



Elektrobit

# EB GUIDE Studio

ユーザーガイド

バージョン6.8.0.190618155600



Elektrobit Automotive GmbH  
Am Wolfsmantel 46  
D-91058 Erlangen  
GERMANY

Phone: +49 9131 7701-0  
Fax: +49 9131 7701-6333  
<http://www.elektrobit.com>

## Legal notice

Confidential and proprietary information

ALL RIGHTS RESERVED. No part of this publication may be copied in any form, by photocopy, microfilm, retrieval system, or by any other means now known or hereafter invented without the prior written permission of Elektrobit Automotive GmbH.

All brand names, trademarks and registered trademarks are property of their rightful owners and are used only for description.

Copyright 2019, Elektrobit Automotive GmbH.

# 目次

1. 本書について .....	17
1.1. 対象者: モデラー .....	17
1.2. ユーザーマニュアルの構成 .....	17
1.3. 本書で使用する表記スタイル .....	18
1.4. ネーミングルール .....	20
1.5. パスに関する規則 .....	21
2. 安全で正しい使い方 .....	22
2.1. 使用目的 .....	22
2.2. 考えられる誤用 .....	22
3. サポート .....	23
4. EB GUIDEの基本 .....	24
4.1. EB GUIDE product line .....	24
4.2. EB GUIDE Studio .....	24
4.2.1. ヒューマンマシンインターフェースの動作のモデル化 .....	24
4.2.2. ヒューマンマシンインターフェースの外観のモデル化 .....	25
4.2.3. データの処理 .....	25
4.2.4. EB GUIDEモデルのシミュレート .....	25
4.2.5. EB GUIDEモデルのエクスポート .....	26
4.3. EB GUIDE TF .....	26
5. チュートリアル: はじめに .....	28
5.1. EB GUIDEの起動 .....	28
5.2. プロジェクトの作成 .....	29
5.3. ヒューマンマシンインターフェースの動作のモデル化 .....	30
5.4. ヒューマンマシンインターフェースの外観のモデル化 .....	33
5.5. シミュレーションを開始する .....	36
6. バックグラウンド情報 .....	37
6.1. 3Dグラフィック .....	37
6.1.1. サポートされている3Dグラフィック形式 .....	37
6.1.2. 3Dグラフィックファイルの設定 .....	37
6.1.3. 3Dグラフィックファイルのインポート .....	38
6.2. アニメーション .....	40
6.2.1. ウィジェットのアニメーション .....	40
6.2.2. ビュー遷移のアニメーション .....	40
6.2.3. スクリプト曲線 .....	41
6.3. アプリケーションとモデルを結ぶアプリケーションプログラミングインターフェース .....	42
6.4. 通信コンテキスト .....	42
6.5. グラフィカルユーザーインターフェイスのコンポーネント .....	43
6.5.1. プロジェクトセンター .....	43
6.5.1.1. ナビゲーションエリア .....	44

6.5.1.2. コンテンツエリア .....	44
6.5.2. プロジェクトエディター .....	44
6.5.2.1. ナビゲーションコンポーネント .....	45
6.5.2.2. 概要コンポーネント .....	46
6.5.2.3. ツールボックスコンポーネント .....	47
6.5.2.4. プロパティコンポーネント .....	47
6.5.2.5. コンテンツエリア .....	48
6.5.2.6. イベントコンポーネント .....	50
6.5.2.7. データプールコンポーネント .....	50
6.5.2.8. アセットコンポーネント .....	51
6.5.2.9. 名前空間コンポーネント .....	51
6.5.2.10. コマンドエリア .....	51
6.5.2.11. 問題検出コンポーネント .....	52
6.5.2.12. VTAコンポーネント .....	53
6.5.3. ドッキング可能なコンポーネント .....	53
6.5.4. EB GUIDE Monitor .....	54
6.6. データプール .....	57
6.6.1. 概念 .....	57
6.6.2. データプールアイテム .....	57
6.6.3. ウィンドウ表示リスト .....	57
6.7. EB GUIDEモデルとEB GUIDEプロジェクト .....	58
6.7.1. ストレージ形式 .....	59
6.7.2. EB GUIDEプロジェクトの検証基準 .....	60
6.7.2.1. EB GUIDEを開くときの検証 .....	61
6.7.2.2. 問題検出コンポーネントを使用した検証 .....	61
6.8. イベント処理 .....	61
6.8.1. イベントシステム .....	61
6.8.2. イベント .....	62
6.9. 拡張機能 .....	64
6.9.1. EB GUIDE Studio拡張機能 .....	64
6.9.2. EB GUIDE GTF拡張機能 .....	65
6.9.3. EB GUIDE Monitor拡張機能 .....	65
6.10. ガンマ補正レンダリング .....	66
6.10.1. 概念 .....	66
6.10.2. EB GUIDE Studioでのガンマ補正 .....	67
6.11. イメージベースドライティング .....	67
6.11.1. IBLGenerator、ファイル形式とインポート .....	68
6.11.2. OpenGLレンダラーによるIBLの制限 .....	68
6.12. 言語 .....	68
6.12.1. EB GUIDE Studioの表示言語 .....	68
6.12.2. EB GUIDEモデルの言語 .....	69
6.12.3. 言語依存テキストのエクスポートとインポート .....	69

6.13. 名前空間 .....	70
6.14. Photoshopファイル形式のサポート .....	71
6.15. リソース管理 .....	72
6.15.1. フォント .....	72
6.15.1.1. ビットマップフォント .....	73
6.15.1.2. マルチフォントサポート .....	73
6.15.2. 3Dグラフィックのイメージベースドライティング .....	74
6.15.3. イメージ .....	74
6.15.3.1. 9-patchイメージ .....	75
6.15.4. 3Dグラフィック用メッシュ .....	76
6.16. スクリプト言語EB GUIDEスクリプト .....	76
6.16.1. アプリケーションの機能とエリア .....	76
6.16.2. プレフィックスと識別子 .....	76
6.16.3. コメント .....	77
6.16.4. データ型 .....	78
6.16.5. 式 .....	78
6.16.6. 定数と参照 .....	79
6.16.7. 算術式と論理式 .....	80
6.16.8. L値とR値 .....	80
6.16.9. ローカル変数 .....	81
6.16.10. Whileループ .....	82
6.16.11. If-then-else .....	82
6.16.12. 外部関数呼び出し .....	83
6.16.13. データプールアクセス .....	84
6.16.14. ウィジェットプロパティ .....	85
6.16.15. リスト .....	86
6.16.16. イベント .....	87
6.16.17. 文字列の書式設定 .....	89
6.16.18. 標準ライブラリ .....	89
6.17. スクリプト値 .....	89
6.18. ショートカット、ボタン、アイコン .....	91
6.18.1. ショートカット .....	91
6.18.2. コマンドラインオプション .....	92
6.18.2.1. のコマンドラインオプション Studio.Console.exe .....	92
6.18.2.2. のコマンドラインオプション Monitor.Console.exe .....	93
6.18.3. ボタン .....	94
6.18.4. アイコン .....	95
6.19. スキン .....	96
6.20. ステートマシンとステート .....	96
6.20.1. ステートマシン .....	96
6.20.1.1. ハプティックステートマシン .....	96
6.20.1.2. ロジックステートマシン .....	97

6.20.1.3. 動的ステートマシン	97
6.20.2. ステート	97
6.20.2.1. 混合ステート	97
6.20.2.2. ビューステート	99
6.20.2.3. 初期ステート	99
6.20.2.4. 最終ステート	100
6.20.2.5. 選択ステート	101
6.20.2.6. 履歴ステート	102
6.20.3. 遷移	105
6.20.4. ステートマシンの実行	109
6.20.5. EB GUIDE の記法とUML記法の比較	113
6.20.5.1. サポートされている要素	114
6.20.5.2. サポートされない要素	114
6.20.5.3. 偏差	114
6.21. タッチ入力	115
6.21.1. 非パスジェスチャー	115
6.21.2. パスジェスチャー	115
6.21.3. 入力処理とジェスチャー	116
6.21.4. マルチタッチ入力	116
6.22. ウィジェット	117
6.22.1. ビュー	117
6.22.2. ウィジェットのカテゴリ	118
6.22.3. ウィジェットプロパティ	120
6.22.4. ウィジェットテンプレート	121
6.22.5. ウィジェット機能	122
6.22.5.1. フォーカスウィジェット機能カテゴリ	123
6.22.5.2. リスト管理ウィジェット機能カテゴリ	124
7. ヒューマンマシンインターフェースの動作のモデル化	126
7.1. ステートマシンのモデリング	126
7.1.1. ステートマシンの追加	126
7.1.2. 動的ステートマシンの追加	126
7.1.3. ステートマシンに対するエントリーアクションの定義	127
7.1.4. ステートマシンに対する終了アクションの定義	128
7.1.5. ステートマシンの削除	128
7.2. モデリングステート	128
7.2.1. ステートの追加	128
7.2.2. 混合ステートへのステートの追加	129
7.2.3. 選択ステートの追加	130
7.2.4. ステートに対するエントリーアクションの定義	132
7.2.5. ステートに対する終了アクションの定義	132
7.2.6. ステートマシンからのモデル要素の削除	133
7.3. ステート間を遷移で接続	133

7.3.1. 2つの状態間に遷移を追加 .....	133
7.3.2. 遷移の移動 .....	134
7.3.3. 遷移に対するトリガーの定義 .....	135
7.3.4. 遷移への条件の追加 .....	136
7.3.5. 遷移へのアクションの追加 .....	137
7.3.6. ステートへの内部遷移の追加 .....	139
8. ヒューマンマシンインターフェースの外観のモデル化 .....	140
8.1. ウィジェットの操作 .....	140
8.1.1. ビューの追加 .....	140
8.1.2. ビューへの基本ウィジェットの追加 .....	141
8.1.2.1. 四角形を追加する .....	141
8.1.2.2. 楕円を追加する .....	141
8.1.2.2.1. 楕円を編集する .....	141
8.1.2.3. イメージを追加する .....	142
8.1.2.4. ラベルを追加する .....	144
8.1.2.4.1. ラベルのフォントの変更 .....	145
8.1.2.5. コンテナを追加する .....	145
8.1.2.6. インスタンスエータの追加 .....	146
8.1.2.7. アニメーションの追加 .....	148
8.1.2.8. スクリプト曲線によるアニメーションの追加 .....	149
8.1.2.9. アルファマスクの追加 .....	151
8.1.3. ビューへの3Dウィジェットの追加 .....	152
8.1.3.1. ビューへのシーングラフの追加 .....	152
8.1.4. ビューに.psdファイルをインポートする .....	153
8.1.5. .psdファイルからイメージを抽出する .....	154
8.1.6. IBLファイルのインポート .....	155
8.1.7. ビューからのウィジェットの削除 .....	156
8.2. ウィジェットプロパティの操作 .....	157
8.2.1. ウィジェットの配置 .....	157
8.2.2. ウィジェットのサイズの変更 .....	158
8.2.3. ウィジェットプロパティ間のリンク設定 .....	159
8.2.4. ウィジェットプロパティとデータプールアイテムのリンク設定 .....	160
8.2.5. ウィジェットへのユーザー定義プロパティの追加 .....	162
8.2.5.1. タイプのユーザー定義プロパティの追加 <code>Function () : bool</code> .....	163
8.2.6. ユーザー定義プロパティの名前の変更 .....	164
8.2.7. タイプリストのプロパティの編集 .....	165
8.2.8. マルチフォントサポートの管理 .....	166
8.2.9. ウィジェットの順序および可視性の管理 .....	168
8.3. ウィジェット機能を追加してウィジェットを拡張する .....	170
8.3.1. ウィジェット機能の追加 .....	170
8.3.2. ウィジェット機能の削除 .....	172
8.4. 言語サポートとの連携 .....	173

8.4.1. EB GUIDEモデルへの言語の追加 .....	174
8.4.2. データプールアイテムに言語サポートを追加する .....	175
8.4.3. 言語の削除 .....	175
8.5. スキンのサポートの操作 .....	176
8.5.1. EB GUIDEモデルへのスキンの追加 .....	176
8.5.2. データプールアイテムにスキンのサポートを追加する .....	177
8.5.3. スキンを切り替える .....	178
8.5.4. スキンを削除する .....	178
8.6. ビュー遷移のアニメーション化 .....	179
8.6.1. 開始アニメーションの追加 .....	179
8.6.2. 変更アニメーションの追加 .....	179
8.6.3. アニメーションの再配置 .....	180
8.7. ウィジェットの再利用 .....	181
8.7.1. テンプレートの追加 .....	181
8.7.2. テンプレートインターフェースの定義 .....	182
8.7.3. テンプレートの使用 .....	183
8.7.4. テンプレートの削除 .....	183
9. データの処理 .....	185
9.1. イベントの追加 .....	185
9.2. イベントへのパラメータの追加 .....	185
9.3. イベントへの対応 .....	186
9.4. イベントへのキーのマッピング .....	187
9.5. イベントの削除 .....	188
9.6. データプールアイテムの追加 .....	189
9.7. リストタイプのデータプールアイテムの編集 .....	189
9.8. プロパティのスクリプト値への変換 .....	190
9.9. 外部通信の確立 .....	191
9.10. データプールアイテム間のリンク設定 .....	192
9.11. データプールアイテムの削除 .....	193
9.12. 名前空間の操作 .....	194
9.12.1. 名前空間の追加 .....	194
9.12.2. 名前空間へのモデル要素の追加 .....	195
9.12.3. 名前空間どうしでのモデル要素の移動 .....	195
9.12.4. 名前空間の削除 .....	196
10. プロジェクトの処理 .....	197
10.1. プロジェクトの作成 .....	197
10.2. プロジェクトを開く .....	198
10.2.1. ファイルエクスプローラからプロジェクトを開く .....	198
10.2.2. プロジェクトをEB GUIDE Studioに開く .....	198
10.3. モデル要素の名前を変更する .....	199
10.4. EB GUIDEモデルの検証およびモデル実行 .....	199
10.4.1. EB GUIDEモデルの検証 .....	200



10.4.1.1. EB GUIDE Studioを使用したEB GUIDEモデルの検証 .....	200
10.4.1.2. コマンドラインを使用したEB GUIDEモデルの検証 .....	201
10.4.2. シミュレーションの開始と停止 .....	201
10.5. EB GUIDEモデルのエクスポート .....	201
10.5.1. EB GUIDE Studioを使用したEB GUIDEモデルのエクスポート .....	202
10.5.2. コマンドラインを使用したEB GUIDEモデルのエクスポート .....	202
10.6. EB GUIDE Studioの表示言語の変更 .....	203
10.7. プロファイルの設定 .....	203
10.7.1. プロファイルの追加 .....	203
10.7.2. ライブラリの追加 .....	204
10.7.3. シーンの設定 .....	206
10.8. 言語依存テキストのエクスポートとインポート .....	207
10.8.1. 言語依存テキストのエクスポート .....	208
10.8.2. 言語依存テキストのインポート .....	209
10.8.2.1. EB GUIDE Studioを使用した言語依存テキストのインポート .....	209
10.8.2.2. コマンドラインを使用した言語依存テキストのインポート .....	210
10.9. EB GUIDE Monitorを操作する .....	211
10.9.1. EB GUIDE Monitorでのイベント発行 .....	211
10.9.2. EB GUIDE Monitorでのデータプールアイテムの値の変更 .....	212
10.9.3. EB GUIDE Monitorでのスクリプトの開始 .....	213
10.9.3.1. EB GUIDE Monitorでのスクリプトファイルの記述 .....	213
10.9.4. スタンドアロンアプリケーションとしてEB GUIDE Monitorを起動する .....	218
10.9.5. EB GUIDE Monitorへの設定の読み込み .....	219
10.9.6. ウォッチリストのエクスポートおよびインポート .....	220
11. ベストプラクティス .....	222
11.1. ベストプラクティス: Conditional script .....	222
12. チュートリアル .....	223
12.1. チュートリアル: 動的ステートマシンの追加 .....	223
12.2. チュートリアル: EB GUIDEスクリプトを使用したボタン動作のモデル化 .....	231
12.3. チュートリアル: パスジェスチャーをモデル化する .....	238
12.4. チュートリアル: 動的コンテンツを使用したリストの作成 .....	241
12.5. チュートリアル: 画面内で楕円を移動する .....	249
12.6. チュートリアル: データプールアイテムに言語依存テキストを追加する .....	251
12.7. チュートリアル: 3Dグラフィックの操作 .....	255
12.8. チュートリアル: ガンマの正しいレンダリング .....	260
12.9. チュートリアル: 表示遷移アニメーションの使用 .....	263
12.10. チュートリアル: アニメーションでのスクリプト曲線の使用 .....	272
13. リファレンス .....	277
13.1. Androidイベント .....	277
13.2. データプールアイテム .....	278
13.3. データタイプ .....	278
13.3.1. ブール値 .....	278

13.3.2. 色 .....	279
13.3.3. 条件スクリプト .....	279
13.3.4. 浮動小数点数 .....	280
13.3.5. フォント .....	280
13.3.6. Function () : bool .....	281
13.3.7. IBL .....	281
13.3.8. イメージ .....	281
13.3.9. 整数 .....	281
13.3.10. メッシュ .....	282
13.3.11. 文字列 .....	282
13.3.12. リスト .....	283
13.4. EB GUIDEスクリプト .....	284
13.4.1. EB GUIDEスクリプトのキーワード .....	284
13.4.2. EB GUIDEスクリプト 演算子の優先順位 .....	285
13.4.3. EB GUIDEスクリプト標準ライブラリ .....	285
13.4.3.1. EB GUIDEスクリプトの関数A~B .....	286
13.4.3.1.1. abs .....	286
13.4.3.1.2. absf .....	286
13.4.3.1.3. acosf .....	286
13.4.3.1.4. animation_before .....	286
13.4.3.1.5. animation_beyond .....	287
13.4.3.1.6. animation_cancel .....	287
13.4.3.1.7. animation_cancel_end .....	287
13.4.3.1.8. animation_cancel_reset .....	287
13.4.3.1.9. animation_pause .....	288
13.4.3.1.10. animation_play .....	288
13.4.3.1.11. animation_reverse .....	288
13.4.3.1.12. animation_running .....	288
13.4.3.1.13. animation_set_time .....	289
13.4.3.1.14. asinf .....	289
13.4.3.1.15. atan2f .....	289
13.4.3.1.16. atan2i .....	290
13.4.3.1.17. atanf .....	290
13.4.3.1.18. bool2string .....	290
13.4.3.2. EB GUIDEスクリプトの関数C~H .....	290
13.4.3.2.1. ceil .....	290
13.4.3.2.2. changeDynamicStateMachinePriority .....	291
13.4.3.2.3. character2unicode .....	291
13.4.3.2.4. clampf .....	291
13.4.3.2.5. clampi .....	292
13.4.3.2.6. clearAllDynamicStateMachines .....	292
13.4.3.2.7. color2string .....	292

13.4.3.2.8. cosf .....	293
13.4.3.2.9. deg2rad .....	293
13.4.3.2.10. expf .....	293
13.4.3.2.11. float2string .....	293
13.4.3.2.12. floor .....	294
13.4.3.2.13. fmod .....	294
13.4.3.2.14. focusMoveTo .....	294
13.4.3.2.15. focusNext .....	294
13.4.3.2.16. focusPrevious .....	295
13.4.3.2.17. format_float .....	295
13.4.3.2.18. format_int .....	296
13.4.3.2.19. frac .....	296
13.4.3.2.20. getConfigItem .....	297
13.4.3.2.21. getFontAscender .....	297
13.4.3.2.22. getFontDescender .....	297
13.4.3.2.23. getFontLineGap .....	298
13.4.3.2.24. getImageHeight .....	298
13.4.3.2.25. getImageWidth .....	298
13.4.3.2.26. getLabelTextHeight .....	298
13.4.3.2.27. getLabelTextWidth .....	299
13.4.3.2.28. getLineCount .....	299
13.4.3.2.29. getLineHeight .....	299
13.4.3.2.30. getProductString .....	300
13.4.3.2.31. getTextHeight .....	300
13.4.3.2.32. getTextLength .....	300
13.4.3.2.33. getTextWidth .....	301
13.4.3.2.34. getVersionString .....	301
13.4.3.2.35. has_list_window .....	301
13.4.3.2.36. hsba2color .....	302
13.4.3.3. EB GUIDEスクリプトの関数I〜R .....	302
13.4.3.3.1. int2float .....	302
13.4.3.3.2. int2string .....	302
13.4.3.3.3. isDynamicStateMachineActive .....	303
13.4.3.3.4. isWidgetOnActiveStatemachine .....	303
13.4.3.3.5. language .....	303
13.4.3.3.6. lerp .....	303
13.4.3.3.7. localtime_day .....	304
13.4.3.3.8. localtime_hour .....	304
13.4.3.3.9. localtime_minute .....	304
13.4.3.3.10. localtime_month .....	305
13.4.3.3.11. localtime_second .....	305
13.4.3.3.12. localtime_weekday .....	305

13.4.3.3.13. localtime_year .....	305
13.4.3.3.14. log10f .....	306
13.4.3.3.15. logf .....	306
13.4.3.3.16. maxf .....	306
13.4.3.3.17. maxi .....	306
13.4.3.3.18. minf .....	307
13.4.3.3.19. mini .....	307
13.4.3.3.20. nearbyint .....	307
13.4.3.3.21. popDynamicStateMachine .....	308
13.4.3.3.22. powf .....	308
13.4.3.3.23. pushDynamicStateMachine .....	308
13.4.3.3.24. rad2deg .....	308
13.4.3.3.25. rand .....	309
13.4.3.3.26. rgba2color .....	309
13.4.3.3.27. round .....	309
13.4.3.4. EB GUIDEスクリプトの関数S~W .....	310
13.4.3.4.1. saturate .....	310
13.4.3.4.2. seed_rand .....	310
13.4.3.4.3. shutdown .....	310
13.4.3.4.4. sinf .....	310
13.4.3.4.5. skin .....	311
13.4.3.4.6. smoothstep .....	311
13.4.3.4.7. sqrtf .....	311
13.4.3.4.8. string2float .....	312
13.4.3.4.9. string2int .....	312
13.4.3.4.10. string2string .....	312
13.4.3.4.11. substring .....	313
13.4.3.4.12. system_time .....	313
13.4.3.4.13. system_time_ms .....	313
13.4.3.4.14. tanf .....	314
13.4.3.4.15. trace_dp .....	314
13.4.3.4.16. trace_string .....	314
13.4.3.4.17. transformToScreenX .....	314
13.4.3.4.18. transformToScreenY .....	315
13.4.3.4.19. transformToWidgetX .....	315
13.4.3.4.20. transformToWidgetY .....	315
13.4.3.4.21. trunc .....	316
13.4.3.4.22. widgetGetChildCount .....	316
13.5. イベント .....	316
13.5.1. キーイベントの10進コード .....	317
13.6. model.json設定ファイル .....	318
13.6.1. EB GUIDE Studioでのサンプルmodel.json .....	327

13.7. OpenGL ES拡張機能 .....	328
13.8. platform.json設定ファイル .....	331
13.8.1. EB GUIDE Studioでのサンプルplatform.json .....	334
13.9. シーン .....	335
13.10. がサポートするタッチスクリーンタイプ EB GUIDE GTF .....	338
13.11. tracing.json設定ファイル .....	339
13.11.1. 重大度レベル .....	341
13.11.2. tracing.jsonの例 .....	342
13.12. ウィジェット .....	342
13.12.1. ビュー .....	342
13.12.2. 基本ウィジェット .....	343
13.12.2.1. アルファマスク .....	344
13.12.2.2. アニメーション .....	345
13.12.2.2.1. コンスタント曲線 .....	346
13.12.2.2.2. 高速開始曲線 .....	346
13.12.2.2.3. 低速開始曲線 .....	347
13.12.2.2.4. 二次曲線 .....	347
13.12.2.2.5. 正弦曲線 .....	348
13.12.2.2.6. スクリプト曲線 .....	348
13.12.2.2.7. リニア曲線 .....	349
13.12.2.2.8. リニア補間曲線 .....	349
13.12.2.3. コンテナ .....	350
13.12.2.4. 楕円 .....	350
13.12.2.5. イメージ .....	351
13.12.2.6. インスタンスエータ .....	351
13.12.2.7. ラベル .....	352
13.12.2.8. 四角形 .....	352
13.12.3. アニメーション .....	353
13.12.3.1. アニメーション .....	353
13.12.3.2. コンスタント曲線 .....	354
13.12.3.3. 高速開始曲線 .....	354
13.12.3.4. 低速開始曲線 .....	355
13.12.3.5. 二次曲線 .....	355
13.12.3.6. 正弦曲線 .....	356
13.12.3.7. スクリプト曲線 .....	357
13.12.3.8. リニア曲線 .....	357
13.12.3.9. リニア補間曲線 .....	357
13.12.4. 3Dウィジェット .....	358
13.12.4.1. 環境光 .....	358
13.12.4.2. カメラ .....	358
13.12.4.3. 指向性ライト .....	359
13.12.4.4. イメージベースドライト .....	359

13.12.4.5. 材質 .....	360
13.12.4.6. メッシュ .....	360
13.12.4.7. PBR GGX材質 .....	361
13.12.4.8. PBR Phong材質 .....	362
13.12.4.9. 点ライト .....	363
13.12.4.10. シーングラフ .....	363
13.12.4.11. シーングラフノード .....	364
13.12.4.12. スポットライト .....	365
13.13. ウィジェット機能 .....	365
13.13.1. 共通 .....	365
13.13.1.1. 子の可視性の選択 .....	365
13.13.1.2. 有効 .....	366
13.13.1.3. フォーカス .....	366
13.13.1.4. 複数行 .....	366
13.13.1.5. 押下 .....	367
13.13.1.6. 選択 .....	367
13.13.1.7. 選択グループ .....	368
13.13.1.8. スピン .....	368
13.13.1.9. テキストの切り捨て .....	369
13.13.1.10. タッチ .....	369
13.13.2. 効果 .....	370
13.13.2.1. 枠 .....	370
13.13.2.2. 配色 .....	371
13.13.3. フォーカス .....	371
13.13.3.1. 自動フォーカス .....	372
13.13.3.2. ユーザー定義フォーカス .....	372
13.13.4. ジェスチャー .....	373
13.13.4.1. フリックジェスチャー .....	373
13.13.4.2. ホールドジェスチャー .....	374
13.13.4.3. ロングホールドジェスチャー .....	374
13.13.4.4. パスジェスチャー .....	375
13.13.4.4.1. ジェスチャーID .....	376
13.13.4.5. ピンチジェスチャー .....	377
13.13.4.6. 回転ジェスチャー .....	377
13.13.5. 入力処理 .....	378
13.13.5.1. ジェスチャー .....	378
13.13.5.2. キー押下 .....	378
13.13.5.3. キーリリース .....	379
13.13.5.4. キーのステータス変更 .....	379
13.13.5.5. キーUnicode .....	380
13.13.5.6. 内部へ移動 .....	380
13.13.5.7. 外部へ移動 .....	381

13.13.5.8. 内部で移動 .....	381
13.13.5.9. 可動 .....	381
13.13.5.10. 回転 .....	382
13.13.5.11. タッチの喪失 .....	382
13.13.5.12. タッチ移動 .....	383
13.13.5.13. タッチ押下 .....	383
13.13.5.14. タッチリリース .....	384
13.13.5.15. タッチのステータス変更 .....	384
13.13.6. レイアウト .....	385
13.13.6.1. 絶対レイアウト .....	385
13.13.6.2. ボックスレイアウト .....	386
13.13.6.3. フローレイアウト .....	386
13.13.6.4. グリッドレイアウト .....	387
13.13.6.5. レイアウト余白 .....	388
13.13.6.6. リストレイアウト .....	388
13.13.6.7. 拡大縮小モード .....	390
13.13.7. リスト管理 .....	390
13.13.7.1. ラインインデックス .....	390
13.13.7.2. リストインデックス .....	391
13.13.7.3. テンプレートインデックス .....	391
13.13.7.4. ビューポート .....	391
13.13.8. 3D .....	392
13.13.8.1. カメラビューポート .....	392
13.13.8.2. アンビエントテクスチャ .....	392
13.13.8.3. ディフューズテクスチャ .....	393
13.13.8.4. エミッシブテクスチャ .....	394
13.13.8.5. ライトマップテクスチャ .....	395
13.13.8.6. 金属テクスチャ .....	396
13.13.8.7. ノーマルマップテクスチャ .....	397
13.13.8.8. 不透明テクスチャ .....	398
13.13.8.9. リフレクションテクスチャ .....	399
13.13.8.10. ラフネス(粗さ)テクスチャ .....	400
13.13.8.11. 光沢テクスチャ .....	401
13.13.8.12. スペキュラテクスチャ .....	402
13.13.8.13. Texture coordinate transformation .....	403
13.13.8.14. トーンマッピング .....	404
13.13.9. 変換 .....	405
13.13.9.1. ピボット .....	405
13.13.9.2. 回転 .....	405
13.13.9.3. 拡大縮小 .....	406
13.13.9.4. せん断 .....	406
13.13.9.5. 変換 .....	406

14. EB GUIDE Studioのインストール .....	408
14.1. バックグラウンド情報 .....	408
14.1.1. 制限 .....	408
14.1.2. システム要件 .....	408
14.2. EB GUIDEのダウンロード .....	409
14.3. EB GUIDEのインストール .....	409
14.4. アンインストールする EB GUIDE .....	410
用語集 .....	412
インデックス .....	417



# 1. 本書について

## 1.1. 対象者: モデラー

モデラーは、ヒューマンマシンインターフェース(HMI)を作るためにEB GUIDE Studioを使用します。EB GUIDEでは、HMIはEB GUIDEモデルと呼ばれます。アプリケーションと通信するには、イベントメカニズムを使ってイベントを確認する方法、データプールを使ってデータプールアイテムを利用する方法、ユーザー固有のEB GUIDEスクリプト関数を呼び出す方法のいずれかを使用します。

モデラーは、以下のタスクを担います。

- ▶ ウィジェットとビューのアーキテクチャを使用して、ディスプレイのグラフィカル要素を指定します。
- ▶ ヒューマンマシンインターフェースの最適化について、デザイナーおよびユーザビリティの専門家と打ち合わせをします。
- ▶ グラフィカル要素をいつ表示するかをステートマシン機能を使って指定します。
- ▶ コントロールパネルやタッチスクリーンなどの装置から入力を受け取った要素がどのように反応するかを定義します。
- ▶ これらの要素が、情報をハードウェア、またはソフトウェアアプリケーション(ナビゲーションユニットのようなサービスを提供するもの)から受け取る方法を定義します。
- ▶ モデル要素間や、入力デバイス、出力デバイスとのインターフェースを定義します。

モデラーには以下に関する深い知識があります。

- ▶ EB GUIDE Studio 機能
- ▶ UMLステートマシンの概念
- ▶ ドメインの仕様と要件
- ▶ やり取りされるデータとEB GUIDE GTF通信メカニズム
- ▶ プロジェクトで3Dグラフィックを使用している場合、3Dグラフィックの仕様

## 1.2. ユーザーマニュアルの構成

次の構成で情報が記されています。

- ▶ バックグラウンド情報

バックグラウンド情報では、具体的なトピックや重要な事実を紹介しています。こうした情報により、関連する手順を実行できます。

▶ 手引書

この手引書では、具体的なタスクを段階的に説明し、EB GUIDEの使用方法を示します。手順は、タイトルで体言止め(英語では動詞のing形)になっているものです(例えば、「EB GUIDE Studioの起動」)。

▶ チュートリアル

チュートリアルは手引書を拡張したものです。複雑なタスクの説明が記されています。チュートリアルの見出しは「チュートリアル:」で始まります(例えば、「チュートリアル: ボタンの作成」)。

▶ リファレンス

リファレンスは、詳細な技術的パラメータおよび表、それにEB GUIDE Monitorアプリケーションプログラミングインターフェイスに関するドキュメントを提供します。

▶ デモ

デモは、アプリケーションの記述方法や対話処理の流れを理解するのに役立ちます。デモはEB GUIDE GTF SDKに含まれています。

## 1.3. 本書で使用する表記スタイル

このドキュメントでは、重要な情報を示すために次の絵文字およびシグナル語が使用されます。

シグナル語**警告**は、正常に設定するための必須の情報を示します。

警告



ソースと問題の種類

ソフトウェアに発生する可能性があること

その問題によってどのような状態になるか

問題を回避する方法

シグナル語**注意**は、対象事項に関する重要な情報を示します。

注記



重要な情報

対象事項に関する重要な情報を示します。

シグナル語**ヒント**は、役に立つヒント、コツ、およびショートカットを提供します。

ティップ



役立つヒント  
役立つヒントを示します

このドキュメントでは、一部の単語や語句を[太字]、**斜体**、または等幅フォントで表記しています。

これらの表記の意味については、次の例をご覧ください。

デフォルトのテキストは、Arial Regularフォントで表記します。

フォント	説明	使用例
斜体	新しい用語または重要な用語を強調する	設定の <b>基本構成要素</b> は、モジュール設定です。
太字	GUI要素およびキーボードのキー	1. [Project]ドロップダウンリストボックスで Project_Aを選択します。 2. [Enter]キーを押します。
等幅フォント (Courier)	ファイル名やディレクトリ名、および章のタイトルを示す	スクリプトの配置先: <code>function_name \abcdirectory</code> 。
等幅フォント (Courier)	ユーザーの入力、コード、およびファイルのディレクトリ	<code>CC_FILES_TO_BUILD = (PROJECT_- PATH) \source\network\can_node.- c CC_FILES_TO_BUILD += \$(PROJECT_- PATH) \source\network\can_config.c</code>  このモジュールは <code>BswM_Dcm_- RequestSessionMode()</code> 関数を呼び出します。  プロジェクト名は、Project_Testと入力します。
角括弧 [ ]	オプションのパラメータがあるコマンド構文でオプションパラメータを示す	<code>insertBefore [&lt;opt&gt;]</code>
波括弧 { }	必須のパラメータがあるコマンド構文で必須パラメータを示す	<code>insertBefore {&lt;file&gt;}</code>
三点リーダー	複数のパラメータを持つコマンド構文の場合に、パラメータがさらに続くことを示す	<code>insertBefore [&lt;opt&gt;...]</code>
破断線	使用可能なパラメータのいずれかを選択するコマンド構文で、使用可能なすべてのパラメータを示す	<code>allowinvalidmarkup {on off}</code>



これは操作方法一を示しています。

足あとの付いたバーの下には操作方法を段階的に説明する内容が続きます。

前提条件:

- この行には操作方法の前提条件を示します。

#### ステップ 1

操作方法の説明が入ります。

#### ステップ 2

操作方法の説明が入ります。

#### ステップ 3

操作方法の説明が入ります。

## 1.4. ネーミングルール

EB GUIDEドキュメンテーションでは、以下のディレクトリ名を使用します。

- ▶ EB GUIDEのインストール先ディレクトリは、`$GUIDE_INSTALL_PATH`と表されます。

例えば、次のようになります。

```
C:/Program Files/Elektrobit/EB GUIDE Studio 6.8
```

- ▶ EB GUIDE SDKプラットフォームのディレクトリは、`$GTF_INSTALL_PATH`と表されます。名前のパターンは、`$GTF_INSTALL_PATH/platform/<platform name>`となります。

例えば、次のようになります。

```
C:/Program Files/Elektrobit/EB GUIDE Studio 6.8/platform/win64
```

- ▶ EB GUIDEプロジェクトの保存先ディレクトリは、`$GUIDE_PROJECT_PATH`と表されます。

例えば、次のようになります。

```
C:/Users/[user name]/Documents/EB GUIDE 6.8/projects/
```

- ▶ EB GUIDEモデルのエクスポート先ディレクトリは、`$EXPORT_PATH`と表されます。

## 1.5. パスに関する規則

EB GUIDE Studioは、Windows 10で260文字を超えるパス名を扱えます。フルパス名は260文字を超えることができても、パス内の単一のファイルまたはディレクトリの名前には依然として248文字の制限があります。

注記



### Windows 7の長いパス名

Windows 7では、長いパス名は扱えません。長いパス名を使用するには、Windows 10でEB GUIDE Studioを実行します。Windows 10で長いパス名を有効にする方法については、Windows 10のドキュメントをご覧ください。

## 2. 安全で正しい使い方

### 2.1. 使用目的

- ▶ EB GUIDE StudioおよびEB GUIDE GTFは、インフォテイメントヘッドユニット、計器パネル、特定の工業アプリケーション向けにヒューマンマシンインターフェースを開発するプロジェクトで使うことを意図しています。
- ▶ ライセンスでカバーされる範囲によりますが、主なユースケースは、大量生産、受注生産、プロトタイプングでの使用です。

### 2.2. 考えられる誤用

#### WARNING



#### 誤用の可能性と賠償責任

お客様は、このソフトウェアを、意図された用途に従い、適用可能なライセンス使用許諾契約書で許可されたとおりに常に使用するものとします。Elektrobit Automotive GmbHは、一切の賠償責任を負わないものとし、このソフトウェアが適用可能なライセンス使用許諾契約書に従わずに使用された場合も、それについて責任を保持しません。

- ▶ Elektrobit Automotive GmbHから提供されるEB GUIDE product lineは、ISO 26262/A-SILに規定された安全関連システムにヒューマンマシンインターフェースを実装する目的には使用しないでください。
- ▶ EB GUIDE product lineは、DO-178B、SIL、A-SILなどの認定が必要とされる安全関連システムでの使用を意図していません。

このような環境でEB GUIDE GTFを使用することを禁じます。具体的なアプリケーションに対する適合性が不確かな場合は、[第3章「サポート」](#)の説明に従ってElektrobit Automotive GmbHまでお問い合わせください。

## 3. サポート

EB GUIDEサポートは次のような形で利用できます。

▶ コミュニティエディションの場合:

当社の記事、ブログ、およびフォーラム内の包括的な情報を検索します。

▶ エンタープライズエディションの場合:

サポート上の合意事項に従って当社までお問い合わせください。

サポートが必要な場合は、EB GUIDEインストールのバージョン番号をお手元にご用意ください。バージョン番号を確認するには、プロジェクトセンターに移動し、[ヘルプ]をクリックします。ダイアログの右下隅にバージョン番号が記されています。

## 4. EB GUIDEの基本

EB GUIDEは、ヒューマンマシンインターフェース(HMI)の開発プロセスに携わるユーザーをさまざまな局面で支援します。EB GUIDE商品ラインは、グラフィカルユーザーインターフェースまたは音声ユーザーインターフェース向けのツールとプラットフォームを提供しています。EB GUIDE商品ラインは、インフォテイメントヘッドユニット、計器パネル、特定の工業アプリケーション向けにユーザーインターフェースを開発するプロジェクトで使うことを意図しています。主に、大量生産、受注生産、プロトタイプングで使用されます。

### 4.1. EB GUIDE product line

EB GUIDE product lineは、以下のソフトウェア要素から構成されます。

- ▶ EB GUIDE Studio
- ▶ EB GUIDE TF

EB GUIDE Studioは、PCで動作するモデリングツールです。EB GUIDE Studioを使うと、機能へのアクセスをユーザーに提供する中心的な制御要素としてヒューマンマシンインターフェース機能全体をモデリングできます。

EB GUIDE TFは、EB GUIDE Studioで作成されたEB GUIDEモデルを実行します。EB GUIDE TFは、開発用PCや各種組み込みハードウェアで使用できます。

EB GUIDE Studioで作成されたEB GUIDEモデルと、EB GUIDE TFで実行されるエクスポートされたEB GUIDEモデルは完全に分離されます。相互に対話は行われますが、相手側の動作をブロックすることはできません。

### 4.2. EB GUIDE Studio

#### 4.2.1. ヒューマンマシンインターフェースの動作のモデル化

EB GUIDEモデルの動的な動作は、ステートマシンにステートを配置し、複数のステートを組み合わせることで指定します。

##### ステートマシン

ステートマシンは、決定性有限オートマトンであり、システムの動的な動作を表します。EB GUIDE Studioでは、ハプティックステートマシンなど、複数のタイプのステートマシンを使用できます。ハプティックステートマシンを使うと、グラフィカルユーザーインターフェースを指定できます。

##### ステート

ステートは遷移でリンクされます。遷移は、ステート間の接続であり、ステートの変化をトリガーします。



## 4.2.2. ヒューマンマシンインターフェースの外観のモデル化

EB GUIDE Studioで、EB GUIDEモデルのグラフィカルユーザーインターフェースと音声ユーザーインターフェースを定義します。

### ウィジェット

グラフィカルユーザーインターフェースを作成するために、EB GUIDE Studioはウィジェットを提供しています。ウィジェットは、見え方を定義するモデル要素です。テキストラベルやイメージなどの情報を表示するのに主に使用します。また、ボタンやスライダーなど、システムの動作を操作する手段も提供できます。複数のウィジェットを組み合わせて、ビューと呼ばれる構造を作成します。

### スピージェット

音声ユーザーインターフェースを作成するために、EB GUIDE Studioはスピージェットを提供しています。スピージェットを使用して、音声ダイアログの基本要素を指定します。ユーザー入力としてのスピーチ認識とシステム出力としてのスピーチ合成です。プロンプトスピージェットを使って、Text-to-Speechシンセサイザ(TTS)で再生されるテキストをモデリングできます。コマンドスピージェットは、スピーチ認識機能が理解できる文法のモデリングに使用します。関連性のあるスピージェットは、モデル要素全体でグループ化されています。このグループをトークと呼びます。

## 4.2.3. データの処理

ヒューマンマシンインターフェースとアプリケーションとの間の通信は、データプールとイベントシステムを使って実装されています。

### データプール

データプールは、すべての表示データと詳細な内部情報を格納する組み込みのデータベースです。データプールアイテムはデータを格納し、やり取りします。

### イベントシステム

イベントは一時的なトリガーです。イベントは双方に送信され、特定の事態が発生したことを通知します。

アプリケーションソフトウェアは、アプリケーションプログラミングインターフェースを通じてイベントやデータプールにアクセスできます。

## 4.2.4. EB GUIDEモデルのシミュレート

EB GUIDE Studioでは、シミュレーション中にEB GUIDEモデルの機能をテストできます。マウスでクリックするだけでシミュレーションを開始し、EB GUIDEモデルがどのように見えるかすぐに確認できます。

シミュレーションは、マウス、キーボード、タッチスクリーンなどの入力デバイスを使用して操作することができます。

EB GUIDE Monitorを使用してEB GUIDEモデルを制御し、次の操作を実行することもできます。

- ▶ データプールアイテムの値を変更することによって、表示されるデータを変更する
- ▶ イベントを発行することによって、ユーザー入力をモデル実行する
- ▶ ログのすべての変更を追跡する
- ▶ スクリプトを開始する

EB GUIDE Monitorは、スタンドアロンアプリケーションとして使用することもできます。

### 4.2.5. EB GUIDEモデルのエクスポート

EB GUIDEモデルを対象デバイスで使うには、EB GUIDEモデルをEB GUIDE Studioからエクスポートし、対象デバイスで認識できる形式に変換する必要があります。エクスポートの過程で、すべての関連データが一連のASCIIファイルとしてエクスポートされます。

## 4.3. EB GUIDE TF

EB GUIDE TFは、EB GUIDEモデルの実行に必要なGtfStartup実行可能ファイルと一連のライブラリで構成されています。

EB GUIDE Studioで選択したプロジェクトタイプに応じて、以下を実行します。

- ▶ EB GUIDE GTF

EB GUIDE Graphics Target Frameworkは、グラフィカルヒューマンマシンインターフェースモデルを実行するランタイム環境です。

- ▶ EB GUIDE STF

EB GUIDE Speech Target Frameworkは、ヒューマンマシンインターフェースに含まれるスピーチ機能を実行するランタイム環境です。

EB GUIDE TFのプログラムコードは、大半がプラットフォームに依存しません。別のシステムへのコードの移植は、非常に簡単です。

EB GUIDEモデルファイルを入れ替えるだけで、ヒューマンマシンインターフェース全体を入れ替えることが可能です。EB GUIDE TFの再コンパイルは不要です。変更したEB GUIDEモデルをEB GUIDE Studioから再エクスポートするだけで、必要な作業は完了です。

EB GUIDE TFでは、次のハードウェア抽象化を使用します。

- ▶ OSの抽象化

オペレーティングシステム(OS)のプラットフォーム依存関係は、OSアブストラクションレイヤー(GtFOSAL)でカプセル化されています。EB GUIDE TFで利用されるオペレーティングシステム機能には、ファイルシステムやTCPソケットなどがあります。

▶ GLの抽象化

グラフィックサブシステムのプラットフォーム依存関係は、レンダラーにカプセル化されています。EB GUIDEモデルには、ジオメトリやライティングなどの情報を持つ要素プロパティが含まれます。エクスポートされたEB GUIDEモデルに含まれるデータは、レンダラーに渡され、処理されてデジタルイメージに出力されます。レンダラーは、ハードウェアで提供される実際のグラフィカルシステムを抽象化したものです。EB GUIDE TFは、異なるハードウェアに対応するさまざまなレンダラーをサポートしています。

▶ オーディオの抽象化

音声ユーザーインターフェースは、オーディオハードウェアにアクセスする必要があります。オーディオの抽象化によって、マイクとスピーカーへのアクセスが可能になります。EB GUIDE STFには、音声認識とText-to-Speech音声出力機能が実装されています。この目的で、EB GUIDE STFにはサードパーティの音声エンジンが組み込まれています。

## 5. チュートリアル: はじめに

### NOTE



#### デフォルトのウィンドウレイアウト

すべての手順およびスクリーンショットでは、デフォルトのウィンドウレイアウトが使用されています。手順に従う場合は、EB GUIDE Studioウィンドウをレイアウト > **Reset to default layout**を選択することによって、デフォルトレイアウトにリセットすることをお勧めします。

このセクションでは、EB GUIDE Studioによるヒューマンマシンインターフェースのモデル化について、概要を簡単に紹介します。EB GUIDE Studioの起動方法、プロジェクトの作成方法、EB GUIDEモデルの動作および外観のモデル化の方法、EB GUIDEモデルの実行方法を説明しています。

所要時間: 20分

### 5.1. EB GUIDEの起動



#### EB GUIDEの起動

Prerequisite:

- EB GUIDEのインストールが完了していること。

#### Step 1

Windowsのスタートメニューで、すべてのプログラムをクリックします。

#### Step 2

**Elektrobit**メニューで、起動するバージョンをクリックします。

EB GUIDE Studioが起動します。プロジェクトセンターが表示されます。

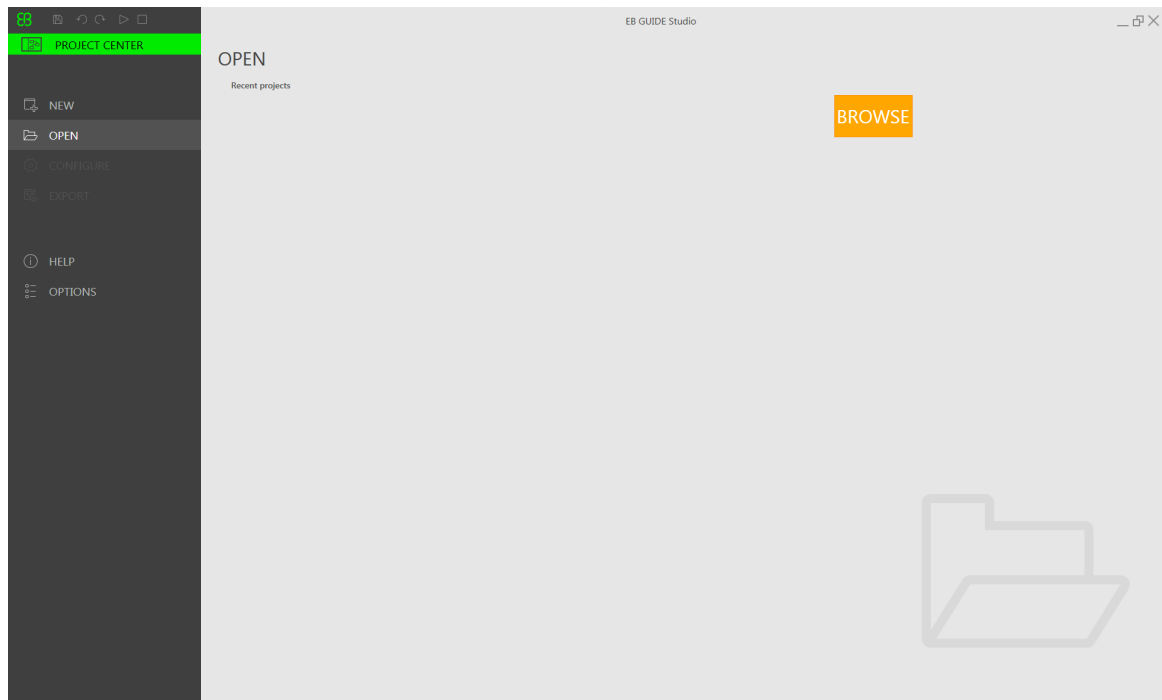


Figure 5.1. プロジェクトセンター

## 5.2. プロジェクトの作成



### プロジェクトの作成

#### Prerequisite:

- EB GUIDE Studioが起動されていること。
- C:/tempのディレクトリが作成されていること。

#### Step 1

プロジェクトセンターのナビゲーションエリアで、新規をクリックします。

#### Step 2

コンテンツエリアで、場所としてC:/tempディレクトリを選択します。

#### Step 3

MyProjectというプロジェクト名を入力します。

#### Step 4

作成をクリックします。

プロジェクトが作成されます。プロジェクトエディターが開き、空プロジェクトが表示されます。

メインステートマシンがデフォルトで追加され、コンテンツエリアに表示されます。

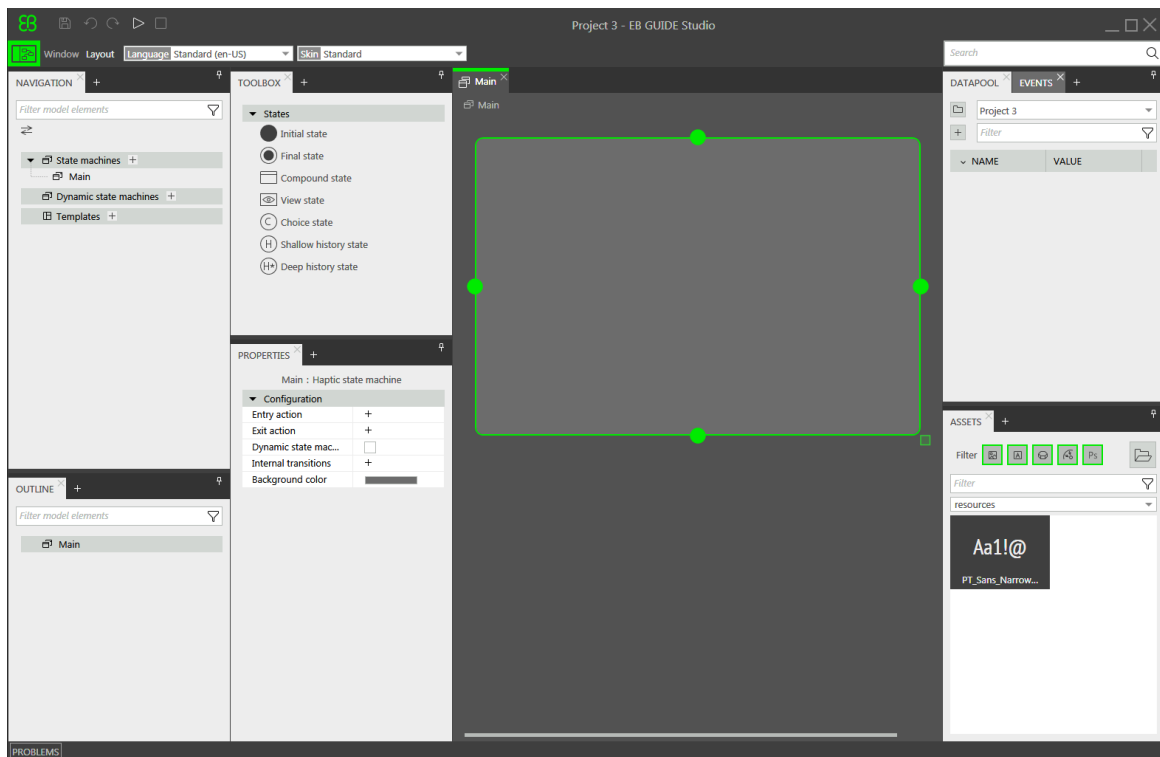


Figure 5.2. メインステートマシンが表示されたプロジェクトエディター

## 5.3. ヒューマンマシンインターフェースの動作のモデル化

EB GUIDEモデルの動作は、ステートマシンによって定義されます。では、UMLに似た構文を使用してこれを行います。

このセクションでは、定義されたビューを起動時に表示し、ボタンを押すと別のビューに変わるステートマシンをモデル化する方法を説明します。



### ステートマシンへのステートの追加

EB GUIDEには、さまざまなステートが用意されています。このセクションでは、3種類のステートを示します。初期ステートは、ステートマシンのスタート地点となります。ビューステートは、デフォルトのビューを表示します。また、ステートマシンの最終ステートは、ステートマシンの終了処理を行います。

Prerequisite:

- プロジェクトMyProjectが作成されていること。

- コンテンツエリアに、メインステートマシンが表示されていること。

#### Step 1

ツールボックスからビューステートをドラッグし、ステートマシンにドロップします。

View state 1と共に、ビューがEB GUIDEモデルに追加されます。

#### Step 2

ステップ1を繰り返します。

View state 2 が追加されます。

#### Step 3

ツールボックスから初期ステートをドラッグし、ステートマシンにドロップします。

#### Step 4

ツールボックスから最終ステートをドラッグし、ステートマシンにドロップします。

メインステートマシンに追加した4つのステートは、コンテンツエリアとナビゲーションコンポーネントの両方に、それぞれステートチャートと階層的ツリービューとして表示されます。

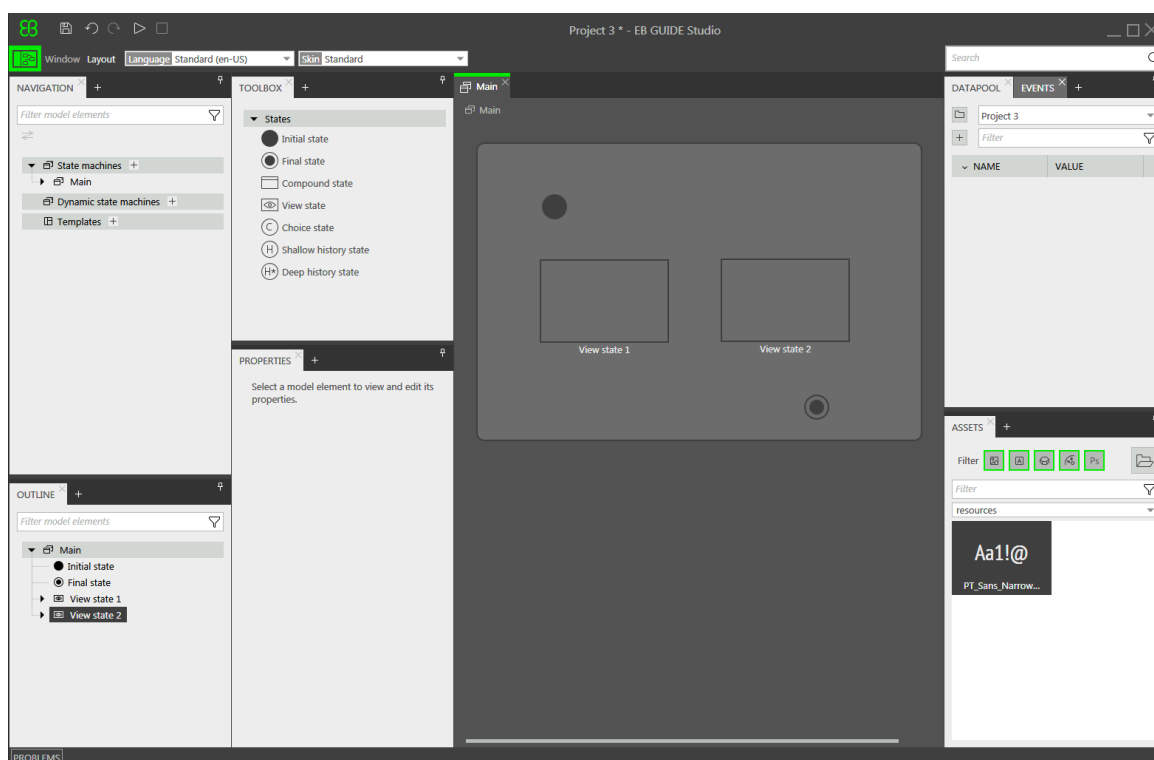


Figure 5.3. ステートが表示されたプロジェクトエディター



### 遷移の追加

遷移は、ステート間の接続であり、ステートの変化をトリガーします。各種の遷移タイプがあります。このセクションでは、デフォルト遷移とイベントトリガー遷移を扱います。

#### Prerequisite:

- コンテンツエリアに、メインステートマシンが表示されていること。
- メインステートマシンに、初期ステート、2つのビューステート、および最終ステートが含まれていること。

