDB Hardware Debugging] [pid: 6]

QEDisp\_RX72N\_Envision.x [1] [cores: 0]

Thread #1 1 (single core) [core: 0] (Suspended : シグナル : SIGILL)

PowerON\_Reset\_PC() at resetprg.c:204 0xffc00000

rx-elf-gdb -rx-force-isa=v3 -rx-force-double-fpu (7.8.2)

Renesas GDB server (Host)

デバッグ(D)

デバッグの構成(B)...

お気に入りの編成(V)...

174

175

176

177

178

204 ffc00000

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223 ffc0000e

224

225

226

227 ffc00017

228

229

230

extern void R\_BSP\_MAIN\_FUNCTION(void);

#endif

/\* Function name: PowerON\_Reset\_PC \*/

R\_BSP\_POR\_FUNCTION(R\_BSP\_STARTUP\_FUNCTION)

{

/\* Stack pointers are setup prior to calling this function - see comments ab

/\* You can use auto variables in this function but such variables other than

/\* will be unavailable after you change the stack from the I stack to the U

/\* The bss sections have not been cleared and the data sections have not bee

/\* and constructors of C++ objects have not been executed until the \_INITSC

#if defined(\_\_GNUC\_\_)

#if BSP\_CFG\_USER\_STACK\_ENABLE == 1

INTERNAL\_NOT\_USED(ustack\_area);

#endif

INTERNAL\_NOT\_USED(istack\_area);

#endif

#if defined(\_\_CCRX\_\_) || defined(\_\_GNUC\_\_)

/\* Initialize the Interrupt Table Register \*/

R\_BSP\_SET\_INTB(R\_BSP\_SECTOP\_INTVECTTBL);

#ifdef BSP\_MCU\_EXCEPTION\_TABLE

/\* Initialize the Exception Table Register \*/

R\_BSP\_SET\_EXTB(R\_BSP\_SECTOP\_EXCEPTVECTTBL);

#endif

#ifdef BSP\_MCU\_FLOATING\_POINT

変数 ×

名前

型

値

コンソール ×

レジスター

問題

スマート・ブラウザー

Debugger Console

メモリ

QEDisp\_RX72N\_Envision HardwareDebug [Renesas GDB Hardware Debugging] [pid: 6]

ターゲット接続状況 - OK

ターゲット接続状況 - OK

ダウンロード開始

ダウンロード終了

ハードウェア・ブレークポイントをアドレス0xffc49525に設定します。

中断中

書き込み可能

スマート挿入

setAppWizard

- RX72N Envision Kitの詳細

<https://www.renesas.com/jp/ja/products/software-tools/boards-and-kits/eval-kits/rx72n-envision-kit.html>

- RX65N / RX72N Envision Kit用emWinスタートアップガイド (PDF)

[https://www.embitek.co.jp/technote/emwin/StartupGuide\\_emWin\\_EnvisionKit.pdf](https://www.embitek.co.jp/technote/emwin/StartupGuide_emWin_EnvisionKit.pdf)

- RXファミリ QE for Display [RX] アプリケーションノート

<https://www.renesas.com/jp/ja/document/apn/rx-family-qe-display-rx-application-note>

- AppWizardユーザマニュアル

[https://www.segger.com/doc/UM03003\\_AppWizard.html](https://www.segger.com/doc/UM03003_AppWizard.html)

- emWinユーザマニュアル

[https://www.segger.com/doc/UM03001\\_emWin.html](https://www.segger.com/doc/UM03001_emWin.html)



## ソースパッケージアップグレードのご案内

OEM版emWinライブラリパッケージをご利用のユーザー様がソース付きのemWin PROパッケージへ通常価格の **割引** 価格でアップグレード可能です。

※ ライセンス製品価格・割引率は、サポート窓口対応や諸条件、為替相場により提示価格が変動する場合があります。

ソースアップグレードのメリット：

- ソース付きプロジェクトでGUIアプリケーションのデバッグ操作が簡単
- ソースを参考に新しいウィジェット作成可能
- emWinライブラリの最適化はさらにチューニング可能
- お困りの際に技術サポートへの問い合わせ可能

お気軽に以下窓口へお問い合わせください。

# 株式会社エンビテック

代理店販売（デバッグ、RTOS、GUI、ミドルウェア）

TEL: 03-6240-2655 / FAX: 03-6240-2656

E-mail : [sales@embitek.co.jp](mailto:sales@embitek.co.jp)

<https://www.embitek.co.jp>