#### Step 1

初期ステートを遷移のソースステートとして選択します。

#### Step 2

緑色のドラッグ点をクリックし、マウスボタンを押したままにします。

#### Step 3

ターゲットステートView state 1までマウスをドラッグします。

#### Step 4

ターゲットステートが緑色で強調表示されたら、マウスボタンを離します。

遷移が作成され、緑色の矢印として表示されます。

#### Step 5

View state 1とView state 2の間に遷移を追加します。

View state 1を選択して、ターゲットステートとしてView state 2にステップ2~4を繰り返します。

#### Step 6

View state 1とView state 2の間の遷移を選択します。

次のステップでは、遷移をイベントに関連付けます。

#### Step 7

プロパティコンポーネントに移動し、トリガーコンボボックスにEvent 1と入力して、イベントの追加をクリックします。

Event 1というイベントが作成され、遷移トリガーとして追加されます。Event 1が発行されたときには、必ずこの遷移が実行されます。

#### Step 8

View state 2と最終ステートの間に遷移を追加します。

View state 2を選択して、ターゲットステートとして最終ステートにステップ2~4を繰り返します。

新規イベントEvent 2をトリガーとして追加します。

この時点で、ステートマシンは次の図のようになっています。



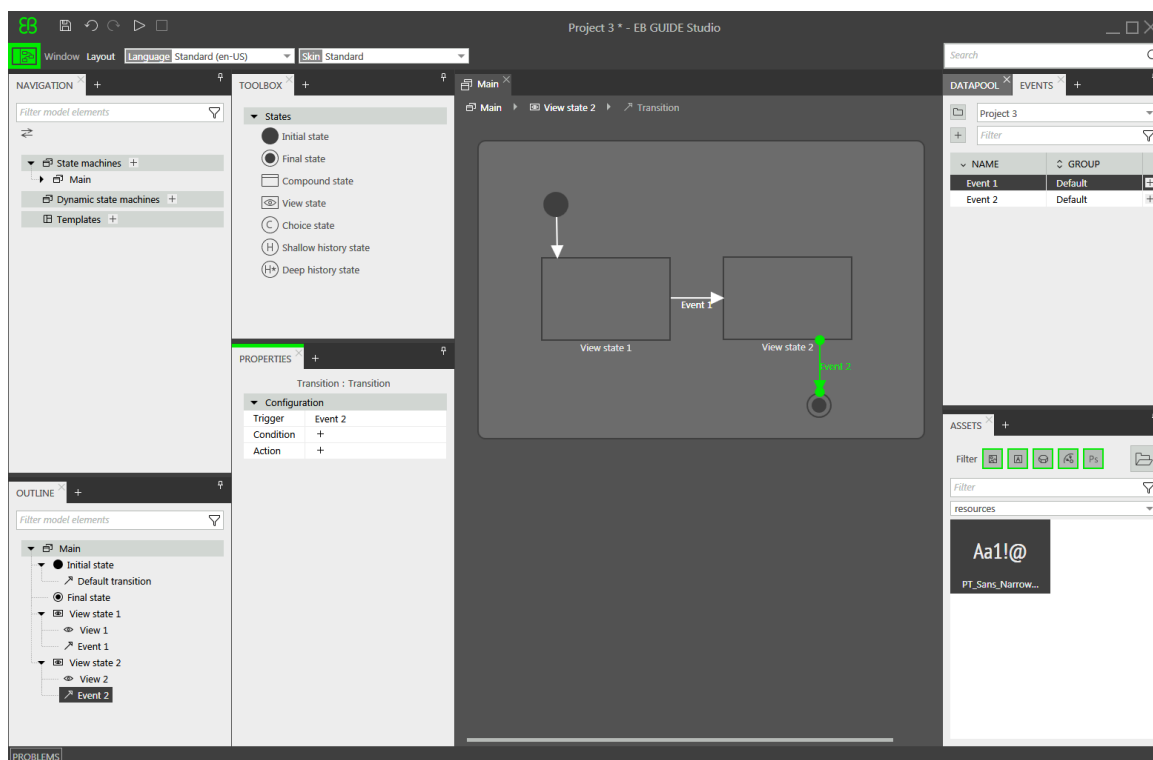


Figure 5.4. イベントを伴う遷移でリンクされたステート

これで、基本的なステートマシンの動作が定義されました。

## 5.4. ヒューマンマシンインターフェースの外観のモデル化

上記のセクションで作成したステートマシンには、2つのビューステートが含まれています。このセクションでは、ビューをモデル化する方法を説明します。



ビューを開く

Prerequisite:

- View state 1 がモデルに追加されていること。

### Step 1

View state 1をダブルクリックします。

コンテンツエリアにView 1が表示されていること。



## ビューへのボタンの追加

EB GUIDE Studioには、ビューの外観をモデル化するさまざまなオプションがあります。

1つの例として、このセクションでは四角形をビューに追加する方法を示します。この四角形は、ユーザーの入力に反応し、ボタンとして機能します。

Prerequisite:

- コンテンツエリアにView 1が表示されていること。

### Step 1

ツールボックスから四角形をドラッグし、ビューにドロップします。

### Step 2

プロパティコンポーネントでウィジェット機能プロパティカテゴリに移動し、追加/削除をクリックします。

ウィジェット機能ダイアログが表示されます。

### Step 3

使用可能なウィジェット機能の下で、入力処理カテゴリを展開し、タッチリリースを選択します。

承認をクリックします。

関連するウィジェット機能のプロパティがプロパティコンポーネントに追加されます。

### Step 4

プロパティコンポーネントで、touchPolicyドロップダウンリストボックスからPress then reactを選択します。

この四角形はモデル実行モードでタッチ入力に反応します。

### Step 5

touchShortReleasedプロパティに移動し、編集をクリックします。

### Step 6

次のEB GUIDEスクリプトを入力します。

```
function(v:touchId::int, v:x::int, v:y::int, v:fingerId::int)
{
  fire_delayed 500, ev:"Event 1"()
  true
}
```

モデル実行モードで四角形にタッチすると、500ミリ秒後にEvent 1が発行されます。

### Step 7

承認をクリックします。

### Step 8

プロパティコンポーネントで、fillColorプロパティとして赤を選択します。

### Step 9

ナビゲーションコンポーネントで、View 2をダブルクリックします。

コンテンツエリアにView 2が表示されていること。

### Step 10

ステップ1~5を繰り返します。

### Step 11

次のEB GUIDEスクリプトを入力します。

```
function(v:touchId::int, v:x::int, v:y::int, v:fingerId::int)
{
    fire_delayed 500, ev:"Event 2"()
    true
}
```

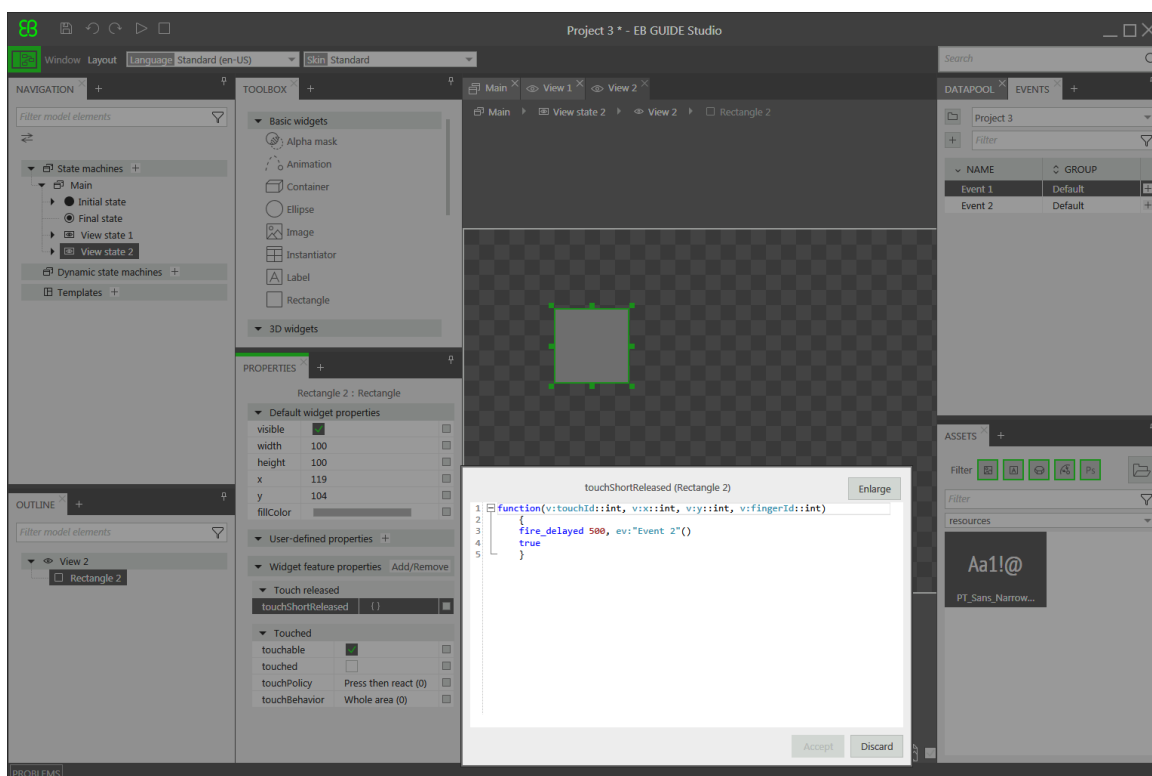


Figure 5.5. EB GUIDEスクリプトを持つウィジェットプロパティ

### Step 12

承認をクリックします。

モデル実行モードで四角形にタッチすると、500ミリ秒後にEvent 2が発行されます。

### Step 13

プロパティコンポーネントで、fillColorプロパティとして青を選択します。


## 5.5. シミュレーションを開始する

EB GUIDEでは、対象デバイスにエクスポートする前にモデルをPC上で実行できます。




### シミュレーションを開始する

#### Step 1


プロジェクトを保存するには、コマンドエリアでをクリックします。

#### Step 2

コマンドエリアでをクリックします。

EB GUIDEモデルが開始され、モデル化した動作と外観が表示されます。

最初に、View 1が表示されます。赤い四角形をクリックすると、画面がView 2に変化します。これは、クリックによってEvent 1が発行され、Event 1によってView state 1からView state 2への遷移が実行されるためです。

次にView 2が表示されます。View 2で青い四角形をクリックすると、ステートマシンの終了処理が行われます。これは、クリックによってEvent 2が発行され、Event 2によってView state 2から最終状態への遷移が実行されるためです。シミュレーションのウィンドウは開いたままです。シミュレーションを停止するには、をクリックします。

## 6. バックグラウンド情報

この章では、トピックスをアルファベット順で並べています。

### NOTE



デフォルトのウィンドウレイアウト

すべての手順およびスクリーンショットでは、デフォルトのウィンドウレイアウトが使用されています。手順に従う場合は、EB GUIDE Studioウィンドウをレイアウト > **Reset to default layout**を選択することによって、デフォルトレイアウトにリセットすることをお勧めします。

### 6.1. 3Dグラフィック

EB GUIDE Studioでは、EB GUIDEプロジェクトに3Dグラフィックを使用できます。

#### 6.1.1. サポートされている3Dグラフィック形式

OpenGL ES 2.0以降のレンダラーに限り、3Dグラフィックを表示できます。サポートされている3Dグラフィック形式は、COLLADA (.dae)およびFilmbox (.fbx)です。失敗を防ぐため、Filmbox形式を使用します。

#### 6.1.2. 3Dグラフィックファイルの設定

3DオブジェクトをEB GUIDE Studioのビューに表示するには、次のオプションで3Dグラフィックファイルを作成する必要があります。

- ▶ 透視図カメラ
- ▶ メッシュおよび少なくとも1つの材質を含む少なくとも1つのオブジェクト
- ▶ 1つ以上の光源

### TIP



シーングラフのガンマ補正

gammaプロパティを使用すると、モニターまたはディスプレイデバイスの輝度応答に合うようにシーングラフの輝度出力を調整して、視覚に関して最適な結果を得ることができます。値は、0.0より大きくなければならず、デフォルトの2.2に設定されています。これは、ほとんどのディスプレイに適した値です。

3Dグラフィックファイルを作成するには、サードパーティの3Dモデリングソフトウェアを使用します。

3Dグラフィックファイルは、以下に示す多種多様な追加コンテンツをサポートしています。

- ▶ 位置、法線、従法線、接線、および1つのテクスチャチャンネルを含む3Dオブジェクト

- ▶ 指向性光源
- ▶ イメージベースドライツソース
- ▶ 環境光源
- ▶ 定数、線形、二次、および三次減衰を含む点光源
- ▶ 円錐角、定数、線形、二次、および三次減衰を含むスポット光源
- ▶ ビュー、ニアプレーン、およびファープレーンのフィールドに対応する透視図カメラのサポート
- ▶ テクスチャ: エミッシブ、ディフューズ、スペキュラ、ノーマルマップ、オパシティ、リフレクションキューブ、およびライツマップ

**TIP**



**3Dグラフィックファイルのセットアップ**

オパシティマップには有効なアルファチャネルが必要であることを注意してください。

### 6.1.3. 3Dグラフィックファイルのインポート

3Dグラフィックをビューに追加するには、シーングラフを使用して3Dグラフィックファイルをインポートする必要があります。インポート中、EB GUIDE Studioは3Dグラフィックファイルをシーングラフを親ノードに持つウィジェットツリーに変換します。例えばカメラ、材質、メッシュなどの3Dグラフィックファイルのコンテンツに対し、EB GUIDE Studioはそれぞれウィジェットを作成します。インポートする3Dグラフィックファイルの3Dシーンにアニメーションが含まれている場合、EB GUIDE Studioはリニアキー値補間曲線を使用してそれらのアニメーションをインポートします。この曲線は、タイプが浮動小数点数、整数、または色であるアニメーション化プロパティに適用されます。リニアキー値補間曲線は、その他のアニメーション曲線と同じように適用することはできません。この曲線は、3Dグラフィックのアニメーションをインポートする場合にのみ使用されます。

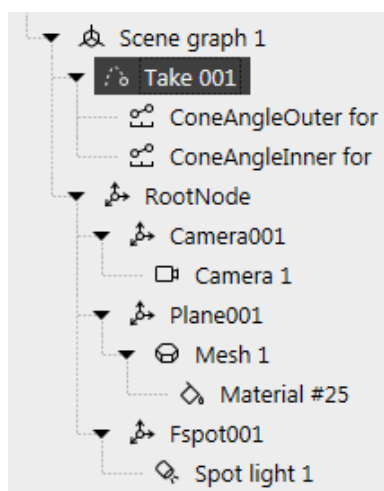


Figure 6.1. ナビゲーションコンポーネントに表示されるシーングラフの例

## NOTE



### 制限

次のことに注意してください。

- ▶ EB GUIDE Studioでは、メッシュごとに1つの材質のみ指定できます。3Dグラフィックでメッシュ1つにつき材質が複数指定されている場合、EB GUIDE Studioは材質ごとにメッシュを追加で作成します。
- ▶ .fbxファイルのインポート時にはデフォルトの材質ウィジェットのみが作成されます。3Dモデルにその他のタイプの材質がある場合は、EB GUIDE Studioによってデフォルトの材質のみが追加され、そのプロパティはデフォルト値に設定されます。EB GUIDE Studioでは、PBR Phong材質およびPBR GGX材質ウィジェットを使用して、その他のタイプの材質を追加できます。
- ▶ EB GUIDE Studioでメッシュに複数の材質が追加されている場合は、最上位の材質ウィジェットのみがレンダリングされます。

3Dグラフィックファイルをインポートすると、ディレクトリ\$GUIDE\_PROJECT\_PATH/<project name>/resourcesにサブディレクトリが作成されます。サブディレクトリには、インポートされた.fbxファイルの名前が付けられます。サブディレクトリの名前に作成した日時を追加することもできます。



### Example 6.1.

#### インポートディレクトリの命名

3Dグラフィックファイルは、car.fbxと呼ばれます。EB GUIDE Studioに3Dグラフィックファイルをインポートすると、\$GUIDE\_PROJECT\_PATH/<project name>/resourcesにcar\_20160102\_103029という名前のサブディレクトリが作成されます。

サブディレクトリには、以下の項目が収められます。

- ▶ .ebmeshファイルとなったメッシュ
- ▶ .pngまたは.jpgファイルとなったテクスチャ

3Dグラフィックに追加でテクスチャを使用するには、\$GUIDE\_PROJECT\_PATH/<project name>/resourcesにテクスチャをコピーします。テクスチャには.pngまたは.jpgイメージを使用してください。

複数の3Dグラフィックを1つのシーングラフにインポートすることもできます。

3Dウィジェットはインポート後に追加、変更、削除することができます。

詳細については、[6.22「ウィジェット」](#)、[13.12.4「3Dウィジェット」](#)、および[13.13.8「3D」](#)をご覧ください。

手順については、[8.1.3.1「ビューへのシーングラフの追加」](#)および[12.7「チュートリアル: 3Dグラフィックの操作」](#)をご覧ください。

## 6.2. アニメーション

アニメーションを使用すると、EB GUIDEモデルに動きと視覚効果を付けることができます。EB GUIDEでは、さまざまな使用例でアニメーションを使用できます。ビュー内のウィジェットをアニメーション化することや、あるビューから別のビューへの遷移をアニメーション化することができます。

### 6.2.1. ウィジェットのアニメーション

ウィジェットのアニメーション化とは、ウィジェットをビュー上で移動させることです。この移動は曲線で定義されます。このため、ツールボックスの基本ウィジェットカテゴリには、アニメーションと呼ばれるウィジェットが含まれています。どのアニメーションに対しても、コンスタント曲線、リニア補間曲線、正弦曲線など、一連の曲線を追加できます。曲線は、targetウィジェットプロパティを持ち、そのtargetプロパティの時間的な変化を記述したものです。

各アニメーションには、1つ以上の曲線が関連付けられています。

とりわけ、ウィジェットにアニメーション化すると次の操作が行えます。

- ▶ ビュー内でウィジェットを移動する
- ▶ ウィジェットのサイズを変更する
- ▶ ウィジェットの色を段階的に変更する

アニメーションは、EB GUIDEスクリプト `f:animation_play`、`f:animation_pause`、`f:animation_cancel`などの関数によって制御されます。

#### TIP



#### 並列アニメーション

EB GUIDEでは、アニメーションが並列アニメーションになっており、複数の曲線が並行して実行されます。これは、複数のアニメーションの曲線が同じウィジェットプロパティを対象として使用している場合、それらの曲線は同じタイミングでそのtargetプロパティの値を上書きすることを意味します。

アニメーションと曲線のプロパティについては、[13.12.2.2「アニメーション」](#)をご覧ください。

手順については、[8.1.2.7「アニメーションの追加」](#)をご覧ください。

独自のカスタム曲線を作成する場合は、[13.12.2.6「スクリプト曲線」](#)をご覧ください。詳しいバックグラウンド情報については、[6.2.3「スクリプト曲線」](#)をご覧ください。

### 6.2.2. ビュー遷移のアニメーション



ビュー遷移のアニメーション化とは、ビューステートの開始時または終了時にムーブまたはフェードするアニメーションを定義することです。ビューを変更すると、それらのアニメーションがトリガーされます。

ビューステートおよびビューテンプレートのView Transition Animationを定義します。ビューテンプレートを再利用するたびに、インスタンスはテンプレートのView Transition Animationを継承します。

View Transition Animationにはさまざまなタイプがあります。

Table 6.1. アニメーションのタイプ

アニメーションのタイプ	説明
開始アニメーション	このアニメーションは、アニメーション付きのビューステートにエンタリーすると再生されます。このアニメーションで操作できるのは、追加されたビューのウィジェットプロパティとウィジェット機能プロパティのみです。
終了アニメーション	このアニメーションは、アニメーション付きのビューステートが終了すると再生されます。このアニメーションで操作できるのは、追加されたビューのウィジェットプロパティとウィジェット機能プロパティのみです。
変更アニメーション	このアニメーションはビューステートが変更されると再生されます。このアニメーションでは遷移元ビューと遷移先ビューのプロパティを操作できます。遷移元ビューは、アニメーションが追加されるビューステートです。遷移先ビューは、別のビューステートまたはビューテンプレートにすることができます。
ポップアップオンアニメーション	ビューテンプレートビューと動的ステートマシンビューにのみ使用可能です。このアニメーションは、それぞれの動的ステートマシンが有効化されると(動的ステートマシンがプッシュされると)再生されます。このアニメーションでは、追加されたビューのプロパティとウィジェットプロパティを操作できます。
ポップアップオフアニメーション	ビューテンプレートビューと動的ステートマシンビューにのみ使用可能です。このアニメーションは、それぞれの動的ステートマシンを終了すると(動的ステートマシンがポップされると)再生されます。このアニメーションでは、追加されたビューのプロパティとウィジェットプロパティを操作できます。

ビューテンプレートのアニメーションプロパティについては、[13.12.1「ビュー」](#)をご覧ください。

手順については、[8.6「ビュー遷移のアニメーション化」](#)をご覧ください。

### 6.2.3. スクリプト曲線

デフォルトのアニメーション曲線には、豊富なカスタマイズオプションがすでに用意されています。また、EB GUIDE Studioではスクリプト曲線機能によって独自のアニメーション曲線を定義することもできます。この機能により、EB GUIDEスクリプトを使用した独自の曲線を定義できます。

スクリプト曲線アニメーションについての理解を深めるには、EB GUIDE Monitorで`v:diff`および`v:t_anim`の出力を取得します。手順については、[8.1.2.8「スクリプト曲線によるアニメーションの追加」](#)をご覧ください。チュートリアルについては、[12.10「チュートリアル: アニメーションでのスクリプト曲線の使用」](#)をご覧ください。

## 6.3. アプリケーションとモデルを結ぶアプリケーションプログラミングインターフェース

EB GUIDEでは、アプリケーションとEB GUIDE TFとの間のすべての通信データがアプリケーションプログラミングインターフェース(API)で抽象化されます。アプリケーションとは、例えばメディアプレーヤーであったりナビゲーションであったりします。

アプリケーションプログラミングインターフェースは、データプールアイテムとイベントで定義します。イベントはヒューマンマシンインターフェースとアプリケーションの間を送信されます。



### Example 6.2.

アプリケーションプログラミングインターフェースの内容

- ▶ `START_TRACK`: アプリケーションに送信されるイベント。パラメータ`track`には、再生するトラックの数が設定されます。
- ▶ `TRACK_STOPPED`: 再生されたトラックが終了したときに、アプリケーションからヒューマンマシンインターフェースに送信されるイベント。
- ▶ `MEDIA_CURRENT_TRACK`: アプリケーションから書き込まれる動的なデータプールアイテム。
- ▶ `MEDIA_PLAY_SPEED`: 再生速度を定義する動的なデータプールアイテム。この値は、ヒューマンマシンインターフェースを通じてユーザーが設定します。

## 6.4. 通信コンテキスト

通信コンテキストは、通信が行われる環境を記述します。例えばメディアやナビゲーションアプリケーションなどは通信コンテキストです。これらはヒューマンマシンインターフェイスモデルを使って通信を行います。ある通信コンテキストによる変更は、ライターアプリケーションで発行され、リーダーアプリケーションで更新されるまで、他の通信コンテキストには見えません。

通信コンテキストには、識別のため、プロジェクトの設定で固有の名前と数値ID(0~255)が割り振られます。

手順については、[9.9「外部通信の確立」](#)をご覧ください。

## 6.5. グラフィカルユーザーインターフェイスのコンポーネント

EB GUIDE Studioのグラフィカルユーザーインターフェイスは、プロジェクトセンターとプロジェクトエディターという2つのコンポーネントに分割されています。プロジェクトセンターでは、EB GUIDEプロジェクトの管理、オプションの設定、対象デバイスにコピーするためのEB GUIDEモデルのエクスポートを行います。プロジェクトエディターでは、ヒューマンマシンインターフェイスの外観と動作をモデリングします。

### 6.5.1. プロジェクトセンター

プロジェクトセンターは、EB GUIDE Studioの起動後に表示される最初の画面です。プロジェクト関連のすべての機能が、プロジェクトセンターにあります。プロジェクトセンターは、ナビゲーションエリアとコンテンツエリアという2つの要素で構成されています。

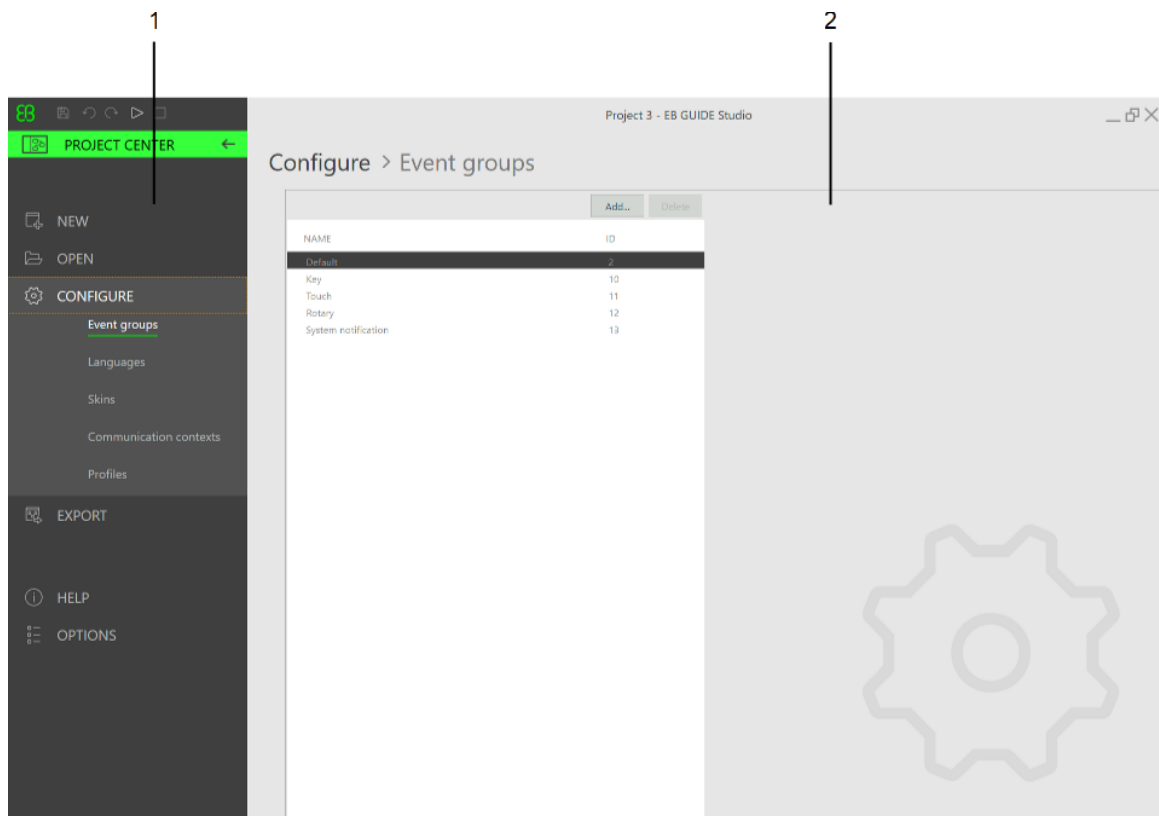


Figure 6.2. ナビゲーションエリア(1)とコンテンツエリア(2)で構成されるプロジェクトセンター

### 6.5.1.1. ナビゲーションエリア

プロジェクトセンターのナビゲーションエリアは、設定やエクスポートといった機能タブで構成されています。ナビゲーションエリア内のタブをクリックすると、対応する機能と設定がコンテンツエリアに表示されます。

### 6.5.1.2. コンテンツエリア

プロジェクトセンターのコンテンツエリアでは、プロジェクト管理と設定を行います。例えば、プロジェクトを保存するディレクトリを選択したり、EB GUIDEモデルの起動時の動作を定義したりします。コンテンツエリアの外観は、ナビゲーションエリアで選択されているタブによって異なります。

## 6.5.2. プロジェクトエディター

プロジェクトを作成すると、プロジェクトエディターが表示されます。プロジェクトエディターでは、ヒューマンマシンインターフェースの動作と外観のモデリングを行います。ステートマシンをモデリングし、ビューを作成し、イベントとデータプールを管理します。プロジェクトエディターは、以下のエリアおよびコンポーネントで構成されています。プロジェクトのすべてのコンポーネントはドッキングまたはフローティングにすることができ、コンテンツエリアを除くプロジェクトエディターの任意の位置に配置できます。

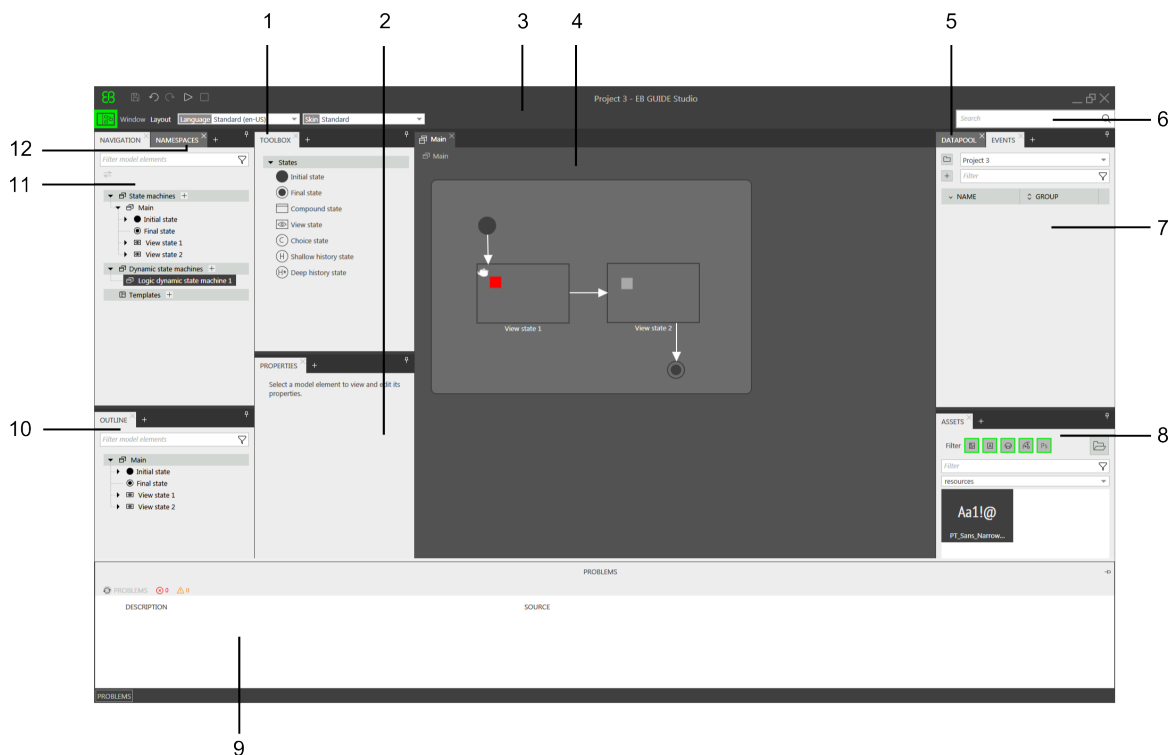


Figure 6.3. プロジェクトエディターとそのエリアおよびコンポーネント

- 1 ツールボックスコンポーネント
- 2 プロパティコンポーネント
- 3 コマンドエリア
- 4 コンテンツエリア
- 5 データプールコンポーネント
- 6 検索ボックス
- 7 イベントコンポーネント
- 8 アセットコンポーネント
- 9 問題検出コンポーネント
- 10 概要コンポーネント
- 11 ナビゲーションコンポーネント
- 12 名前空間コンポーネント

#### 6.5.2.1. ナビゲーションコンポーネント

ナビゲーションコンポーネントには、EB GUIDEモデルのステート、ビュー、アニメーション、遷移などのモデル要素が階層構造として表示され、任意の要素に移動できます。モデル要素をダブルクリックすると、そのモデル要素がコンテンツエリアに表示されます。

ナビゲーションコンポーネントには、EB GUIDEモデルのすべてのグラフィカル要素と非グラフィカル要素の概要が表示され、ステートマシン階層が反映されます。

EB GUIDEモデルに要素(ステートマシン、動的ステートマシン、テンプレートなど)を追加する場所でもあります。ウィジェットやアニメーションなど、ツールボックスにある要素をドラッグアンドドロップ操作によって追加できます。

**NOTE**



**フィルタボックス**

コンポーネントの最上部には、コンポーネント内の要素を検索するためのフィルタボックスがあります。

コンポーネント内の要素をクリックして**F3**キーを押すと、参照検索が始まります。検索結果 ウィンドウが開き、選択した要素のEB GUIDEモデル内での出現箇所がすべてリストされます。

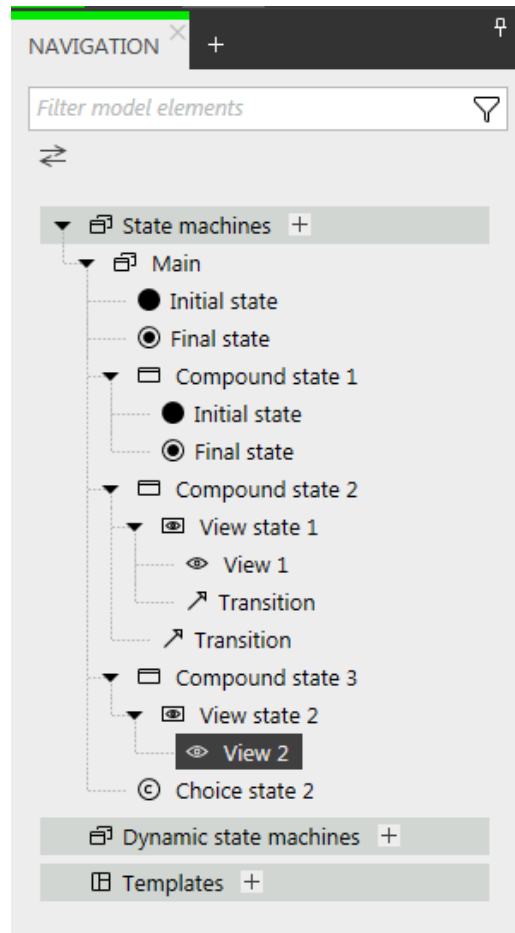


Figure 6.4. プロジェクトエディターのナビゲーションコンポーネント

### 6.5.2.2. 概要コンポーネント

ナビゲーションコンポーネント、または現在コンテンツエリアに表示されているエディターコンポーネントで選択されたツリー部分に含まれている構造およびモデル要素のみが表示されます。

**NOTE**



**フィルタボックス**

コンポーネントの最上部には、コンポーネント内の要素を検索するためのフィルタボックスがあります。

コンポーネント内の要素をクリックして**F3**キーを押すと、参照検索が始まります。検索結果 ウィンドウが開き、選択した要素のEB GUIDEモデル内での出現箇所がすべてリストされます。

### 6.5.2.3. ツールボックスコンポーネント

モデリングに必要なツールはすべて、ツールボックスコンポーネント(ツールボックスとも呼ばれます)にあります。コンテンツエリアに表示されている要素に応じて、ツールボックスでは異なる一連のツールが提供され、それらはコンテンツエリアまたはナビゲーションコンポーネントにドラッグできます。例えば、ツールボックスには次のものが含まれます。

- ▶ コンテンツエリアにステートマシンが表示されている場合、ツールボックスには、ステートマシンに追加できるステートが含まれます。
- ▶ コンテンツエリアにビューが表示されている場合、ツールボックスには、ビューに整列できるウィジェットが含まれます。
- ▶ コンテンツエリアにスクリプト値プロパティが表示されている場合、ツールボックスには、挿入可能なEB GUIDE スクリプト関数が含まれます。

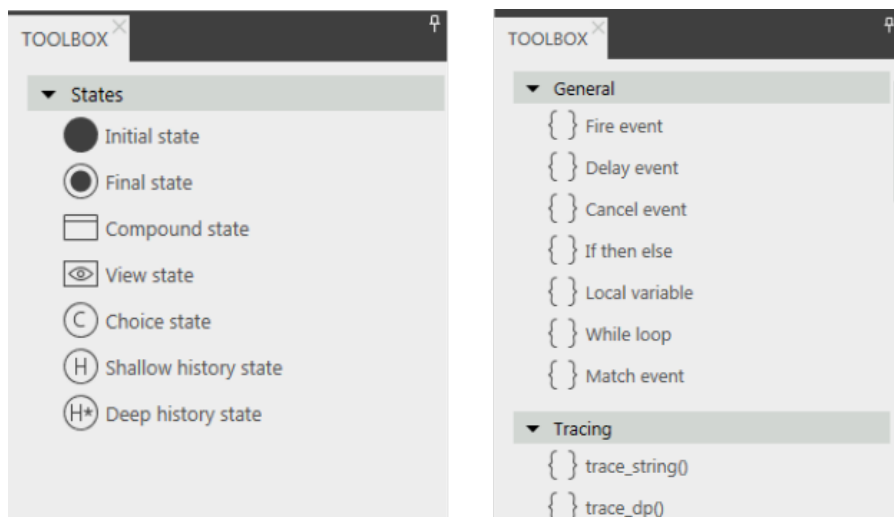


Figure 6.5. プロジェクトエディターのツールボックス

### 6.5.2.4. プロパティコンポーネント

プロパティコンポーネントには、ウィジェットやステートなど、選択されたモデル要素のプロパティが表示されます。プロパティはカテゴリでグループ化され、プロパティコンポーネントで編集できます。

プロパティをクリックして**F3**キーを押すと、参照検索が開始されます。検索結果ウィンドウが開き、選択したプロパティのEB GUIDEモデル内での出現箇所がすべてリストされます。

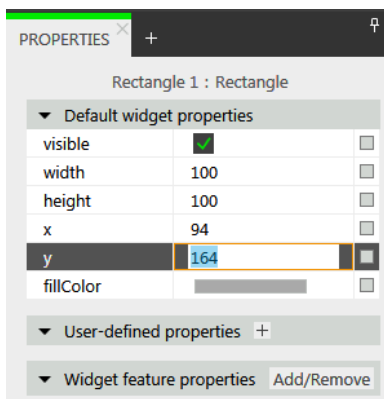


Figure 6.6. ウィジェットのプロパティが表示されたプロパティコンポーネント

#### 6.5.2.5. コンテンツエリア

コンテンツエリアの表示内容は、ナビゲーションコンポーネントでの選択内容によって異なります。モデル要素を編集する場合、ナビゲーションコンポーネントでそのモデル要素をダブルクリックするとコンテンツエリアにそれが表示されます。例えば、状態マシンの状態をモデリングする場合は、ビュー内にウィジェットを整列させるか、コンテンツエリアでEB GUIDEスクリプトを編集します。



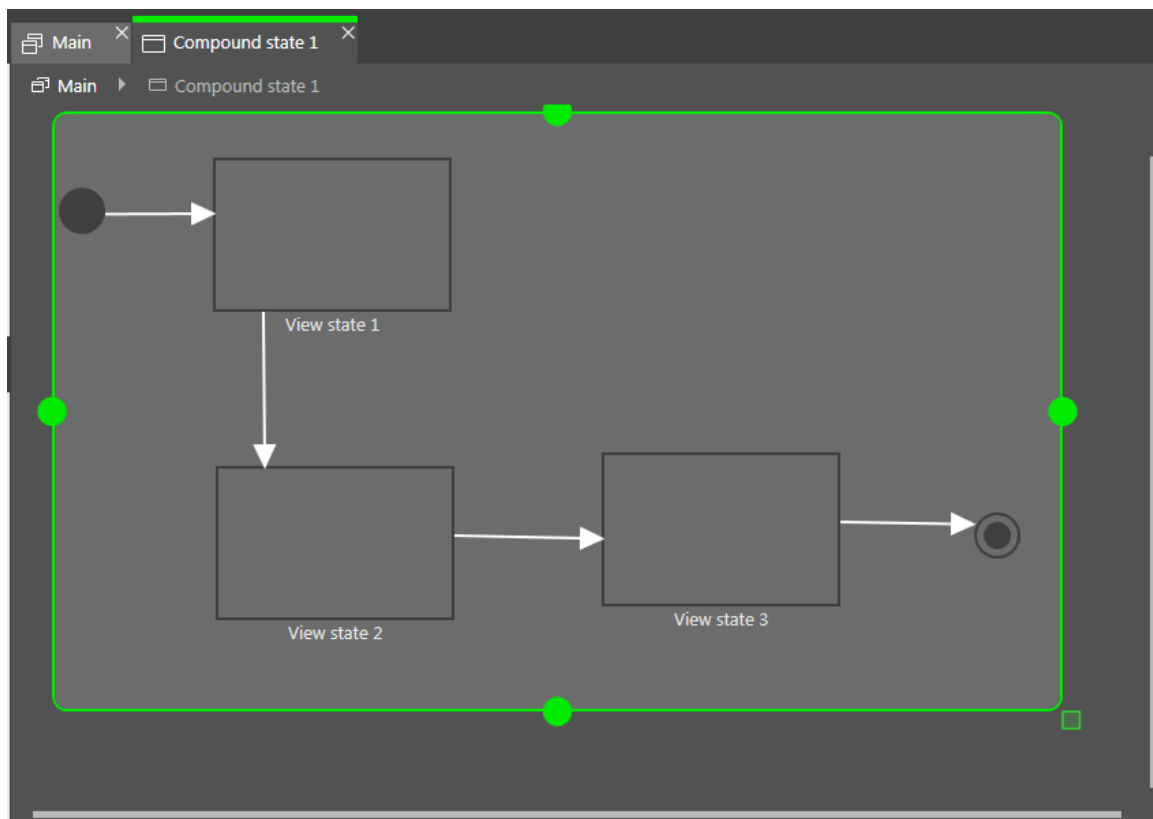


Figure 6.7. プロジェクトエディターのコンテンツエリア

コンテンツエリアに開いているビューがあり、そのビューにアニメーションが含まれている場合は、アニメーションエディターが開かれます。アニメーションエディターでは、曲線をウィジェットプロパティに追加できます。また、プレビュー内のハンドルを移動することで、曲線のdelayおよびdurationプロパティを編集することもできます。

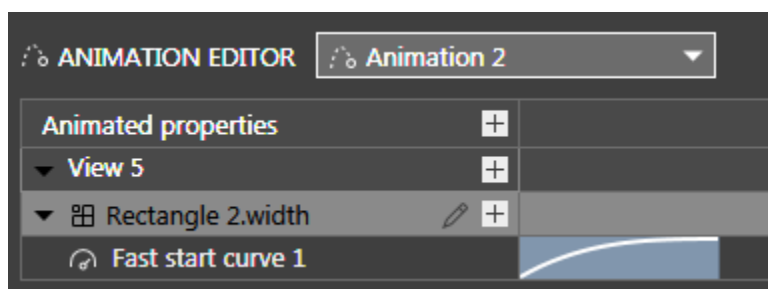


Figure 6.8. アニメーションエディター

リファレンス検索を開始するには、コンテンツエリア内の状態またはウィジェットをクリックして**F3**キーを押します。検索結果ウィンドウが開き、選択した状態またはウィジェットのEB GUIDEモデル内でのすべての出現箇所のリストが表示されます。

### 6.5.2.6. イベントコンポーネント

ここでは、選択されている名前空間にイベントを追加したり、プロパティ(イベント表の名前、グループ、タイプ、パラメータ名など)を編集できます。

#### NOTE



#### フィルタボックス

コンポーネントの最上部には、コンポーネント内の要素を検索するためのフィルタボックスがあります。

コンポーネント内の要素をクリックして**F3**キーを押すと、参照検索が始まります。検索結果 ウィンドウが開き、選択した要素のEB GUIDEモデル内での出現箇所がすべてリストされます。

#### TIP



#### モデル要素の複数選択

複数の要素を選択するには、**Ctrl**キーを押しながら 選択したい要素をクリックします。また、**Shift**キーを押しながら それぞれの要素をクリックしたり、上矢印キーまたは 下矢印キーを使用したりすることもできます。

イベントコンポーネントでは、以下のビューオプションの切り替えができます。

- ▶ 選択されている名前空間のイベントのみを表示する。
- ▶ 選択されている名前空間のイベントとそのサブ名前空間を表示する。
- ▶ すべての名前空間のイベントを表示する。

### 6.5.2.7. データプールコンポーネント

ここでは、選択されている名前空間にデータプールアイテムを追加したり、プロパティ(名前、値など)を編集したりできます。データプールアイテムへのリンクの追加、値からスクリプトへの変換、および言語サポートやスキンのサポートの追加を行うこともできます。

#### NOTE



#### フィルタボックス

コンポーネントの最上部には、コンポーネント内の要素を検索するためのフィルタボックスがあります。

コンポーネント内の要素をクリックして**F3**キーを押すと、参照検索が始まります。検索結果 ウィンドウが開き、選択した要素のEB GUIDEモデル内での出現箇所がすべてリストされます。

#### TIP



#### モデル要素の複数選択

複数の要素を選択するには、**Ctrl**キーを押しながら 選択したい要素をクリックします。また、**Shift**キーを押しながら それぞれの要素をクリックしたり、上矢印キーまたは 下矢印キーを使用したりすることもできます。

データプールコンポーネントでは、以下のビューオプションの切り替えができます。

- ▶ 選択されている名前空間のデータプールアイテムのみを表示する。
- ▶ 選択されている名前空間のデータプールアイテムとそのサブ名前空間を表示する。
- ▶ すべての名前空間のデータプールアイテムを表示する。

### 6.5.2.8. アセットコンポーネント

ここでは、イメージ、フォント、.ebmeshファイル、.psdファイルなどのリソースを追加できます。`$GUIDE_PROJECT_PATH/<project name>/resources`ディレクトリとそのサブディレクトリにあるすべてのリソースファイルがコンポーネントのプレビューエリアに表示されます。ドラッグアンドドロップ操作によってモデルにリソースを追加できます。

#### NOTE



#### フィルタボックス


コンポーネントの最上部には、コンポーネント内の要素を検索するためのフィルタボックスがあります。

### 6.5.2.9. 名前空間コンポーネント

ここでは、名前空間を作成、移動、および削除できます。デフォルトでは、ルート名前空間が追加されています。

### 6.5.2.10. コマンドエリア

コマンドエリアには、次のものがあります。

- ▶ プロジェクトセンターを開くための  ボタン
- ▶ モデルの要素を検索してそれらにジャンプするための検索ボックス
- ▶ 詳細なメニュー

#### 検索ボックス

検索ボックスを使うと、モデル要素を検索することができます。検索ボックスは次のように使用します。

- ▶ 検索ボックスをクリックするか、**Ctrl** + **F**キーのショートカットを使用して検索ボックスにジャンプします。検索するモデル要素の名前を入力します。

または、モデル要素を選択して**F3**キーを押すこともできます。検索結果ウィンドウが開き、検索結果が表示されます。

- ▶ ヒットリスト内のモデル要素をダブルクリックし、ジャンプします。

検索結果ウィンドウの左側には、見つかったモデル要素がカテゴリ別に分類された状態でリストされます。上部のフィルタボタンを使用して、カテゴリの表示/非表示を切り替えます。モデル要素を選択して、プレビューを表示するか、読み取り専用モードでモデル要素のプロパティを表示します。

検索結果ウィンドウを閉じると最後に使用した検索語、フィルタ設定、対応するヒットリストが保存され、次に検索結果ウィンドウを開いたときに表示されます。次に開くまでの間にモデル要素が変更された場合、再度検索を実行する必要があります。

検索では大文字と小文字は区別されません。

アスタリスク\*を使用してワイルドカード検索を行う場合、次のルールが適用されます。

- ▶ tと入力して検索すると、tが含まれる要素の名前が返されます。
- ▶ \*tと入力して検索すると、tで終わる要素の名前が返されます。
- ▶ t\*と入力して検索すると、tで始まる要素の名前が返されます。

検索できるのは以下のモデル要素カテゴリです。

Table 6.2. 検索ボックス内のカテゴリ

カテゴリ	説明
ステート	ヒットリストには、見つかったステートの子要素も表示されます。
ビュー	ヒットリストには、見つかったビューの子要素も表示されます。
テンプレート	ヒットリストには、見つかったテンプレートの子要素も表示されません。
イベント	プレビューには、イベントのプロパティが表示されます。
データプールアイテム	プレビューには、データプールアイテムのプロパティが表示されます。
スクリプト	プレビューには、テキストを含むスクリプトのコンテンツが表示されます。見つかったテキストは強調表示されます。
プロパティ	プレビューには、プロパティが属するウィジェットが表示されます。
表示遷移アニメーション	プレビューにより、表示遷移アニメーションが属しているビューが表示されます。

### 6.5.2.11. 問題検出コンポーネント

問題検出コンポーネントでは、モデルが有効であるかどうかをチェックできます。現在開いているEB GUIDEモデルの考えられるエラーと警告が表示されます。問題が発生する部分に直接ジャンプするには、説明をダブルクリックします。

### 6.5.2.12. VTAコンポーネント

VTA (View Transition Animations)コンポーネントでは、ビューステートまたはビューテンプレートのビュー遷移アニメーションを編集できます。さまざまなアニメーションタイプを選択できます。条件がtrueの各アニメーションタイプの最初のタイプなど、関連するすべてのアニメーションタイプが同時に開始されます。

VTAコンポーネントはデフォルトレイアウトには表示されません。VTAコンポーネントを開くには、レイアウトメニューのVTA (View Transition Animations)を選択します。

### 6.5.3. ドッキング可能なコンポーネント

プロジェクトのすべてのコンポーネントは、タブとしてドッキングしたり、ドッキング解除によってフローティングコンポーネントにしたりすることができます。コンポーネントはフローティングコンポーネントとして、コンテンツエリアを除くプロジェクトセンターの任意の部分にドラッグできます。

ドッキングコントロールの矢印はドッキングの位置を選択するときに役立ち、ライブプレビューにはレイアウトがどのようになるかが表示されます。

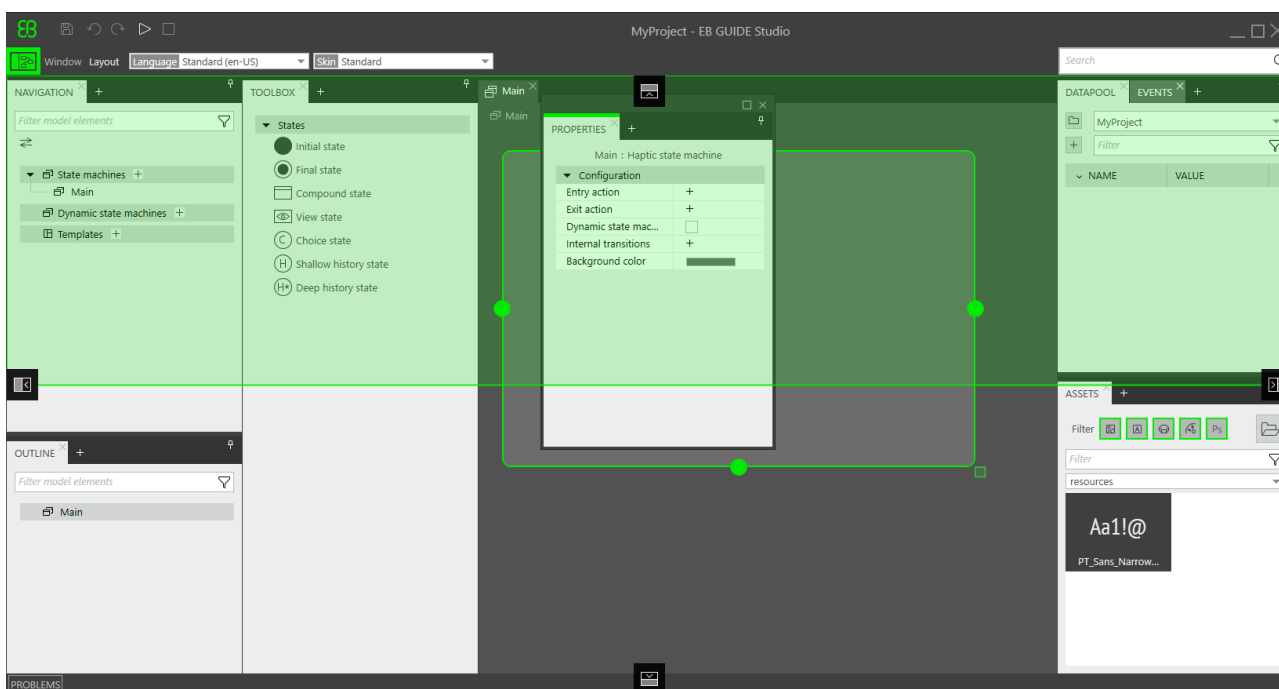


Figure 6.9. ドッキングコントロールとライブプレビュー

**NOTE**



**デフォルトレイアウト**

デフォルトレイアウトに復帰するには、コマンドエリアに移動して、レイアウト > **Reset to default layout**を選択します。

**NOTE**



**自動非表示**

プロジェクトエディターで使用可能な領域を広げるには、コンポーネントを非表示にすることができます。

- ▶ コンポーネントまたはコンポーネントグループを非表示にするには、ピン記号をクリックします。
- ▶ 非表示のコンポーネントを表示するには、マウスをタブの上に合わせて、ピン記号を再びクリックします。

## 6.5.4. EB GUIDE Monitor

EB GUIDEは、モデル実行時にEB GUIDEモデルを監視および制御するためのEB GUIDE Monitorツールを提供しています。EB GUIDE Monitorには、EB GUIDEモデルのデータプール、イベントシステム、およびステートマシンと通信するためのメカニズムが含まれています。

EB GUIDE Monitorは、EB GUIDEモデルのモデル実行時にEB GUIDE Studioで自動的に起動されます。EB GUIDE Monitorは、スタンドアロンアプリケーションとして使用することもできます。

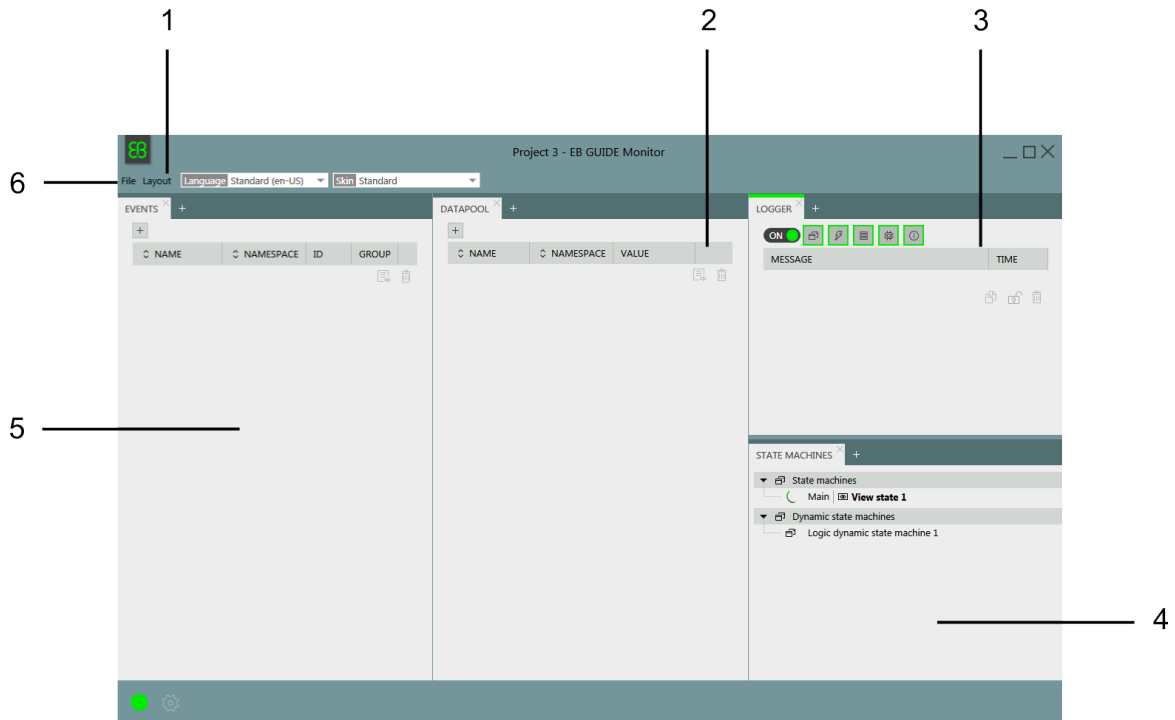


Figure 6.10. EB GUIDE Monitorのデフォルトレイアウト

- 1 レイアウトメニュー
- 2 データプールコンポーネント
- 3 ロガーコンポーネント
- 4 ステートマシンコンポーネント
- 5 イベントコンポーネント
- 6 ファイルメニュー

EB GUIDE Monitorには、次のコンポーネントが含まれています。

- ▶ イベントコンポーネントでは、イベントを検索および発行できます。イベントにパラメータがある場合は、そのパラメータを変更したうえでこのイベントを発行することができます。
- ▶ データプールコンポーネントでは、データプールアイテムを検索して、それらの値を変更できます。
- ▶ ロガーコンポーネントでは、すべての変更、情報メッセージ、エラー、および警告が追跡されます。コンポーネントの最上部には、コンポーネント内のエントリーにフィルタをかけるためのフィルタボタンがあります。自動スクロール機能を有効または無効にするには、コンポーネントの下部にある自動スクロールのチェックボックスをオンまたはオフにします。

**NOTE**



**ログメッセージの数の編集**

\$INSTALL\_PATH\tools\monitor\Monitor.exe.config設定ファイルでは、ロガーがメモリ上限に達した場合のEB GUIDE Monitorの動作を定義できます。そのためには、以下のパラメータを変更します。

- ▶ limit は、ロガーコンポーネントに表示されるエントリーの数を定義します。
- ▶ removeCount は、上限に達した場合に削除されるエントリーの数を定義します。

- ▶ ステートマシンコンポーネントには、現在アクティブなステートとステートマシンが表示されます。
- ▶ スクリプトコンポーネントでは、スクリプトを開始したり、出力スクリプトメッセージを確認したりできます。スクリプトコンポーネントは、デフォルトレイアウトには表示されません。このコンポーネントを追加するには、レイアウト > スクリプトをクリックします。

コンポーネントを再配置したり、プロジェクトのニーズに応じて新しいコンポーネントを追加したりできます。EB GUIDE Monitorウィンドウ内でコンポーネントをドッキングおよびドッキング解除することもできます。

**NOTE**



**デフォルトのウィンドウレイアウト**

すべての手順およびスクリーンショットでは、デフォルトのウィンドウレイアウトが使用されています。手順に従う場合は、EB GUIDE Monitorウィンドウをレイアウト > **Reset to default layout**を選択することによって、デフォルトレイアウトにリセットすることをお勧めします。

**TIP**



**EB GUIDE Monitorウィンドウのサイズのリセット**

EB GUIDE Monitorウィンドウのサイズと画面上での位置は、C:\<user>\AppData\Local\Temp\eb\_guide\_simulation\_export\<project>に個々のEB GUIDEプロジェクトごとに保存されています。

デフォルト値のサイズと位置をリセットするには、このフォルダーにあるmonitor\_layout.xmlおよびmonitor\_settings.xmlを削除します。次のシミュレーションの際に、新しいmonitor\_layout.xmlおよびmonitor\_settings.xmlファイルがデフォルトのサイズと位置の値を使用して作成されます。

EB GUIDE Monitorウィンドウの左下隅には、接続ステータスを示す次のボタンがあります。

ボタン	ステータス
	EB GUIDE Monitorが接続されます。 ボタンをクリックすると、EB GUIDE Monitorが切断されます。
	EB GUIDE Monitorは切断されています。 ボタンをクリックすると、EB GUIDE Monitorが接続されます。
	EB GUIDE Monitorは切断されています。 ボタンをクリックすると、EB GUIDE Monitorの接続設定を設定できます。



コマンドエリアのドロップダウンボックスを使用すると、言語およびスキンを変更することもできます。

手順については、[10.9「EB GUIDE Monitorを操作する」](#)をご覧ください。

EB GUIDE Monitor APIについては、`$GUIDE_INSTALL_PATH/doc/monitor/monitor_api.chm`をご覧ください。

## 6.6. データプール

### 6.6.1. 概念

モデルは実行中にさまざまなアプリケーションと通信します。この通信を可能にするため、EB GUIDEモデルはインターフェイスを提供しなければなりません。データプールは、データプールアイテムにアクセスしてデータを交換できるようにするインターフェイスです。データプールアイテムは、値を格納し、ヒューマンマシンインターフェイスとアプリケーションとの間で通信を成立させます。データプールアイテムは、EB GUIDEモデルで定義されます。

### 6.6.2. データプールアイテム

データプールアイテムは、以下の目的に使用します。

- ▶ データをアプリケーションからヒューマンマシンインターフェイスへ送信します
- ▶ データをヒューマンマシンインターフェイスからアプリケーションへ送信します
- ▶ ヒューマンマシンインターフェイスまたはアプリケーションだけで使用されるデータを格納します

手順については、[9.6「データプールアイテムの追加」](#)をご覧ください。

通信チャンネルを開くには、ライターおよびリーダアプリケーションを使用します。

内部通信は、データを格納するために使用されます。2つのアプリケーションを使うと、外部通信が確立されます。

手順については、[9.9「外部通信の確立」](#)をご覧ください。

### 6.6.3. ウィンドウ表示リスト

EB GUIDE product lineは、ウィンドウ表示リストの概念をサポートします。多くの場合、ウィンドウ表示リストの操作モードは、大きなリスト(例えば、ディレクトリ内のすべてのMP3タイトル)を表示する場合のメモリ消費量を削減するために使用されます。通常、こうしたリストは1つのアプリケーション(例えば、メディアアプリケーション)によって提供され、別のアプリケーション(例えば、ヒューマンマシンインターフェイス)によってその一部のみが表示されます。

ライターアプリケーションは、仮想リストの長さとうィンドウの数を定義します。こうしたウィンドウには、そのリストの一部だけが含まれる可能性があります。リーダーアプリケーションは、各ウィンドウの対象範囲内にある場所からのみデータの読み取りを行います。他の場所からの読み取りは失敗します。そのようなユースケースでは、現在必要とされているリストの部分に関する情報をリーダーアプリケーションがライターアプリケーションに通知する必要があります。例えば、ヒューマンマシンインターフェイスは、完全なリスト内での現在のカーソル位置を提供するアプリケーション呼び出しを行うことができます。



### 例6.3 ウィンドウ表示リスト

オーディオプレーヤーデバイスのMP3 タイトルリストには、1,000,000 の要素があります。ヒューマンマシンインターフェイスは、ヘッドユニットディスプレイ、計器パネルディスプレイ、ヘッドアップディスプレイという3つの異なるディスプレイにこのリストを並行して表示する必要があります。

各ディスプレイは、別々に制御され、表示される行の数や完全なリスト内でのカーソル位置が異なります。

3つのカーソルのいずれかが動いたときに、ヒューマンマシンインターフェイスは新しい位置をイベントによってメディアアプリケーションに非同期で送信します。メディアアプリケーションは、3つのウィンドウをリストに提供します。3つのウィンドウのそれぞれは、3つのディスプレイの1つに関連付けられています。ウィンドウの更新は、カーソルが移動した後に少し遅延します。そのため、特定のディスプレイによって表示される行の周辺まで対象範囲を拡げるようなウィンドウ位置とウィンドウサイズの使用をお勧めします。

## 6.7. EB GUIDEモデルとEB GUIDEプロジェクト

EB GUIDEモデルとは、ヒューマンマシンインターフェイスの外観と動作を記述するすべての要素をまとめたものです。このモデルは全体がEB GUIDE Studio内で構築されます。EB GUIDEモデルは、PC上でシミュレートできます。

EB GUIDEモデルを対象デバイスで実行するには、EB GUIDEモデルをエクスポートし、結果として得られるバイナリファイルを対象デバイスにコピーします。

EB GUIDEプロジェクトは、EB GUIDEモデルと、モデリングに必要な設定で構成されます。このプロジェクトには、プロジェクト特有のオプション、カスタマイズ、リソース、またグラフィカルプロジェクトの場合にはハプティックパネルも含まれます。

EB GUIDEプロジェクトには、EB GUIDEモデル内で設定され、リンクされたオブジェクトが含まれます。これらのオブジェクトをEB GUIDEモデル要素と呼びます。例えば、以下のものはEB GUIDEモデル要素です。

- ▶ データプールアイテム
- ▶ イベント
- ▶ ステート
- ▶ ステートマシン

- ▶ ウィジェット
- ▶ リソース
- ▶ 言語

## 6.7.1. ストレージ形式

EB GUIDEプロジェクトは、ファイル形式がEB GUIDE独自のものである複数のファイルに保存されます。このファイル形式は、次の2つのファイル拡張子で表されます。

- ▶ `.ebguide` : EB GUIDEプロジェクトファイル用
- ▶ `.gdata` : その他すべてのプロジェクトファイル用

EB GUIDE Studioのストレージ形式は、次の疑似EBNF構文で定義されます。

```
INT = [0-9]+ ;

HEXINT = '0' ( 'x' | 'X' ) [a-fA-F0-9]+

FLOAT = <as represented in the C# specification> ;

STRING = " " ; //escape characters are supported as specified in MSDN

SUFFIX = [a-zA-Z_-][a-zA-Z0-9_-]* ;

COLOR = [a-fA-F0-9]{8} ;

IDENTIFIER = ( '_' | [a-zA-Z] ) ( [a-zA-Z] | [0-9] | '_' | '$' | '.' ) * ;

file = header object ;

header = 'EBGUIDE' INT '.' INT '.' INT '.' INT SUFFIX ';' ;

object = type '(' objectId ')' '{' propertyList '}' ;

type = identifier [ '<' type { ',' type } '>' ] ;

property = identifier ':' value ;

value = bool
      | int
      | float
      | string
      | color
      | object
```

```
| externalObject
| nullObject
| objectReference
| propertyReference
| list ;

string : STRING { '\\' STRING } ;

int = [ '+' | '-' ] INT
      | HEXINT ;

color = '#' COLOR ;

float = [ '+' | '-' ] FLOAT ;

bool = 'true' | 'false' ;

externalObject = '(' objectId ')' ;

nullObject = type '(' 'none' ')' ;

objectReference = '@' objectId '(' type ')' ;

propertyReference = identifier '@' objectId '(' type ')' ;

list = type '[' [ value { ',' value } ] ']' ;

identifier = IDENTIFIER | STRING ;

objectId = GUID ; //encoded as hex digits in
                //the XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX format
```

EB GUIDEのプロジェクトディレクトリ\$GUIDE\_PROJECT\_PATH/<project name>/には、以下のものが含まれません。

- ▶ 設定、名前空間、ステートマシンに関連するファイルがあるディレクトリ
- ▶ プロジェクト固有のリソースがあるresourceディレクトリ。詳しくは、[6.15「リソース管理」](#)をご覧ください。
- ▶ コンテキスト、データプール、イベントシステム、言語、スキンのデータがある.gdataファイル
- ▶ .xliffファイル、.psdファイル、.fbxファイルの読み込みエラー、移行またはインポートメッセージに関する情報を含む.txtファイル

## 6.7.2. EB GUIDEプロジェクトの検証基準

EB GUIDE Studioは、EB GUIDEプロジェクトに対して2つのタイプの検証チェックを実行します。

### 6.7.2.1. EB GUIDEを開くときの検証

EB GUIDEプロジェクトを開くと、EB GUIDE Studioによって構造に関するいくつかの検証が実行されます。例えば、次のようなものです。

- ▶ .ebguideプロジェクトファイルが存在しない、または複数の.ebguideファイルが同じフォルダーに置かれている
- ▶ オブジェクトIDが重複している
- ▶ 子オブジェクトがEB GUIDEプロジェクト内に見つからない
- ▶ 重複するプロパティ名がある
- ▶ リストアイテムの値に一貫性がない
- ▶ .gdataファイルでのEB GUIDE Studioバージョン番号が.ebguideファイルでのEB GUIDE Studioバージョン番号と対応していない
- ▶ 不明なタイプが参照されている

これらの基準のいずれかに一致する場合は、EB GUIDEプロジェクトを開くことができず、エラーのタイプとプロジェクトファイル内でのそのエラーの位置を示すログファイル\$GUIDE\_PROJECT\_PATH/<project name>/<project name>\_LoadingErrorLog.txtが作成されます。

EB GUIDEプロジェクトを開く方法については、[10.2「プロジェクトを開く」](#)をご覧ください。

### 6.7.2.2. 問題検出コンポーネントを使用した検証

EB GUIDEプロジェクトがすでに開かれている場合は、問題検出コンポーネントでEB GUIDEモデルを検証できません。例えば、次のようなエラーがあります。

- ▶ EB GUIDEスクリプトの使用法が無効である
- ▶ デフォルト遷移が見つからない
- ▶ アイテムのリンク先が見つからない

何らかのエラーが見つかった場合、EB GUIDEモデルのシミュレーションやエクスポートは行えません。

EB GUIDEモデルの検証方法については、[10.4「EB GUIDEモデルの検証およびモデル実行」](#)をご覧ください。

## 6.8. イベント処理

### 6.8.1. イベントシステム

イベントシステムは、アプリケーション内部での通信、またはアプリケーション同士での通信をサポートする非同期メカニズムです。

EB GUIDE イベントシステムでは、すべてのイベントを送信された順序どおりに伝達します。イベントを、異なるサブスクライバに事前に定義された順序で伝達することはできません。

## 6.8.2. イベント

EB GUIDE のイベントは、一意のイベントIDを持ち、イベントグループに属しているモデル要素です。イベントIDは、イベントを送受信するためにEB GUIDE GTFによって使用されます。



### Example 6.4. イベントの使用

回転ボタンや一連のハードキー(左、右、上、下、Enterなど)があるヒューマンマシンインターフェースでは、ユーザーが次に何を操作すべきか明確ではない場合があります。そのため、通常、このようなシステムでは現在アクティブなディスプレイ要素が強調表示されます。例えば、YESボタンは色付きの枠でアクティブとマークされます。EB GUIDE Studioでは、フォーカスウィジェット機能を使用してこの強調表示機能をモデル化します。現在フォーカスされている要素、つまりfocusedプロパティがtrueに設定されている要素はアクティブな要素でもあります。また、フォーカスパスを形成するこの要素の親もアクティブです。フォーカスされている要素がキー入力や回転入力を処理できない場合、入力は逆方向つまりルート要素に向かってフォーカスパスに沿って処理されます。フォーカスパスのいずれかの要素が入力を処理すると、このフォーカスは処理済みと見なされます。

タッチ入力を使用するヒューマンマシンインターフェースでは、操作は特定の位置で1つの要素を使用して行われます。例えば、タッチスクリーンのYESボタンを押すとき、この入力に不明瞭な点はありません。なぜなら、ディスプレイが押された位置に基づいて、システムはYESボタンで操作が行われたことを認識するからです。

イベントグループIDの0~65535は、EB GUIDE product lineでの内部使用のために予約されています。次の表に示すイベントグループは例外です。

Table 6.3. 許可されるイベントグループとID

イベントグループ	ID	詳細									
デフォルト	2	内部グループ、つまりシーンを実行中のコアのみがイベントを受信します。									
キー入力イベント	10	次のパラメータを設定できます。 <table border="1" data-bbox="667 1787 1437 2000"> <thead> <tr> <th>パラメータ</th> <th>タイプ</th> <th>詳細</th> </tr> </thead> <tbody> <tr> <td>display</td> <td>整数</td> <td>入力イベントを受信するシーンID</td> </tr> <tr> <td>status</td> <td>整数</td> <td>0: キー押下 1: キーリリース</td> </tr> </tbody> </table>	パラメータ	タイプ	詳細	display	整数	入力イベントを受信するシーンID	status	整数	0: キー押下 1: キーリリース
パラメータ	タイプ	詳細									
display	整数	入力イベントを受信するシーンID									
status	整数	0: キー押下 1: キーリリース									

イベントグループ	ID	詳細		
		パラメータ	タイプ	詳細
				2: キーUnicode
タッチ入力イベント	11	次のパラメータを設定できます。		
		パラメータ	タイプ	詳細
		display	整数	入力イベントを受信するシーンID
		status	整数	0: タッチ押下 1: タッチ移動 2: タッチリリース 3: 近接移動 4: 新規タッチ 5: タッチ終了
		x	整数	タッチイベントのX座標
		y	整数	タッチイベントのY座標
		fingerId	整数	マルチタッチサポートの複数の 並行タッチ位置をトラッキングする 番号
回転入力イベント	12	次のパラメータを設定できます。		
		パラメータ	タイプ	詳細
		display	整数	入力イベントを受信するシーンID
		increment	整数	増分値
システム通知イベント	13	Androidなどの画面回転、ライフサイクル管理、シャットダウンといったシステムイベントに使用されます。		

以下の図に、EB GUIDE Studioでタッチ、キー、および回転イベントをモデル化する方法を示します。

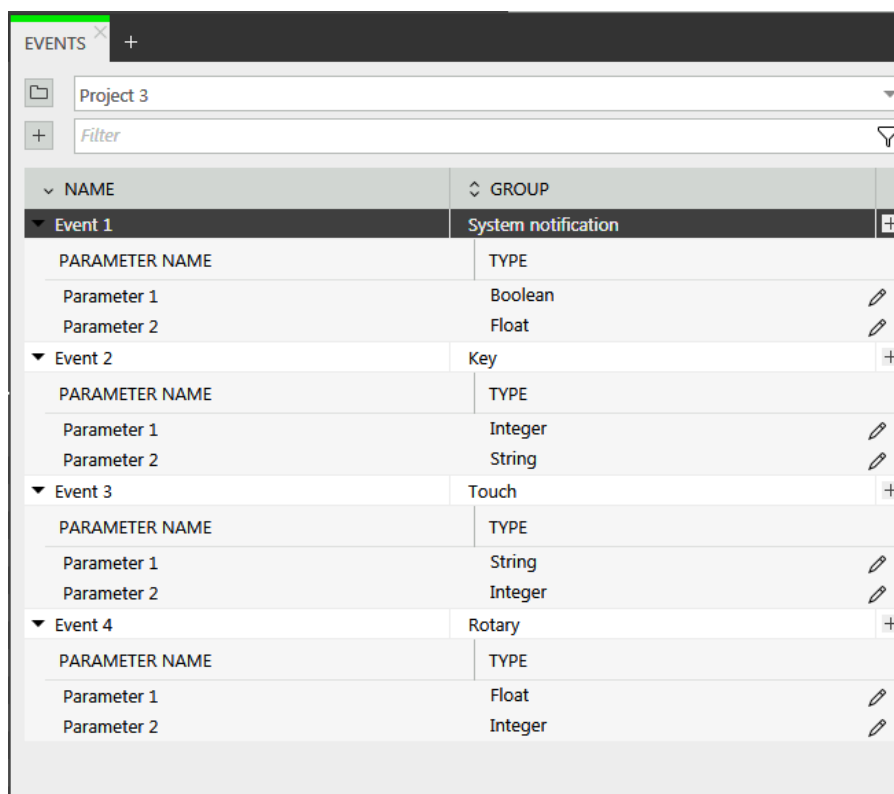


Figure 6.11. イベントグループおよびイベントIDの例

残りの範囲のグループIDは、顧客固有のアプリケーションで使用できます。

手順については、以下をご覧ください。

- ▶ [9.1「イベントの追加」](#)
- ▶ [9.3「イベントへの対応」](#)

リファレンスについては、[13.5「イベント」](#)をご覧ください。

## 6.9. 拡張機能

### 6.9.1. EB GUIDE Studio拡張機能

EB GUIDE Studio拡張機能は、EB GUIDE Studioに対する補足であり、すべてのEB GUIDEモデルで有効です。EB GUIDE Studio拡張機能はEB GUIDE GTFとは無関係です。

詳しくは、[第3章「サポート」](#)までお問い合わせください。



## 6.9.2. EB GUIDE GTF拡張機能

EB GUIDE GTF拡張機能はEB GUIDE GTFに対する補足であり、EB GUIDE Studioに追加の機能を提供しますが、1つのEB GUIDEモデルに対してのみ有効です。EB GUIDE GTF機能拡張はEB GUIDE GTFをベースにしています。

以下はEB GUIDE GTF拡張機能の主な例です。

- ▶ ウィジェットの.new機能
- ▶ 新しいEB GUIDEスクリプト関数

EB GUIDE GTF拡張機能はダイナミックリンクライブラリ(.dll)、または共有オブジェクト(.so)ファイルです。

EB GUIDE GTF拡張機能は、サードパーティライブラリを含め、以下のディレクトリに配置してください。

```
$GUIDE_PROJECT_PATH/<project name>/resources/target
```

詳細および手順については、EB GUIDE GTFユーザーガイドをご覧ください。

EB GUIDE GTF拡張機能の例については、EB GUIDE SDKの使用例をご覧ください。すべてのEB GUIDE SDKサンプルが付属するEB\_GUIDE\_Examples.zipアーカイブは、<https://www.elektrobit.com/ebguide/learn/resources/>からダウンロードできます。なお、EB GUIDE SDKサンプルの操作方法については、EB GUIDE Studio Howto Using examples in EB GUIDE Studio.pdfファイル(上記の.zipアーカイブに収録)をご覧ください。

クラスとインターフェースの詳細については、EB GUIDE GTFアプリケーションプログラミングインターフェースに関するドキュメントをご覧ください。

## 6.9.3. EB GUIDE Monitor拡張機能

EB GUIDE Monitor拡張機能はEB GUIDE Monitorに追加の機能を提供します。

以下はEB GUIDE Monitor拡張機能の主な例です。

- ▶ EB GUIDE Monitorの追加コンポーネント
- ▶ シミュレーション中にスクリーンショットを作成する拡張機能

EB GUIDE Monitor拡張機能の例については、EB GUIDE SDKの使用例をご覧ください。すべてのEB GUIDE SDKサンプルが付属するEB\_GUIDE\_Examples.zipアーカイブは、<https://www.elektrobit.com/ebguide/learn/resources/>からダウンロードできます。なお、EB GUIDE SDKサンプルの操作方法については、EB GUIDE Studio Howto Using examples in EB GUIDE Studio.pdfファイル(上記の.zipアーカイブに収録)をご覧ください。

クラスとインターフェースの詳細については、EB GUIDE Monitorアプリケーションプログラミングインターフェースに関するドキュメントをご覧ください。

## 6.10. ガンマ補正レンダリング

### 6.10.1. 概念

ガンマ補正は、レンダリングパイプラインで重要な役割を果たします。ガンマ補正は、画面およびイメージカラーストレージでの色の再現に影響します。ガンマは画面の色値と知覚明度の関係を表します。ガンマの例については、[figure 6.12. “ガンマの例”](#)をご覧ください。

人間の視覚システム(HVS)は同じ動作を示します。明るいイメージ領域の輝度差よりも暗いイメージ領域の輝度差により敏感になります。一般的な8ビットのイメージ形式(JPEG、PNG)ではこの事実を利用して非線形伝達関数を使用するsRGB色空間に色を格納し、暗いイメージ領域の精度を向上させています。これは、線形のテクスチャカラー入力に依存している3Dライティング計算とテクスチャアルファブレンディングに影響します。そのため、EB GUIDEはガンマ補正レンダリングを使用して[figure 6.13. “sRGBテクスチャの例”](#)に示すようにこれらの影響を相殺します。

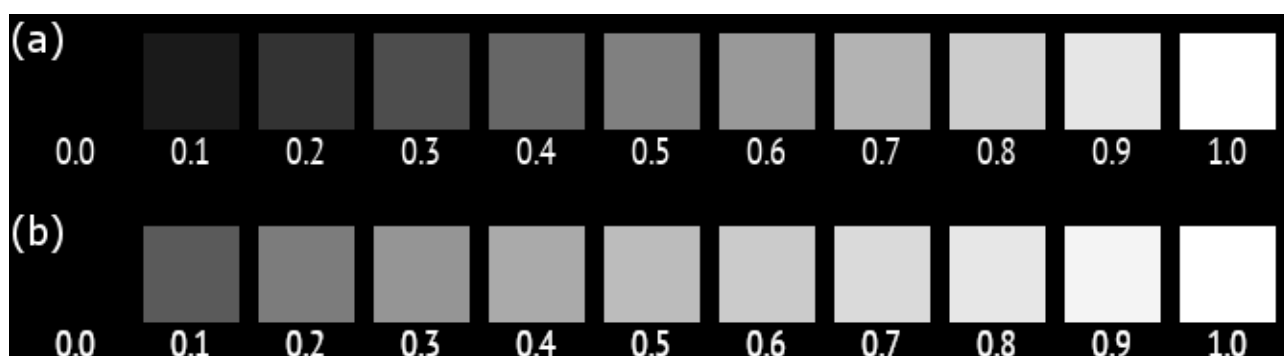


Figure 6.12. ガンマの例

(a)色が付いた四角形の下値はグレーレベルを示します。色値と表示輝度は非線形関係にあるため、50%の明るさは正しく校正されたモニターでおよそ0.7グレーレベルになります。画面の非線形輝度応答にかかわらず、相対的な差は知覚的に均一です。(b)表示の前にガンマエンコーディングされる色値です。エンコーディングガンマがディスプレイのガンマを相殺するため、50%の明るさは0.5グレーレベルになります。

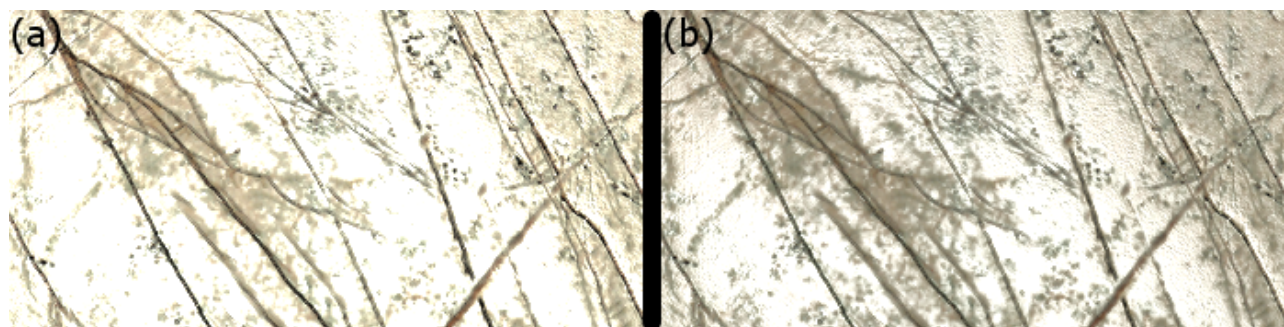


Figure 6.13. sRGBテクスチャの例

(a) sRGBテクスチャはライティング計算で修正されず、出力はガンマ補正されません。ライティングは過飽和状態で、詳細は不鮮明になります。(b) sRGBテクスチャはライティング前にリニアライズされ、結果がガンマ補正されます。詳細と表面構造が可視化されます。

## 6.10.2. EB GUIDE Studioでのガンマ補正

EB GUIDE Studioでは、ガンマ補正レンダリングで次のようにレンダリングパイプラインを設定する必要があります。

- ▶ 出力設定がディスプレイ自体のガンマエンコーディングを制御します。
- ▶ sRGBイメージとして扱うイメージおよびテクスチャリソースを入力設定でEB GUIDE Studioに示し、レンダリングパイプラインがレンダリング操作のためにそれらを正しくリニアライズするようにします。

入力エンコーディングを設定するには、使用される各イメージまたはテクスチャ向けに設定する必要があります。イメージ形式自体はsRGBエンコーディングに関する情報を提供しないことに注意してください。あらかじめこの情報を取得しておく必要があります。手順については、[12.8「チュートリアル:ガンマの正しいレンダリング」](#)をご覧ください。

## 6.11. イメージベースドライティング

イメージベースドライティング (IBL) は、イメージを3Dオブジェクトのライトとして使用できるようにする技法です。EB GUIDE Studioでは、イメージベースドライウィジェットによってIBLが適用されます。このウィジェットはシーングラフノードに適用できます。1つのシーングラフに複数のイメージベースドライを持たせることはできません。複数を追加しても、階層で最初のイメージベースドライのみがシーンで使用されます。



Figure 6.14. イメージベースドライティングの例左: 3点ライトで照らされたセラミックPBR GGX材質のティーポット。中央: イメージベースドライティング (IBL) を使用すると、ティーポットが仮想環境によって照らされ、セラミックPBR GGX材質が写実的になります。右: さらに、材質パラメータを空間的に変動させるためにテクスチャを使用。

IBLデータタイプの詳細については、[13.3.7「IBL」](#)をご覧ください。イメージベースドライウィジェットの詳細については、[13.12.4.4「イメージベースドライ」](#)をご覧ください。

## 6.11.1. IBLGenerator、ファイル形式とインポート

ライティング情報を格納するには、ハイダイナミックレンジイメージデータをサポートするイメージ形式が必要です。EB GUIDE Studioでは、次の2つのIBL形式をサポートしています。

- ▶ Portable Float Map(.pfm)
- ▶ RGBE(.hdr)

RGBE形式の場合、EB GUIDEではXYZ色空間をサポートしていません。-Y +X方向のみがサポートされます。

EB GUIDE StudioでこれらのIBLファイルを使用するには、.ebibl形式に変換する必要があります。この変換にはIBLGeneratorを使用します。IBLGeneratorは、コマンドラインによって制御され、\$GUIDE\_INSTALL\_PATH\toolsにあるインストール環境に含まれています。手順については、[8.1.6「IBLファイルのインポート」](#)をご覧ください。

IBLファイルには、立方体、球体、または経緯度パラメタリゼーションのいずれかでイメージを含めることができます。IBLGeneratorでは、パラメタリゼーションタイプを選択できます。IBLGeneratorに用意されているすべてのオプションを確認するには、コマンドラインに次のように入力します。IBLGenerator.exe -h

## 6.11.2. OpenGLレンダラーによるIBLの制限

OpenGL 3レンダラーは常にIBLをサポートしています。ただし、OpenGLレンダラーを使用する場合は、OpenGL ES 2.0ドライバが以下のOpenGL拡張機能をサポートしている必要があります。これらの拡張機能が1つでもサポートされていない場合、イメージベースドライトウィジェットは無視されます。

- ▶ GL\_EXT\_shader\_texture\_lod
- ▶ GL\_EXT\_texture\_rg
- ▶ GL\_OES\_texture\_float
- ▶ GL\_OES\_texture\_half\_float

## 6.12. 言語

### 6.12.1. EB GUIDE Studioの表示言語

EB GUIDE Studioにはグラフィカルユーザーインターフェースを表示する言語が数多く用意されています。表示言語はプロジェクトセンターのオプションタブで選択できます。

手順については、[10.6「EB GUIDE Studioの表示言語の変更」](#)をご覧ください。

## 6.12.2. EB GUIDEモデルの言語

ほとんどのヒューマンマシンインターフェースでは、テキストをユーザーの優先する言語で表示できます。そのような言語の管理機能もEB GUIDE Studioによって提供されます。

プロジェクト設定にEB GUIDEモデルの言語を追加します。その後、テキストをエクスポートしてローカライズ担当のサービスプロバイダーに送信し、翻訳済みテキストをEB GUIDEモデルにインポートします。

言語サポートをすべてのデータプールアイテムタイプに追加することが可能です。そうすれば、EB GUIDEモデルによってテキストを異なる言語で表示できます。データプールアイテムは、各言語の値を定義します。言語をサポートするには、言語サポートプロパティをそれぞれのデータプールアイテムで選択します。エクスポートされるEB GUIDEモデルの言語は、ランタイムに変更できます。

### NOTE



スキンのサポートは使用できません

データプールアイテムに言語サポートを定義した場合、同じアイテムにスキンのサポートを追加することはできません。

詳細については、[8.4.1「EB GUIDEモデルへの言語の追加」](#)および[6.12.3「言語依存テキストのエクスポートとインポート」](#)をご覧ください。



### Example 6.5.

他言語ヒューマンマシンインターフェースの言語依存テキスト

プロジェクト設定には、3つの言語が追加されています。英語、ドイツ語、フランス語が追加されていること。これで、ヒューマンマシンインターフェースの変化に伴って変化するラベルをモデル化できます。そのためには、ラベルのtextプロパティを、英語の値Welcome、ドイツ語のWillkommenとフランス語のBienvenueという値を持つデータプールアイテムにリンクします。

手順については、[12.6「チュートリアル: データプールアイテムに言語依存テキストを追加する」](#)をご覧ください。

## 6.12.3. 言語依存テキストのエクスポートとインポート

すべての言語依存テキストのエクスポート、編集、翻訳、およびインポートを行うには、EB GUIDE Studioのエクスポート機能とインポート機能を使用します。テキストは.xliffファイルにエクスポートされます。.xliff (XML Localization Interchange File Format)は、抽出されたテキストを格納し、ローカライズプロセスのステップ間でデータを搬送するためのXMLベースの形式です。.xliffは、ローカライズ担当のサービスプロバイダーに送信でき、任意の翻訳ツールで解読できます。

変換後、翻訳済み .xliff ファイルを EB GUIDE モデルにインポートします。

手順については、[10.8「言語依存テキストのエクスポートとインポート」](#)をご覧ください。

.xliff ファイルの構成は、次のようになっています。

- ▶ ヘッダーには、ソースおよびターゲット言語に関するメタデータが含まれています。
  - ▶ `source-language` および `target-language` タグは、言語名の表現に関する ISO 639 標準規格と国コードの表現に関する ISO 3166-1 標準規格に従っています。
  - ▶ プロジェクトと言語のすべてペアについて、英数字からなる一意の `sourceLanguageid` および `targetLanguageid` が作成されます。これらの ID は、別のプロジェクトまたはターゲット言語からの .xliff ファイルの意図しないインポートを防止します。
- ▶ `trans-unit` 要素には、ローカライズ可能なデータが含まれます。`trans-unit` 要素は、ソーステキストを格納する `source` 要素と、翻訳されたテキストを格納する `target` 要素を保持します。EB GUIDE モデルに新しい言語が追加されると、`target` 要素にソース言語が設定されます。そのため、.xliff ファイルのエクスポート時には、まだ翻訳されていないすべての `target` 要素にソース言語が表示されます。

## 6.13. 名前空間

EB GUIDE Studio では、名前空間を使用して、データプールアイテムやイベントのようなモデル要素のグループを作成します。こうしたグループには通常、定義された機能があります。それぞれの名前空間では、異なる名前空間のモデル要素に同じ名前を持たせることができるように、モデル要素のネーミングスコープを作成しています。

各モデル要素は、厳密に1つの名前空間のみに属しています。

ルート名前空間は、デフォルトの名前空間であり、削除することも名前を変更することもできません。ルート名前空間は EB GUIDE プロジェクトと同じ名前を持ちます。その他すべての名前空間は、この名前空間から派生したものです。以下の場合、モデル要素が必ずデフォルトの名前空間に追加されます。

- ▶ コンテキストメニューのデータプールアイテムへのリンクの追加を選択し、新しいデータプールアイテムを作成する場合
- ▶ 遷移または内部遷移のトリガーコンボボックスでイベントを作成する場合

別の名前空間にモデル要素を移動できます。

### NOTE



#### モデル要素の移動

ある名前空間からモデル要素を別の名前空間に移動し、移動先の名前空間にすでに同じ名前の要素が含まれていた場合、この移動の操作は失敗し、エラーメッセージが表示されます。



#### Example 6.6.

### 名前空間ツリー

[figure 6.15. “名前空間ツリーの例”](#)に名前異空間の例を示します。`myProject`名前空間は、デフォルトの名前空間であり、EB GUIDEプロジェクトの名前でもあります。一部の名前空間は、他の名前空間内にネストされています。

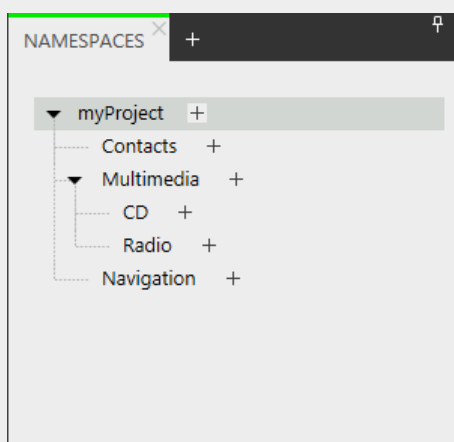


Figure 6.15. 名前空間ツリーの例

詳しくは、[6.5「グラフィカルユーザーインターフェイスのコンポーネント」](#)をご覧ください。

手順については、[9.12「名前空間の操作」](#)をご覧ください。

## 6.14. Photoshopファイル形式のサポート

EB GUIDE Studioは、.psdファイル形式をサポートします。サポートされる色空間は、8ビット、16ビット、32ビットのRGB、それにCMYKです。.psdファイルを直接インポートしたり、.psdファイルからイメージを抽出したりできます。

### インポート

.psdファイルにある要素は直接モデル内に配置され、ウィジェットツリーが作成されます。ウィジェットツリーは、.psdファイルのレイヤーから作成されたコンテナ、イメージ、ラベルで構成されます。手順については、[8.1.4「ビューに.psdファイルをインポートする」](#)をご覧ください。次のことに注意してください。

- ▶ .psdファイルのレイヤーが非表示に設定されている場合、対応するコンテナまたはイメージの`visible`プロパティの横にあるチェックボックスは選択解除されています。
- ▶ .psdファイルのテキストレイヤーは、ラベルとしてインポートされます。対応するラベルは、インポート後にウィジェットツリーで確認できます。
- ▶ .psdファイルのイメージレイヤーは、イメージとしてインポートされます。
- ▶ .psdファイルのグループレイヤーは、コンテナとしてインポートされます。コンテナの名前は対応するグループレイヤーと同じものになります。コンテナには、イメージ、ラベル、またはその他のコンテナを含めることができます。

## 抽出

.psd ファイルにあるイメージが含まれているサブディレクトリが作成されます。ただし、作業しているEB GUIDE モデルは変更されません。手順については、[8.1.5「.psdファイルからイメージを抽出する」](#)をご覧ください。

## 制限事項

EB GUIDE Studioでは、Photoshopファイル形式が持つ次の機能をサポートしていません。

- ▶ レイヤーエフェクト、フィルタ、テキストチャ
- ▶ RGBおよびCMYK以外のカラーモデル
- ▶ マスク
- ▶ 1つのレイヤーに適用される複数のマスク(レイヤーマスクおよびベクトルマスク)
- ▶ テキストスタイリングおよびフォント
- ▶ 色チャンネルのみの使用

## 6.15. リソース管理

リソースは、EB GUIDE内で作成されないものの、プロジェクトで必要とされるコンテンツのことです。EB GUIDEプロジェクトのすべてのリソースは、リソースディレクトリ内に配置します。

リソースディレクトリは、`$GUIDE_PROJECT_PATH/<project name>/resources`にあります。

EB GUIDEは次のタイプのリソースファイルをサポートします。

1. フォント
2. .ebibl 3Dグラフィック用ファイル形式
3. イメージ
4. 3Dグラフィック用メッシュ
5. .psd ファイル形式

リソースをプロジェクトで使用するには、リソースファイルをディレクトリ`$GUIDE_PROJECT_PATH/<project name>/resources`に追加します。

### 6.15.1. フォント

フォントをプロジェクトで使用するには、フォントをディレクトリ`$GUIDE_PROJECT_PATH/<project name>/resources`に追加します。

サポートされているフォントタイプは、TrueTypeフォント(\*.ttf、\*.ttc)、OpenTypeフォント(\*.otf)、およびビットマップフォント(\*.fnt)です。



手順については、[8.1.2.4.1「ラベルのフォントの変更」](#)をご覧ください。

### 6.15.1.1. ビットマップフォント

EB GUIDE Studioのバージョン3.0では、Angelcodeによる\*.fntビットマップフォントをサポートしています。ビットマップフォントを作成するには、サードパーティのフォントジェネレータ(Angelcodeビットマップフォントジェネレータなど)を使用します。詳しくは、<http://www.angelcode.com>をご覧ください。

生成されたフォントの以下の設定項目について確認します。

- ▶ 適切なフォントサイズが定義されている。
- ▶ 文字セットがUnicodeである。
- ▶ フォント記述子がバイナリである。
- ▶ テクスチャが8ビットの.pngファイルとして提供されている。

次のことに注意してください。

- ▶ EB GUIDE Studioでは、ラベルのfontプロパティを使用してビットマップフォントのフォントサイズを変更することができません。つまり、.fntフォントを生成する際にサイズを定義する必要があります。
- ▶ ストロークウィジェット機能は、ビットマップフォントに適用されません。フォント用に特定のアウトラインが必要な場合は、.fntフォントの生成時に定義します。
- ▶ \$GUIDE\_PROJECT\_PATH/resourcesディレクトリ内に、サードパーティツールで生成した.fntビットマップフォントおよび.pngテクスチャファイル用のサブディレクトリを1つ作成します。EB GUIDE Studioでは、.pngファイルが.fntファイルと同じディレクトリ内にあるものと想定しています。

複数のビットマップフォントがある場合は、そうしたフォントごとにサブディレクトリを作成します。

### 6.15.1.2. マルチフォントサポート

EB GUIDE Studioでは、マルチフォントサポートを使用してフォントの独自の組み合わせを作成できます。この機能は、例えば、選択したフォントに必要なすべての文字が用意されていない場合などに役立ちます。そうした場合は、欠落している文字を別のフォントの文字で置き換えることができます。

マルチフォントサポートは、次のモデル要素に追加できます。

- ▶ タイプフォントのプロパティとフォントリストのエントリー
- ▶ タイプフォントのユーザー定義プロパティとフォントリストのエントリー
- ▶ タイプフォントのデータプールアイテムまたはフォントリストのエントリー

指定のフォントで使用するUnicode文字範囲を次のように定義できます。

- ▶ 単一のUnicode文字を使用: 0000など。
- ▶ カンマで区切った複数のUnicode文字を使用: 0000, 0001など。
- ▶ Unicode文字の範囲を使用: 0000-FFFFなど。
- ▶ カンマで区切った複数の範囲を使用: 0000-0022, 0045-0055など。

これらの文字は16進数形式を使用して指定します。

フォントは\$GUIDE\_PROJECT\_PATH/resourcesに用意されています。

マルチフォントサポートを追加すると、デフォルトのマルチフォント値が自動的に追加されます。デフォルトのマルチフォント値は削除できません。その優先度や範囲を編集することもできません。ただし、デフォルト値のサイズおよびフォントは編集できます。

▼ font	18 arialbd.ttf 25 fireflysung.ttf 12 a...			■
PRI0	FONT	RANGE	+	
0	18 arialbd.ttf	0000-00020	×	
1	25 fireflysung.ttf	00FF	×	
2	12 arial.ttf	FFFF	×	
<b>Def.</b>	30 PT_Sans_Na...	0-FFFFFFFF		

Figure 6.16. マルチフォントサポートを追加したタイプフォントのプロパティの例

マルチフォントサポートの使用方法については、[8.2.8「マルチフォントサポートの管理」](#)をご覧ください。

## 6.15.2. 3Dグラフィックのイメージベースドライティング

EB GUIDE Studioでは、イメージベースドライティングを使用できます。外部コマンドラインツールのIBLGeneratorは、.pfmまたは.hdrファイルを入力データとして受け取り、IBLリソースを表す.ebiblファイルを作成します。IBLリソースは、イメージベースドライウィジェットのiblプロパティによって使用されます。

.ebiblファイルの取得方法については、[8.1.6「IBLファイルのインポート」](#)をご覧ください。

バックグラウンド情報については、[6.11「イメージベースドライティング」](#)をご覧ください。

## 6.15.3. イメージ

イメージをプロジェクトで使用するには、イメージを\$GUIDE\_PROJECT\_PATH/<project name>/resourcesディレクトリに追加します。別のディレクトリにあるイメージを選択した場合は、そのイメージがこのディレクトリにコピーされます。

サポートされているイメージ形式は、Portable Network Graphic (\*.png)、JPEG (\*.jpg)、および9-patchイメージ (\*.9.png)です。

手順については、[8.1.2.3「イメージを追加する」](#)をご覧ください。

### 6.15.3.1. 9-patchイメージ

EB GUIDE Studioでは、9-patchイメージ方式に準拠する追加のメタ情報が含まれるイメージをサポートしています。9-patchイメージは伸縮可能な.pngイメージです。9-patchイメージには2つの黒色マーカーがあり、1つはイメージの上端を、もう1つはイメージの左端を示します。マーカーが示す範囲の外側は、拡大縮小の対象ではありません。マーカーの範囲内が拡大縮小されます。マーカーはEB GUIDE Studioに表示されません。

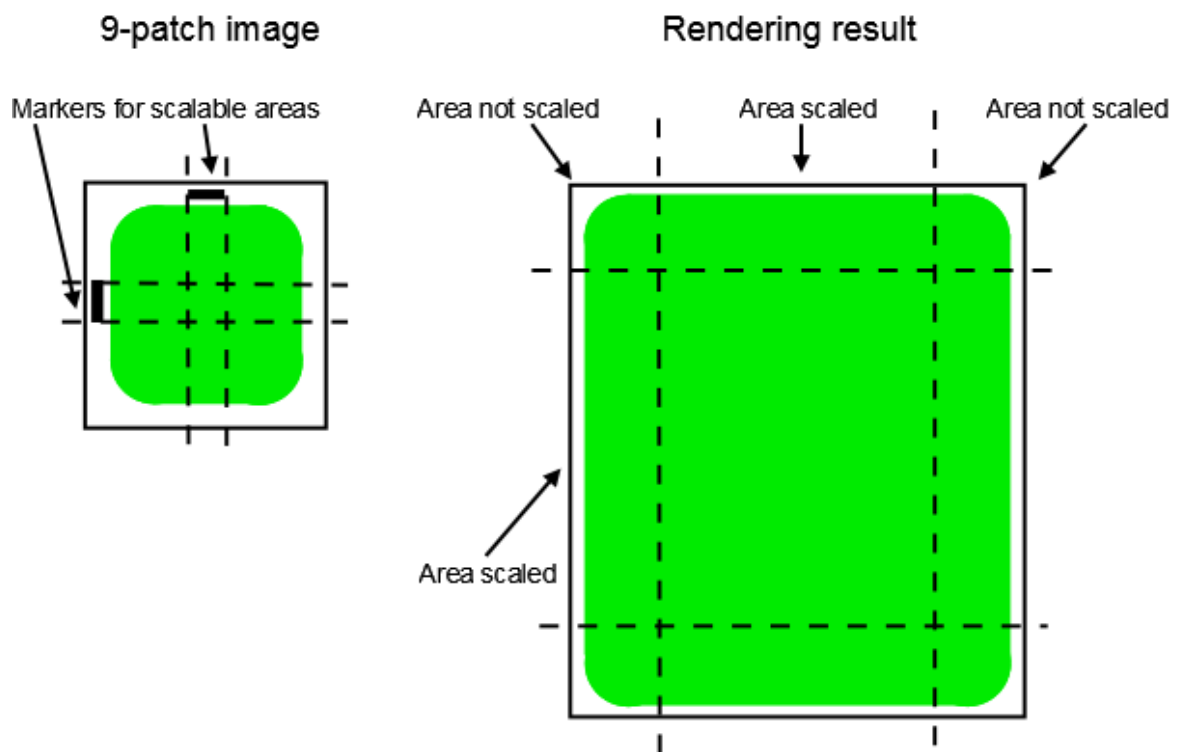


Figure 6.17. 9-patchの例

9-patchイメージを操作する際の注意事項を以下に示します。

- ▶ 9-patch処理はOpenGL ES 2.0以降のレンダラーでのみ機能します。
- ▶ 9-patch処理は.pngイメージにのみ適用されます。
- ▶ 9-patchイメージの場合は、\*.9.png拡張子が必須となります。
- ▶ イメージの上端と左端を示すマーカーを0個、1個、またはそれ以上指定できます。9-patch定義では、イメージの右端および下端の位置にテキストエリア用のマーカーを含めることもできます。これらのマーカーは、EB GUIDE Studioでは評価されません。

手順については、[8.1.2.3「イメージを追加する」](#)をご覧ください。

### 6.15.4. 3Dグラフィック用メッシュ

3DグラフィックファイルをEB GUIDE Studioにインポートすることができます。EB GUIDE Studioに3Dグラフィックファイルをインポートすると、`$GUIDE_PROJECT_PATH/<project name>/resources`にサブディレクトリが作成されます。メッシュは3Dグラフィックファイルで定義されていたとおりに、`.ebmesh`ファイルとしてインポートされます。詳細については、[6.1.3「3Dグラフィックファイルのインポート」](#)をご覧ください。

手順については、[8.1.3.1「ビューへのシーングラフの追加」](#)をご覧ください。

## 6.16. スクリプト言語EB GUIDEスクリプト

EB GUIDEスクリプトはEB GUIDEの組み込み型のスクリプト言語です。この章では、EB GUIDEスクリプト言語の機能、構文、使用方法を説明します。

### 6.16.1. アプリケーションの機能とエリア

EB GUIDEスクリプトは、例えば以下のようなプロジェクトのさまざまな場所で使用できます。

- ▶ ウィジェットのプロパティ内
- ▶ ステートマシン内(遷移またはステートの一部として)
- ▶ データプールアイテム内

EB GUIDEスクリプトの全機能がすべてのクラスで使用できるわけではありません。例えば、ローカルウィジェットプロパティにアクセスできるのは、スクリプトがウィジェットの一部分である場合にに限られます。一方、データプールへのアクセスはどこからでも可能です。

EB GUIDEスクリプトを使うと、モデル要素を直接操作できます。例えば、次の操作が可能です。

- ▶ イベントの発行
- ▶ データプールアイテムの書き込み
- ▶ ウィジェットプロパティの変更

### 6.16.2. プレフィックスと識別子

EB GUIDEでは、種類の異なるオブジェクトに同じ名前を付けることができます。例えば、イベントとデータプールアイテムにNapoleonという同じ名前を付けてもかまいません。これを可能にするには、EB GUIDEスクリプト内のすべての識別子(オブジェクトの名前)がプレフィックスを持つ必要があります。プレフィックスは、オブジェクトのタイプを定義するもので、その後にコロンが続きます。

EB GUIDEスクリプトで使われる一連のプレフィックスは固定されており、新しいプレフィックスを導入することはできません。使用可能なプレフィックス:

- ▶ ev: イベント
- ▶ dp: データプールアイテム
- ▶ f: ユーザー定義アクション(外部関数)
- ▶ v: ローカル変数

例えば、ev:NapoleonならNapoleonという名前のイベントで、dp:NapoleonならNapoleonという名前のデータプールアイテムという意味になります。

プレフィックスがない識別子は文字列定数です。

EB GUIDEでは、識別子にスペースや句読点を含め、多数の文字を使用できます。そのため、EB GUIDEスクリプトでは識別子を引用符で囲むことができます。例えば文字、数字、アンダースコアのみで構成される有効なC言語識別子のように、特殊記号が含まれない識別子であれば、引用符で囲む必要はありません。



#### Example 6.7. EB GUIDEスクリプトの識別子

```
dp:some_text = foo; // foo is a string here
dp:some_text = "foo"; // this statement is identical to the one above
dp:some_text = v:foo; // foo is the name of a local variable
// of course you can quote identifiers, even if it is not strictly necessary
dp:some_text = v:"foo";
// again, a string constant
dp:some_text = "string with spaces, and -- punctuation!";
// identifiers can also contain special characters, but you have to quote them
dp:some_text = v:"identifier % $ with spaces @ and punctuation!";
```

### 6.16.3. コメント

EB GUIDEスクリプトでは、C言語式のブロックコメントと、C++式の行コメントを使用できます。ブロックコメントは入れ子で記述できません。



#### Example 6.8.

### EB GUIDEスクリプトのコメント

```
/* this is a C style block comment */  
// this is a C++ style line comment
```

「todo」という文字列を含むすべてのEB GUIDEスクリプトコメントについて、EB GUIDE Studioではプロジェクトを検証したとき問題検出コンポーネントに警告が表示されます。この機能を使用し、すべての未処理タスクをマークしてそれらを一覧表示することができます。

#### NOTE



#### 条件スクリプトのデフォルトコメント

デフォルトでは、Conditional scriptタイプのデータプールアイテムまたはプロパティには、`// todo: auto generated return value, please adapt`というコメントが含まれています。警告が表示されないようにするには、必要なEB GUIDEスクリプトコードを入力した後、コメントからtodoの文字列を削除します。

## 6.16.4. データ型

EB GUIDEスクリプトは、強く型付けされた、静的型付けプログラミング言語です。すべての式は、明確に定義されたデータ型を持ちます。予期しないデータ型を与えるとエラーになります。

EB GUIDEスクリプトでは、次のデータ型がサポートされています。

- ▶ 整数
- ▶ Unicode文字列(string)
- ▶ 参照カウント付きのオブジェクト
- ▶ 上記および以下のデータ型への型定義
  - ▶ 色(32ビットのRGBA値を表す整数)
  - ▶ ブール値
  - ▶ 各モデル要素のID: データプールアイテム、ビュー、ステートマシン、ポップアップ(いずれも整数型)
- ▶ void(別名unit型)。例えば、Haskellなどの関数型言語で使用
- ▶ ウィジェット参照、イベント参照。これらはレコード型で、そのフィールドにアクセスするには、CやJavaで一般的な方法であるdot記法を使用します。これらのデータ型で新規のオブジェクトを直接作成することはできません。必要となった時点で自動的に作成されます。

すべてのデータ型とデータ型定義には互換性がなく、型のキャストはありません。そのため、スクリプトのコンパイルが正常に完了した後で型の安全性が保証されます。

## 6.16.5. 式

EB GUIDEスクリプトは、式を基盤として機能します。すべての言語構造は式です。複数の式を演算子で連結して、大きな式を記述できます。

式を評価するとは、式をそれに相当する値に置き換えることを意味します。



**Example 6.9.**  
整数値の評価

```
1 + 2 // when this expression is evaluated, it yields the integer 3
```

## 6.16.6. 定数と参照

基本的な式は、整数、色、ブール値、文字列定数、そしてモデル要素への参照です。

void型も定数値を持ち、その書き込み方法は次の2通りありますが、評価上の意味は同じです。

- ▶ 左右の波括弧を使用 {}
- ▶ キーワードを使用 unit



**Example 6.10.**  
定数の使用

```
"hello world" // a string constant
true          // one of the two boolean constants
ev:back       // the event named "back" of type event_id
dp:scrollIndex // the datapool item named "scrollIndex",
               // the type is whichever type the dp item has
5             // integer constants have a dummy type "integer constant"
5::int        // typecast your constants to a concrete type!
color:255,255,255,255 // the color constant for white in RGBA format

// the following are two ways to express the same
                if( true )
{
}
else
{
}

if( true )
    unit
else
    unit
```

## 6.16.7. 算術式と論理式

EB GUIDEスクリプトでは、次の算術式がサポートされています。

- ▶ 加算 (+)、減算 (-)、乗算 (\*)、除算 (/)、剰余 (%) を整数型の式に実行できます。
- ▶ 論理演算子のOR (||)、AND (&&)、NOT (!) は、ブール型の式に実行できます。
- ▶ 整数または文字列は、比較演算子のより大きい (>)、より小さい (<)、以上 (>=)、以下 (<=) を使って比較できます。
- ▶ データ型は、等価演算子の等しい (==) または等しくない (!=) を使って比較できます。

文字列は、等価演算子を使って比較できます。大文字と小文字は区別されません(例: (=Aa=))。

### NOTE



#### 等価演算子の使用

イベントおよび例えば3Dグラフィック、フォント、イメージなどのリソースデータ型は、等価演算子 (==) または (!=) を使って比較できません。

- ▶ 文字列どうしは (+) 演算子で連結できます。



### Example 6.11.

#### 算術式と論理式

```
10::int + 15::int // arithmetic expression of type int
dp:scrollIndex % 2 // arithmetic expression of type int,
// the concrete type depends on the type
// of dp:scrollIndex
"Morning Star" == "Evening Star" // type bool and value false (wait, what?)
"name" =Aa= "NAME" // type bool and value true
!true // type bool, value false
!(0 == 1) // type bool, value true
// as usual, parenthesis can be used to group expressions
((10 + dp:scrollIndex) >= 50) && (!dp:buttonClicked)
// string concatenation
"Napoleon thinks that " + "the moon is made of green cheese"
f:int2string(dp:speed) + " km/h" // another string concatenation
```

## 6.16.8. L値とR値

EB GUIDEスクリプトの式には、l-valuesおよびr-valuesの2種類があります。L値はアドレスを持ち、代入演算の左辺に配置できます。R値はアドレスを持たず、代入演算の左辺に配置できません。

- ▶ L値はデータプール参照、ローカルウィジェットプロパティ、またはローカル変数です。



- ▶ R値はイベントパラメータまたは定数式(文字列定数や整数定数など)です。

## 6.16.9. ローカル変数

let式でローカル変数を導入します。この式は一連の変数宣言とin式で構成され、ローカル変数はこのin式内のみで参照できます。変数はL値であり、代入演算の左辺で使用できます。変数のプレフィックスはv:です。let式の構文には、次の特徴があります。

```
let v:<identifier> = <expression> ;  
    [ v:<identifier> = <expression> ; ]...  
in  
    <expression>
```

let式のデータ型と値は、in式の使用時と同じです。

let式は入れ子にすることができます。外側のlet式の変数は、内側の式でも参照できます。



### Example 6.12. let式の使用方法

```
// assign 5 to the datapool item "Napoleon"  
let v:x = 5 in dp:Napoleon = v:x;  
  
// define several variables at once  
let v:morning_star = "Venus";  
    v:evening_star = "Venus";  
in  
    v:morning_star == v:evening_star; // Aha!  
  
let v:x = 5;  
    v:y = 20 * dp:foo;  
in  
{  
    // Of course you may have a sequence as the in expression,  
    // but parenthesis or braces are required then.  
    v:x = v:y * 10;  
    dp:foo = v:x;  
}  
// Because let expression also have types and values, we can have them  
// at the right hand side of assignments.  
dp:x = let v:sum = dp:x + dp:y + dp:z  
        in v:sum; // this is the result  
        // of the let expression  
  
// A nested let expression  
let v:x = dp:x + dp:y;
```

```
v:a = 5;
in
{
  let v:z = v:x + v:a;
  in
  {
    dp:x = v:z;
  }
}
```

## 6.16.10. Whileループ

while EB GUIDEスクリプトのループ構文は、CやJavaと似ています。条件式と実行式から構成されます。以下の構文を使用します。

```
while (<condition expression> ) <do expression>
```

条件式がfalseとみなされるまで、実行式が繰り返し評価されます。condition expressionはブール型で、実行式はvoid型でなければなりません。while式はvoid型であり、代入演算の左辺または右辺に記述することはできません。



### Example 6.13. whileループの使用方法

```
// Assume dp:whaleInSight is of type bool
while( ! dp:whaleInSight )
{
  dp:whaleInSight = f:lookAtHorizon();
}
```

## 6.16.11. If-then-else

if-then-else EB GUIDEスクリプトのは、CやJavaの三項条件演算子(?:)と似た動作をします。

if-then-else式は、次のサブ式で構成されます。

- ▶ 条件式
- ▶ then式
- ▶ else 式

以下の構文を使用します。

```
if ( < condition expression > ) <then expression> else <else expression>
```

if-then-else は、次のように動作します。

1. 最初に条件式が評価されます。この式はブール型でなければなりません。
2. 条件がtrueであれば、then式が評価されます。
3. 条件がfalseであれば、else式が評価されます。

if-then-else そのものが1つの式です。式全体のデータ型は、then式とelse式のデータ型です。この2つは一致していなければなりません。if-then-else式の値は、then式の値またはelse式の値で、どちらの値になるかは上記の規則に従います。

if-then-elseでは、else分岐を省略できる特殊なケースがあります。この記述が許されるのは式がvoid型の場合で、スクリプトから戻り値を帰すことはできません。



#### Example 6.14. の使用方法 if-then-else

```
// Assume dp:whaleInSight is of type bool
// and dp:user is of type string.
if( dp:whaleInSight && dp:user == "Captain Ahab" )
{
    dp:mode = "insane";
}
else
{
    dp:mode = "normal";
}

// Because if-then-else is also an expression,
// we may simplify the previous example:
dp:mode = if( dp:whaleInSight && dp:user == "Captain Ahab" )
    "insane"
    else
    "normal"

if ( <expression> ) <expression> // This is the reduced way of
    writing if-then-else
//It is an alternative to the following
if( <expression> ) { <expression> ; {} } else {}
```

## 6.16.12. 外部関数呼び出し

C言語で書かれた関数(外部関数と呼ぶ)を使って、EB GUIDEスクリプトの機能を拡張できます。

f: を先頭に付けた識別子は、外部関数の名前として扱われます。外部関数には引数リストと戻り値があり、これはC言語の関数と同じです。外部関数の構文は以下のとおりです。

```
f:<identifier> ( <expression> [ , <expression> ] ... )
```



### Example 6.15.

#### 外部関数の呼び出し

```
// write some text to the connection log
f:trace_string("hello world");
// display dp:some_index as the text of a label
v:this.text = f:int2string(dp:some_index);

// passing different parameters of matching type
f:int2string(v:this.x)
f:int2string(4)
f:int2string(dp:myInt)
f:int2string(v:myVar)

//passing parameters of different types
// starts an animation (parameter type GtfTypeRecord) from a script
// located in its parent widget
f:animation_play(v:this->Animation);

// checks the number of child widgets of a widget (parameter type widget)
f:widgetGetChildCount(v:this);

// traces debugging information about a datapool item (parameter type dp_id)
// to the connection log; uses the address of the datapool item as parameter
f:trace_dp(&dp:myFlag);
```

## 6.16.13. データプールアクセス

EB GUIDEスクリプトで書かれたスクリプトは、データプールアイテムを読み書きできます。プレフィックスdp:を持つ識別子は、データプールアイテム式と呼ばれます。この式のデータ型はdatapool item of type Xで、この「X」は参照先のデータプールエントリーのデータ型です。識別子にはデータプールアイテムの名前のみを含めることができます。あるいは、そのデータプールアイテムがデフォルトの名前空間にない場合、名前空間の名前の後にデータプールアイテムの名前が続きます。

X型のデータプールアイテムを代入演算の左辺に使い、X型の式を右辺に使った場合、データプールアイテムの値が書き込まれます。

プログラム内でデータプールアイテムを代入演算の左辺ではない場所に使った場合、データプールアイテムの値が読み取られません。



### Example 6.16.

#### データプールアイテム値の代入

```
// Assume intA to be of type int. Assign 10 to it.
dp:intA = 10;
// Assume strA to be of type string. Assign the string "blah" to it.
dp:strA = blah; // Yes, we can omit the quotes, remember?
dp:strA = 42; // Error: integer cannot be assigned to string

// Assign the value of the datapool item intB to intA.
// Both datapool items must have the same type.
dp:intA = dp:intB;
// Multiply the value of intB by two and assign it to intA.
dp:intA = 2 * dp:intB;
// Use the value of a datapool item in an if-clause.
if( dp:speed > 100 )
{
    // ...
}
```

以下の演算子がデータプールアイテムに使用できます。

- ▶ 参照演算子(&)はデータプールアイテムに使用できます。この演算子を使うと、データプールアイテムの値ではなくアドレスを参照できます。参照演算子は、`dp_id`型のパラメータを渡すために外部関数で使用します。
- ▶ リダイレクトリンク演算子(=>)は、データプールアイテムのリンクターゲットを変更します。リンクソースにすることができるのは、すでにリンクされているデータプールアイテムのみです。

## 6.16.14. ウィジェットプロパティ

スクリプトがウィジェットの一部である場合は、スクリプトからそのウィジェットのプロパティにアクセスできます。EB GUIDEスクリプトでは、このプロパティにドット記法でアクセスするために`v:this`という変数が作成されます。

スクリプトは、ウィジェットのプロパティに接続されると、そのウィジェットの一部となります。クリックやボタン操作などの入力への反応としてスクリプトを接続するケースがこれに該当します。



### Example 6.17.

#### ウィジェットプロパティの設定

```
// assume this script is part of a widget
v:this.x = 10; // if the widget has an x-coordinate

v:this.text = "hello world"; // if the widget is a label and has a text property
// assume testEvent has one integer parameter
fire ev:testEvent(v:this.x);
```

スクリプトがウィジェットの一部分である場合は、スクリプトからウィジェットツリーにある他のウィジェットのプロパティにもアクセスできます。

アロー演算子(->)を使って、ウィジェットツリー内の他のウィジェットを参照します。以下の構文を使用します。

```
<expression> -> <expression>
```

式の左辺はウィジェットを参照し、右辺は子ウィジェットの名前を文字列で記述する必要があります。親ウィジェットへ移動するには、右側に^記号を付けます。アロー式全体が1つのウィジェットを参照します。

ウィジェットツリーの別の場所を参照すると、ランタイムのパフォーマンスが低下することがあります。複数のプロパティを効率よく操作するには、ウィジェットを変数に代入します。



### Example 6.18.

#### ウィジェットプロパティへのアクセス

```
v:this.x          // access the properties of the current widget
v:this->^.x       // access the x property of the parent widget
v:this->^->caption.text // access the text property of a label called caption,
                    // read: "go-to parent, go-to caption, text"

// Modify several properties of the caption.
// This way, the navigation to the caption is only performed once.
let v:cap = v:this->^->caption
in
{
  v:cap.textColor = color:0,0,0,255;
  v:cap.x += 1;
  v:cap.y += 1;
}
```

## 6.16.15. リスト

データプールアイテムとウィジェットプロパティには、リストを格納できます。添字演算子([])を使うと、リスト要素にアクセスできます。以下の構文を使用します。

```
<expression> [ <expression> ]
```

最初の式ではリストの型を評価し、2番目の式では整数値を評価する必要があります。リストがlist A型である場合、リスト添字式全体はA型でなければなりません。

リスト添字式を代入演算の左辺に使うと、参照先リスト要素の値が書き込まれます。

lengthキーワードは、リストにある要素の数を返します。このキーワードをリスト式の前面に配置すると、式全体が整数型になります。



### Example 6.19. リスト

```
// Assume this widget is a label and dp:textList is a list of strings
v:this.text = dp:textList[3];

dp:textList[1] = v:this.text; // writing the value of the list element

v:this.width = length dp:textList;// checking the length of the list
dp:textList[length dp:textList - 1] = "the end is here";
```

EB GUIDEスクリプトでは現在、リスト要素の追加と削除はサポートされていません。

リストの終端を超えてリスト要素にアクセスしようとすると、スクリプトの実行が即座に中止されます。リストへのアクセスが常に範囲を超えないように注意してください。

## 6.16.16. イベント

EB GUIDEスクリプトには、イベントを処理する以下の式が用意されています。

- ▶ `fire`式はイベントを送信します。以下の構文を使用します。

```
fire ev:<identifier> ( <parameter list> )
```

イベントにパラメータを指定できますが、必須ではありません。`fire`式のパラメータリストは、発火したイベントのパラメータと一致する必要があります。イベントにパラメータがなければ、括弧の中は空白にします。



### Example 6.20. `fire`式の使用

```
fire ev:toggleView(); // the event "toggleView" has no parameters
fire ev:mouseClick(10, 20); // "mouseClick" has two integer parameters
fire ev:userNameEntered("Ishmael"); // string event parameter
```

- ▶ `fire_delayed`式は、指定された時間が経過した後でイベントを送信します。以下の構文を使用します。

```
fire_delayed <time> , ev:<identifier> ( <parameter list> )
```

`time`パラメータは、この遅延時間をミリ秒単位で指定する整数値です。



### Example 6.21. `fire_delayed`式の使用

```
fire_delayed 3000, ev:mouseClick(10, 20); // send the event "mouseClick"
//in 3 seconds.
```

- ▶ `cancel_fire`式は、遅延中のイベントを取り消します。以下の構文を使用します。

```
cancel_fire ev:<identifier>
```

- ▶ `match_event`式では、スクリプトの実行がイベントによってトリガーされたものかどうかをチェックします。以下の構文を使用します。

```
match_event v:<identifier> = ev:<identifier>
in
    <expression>
else
    <expression>
```

`match_event`式の型は、`in`式と`else`式の型です。この2つは一致していなければなりません。

`match_event`式では、`else`分岐を省略できる特殊なケースがあります。この記述が許されるのは式が`void`型の場合で、スクリプトから戻り値を帰すことはできません。



#### Example 6.22.

##### `match_event`式の使用

```
match_event v:theEvent = ev:toggleView in
{
    // this code will be executed when the "toggleView" event
    // has triggered the script
    dp:infoText = "the view has been changed";
}
else {}

match_event ( <expression> ) in <expression> //special form
    //without an else branch
    //The special form is an alternative way to express the following
    match_event ( <expression> ) in { <expression> ; {} } else {}
```

識別子にはイベントの名前のみを含めることができます。あるいは、そのイベントがデフォルトの名前空間にない場合、名前空間の名前の後にイベントの名前が続きます。

EB GUIDEスクリプトがパラメータ付きのイベントによってトリガーされた場合は、`match_event`式の`in`式でそれらのパラメータにアクセスできます。C言語で構造体のフィールドにアクセスする場合と同じように、ドット記法でパラメータから値を読み取ります。イベントのパラメータは、`else`式で使用できません。



#### Example 6.23.

##### イベントパラメータ

```
// assume that "mouseClick" has two parameters: x and y
match_event v:event = ev:mouseClick in
{
    dp:rectX = v:event.x;
    dp:rectY = v:event.y;
```



```
}
```

## 6.16.17. 文字列の書式設定

EB GUIDEスクリプトでは、文字列の書式設定に、連結演算子(+)やさまざまなデータ文字列変換関数を使います。EB GUIDEスクリプト標準ライブラリには、整数から文字列への簡易な変換を実行する`int2string`関数が付属します。



### Example 6.24.

#### 文字列の書式設定

```
// Assume this widget is a label and has a text property.  
// Further assume that the datapool item dp:time_hour and  
// dp:time_minute hold the current time.  
v:this.text = "the current time is: " + f:int2string(dp:time_hour)  
            + ":" + f:int2string(dp:time_minute);
```

## 6.16.18. 標準ライブラリ

EB GUIDEスクリプトには、例えば以下に示すような一連の外部関数で構成される標準ライブラリが付属します。

- ▶ 文字列の書式設定
- ▶ 言語管理
- ▶ トレース
- ▶ 時刻と日付
- ▶ 乱数生成

詳細については、[13.4.3「EB GUIDEスクリプト標準ライブラリ」](#)をご覧ください。

## 6.17. スクリプト値

スクリプト値は、ウィジェットプロパティまたはデータプールアイテムの値を、別の表記に変換したものです。この変換によって、ウィジェットプロパティまたはデータプールアイテムは、他のモデルの要素を使用して、自身の値を評価したり、イベントやプロパティの更新に反応したりしています。スクリプト値はEB GUIDEスクリプトのスクリプト言語で記述されます。

EB GUIDEのプロパティは、スクリプト値に変換することも、スクリプト値から元の値に戻すこともできます。

手順については、[9.8「プロパティのスクリプト値への変換」](#)をご覧ください。

スクリプト値を編集するには、EB GUIDE StudioのEB GUIDEスクリプトエディターを使用します。このエディターはいくつかに分割されています。

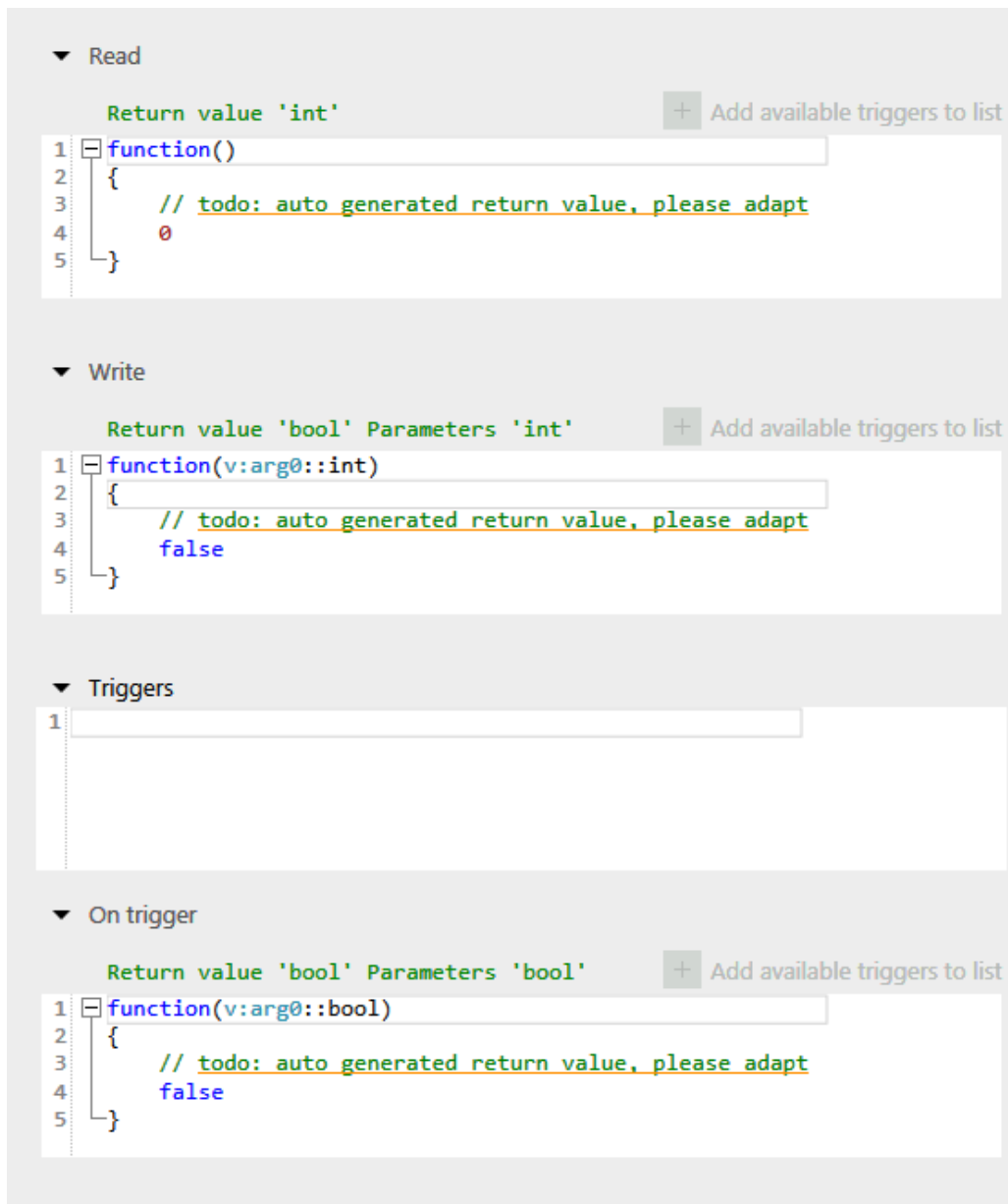


Figure 6.18. EB GUIDE StudioのEB GUIDEスクリプト

- ▶ **Read**スクリプトはスクリプト値のプロパティが読み取られると呼び出されます。プロパティがリストタイプの場合には、パラメータにリストインデックスが含まれます。

**Read**スクリプトの戻り値は、プロパティの現在の値を表します。

- ▶ **Write**スクリプトは、スクリプト値のプロパティの書き込み時に呼び出されます。

新しいプロパティ値は**Write** スクリプトのパラメータになります。プロパティがリストタイプの場合は、パラメータにリストインデックスが含まれます。

**Write**スクリプトの戻り値によって、プロパティの変更通知が次のように制御されます。

- ▶ true: 変更通知をトリガーする
- ▶ false: 変更通知をトリガーしない
- ▶ **Trigger**スクリプトには、**On trigger**スクリプトの実行をトリガーするイベント、データプールアイテム、およびウィジェットプロパティのリストが含まれます。

使用可能なすべてのトリガーをトリガースクリプトに追加をクリックすると、対応するスクリプトで強調表示されているすべてのトリガーがトリガースクリプトに追加されます。

- ▶ **On trigger**スクリプトはイベントをトリガーしたのち、またはプロパティを更新したのち、初期化時に呼び出されません。

**On trigger**スクリプトのパラメータは、スクリプト実行の原因を表します。実行は、初期化、または**Trigger**リストに含まれるいずれかのトリガーによって引き起こされます。

**On trigger**スクリプトの戻り値によって、プロパティの変更通知が次のように制御されます。

- ▶ true: 変更通知をトリガーする
- ▶ false: 変更通知をトリガーしない
- ▶ **Length**スクリプトは、リストタイプのプロパティでのみ使用できます。

**Length**スクリプトの戻り値は、リストの現在の長さを表します。

## 6.18. ショートカット、ボタン、アイコン

### 6.18.1. ショートカット

次の表に、EB GUIDE Studioで使用できるショートカットとその意味を示します。

Table 6.4. ショートカット

ショートカット	説明
<b>Ctrl+C</b>	選択内容をコピー
<b>Ctrl+F</b>	検索ボックスにジャンプ
<b>Ctrl+S</b>	保存
<b>Ctrl+V</b>	コピーした選択内容を貼り付け

ショートカット	説明
<b>Ctrl+Y</b>	やり直し
<b>Ctrl+Z</b>	元に戻す
<b>Ctrl+Enter</b>	トリガーフィルタボックスに新しいイベントを追加します。
<b>Ctrl+Shift+Insert</b>	名前空間コンポーネントで、新しい名前空間を子として既存の名前空間に追加します。
<b>Alt+F4</b>	アクティブなウィンドウを閉じる
<b>Shift+F1</b>	EB GUIDE TFのユーザーマニュアルを開く
<b>F1キー</b>	EB GUIDE Studioのユーザーマニュアルを開く
<b>Shift+F2</b>	データプールまたはイベントコンポーネント、および選択した要素が使用されているすべての場所(例えば、EB GUIDEスクリプト)の選択した要素の名前を変更します。データプールアイテムとイベントに使用できます。
<b>F2キー</b>	選択された要素の名前変更
<b>F3キー</b>	EB GUIDEモデル内における選択した要素の出現箇所をすべて検索
<b>F5キー</b>	シミュレーションの開始
<b>F6キー</b>	検証
<b>Deleteキー</b>	選択したモデル要素をコンテンツエリアまたはコンポーネントから削除
-	ナビゲーションまたは概要コンポーネント内の選択した要素を折りたたむ
*と+	ナビゲーションまたは概要コンポーネント内の選択したモデル要素を展開
上/下/左/右方向キー	コンテンツエリア内の選択したステートまたはウィジェットを上/下/左/右の方向に1ピクセル移動
<b>Ctrl</b> キーを押しながら左マウスボタンをクリック  <b>Shift</b> キーを押しながら、左マウスボタンをクリック、あるいは上矢印または下矢印キー	データプールおよびイベントコンポーネントで複数の要素を選択します。

## 6.18.2. コマンドラインオプション

### 6.18.2.1. のコマンドラインオプション `Studio.Console.exe`

次の表に、EB GUIDE Studioの`Studio.Console.exe`で利用できるコマンドラインオプションとその意味を示します。未定義のコマンドラインオプションは無視されます。

コマンドラインの一般的な構文は次のとおりです。

```
Studio.Console.exe <option> "project_name.ebguide"
```

Table 6.5. のコマンドラインオプション Studio.Console.exe

オプション	説明
-c <logfile dir>	EB GUIDEモデルを検証し、logfile dirで指定されたディレクトリにログファイルを書き込む
-e <destination dir>	EB GUIDEモデルを出力先ディレクトリにエクスポート destination dir コマンドラインオプション-pとともに使用。以下の例をご覧ください。
-h	ヘルプメッセージを表示
-l <language file>	language file (.xliff)として保存された1つの言語ファイルをEB GUIDEモデルにインポートし、ログファイルを作成します
-m	プロジェクトの移行を許可
-o	プロジェクトファイルを開く
-p <profile>	エクスポート中にprofileを指定されたプロファイルとして使用



#### Example 6.25. コマンドラインオプション

コマンドラインStudio.Console.exe -e "C:/temp/exported\_project" -p "target\_profile" -o "project\_name.ebguide"は、プロファイルtarget\_profileを使用してproject\_name.ebguideを指定された出力先ディレクトリC:/temp/exported\_projectにエクスポートします。

手順については、以下をご覧ください。

- ▶ [10.4.1.2「コマンドラインを使用したEB GUIDEモデルの検証」](#)
- ▶ [10.5.2「コマンドラインを使用したEB GUIDEモデルのエクスポート」](#)
- ▶ [10.8.2.2「コマンドラインを使用した言語依存テキストのインポート」](#)

### 6.18.2.2. のコマンドラインオプション Monitor.Console.exe

次の表に、EB GUIDE MonitorのMonitor.Console.exeで使用できるコマンドラインオプションとその意味を示します。未定義のコマンドラインオプションは無視されます。

コマンドラインの一般的な構文は次のとおりです。

```
Monitor.Console.exe <option> "monitor.cfg"
```

Table 6.6. のコマンドラインオプション Monitor.Console.exe

オプション	説明
-c <host:port>	EB GUIDEモデルを実行中のEB GUIDE GTFプロセスに接続
-h	ヘルプメッセージを表示
-l <language>	EB GUIDE Monitorの言語を en (英語)、ja(日本語)、ko(韓国語)、zh-cn(中国語)のいずれかに設定
-o	設定ファイルを開く monitor.cfg
-s	定義済みスクリプトのすべてのメソッドを実行



**Example 6.26.**  
コマンドラインオプション












コマンドラインMonitor.Console.exe -l koは、EB GUIDE Monitorの言語を韓国語に設定します。






EB GUIDE Monitorの使用方法については、[10.9「EB GUIDE Monitorを操作する」](#)をご覧ください。

### 6.18.3. ボタン

次の表に、EB GUIDE StudioおよびEB GUIDE Monitorで使用されているボタンとその意味を示します。

Table 6.7. EB GUIDE Studioのボタン

ボタン	説明
	元に戻す
	やり直し
	保存
	プロジェクトを検証
	シミュレーションを開始
	シミュレーションを停止
	プロジェクトセンターを開く
	EB GUIDEスクリプトエディターを開きます。
	コンテンツエリアとナビゲーションコンポーネントの同期を取る
	イベント、データプールアイテム、またはステートマシンを追加
	プロパティに関連付けられたコンテキストメニューを開く  ボタンの色の意味は次のとおりです。

ボタン	説明
	<ul style="list-style-type: none"> <li> プロパティがローカルである</li> <li> プロパティが別のプロパティにリンクされている</li> <li> プロパティがデータプールアイテムにリンクされている</li> <li> プロパティ値テンプレート値と等しい</li> </ul>
	イベントを発行する

## 6.18.4. アイコン

次の表に、EB GUIDE Studioで使用されているアイコンとその意味を示します。

Table 6.8. EB GUIDE Studioのアイコン

アイコン	説明
	ビューステートまたはビューテンプレートの終了アニメーションを示す
	ビューステートまたはビューテンプレートの開始アニメーションを示す
	ビューステートまたはビューテンプレートの変更アニメーションを示す
	ステートマシンまたはステートのエントリーアクションを示す
	ステートマシンまたはステートの終了アクションを示す
	エントリーアクションまたは終了アクションを削除するコンテキストメニューを開く
	動的ステートマシンリストが有効になっていることを示す
	テンプレートを示す
	遷移を示す
	内部遷移を示す
	ウィジェットテンプレート: プロパティがウィジェットテンプレートインターフェースに追加されていることを示す
	ダイアログを開いてマスターイメージを選択する
	EB GUIDEスクリプトエディターで関数を示す
	EB GUIDEスクリプトエディターでキーワードを示す
	EB GUIDEスクリプトエディターで変数を示す

## 6.19. スキン

スキンを使用すると、同じEB GUIDEモデルに異なるデータプール値を定義することによって、異なるヒューマンマシンインターフェースを定義できます。このようにして、同じヒューマンマシンインターフェースにさまざまな外観(夜用モードと昼用モードのスキンなど)を定義できます。

ランタイム中にスキンを切り替えると、異なるデータプール値の効果を確認できます。

スキンのサポートは標準のデータプール値にのみ使用でき、スクリプト値またはリンクされたデータプールアイテムには使用できません。

注記



言語サポートは使用できません

データプールアイテムにスキンのサポートを定義した場合、同じアイテムに言語サポートを追加することはできません。

手順については、[8.5「スキンのサポートの操作」](#)をご覧ください。

## 6.20. ステートマシンとステート

NOTE



ステートマシンとステートの背景色の変更

EB GUIDE Studioでは、次の背景色を変更できます。

- ▶ メインステートマシン
- ▶ ビューステート
- ▶ 混合ステート

背景色を変更するには、プロパティコンポーネントのBackground colorドロップダウンリストボックスから色を選択します。

### 6.20.1. ステートマシン

ステートマシンは、決定性有限オートマトンであり、システムの動的な動作を表します。EB GUIDEのステートマシンは、階層的に順序付けられた任意の数のステートと、ステート間の遷移で構成されます。

EB GUIDEでは、次のタイプのステートマシンを作成できます。

#### 6.20.1.1. ハプティックステートマシン

ハプティックステートマシンを使うと、GUIの仕様を提供できます。



### 6.20.1.2. ロジックステートマシン

ロジックステートマシンを使うと、GUIなしでロジックを指定できます。

### 6.20.1.3. 動的ステートマシン

動的ステートマシンは、他のステートマシンと並行して動作します。

動的ステートマシンは、システムの起動時に自動的に開始されません。動的ステートマシンの開始と停止は、別のステートマシンから実行されます。

動的ステートマシンには、次の2種類があります。

- ▶ ハプティック動的ステートマシン
- ▶ ロジック動的ステートマシン

手順については、[12.1「チュートリアル: 動的ステートマシンの追加」](#)をご覧ください。

## 6.20.2. ステート

EB GUIDE では、ステートという概念を使用します。ステートは、ステートマシンのステータスと動作を決定します。ステートは遷移でリンクされます。遷移は、ステート間の接続であり、ソースステートからターゲットステートへのステート変更を定義します。

ステートには次のプロパティがあります。

- ▶ エントリーアクション
- ▶ 終了アクション
- ▶ 内部遷移

### 6.20.2.1. 混合ステート

混合ステートは、内部に他のステートを子ステートとして持つことができます。混合ステートは階層構造で、任意の数の子ステートを作成できます。どのタイプのステートでも混合ステートに入れ子にすることができます。

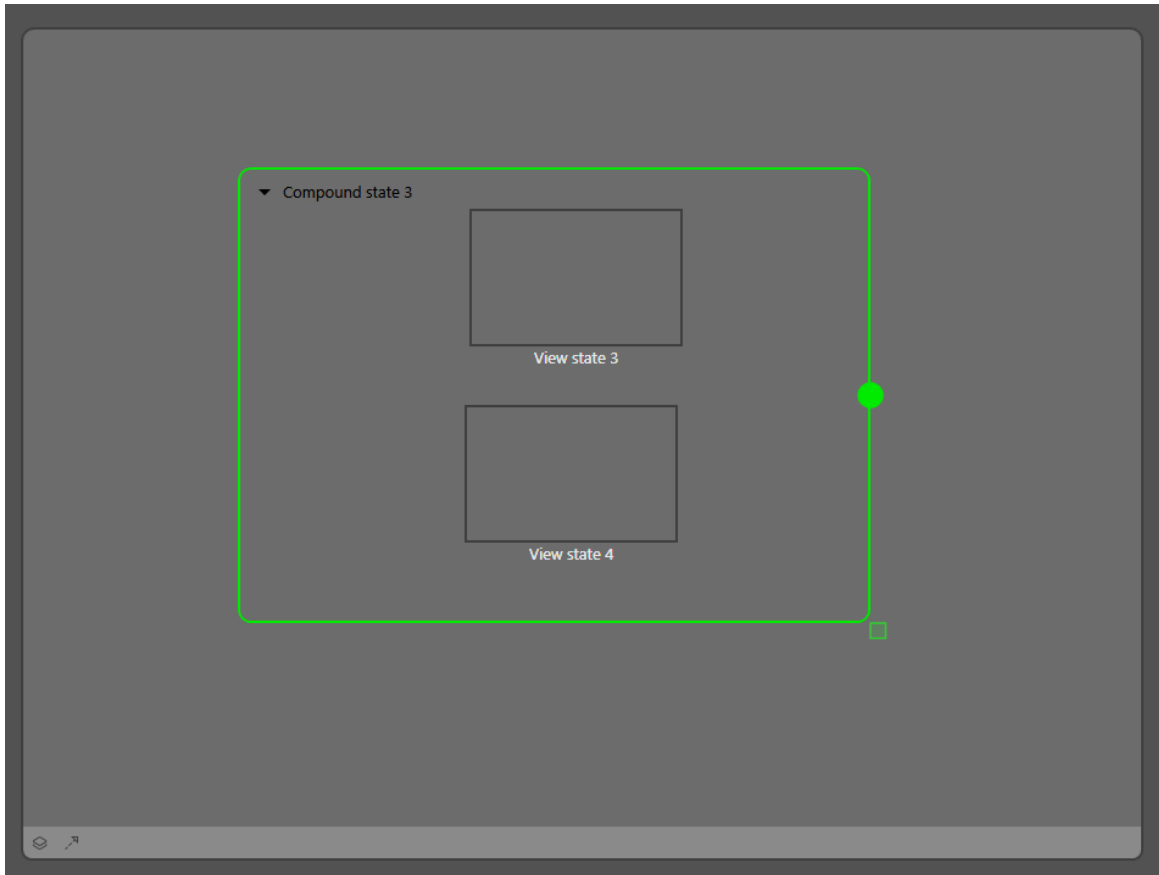


図6.19 混合状態

[ナビゲーション]コンポーネントには、状態の階層がツリー構造で表示されます。

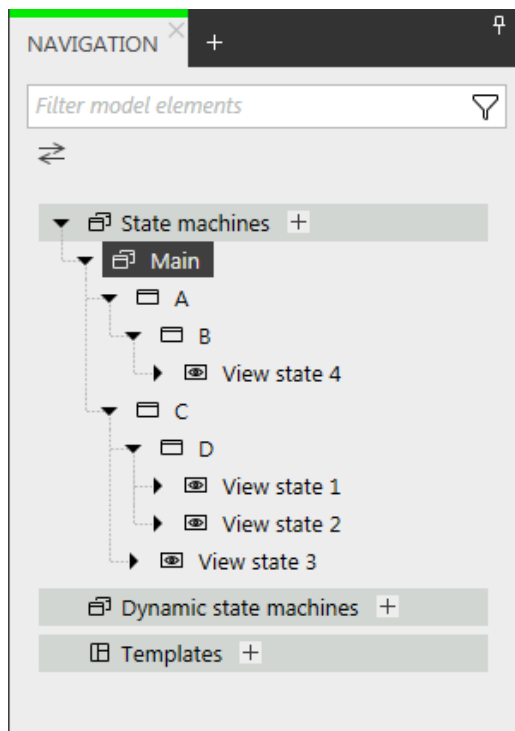


図6.20 ツリー構造で表示された状態階層

混合状態には、入力または出力遷移、および内部遷移をいくつでも作成できます。子状態は親状態の遷移を継承します。

### 6.20.2.2. ビュー状態

ビュー状態はビューを格納します。ビューはプロジェクト固有のヒューマンマシンインターフェース画面を提供します。ビューは、対応するビュー状態がアクティブになっている場合に表示されます。ビューは、ユーザーとシステムを結ぶインターフェースであるウィジェットから構成されます。

### 6.20.2.3. 初期状態

初期状態は、状態マシンのスタート地点となります。初期状態には、最初の状態を指すデフォルト遷移があります。初期状態へエントリーする遷移はありません。

初期状態は、混合状態のスタート地点として使うことができます。混合状態へは次の方法でエントリーします。

- ▶ 混合状態への遷移を使う(初期状態必須)
- ▶ 混合状態の子状態への遷移を使う

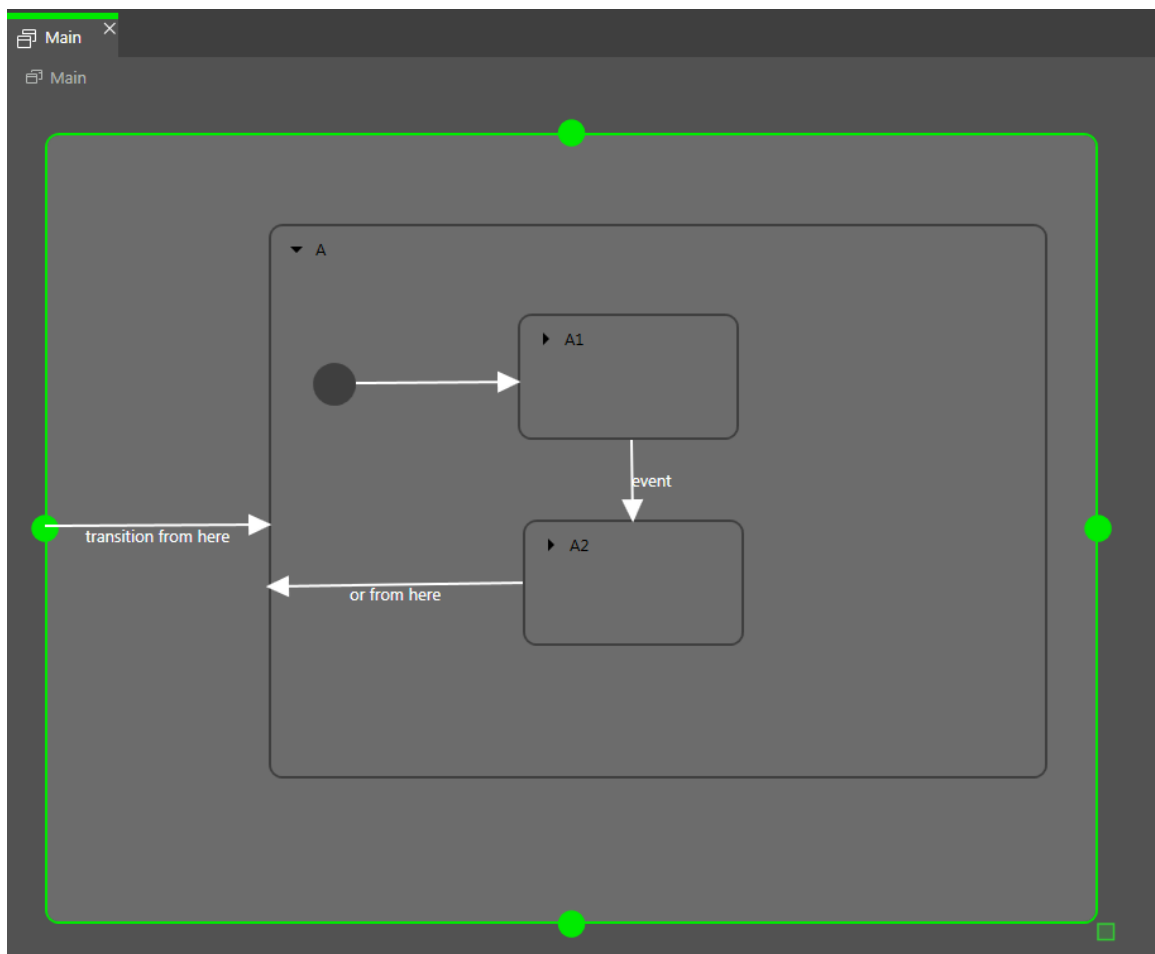


図6.21 初期状態の例

#### 6.20.2.4. 最終状態

最終状態は混合状態を終了するために使用します。状態マシンの最終状態へエントリーすると、状態マシンの動作は終了します。混合状態内の履歴状態はリセットされます。最終状態から他の状態へ向かう遷移はありません。

混合状態には最終状態を1つだけ作成できます。最終状態がトリガーされるのは、以下の状況です。

- ▶ 子状態から混合状態の外部への遷移が発生した(イベントZによる遷移)
- ▶ 混合状態からの出力遷移が発生した(イベントYによる遷移)
- ▶ 混合状態内で最終状態への遷移が発生した(イベントXによる遷移)

最終状態を含む混合状態には、出力遷移が必要です。

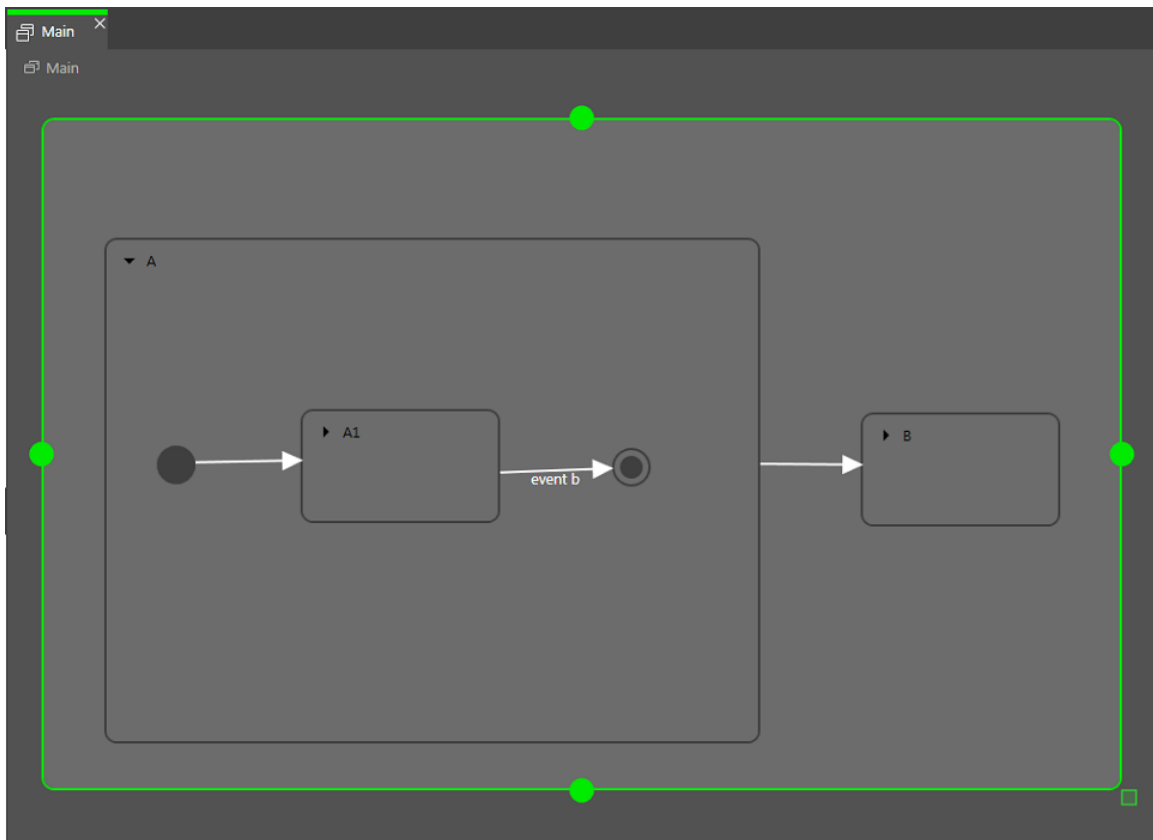


図6.22 混合状態内での最終状態の使用方法

### 6.20.2.5. 選択状態

選択状態は、動的な条件分岐を実現します。条件に基づいてイベントを発行する場合に使用します。選択状態は、遷移元の状態と遷移先の状態を接続するものです。1つの選択状態に複数の入出力遷移を設定できます。外へ向かう遷移には条件が与えられ、その条件が`true`と評価されない限り、遷移は実行されません。出力遷移の1つに`else`遷移があります。他のすべての条件が`false`と評価されると、この遷移が実行されます。`else`遷移は必須です。

外へ向かう複数の遷移が`true`と評価されることがあるため、遷移を評価する順序を定義する必要があります。

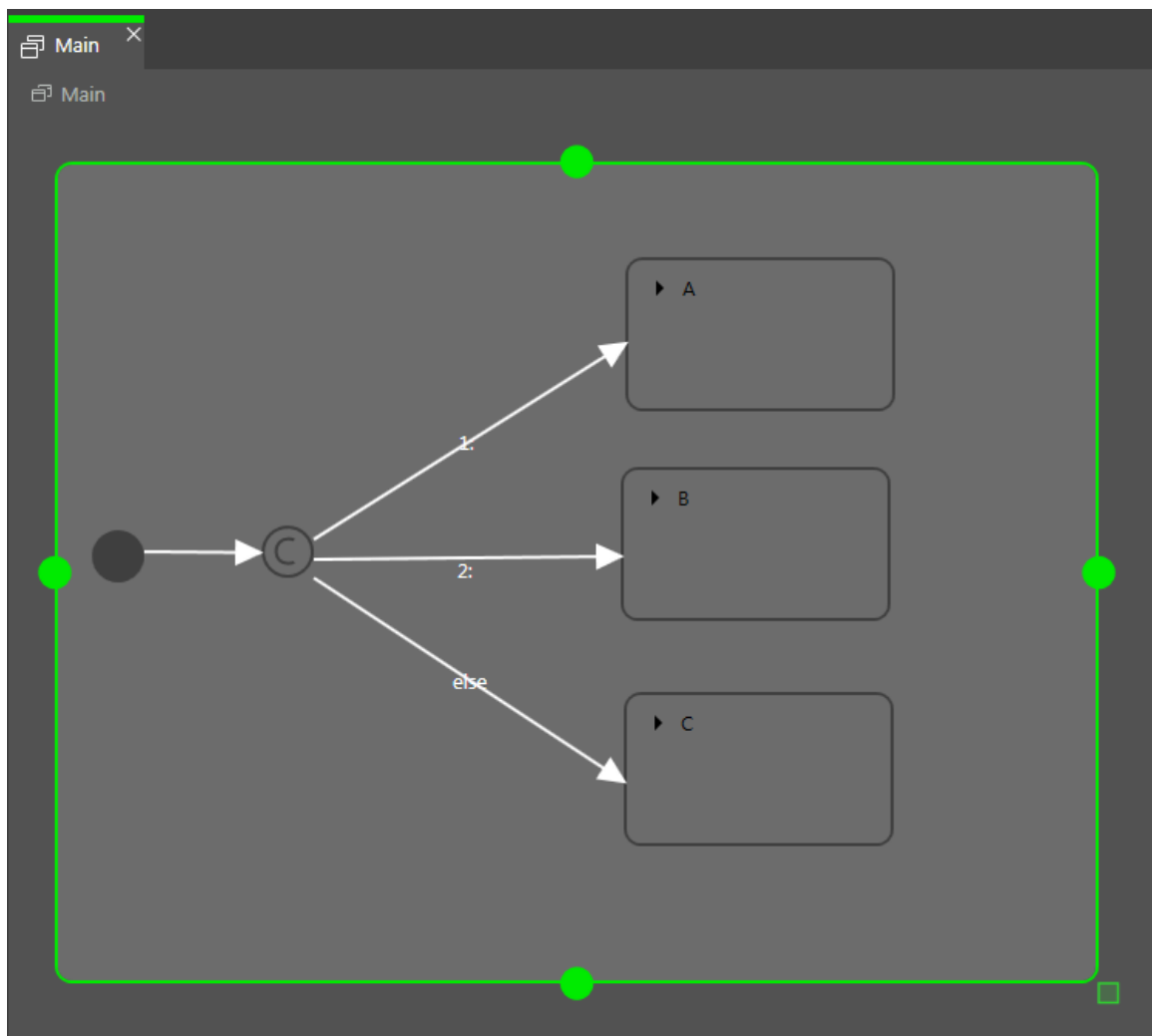


図6.23 入力遷移と出力遷移を持つ選択状態

### 6.20.2.6. 履歴状態

EB GUIDE には、2種類の履歴状態がサポートされています。

- ▶ 浅い履歴状態には、最新のアクティブなサブ状態(混合状態を終了する直前にアクティブだったサブ状態)が保存されます。
- ▶ 深い履歴状態には、混合状態と、混合状態を終了する直前のそのサブ階層全体が保存されます。

履歴状態の親状態へ初めてエントリーしたときに、最後にアクティブだった子状態が復帰されます。

浅い履歴状態は、混合状態を終了する直前にアクティブだった状態だけを覚えています。状態の階層は記憶できません。

浅い履歴ステートは、混合ステート内に記録された最後のアクティブステートを復帰します。この履歴ステートには、外へ向かう無条件のデフォルト遷移があるほか、複数のエントリー遷移を指定できます。

混合ステートへ初めてエントリーする時点では、浅い履歴ステートは空です。空の浅い履歴ステートへエントリーすると、そのステートのデフォルト遷移によって次のステートが決定されます。



### 例6.27 浅い履歴ステート

浅い履歴ステートは次のように使用できます。

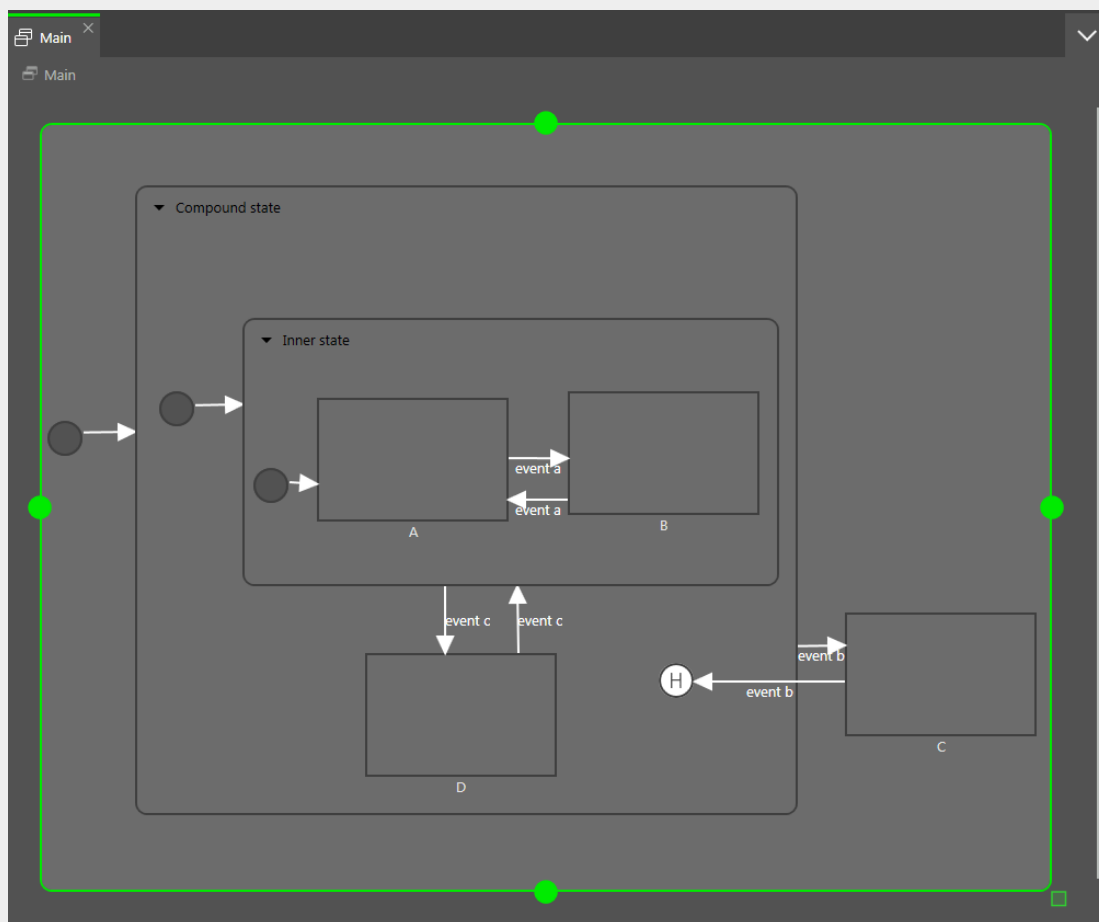


図6.24 浅い履歴ステート

- ▶ ケース1: アクティブなステートはDです。
  1. event b が発行され、ステートCにエントリーします。
  2. event b が再び発行され、浅い履歴ステートにエントリーします。
  3. ステートマシンは、浅い履歴ステートからステートDにエントリーします。ステートDがCompound State内の最後のアクティブステートだったからです。
- ▶ ケース2: アクティブなステートはBです。

1. event b が発行され、状態Cにエントリーします。
2. event b が再び発行され、浅い履歴状態にエントリーします。
3. 状態マシンは、浅い履歴状態からInner stateにエントリーします。浅い履歴状態は、最後にアクティブだった状態を記憶していますが、階層を記憶できないからです。
4. Inner stateにエントリーすると状態Aに遷移します。

深い履歴状態では階層履歴も記録されます。



### 例6.28 深い履歴状態

深い履歴状態は次のように使用できます。

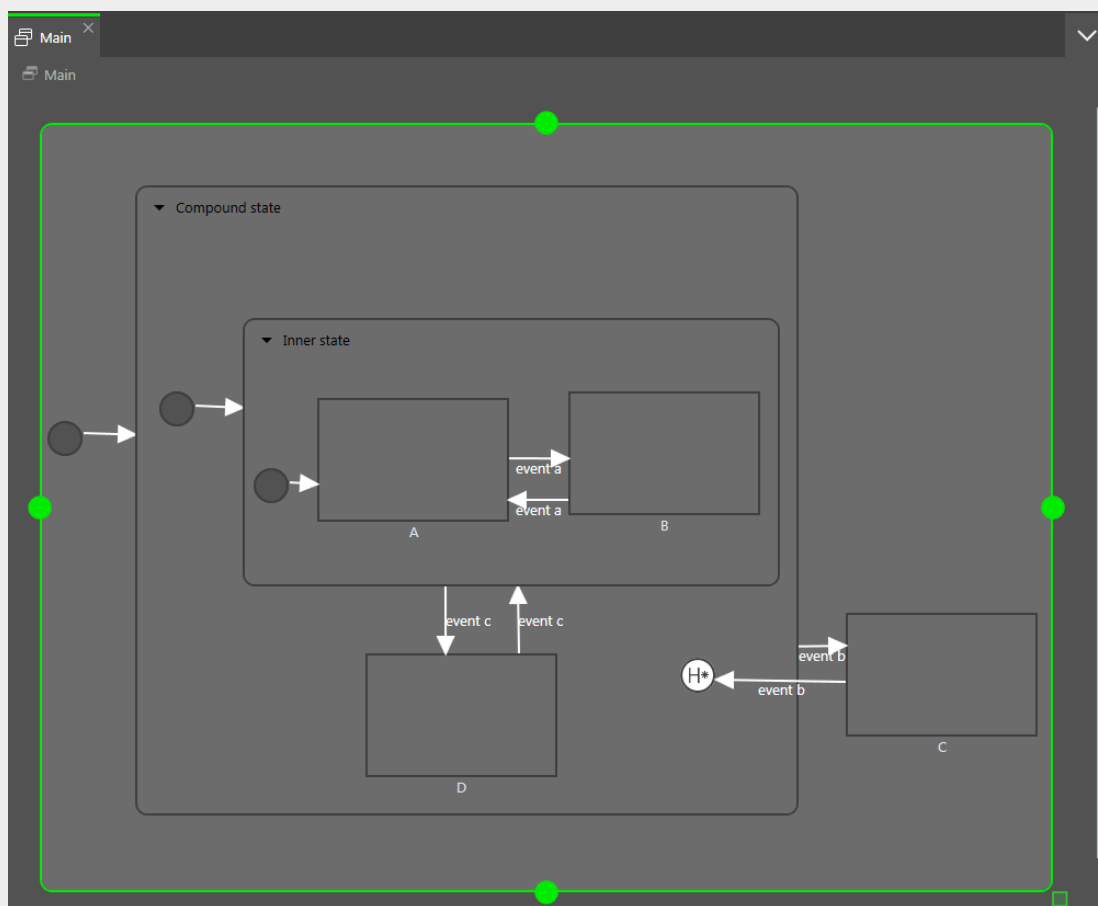


図6.25 深い履歴状態

- ▶ ケース1: アクティブな状態はDです。
1. event b が発行され、状態Cにエントリーします。
  2. event b が再び発行され、深い履歴状態にエントリーします。



3. ステートマシンは、深い履歴ステートからステートDにエントリーします。ステートDがCompound State内の最後のアクティブステートだったからです。

▶ ケース2: アクティブなステートはBです。

1. event b が発行され、ステートCにエントリーします。

2. event b が再び発行され、深い履歴ステートにエントリーします。

3. ステートマシンは、深い履歴ステートからステートBにエントリーします。ステートBが最後のアクティブステートであり、深い履歴ステートはステート階層を記憶しているからです。

ステートには浅い履歴ステートと深い履歴ステートのどちらかを設定できます。親ステートに履歴ステートを設定した場合も、その子ステートに別の履歴ステートを設定できます。

### 6.20.3. 遷移

遷移は、ソースステートとターゲットステートを直接結ぶ関連付けです。ステートマシンを現在のステートから別のステートへと導きます。遷移には次のプロパティがあります。

▶ 遷移を実行するトリガー

トリガーはイベント、またはデータプールアイテムの変更です。

▶ 遷移を実行するためにtrueと評価される必要がある条件

▶ 遷移時に実行されるアクション

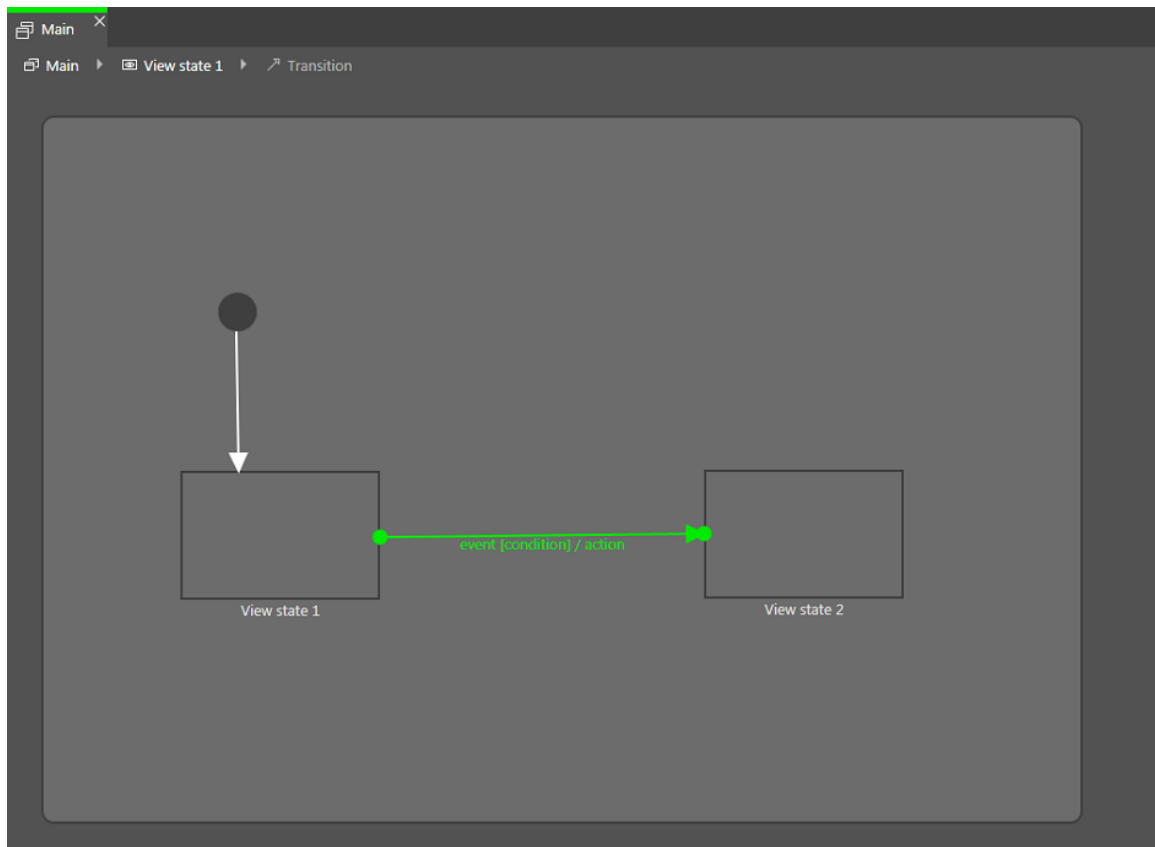


Figure 6.26. 遷移

**NOTE**



**遷移の決定性**

特定のソースステートから同じイベントに複数の遷移を設定することは、たとえ条件に違いがあっても不可能です。複数の条件を使ってステートマシンを複数の遷移先ステートにジャンプさせたい場合は、選択ステートを使ってください。

ステートは親ステートからすべての遷移を継承します。複数のステートが別のステートへの遷移を共有する場合、混合ステートを使って遷移をまとめると条件の数を減らせます。



**Example 6.29.**

### 遷移の継承

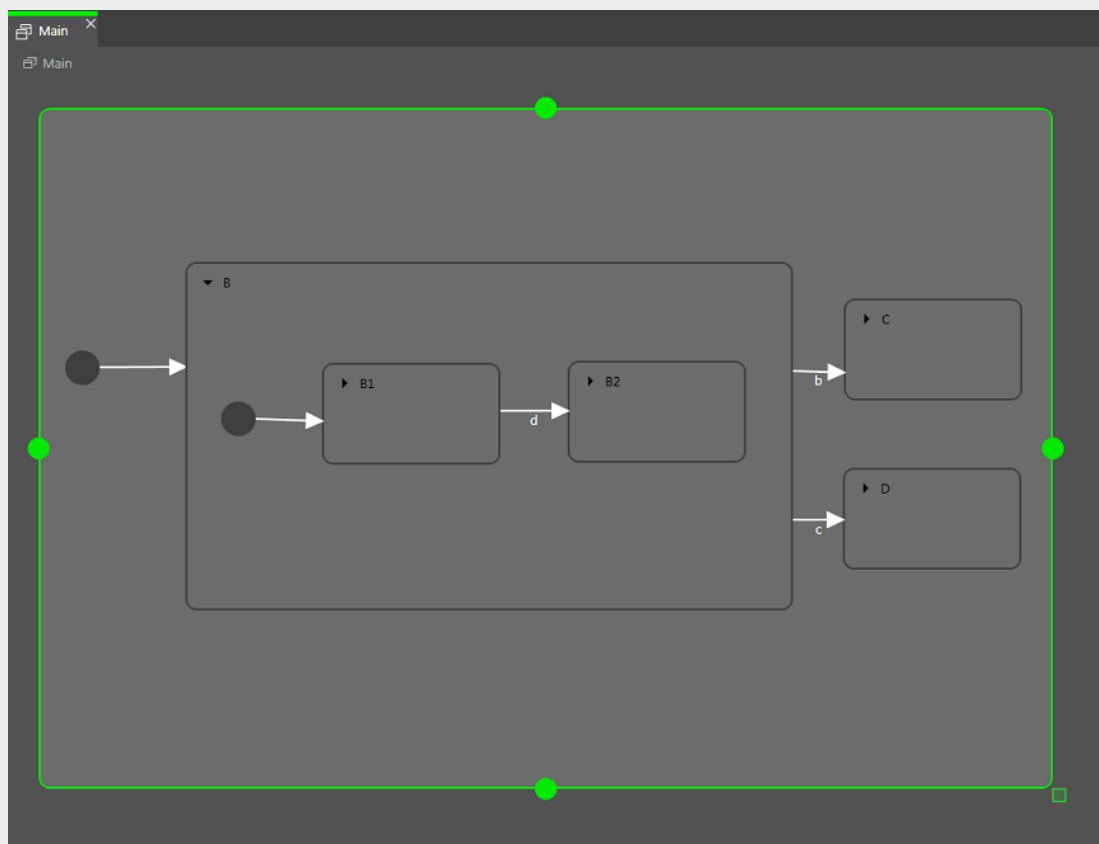


Figure 6.27. 遷移の継承

イベントbの発行時にステートマシンがState B1にある場合、State Cへの遷移が実行されます。子ステートState B1とState B2が、ステートState Bの遷移を継承するためです。

子ステートからの内部遷移に使われるイベントが、親ステートからの外部遷移のイベントと同じ場合、遷移の継承がオーバーライドされます。



#### Example 6.30.

### 遷移のオーバーライド

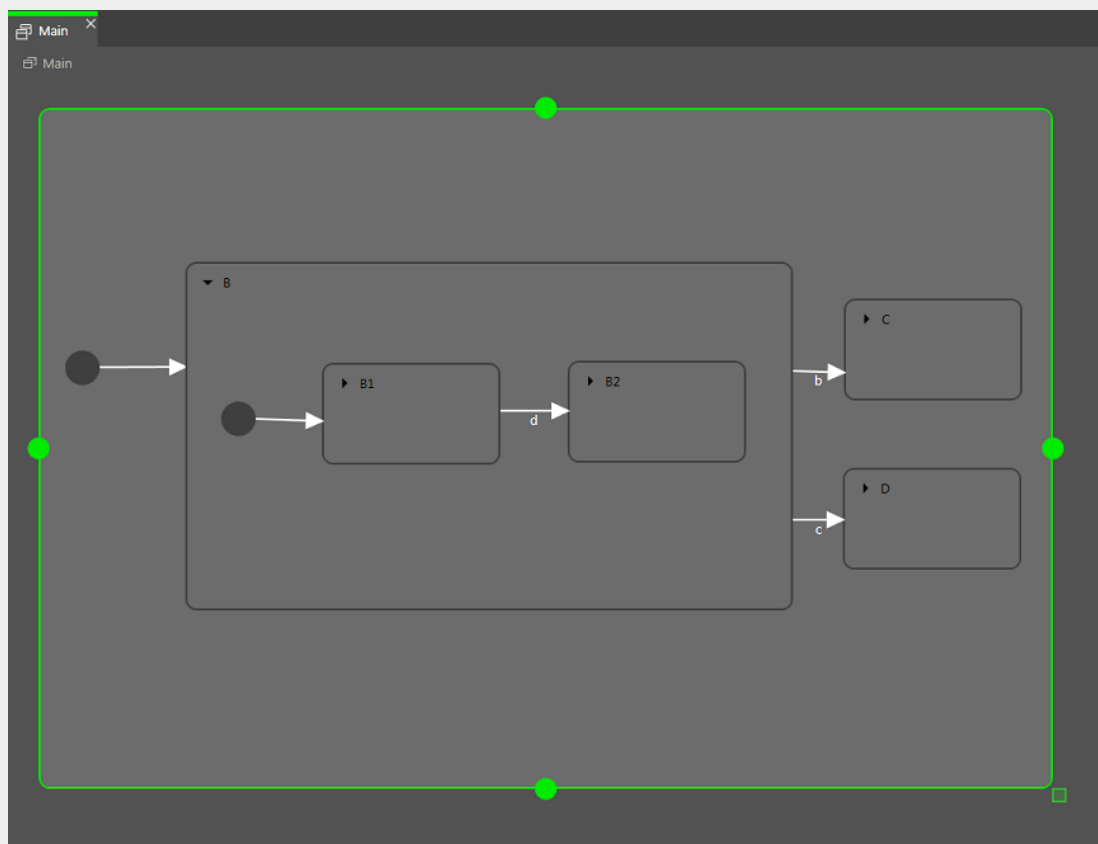


Figure 6.28. 遷移のオーバーライド

イベントdの発行時にステートマシンがState Bのステートにある場合、State Cへの遷移が実行されます。

イベントdの発行時にステートマシンがState B1のステートにある場合、State B2への遷移が実行されず(State Cには遷移しません)。2つの遷移の名前が同一であるため、内側の遷移が外側の遷移よりも優先されます。

#### NOTE

##### 実行の階層



ステートマシンでは、同じイベントを使う遷移の実行階層は、常に内側から外側へ向かいます。つまり、内部遷移が外部遷移より優先されるということです。

遷移には次のタイプがあります。

#### ▶ デフォルト遷移

デフォルト遷移は自動的にトリガーされます。イベントやデータプールアイテムの更新ではトリガーされません。条件はありませんが、アクションを指定できます。この遷移は、初期ステート、最終ステート、選択ステート、履歴ステートで使用されます。

▶ 選択遷移

選択遷移は、与えられた条件によって外部へ遷移します。ソース状態は選択状態です。選択遷移は、条件の評価によってトリガーされます。その結果としてアクションが実行されます。条件が`true`と評価された最初の選択遷移が実行されます。

▶ else遷移

else遷移は、選択遷移と対になる必須の遷移です。どの選択状態にも1つのelse遷移が必要であり、すべての選択遷移の条件が`false`と評価された場合にelse遷移が実行されます。

▶ 内部遷移

内部遷移は、ターゲット状態を持たない遷移です。したがって、アクティブ状態を変更しません。内部遷移の目的は、現在の状態に留まったままイベントに反応することです。この遷移に条件を指定し、その結果としてアクションを実行できます。

状態の同じイベントに対して複数の内部遷移を指定できます。実行の順序を定義します。

▶ 自己遷移

自己遷移は、ソース状態とターゲット状態が同じ状態である遷移です。内部遷移とは違い、自己遷移は状態を脱してから同じ状態へ再度エントリーします。したがって、エントリーアクションと終了アクションが実行されます。

## 6.20.4. ステートマシンの実行

ステートマシンが実行されると、実行中は常に1つのアクティブ状態が存在します。ステートマシンはイベント主導型です。

ステートマシンの動作サイクルは、次のようなものです。

1. ステートマシンは、初期状態へエントリーすると開始されます。
2. ステートマシンは、イベントの発生を待ちます。
  - a. 内部遷移が見つかります。
    - i. 現在の状態を出発点に、現在のイベントでトリガーされ、条件が`true`と評価される最初の内部遷移を探します。該当する遷移が見つかったら、それが実行されます。
    - ii. 遷移が見つからない場合は、親状態に移動し、現在のイベントでトリガーされ、条件が`true`と評価される最初の内部遷移を探します。
    - iii. 遷移が見つからなければ、最上位の状態に達するまで前のステップを繰り返します。
  - b. 内部遷移が処理されます。

内部遷移を実行すると、それに関連付けられたアクションのトリガーだけが行われます。状態の終了と再エントリーは行われません。

- c. 遷移が見つかります。
  - i. 現在の状態を出発点に、現在のイベントでトリガーされ、条件が`true`と評価される最初の遷移を探します。該当する遷移が見つかったら、それが実行されます。
  - ii. 遷移が見つからなければ、親状態に移動し、遷移を探します。
  - iii. 条件を満たす遷移が見つかるまで、前のステップを繰り返します。
- d. 遷移が処理されます。

遷移を実行すると、状態マシンは現在の状態から別の状態に変わります。ソース状態が終了し、ターゲット状態へエントリーします。

遷移が実行されるのは、対応するイベントが発生し、条件が`true`と評価された場合だけです。

遷移は、状態階層内の複数の混合状態を終了またはエントリーできます。終了とエントリーの段階的な処理に挟まれる形で遷移のアクションが実行されます。

状態へエントリーするには、後続の遷移が必要です。例えば、混合状態へエントリーする場合、初期状態の遷移を後続の遷移として実行する必要があります。後続の遷移を連続して複数実行することが可能です。

- 3. 最終状態に達した時点で、状態マシンは停止します。

遷移が状態階層内で複数の状態を横断して使われる場合、段階的な終了とエントリーのアクションが実行されます。



#### 例6.31

### 遷移の実行

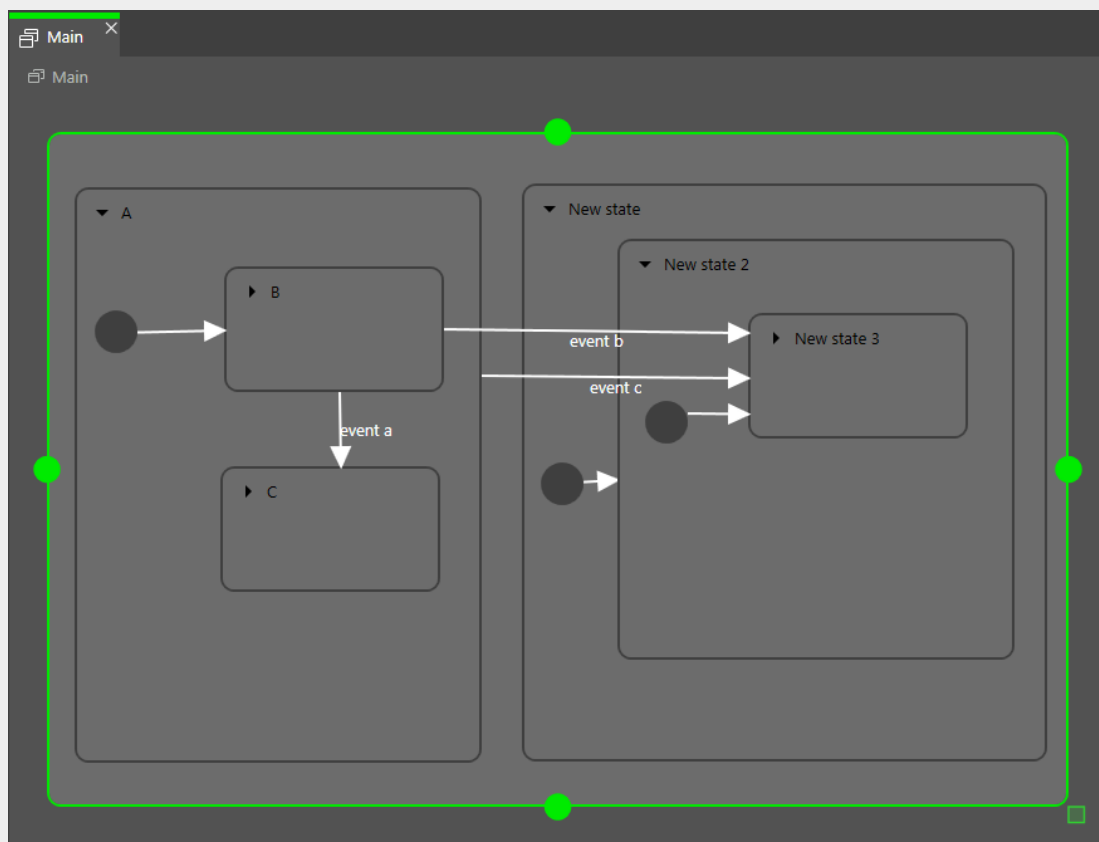


図6.29 遷移の実行

event aが発行されると、次のことが生じます。

1. ステートBが終了します。
2. ステートCが開始します。

event bが発行されると、次のことが生じます。

1. ステートBが終了します。
2. ステートAが終了します。
3. ステートNew stateが開始します。
4. ステートNew state 2が開始します。
5. ステートNew state 3が開始します。

event cが発行されると、次のことが生じます。

1. ステートBまたはステートCがアクティブの場合は、ステートBまたはステートCが終了します。
2. ステートAが終了します。

3. ステートNew stateが開始します。
4. ステートNew state 2が開始します。
5. ステートNew state 3が開始します。



### 例6.32 遷移の実行

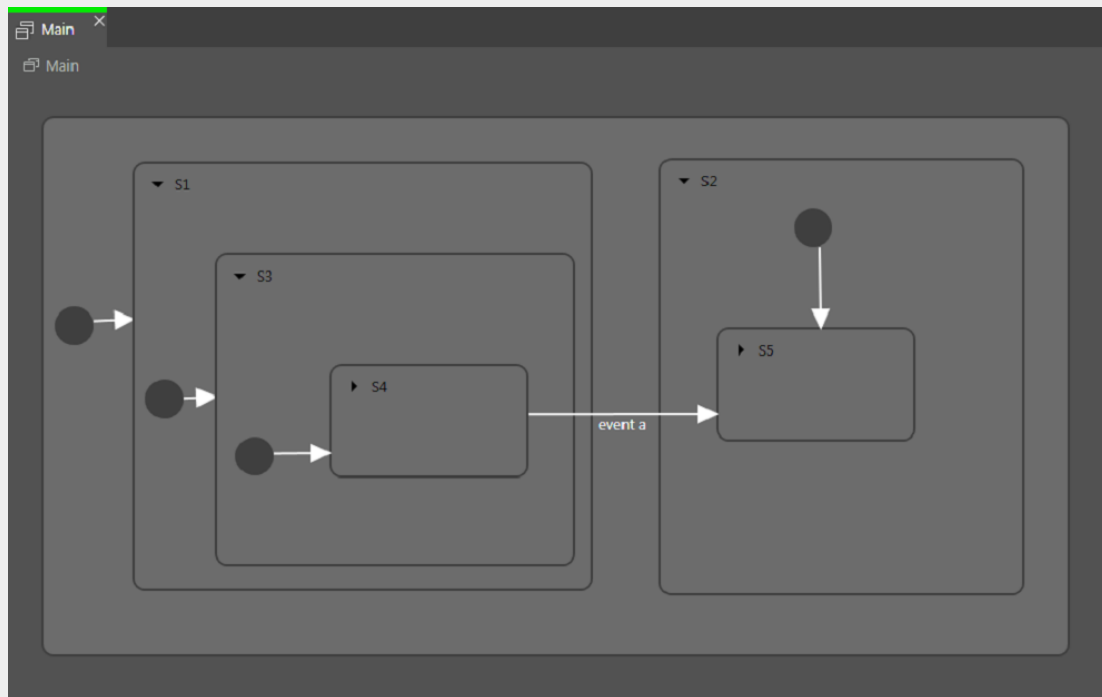


図6.30 遷移の実行

event aによって遷移がトリガーされた場合、次のことが生じます。

1. ステートS4が終了します。
2. ステートS3が終了します。
3. ステートS1が終了します。
4. ステートS2が開始します。
5. ステートS5が開始します。



### 例6.33



### 遷移の実行

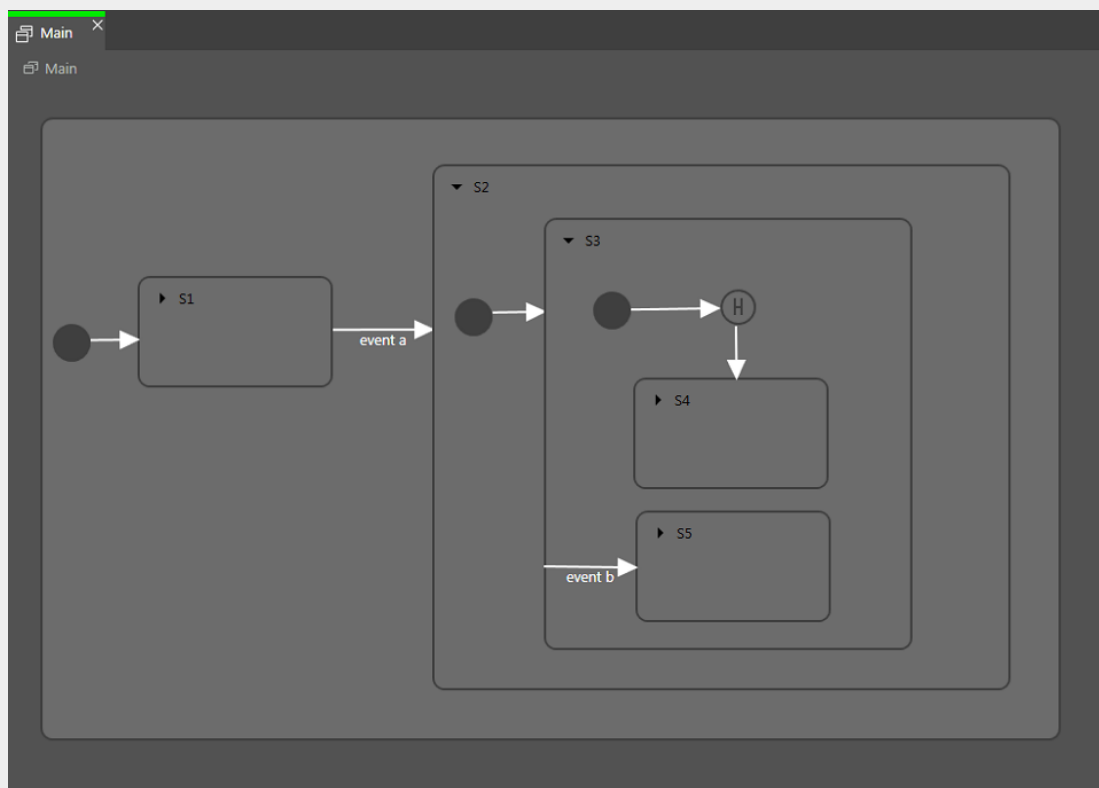


図6.31 遷移の実行

event aによって遷移がトリガーされると、次の遷移シーケンスが生じます。

1. ステートマシンはステートS2に移動します。
2. デフォルト遷移によってステートS3に遷移します。
3. 次のデフォルト遷移によって浅い履歴ステートにエントリーします。
4. 浅い履歴ステートは、ステートS3の最後のアクティブステートである、ステートS4またはステートS5を復帰します。

それぞれのステップで、開始-終了カスケードが個別に実行されます。

## 6.20.5. EB GUIDE の記法とUML記法の比較

ここではEB GUIDE記法を統一モデリング言語(Unified Modeling Language; UML) 2.5記法と比較して紹介します。

### 6.20.5.1. サポートされている要素

以下の表に、EB GUIDEでサポートされているすべてのUML 2.5要素を示します。一部の要素はUML 2.5の命名規則に従っていませんが、機能はまったく同じです。

での名前 EB GUIDE	UML 2.5での名前
初期ステート	初期(仮ステート)
最終ステート	最終ステート
混合ステート	ステート
選択ステート	選択(仮ステート)
深い履歴ステート	深い履歴(仮ステート)
浅い履歴ステート	浅い履歴(仮ステート)
内部遷移	内部遷移
遷移	外部/ローカル遷移 <sup>a</sup>

<sup>a</sup>EB GUIDE では、外部遷移とローカル遷移を区別しません。

### 6.20.5.2. サポートされない要素

以下のUML 2.5要素はEB GUIDEでサポートされません。

- ▶ ジョイン
- ▶ フォーク
- ▶ 交差
- ▶ エントリーポイント
- ▶ イグジットポイント
- ▶ 停止

### 6.20.5.3. 偏差

UML 2.5記法の一部の要素はEB GUIDEに実装されていません。ただし、それらの機能はEB GUIDEの概念によってモデリングされています。

UML 2.5での概念	を使用した回避策 EB GUIDE
並行ステート	この概念は、動的ステートマシンを使って実装されています。
遷移ごとのトリガー数	この概念は、EB GUIDEスクリプトを使ってデータプールアイテムまたはビューとして実装されています。

UML 2.5での概念	を使用した回避策 EB GUIDE
遷移時のタイムトリガー	この概念は、EB GUIDEスクリプト( <code>fire_delayed</code> )を使ってステートマシン、データプールアイテム、遷移、またはビューとして実装されています。

## 6.21. タッチ入力

EB GUIDEでは、2種類のタッチ入力がサポートされています。タッチジェスチャーとマルチタッチ入力です。

各タッチジェスチャーは、EB GUIDE Studioにおいてウィジェット機能として表されます。ウィジェット機能を有効にすると、一連のプロパティがウィジェットに追加されます。

ジェスチャーは、次の2つの基本タイプに分類されます。

- ▶ 非パスジェスチャー
- ▶ パスジェスチャー

### 6.21.1. 非パスジェスチャー

EB GUIDEには、以下の非パスジェスチャーが実装されています。

- ▶ フリック
- ▶ ピンチ
- ▶ 回転
- ▶ ホールド
- ▶ ロングホールド

パスジェスチャーには、マルチタッチジェスチャーとシングルタッチジェスチャーがあります。マルチタッチジェスチャーには、マルチタッチ入力をサポートする入力デバイスが必要です。シングルタッチジェスチャーには、サポートされている任意の入力デバイスが使用できます。

各ジェスチャーは、互いに独立して反応します。複数のジェスチャーが有効である場合、モデラーはEB GUIDEの動作の一貫性を保証する必要があります。

### 6.21.2. パスジェスチャー

パスジェスチャーとは、タッチスクリーン上に指で描画されたか、その他の入力デバイスによって入力された形状のことです。ウィジェットに対してウィジェット機能が有効である場合、ユーザーはそのウィジェット上で開始する形状を入力できます。設定可能な最小矩形よりも大きい形状でなければ、パスジェスチャーとしての認識対象にはなりません。入力形状が既知の形状集合とマッチングされ、マッチすればジェスチャーとして認識されます。

手順については、[12.3「チュートリアル:パスジェスチャーをモデル化する」](#)をご覧ください。

### 6.21.3. 入力処理とジェスチャー

ジェスチャー認識は、通常の入力処理と並行して実行されます。各ジェスチャーは、ジェスチャー関連のコンタクトを通常の入力処理から除外するように要求できます。ジェスチャーがコンタクトの除外を要求するタイミングは、実際のジェスチャーによって異なり、一部のジェスチャーに対してはこれを設定することが可能です。

コンタクトの除外は、ジェスチャーを行う指に対してのみ適用されます。コンタクトが除外されると、そのコンタクトに対するリリースイベントが受信されるまで、通常の入力処理においてそれが無視されます。つまり、近接性をサポートしないタッチスクリーン上では、除外されたコンタクトによってその他のタッチ反応がトリガーされることはありません。

#### TIP



#### 通常の入力処理からのコンタクトの除外

ボタンとウィジェット機能を備えるウィンドウに対して、ジェスチャーを行う場合を考えます。コンタクトがジェスチャー関連のものである場合、そのコンタクトがボタン上でリリースされたとしても、それによってボタンに関連付けられたアクションがトリガーされることがあってはいけません。

### 6.21.4. マルチタッチ入力

EB GUIDEでは、互換性のあるマルチタッチ入力デバイスが使用されれば、マルチタッチ入力を処理することができます。

マルチタッチとは、入力デバイスの2点以上のコンタクトを画面上で認識してトラッキングする機能のことです。典型的な例としては、タッチスクリーンを複数の指でタッチする場合があります。

#### ▶ マルチタッチイベントの処理

マルチタッチイベントは、マウスやシングルタッチのタッチスクリーンからのイベントと同じように、タッチイベントのメカニズムを使用してディスパッチされます。唯一の相違点は、各コンタクトが互いに独立してタッチ反応をトリガーすることです。個々のコンタクトを区別できるように、各タッチ反応にfingeridというパラメータが与えられます。

#### ▶ Finger ID

入力デバイスによってトラッキングされる各コンタクトには、それを識別する番号が割り当てられます。この識別子はfingeridと呼ばれ、入力デバイスごとに一意です。ただし、もう使用されていない同じ値が、後で別のコンタクトに割り当てられることはあります。

マルチタッチ入力が有効である場合に、エンドユーザーが入力可能なその他のタッチ操作シーケンスについて説明します。そのようなシーケンスとしては、以下のものがあります。

- ▶ エンドユーザーは、リストをスクロールしながらボタンを押すなど、インターフェースの複数の要素を同時に操作することができます。

- ▶ エンドユーザーは、1つのウィジェット上に複数の指を置くことができます。

これが生じる2つの典型的な操作は、スクロールとドラッグです。fingeridを使用することで、これらの操作を正しく処理できます。求められる動作によって、考えられるソリューションとしては以下のようなものがあります。

- ▶ ウィジェットを最初に押した指のみに、スクロールとドラッグの操作を許可する。
- ▶ 必ず最後にウィジェット上に置かれた指によって、スクロールとドラッグの操作を行う。上の方法を少し変更するだけで、簡単にこれが実現できます。

## 6.22. ウィジェット

ウィジェットは、EB GUIDEモデルを構成する基本的なグラフィカル要素です。

ウィジェットをカスタマイズできます。ウィジェットのプロパティを編集すれば、ニーズに合ったウィジェットに変更できます。次の要素は、タッチ時または移動時のプロパティの例です。

- ▶ サイズ
- ▶ 色
- ▶ レイアウト
- ▶ 動作

ウィジェットを組み合わせることができます。小さなブロックを積み重ねて複雑な構造物を作成できます。例えば、次の要素のボタンを作成できます。

- ▶ 楕円
- ▶ イメージ
- ▶ ラベル
- ▶ 四角形

ウィジェットを入れ子にすることができます。ウィジェット階層では、下位のウィジェットを子ウィジェットと呼び、上位のウィジェットを親ウィジェットと呼びます。

### 6.22.1. ビュー

ビューは、シーンの最上位ウィジェットです。モデリング時に、次の要素がビューに配置されます。

- ▶ 基本ウィジェット
- ▶ 3Dウィジェット
- ▶ アニメーション
- ▶ ウィジェットテンプレート

ビューは、1つのビューステートに関連付けられます。ビューは、ビューステートなしで存在できません。

**NOTE**




**ビューのサイズ変更**

EB GUIDE Studioでは、一部の領域を詳細に把握したりより広い領域を確認するためにビューのサイズを拡大または縮小することができます。拡大または縮小を行うには、スライダーを使うか、ビューの一番下にあるテキストボックスをクリックします。デフォルトのズームレベルは100%になっています。また、**Ctrl++**キーで拡大、**Ctrl+-**キーで縮小、**Ctrl+0**キーでズームレベル100%へのリセットを行うこともできます。

**NOTE**



**マスターイメージ上の要素の整列**

EB GUIDE Studioでは、基本ウィジェットや3Dウィジェットといった要素を均等に整列させるためにビューにマスターイメージを追加できます。ビューにマスターイメージを追加するには、ビューの下部にある をクリックします。マスターイメージを非表示にするには、チェックボックスをオンまたはオフにします。ビューを閉じる場合は、マスターイメージを再度追加する必要があります。

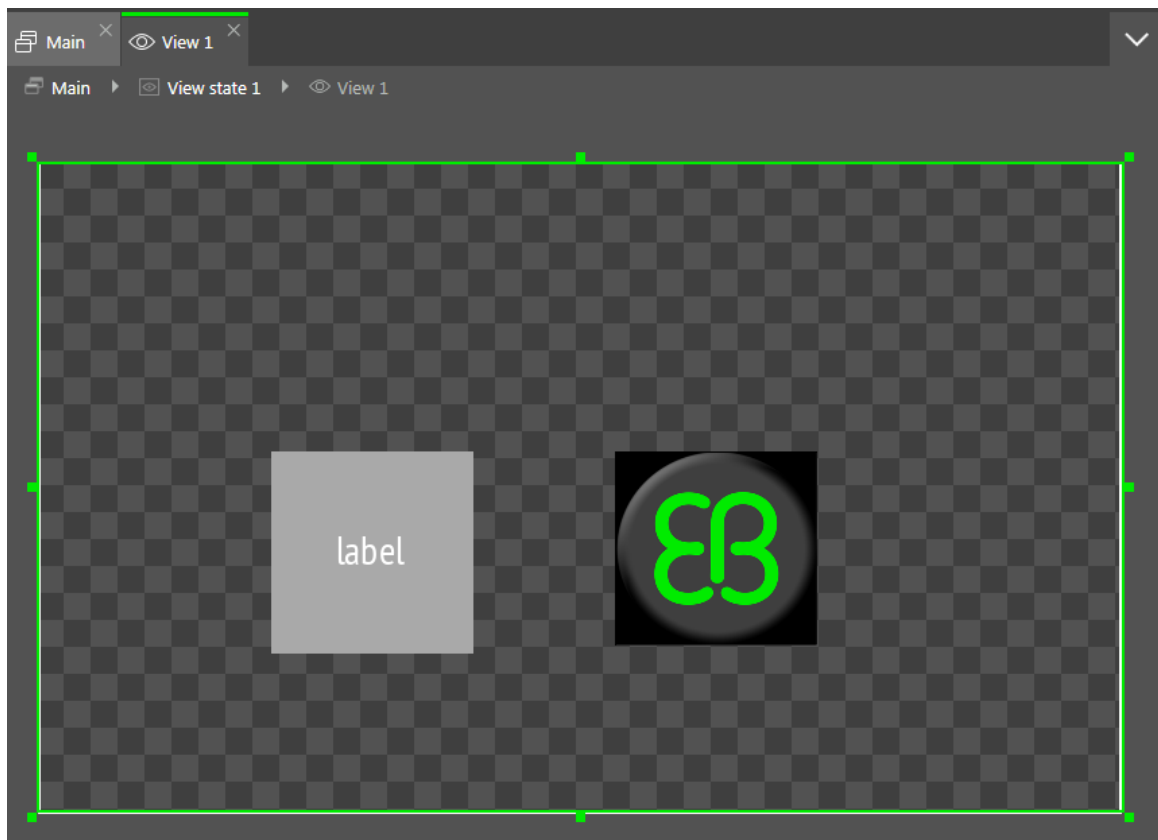


Figure 6.32. 四角形、ラベル、イメージを含むビュー

## 6.22.2. ウィジェットのカテゴリ

ツールボックス内で、ウィジェットはカテゴリによって分類されています。以下のカテゴリがあります。

▶ 基本ウィジェット

次の要素は基本ウィジェットです。

- ▶ アルファマスク
- ▶ アニメーション
- ▶ コンテナ
- ▶ 楕円
- ▶ イメージ
- ▶ インスタンスエータ
- ▶ ラベル
- ▶ 四角形

▶ 3Dウィジェット

3Dグラフィックを表示するには、**3Dウィジェット**カテゴリに含まれるウィジェットを使用します。**3Dウィジェット**は次の要素です。

- ▶ 環境光
- ▶ カメラ
- ▶ 指向性ライト
- ▶ イメージベースドライバ
- ▶ 材質
- ▶ メッシュ
- ▶ PBR GGX材質
- ▶ PBR Phong材質
- ▶ 点ライト
- ▶ シーングラフ
- ▶ シーングラフノード
- ▶ スポットライト

**NOTE**



サポートされているレンダラー

3Dグラフィックを表示するには、OpenGL ES 2.0以降が必要です。グラフィックドライバがレンダラーのバージョンと互換性があることを確認してください。

▶ ウィジェットテンプレート

このテンプレートカテゴリには、ウィジェットテンプレートが含まれます。このカテゴリは、ウィジェットテンプレートが定義されている場合にのみ表示されます。

#### ▶ カスタムウィジェット

カスタムウィジェットカテゴリにはカスタマイズされたウィジェットが含まれているため、カスタマイズされたウィジェットがプロジェクトに追加されたときにのみ表示されます。詳細については、弊社のWebサイト<https://www.elektrobit.com/ebguide/learn/resources/>をご覧ください。

手順については、[8.1「ウィジェットの操作」](#)をご覧ください。

### 6.22.3. ウィジェットプロパティ

ウィジェットは、外観や動作を指定する一連のプロパティによって定義されます。プロパティコンポーネントには、現在フォーカスがあるウィジェットのプロパティが表示され、プロパティを編集する機能が提供されています。

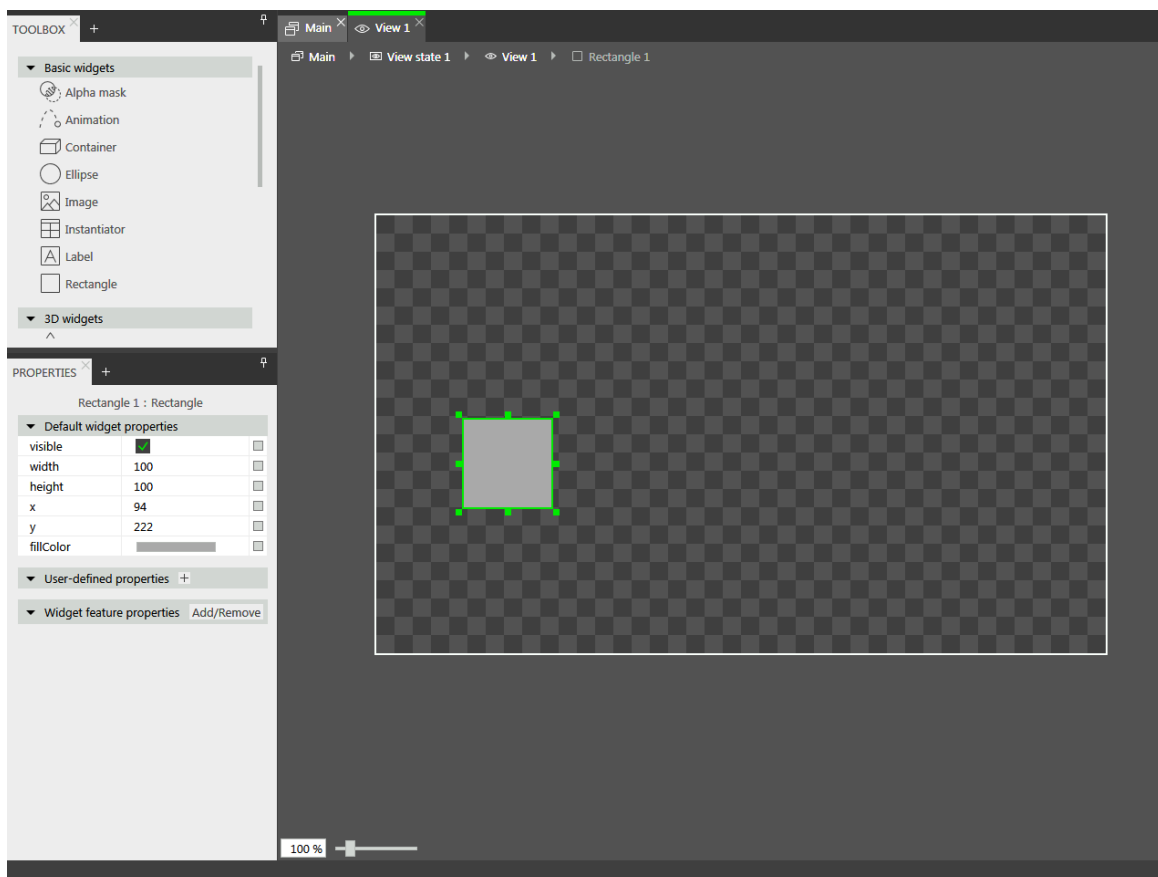


Figure 6.33. 四角形とそのプロパティ

ウィジェットプロパティには3つのタイプがあります。



- ▶ デフォルトウィジェットプロパティは、ウィジェットインスタンスと共に作成されます。すべてのウィジェットのデフォルトプロパティについては、[13.12「ウィジェット」](#)をご覧ください。
- ▶ ユーザー定義ウィジェットプロパティは、デフォルトプロパティに加えて、モデラーが作成するプロパティです。
- ▶ ウィジェット機能プロパティは、モデラーがウィジェット機能をウィジェットに追加する際にEB GUIDE Studioによって作成されます。ウィジェット機能プロパティは、カテゴリに分かれています。ウィジェット機能は、ウィジェットの外観と動作に関して別の機能を追加します。



#### Example 6.34.

##### タッチウィジェット機能

タッチウィジェット機能は、ウィジェットがタッチに反応するか、反応する場合はどう反応するかを定義します。プロパティは4つ追加されます。ブール値プロパティ`touchable`は、ウィジェットがタッチ入力に反応するかどうかを定義します。ブール値プロパティ`touched`は、ウィジェットが現在タッチされている場合、EB GUIDEによってランタイムに設定されます。2つの整数プロパティ、`touchPolicy`および`touchBehavior`は、ウィジェットがタッチ入力にどう反応するかを定義します。

## 6.22.4. ウィジェットテンプレート

ウィジェットテンプレートでは、EB GUIDEモデルで何度も使用できるカスタマイズされたウィジェットを定義できます。既存のウィジェットに基づいてテンプレートを定義したり、既存のテンプレートから新しいテンプレートを派生させたりすることができます。テンプレートを作成した後は、必要に応じて、例えばプロパティやウィジェット機能の追加などで変更します。このため、ウィジェットテンプレートでは、複雑なウィジェットのライブラリを構築できます。

ウィジェットテンプレートは、テンプレートインターフェースを備えています。テンプレートインターフェースにはテンプレートのプロパティが含まれます。これらのプロパティは、ウィジェットインスタンスで表示およびアクセスが可能です。こうして、ウィジェットインスタンスはそのテンプレートのインターフェースからプロパティを継承します。継承されたプロパティは、テンプレートプロパティと呼ばれます。テンプレートプロパティは■ボタンでマークされます。

テンプレートプロパティの値を変更すると、そのプロパティはローカルプロパティに変わります。ローカルプロパティは■ボタンでマークされます。



#### Example 6.35.

##### ウィジェットテンプレートのプロパティとそのインスタンスの関係

SquareウィジェットテンプレートをEB GUIDEモデルに追加します。Squareに`color`というプロパティを追加します。`color`はテンプレートインターフェースに追加されます。`color`の値を`red`に設定します。

Squareウィジェットテンプレートのインスタンスをビューに追加します。インスタンスはBlueSquareという名前になります。

- ▶ BlueSquare には、`color`が値`red`で継承されます。

- ▶ Squareテンプレートでcolorの値をgreenに変更します。  
=> BlueSquareでもcolorの値がgreenに変わります。
- ▶ BlueSquareでcolorの値をblueに変更します。  
Squareテンプレートでcolorの値をyellowに変更します。  
=> BlueSquareのcolorの値はblueのままです。

手順については、[8.7「ウィジェットの再利用」](#)をご覧ください。

## 6.22.5. ウィジェット機能

ウィジェット機能を使用してウィジェットおよびウィジェットテンプレートの機能を拡張できます。ウィジェット機能には、事前定義済みのウィジェットプロパティがあります。ウィジェット機能は、カテゴリにグループ化されています。

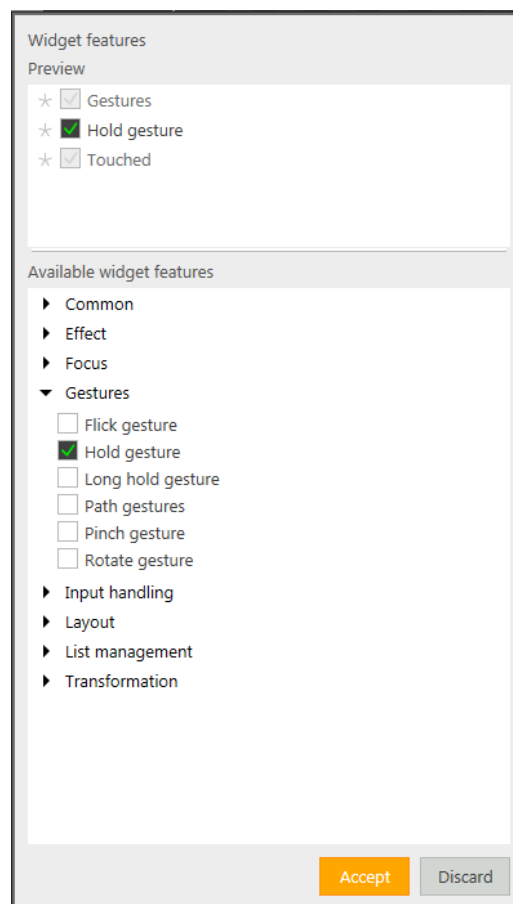


Figure 6.34. ウィジェット機能

ウィジェット機能をウィジェットテンプレートに追加すると、作成されるウィジェットテンプレートのインスタンスに、追加されたウィジェット機能が継承されます。ウィジェットテンプレートのインスタンス、またはテンプレートから作成されたテンプレートには、ウィジェット機能を追加できません。

ウィジェット機能を使用する場合の制限は次のとおりです。

- ▶ ウィジェット機能には継承階層がありません。
- ▶ 1つのウィジェットにウィジェット機能を複数回追加することはできません。
- ▶ 一部のウィジェット機能には相互依存関係があります。これは、あるウィジェット機能を追加するには別のウィジェット機能を追加する必要があること、またはウィジェット機能同士が相互に排他的であることを意味します。
- ▶ ウィジェット機能を特定タイプのウィジェットに制限できます。
- ▶ ランタイム中にウィジェット機能を有効または無効にすることはできません。

デフォルトでは、すべてのウィジェット機能は無効にされます。特定のウィジェット機能が必要な場合は、それをウィジェットに追加する必要があります。

手順については、[8.3「ウィジェット機能を追加してウィジェットを拡張する」](#)をご覧ください。すべてのウィジェット機能の一覧については、[13.13「ウィジェット機能」](#)をご覧ください。

#### 6.22.5.1. フォーカスウィジェット機能カテゴリ

EB GUIDE Studioでは、ウィジェットのフォーカス管理はフォーカスウィジェット機能(自動フォーカスとユーザー定義フォーカス)を使用してモデリングします。

次の2つのフォーカス方向を使用できます。

1. 順方向: 次のフォーカス可能なウィジェットがフォーカスされます。
2. 逆方向: 前のフォーカス可能なウィジェットがフォーカスされます。

自動フォーカスおよびユーザー定義フォーカスウィジェット機能は、順方向でフォーカスを処理する方法の設定を提供します。逆方向の場合は、同じフォーカス順序が逆方向で使用されます。

フォーカスウィジェット機能には次の特性があります。

##### 自動フォーカス

このポリシーでは、フォーカスは1番上の行から始まり左から右という順序でフォーカス可能なウィジェットに適用されます。順序はウィジェットツリーの構造によって定義されます。

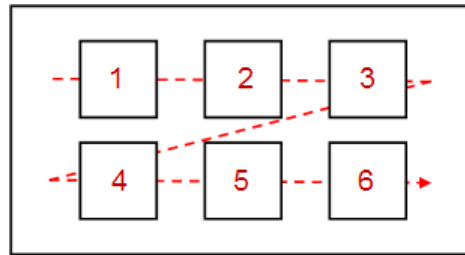


Figure 6.35. 自動フォーカスウィジェット機能のポリシー

フォーカス可能な子ウィジェットはスキップできません。非表示のウィジェット、`focused`プロパティが無効にされているウィジェット、およびフォーカスウィジェット機能がないウィジェットは、有効なフォーカス可能なウィジェットとして認識されません。このため、それらは現在フォーカスされるウィジェットを決定するときにスキップされます。

#### ユーザー定義フォーカス

ビューが複雑なために、自動フォーカスポリシーに従ったフォーカス順序がきわめて難しいことがあります。この場合は、ユーザー定義フォーカスの順序を指定しておく役に立ちます。

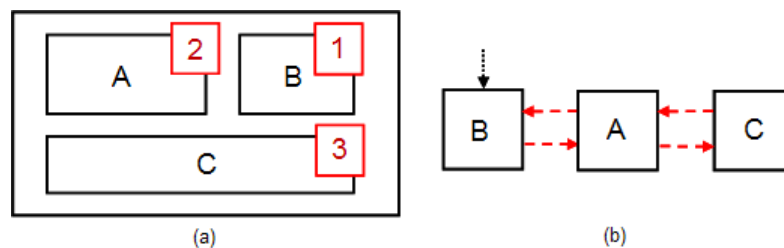


Figure 6.36. ユーザー定義フォーカスウィジェット機能のポリシー

[figure 6.36. “ユーザー定義フォーカスウィジェット機能のポリシー”](#)では、(a)はビューを示し、(b)はフォーカス順序を示しています。フォーカスの変更が処理される順序は、ウィジェットツリー構造と異なることがあります。

ウィジェット階層内のウィジェットがフォーカス可能とマークされる場合、それらはフォーカス階層の一部です。このフォーカス階層は、フォーカス可能なウィジェット、フォーカスポリシー、およびその階層内でフォーカスを処理する方法を定義する自動フォーカスウィジェット機能またはユーザー定義フォーカスウィジェット機能で構成されています。フォーカス階層を入れ子にすることができます。

#### 6.22.5.2. リスト管理ウィジェット機能カテゴリ

ラインインデックスおよびテンプレートインデックスウィジェット機能を使用すると、データ(イメージ、曲のタイトルなど)を対応する動的に作成されたインスタシエータのラインテンプレートに接続できます。

#### ラインインデックス

ラインインデックスウィジェット機能は、インスタシエータウィジェットのラインテンプレートをカスタマイズするために使用されます。ラインインデックスウィジェット機能は、リストまたは表の各ラインの一意の位置を定義します。



#### Example 6.36.

##### ラインインデックスウィジェット機能

リストをモデリングする場合は、リストプロパティのエントリーを反映した特定の値がリストの各エントリーにあることが予想されます。リスト内の特定のエントリーにアクセスするには、ラインテンプレートのインスタンスがそのエントリーがどのインスタシエータの子であるかを認識している必要があります。ラインインデックスウィジェット機能は、`lineIndex`プロパティを追加します。インスタシエータがラインテンプレートのインスタンスを作成するときに、`lineIndex`に値が設定されます。インデックスは最初のインスタンスをゼロとして開始されます。インスタシエータに2つの要素がある場合、2番目の要素は`lineIndex`値に1を受け取ります。

手順については、[12.4「チュートリアル: 動的コンテンツを使用したリストの作成」](#)をご覧ください。

#### テンプレートインデックス

テンプレートインデックスウィジェット機能では、複雑なデータ抽象化が可能です。非常に複雑なリストまたは表の場合は、1つのエントリーまたは一連のエントリーを視覚化するために複数のデータリストが必要となります。例えば、イメージとテキストが混在したコンテンツを持つ表では、イメージのリストと文字列のリストが必要となります。そのような複雑な場合をカバーするために、テンプレートインデックスウィジェット機能は`lineTemplateIndex`プロパティを提供しています。



#### Example 6.37.

##### テンプレートインデックスウィジェット機能

`lineMapping`プロパティに`0|1`を設定し、`numItems`プロパティに5を設定してインスタシエータを使用してリストをモデリングする場合、`lineTemplateIndex`は`0|0|1|1|2`となります。

## 7. ヒューマンマシンインターフェースの動作のモデル化

注記



デフォルトのウィンドウレイアウト

すべての手順およびスクリーンショットでは、デフォルトのウィンドウレイアウトが使用されています。手順に従う場合は、EB GUIDE Studioウィンドウを [レイアウト] > [Reset to default layout] を選択することによって、デフォルトレイアウトにリセットすることをお勧めします。

### 7.1. ステートマシンのモデリング

#### 7.1.1. ステートマシンの追加



ステートマシンの追加

##### ステップ 1

[ナビゲーション]コンポーネントで[ステートマシン]に移動し、+をクリックします。

メニューが展開されます。

##### ステップ 2

ステートマシンのタイプを選択します。

選択したタイプの新しいステートマシンが追加されます。

##### ステップ 3

ステートマシンの名前を変更します。

#### 7.1.2. 動的ステートマシンの追加

動的ステートマシンは他のステートマシンと平行して動作し、ランタイムの間に開始(プッシュ)したり終了(ポップ)したりできます。



動的ステートマシンの追加

動的ステートマシンは、例えば通常の画面をオーバーレイしてエラーメッセージを表示する場合に使用します。

前提条件:

- ステートマシンがあり、ビューステートまたは混合ステートがEB GUIDEモデルに追加されていること。

#### ステップ 1

[ナビゲーション]コンポーネントで[動的ステートマシン]に移動し、+をクリックします。

メニューが展開されます。

#### ステップ 2

動的ステートマシンのタイプを選択します。

選択したタイプの新しい動的ステートマシンが追加されます。

#### ステップ 3

[ナビゲーション]コンポーネントで、動的ステートマシンと平行して動作させるステートマシン、ビューステート、または混合ステートをクリックします。

#### ステップ 4

[プロパティ]コンポーネントで、Dynamic state machine listチェックボックスを選択します。

以上の操作が終了したら、動的ステートマシンに関連付けられたEB GUIDEスクリプト関数を使用します。

詳細については、[12.1「チュートリアル: 動的ステートマシンの追加」](#)をご覧ください。

## 7.1.3. ステートマシンに対するエントリーアクションの定義



### ステートマシンに対するエントリーアクションの定義

#### ステップ 1

ステートマシンを選択します。

#### ステップ 2

[プロパティ]コンポーネントで、[エントリーアクション]プロパティに移動し、+ をクリックします。

スクリプトエディターが開きます。

#### ステップ 3

EB GUIDEスクリプトを使用してアクションを入力します。

バックグラウンド情報については、[6.16「スクリプト言語EB GUIDEスクリプト」](#)をご覧ください。

#### ステップ 4

[承認]をクリックします。

ステートマシンに対するエントリーアクションが定義されました。

## 7.1.4. ステートマシンに対する終了アクションの定義



### ステートマシンに対する終了アクションの定義

#### ステップ 1

ステートマシンを選択します。

#### ステップ 2

[プロパティ]コンポーネントで、[終了アクション]プロパティに移動し、+ をクリックします。

スクリプトエディターが開きます。

#### ステップ 3

EB GUIDEスクリプトを使用してアクションを入力します。

バックグラウンド情報については、[6.16「スクリプト言語EB GUIDEスクリプト」](#)をご覧ください。

#### ステップ 4

[承認]をクリックします。

ステートマシンに対する終了アクションが定義されました。

## 7.1.5. ステートマシンの削除



### ステートマシンの削除

#### ステップ 1

[ナビゲーション]コンポーネントで、ステートマシンを右クリックします。

#### ステップ 2

コンテキストメニューの[削除]をクリックします。

ステートマシンが削除されました。

## 7.2. モデリングステート

### 7.2.1. ステートの追加





## ステートの追加

前提条件:

- コンテンツエリアにステートマシンが表示されていること。

ステップ 1

[ツールボックス]からステートをドラッグし、ステートマシンにドロップします。

ステートがステートマシンに追加されます。

## 注記



初期ステート、最終ステート、および履歴ステートの一意性

初期ステート、最終ステート、および履歴ステートは、混合ステート1つにつき一度だけ挿入できます。

## ティップ



ステートのコピーと検索

既存のステートをコピーして貼り付けるには、コンテキストメニューまたはCtrl+CキーとCtrl+Vキーを使用するという方法もあります。

EB GUIDEモデル内で特定のステートを検索するには、ステートの名前を検索ボックスに入力するか、Ctrl+Fキーを使用します。ステートにジャンプするには、ヒットリスト内のステートをダブルクリックします。

## 7.2.2. 混合ステートへのステートの追加



## 混合ステートへのステートの追加

ステート階層を作成するには、ステートを別のステートの子として作成します。混合ステートにステートを追加することによって、これを行います。

前提条件:

- コンテンツエリアにステートマシンが表示されていること。
- ステートマシンには混合ステートが含まれていること。

ステップ 1

[ナビゲーション]コンポーネントで混合ステートをダブルクリックします。

混合ステートがコンテンツエリアに展開されます。

### ステップ 2

[ツールボックス]からステートをドラッグし、混合ステートにドロップします。

ステートは、混合ステートの子ステートとして追加されます。

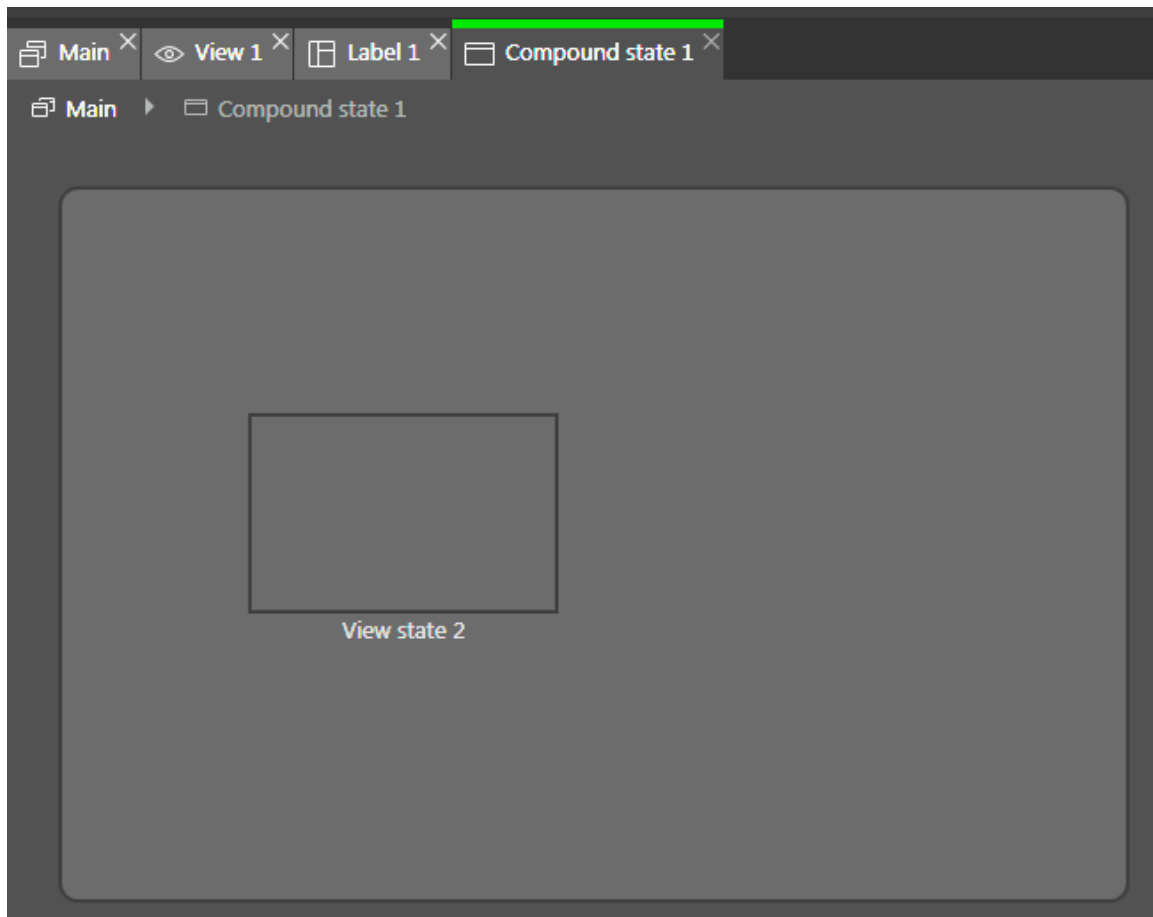


図7.1 入れ子のビューステートを持つ混合ステート

## 7.2.3. 選択ステートの追加



### 選択ステートの追加

前提条件:

- コンテンツエリアにステートマシンが表示されていること。
- ステートマシンには、少なくとも2つのステートが含まれています。

### ステップ 1

[ツールボックス]から選択ステートをドラッグし、ステートマシンにドロップします。

### ステップ 2

出力遷移に条件を追加します。詳細については、[7.3.4「遷移への条件の追加」](#)をご覧ください

この条件には優先度1が割り当てられます。ステートマシンが選択ステートにエントリーすると、優先度1の条件が最初に評価されます。

### ステップ 3

選択遷移をさらに追加するには、上記の2つのステップを繰り返します。

新しい選択遷移には、以前に作成された遷移よりも低い優先度が割り当てられます。

### ステップ 4

選択ステートからの出力遷移を追加します。

### ステップ 5

[ナビゲーション]コンポーネントで、遷移を右クリックします。コンテキストメニューの[elseに変換]をクリックします。

else遷移が追加されました。else遷移は、外へ向かう選択遷移に割り当てられているすべての条件がfalseと評価されたときに実行されます。

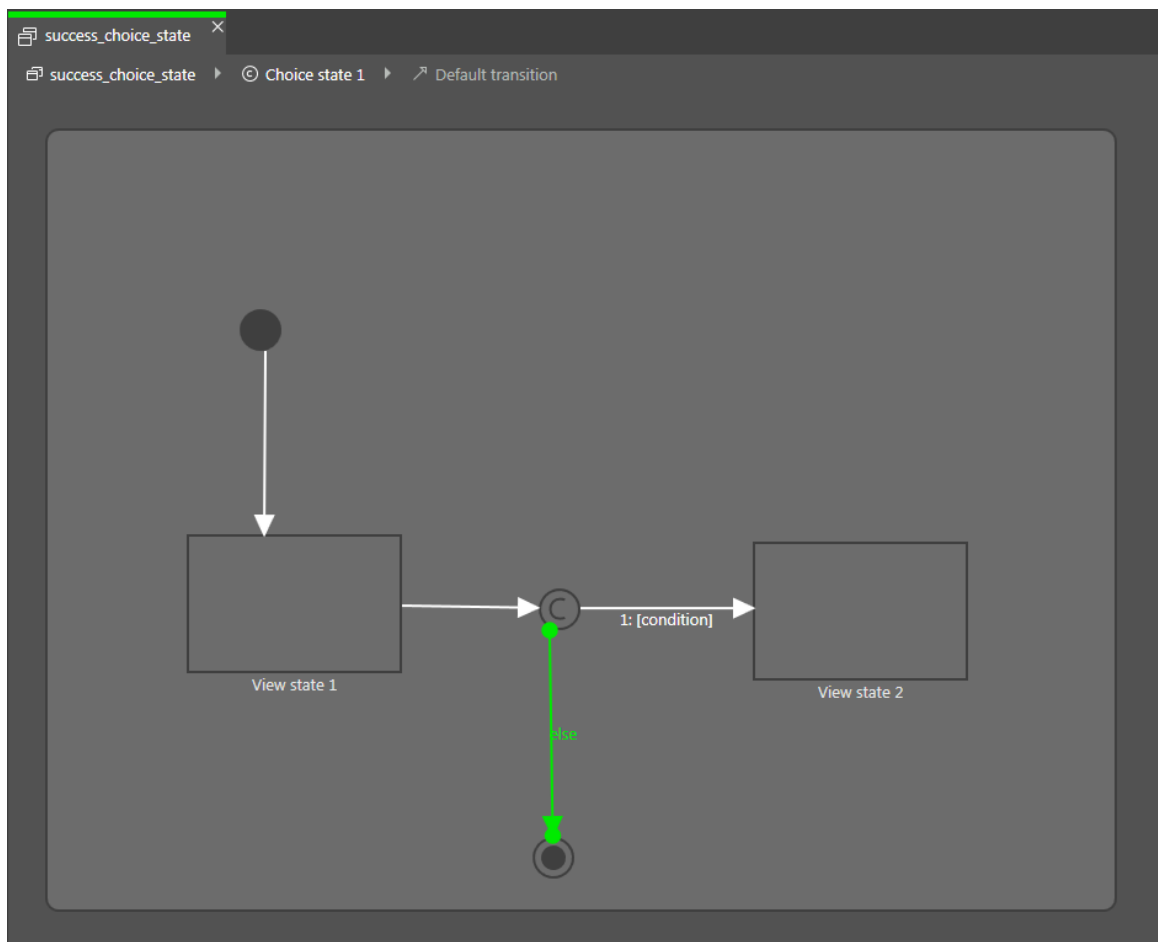


図7.2 選択遷移を持つ選択ステート

## 7.2.4. ステートに対するエントリーアクションの定義



### ステートに対するエントリーアクションの定義

ビューステートと混合ステートに対し、エントリーアクションを定義することができます。エントリーアクションは、そのステートの開始時に必ず実行されます。

前提条件:

- ステートマシンにビューステートまたは混合ステートが含まれていること。

#### ステップ 1

ステートを選択します。

#### ステップ 2

[プロパティ]コンポーネントで、Entry actionプロパティに移動し、+ をクリックします。

スクリプトエディターが開きます。

#### ステップ 3

EB GUIDEスクリプトを使用してアクションを入力します。

バックグラウンド情報については、[6.16「スクリプト言語EB GUIDEスクリプト」](#)をご覧ください。

#### ステップ 4

[承認]をクリックします。

## 7.2.5. ステートに対する終了アクションの定義



### ステートに対する終了アクションの定義

ビューステートおよび混合ステートに対し、終了アクションを定義することができます。終了アクションは、そのステートの終了時に必ず実行されます。

前提条件:

- ステートマシンにビューステートまたは混合ステートが含まれていること。

#### ステップ 1

ステートを選択します。

#### ステップ 2

[プロパティ]コンポーネントで、Exit actionプロパティに移動し、+ をクリックします。

スクリプトエディターが開きます。

### ステップ 3

EB GUIDEスクリプトを使用してアクションを入力します。

バックグラウンド情報については、[6.16「スクリプト言語EB GUIDEスクリプト」](#)をご覧ください。

### ステップ 4

[承認]をクリックします。

## 7.2.6. ステートマシンからのモデル要素の削除



### ステートマシンからのモデル要素の削除

前提条件:

- ステートマシンに、少なくとも1つのモデル要素が含まれていること。

### ステップ 1

[ナビゲーション]コンポーネントで、モデル要素を右クリックします。

### ステップ 2

コンテキストメニューの[削除]をクリックします。

モデル要素が削除されます。

## 7.3. ステート間を遷移で接続

### 7.3.1. 2つのステート間に遷移を追加



### 2つのステート間に遷移を追加

遷移によって、ソースステートとターゲットステートを接続します。

前提条件:

- コンテンツエリアにステートマシンが表示されていること。
- ステートマシンには、少なくとも2つのステートが含まれています。

### ステップ 1

遷移のソースステートとなるステートを選択します。

### ステップ 2

緑色のドラッグ点をクリックし、マウスボタンを押したままにします。

### ステップ 3

ターゲットステートまでマウスをドラッグします。

### ステップ 4

ターゲットステートが緑色で強調表示されたら、マウスボタンを離します。

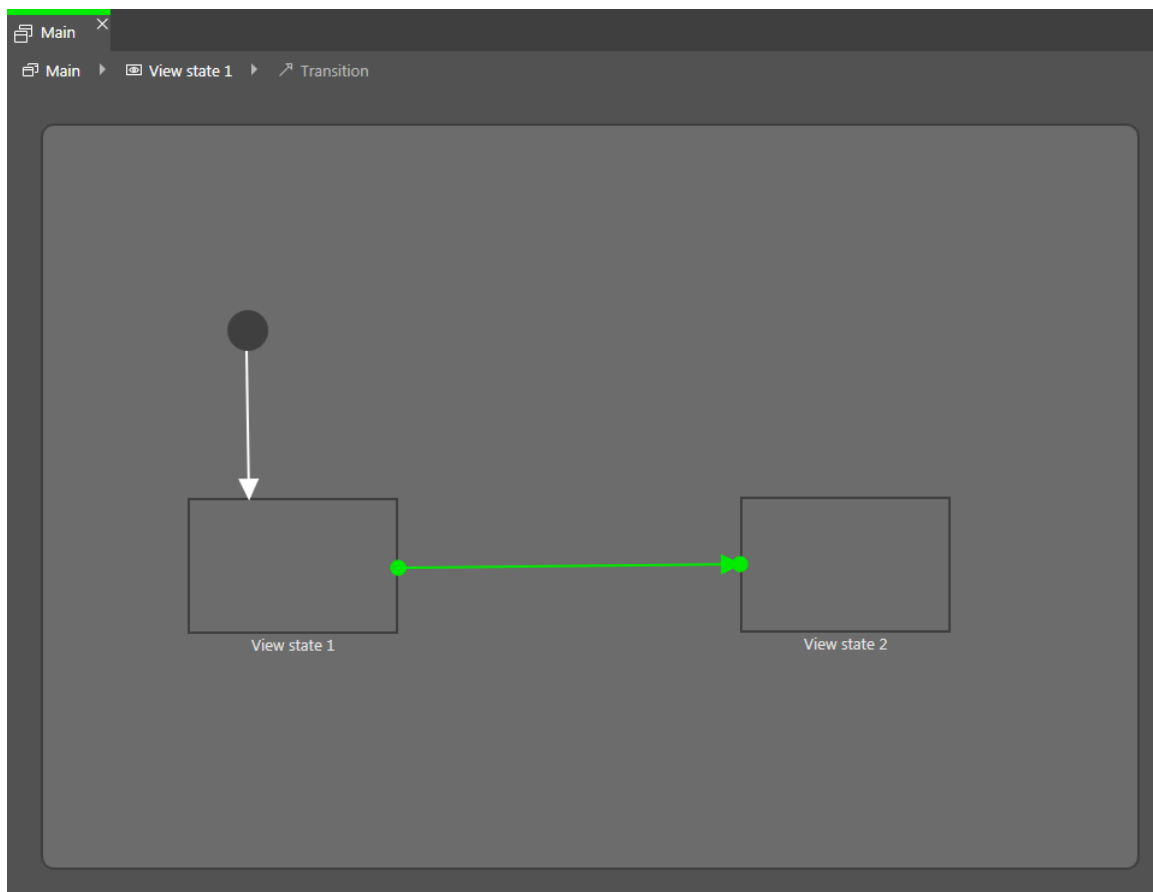


図7.3 遷移

遷移が追加され、緑色の矢印として表示されます。

#### ティップ



ステートマシンに遷移を接続

ステートマシンは、最上層の混合ステートです。したがって、ステートマシンの枠に対して入出力する遷移を作成できます。そのような遷移は、ステートマシン内のすべてのステートに継承されます。

## 7.3.2. 遷移の移動



### 遷移の移動

遷移の移動は、一方の終点を動かすことによって行います。

前提条件:

- コンテンツエリアにステートマシンが表示されていること。
- ステートマシンには、少なくとも2つのステートが含まれています。
- ステート間は遷移によって接続されていること。

#### ステップ 1

コンテンツエリアで、遷移をクリックします。

2つの緑色のドラッグ点が表示されます。

#### ステップ 2

移動させるドラッグ点をクリックし、マウスボタンを押したままにします。

#### ステップ 3

異なるステートまでマウスをドラッグします。

#### ステップ 4

ステートが緑色で強調表示されたら、マウスボタンを離します。

すると、遷移が移動します。

## 7.3.3. 遷移に対するトリガーの定義



### 遷移に対するトリガーの定義

遷移に対し、それをトリガーするイベントを定義できます。

前提条件:

- ステートマシンに少なくとも2つのステートが含まれていること。
- ステート間は遷移によって接続されていること。

#### ステップ 1

遷移をクリックします。

#### ステップ 2

[プロパティ]コンポーネントで、[トリガー]の横にあるボックスをクリックします。

### ステップ 3

イベントをクリックします。

新しいイベントを作成する場合は、名前を入力し、[イベントの追加]をクリックします。

イベントが、遷移トリガーとして追加されます。

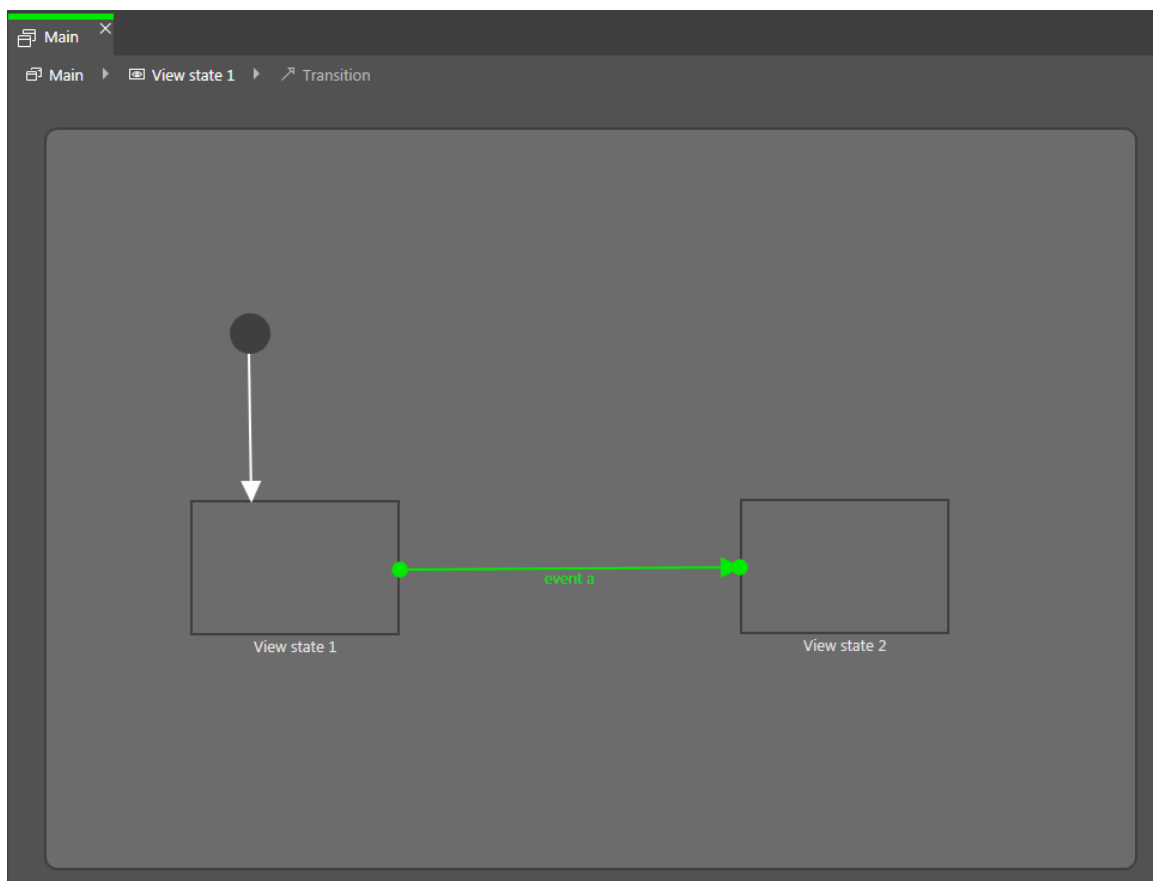


図7.4 トリガー付きの遷移

## 7.3.4. 遷移への条件の追加



### 遷移への条件の追加

各遷移に対し、その遷移を実行するために満たすべき条件を定義することができます。

前提条件:

- ステートマシンに少なくとも2つのステートが含まれていること。
- ステート間は遷移によって接続されていること。



#### ステップ 1

遷移をクリックします。

#### ステップ 2

遷移に条件を追加するには、[プロパティ]コンポーネントに移動します。Conditionプロパティの横にある+ をクリックします。

スクリプトエディターが開きます。

#### ステップ 3

EB GUIDEスクリプトを使用して条件を入力します。

バックグラウンド情報については、[6.16「スクリプト言語EB GUIDEスクリプト」](#)をご覧ください。

#### ステップ 4

[承認]をクリックします。

条件が遷移に追加されます。

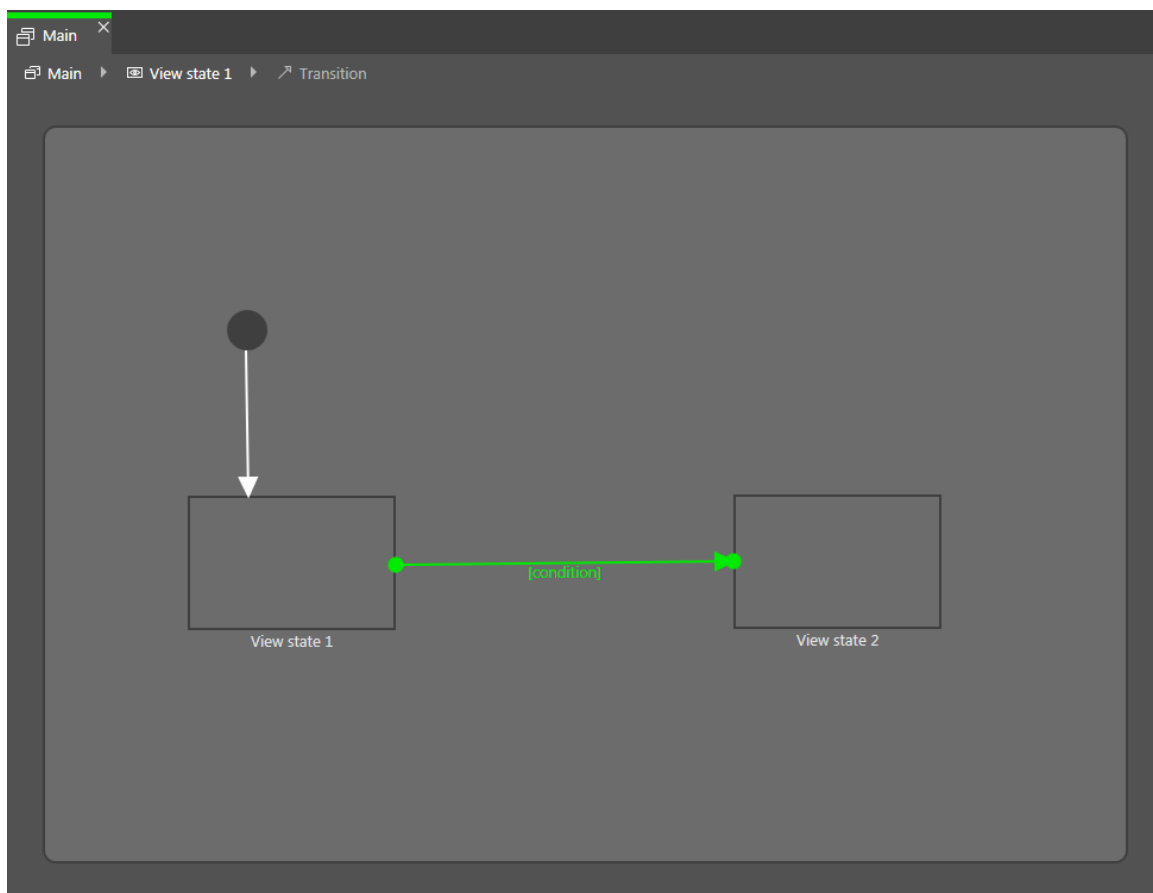


図7.5 条件付きの遷移

### 7.3.5. 遷移へのアクションの追加



## 遷移へのアクションの追加

各遷移に対し、その遷移時に実行されるアクションを定義することができます。

前提条件:

- ステートマシンに少なくとも2つのステートが含まれていること。
- ステート間は遷移によって接続されていること。

### ステップ 1

遷移をクリックします。

### ステップ 2

遷移にアクションを追加するには、[プロパティ]コンポーネントに移動します。Actionプロパティの横にある+ をクリックします。

スクリプトエディターが開きます。

### ステップ 3

EB GUIDEスクリプトを使用してアクションを入力します。

バックグラウンド情報については、[6.16「スクリプト言語EB GUIDEスクリプト」](#)をご覧ください。

### ステップ 4

[承認]をクリックします。

アクションが遷移に追加されます。

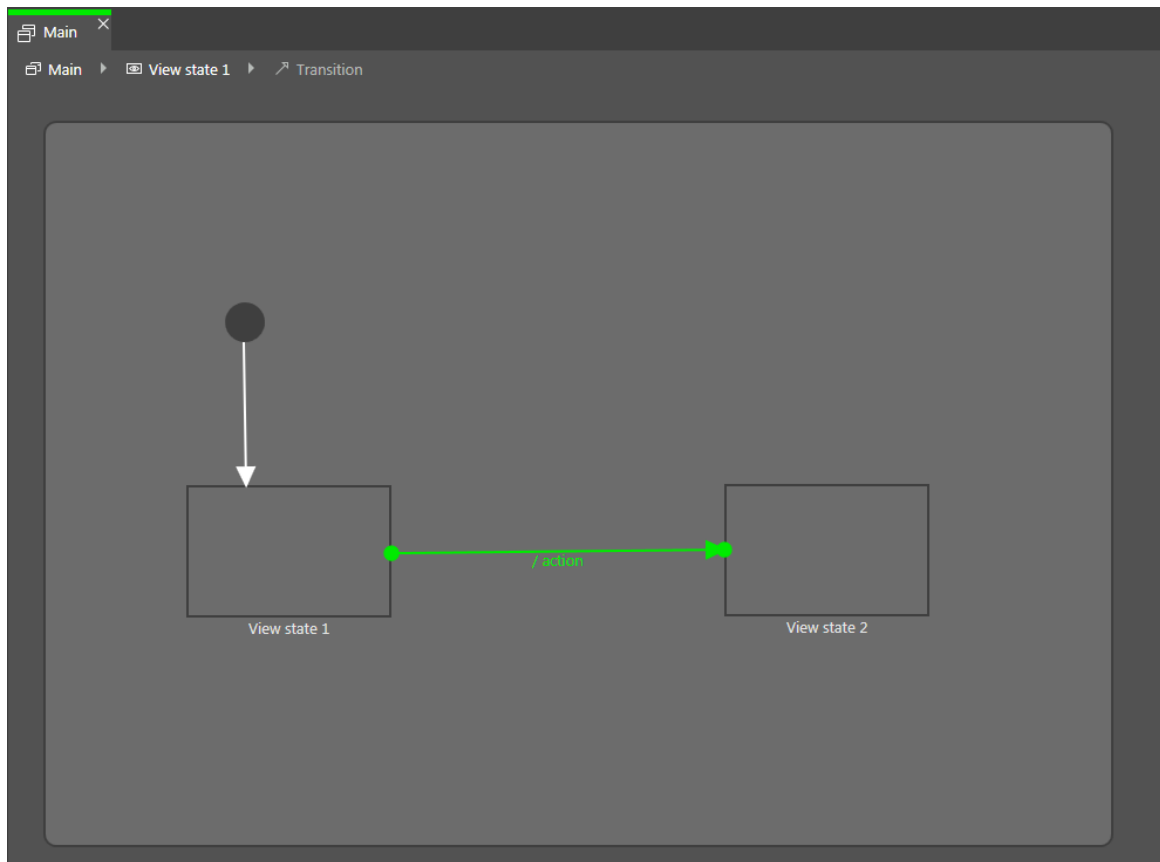


図7.6 アクション付きの遷移

### 7.3.6. ステートへの内部遷移の追加



#### ステートへの内部遷移の追加

前提条件:

- ステートマシンにステートが含まれていること。

#### ステップ 1

ステートを選択します。

#### ステップ 2

[プロパティ]コンポーネントで、[内部遷移]に移動し、+ をクリックします。

スクリプトエディターが開きます。

内部遷移がステートに追加されます。内部遷移が[ナビゲーション]コンポーネントに表示されます。

## 8. ヒューマンマシンインターフェースの外観のモデル化

### 注記



#### デフォルトのウィンドウレイアウト

すべての手順およびスクリーンショットでは、デフォルトのウィンドウレイアウトが使用されています。手順に従う場合は、EB GUIDE Studioウィンドウを [レイアウト] > [Reset to default layout] を選択することによって、デフォルトレイアウトにリセットすることをお勧めします。

### 8.1. ウィジェットの操作

#### ティップ



#### ビューおよびウィジェットのコピーと検索

既存のビューまたはウィジェットをコピーして貼り付けるには、コンテキストメニューまたはCtrl + CキーとCtrl + Vキーを使用できます。

EB GUIDEモデル内で特定のビューまたはウィジェットを検索するには、ビューまたはウィジェットの名前を検索ボックスに入力するか、Ctrl + Fキーを使用します。ビューまたはウィジェットにジャンプするには、ヒットリスト内のビューまたはウィジェットをダブルクリックします。

#### 8.1.1. ビューの追加



##### ビューの追加

#### 前提条件:

- コンテンツエリアにステートマシンが表示されていること。

#### ステップ 1

[ツールボックス]からビューステートをドラッグし、ステートマシンにドロップします。

ビューステートと共に、ビューがEB GUIDEモデルに追加されます。

#### ステップ 2

[ナビゲーション]コンポーネントでビューをクリックします。

#### ステップ 3

F2キーを押し、ビューの名前を変更します。

#### ステップ 4

コンテンツエリアでビューステートをダブルクリックします。

コンテンツエリアに新しいビューが表示されます。

## 8.1.2. ビューへの基本ウィジェットの追加

基本ウィジェットの詳細については、[13.12.2「基本ウィジェット」](#)をご覧ください。

### 8.1.2.1. 四角形を追加する



#### 四角形を追加する

前提条件:

- コンテンツエリアにビューが表示されていること。

#### ステップ 1

[ツールボックス]から四角形をドラッグし、ビューにドロップします。

四角形がビューに追加されます。

### 8.1.2.2. 楕円を追加する



#### 楕円を追加する

前提条件:

- コンテンツエリアにビューが表示されていること。

#### ステップ 1

[ツールボックス]から楕円をドラッグし、ビューにドロップします。

ウィジェットがビューに追加されます。

#### 8.1.2.2.1. 楕円を編集する

楕円の扇型のみを描画したり、楕円の弧を変更したりできます。



### 扇型を作成する

前提条件:

- ビューに楕円が含まれていること。

#### ステップ 1

楕円をクリックし、[プロパティ]コンポーネントに移動します。

#### ステップ 2

扇型の角度をcentralAngleテキストボックスに入力します。

#### ステップ 3

扇型の方向をsectorRotationテキストボックスに入力します。

扇型を作成しました。



### 円弧を作成する

前提条件:

- ビューに楕円が含まれていること。

#### ステップ 1

楕円をクリックし、[プロパティ]コンポーネントに移動します。

#### ステップ 2

幅(0~50)をarcWidthテキストボックスに入力します。

円弧を作成しました。

### 8.1.2.3. イメージを追加する



### [ツールボックス]を使用してイメージを追加する

前提条件:

- イメージファイルは、\$GUIDE\_PROJECT\_PATH/<project name>/resourcesディレクトリに配置します。サポートされているファイルタイプについては、[6.15.3「イメージ」](#)をご覧ください。
- コンテンツエリアにビューが表示されていること。

#### ステップ 1

[ツールボックス]からイメージをビューにドラッグしてドロップします。

### ステップ 2

[プロパティ]コンポーネントで、imageコンボボックスからイメージを選択します。または、[アセット]コンポーネントから別のイメージをドラッグして、imageドロップダウンリストボックスにドロップします。

ビューにイメージが表示されます。



[アセット]コンポーネントを使用してイメージを追加する

前提条件:

- イメージファイルは、`$GUIDE_PROJECT_PATH/<project name>/resources`ディレクトリに配置します。サポートされているファイルタイプについては、[6.15.3「イメージ」](#)をご覧ください。
- コンテンツエリアにビューが表示されていること。

### ステップ 1

[アセット]コンポーネントからイメージファイルをドラッグしてビューにドロップします。

ビューにイメージが表示されます。

### ステップ 2

イメージファイルを変更するには、[プロパティ]コンポーネントに移動して、imageコンボボックスからイメージを選択します。または、[アセット]コンポーネントから別のイメージをドラッグして、imageコンボボックスにドロップします。

ビューにイメージが表示されます。



9-patchイメージを追加する

前提条件:

- 9-patchイメージファイルは、`$GUIDE_PROJECT_PATH/<project name>/resources`ディレクトリに配置します。9-patchイメージのバックグラウンド情報については、[6.15.3.1「9-patchイメージ」](#)をご覧ください。
- コンテンツエリアにビューが表示されていること。
- イメージがEB GUIDEモデルに追加されます。

### ステップ 1

イメージを選択し、[プロパティ]コンポーネントに移動します。

### ステップ 2

imageコンボボックスで9-patchイメージを選択します。

### ステップ 3

[ウィジェット機能プロパティ]に移動して、[追加/削除]をクリックします。

[ウィジェット機能]ダイアログが表示されます。

#### ステップ 4

[使用可能なウィジェット機能]で、[レイアウト]カテゴリを展開して[拡大縮小モード]を選択します。

#### ステップ 5

[承認]をクリックします。

関連するウィジェットプロパティがイメージに追加され、[プロパティ]コンポーネントに表示されます。

#### ステップ 6

[プロパティ]コンポーネントで、scaleModeプロパティとしてfit to Size (1)を選択します。

#### 注記



#### 9-patchイメージを追加する

[拡大縮小モード]ウィジェット機能を追加していない場合、またはscaleModeプロパティにoriginal Size (0)またはkeep aspect ratio (2)を選択した場合、9-patchイメージは通常の.pngイメージのように拡大縮小されます。

### 8.1.2.4. ラベルを追加する



[ツールボックス]を使用してラベルを追加する

前提条件:

- コンテンツエリアにビューが表示されていること。

#### ステップ 1

[ツールボックス]からラベルをドラッグし、ビューにドロップします。

ビューにラベルが追加されます。ラベルのフォントは、デフォルトフォントのPT\_Sans\_Narrow.ttfです。



[アセット]コンポーネントを使用してラベルを追加する

前提条件:

- フォントファイルは、\$GUIDE\_PROJECT\_PATH/<project name>/resourcesディレクトリに配置します。サポートされているファイルタイプについては、[6.15.1「フォント」](#)をご覧ください。
- コンテンツエリアにビューが表示されていること。

#### ステップ 1

[アセット]コンポーネントからフォントファイルをドラッグしてビューにドロップします。

選択したフォントでビューにラベルが表示されます。



## 8.1.2.4.1. ラベルのフォントの変更



## ラベルのフォントの変更

前提条件:

- フォントファイルは、\$GUIDE\_PROJECT\_PATH/<project name>/resourcesディレクトリに配置します。サポートされているファイルタイプについては、[6.15.1「フォント」](#)をご覧ください。
- EB GUIDEモデルにビューステートが含まれていること。
- ビューにラベルが含まれていること。

## ステップ 1

ビューでラベルを選択します。

## ステップ 2

[プロパティ]コンポーネントで、fontコンボボックスからフォントを選択します。

または、[アセット]コンポーネントからフォントファイルをドラッグして、fontコンボボックスにドロップします。

ビューに表示されているラベルのフォントが変更されます。 .fntビットマップフォントを選択した場合、フォントのサイズは固定されており、ラベルのfontプロパティでフォントサイズを変更することはできません。

## 注記



## テキストの高さと行間の計算

次の図は、EB GUIDE Studioでテキストの高さ、行の高さ、および行間が計算される方法を示しています。ラベルのフォントスタイル、サイズ、または行間を変更するときは、これを考慮してください。

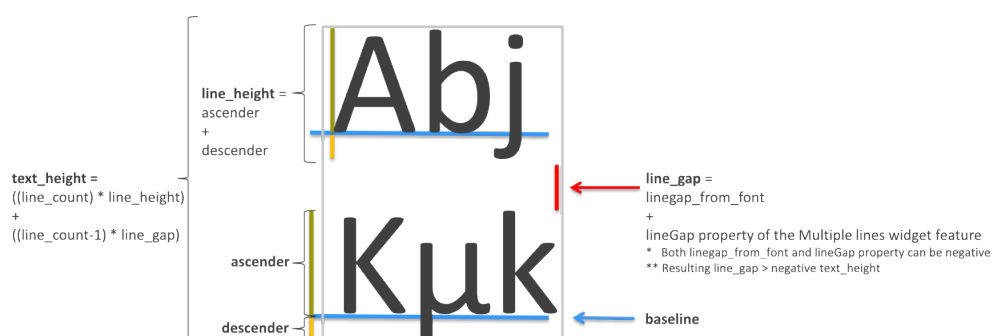


図8.1 テキストの高さ、行の高さ、および行間の計算

マルチフォントサポートの使用方法については、[8.2.8「マルチフォントサポートの管理」](#)をご覧ください。

## 8.1.2.5. コンテナーを追加する



### コンテナを追加する

コンテナを使用するとウィジェットをグループ化できます。

前提条件:

- コンテンツエリアにビューが表示されていること。

#### ステップ 1

[ツールボックス]からコンテナをドラッグし、ビューにドロップします。

#### ステップ 2

コンテンツエリアで、コンテナの四隅のいずれかをドラッグしてコンテナを拡大します。

#### ステップ 3

[ツールボックス]から2つ以上のウィジェットをドラッグし、コンテナにドロップします。

ウィジェットがコンテナの子のウィジェットとしてモデリングされます。コンテナを移動すると、子のウィジェットも一緒に移動します。

### 8.1.2.6. インスタンスエータの追加



### インスタンスエータの追加

前提条件:

- コンテンツエリアにビューが表示されていること。

#### ステップ 1

[ツールボックス]からインスタンスエータをビューにドラッグしてドロップします。

#### ステップ 2

[ツールボックス]からウィジェットをインスタンスエータにドラッグしてドロップします。

ウィジェットはラインテンプレートとして機能します。

#### ステップ 3

インスタンスエータを選択し、[プロパティ]コンポーネントに移動します。

##### ステップ 3.1

numItemsプロパティには、1よりも大きい値を入力してください。

##### ステップ 3.2

次のいずれかのウィジェット機能を、インスタンスエータに追加します。

- ▶ [ボックスレイアウト]

- ▶ [フローレイアウト]
- ▶ [グリッドレイアウト]
- ▶ [リストレイアウト]

詳細については、[8.3.1「ウィジェット機能の追加」](#)をご覧ください。

子ウィジェットが、numItemsプロパティで指定された回数だけ、ウィジェット機能でインスタンスータに指定されたレイアウトで、ビューに表示されます。

#### ステップ 4

[ツールボックス]からウィジェットをインスタンスータにドラッグしてドロップします。

2番目のラインテンプレートとして機能する2番目の子ウィジェットを追加しました。

#### ステップ 5

インスタンスータを選択し、[プロパティ]コンポーネントに移動します。

##### ステップ 5.1

lineMappingを選択し、をクリックします。

##### ステップ 5.2

[追加]ボタンをクリックします。

新しいエントリーが表に追加されます。

##### ステップ 5.3

Valueテキストボックスに、0と入力します。

##### ステップ 5.4

[追加]ボタンをクリックします。

新しいエントリーが表に追加されます。

##### ステップ 5.5

Valueテキストボックスに、1と入力します。

ラインテンプレートがインスタンス化される順序を定義しました。



#### 例8.1

##### インスタンス化の順序

lineMappingプロパティはインスタンス化の順序を定義します。例えば、値1|0を入力した場合、インスタンスータはラインテンプレート1を最初の子ウィジェットとしてインスタンス化し、ラインテンプレート0を2番目の子ウィジェットとしてインスタンス化します。

lineMappingプロパティは、反復して適用されます。これは、numItemsプロパティに10と入力した場合、結果は1|0|1|0|1|0|1|0|1|0という順序になることを意味します。

インスタンスータの使い方を示す詳細なサンプルについては、[12.4「チュートリアル: 動的コンテンツを使用したリストの作成」](#)をご覧ください。

## 注記



ラインテンプレートのプロパティをリンクする  
リンクの規則を次に示します。

- ▶ ラインテンプレート同士のプロパティをリンクすることはできません。
- ▶ インスタンスエータの外部からそのラインテンプレートにリンクすることはできません。
- ▶ ラインテンプレートから対応するインスタンスエータにリンクできます。

## 8.1.2.7. アニメーションの追加



## アニメーションの追加

曲線の詳細や曲線プロパティの記述については、[13.12.2.2「アニメーション」](#)をご覧ください。

前提条件:

- コンテンツエリアにビューが表示されていること。

## ステップ 1

[ツールボックス]から基本ウィジェットの1つをビューにドラッグします。

## ステップ 2

[ツールボックス]からアニメーションをドラッグし、追加したウィジェットにドロップします。

## ステップ 3

[アニメーション]エディターに移動し、[アニメーション化プロパティ]の横にある+をクリックします。

メニューが展開されます。

## ステップ 4

[アニメーションプロパティ]の下でアニメーション化するプロパティを選択し、[アニメーション曲線]の下で該当する曲線を選択します。

## ステップ 5

[承認]をクリックします。

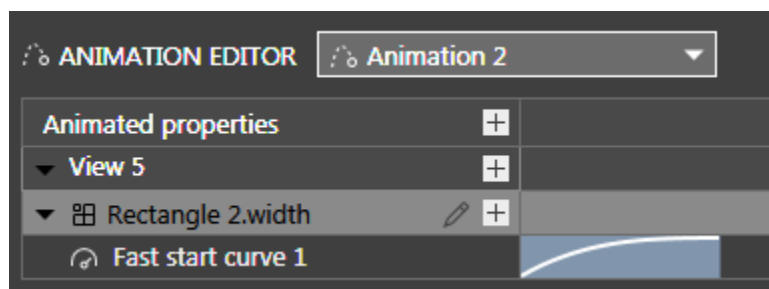


図8.2 サンプル曲線が表示された[アニメーション]エディター

### ステップ 6

基本ウィジェットを選択し、Conditional scriptタイプのユーザー定義プロパティを追加します。詳細については、[8.2.5「ウィジェットへのユーザー定義プロパティの追加」](#)をご覧ください。

### ステップ 7

条件スクリプトの横にある[値]列を選択し、{} をクリックします。

EB GUIDEスクリプトエディターが開きます。

### ステップ 8

[On trigger]セクションに次のEB GUIDEスクリプトを入力します。

```
function(v:arg0::bool)
{
  f:animation_play(v:this->"Animation 1")
}
```

最初に追加されるアニメーションのデフォルト名は[Animation 1]です。Step 2で追加したアニメーションに別の名前が付いている場合は、[On trigger]スクリプトで名前を書き換えてください。

### ステップ 9

シミュレーションを開始します。

リンク先のウィジェットのプロパティが、追加した曲線で指定されたとおりに徐々に変化します。

参考までに、アニメーションや曲線のプロパティは変更できます。

### ステップ 10

曲線の動作を変更するには、[プロパティ]コンポーネントでアニメーションまたは曲線のプロパティを編集します。

[アニメーション]エディターに曲線の動作のプレビューが表示されます。

アニメーションの具体的なサンプルについては、[12.5「チュートリアル: 画面内で楕円を移動する」](#)をご覧ください。

## 8.1.2.8. スクリプト曲線によるアニメーションの追加



### スクリプト曲線の出力の取得

前提条件:

- コンテンツエリアに、[メイン]状態マシンが表示されていること。
- [メイン]状態マシンに、初期状態とビュー状態が含まれていること。
- 初期状態にビュー状態への遷移があること。

### ステップ 1

ビューの名前をFirstViewに変更し、このビューを開きます。

### ステップ 2

[ツールボックス]からアニメーションをFirstView内にドラッグします。

### ステップ 3

[アニメーションエディター]で、[アニメーション化プロパティ]の横にある $\oplus$ をクリックし、次にFirstViewをクリックします。

メニューが展開されます。

### ステップ 4

FirstViewの下でyプロパティ、[スクリプト曲線]の順に選択します。

### ステップ 5

[承認]をクリックします。

アニメーションが[アニメーションエディター]に追加されます。

### ステップ 6

新しいアニメーションの名前をScriptCurveMonitoringに変更します。

### ステップ 7

[プロパティ]コンポーネントで、[デフォルトウィジェットプロパティ]に移動し、curveプロパティの横にある $\{\}$  をクリックします。EB GUIDEスクリプトエディターが表示されます。

次のEB GUIDEスクリプトを入力します。

```
function(v:diff::int, v:t_anim::int)
{
  f:trace_string("Diff : "+ f:int2string(v:diff) + " t_anim: " + f:int2string(v:t_anim))
  0::int
}
```

[承認]をクリックします。

### ステップ 8

[プロパティ]コンポーネントで[ユーザー定義プロパティ]カテゴリに移動し、 $\oplus$ をクリックします。

メニューが展開されます。

### ステップ 9

メニューで、Conditional scriptを選択します。

Conditional scriptタイプのユーザー定義プロパティがアニメーションに追加されます。名前をPlayAnimationに変更します。

### ステップ 10

PlayAnimationプロパティの横にある $\{\}$  をクリックします。

EB GUIDEスクリプトエディターが表示されます。

### ステップ 11

[On trigger]セクションに次のEB GUIDEスクリプトを入力します。

```
function(v:arg0::bool)
{
  f:animation_play(v:this)
  false
}
```




## EB GUIDEモデルの保存およびテスト


前提条件:

- 前のセクションの手順を完了していること。

### ステップ 1

プロジェクトを保存するには、コマンドエリアで をクリックします。

### ステップ 2

シミュレーションを開始するには、コマンドエリアで をクリックします。

EB GUIDE Monitorで、[ログ]コンポーネントを参照します。v:diffは、アニメーションが16ミリ秒ごとに実行されることを示しています。v:t\_animは、開始してからこの時点までにアニメーションが実行された時間を示しています。以下の図をご覧ください。

☞ traceString 'Diff : 0 t_anim: 0'	12:21:07.779
☞ traceString 'Diff : 16 t_anim: 16'	12:21:07.779
☞ traceString 'Diff : 16 t_anim: 32'	12:21:07.779
☞ traceString 'Diff : 17 t_anim: 49'	12:21:07.779
☞ traceString 'Diff : 19 t_anim: 68'	12:21:07.779
☞ traceString 'Diff : 16 t_anim: 84'	12:21:07.779
☞ traceString 'Diff : 16 t_anim: 100'	12:21:07.779

図8.3 EB GUIDE Monitorメッセージ

## 8.1.2.9. アルファマスクの追加



## アルファマスクの追加

アルファマスクの詳細については、[13.12.2.1「アルファマスク」](#)をご覧ください。

**前提条件:**

- \$GUIDE\_PROJECT\_PATH/<project name>/resourcesディレクトリにイメージが含まれていること。
- コンテンツエリアにビューが表示されていること。

**ステップ 1**

[ツールボックス]からアルファマスクをドラッグし、ビューにドロップします。

**ステップ 2**

[プロパティ]コンポーネントに移動し、imageドロップダウンリストボックスからイメージを選択します。

**注記****サポートされているアルファマスク用イメージファイル形式**

使用可能なイメージファイル形式は、レンダラーの実装によって異なります。レンダラー (OpenGL ESバージョン2.0以降向け)では、.pngファイルおよび .jpgファイルがサポートされます。RGBイメージは、グレースケールイメージに変換されてからアルファマスクとして使用されます。グレースケールイメージはそのまま使用されます。イメージのアルファチャネルは無視されます。

アルファマスク機能は9-patchイメージには適用されません。9-patchイメージは PNGおよび JPEGファイル形式と同じ方法で処理されます。

**ステップ 3**

[ツールボックス]にある基本ウィジェットのいずれかを子ウィジェットとしてアルファマスクに追加します。

アルファチャネル(子ウィジェットのオパシティ)は、アルファマスクを使って制御されます。

## 8.1.3. ビューへの3Dウィジェットの追加

### 8.1.3.1. ビューへのシーングラフの追加

**ビューへのシーングラフの追加**

制限事項と推奨事項については、[6.1.2「3Dグラフィックファイルの設定」](#)をご覧ください。

**前提条件:**

- 3Dグラフィックファイルが使用可能になっていること。ファイルに、カメラ、光源、およびメッシュと少なくとも1つの材質を含む1つのオブジェクトが含まれていること。サポートされている3Dグラフィックファイル形式については、[6.1.1「サポートされている3Dグラフィック形式」](#)をご覧ください。
- コンテンツエリアにビューが表示されていること。



#### ステップ 1

[ツールボックス]からシーングラフをビューにドラッグしてドロップします。

ビューに空の矩形が表示されます。

#### ステップ 2

[プロパティ]コンポーネントで[ファイルのインポート]をクリックします。

ダイアログが開きます。

#### ステップ 3

3Dグラフィックファイルが格納されているディレクトリに移動します。

#### ステップ 4

3Dグラフィックファイルを選択します。

#### ステップ 5

[開く]をクリックします。

インポートが開始します。ダイアログが開きます。

#### ステップ 6

[OK]をクリックします。

ビューに3Dグラフィックが表示されます。インポートしたウィジェットツリーが、シーングラフを親ノードにして[ナビゲーション]コンポーネントに表示されます。インポートされる3Dシーンにアニメーションがある場合は、リニアキー値補間整数曲線またはリニアキー値補間浮動小数点数曲線が追加されます。EB GUIDE Studioでそれらの曲線の基礎となるキー値ペアを変更することはできません。

#### ティップ



#### 複数のインポート

複数の3Dグラフィックを1つのシーングラフにインポートすることもできます。

インポート後、複数の3Dグラフィックは重ねてレンダリングされます。3Dオブジェクトを個別に表示するには、RootNodeのvisibleプロパティを使用します。

## 8.1.4. ビューに.psdファイルをインポートする



#### ビューに.psdファイルを追加する

バックグラウンド情報については、[6.14「Photoshopファイル形式のサポート」](#)をご覧ください。

前提条件:

- .psdファイルが\$GUIDE\_PROJECT\_PATH/<project name>/resourcesにあること。
- コンテンツエリアにビューが表示されていること。

### ステップ 1

[アセット]コンポーネントで、対応するフォルダーを選択します。

### ステップ 2

プレビューエリアで、.psdファイルをドラッグして、コンテンツエリアにドロップします。

インポートステータスを示すメッセージが表示されます。

### ステップ 3

[OK]をクリックします。

インポートに成功した場合、[ナビゲーション]コンポーネントに新しいウィジェットツリーが表示されます。最上位要素は、.psdファイルの名前が反映されたコンテナです。

\$GUIDE\_PROJECT\_PATH/<project name>/resourcesフォルダー内にサブフォルダーが作成されます。ここには、抽出されたイメージのすべてが含まれます。

#### 注記



#### 複数行

テキストレイヤーに1行を超えるテキストが含まれている場合は、[複数行] ウィジェット機能を追加してプロパティを適宜編集する必要があります。詳細については、[8.3.1「ウィジェット機能の追加」](#)および[13.13.1.4「複数行」](#)をご覧ください。

## 8.1.5. .psdファイルからイメージを抽出する



### .psdファイルからイメージを抽出する

イメージをインポートせずに .psdファイルから抽出すると、ウィジェットツリーは作成されません。バックグラウンド情報については、[6.14「Photoshopファイル形式のサポート」](#)をご覧ください。

前提条件:

- .psdファイルが\$GUIDE\_PROJECT\_PATH/<project name>/resourcesまたはサブフォルダー内にあること。

### ステップ 1

[アセット]コンポーネントで、対応するフォルダーを選択します。

### ステップ 2

.psdファイルを右クリックし、[.psdファイルからイメージを抽出]を選択します。

インポートステータスを示すメッセージが表示されます。

### ステップ 3

[OK]をクリックします。

\$GUIDE\_PROJECT\_PATH/<project name>/resourcesフォルダーに、抽出されたすべてのイメージが含まれるサブフォルダーが作成されます。このサブフォルダーの名前は、.psdファイルの名前を反映したものになっています。

## 8.1.6. IBLファイルのインポート

バックグラウンド情報については、[6.11「イメージベースドライティング」](#)をご覧ください。



### IBLファイルのインポート

IBLファイルをインポートするには、まずIBLファイルを .ebibl形式に変換する必要があります。この処理はIBLGeneratorを使用して行います。

前提条件:

- EB GUIDEプロジェクトが作成されます。
- IBLファイルが .pfmまたは .hdr形式であること。
- 管理者権限があること。
- シーングラフウィジェットがEB GUIDEモデルに追加されていること。
- 3Dファイルがインポートされていること。

#### ステップ 1

管理者としてコマンドラインプロンプトを開きます。

#### ステップ 2

IBLGeneratorのインストールパスに移動します。これはtoolsディレクトリ内のEB GUIDE Studioインストールディレクトリ(\$GUIDE\_INSTALL\_PATH\tools\IBLGenerator)にあります。

#### ステップ 3

ファイルを .ebibl形式に変換するコマンドを入力します。そのコマンドは次のようになります。

```
IBLGenerator.exe -i yourfile.hdr -o yourfile.ebibl -p latlong -q 1
```

- ▶ -i: 入力ファイル名
- ▶ -o: 出力ファイル名
- ▶ -p: パラメタリゼーションタイプ。その他のタイプには立方体および球体があります。
- ▶ -q: 品質レベル。品質レベルは、1が最低で10が最高の品質となります。品質レベルが高くなると、かなりの処理時間が必要になります。

.ebiblファイルは、指定したディレクトリに配置されます。

## ティップ

**IBLGeneratorヘルプ**

IBLGeneratorのオプションの一覧を確認するには、`-h`パラメータを指定して実行します。

```
IBLGenerator.exe -h
```

ステップ 4

.ebiblファイルをEB GUIDEモデルのリソースディレクトリにコピーします。これで、.ebiblファイルをシーングラフノードで使用できるようになります。

ステップ 5

EB GUIDE Studioの[ツールボックス]コンポーネントで、[3Dウィジェット]からイメージベースドライトをシーングラフノード内にドラッグします。

ステップ 6

[プロパティ]コンポーネントで、iblプロパティの横にある.ebiblファイルを選択します。

IBLファイルのインポートが終了します。

## ティップ

**IBLのための最良の結果**

最良の結果を得られるよう、イメージベースドライトのプロパティを適合させ、PBR GGX材質またはPBR Phong材質を使用します。

イメージベースドライティングのみでシーンを照らすには、その他すべての光源を無効にします。

## 8.1.7. ビューからのウィジェットの削除



## ビューからのウィジェットの削除

前提条件:

- EB GUIDEモデルにウィジェットが含まれていること。

ステップ 1

[ナビゲーション]コンポーネントで、ウィジェットを右クリックします。

ステップ 2

コンテキストメニューの[削除]をクリックします。

ウィジェットが削除されます。

## ティップ



## コンテンツエリアからのウィジェットの削除

コンテンツエリアでウィジェットを選択してDeleteキーを押すことで、ウィジェットを削除することもできます。

## 8.2. ウィジェットプロパティの操作

### 8.2.1. ウィジェットの配置



## ウィジェットの配置

ウィジェットの配置とは、ウィジェットのxおよびyプロパティを調整することです。xとyの値が両方とも0になっている原点は、親ウィジェットの左上隅です。

前提条件:

- コンテンツエリアにビューが表示されていること。
- ビューにウィジェットが含まれていること。

#### ステップ 1

ウィジェットを選択します。

[プロパティ]コンポーネントに、選択したウィジェットのプロパティが表示されます。

#### ステップ 2

ウィジェットのx座標を定義するには、xテキストボックスに値を入力します。

#### ステップ 3

ウィジェットのy座標を定義するには、yテキストボックスに値を入力します。

#### ステップ 4

テキストボックスの外側をクリックします。

コンテンツエリアで、入力した場所にウィジェットが表示されます。

## ティップ



## 別の方法

ウィジェットを目見当で配置するには、コンテンツエリアのウィジェットを選択し、マウスを使って移動します。

## 8.2.2. ウィジェットのサイズの変更



## ウィジェットのサイズの変更

## 前提条件:

- コンテンツエリアにビューが表示されていること。
- ビューにウィジェットが含まれていること。

## ステップ 1

ウィジェットを選択します。

[プロパティ]コンポーネントに、選択したウィジェットのプロパティが表示されます。

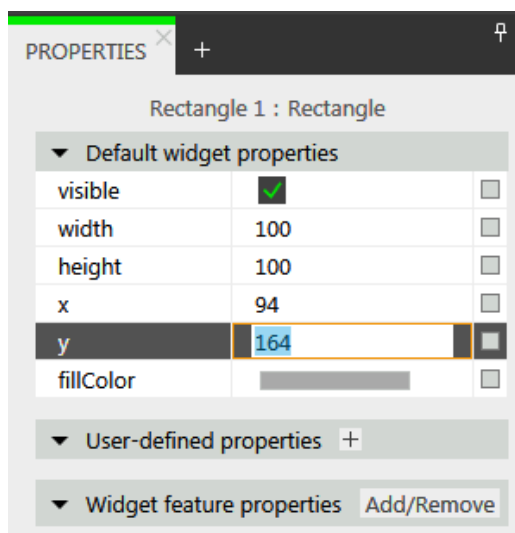


図8.4 イメージのプロパティ

## ステップ 2

ウィジェットの高さを定義するには、heightテキストボックスに値を入力します。

## ステップ 3

ウィジェットの幅を定義するには、widthテキストボックスに値を入力します。

**ステップ 4**

テキストボックスの外側をクリックします。

コンテンツエリアで、入力したサイズでウィジェットが表示されます。

**注記****負の値**

heightおよびwidthプロパティに負の値を使用しないでください。EB GUIDE Studioは負の値を0として扱うため、各ウィジェットが描出されなくなってしまいます。

**ティップ****別の方法**

ウィジェットのサイズを目見当で変更するには、コンテンツエリアのウィジェットを選択し、角の1つをマウスを使ってドラッグします。

### 8.2.3. ウィジェットプロパティ間のリンク設定



#### ウィジェットプロパティ間のリンク設定

2つのウィジェットプロパティが常に同じ値を持つようにするために、2つのウィジェットプロパティをリンクできます。例えば、次の手順は、四角形のwidthプロパティをビューのwidthプロパティとリンクする方法を示しています。

同一ビュー内にあるウィジェットのプロパティのみリンクできます。

インスタンシエータの子ウィジェットのプロパティにリンクすることはできません。

前提条件:

- EB GUIDEモデルにビューステートが含まれていること。
- ビューに四角形が含まれていること。
- 四角形のwidthプロパティがスクリプト値ではないこと。

**ステップ 1**

四角形をクリックします。

[プロパティ]コンポーネントに、四角形のプロパティが表示されます。

**ステップ 2**

[プロパティ]コンポーネントでwidthプロパティに移動し、プロパティの横にある■ボタンをクリックします。

メニューが展開されます。

### ステップ 3

メニューの[ウィジェットプロパティへのリンクを追加]をクリックします。

ダイアログが開きます。

### ステップ 4

ダイアログ内で、ビューに移動し、そのwidthプロパティをクリックします。

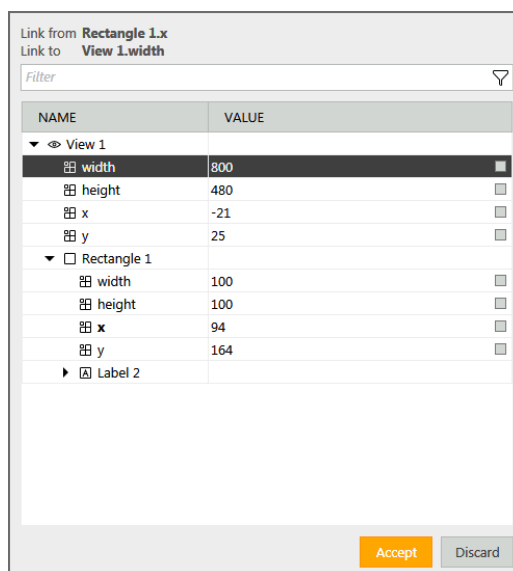



図8.5 ウィジェットプロパティ間のリンク設定

### ステップ 5


[承認]をクリックします。

ダイアログが閉じられます。 ボタンがwidthプロパティの横に表示されます。これは、四角形のwidthプロパティがビューのwidthプロパティにリンクされたことを示します。ビューの幅を変更するたびに四角形の幅が変更され、逆に四角形の幅を変更するたびにビューの幅が変更されます。

#### 注記




リンクソースとリンクターゲット

 ボタンはリンクソースの横にのみ表示されます。リンクターゲットには表示されません。

#### ティップ



リンクの削除

リンクを削除するには、 ボタンをもう一度クリックします。開いたメニューで、[リンクの削除]をクリックします。

## 8.2.4. ウィジェットプロパティとデータプールアイテムのリンク設定





### ウィジェットプロパティとデータプールアイテムのリンク設定

ウィジェットプロパティとデータプールアイテムが常に同じ値を持つようにするために、ウィジェットプロパティとデータプールアイテムをリンクできます。例えば、次の手順は、イメージのimageプロパティと新しいデータプールアイテムをリンクする方法を示しています。

前提条件:


- EB GUIDEモデルにビューステートが含まれていること。
- ビューにイメージが含まれていること。
- イメージのimageプロパティがスクリプト値ではないこと。

#### ステップ 1

イメージをクリックします。

[プロパティ]コンポーネントに、イメージのプロパティが表示されます。

#### ステップ 2

[プロパティ]コンポーネントでimageプロパティに移動し、プロパティの横にある  ボタンをクリックします。

メニューが展開されます。

#### ステップ 3

メニューの[データプールアイテムへのリンクを追加]をクリックします。

ダイアログが開きます。

#### ステップ 4

新しいデータプールアイテムを追加するには、テキストボックスに名前を入力します。

#### ステップ 5

[データプールアイテムを追加]をクリックします。

#### ステップ 6

[承認]をクリックします。

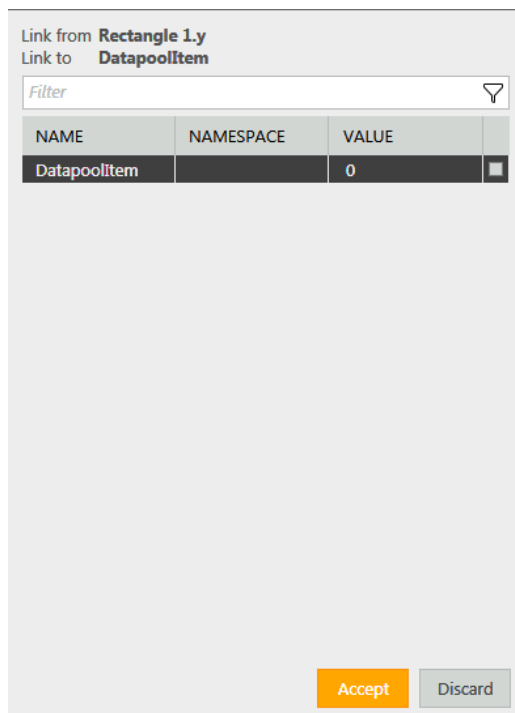


図8.6 データプールアイテムとのリンク設定

新しいデータプールアイテムが追加されます。

#### ステップ 7

ダイアログが閉じられます。■ボタンがimageプロパティの横に表示されます。これは、imageプロパティがデータプールアイテムにリンクされたことを示します。イメージを変更するたびにデータプールアイテムが変更され、逆にデータプールアイテムを変更するたびにイメージが変更されます。

#### 注記



リンクソースとリンクターゲット

■ボタンはリンクソースの横にのみ表示されます。リンクターゲットには表示されません。

#### ティップ



リンクの削除

リンクを削除するには、■ボタンをもう一度クリックします。開いたメニューで、[リンクの削除]をクリックします。

## 8.2.5. ウィジェットへのユーザー定義プロパティの追加



### ウィジェットへのユーザー定義プロパティの追加

前提条件:

- EB GUIDEモデルにビューステートが含まれていること。
- ビューにウィジェットが含まれていること。

#### ステップ 1

ウィジェットを選択します。

[プロパティ]コンポーネントに、選択したウィジェットのプロパティが表示されます。

#### ステップ 2

[プロパティ]コンポーネントで[ユーザー定義プロパティ]カテゴリに移動し、+をクリックします。

メニューが展開されます。

#### ステップ 3

メニューでユーザー定義プロパティのタイプをクリックします。

選択したタイプの新しいウィジェットプロパティがウィジェットに追加されます。

#### ステップ 4

プロパティの名前を変更します。

### 8.2.5.1. タイプのユーザー定義プロパティの追加 `Function () : bool`



### タイプのユーザー定義プロパティの追加 `Function () : bool`

`Function () : bool`タイプのプロパティは、パラメータを持たない、ブール値を返す関数です。この関数をEB GUIDEスクリプトで呼び出すには、ウィジェットプロパティの後に引数リストを指定します。

前提条件:

- EB GUIDEモデルにビューステートが含まれていること。
- ビューにウィジェットが含まれていること。

#### ステップ 1

ウィジェットを選択します。

[プロパティ]コンポーネントに、選択したウィジェットのプロパティが表示されます。

#### ステップ 2

[プロパティ]コンポーネントで[ユーザー定義プロパティ]カテゴリに移動し、+をクリックします。

メニューが展開されます。

#### ステップ 3

メニューで `Function () : bool` をクリックします。

`Function () : bool` タイプの新しいウィジェットプロパティがウィジェットに追加されます。

#### ステップ 4

プロパティの名前を変更します。

#### ステップ 5

データプールアイテムの横にある `[値]` 列を選択し、`{}` をクリックします。

EB GUIDE スクリプトエディターが開きます。

#### ステップ 6

EB GUIDE スクリプトを使用して新しい関数の動作を定義します。

#### ステップ 7

`[承認]` をクリックします。



#### 例 8.2

タイプのプロパティの呼び出し `Function () : bool`

EB GUIDE モデルには、`Background color` という四角形があります。その四角形に `Function () : bool` タイプのプロパティを追加しました。このプロパティは `change` と呼ばれます。

EB GUIDE モデルの EB GUIDE スクリプトコードでは、次のようにプロパティでスクリプトを呼び出せます。

```
"Background color".change()
```

## 8.2.6. ユーザー定義プロパティの名前の変更



### ユーザー定義プロパティの名前の変更

前提条件:

- EB GUIDE モデルに、ユーザー定義プロパティを持つウィジェットが含まれていること。

#### ステップ 1

[ナビゲーション] コンポーネントで、ユーザー定義プロパティを持つウィジェットを選択します。

#### ステップ 2

[プロパティ] コンポーネントで、プロパティ名を右クリックします。

メニューが展開されます。

### ステップ 3

メニューで[名前の変更]をクリックします。

### ステップ 4

プロパティの名前を入力します。

### ステップ 5

Enterキーを押します。

## 8.2.7. タイプリストのプロパティの編集



### タイプリストのプロパティの編集


タイプリストのプロパティの詳細については、[13.3.12「リスト」](#)をご覧ください。

リソース管理の詳細については、[6.15「リソース管理」](#)および[6.5「グラフィカルユーザーインターフェースのコンポーネント」](#)をご覧ください。

前提条件:

- EB GUIDEモデルにビューステートが含まれていること。
- ビューステートにウィジェットが含まれていること。
- ウィジェットにタイプリストのプロパティがあること。

### ステップ 1

[プロパティ]コンポーネントで、タイプリストのプロパティを選択し、その横にある  ボタンをクリックします。

[編集]ダイアログが開きます。

### ステップ 2

リストに新しいエントリーを追加するには、[追加]ボタンをクリックします。

新しい行が追加されます。

### ステップ 3

値を編集するには、[値]列にある対応する行をクリックします。

#### 注記



#### [アセット]コンポーネント

プロパティがタイプフォントリスト、IBLリスト、イメージリスト、またはメッシュリストのものである場合、[編集]ダイアログには追加の[アセット]コンポーネントが表示されます。[アセット]コンポーネントからアセットを[値]列の対応する行にドラッグアンドドロップできます。

### ステップ 4

リストの編集が終了したら、[承認]ボタンをクリックします。

ダイアログが閉じられます。

## 8.2.8. マルチフォントサポートの管理

詳しくは、[6.15.1.2「マルチフォントサポート」](#)をご覧ください。



### タイプフォント向けマルチフォントサポートの追加

前提条件:


- `$GUIDE_PROJECT_PATH/resources`には複数のフォントが用意されています。
- タイプフォントのデータプールアイテムのプロパティが存在します。

#### ステップ 1

マルチフォントサポートを追加するには、次の操作を行います。

- ▶ マルチフォントサポートをウィジェットプロパティに追加する場合は、[プロパティ]コンポーネントに移動します。
- ▶ マルチフォントサポートをデータプールアイテムに追加する場合は、[データプール]コンポーネントに移動します。

#### ステップ 2

データプールアイテムのプロパティの横にある  ボタンをクリックします。


メニューが展開されます。

#### ステップ 3

メニューの[マルチフォントサポートの追加]をクリックします。

プロパティまたはデータプールアイテムの下に表が表示されます。この表には、デフォルトのマルチフォント値が1つあります。

#### ステップ 4

新しいマルチフォント値を追加するには、 ボタンをクリックします。

新しい行が追加され、デフォルトフォントに基づいた値が設定されます。

#### ステップ 5

この行では、以下の編集および定義ができます。

- ▶ [優先度]列で、フォントエントリの優先度を定義します。
- ▶ [フォント]列で、サイズとフォントを定義します。
- ▶ [範囲]列で、選択されているフォントの影響を受けるUnicode文字を定義します。


### ステップ 6

必要なすべてのフォント値が追加されるまで、手順2~5を繰り返します。

#### ティップ



#### マルチフォントサポートの削除

マルチフォントサポートを削除するには、 ボタンをもう一度クリックします。開いたメニューで、[マルチフォントサポートの削除]をクリックします。



#### タイプフォントリスト向けマルチフォントサポートの追加

タイプリストのプロパティの操作方法については、[8.2.7「タイプリストのプロパティの編集」](#)をご覧ください。

#### 前提条件:


- \$GUIDE\_PROJECT\_PATH/resourcesには複数のフォントが用意されています。
- タイプフォントリストのプロパティまたはデータプールアイテムが存在し、そこに少なくとも1つのフォントリストエントリーがあります。

### ステップ 1

マルチフォントサポートを追加するには、次の操作を行います。


- ▶ マルチフォントサポートをウィジェットプロパティに追加する場合は、[プロパティ]コンポーネントに移動します。
- ▶ マルチフォントサポートをデータプールアイテムに追加する場合は、[データプール]コンポーネントに移動します。

### ステップ 2

プロパティまたはデータプールアイテムを選択し、その横にある  ボタンをクリックします。

[編集]ダイアログが開きます。

### ステップ 3

タイプフォントのエントリーの横にある  ボタンをクリックします。


メニューが展開されます。

### ステップ 4

メニューの[マルチフォントサポートの追加]をクリックします。

プロパティまたはデータプールアイテムの下に表が追加されます。この表には、デフォルトのマルチフォント値が1つあります。

### ステップ 5

新しいマルチフォント値を追加するには、 ボタンをクリックします。

新しい行が追加され、デフォルトフォントに基づいた値が設定されます。

### ステップ 6

この行では、以下の編集および定義ができます。

- ▶ [優先度]列で、フォントエントリーの優先度を定義します。
- ▶ [フォント]列で、サイズとフォントを定義します。
- ▶ [範囲]列で、選択されているフォントの影響を受けるUnicode文字を定義します。


### ステップ 7

必要なすべてのフォント値が追加されるまで、手順3~6を繰り返します。

#### ティップ



#### マルチフォントサポートの削除

マルチフォントサポートを削除するには、 ボタンをもう一度クリックします。開いたメニューで、[マルチフォントサポートの削除]をクリックします。

### ステップ 8

リスト内のエントリーの編集が終了したら、[承認]ボタンをクリックします。

ダイアログが閉じられます。

## 8.2.9. ウィジェットの順序および可視性の管理

EB GUIDE Studioでは、ウィジェットの順序、レイヤー、または可視性を定義するなど、いくつかの操作が可能です。以下の操作が可能であり、それらを適用できるかどうかはユースケースによって異なります。

ビューにおいて他のウィジェットよりも相対的に上位にあるウィジェットを変更する場合は、[ナビゲーション]ツリーを使用します。[ナビゲーション]ツリーでは、ウィジェットの順序により、どのウィジェットが最上位に表示されるかが決まります。同じ分岐上にあるウィジェットどうしでは、ツリーのより高い位置にあるウィジェットが下に表示されます。異なる分岐上にあるウィジェットどうしでは、より高いレベルのウィジェットが上に表示されます。

ポップアップを他のどの要素よりも上に表示する場合は、動的ステートマシンを使用します。手順については、[12.1「チュートリアル: 動的ステートマシンの追加」](#)をご覧ください。

ユーザーの選択内容に従って変化する複雑なメニューがある場合は、Child visibility selectionを使用します。この機能により、ウィジェットの子ウィジェットの可視性を制御できます。この機能は、影響を受けるウィジェットのVisibilityプロパティを上書きします。この機能の使用法としては、次の2つが想定されます。

- ▶ 1つの子ウィジェットを可視化します。このユースケースでは、ウィジェットツリーにウィジェットのインデックスのみが必要です。手順については、[「1つの子ウィジェットの可視化」](#)をご覧ください。
- ▶ 複数の子ウィジェットを可視化します。このユースケースでは、子ウィジェットのグループを定義し、Valueを識別する1つのグループを指定する必要があります。手順については、[「複数の子ウィジェットの可視化」](#)をご覧ください。





### 1つの子ウィジェットの可視化

前提条件:

- EB GUIDEプロジェクトがEB GUIDE Studioで開かれていること。
- EB GUIDEモデルにウィジェットが含まれていること。

#### ステップ 1

[子の可視性選択]ウィジェット機能を親ウィジェットに追加します。[8.3「ウィジェット機能を追加してウィジェットを拡張する」](#)をご覧ください。

#### ステップ 2

[ナビゲーション]コンポーネントで、子ウィジェットがどの位置にあるか確認します。0 が最初の位置になります。

#### ステップ 3

[プロパティ]コンポーネントで、`containerIndex`行を選択し、入力フィールドに子ウィジェットの位置を入力します。

#### ステップ 4

子ウィジェットの位置を`containerIndex`入力フィールドに入力します。

これで、この子ウィジェットは表示される唯一の子ウィジェットになります。



### 複数の子ウィジェットの可視化


前提条件:

- EB GUIDEプロジェクトがEB GUIDE Studioで開かれていること。
- EB GUIDEモデルにウィジェットが含まれていること。

#### ステップ 1

[子の可視性の選択]ウィジェット機能を追加します。[8.3「ウィジェット機能を追加してウィジェットを拡張する」](#)をご覧ください。

#### ステップ 2

[プロパティ]コンポーネントで、`containerMapping`を選択してからをクリックします。

エディターが開きます。

#### ステップ 3

次の手順では、ウィジェットのグループと子ウィジェットとの間のマッピングを定義します。

[インデックス]列には、子ウィジェットのインデックスが含まれます。[値]列に、子ウィジェットのマッピング先にするグループを入力します。

#### ステップ 4

[承認]をクリックします。

#### ステップ 5

containerIndexを選択し、入力フィールドには、表示する必要がある、グループのValueを入力します。

これで、このグループのウィジェットが表示されます。このグループにマッピングされていないウィジェットは表示されません。

## 8.3. ウィジェット機能を追加してウィジェットを拡張する

ウィジェット機能は、ウィジェットの外観と動作に関して別の機能を追加します。ウィジェット機能をウィジェットに追加することは、1つ以上のウィジェットプロパティを追加することを意味します。提供されるウィジェット機能は、ウィジェットのタイプによって異なります。

### 8.3.1. ウィジェット機能の追加



#### ウィジェット機能の追加

前提条件:

- EB GUIDEモデルにウィジェットが含まれていること。

#### ステップ 1

[ナビゲーション]コンポーネントでウィジェットをクリックします。

[プロパティ]コンポーネントに、選択したウィジェットのプロパティが表示されます。

#### ステップ 2

[プロパティ]コンポーネントで[ウィジェット機能プロパティ]カテゴリに移動し、[追加/削除]をクリックします。

[ウィジェット機能]ダイアログが表示されます。

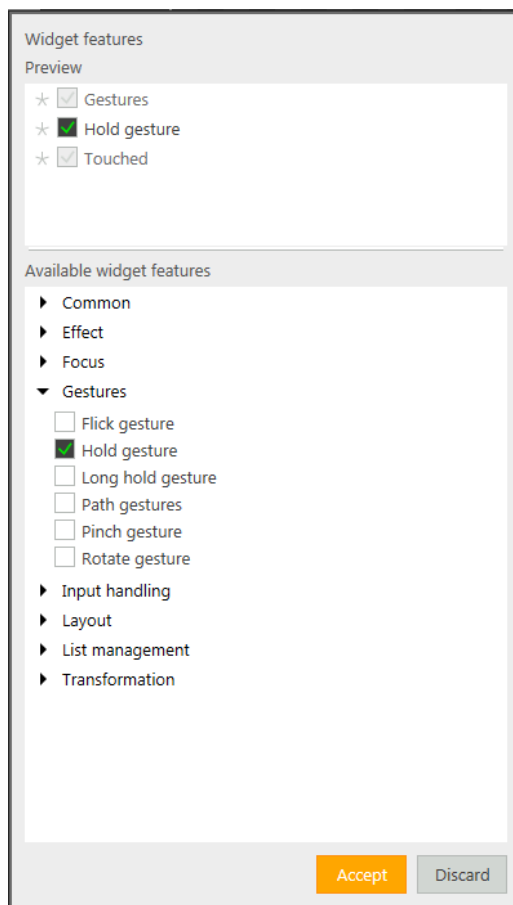


図8.7 ウィジェット機能ダイアログ

### ステップ 3

[使用可能なウィジェット機能]からカテゴリを展開し、追加するウィジェット機能を選択します。

選択したウィジェット機能とそれに合わせて自動的に有効化される依存ウィジェット機能の一覧が、[プレビュー]に表示されます。

[承認]をクリックします。

### ティップ



#### ウィジェット機能間の依存関係

ウィジェット機能の中には、他のウィジェット機能を必要とするものがあります。そのため、あるウィジェット機能をオンにすると、他のウィジェット機能が自動的に選択される場合があります。

例えば、[移動可能]というウィジェット機能を追加するとします。その場合、[タッチ]と[タッチ移動]というウィジェット機能も自動的に追加されます。

カテゴリ別に分類されたウィジェット機能の一覧については、[13.13「ウィジェット機能」](#)をご覧ください。

チュートリアルについては、[以下](#)をご覧ください。

- ▶ [12.3「チュートリアル:パスジェスチャーをモデル化する」](#)
- ▶ [12.4「チュートリアル:動的コンテンツを使用したリストの作成」](#)
- ▶ [12.2「チュートリアル:EB GUIDEスクリプトを使用したボタン動作のモデル化」](#)

### 8.3.2. ウィジェット機能の削除



#### ウィジェット機能の削除

前提条件:

- EB GUIDEモデルにウィジェットが含まれていること。
- 1つ以上のウィジェット機能がウィジェットに追加されていること。

#### ステップ 1

[ナビゲーション]コンポーネントでウィジェットをクリックします。

[プロパティ]コンポーネントに、選択したウィジェットのプロパティが表示されます。

#### ステップ 2

[プロパティ]コンポーネントで[ウィジェット機能プロパティ]カテゴリに移動し、[追加/削除]をクリックします。

[ウィジェット機能]ダイアログが表示されます。

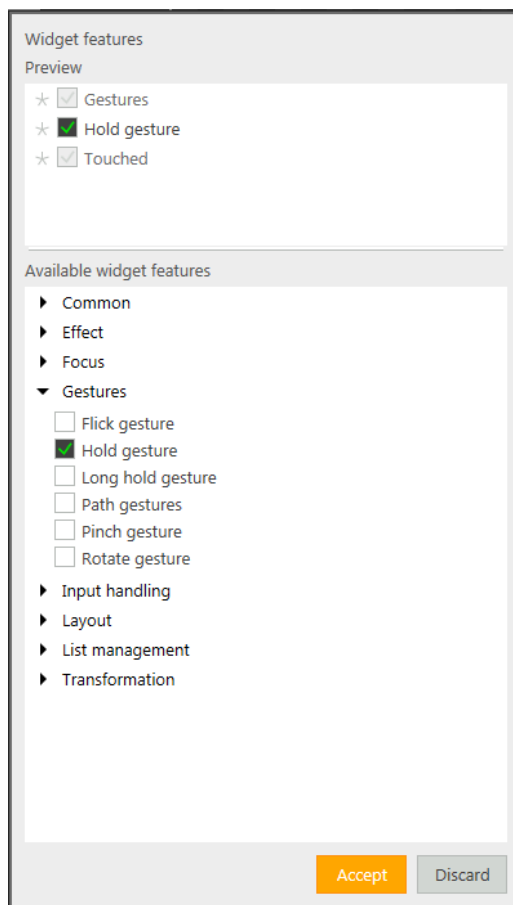


図8.8 ウィジェット機能ダイアログ

### ステップ 3

[プレビュー]で、削除するウィジェット機能を解除します。

[承認]をクリックします。

関連するウィジェット機能のプロパティが[プロパティ]コンポーネントから削除されます。

#### 注記



#### 依存関係を持つウィジェット機能の削除

依存関係により自動的に追加されたウィジェット機能は、自動的に削除されません。それらは直接削除できません。子のウィジェット機能を解除する前に、親のウィジェット機能を解除してください。

## 8.4. 言語サポートとの連携

ランタイム中にEB GUIDEモデルの言語を変更するには、言語サポートと言語依存テキストを追加します。

EB GUIDEモデルの言語については、[6.12.2「EB GUIDEモデルの言語」](#)をご覧ください。

### 8.4.1. EB GUIDEモデルへの言語の追加

注記



スキンのサポートは使用できません


データプールアイテムに言語サポートを定義した場合、同じアイテムにスキンのサポートを追加することはできません。



#### 言語の追加

次の手順は、EB GUIDEモデルに言語を追加する方法を示しています。

##### ステップ 1

をクリックします。

プロジェクトセンターが表示されます。

##### ステップ 2

ナビゲーションエリアで、[設定] > [言語]の順にクリックします。

使用できる言語が表示されます。リスト内の最初の言語は、常にデフォルト言語となります。

##### ステップ 3

コンテンツエリアで、[追加]をクリックします。

言語が表に追加されます。その言語では標準の言語設定が初期値として使用されます。

##### ステップ 4

言語の名前を入力します。

##### ステップ 5

[言語]ドロップダウンリストボックスから言語を選択します。

##### ステップ 6

[国]ドロップダウンリストボックスから国を選択します。

言語がEB GUIDEモデルに追加されます。プロジェクトエディターのコマンドエリアにある[言語]ドロップダウンリストボックスで、新しい言語を選択できます。

ランタイム中に言語を切り替えると、異なるデータプール値の効果を確認できます。詳しくは、[12.6「チュートリアル: データプールアイテムに言語依存テキストを追加する」](#)をご覧ください。

## 8.4.2. データプールアイテムに言語サポートを追加する



### データプールアイテムに言語サポートを追加する

次の手順は、EB GUIDEモデルのデータプールアイテムに言語サポートを追加する方法を示しています。


前提条件:

- EB GUIDEモデルにデータプールアイテムが含まれていること。
- 少なくとも2つの言語がモデルに追加されていること。

#### ステップ 1

プロジェクトエディターで、[データプール]コンポーネントに移動します。


#### ステップ 2

データプールアイテムの[値]プロパティの横にある  ボタンをクリックします。

メニューが展開されます。

#### ステップ 3

メニューの[Add language support]をクリックします。

ダイアログが閉じられます。[値]プロパティの横に  ボタンが表示されます。これは、言語サポートがこのデータプールアイテムに追加され、各言語に異なる値を定義できるようになったことを示します。

データプールアイテムに言語サポートが追加されました。これで、このデータプールアイテムで言語依存の値を定義できます。詳しくは、[12.6「チュートリアル: データプールアイテムに言語依存テキストを追加する」](#)をご覧ください。

## 8.4.3. 言語の削除

注記



デフォルト言語

リスト内の最初の言語は、常にデフォルト言語となり、削除できません。




### 言語の削除

前提条件:

- 少なくとも2つの言語がEB GUIDEモデルに追加されていること。

#### ステップ 1

をクリックします。

プロジェクトセンターが表示されます。

#### ステップ 2

ナビゲーションエリアで、[設定] > [言語]の順にクリックします。

使用できる言語が表示されます。

#### ステップ 3

コンテンツエリアで、言語を選択します。

#### ステップ 4

コンテンツエリアの[削除]をクリックします。

言語が表から削除されます。

## 8.5. スキンのサポートの操作

スキンのサポートを使用すると、モデルに異なるデータプール値を定義できます。このようにして、同じモデルに異なる外観(夜用モードと昼用モードなど)を定義できます。

スキンの詳細については、[6.19「スキン」](#)をご覧ください。

### 8.5.1. EB GUIDEモデルへのスキンの追加

注記




言語サポートは使用できません

データプールアイテムにスキンのサポートを定義した場合、同じアイテムに言語サポートを追加することはできません。



EB GUIDEモデルへのスキンの追加

#### ステップ 1

をクリックします。

プロジェクトセンターが表示されます。

#### ステップ 2

ナビゲーションエリアで[設定] > [スキン]の順にクリックします。



標準スキンが各モデルにデフォルトで追加されます。

#### ステップ 3

コンテンツエリアで[追加]をクリックします。

スキンが表に追加されます。

#### ステップ 4

スキンの名前を入力します。

新しいスキンがEB GUIDEモデルに追加されます。プロジェクトエディターのコマンドエリアにある[スキン]ドロップダウンリストボックスで、新しいスキンを選択できます。

## 8.5.2. データプールアイテムにスキンのサポートを追加する



### データプールアイテムにスキンのサポートを追加する

異なるデータプール値を定義することによってEB GUIDEモデルにさまざまな外観を定義するには、最初にデータプールアイテムにスキンのサポートを追加する必要があります。


前提条件:

- EB GUIDEモデルにデータプールアイテムが含まれていること。
- スキンがモデルに追加されていること。

#### ステップ 1

プロジェクトエディターで、[データプール]コンポーネントに移動します。


#### ステップ 2

データプールアイテムの[値]プロパティの横にある  ボタンをクリックします。

メニューが展開されます。

#### ステップ 3

メニューの[Add skin support]をクリックします。

ダイアログが閉じられます。[値]プロパティの横に  ボタンが表示されます。これは、スキンのサポートがこのデータプールアイテムに追加され、各スキンに異なる値を定義できるようになったことを示します。

#### ステップ 4

データプールアイテムに異なる値を定義するには、[データプール]コンポーネントでデータプールを選択します。

[プロパティ]コンポーネントに、EB GUIDEモデルで使用可能なすべてのスキンが含まれている表が表示されます。

#### ステップ 5

表内の各スキンに値を定義します。

### 8.5.3. スキンを切り替える



#### スキンを切り替える

前提条件:

- EB GUIDEモデルにデータプールアイテムが含まれていること。
- スキンがモデルに追加されていること。

#### ステップ 1

プロジェクトエディターで、コマンドエリアに移動します。

#### ステップ 2

ドロップダウンリストボックスでスキンを選択します。

コンテンツエリアに、このスキンで有効なデータプール値を持つモデルが表示されます。また、モデル実行モードでは特定のスキン値を持つモデルが表示されます。

### 8.5.4. スキンを削除する




#### スキンを削除する

前提条件:

- スキンがモデルに追加されていること。

#### ステップ 1

をクリックします。

プロジェクトセンターが表示されます。

#### ステップ 2

ナビゲーションエリアで[設定] > [スキン]の順にクリックします。

現在のプロジェクトのすべてのスキンが一覧表示されます。

#### ステップ 3

削除するスキンを選択して、[削除]をクリックします。

スキンが表から削除されます。

## 8.6. ビュー遷移のアニメーション化

### 8.6.1. 開始アニメーションの追加



#### 開始アニメーションの追加

このセクションでは、開始アニメーションをビューステートに追加する手順を説明します。この手順は、終了アニメーション、ポップアップオンアニメーション、およびポップアップオフアニメーションにも適用されます。詳細については、[section 6.2.2. “ビュー遷移のアニメーション”](#)および[13.12.1. 「ビュー」](#)をご覧ください。

前提条件:

- ビューステートとビューがEB GUIDEモデルに追加されます。
- [VTA]コンポーネントが開いていること。

#### ステップ 1

[VTA]コンポーネントで、+をクリックします。

#### ステップ 2

コンテキストメニューで、[開始アニメーション]をクリックします。

[アニメーション]エディターが開きます。

[プロパティ]コンポーネントに、追加した開始アニメーションのプロパティが表示されます。

#### ステップ 3

依存ビューで利用可能なすべてのウィジェットプロパティをアニメーション化します。

[アニメーション]エディターで開始アニメーションを定義します。

### 8.6.2. 変更アニメーションの追加



#### 変更アニメーションの追加

このセクションでは、変更アニメーションをビューステートまたはビューテンプレートに追加する手順を説明します。

前提条件:

- [VTA]コンポーネントが開いていること。
- [メイン]ステートマシンに2つのビューステートが含まれていること。

#### ステップ 1

[メイン]ステートマシンでView state 1を選択します。

#### ステップ 2

[VTA]コンポーネントで、+をクリックします。

#### ステップ 3

コンテキストメニューで、[変更アニメーション]を選択します。

ダイアログが開きます。

#### ステップ 4

View 2を選択します。

#### ステップ 5

[承認]をクリックします。


出力先ビューの名前が変更アニメーション名の横に表示されます。

[アニメーション]エディターが開きます。

[プロパティ]コンポーネントに、追加した変更アニメーションのプロパティが表示されます。

出力元ビューがプロジェクトエディターに表示されます。

#### ステップ 6

出力先ビューを編集するには、[VTA]コンポーネントで  をクリックします。

#### ステップ 7

アニメーションプロパティを追加するには、[アニメーション]エディターの+をクリックし、それぞれのカテゴリに追加します。

依存ビューで利用可能なすべてのウィジェットプロパティをアニメーション化します。

### 8.6.3. アニメーションの再配置



#### アニメーションの再配置

このセクションでは、ビューステートまたはビューテンプレートのアニメーションを再配置する手順を説明します。この手順は、終了アニメーション、変更アニメーション、ポップアップオンアニメーション、およびポップアップオフアニメーションにも適用されます。

前提条件:

- [VTA]コンポーネントが開いていること。
- 複数の開始アニメーションが追加されていること。

### ステップ 1

[MTA]コンポーネントで、最初に開始する開始アニメーションの横にある優先順位テキストボックスを選択します。

### ステップ 2

開始アニメーションの値を0に変更します。

開始アニメーションの順序が変更されます。編集した開始アニメーションが最初に開始され、次に続くすべてのエントリは1つずつ値が増えます。

値0は、アニメーションが最初に再生されることを意味します。その後の値は、後に続く順序でアニメーションが再生されることを意味します。

## 8.7. ウィジェットの再利用

テンプレートの詳細については、[6.22.4「ウィジェットテンプレート」](#)をご覧ください。

### 8.7.1. テンプレートの追加



#### テンプレートの追加

#### ステップ 1

[ナビゲーション]コンポーネントで[テンプレート]に移動し、+をクリックします。

メニューが展開されます。

#### ステップ 2

メニューでテンプレートのタイプをクリックします。

選択したタイプの新しいテンプレートが追加されます。コンテンツエリアにテンプレートが表示されます。

#### ステップ 3

テンプレートの名前を変更します。

#### ステップ 4

[プロパティ]コンポーネントでテンプレートのプロパティを編集し、テンプレートインターフェースを定義します。

## ティップ



## テンプレートのテンプレート

テンプレートのタイプを既存のテンプレートにすることができます。このため、EB GUIDEではテンプレートからテンプレートを作成できます。

## ティップ



## テンプレートのコピーと検索

既存のテンプレートをコピーして貼り付けるには、コンテキストメニューまたはCtrl+CキーとCtrl+Vキーを使用するという方法もあります。

EB GUIDEモデル内で特定のテンプレートを検索するには、テンプレートの名前を検索ボックスに入力するか、Ctrl+Fキーを使用します。テンプレートにジャンプするには、ヒットリスト内のテンプレートをダブルクリックします。

## 8.7.2. テンプレートインターフェースの定義



## テンプレートインターフェースの定義

## 前提条件:

- EB GUIDEモデルにテンプレートが含まれていること。

ステップ 1

テンプレートを選択します。

ステップ 2

テンプレートインターフェースにプロパティを追加するには、[プロパティ]コンポーネントでプロパティの横にある■ボタンをクリックします。メニューの[テンプレートインターフェースへの追加]をクリックします。

- アイコンがプロパティの横に表示されます。

ステップ 3

テンプレートインターフェースからプロパティを削除するには、プロパティの横にある■ボタンをクリックします。メニューの[テンプレートインターフェースから削除]を選択します。

- アイコンがプロパティの横に表示されなくなります。

## 注記



## インスタンスエータのテンプレート

インスタンスエータのテンプレートでは、インスタンスエータの子ウィジェットのプロパティをテンプレートインターフェースに追加できません。

### 8.7.3. テンプレートの使用



## テンプレートの使用

## 前提条件:

- コンテンツエリアにビューが表示されていること。
- [ツールボックス]で、ウィジェットテンプレートが使用可能になっていること。
- ウィジェットテンプレートのテンプレートインターフェースに、プロパティが1つ以上存在すること。

ステップ 1

[ツールボックス]からウィジェットテンプレートをビューにドラッグします。

テンプレートのインスタンスがビューに追加されます。[プロパティ]コンポーネントに、テンプレートインターフェースのプロパティが表示されます。

## ティップ





## テンプレートインターフェースを定義する


[プロパティ]コンポーネントにテンプレートインスタンスのプロパティが一切表示されない場合、テンプレートインターフェースにはプロパティがまったく追加されていません。この状況を変えるには、テンプレートインターフェースを定義します。

ステップ 2

[プロパティ]コンポーネントでテンプレートインスタンスのプロパティを編集します。

プロパティを編集すると、ボタンがボタンに変わります。

ステップ 3

プロパティの値をテンプレートの値にリセットするには、プロパティの横にあるボタンをクリックします。メニューの[テンプレート値へリセット]をクリックします。

### 8.7.4. テンプレートの削除



### テンプレートの削除

#### ステップ 1

[ナビゲーション]コンポーネントでテンプレートを右クリックします。

#### ステップ 2

コンテキストメニューの[削除]をクリックします。

テンプレートが削除されます。



## 9. データの処理

### NOTE



#### デフォルトのウィンドウレイアウト

すべての手順およびスクリーンショットでは、デフォルトのウィンドウレイアウトが使用されています。手順に従う場合は、EB GUIDE Studioウィンドウをレイアウト > **Reset to default layout**を選択することによって、デフォルトレイアウトにリセットすることをお勧めします。

### 9.1. イベントの追加



#### イベントの追加

##### Step 1

イベントコンポーネントの名前空間ドロップダウンリストボックスで、イベントの追加先となる名前空間を選択します。

##### Step 2

＋をクリックします。

イベントが表に追加されます。

##### Step 3

イベントの名前を変更します。

### TIP



#### イベントのコピーと検索

既存のイベントをコピーして貼り付けるには、コンテキストメニューまたは**Ctrl+C**キーと**Ctrl+V**キーを使用するという方法もあります。重複を防止するために、貼り付けたイベントはコピーしたイベントとは異なるイベントIDになります。

EB GUIDEモデル内で特定のイベントを検索するには、イベントの名前を検索ボックスに入力するか、**Ctrl+F**キーを使用します。イベントにジャンプするには、ヒットリスト内のイベントをダブルクリックします。

### 9.2. イベントへのパラメータの追加



#### イベントへのパラメータの追加

Prerequisite:

- イベントがEB GUIDEモデルに追加されます。

Step 1

イベントコンポーネントでイベントをクリックします。

Step 2

イベント表で、イベントの横にある+をクリックします。

Step 3

ドロップダウンリストボックスから、パラメータのタイプを選択します。

選択したタイプのパラメータがイベントに追加されます。

Step 4

パラメータの名前を変更します。


## 9.3. イベントへの対応

イベントに対応するには、イベントIDとイベントグループIDを使用します。EB GUIDE TFでは、ランタイムにイベントを送受信するためにIDが利用されます。



### イベントグループの追加

Step 1

をクリックします。

プロジェクトセンターが表示されます。

Step 2

ナビゲーションエリアで、設定 > イベントグループの順にクリックします。

Step 3

コンテンツエリアで追加をクリックします。

イベントグループが表に追加されます。

Step 4

イベントグループの名前を変更します。

Step 5

イベントグループIDを変更するには、IDをダブルクリックし、数値を入力します。



### EB GUIDE TFイベントへの対応

Prerequisite:

- イベントグループが追加されています。
- イベントがEB GUIDEモデルに追加されます。

#### Step 1

イベントコンポーネントでイベントをクリックします。

プロパティコンポーネントに、選択したイベントのプロパティが表示されます。

#### Step 2

Event IDテキストボックスにIDを挿入します。

#### Step 3

イベントコンポーネントに移動し、Groupドロップダウンリストボックスからイベントグループを選択します。

## 9.4. イベントへのキーのマッピング

イベントを発行するには、キーボードのキーを押すか、または対象デバイスのキーを押すか回転ボタンを使用します。

キーボードのキーを押すと反応するようにするには、モデルでこれらのキーイベントのマッピングを定義する必要があります。

各キーに対して、EB GUIDE GTFはC++ヘッダーファイルで数値コードを定義します。コード番号については、`$INSTALL_PATH$/platform/win64/include/gtf/inputmapper/KeyConstants.h`をご覧ください。



### イベントへのキーのマッピング

イベント処理の詳細については、[6.8「イベント処理」](#)および[13.5「イベント」](#)をご覧ください。

Prerequisite:

- イベントグループキー(ID 10)がプロジェクトセンター > 設定 > イベントグループに追加されていること。
- イベントが追加されています。

#### Step 1

KeyConstant.hファイルにマッピングするキーの16進コード番号を検索します。

#### Step 2

10進コード番号を計算します。

#### Step 3

イベントコンポーネントのグループ列で、キー(ID 10)を選択します。

#### Step 4

プロパティコンポーネントのEvent IDテキストボックスに、計算された10進コード番号を入力します。

これで、選択したキーがイベントにマップされます。



#### Example 9.1.

イベントへのF1キーのマッピング

F1の内部16進コード番号は12です。

12の10進コード番号は18です。

プロパティコンポーネントに移動し、Event IDテキストボックスに18と入力します。

これで、F1キーがイベントにマップされます。

#### NOTE



#### EB GUIDE Monitorで表示されないイベント

イベント自体はEB GUIDE Monitorに表示されませんが、イベントによってトリガーされるEB GUIDEスクリプトは反応します。

10進コード番号の詳細については、[13.5.1「キーイベントの10進コード」](#)をご覧ください。

## 9.5. イベントの削除



### イベントの削除

Prerequisite:

- イベントがEB GUIDEモデルに追加されます。

#### Step 1

イベントコンポーネントで、対応する名前空間を選択します。

#### Step 2

イベントを右クリックし、コンテキストメニューの削除を選択します。

イベントが削除されます。

## 9.6. データプールアイテムの追加



### データプールアイテムの追加

#### Step 1

データプールコンポーネントの名前空間ドロップダウンリストボックスで、データプールアイテムの追加先となる名前空間を選択します。

#### Step 2

＋をクリックします。

メニューが展開されます。

#### Step 3

メニューでデータプールアイテムのタイプをクリックします。

新しいデータプールアイテムがそのタイプで追加されます。データプールアイテムは内部で使用できるように準備されています。

#### Step 4

データプールアイテムの名前を変更します。

#### TIP



#### データプールアイテムのコピーと検索

既存のデータプールアイテムをコピーして貼り付けるには、コンテキストメニューまたは**Ctrl** + **C**キーと**Ctrl** + **V**キーを使用するという方法もあります。

EB GUIDEモデル内で特定のデータプールアイテムを検索するには、データプールアイテムの名前を検索ボックスに入力するか、**Ctrl** + **F**キーを使用します。データプールアイテムにジャンプするには、ヒットリスト内のデータプールアイテムをダブルクリックします。

## 9.7. リストタイプのデータプールアイテムの編集



### リストタイプのデータプールアイテムの編集


Prerequisite:

- リストタイプのデータプールアイテムが追加されていること。

#### Step 1

データプールコンポーネントで、リストタイプのデータプールアイテムをクリックします。

#### Step 2

Value列を選択し、をクリックします。

エディターが開きます。

#### Step 3

アイテムをリストデータプールアイテムに追加するには、追加をクリックします。

新しいエントリーが表に追加されます。

#### Step 4

新しいエントリーの値をValueテキストボックスに入力するか、コンボボックスから値を選択します。

#### Step 5

3と4の手順を繰り返し、リストにアイテムをさらに追加します。

#### Step 6

承認をクリックします。

リストのコンテンツは、Value列に表示されます。

## 9.8. プロパティのスクリプト値への変換




### プロパティのスクリプト値への変換

データプールアイテムやウィジェットのプロパティは、スクリプト値に変換したり、元の値に戻したりできます。データプールアイテムの値を変換するには、以下の操作を行います。ウィジェットプロパティでも同じ手順で変換できます。

Prerequisite:

- 1つのデータプールアイテムが追加されています。
- データプールアイテムは言語依存ではありません。
- データプールアイテムはスキン依存ではありません。
- データプールアイテムはリンクされていません。

#### Step 1

データプールコンポーネントで、データプールアイテムをクリックし、ボタンをクリックします。


メニューが展開されます。

#### Step 2

メニューからスクリプトに変換をクリックします。

データプールアイテムがスクリプト値に変換されます。

### Step 3


データプールアイテムの横にある値列を選択し、 をクリックします。

EB GUIDEスクリプトエディターが開きます。

### Step 4

EB GUIDEスクリプトを編集します。

### Step 5

データプールアイテムをプレーン値に変換するには、 ボタンをクリックします。

メニューが展開されます。

### Step 6

メニューで、プレーン値に変換をクリックします。

データプールアイテムがプレーン値に変換されます。

## 9.9. 外部通信の確立


EB GUIDEモデルとアプリケーションの間などに、外部通信を確立するには、通信コンテキストをEB GUIDEモデルに追加します。



### 通信コンテキストの追加

通信コンテキストを使うと、通信チャンネルを開くことができます。

### Step 1

 をクリックします。

プロジェクトセンターが表示されます。

### Step 2

ナビゲーションエリアで、設定 > 通信コンテキストの順にクリックします。

### Step 3

コンテンツエリアで追加をクリックします。

通信コンテキストが表に追加されます。

### Step 4

通信コンテキストの名前を、Mediaなどに変更します。

### Step 5

通信コンテキストを独自スレッドで実行するには、独自スレッドを使用をクリックします。

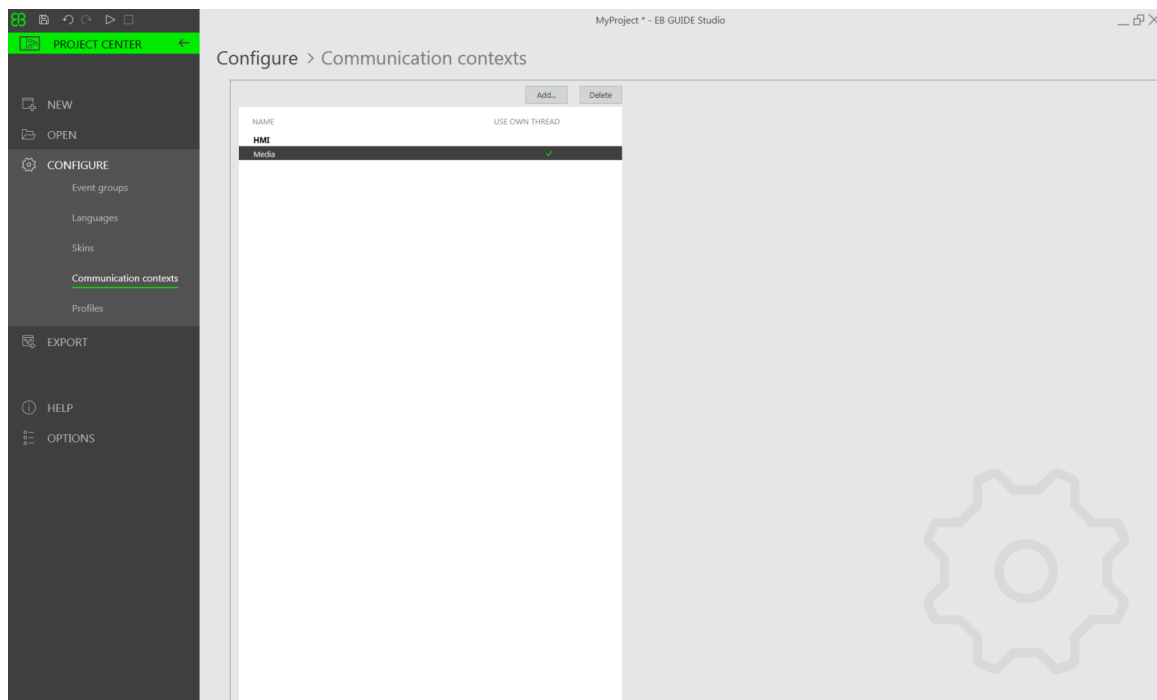


Figure 9.1. 通信コンテキストMedia。

## 9.10. データプールアイテム間のリンク設定



### データプールアイテム間のリンク設定

#### Prerequisite:

- 1つのデータプールアイテムが追加されています。
- データプールアイテムは言語依存ではありません。
- データプールアイテムはスキン依存ではありません。
- データプールアイテムはスクリプト値ではありません。

#### Step 1

データプールコンポーネントでデータプールアイテムをクリックします。

#### Step 2

■ ボタンをクリックします。

メニューが展開されます。

#### Step 3

メニューのデータプールアイテムへのリンクを追加をクリックします。



ダイアログが開きます。

#### Step 4

新しいデータプールアイテムを追加するには、テキストボックスに名前を入力します。

#### Step 5

データプールアイテムを追加をクリックします。

#### Step 6

承認をクリックします。

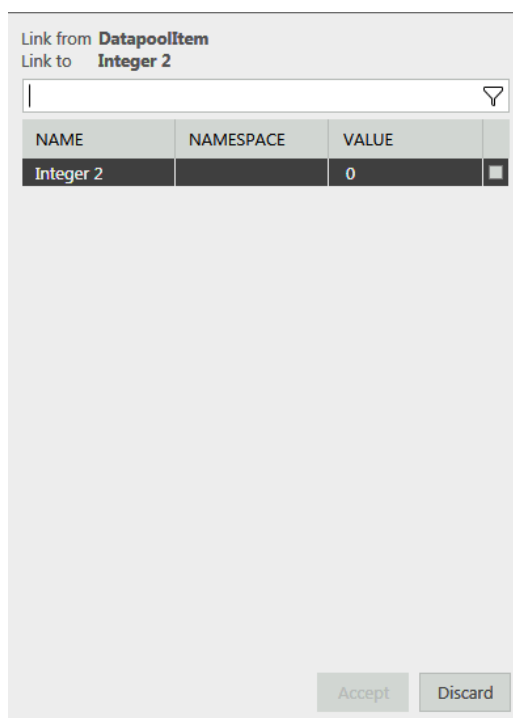



Figure 9.2. データプールアイテム間のリンク設定

ダイアログが閉じられます。Valueプロパティの横に、 ボタンが表示されます。これは、Valueプロパティがデータプールアイテムにリンクされたことを示します。一方のデータプールアイテムの値を変更するたびに、他方のデータプールアイテムの値も変更されます。

## 9.11. データプールアイテムの削除



### データプールアイテムの削除

Prerequisite:

- 1つのデータプールアイテムが追加されています。

#### Step 1

データプールコンポーネントで、対応する名前空間を選択します。

#### Step 2

データプールアイテムを右クリックし、コンテキストメニューの削除を選択します。

データプールアイテムが削除されます。

## 9.12. 名前空間の操作

名前空間の詳細については、[6.5「グラフィカルユーザーインターフェイスのコンポーネント」](#)および[6.13「名前空間」](#)をご覧ください。

### 9.12.1. 名前空間の追加



#### 名前空間の追加

新しい名前空間を子として、ルート名前空間または任意の既存名前空間に追加できます。

#### Step 1

名前空間コンポーネントは、EB GUIDEのデフォルトのウィンドウレイアウトでは使用できないので、次の操作を行います。

コマンドエリアでレイアウトをクリックし、名前空間を選択します。

名前空間コンポーネントが表示されます。

#### Step 2

名前空間コンポーネントで+をクリックします。

名前空間がツリーに追加されます。

#### Step 3

名前空間の名前を変更します

**TIP**



**名前空間の移動**

名前空間を移動するには、ルート名前空間または別の名前空間にドラッグします。名前空間のネーミングルールに従うことで名前の衝突を回避します。

## 9.12.2. 名前空間へのモデル要素の追加

名前空間へのイベントの追加方法については、[9.1「イベントの追加」](#)をご覧ください。

名前空間へのデータプールアイテムの追加方法については、[9.6「データプールアイテムの追加」](#)をご覧ください。

## 9.12.3. 名前空間どうしでのモデル要素の移動



### 名前空間どうしでのモデル要素の移動

Prerequisite:

- モデル要素、イベント、またはデータプールアイテムが名前空間に追加されていること。
- 少なくとも2つの名前空間が存在すること。

#### Step 1

イベントを移動するには、イベントコンポーネントに移動します。

データプールアイテムを移動するには、データプールコンポーネントに移動します。

#### Step 2

対応する名前空間を選択し、モデル要素を右クリックします。

#### Step 3

コンテキストメニューで、名前空間への移動...を選択します。

ダイアログが開きます。

#### Step 4

移動先の名前空間を選択し、承認をクリックします。

モデル要素が移動先の名前空間に移動します。

**NOTE**



モデル要素の移動  
また、モデル要素を別の名前空間にドラッグすることもできます。

## 9.12.4. 名前空間の削除

**WARNING**



名前空間の削除  
名前空間を削除すると、この名前空間に含まれているすべてのモデル要素も削除されます。



### 名前空間の削除

**NOTE**



ルート名前空間  
ルート名前空間は削除できないことに注意してください。

Prerequisite:

- 名前空間がEB GUIDEモデルに追加されていること。

Step 1

ナビゲーションコンポーネントで、名前空間を右クリックします。

Step 2

コンテキストメニューの削除をクリックします。

名前空間が削除されます。

## 10. プロジェクトの処理

### 注記



#### デフォルトのウィンドウレイアウト


すべての手順およびスクリーンショットでは、デフォルトのウィンドウレイアウトが使用されています。手順に従う場合は、EB GUIDE Studioウィンドウを [レイアウト] > [Reset to default layout] を選択することによって、デフォルトレイアウトにリセットすることをお勧めします。

### 10.1. プロジェクトの作成



#### プロジェクトの作成

##### ステップ 1

 をクリックします。

プロジェクトセンターが表示されます。

##### ステップ 2

ナビゲーションエリアで、[新規] をクリックします。

##### ステップ 3

プロジェクト名を入力し、場所を選択します。

##### ステップ 4

[作成] をクリックします。

プロジェクトが作成されます。プロジェクトエディターが開かれ、新しいプロジェクトが表示されます。

## 10.2. プロジェクトを開く

### 注記



#### 無効なプロジェクト

EB GUIDEプロジェクトが有効でない場合、EB GUIDE Studioではそのプロジェクトを開くことができません。エラーメッセージが表示され、エラーを説明するログファイルが\$GUIDE\_PROJECT\_PATH/<project name>/<project name>\_LoadingErrorLog.txtに作成されます。

詳しくは、[6.7.2「EB GUIDEプロジェクトの検証基準」](#)をご覧ください。

### 10.2.1. ファイルエクスプローラからプロジェクトを開く



#### ファイルエクスプローラからプロジェクトを開く

#### 前提条件:

- EB GUIDEプロジェクトが作成されます。

#### ステップ 1

ファイルエクスプローラを開き、開きたいEB GUIDEプロジェクトファイルを選択します。EB GUIDEプロジェクトファイルのファイル拡張子は .ebguide です。

#### ステップ 2

EB GUIDEプロジェクトファイルをダブルクリックします。

プロジェクトがEB GUIDE Studioに開かれます。

### 10.2.2. プロジェクトをEB GUIDE Studioに開く




#### プロジェクトをEB GUIDE Studioに開く

#### 前提条件:

- EB GUIDEプロジェクトが作成されます。

#### ステップ 1

 をクリックします。

プロジェクトセンターが表示されます。

### ステップ 2

ナビゲーションエリアで、[開く]タブをクリックします。

### ステップ 3

[最近使ったプロジェクト]の一覧にあるプロジェクトを選択するか、[参照]をクリックして、開きたいEB GUIDEプロジェクトファイルを選択します。EB GUIDEプロジェクトファイルのファイル拡張子は .ebguide です。

プロジェクトがEB GUIDE Studioに開かれます。

## 10.3. モデル要素の名前を変更する



### モデル要素の名前を変更する

次の手順は、モデル要素(ステート、ステートマシン、ウィジェット、遷移、データプールアイテム、イベントなど)の名前を変更する手順を説明しています。

前提条件:

- モデル要素がEB GUIDEモデルに追加されていること。

### ステップ 1

モデル要素の名前を変更するには、以下の手順を実行します。

- ▶ ウィジェット、ステート、ステートマシンなどのモデル要素の名前を変更するには、[ナビゲーション]コンポーネントで、そのモデル要素を右クリックします。
- ▶ データプールアイテムの名前を変更するには、[データプール]コンポーネントで、データプールアイテムを右クリックします。
- ▶ イベントの名前を変更するには、[イベント]コンポーネントで、データプールアイテムを右クリックします。

コンテキストメニューが開きます。

### ステップ 2

コンテキストメニューで、次のいずれかをクリックします。

- ▶ 選択されているモデル要素のみの名前を変更するには、[名前の変更]をクリックします。
- ▶ 選択されているモデル要素だけでなく、EB GUIDEモデルでのその出現箇所(例えば、EB GUIDEスクリプト内)についても名前を変更するには、[グローバルな名前変更]をクリックします。

## 10.4. EB GUIDEモデルの検証およびモデル実行

EB GUIDEモデルを対象デバイスにエクスポートする前に、PC上でエラーを解決し、モデルをシミュレートします。



## 10.4.1. EB GUIDEモデルの検証

### 10.4.1.1. EB GUIDE Studioを使用したEB GUIDEモデルの検証



#### EB GUIDE Studioを使用したEB GUIDEモデルの検証

EB GUIDEは、[問題検出]コンポーネントに以下の項目を表示します。


- ▶  エラー
- ▶  警告

詳しくは、[6.7.2「EB GUIDEプロジェクトの検証基準」](#)をご覧ください。

#### ステップ 1

[問題検出]コンポーネントを展開するには、[問題]をクリックします。

#### ステップ 2

[問題検出]コンポーネントでをクリックします。

エラーと警告の一覧が表示されます。



図10.1 問題検出コンポーネント

#### ステップ 3


問題の箇所に移動するために、該当行をダブルクリックします。

問題の原因となっている要素が、強調表示されます。

#### ステップ 4

問題を解決します。

#### ステップ 5

をクリックします。



解決した問題は、[問題検出]コンポーネントの一覧から削除されます。

#### ステップ 6

[問題検出]コンポーネントを折りたたむには、[問題]を再度クリックします。

エラーがない場合、EB GUIDEモデルは有効です。警告がいくつかあった場合でも、EB GUIDEモデルは有効です。

### 10.4.1.2. コマンドラインを使用したEB GUIDEモデルの検証



#### コマンドラインを使用したEB GUIDEモデルの検証

##### ステップ 1

コマンドラインで、\$GUIDE\_INSTALL\_PATH/Studioに移動します。

##### ステップ 2

Studio.Console.exe -c "<logfile dir>/log.txt" -o "\$GUIDE\_PROJECT\_PATH/project\_name.ebguide"と入力します。

EB GUIDEモデルが検証され、結果が指定した場所(<logfile dir>)のログファイルに保存されます。

### 10.4.2. シミュレーションの開始と停止



#### シミュレーションの開始と停止

##### ステップ 1

シミュレーションを開始するには、コマンドエリアで▶をクリックします。

シミュレーションとEB GUIDE Monitorが開始されます。シミュレーションは、そのシミュレーション自体の設定で開始されます。

設定を変更するには、プロジェクトセンターに移動し、[設定] > [プロファイル]の順にクリックします。

##### ステップ 2

シミュレーションを停止するには、コマンドエリアで□をクリックします。

シミュレーションとEB GUIDE Monitorが停止します。

## 10.5. EB GUIDEモデルのエクスポート

## 10.5.1. EB GUIDE Studioを使用したEB GUIDEモデルのエクスポート




### EB GUIDE Studioを使用したEB GUIDEモデルのエクスポート

EB GUIDEモデルを対象デバイスにコピーするには、EB GUIDE Studioを使用してそのモデルをエクスポートする必要があります。

EB GUIDEモデルをエクスポートするたびに、必ずプロファイルを選択します。

#### ステップ 1

をクリックします。

プロジェクトセンターが表示されます。

#### ステップ 2

ナビゲーションエリアで、[エクスポート]タブをクリックします。

#### ステップ 3

Profileドロップダウンリストボックスでプロファイルを選択します。

#### ステップ 4

[参照]をクリックし、バイナリファイルのエクスポート先となる場所を選択します。

#### ステップ 5

[フォルダーの選択]をクリックします。

#### ステップ 6

[エクスポート]をクリックします。

選択した場所にバイナリファイルがエクスポートされます。

## 10.5.2. コマンドラインを使用したEB GUIDEモデルのエクスポート



### コマンドラインを使用したEB GUIDEモデルのエクスポート

前提条件:

- EB GUIDEモデルにエラーおよび警告がないこと。

#### ステップ 1

コマンドラインで、`$GUIDE_INSTALL_PATH/Studio`に移動します。

#### ステップ 2

`Studio.Console.exe -e <destination dir> -p <profile> -o "$GUIDE_PROJECT_PATH/project_name.ebguide"`と入力します。


EB GUIDEモデルが選択した場所<destination dir>に指定したプロファイル<profile>でエクスポートされます。

## 10.6. EB GUIDE Studioの表示言語の変更



### EB GUIDE Studioの表示言語の変更

#### ステップ 1

をクリックします。

プロジェクトセンターが表示されます。

#### ステップ 2

ナビゲーションエリアで[オプション]タブをクリックします。

#### ステップ 3

[表示言語]ドロップダウンリストボックスから言語を選択します。

#### ステップ 4

EB GUIDE Studioを再起動します。

再起動後、グラフィカルユーザーインターフェースは選択した言語で表示されます。

## 10.7. プロファイルの設定

EB GUIDE Studio では、EB GUIDEモデルのさまざまなプロファイルを作成することができます。

プロファイルを使用して、以下の操作を行えます。

- ▶ メッセージの送信
- ▶ 読み込む内部ライブラリおよびユーザー定義ライブラリの設定
- ▶ シーンの設定
- ▶ レンダラーの設定

デフォルトのプロファイルは2つあります。[編集]と[シミュレーション]です。

### 10.7.1. プロファイルの追加



### プロファイルの追加

EB GUIDE Studioでプロファイルを追加するには、既存のプロファイルを複製します。

前提条件:

- EB GUIDEプロジェクトが開いていること。
- プロジェクトセンターが表示されます。

#### ステップ 1

ナビゲーションエリアで、[設定] > [プロファイル]の順にクリックします。

#### ステップ 2

コンテンツエリアで[シミュレーション]プロファイルを選択します。

#### ステップ 3

[複製]をクリックします。

プロファイルが表に追加されます。このプロファイルはデフォルトプロファイルである[シミュレーション]の複製です。

#### ステップ 4

表でダブルクリックし、プロファイルの名前をMySimulationに変更します。

#### ステップ 5

[シミュレーションに使用]をクリックします。

MySimulationプロファイルは、PCでのシミュレーションに使用されます。

## 10.7.2. ライブラリの追加

EB GUIDE TFのデフォルトデリバリは、共有ライブラリをサポートするWindows 10、Linux、QNXなどのオペレーティングシステムで動作します。EB GUIDE TFは実行可能ファイルとライブラリに分かれており、ほとんどの顧客プロジェクトにそのまま適合します。

次のタスクでは、EB GUIDEモデルを操作するユーザー定義ライブラリを追加し、追加機能を提供する方法を示します。



### ライブラリの追加: プラットフォーム

このタスクは、現在のプラットフォームのすべてのEB GUIDEモデルに使用できるライブラリを追加する方法を示しています。

前提条件:

- EB GUIDEプロジェクトが開いていること。

- プロジェクトセンターが表示されます。
- ナビゲーションエリアで、[設定] > [プロファイル]タブが選択されていること。
- MySimulationプロファイルが追加されていること。
- MyLibraryAおよびMyLibraryBライブラリが\$GTF\_INSTALL\_PATH/platform/<platform name>/binで使用可能であること。

#### ステップ 1

コンテンツエリアで、MySimulationプロファイルを選択します。

#### ステップ 2

[プラットフォーム]タブをクリックします。

#### ステップ 3

次のコードを入力します。

```
{
  "gtf":
  {
    "core":
    {
      "pluginstoload": ["MyLibraryA", "MyLibraryB"]
    }
  }
}
```

MyLibraryAおよびMyLibraryBライブラリが起動コードに追加されました。

#### 注記



#### JSONオブジェクト表記

EB GUIDE Studio内でplatform.jsonを設定する場合は、JSONオブジェクト表記を使用します。

例えば、[13.8.1「EB GUIDE Studioでのサンプルplatform.json」](#)をご覧ください。

JSON形式の詳細については、<http://www.json.org>をご覧ください。



#### ライブラリの追加: モデル

このタスクは、現在のEB GUIDEモデルのみが使用できるライブラリを追加する方法を示しています。

前提条件:

- EB GUIDEプロジェクトが開いていること。
- プロジェクトセンターが表示されます。
- ナビゲーションエリアで、[設定] > [プロファイル]タブが選択されていること。

- MySimulationプロファイルが追加されていること。
- MyLibraryAおよびMyLibraryBライブラリが\$GUIDE\_PROJECT\_PATH/<project name>/resourcesで使用可能であること。

#### ステップ 1

コンテンツエリアで、MySimulationプロファイルを選択します。

#### ステップ 2

[モデル]タブをクリックします。

#### ステップ 3

次のコードを入力します。

```
{
  "gtf":
  {
    "model":
    {
      "pluginstoload": ["resources/MyLibraryA", "resources/MyLibraryB"]
    }
  }
}
```

MyLibraryAおよびMyLibraryBライブラリが起動コードに追加されました。

#### 注記



#### JSONオブジェクト表記

EB GUIDE Studioでmodel.jsonを設定する場合は、JSONオブジェクト表記を使用します。

例えば、[13.6.1「EB GUIDE Studioでのサンプルmodel.json」](#)をご覧ください。

JSON形式の詳細については、<http://www.json.org>をご覧ください。

## 10.7.3. シーンの設定

EB GUIDE Studioでは、すべてのステートマシンにシーンを設定することができます。

プロジェクトには、次の理由で複数のステートマシンを含めることができます。

- ▶ モデルのロジックを異なるステートマシンに分けるため
- ▶ 複数のディスプレイまたはレイヤーを使用するため



## シーンの設定

前提条件:

- EB GUIDEプロジェクトが開いていること。
- プロジェクトセンターが表示されます。
- ナビゲーションエリアで、[設定] > [プロファイル]タブが選択されていること。

### ステップ 1

コンテンツエリアで、[シーン]タブをクリックします。

### ステップ 2

[ステートマシン]ドロップダウンリストボックスで、メインディスプレイのステートマシン(例えば、[メイン])を選択します。

### ステップ 3

PCデスクトップ上のウィンドウの最初の位置を設定するには、\*およびyに値を入力します。

### ステップ 4

[レンダラー]ドロップダウンリストボックスからレンダラーを選択します。

### ステップ 5

その他のプロパティを調整します。各プロパティの詳細については、[13.9「シーン」](#)をご覧ください。

## 10.8. 言語依存テキストのエクスポートとインポート

EB GUIDEでは、ユーザーの選択した言語でテキストを表示できます。そのためには、EB GUIDEモデルに言語サポートを追加します。次に、言語依存テキストを.xliffファイルにエクスポートし、それらのテキストを翻訳したうえで再びモデルにインポートします。

注記



プロジェクトおよび言語固有のID

プロジェクトと言語のすべてペアについて、英数字からなる一意のsourcelanguageidおよびtargetlanguageidが作成されます。これらのIDは、別のプロジェクトまたはターゲット言語からの.xliffファイルの意図しないインポートを防止します。また、それぞれのデータプールアイテムは各言語の一意の英数字IDを受け取ります。

テキストが翻訳のためにエクスポートされている間にEB GUIDE Studioモデルで言語またはデータプールアイテムが変更された場合でも、翻訳されたテキストは、固有のIDによって適切なデータプールアイテムおよび言語に割り当てることができます。すべてのデータプールアイテムが正しく割り当てられているかどうかを確認するには、インポートのログファイルをご覧ください。

## 10.8.1. 言語依存テキストのエクスポート

ティップ



EB GUIDEモデルの検証

テキストのエクスポートとインポート時のエラーを回避するには、開始前にEB GUIDEモデルを検証します。

詳しくは、[10.4.1.1「EB GUIDE Studioを使用したEB GUIDEモデルの検証」](#)をご覧ください。



### 言語依存テキストのエクスポート

ユーザーの優先する言語でテキストを指定するには、データプールアイテムの言語依存テキストをすべてエクスポートし、そのテキストをローカライズ担当のサービスプロバイダーに渡します。

前提条件:

- 翻訳対象言語がEB GUIDEモデルに追加されていること。詳しくは、[8.4.1「EB GUIDEモデルへの言語の追加」](#)をご覧ください。
- StringタイプまたはString listタイプのデータプールアイテムが追加されます。
- データプールアイテムに言語サポートがあること。詳しくは、[8.4.2「データプールアイテムに言語サポートを追加する」](#)をご覧ください。
- EB GUIDEモデルにエラーおよび警告がないこと。

ステップ 1

をクリックします。

プロジェクトセンターが表示されます。

ステップ 2

ナビゲーションエリアで、[設定] > [言語]の順にクリックします。



### ステップ 3

コンテンツエリアで、翻訳対象となるターゲット言語を選択します。

複数選択が可能です。

### ステップ 4

[エクスポート]をクリックします。

ダイアログが開きます。

### ステップ 5

ファイルのエクスポート先となるディレクトリを選択します。

### ステップ 6

[フォルダーの選択]をクリックします。

結果: エクスポートが開始します。選択したディレクトリにファイルが保存されます。ファイルには言語依存頭字語と `.xliff`形式が含まれます。ソース言語の値とターゲット言語の値がファイルに格納されます。

#### 注記

#### エクスポートされたファイルの構造と内容



- ▶ プロジェクトセンターで選択する言語ごとに異なる `.xliff`ファイルがエクスポートされます。
- ▶ ソース言語はデフォルトの言語です。そのため、テキストがまだ翻訳されていない場合、`target-language`要素には常にソーステキストが含まれます。

EB GUIDEモデルおよび `.xliff`ファイルにおける言語依存テキストの詳細については、[6.12「言語」](#)をご覧ください。

## 10.8.2. 言語依存テキストのインポート

### 10.8.2.1. EB GUIDE Studioを使用した言語依存テキストのインポート




#### EB GUIDE Studioを使用した言語依存テキストのインポート

#### 前提条件:

- 選択されているEB GUIDEモデルの翻訳済み `.xliff`ファイルが少なくとも1つは使用できること。
- 翻訳対象のデータプールアイテムとターゲット言語が依然として存在していること。
- EB GUIDEモデルにエラーおよび警告がないこと。

#### ステップ 1

をクリックします。

プロジェクトセンターが表示されます。

#### ステップ 2

ナビゲーションエリアで、[設定] > [言語]の順にクリックします。

#### ステップ 3

[インポート]をクリックします。

ダイアログが開きます。

#### ステップ 4

翻訳済み.xliffファイルの格納先となるディレクトリを選択します。

#### ステップ 5

翻訳済み.xliffファイルを選択します。

複数選択が可能です。

#### ステップ 6

[開く]をクリックします。

インポートが開始します。ダイアログが開きます。

#### ステップ 7

[閉じる]をクリックします。

これで、言語サポートがあるすべてのデータプールアイテムに、対応する言語依存テキストが表示されます。インポートの詳細については、ログファイルをご覧ください。

### 10.8.2.2. コマンドラインを使用した言語依存テキストのインポート



#### コマンドラインを使用した言語依存テキストのインポート

前提条件:

- 選択されているEB GUIDEモデルの翻訳済み.xliffファイルが少なくとも1つは使用できること。
- 翻訳用に送信されたデータプールアイテムが依然として存在していること。
- EB GUIDEモデルにエラーおよび警告がないこと。

#### ステップ 1

コマンドラインで、\$GUIDE\_INSTALL\_PATH/Studioに移動します。

#### ステップ 2

Studio.Console.exe -l <language file> -o "\$GUIDE\_PROJECT\_PATH/project\_name.-ebguide"と入力します。

インポートが成功した場合、EB GUIDEモデルは変更されます。インポートが失敗した場合、EB GUIDEモデルは変更されません。どちらの場合もログファイルが生成されます。ログファイルの名前には日付とタイムスタンプが追加されます。

## 10.9. EB GUIDE Monitorを操作する

EB GUIDE Monitorの詳細については、[6.5.4「EB GUIDE Monitor」](#)をご覧ください。

### 注記



#### デフォルトのウィンドウレイアウト

すべての手順およびスクリーンショットでは、デフォルトのウィンドウレイアウトが使用されています。手順に従う場合は、EB GUIDE Monitorウィンドウを [レイアウト] > [Reset to default layout]を選択することによって、デフォルトレイアウトにリセットすることをお勧めします。

### 10.9.1. EB GUIDE Monitorでのイベント発行




#### EB GUIDE Monitorでのイベント発行

##### 前提条件:

- EB GUIDEモデルにイベントが含まれていること。
- EB GUIDEモデルのシミュレーションが開始されます。
- EB GUIDE Monitorが開始されます。

##### ステップ 1

EB GUIDE Monitorの[イベント]コンポーネントで、 をクリックします。


ダイアログが開きます。

##### ステップ 2

発行するイベントを選択し、[承認]をクリックします。

イベントがリストに追加されます。

##### ステップ 3

イベントを発行するには、イベントの横にある[イベント]コンポーネントで  をクリックします。

イベントが発行されます。[ロガー]コンポーネントにログメッセージが表示されます。

##### ステップ 4

イベントにパラメータがある場合は、次の操作を行います。


#### ステップ 4.1

■ をクリックして、パラメータを展開します。

#### ステップ 4.2

[値]列のパラメータを変更します。

#### ステップ 4.3

イベントを発行するには、イベントの横にある  をクリックします。

変更後のパラメータでイベントが発行されます。[ロガー]コンポーネントにログメッセージが表示されます。

## 10.9.2. EB GUIDE Monitorでのデータプールアイテムの値の変更



### EB GUIDE Monitorでのデータプールアイテムの値の変更

前提条件:

- EB GUIDEモデルにデータプールアイテムが含まれていること。
- EB GUIDEモデルのシミュレーションが開始されます。
- EB GUIDE Monitorが開始されます。

#### ステップ 1

EB GUIDE Monitorの[データプール]コンポーネントで、+ をクリックします。

ダイアログが開きます。

#### ステップ 2

データプールアイテムを選択し、[承認]をクリックします。

データプールアイテムがリストに追加されます。

#### ステップ 3

[値]列でデータプールアイテムの値を変更します。

注記



サポートされるタイプ

以下のデータタイプのデータプールアイテムを変更することができます。f

- ▶ ブール値
- ▶ 色
- ▶ 整数
- ▶ 浮動小数点数
- ▶ 文字列

データプールアイテムの値が変更されます。[ロガー]コンポーネントにログメッセージが表示されます。

### 10.9.3. EB GUIDE Monitorでのスクリプトの開始



#### EB GUIDE Monitorでのスクリプトの開始

前提条件:

- EB GUIDEモデルのシミュレーションが開始されます。
- EB GUIDE Monitorが開始されます。
- スクリプトが含まれている .csファイルまたは .dllファイルをコンピューターで使用できること。スクリプトのサンプルについては、[10.9.3.1「EB GUIDE Monitorでのスクリプトファイルの記述」](#)をご覧ください。

#### ステップ 1

[スクリプト]コンポーネントを開くには、[レイアウト] > [スクリプト]を選択します。

[スクリプト]コンポーネントがドッキングされたコンポーネントとして開きます。

#### ステップ 2

[スクリプト]コンポーネントで、[Open]ボタンをクリックします。

ファイルエクスプローラが開きます。

#### ステップ 3

.csファイルまたは .dllファイルを選択して、[Open]をクリックします。

ファイルに含まれている適用可能なすべてのメソッドおよび対応するクラスが、[スクリプト]表に一覧で示されます。

#### ステップ 4

メソッドを選択し、開始ボタンをクリックします。

スクリプトが開始されます。[スクリプト出力]エリアにログメッセージが表示されます。

#### 10.9.3.1. EB GUIDE Monitorでのスクリプトファイルの記述

スクリプトのメソッドの詳細については、`$GUIDE_INSTALL_PATH/doc/monitor/monitor_api.chm`のEB GUIDE Monitor APIをご覧ください。

以下に、基本的なEB GUIDE Monitorスクリプト機能のサンプルを示します。

注記



ステートおよびステートマシンでのメソッドの使用

EB GUIDEモデルに同じ名前のステートまたはステートマシンが複数ある場合は、uint ID を使用します。`$EXPORT_PATH/monitor.cfg`で、プロジェクトに関係があるuint IDを検索します。



例10.1

### EB GUIDE Monitorのサンプルスクリプトファイル

サンプルスクリプトMonitorScriptSample.csを以下に示します。

```
namespace MyProject
{
    using System.Threading.Tasks;

    using System.Windows.Media; // necessary for Color type!

    using Elektrobit.Guide.Monitor.Scripting.MonitorContext;

    public class Basic
    {
        public async Task PrintMessage(IMonitorContext monitor) //❶
        {
            await monitor.Write("Hello World");
        }

        public async Task FireEvent(IMonitorContext monitor) //❷
        {
            await monitor.FireEvent("nextView");
        }
    }

    public class Events
    {
        public async Task FireEventWithParameter(IMonitorContext monitor)
        {
            await monitor.FireEvent("setBool", true);
        }

        public async Task WaitForEvent(IMonitorContext monitor) //❸
        {
            var ev = await monitor.WaitForEvent("nextView");
            await monitor.Write("Even occurred: " + ev.EventModel.Name);
        }

        public async Task WaitForEventWithParameters(IMonitorContext monitor)
        {
            var ev = await monitor.WaitForEvent("setBool");

            bool mv1 = ev["value"]; // read parameter via name
            bool mv2 = ev[0]; // read the parameter via index

            await monitor.Write("Parameter 'value' is: " + mv1);
            await monitor.Write("Parameter [0] is: " + mv2);
        }
    }
}
```

```
    }  
}  
  
public class Datapool  
{  
    public async Task WriteDpValue(IMonitorContext monitor) //④  
    {  
        await monitor.WriteDatapool("Boolean 1", true);  
    }  
  
    public async Task ReadDatapoolValue(IMonitorContext monitor) //⑤  
    {  
        bool boolValue = await monitor.ReadDatapool("Boolean 1");  
        string stringValue = await monitor.ReadDatapool("String 1");  
        int integerValue = await monitor.ReadDatapool("Integer 1");  
        float floatValue = await monitor.ReadDatapool("Float 1");  
  
        await monitor.Write("Boolean: " + boolValue);  
        await monitor.Write("String: " + stringValue);  
        await monitor.Write("Integer: " + integerValue);  
        await monitor.Write("Float: " + floatValue);  
    }  
  
    public async Task ReadColor(IMonitorContext monitor)  
    {  
        Color colorValue = await monitor.ReadDatapool("Color 1");  
        await monitor.Write("Boolean: " + colorValue);  
    }  
}  
  
public class StateMachines  
{  
    public async Task WaitForStateChanges(IMonitorContext monitor)  
    {  
        var leftState = await monitor.WaitForStateExit  
            ("Main", "State 1"); //⑥  
        await monitor.Write(string.Format("State {0} left",  
            leftState.Name));  
  
        var enteredState = await monitor.WaitForStateEnter  
            ("Main", "State 2"); //⑦  
        await monitor.Write(string.Format("State {0} entered",  
            enteredState.Name));  
    }  
  
    public async Task WaitForStateMachineChanges(IMonitorContext monitor)
```

```
{
    var startedStateMachine = await monitor.WaitForStateMachineStart
        ("Dynamic state machine 1"); //❸
    await monitor.Write(string.Format("State Machine {0} started",
        startedStateMachine.Name));

    var stoppedStateMachine = await monitor.WaitForStateMachineStop
        ("Dynamic state machine 1"); //❹
    await monitor.Write(string.Format("State Machine {0} stopped",
        stoppedStateMachine.Name));
}
}

public class Advanced
{
    public async Task CaptureScreenshot(IMonitorContext monitor) //❺
    {
        // make sure remote framebuffer is enabled in profile
        uint sceneId = 0;
        await monitor.CaptureScreenshot(sceneId, @"d:/image.png");
    }

    public async Task CountTo10(IMonitorContext monitor)
    {
        for (var i = 0; i < 10; i++)
        {
            await monitor.Write("Hello World: " + i);
            await Task.Delay(1000, monitor.CancellationToken);

            monitor.CancellationToken.ThrowIfCancellationRequested();
        }
    }

    public async Task WaitForEventWithTimeout(IMonitorContext monitor) //❻
    {
        // Disclaimer:
        // this is just one of many opportunities provided by
        // the .NET's "Task Parallel Library"

        var eventWaitTask = monitor.WaitForEvent("nextView");

        await Task.WhenAny(eventWaitTask, Task.Delay(5000));

        if (!eventWaitTask.IsCompleted || eventWaitTask.IsFaulted)
        {
            return;
        }
    }
}
```



```
        await monitor.Write("event occurred");
    }
}

namespace MonitorScripting.EventScripts //❶
{
    using Elektrobit.Guide.Monitor.Scripting.MonitorContext;
    using System.Threading;
    using System.Threading.Tasks;
    using Elektrobit.Guide.Monitor.Model.Event;
    using Elektrobit.Guide.Monitor.Model.Value;

    public class MonitorScripts
    {
        public async Task FireEventInNamespace(IMonitorContext monitor)
        {
            string[] namespacePath = { "Foo_namespace", "go_to_view2" };
            var identifier = new QualifiedIdentifier(namespacePath);

            await monitor.FireEvent(identifier);
        }

        public async Task FireEventInNestedNamespace(IMonitorContext monitor)
        {
            string[] namespacePath =
                { "Foo_namespace", "sub_namespace_under_foo", "go_to_view4" };
            var identifier = new QualifiedIdentifier(namespacePath);

            await monitor.FireEvent(identifier);
        }

        public async Task FireEventInRootNamespace(IMonitorContext monitor)
        {
            await monitor.FireEvent("go_to_view3");
        }
    }
}
}
```

- ❶ メッセージを出力するメソッド
- ❷ イベントを発行するメソッド
- ❸ イベントが発行されるまで待機するメソッド
- ❹ データプール値を書き込むメソッド
- ❺ データプール値を読み取るメソッド
- ❻ ステートにエントリーするまで待機したうえでエントリーを報告するメソッド

- ⑦ ステートが終了するまで待機したうえで終了を報告するメソッド
- ⑧ ステートマシンが開始されるまで待機したうえで開始を報告するメソッド
- ⑨ ステートマシンが停止するまで待機したうえで停止を報告するメソッド
- ⑩ スクリーンショットをキャプチャするメソッド
- ⑪ イベントがタイムアウトするまで待機するメソッド
- ⑫ 名前空間の操作方法の例

## 10.9.4. スタンドアロンアプリケーションとしてEB GUIDE Monitorを起動する

EB GUIDE Monitorは、EB GUIDEモデルのモデル実行中にEB GUIDE Studioで自動的に起動されます。しかし、`$GUIDE_INSTALL_PATH/tools/monitor`のスタンドアロンアプリケーションとして、またはコマンドラインを使って、EB GUIDE Monitorを起動することもできます。



### コマンドラインを使用してEB GUIDE Monitorを起動する

前提条件:

- EB GUIDEのインストールが完了していること。
- EB GUIDEモデルが`$EXPORT_PATH`にエクスポートされていること。

#### ステップ 1

ファイルエクスプローラで、`$GUIDE_INSTALL_PATH/tools/monitor`に移動します。

#### ステップ 2

コマンドラインを開き、次のように入力します。Monitor.exe

EB GUIDE Monitorが起動します。



### EB GUIDE Monitorの表示言語の変更

前提条件:

- EB GUIDE Monitorがスタンドアロンアプリケーションとして起動されていること。

#### ステップ 1

[ファイル] > [表示言語]から言語を選択します。

#### ステップ 2

EB GUIDE Monitorを再起動します。

再起動後、グラフィカルユーザーインターフェースが選択した言語で表示されます。



注記

**EB GUIDE Monitor**による**EB GUIDE Studio**からの言語の継承

EB GUIDE MonitorがEB GUIDE Studioで起動されている場合、グラフィカルユーザーインターフェースの表示言語は変更できません。EB GUIDE Monitorの表示言語はEB GUIDE Studioと同じものになります。

## 10.9.5. EB GUIDE Monitorへの設定の読み込み



### EB GUIDE Monitorへの設定ファイルの読み込み

前提条件:

- EB GUIDE Monitorがスタンドアロンアプリケーションとして起動されていること。
- EB GUIDEモデルが`$EXPORT_PATH`にエクスポートされていること。

#### ステップ 1

[ファイル] > [読み込み設定]を選択します。

ダイアログが開きます。

#### ステップ 2

`$EXPORT_PATH`に移動して、`monitor.cfg`設定ファイルを選択します。

#### ステップ 3

[開く]をクリックします。

プロジェクトの設定がEB GUIDE Monitorに読み込まれます。



### EB GUIDE Monitorへの最近使った設定ファイルの読み込み

前提条件:

- EB GUIDE Monitorがスタンドアロンアプリケーションとして起動されていること。
- 1つ以上の設定が最近使用されていること。

#### ステップ 1

[ファイル] > [最近使った設定ファイル]を選択します。

ダイアログが開きます。

### ステップ 2

それぞれの場所に移動して、設定ファイルを選択します。

設定がEB GUIDE Monitorに読み込まれます。

#### 注記



#### EB GUIDE MonitorのEB GUIDE GTFからの切断

新しい設定を読み込む前に、EB GUIDE Monitorは現在のEB GUIDE GTFから自動的に切断されます。

EB GUIDE Monitorは再接続して、新しい設定を読み込みます。

## 10.9.6. ウォッチリストのエクスポートおよびインポート

プロジェクトに使用するイベントおよびデータプールアイテムは、ウォッチリストに格納されます。他のプロジェクトにアイテムを使用するには、ウォッチリストを.xmlファイルとしてエクスポートし、後でそれらを新しいプロジェクトにインポートします。



### すべてのウォッチリストのエクスポート

前提条件:

- EB GUIDE Monitorが起動されていること。
- EB GUIDEモデルが[データプール]タブまたは[イベント]タブに格納されたアイテムですすでにセットアップされていること。

#### ステップ 1

すべてのウォッチリストをエクスポートするには、[ファイル] > [すべてのウォッチリストをエクスポート]を選択します。

ダイアログが開きます。

#### ステップ 2

出力先ディレクトリを選択して、ファイル名を入力します。

すべてのデータプールアイテムとイベントがエクスポートされます。



### 単一のウォッチリストのエクスポート

前提条件:

- EB GUIDE Monitorが起動されていること。
- EB GUIDEモデルが[データプール]コンポーネントまたは[イベント]コンポーネントに格納されたアイテムですすでにセットアップされていること。

#### ステップ 1

エクスポートするアイテムが含まれている[データプール]コンポーネントまたは[イベント]コンポーネントに移動します。

#### ステップ 2

このタブのアイテムリストを保存するには、[エクスポート]をクリックします。

ダイアログが開きます。

出力先ディレクトリを選択して、ファイル名を入力します。

コンポーネントのデータプールアイテムまたはイベントがエクスポートされます。



### ウォッチリストのインポート

前提条件:

- EB GUIDE Monitorが起動されていること。
- エクスポートされたウォッチリストがすでに使用可能であること。

#### ステップ 1

ウォッチリストをインポートするには、[ファイル] > [ウォッチリストをインポート]を選択します。

ダイアログが開きます。

#### ステップ 2

インポートするウォッチリストファイルを選択します。

データプールアイテムまたはイベントが新しい[データプール]コンポーネントまたは[イベント]コンポーネントで開きます。

#### 注記



レイアウトはインポートされない

インポートされるのはデータプールアイテムとイベントだけです。レイアウトはインポートされません。

デフォルトレイアウトは、新しく開く[データプール]コンポーネントと[イベント]コンポーネントに使用されます。

# 11. ベストプラクティス

この章では、トピックスをアルファベット順で並べています。

## NOTE



デフォルトのウィンドウレイアウト

すべての手順およびスクリーンショットでは、デフォルトのウィンドウレイアウトが使用されています。手順に従う場合は、EB GUIDE Studioウィンドウをレイアウト > **Reset to default layout**を選択することによって、デフォルトレイアウトにリセットすることをお勧めします。

## 11.1. ベストプラクティス: Conditional script

データプールアイテムおよびウィジェットのプロパティがスクリプト値に変換されている場合、そのプロパティの読み取りのたびにEB GUIDEスクリプトが実行されます。一部のユースケースでは、EB GUIDEスクリプトの実行回数を最小限に抑えてパフォーマンスを向上するために、次の操作を行います。

1. スクリプト値がある場合は、そのプロパティタイプをリセットします。スクリプト値に変換されておらず、プレーン値を持つプロパティを使用します。詳細については、[9.8「プロパティのスクリプト値への変換」](#)および[8.2.5「ウィジェットへのユーザー定義プロパティの追加」](#)をご覧ください。
2. 現在値を計算および設定するには、Conditional scriptタイプのユーザー定義プロパティを追加します。このアクションは、例えば、初期化の際や入力プロパティの変更時など、必要な場合にのみ実行する必要があります。

## 12. チュートリアル

### NOTE



デフォルトのウィンドウレイアウト

すべての手順およびスクリーンショットでは、デフォルトのウィンドウレイアウトが使用されています。手順に従う場合は、EB GUIDE Studioウィンドウをレイアウト > **Reset to default layout**を選択することによって、デフォルトレイアウトにリセットすることをお勧めします。

### 12.1. チュートリアル: 動的ステートマシンの追加

### NOTE



デフォルトのウィンドウレイアウト

すべての手順およびスクリーンショットでは、デフォルトのウィンドウレイアウトが使用されています。手順に従う場合は、EB GUIDE Studioウィンドウをレイアウト > **Reset to default layout**を選択することによって、デフォルトレイアウトにリセットすることをお勧めします。

動的ステートマシンを使用すると、ランタイム中にポップアップを表示できます。動的ステートマシンは、例えば通常の画面にオーバーレイするエラーメッセージを表示する場合などに使用します。

ここからは、動的ステートマシンを作成する手順について、動的ステートマシンをモデリングし、音量を調節する操作を例に挙げて説明します。ここで説明する順番どおりに操作してください。

所要時間: 20分



#### イベントとデータプールアイテムを追加する

このセクションでは、イベントとデータプールアイテムを追加する手順について説明します。これらのイベントを使用して、後ほど音量の調節を行います。データプールアイテムは、後のセクションでグラフィック要素の位置を変更するために使用します。

#### Step 1

イベントコンポーネントに移動して、**+**をクリックします。

イベントが表に追加されます。

#### Step 2

イベントの名前をVolume upに変更します。

#### Step 3

イベントを追加し、その名前をVolume downに変更します。

#### Step 4

イベントを追加し、その名前をClose volume controlに変更します。

#### Step 5

データプールコンポーネントに移動して、十をクリックします。

メニューが展開されます。

#### Step 6

メニューで整数をクリックします。

Integerタイプのデータプールアイテムが追加されます。

#### Step 7

データプールアイテムの名前をVolume indicatorに変更します。

これで、3つのイベントとデータプールアイテムが追加されました。



### 動的ステートマシンを追加して動作をモデリングする

ここからは、動的ステートマシンを追加する手順について説明します。ハプティック動的ステートマシンのモデリングを行い、音量の調節に使用します。

Prerequisite:

- 前のセクションの手順を完了していること。

#### Step 1

ナビゲーションコンポーネントで動的ステートマシンに移動して、十をクリックします。

メニューが展開されます。

#### Step 2

メニューでハプティック動的ステートマシンをクリックします。

ハプティック動的ステートマシンがコンテンツエリアに追加されて表示されます。

#### Step 3

動的ステートマシンの名前をVolume controlに変更します。

#### Step 4

ツールボックスから初期ステートをドラッグし、動的ステートマシンにドロップします。

#### Step 5

ツールボックスからビューステートをドラッグし、動的ステートマシンにドロップします。

ビューステートと共に、ビューがEB GUIDEモデルに追加されます。

#### Step 6

ナビゲーションコンポーネントでビューステートをクリックします。

#### Step 7

F2キーを押し、ビューステートの名前をVolumeに変更します。

#### Step 8

初期ステートからVolumeビューステートへの遷移を追加します。





## スライダーをモデリングする

このセクションでは、水平型のスライダーインジケータをモデリングする手順を説明します。スライダーインジケータはランタイム中に音量を表示します。

スライダーインジケータは2つの四角形で構成されます。一方はスライダーの背景、もう一方は音量です。

Prerequisite:

- 前のセクションの手順を完了していること。

### Step 1

ナビゲーションコンポーネントでVolumeビューステートを展開します。ビューをダブルクリックします。

コンテンツエリアにビューが表示されます。

### Step 2

ツールボックスから四角形をドラッグし、ビューにドロップします。

### Step 3

ナビゲーションコンポーネントで四角形をクリックし、F2キーを押します。

### Step 4

四角形の名前をSlider backgroundに変更します。

### Step 5

Slider backgroundの外観を変更するには、四角形をクリックし、プロパティコンポーネントに移動します。

#### Step 5.1

widthテキストボックスに500と入力します。

#### Step 5.2

xテキストボックスに125と入力します。

#### Step 5.3

yテキストボックスに300と入力します。

### Step 6

ツールボックスから四角形をドラッグし、ナビゲーションコンポーネントのSlider backgroundにドロップします。

四角形が、Slider backgroundの子ウィジェットとして追加されます。

### Step 7

ナビゲーションコンポーネントで四角形をクリックし、F2キーを押します。

### Step 8

四角形の名前をIndicatorに変更します。

### Step 9

Indicatorの外観を変更するには、四角形をクリックし、プロパティコンポーネントに移動します。

#### Step 9.1

widthテキストボックスに40と入力します。

Step 9.2

heightテキストボックスに80と入力します。

Step 9.3

xプロパティの横にある■ボタンをクリックします。

メニューが展開されます。

Step 9.4

メニューのデータプールアイテムへのリンクを追加をクリックします。

ダイアログが開きます。

Step 9.5

リストから、Volume indicatorデータプールアイテムを選択します。

Step 9.6

承認をクリックします。

ダイアログが閉じられます。■ボタンがxプロパティの横に表示されます。これで、xの値とVolume indicatorの値がリンクされました。

Step 9.7

yテキストボックスに10と入力します。

Step 9.8

fillColorプロパティでは黒を選択します。

2つの四角形がビューに追加され、四角形の外観が変更されました。

Step 10

データプールコンポーネントで、Volume indicatorデータプールアイテムをクリックします。

Step 11

Valueテキストボックスに、10と入力します。

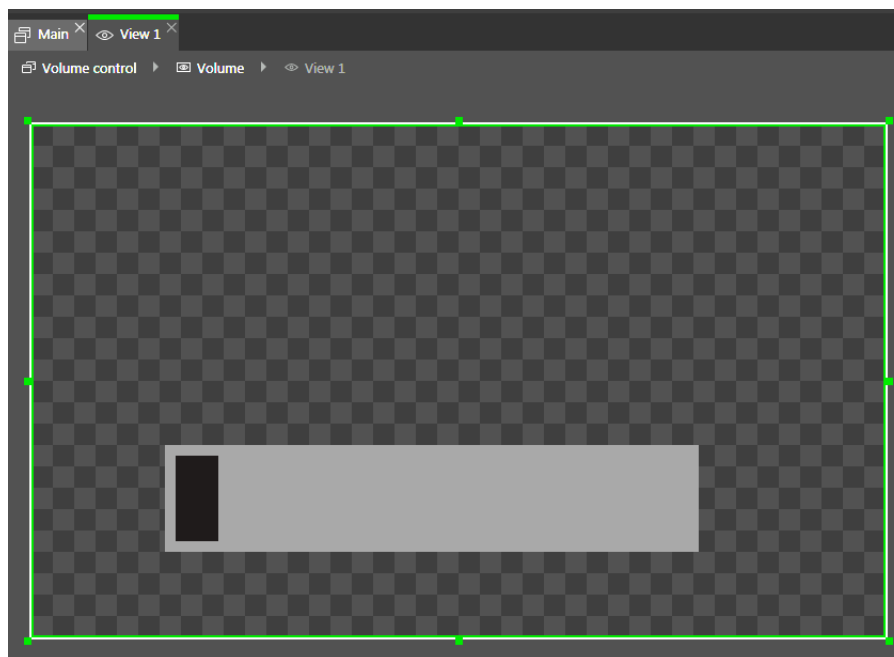


Figure 12.1. 2つの四角形を持つView 1の外観

コンテンツエリアで、四角形Indicatorの位置が変わります。

Volume indicatorデータプールアイテムで、四角形Indicatorの×の位置を調節できます。



#### メインステートマシンにステートを追加する

このセクションでは、メインステートマシンに初期ステートとビューステートを追加します。ビューステートを使用し、動的ステートマシンが他のステートマシンと平行して動作するようにします。

Prerequisite:

- 前のセクションの手順を完了していること。

##### Step 1

ナビゲーションコンポーネントでメインをダブルクリックします。

メインステートマシンがコンテンツエリアに表示されます。

##### Step 2

ツールボックスから初期ステートをドラッグし、ステートマシンにドロップします。

##### Step 3

ツールボックスからビューステートをドラッグし、ステートマシンにドロップします。

ビューステートと共に、ビューがEB GUIDEモデルに追加されます。

#### Step 4

ビューステートの名前をHomeに変更します。

#### Step 5

コンテンツエリアで初期ステートをクリックします。

#### Step 6

初期ステートからHomeビューステートへの遷移を追加します。

#### Step 7

ナビゲーションコンポーネントでメインをクリックします。

#### Step 8

プロパティコンポーネントで、Dynamic state machine listチェックボックスを選択します。

ここまでの操作が完了すると、動的ステートマシンに関連付けられたEB GUIDEスクリプト関数を使用できます。

初期ステートとビューステートがメインステートマシンに追加され、ハプティック動的ステートマシンがメインステートマシンと並行して動作するようになりました。



### 内部遷移をメインステートマシンに追加する

このセクションでは、内部遷移の追加を行います。内部遷移を使用すると、動的ステートマシンをランタイム中に開始(プッシュ)および終了(ポップ)できます。

Prerequisite:

- 前のセクションの手順を完了していること。

#### Step 1

ナビゲーションコンポーネントでメインステートマシンをクリックします。

#### Step 2

プロパティコンポーネントで、内部遷移に移動し、+ をクリックします。

内部遷移がステートマシンに追加されます。内部遷移がナビゲーションコンポーネントに表示されます。

#### Step 3

内部遷移をさらに2つ追加します。

#### Step 4

ナビゲーションコンポーネントで1つ目の内部遷移をクリックします。

##### Step 4.1

プロパティコンポーネントに移動します。

##### Step 4.2

トリガーコンボボックスで、Volume upイベントを検索してダブルクリックします。

##### Step 4.3

アクションプロパティの横にある+ をダブルクリックします。

#### Step 4.4

次のEB GUIDEスクリプトを入力します。

```
function()
{
  dp:"Volume indicator" = dp:"Volume indicator" + 20
  f:pushDynamicStateMachine(popup_stack:Main, sm:"Volume control", 0)
}
```

#### Step 4.5

承認をクリックします。

アクションが遷移に追加されます。ナビゲーションコンポーネントで内部遷移の名前が `Volume up` に変わります。

#### Step 5

ナビゲーションコンポーネントで2つ目の内部遷移をクリックします。

##### Step 5.1

プロパティコンポーネントに移動します。

##### Step 5.2

トリガーコンボボックスで、`Volume down` イベントを検索してダブルクリックします。

##### Step 5.3

アクションプロパティの横にある **+** をクリックします。

##### Step 5.4

次のEB GUIDEスクリプトを入力します。

```
function()
{
  dp:"Volume indicator" = dp:"Volume indicator" - 20
  f:pushDynamicStateMachine(popup_stack:Main, sm:"Volume control", 0)
}
```

##### Step 5.5

承認をクリックします。

アクションが遷移に追加されます。ナビゲーションコンポーネントで内部遷移の名前が `Volume down` に変わります。

#### Step 6

ナビゲーションコンポーネントで3つ目の内部遷移をクリックします。

##### Step 6.1

プロパティコンポーネントに移動します。

##### Step 6.2

トリガーコンボボックスで、`Close volume control` イベントを検索してダブルクリックします。

### Step 6.3

アクションプロパティの横にある+ をクリックします。

### Step 6.4

次のEB GUIDEスクリプトを入力します。

```
function ()  
{  
  f:popDynamicStateMachine (popup_stack:Main,sm:"Volume control")  
}
```

### Step 6.5

承認をクリックします。

アクションが遷移に追加されます。ナビゲーションコンポーネントで内部遷移の名前が Close volume control に変わります。

3つの内部遷移を追加して、動的ステートマシンの開始と終了を可能にしました。Volume up と Volume down の内部遷移は、四角形Indicatorの位置を変更します。

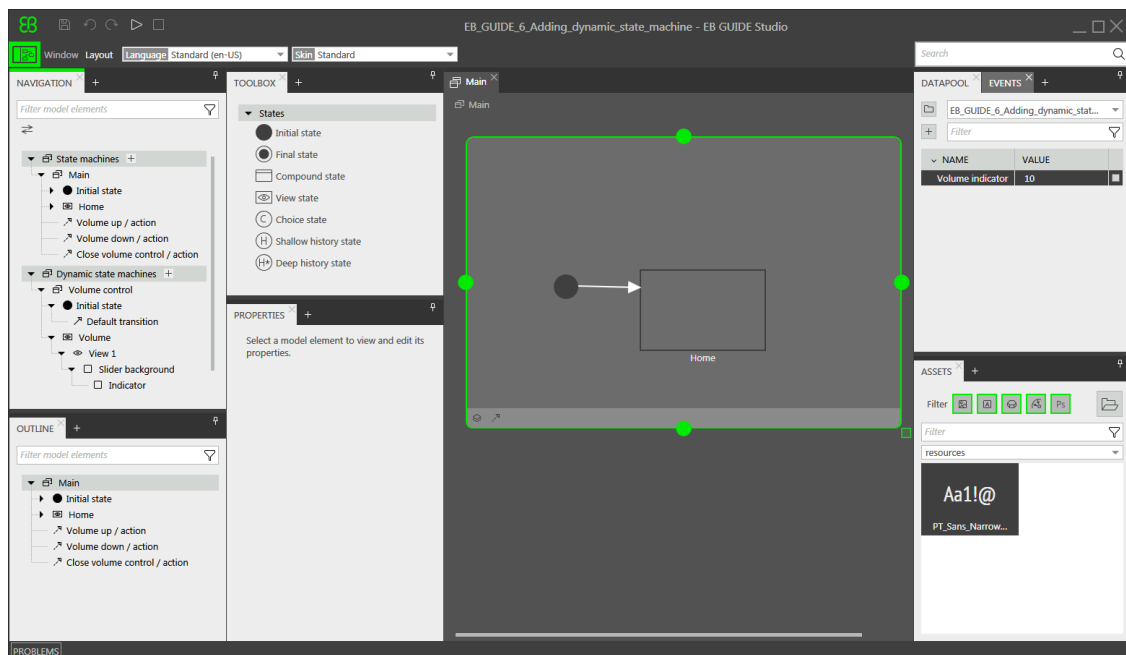


Figure 12.2. すべてのモデル要素が揃ったEB GUIDEモデル



## EB GUIDEモデルのシミュレーションとテストを開始する

Prerequisite:

- 前のセクションの手順を完了していること。

シミュレーションを開始するには、コマンドエリアで▶をクリックします。

シミュレーションとEB GUIDE Monitorが開始されます。EB GUIDEモデルにHomeビューステートが表示されます。

#### Step 1

EB GUIDE Monitorのイベントコンポーネントで、+ をクリックし、Volume upイベントを選択して、このイベントを発行するための🔊 をクリックします。

動的ステートマシンが起動し、スライダーインジケータが表示されます。動的ステートマシンはHomeビューステートをオーバーレイします。

Volume upまたはVolume downイベントを発行すると、黒いIndicator四角形が動きます。Close volume controlイベントが発行されると、スライダーがビューから消えます。

メインステートマシンにステートを追加した場合も、Volume control動的ステートマシンが他のステートをオーバーレイします。

## 12.2. チュートリアル: EB GUIDEスクリプトを使用したボタン動作のモデル化

### NOTE



#### デフォルトのウィンドウレイアウト

すべての手順およびスクリーンショットでは、デフォルトのウィンドウレイアウトが使用されています。手順に従う場合は、EB GUIDE Studioウィンドウをレイアウト > **Reset to default layout**を選択することによって、デフォルトレイアウトにリセットすることをお勧めします。

EB GUIDEスクリプトを使うと、ランタイムの最中にプロパティの値、アクション、または条件を変化させ、それらを評価することができます。

このセクションでは、EB GUIDEスクリプトを使用してボタンの動作をモデル化する手順を説明します。このボタンは、クリックするとサイズが大きくなり、定義した最大限のサイズに達したら元のサイズに戻ります。失敗を防ぐため、ここで説明する順番どおりに操作してください。

所要時間: 10分



### ウィジェットの追加

Prerequisite:

- メインステートマシンに、初期ステートとビューステートが含まれていること。
- 初期ステートにビューステートへの遷移があること。

- コンテンツエリアにビューが表示されます。

#### Step 1

ツールボックスから四角形をドラッグし、ビューにドロップします。

#### Step 2

ナビゲーションコンポーネントで四角形をクリックし、**F2**キーを押して四角形の名前をBackgroundに変更します。

#### Step 3

ツールボックスから四角形をドラッグし、ナビゲーションコンポーネント内でBackground四角形の子ウィジェットとして配置します。

#### Step 4

ナビゲーションコンポーネントで新しい四角形をクリックし、**F2**キーを押して四角形の名前をButtonに変更します。

#### Step 5

ツールボックスからラベルをドラッグし、ナビゲーションコンポーネント内でButton四角形の子ウィジェットとして配置します。

#### Step 6

ナビゲーションコンポーネントでラベルをクリックし、**F2**キーを押してラベルの名前をButton textに変更します。

ウィジェットの階層が、次のように変わります。



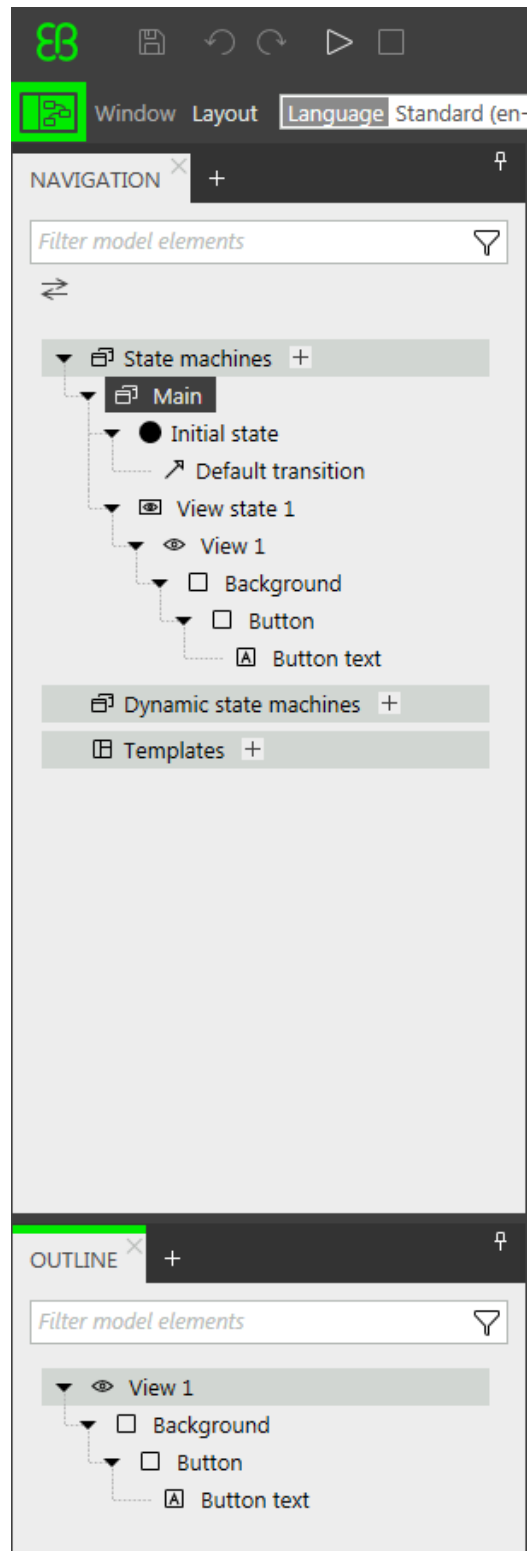


Figure 12.3. ウィジェットの階層



## 背景を設定する

Prerequisite:

- 前のセクションの手順を完了していること。

### Step 1

ナビゲーションコンポーネントでBackground四角形をクリックし、プロパティコンポーネントに移動します。

### Step 2

widthプロパティの横にある■ボタンをクリックします。

メニューが展開されます。

### Step 3

メニューのウィジェットプロパティへのリンクを追加をクリックします。

ダイアログが開きます。

### Step 4

ダイアログ内で、ビューに移動し、そのwidthプロパティをクリックします。

### Step 5

承認をクリックします。

ダイアログが閉じられます。■ボタンがwidthプロパティの横に表示されます。

### Step 6

Background四角形のheightプロパティを、ビューのheightプロパティにリンクします。

### Step 7

Background四角形のxプロパティを、ビューのxプロパティにリンクします。

### Step 8

Background四角形のyプロパティを、ビューのyプロパティにリンクします。

これで、Background四角形がビューのサイズと位置にぴったり重なります。



## ボタンの最大幅を定義する

データプールアイテムにボタンの最大幅の値を格納します。この値はランタイムの最中に変更できます。

Prerequisite:

- 前のセクションの手順を完了していること。

### Step 1

データプールコンポーネントで、+をクリックします。

メニューが展開されます。

#### Step 2

メニューで整数をクリックします。

Integerタイプの新しいデータプールアイテムが追加されます。

#### Step 3

データプールアイテムの名前をMaximum widthに変更します。

#### Step 4

Valueテキストボックスに、400と入力します。



### ボタンを設定する

Prerequisite:

- 前のセクションの手順を完了していること。

#### Step 1

ナビゲーションコンポーネントでButton四角形をクリックし、プロパティコンポーネントに移動します。

##### Step 1.1

heightテキストボックスに50と入力します。

##### Step 1.2

xテキストボックスに350と入力します。

##### Step 1.3

yテキストボックスに215と入力します。

##### Step 1.4

fillColorプロパティで青を選択します。

これで、ボタンが青色になりました。

#### Step 2

ウィジェット機能プロパティカテゴリで追加/削除をクリックします。

ウィジェット機能ダイアログが表示されます。

#### Step 3

使用可能なウィジェット機能で、入力処理カテゴリを展開してタッチ押下ウィジェット機能を選択します。

#### Step 4

承認をクリックします。

関連するウィジェット機能プロパティがButton四角形に追加され、プロパティコンポーネントに表示されます。

#### Step 5

touchPressedプロパティの横にある値列を選択し、{} をクリックします。

EB GUIDEスクリプトエディターが開きます。

### Step 6

既存のEB GUIDEスクリプトを次のコードで置き換えます。

```
function(v:touchId::int, v:x::int, v:y::int, v:fingerId::int)
{
    if (v:this.width > dp:"Maximum width") // If the button has grown
        // beyond its maximum size...
    {
        // ...reset its dimensions to the default values.
        v:this.height = 50
        v:this.width = 100
        v:this.x = 350
        v:this.y = 215
    }
    else // Otherwise...
    {
        // ... increase button size...
        v:this.width += 80
        v:this.height += 40

        // ...and move the button to keep it centered.
        v:this.x -= 40
        v:this.y -= 20
    }
    false
}
```

### Step 7

承認をクリックします。

Button四角形を設定し、ランタイム中にButton四角形のサイズを変更するEB GUIDEスクリプトを作成しました。



#### ボタンのテキストを設定する

Prerequisite:

- 前のセクションの手順を完了していること。

#### Step 1

ナビゲーションコンポーネントでButton textラベルをクリックし、プロパティコンポーネントに移動します。

#### Step 2

textテキストボックスにgrow!と入力します。

#### Step 3

Button textラベルのwidthプロパティを、Button四角形のwidthプロパティにリンクします。

#### Step 4

Button textラベルのheightプロパティを、Button四角形のheightプロパティにリンクします。

#### Step 5

xテキストボックスに0と入力します。

#### Step 6

yテキストボックスに0と入力します。

#### Step 7

horizontalAlignプロパティの横にあるcenter (1) を選択します。

以上の手順で、Button textラベルとButton四角形のサイズと位置が同じになりました。




### EB GUIDEモデルの保存およびテスト


Prerequisite:

- 前のセクションの手順を完了していること。

#### Step 1

プロジェクトを保存するには、コマンドエリアで をクリックします。

#### Step 2

シミュレーションを開始するには、コマンドエリアで をクリックします。

結果:

シミュレーションを開始すると作成したEB GUIDEモデルが開始され、次のように動作します。

1. 最初に、グレーの画面の中央に青色のボタンが次のように表示されます。

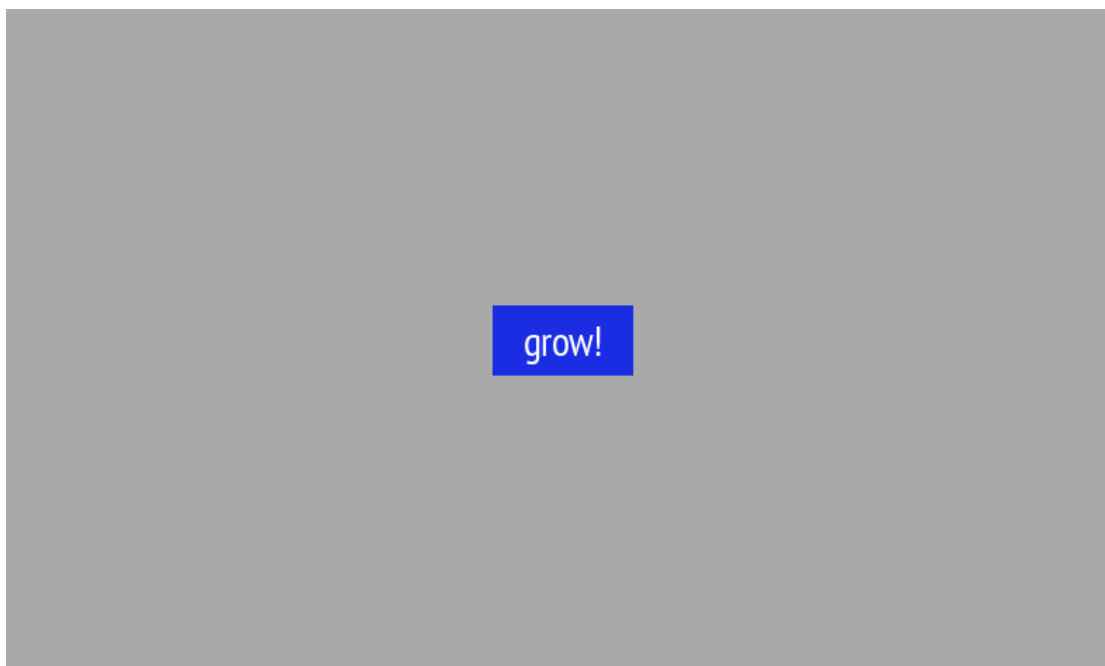


Figure 12.4. 結果

2. ボタンをクリックすると、ボタンのサイズが大きくなります。位置は画面中央から変化しません。
3. ボタンの幅がMaximum widthデータプールアイテムの値に達すると、ボタンは再び小さくなり、元のサイズと位置に戻ります。

## 12.3. チュートリアル: パスジェスチャーをモデル化する

### NOTE



#### デフォルトのウィンドウレイアウト

すべての手順およびスクリーンショットでは、デフォルトのウィンドウレイアウトが使用されています。手順に従う場合は、EB GUIDE Studioウィンドウをレイアウト > **Reset to default layout**を選択することによって、デフォルトレイアウトにリセットすることをお勧めします。

パスジェスチャーとは、タッチスクリーン上に指で描画されたか、その他の入力デバイスによって入力された形状のことです。

このセクションでは、パスジェスチャーをモデル化する手順を説明します。

所要時間: 10分



## ウィジェットの追加およびデフォルトウィジェットプロパティの設定

Prerequisite:

- メインステートマシンに、初期ステートとビューステートが含まれていること。
- 初期ステートにビューステートへの遷移があること。
- コンテンツエリアにビューが表示されていること。

### Step 1

ツールボックスから四角形をドラッグし、ビューにドロップします。

### Step 2

ツールボックスからラベルをドラッグし、四角形にドロップします。

ラベルは、四角形の子ウィジェットとして追加されます。

プロパティコンポーネントに、ラベルのプロパティが表示されます。

### Step 3

プロパティコンポーネントで、widthテキストボックスに500と入力します。

### Step 4

四角形を選択します。

プロパティコンポーネントに、四角形のプロパティが表示されます。

### Step 5

widthテキストボックスに500と入力します。

### Step 6

プロパティコンポーネントでfillColorに移動し、赤色を選択します。

2つのウィジェットを追加し、デフォルトウィジェットプロパティを設定しました。



## 四角形にウィジェット機能を追加する

ユーザーがウィジェット上で開始する形状を入力できるようにするには、パスジェスチャーウィジェット機能を四角形に追加します。入力形状が既知の形状集合とマッチングされ、マッチすればジェスチャーとして認識されます。

Prerequisite:

- 前のセクションの手順を完了していること。

### Step 1

四角形を選択します。

プロパティコンポーネントに、四角形のプロパティが表示されます。

#### Step 2

プロパティコンポーネントでウィジェット機能プロパティに移動し、追加/削除をクリックします。

ウィジェット機能ダイアログが表示されます。

#### Step 3

使用可能なウィジェット機能の下で、ジェスチャーカテゴリを展開し、Path gesturesを選択します。

タッチウィジェット機能が自動的に選択されます。ジェスチャーウィジェット機能でこれが必要なためです。

#### Step 4

承認をクリックします。

関連するウィジェット機能プロパティが四角形に追加され、プロパティコンポーネントに表示されます。

#### Step 5

パスジェスチャーウィジェット機能に対し、次のプロパティを編集します。

##### Step 5.1

onPathプロパティの横にある値列を選択し、`{}` をクリックします。

EB GUIDEスクリプトエディターが開きます。

##### Step 5.2

次のEB GUIDEスクリプトを入力します。

```
function (v:gestureId::int)
{
  v:this->"Label 1".text = "recognized path gesture #"
  + f:int2string (v:gestureId);
}
```

##### Step 5.3

承認をクリックします。

##### Step 5.4

onPathStartプロパティの横にある値列を選択し、`{}` をクリックします。

EB GUIDEスクリプトエディターが開きます。

##### Step 5.5

次のEB GUIDEスクリプトを入力します。

```
function ()
{
  v:this->"Label 1".text = "path gesture start";
}
```

##### Step 5.6

承認をクリックします。



#### Step 5.7

onPathNotRecognizedプロパティの横にある値列を選択し、`{ }` をクリックします。

EB GUIDEスクリプトエディターが開きます。

#### Step 5.8

次のEB GUIDEスクリプトを入力します。

```
function ()
{
  v:this->"Label 1".text = "shape not recognized";
}
```

#### Step 5.9

承認をクリックします。

#### Step 6

シミュレーションを開始するには、コマンドエリアで▶ をクリックします。

シミュレーションとEB GUIDE Monitorが開始されます。反応を確認するには、四角形の内部でマウスを使って図形を描画します。

## 12.4. チュートリアル: 動的コンテンツを使用したリストの作成

### NOTE



#### デフォルトのウィンドウレイアウト

すべての手順およびスクリーンショットでは、デフォルトのウィンドウレイアウトが使用されています。手順に従う場合は、EB GUIDE Studioウィンドウをレイアウト > **Reset to default layout**を選択することによって、デフォルトレイアウトにリセットすることをお勧めします。

インスタシエータを使うと、ランタイム中にリストを動的に作成することができます。リストタイプのデータプールアイテムに基づき、インスタシエータが事前定義済みのレイアウトでリストのすべての要素を表示します。データプールアイテムのコンテンツが変更されると、インスタシエータの表示も変更されます。

このセクションでは、動的コンテンツを使用してリストを作成する手順について説明します。リストの各要素は、ラベルの付いた四角形です。

所要時間: 15分



## データプールアイテムの追加

このセクションでは、`String list`タイプのデータプールアイテムを追加する手順について説明します。データプールアイテムは、インスタシエータのすべてのリスト要素に値を提供します。データプールアイテムのコンテンツが変更されると、インスタシエータの表示も変更されます。

Prerequisite:

- メインステートマシンに、初期ステートとビューステートが含まれていること。
- 初期ステートにビューステートへの遷移があること。

### Step 1

リストのコンテンツを表示するために、`String list`タイプのデータプールアイテムを追加します。

データプールコンポーネントで、**+**をクリックします。

メニューが展開されます。

### Step 2


メニューから文字列リストをクリックします。

`String list`タイプの新しいデータプールアイテムが追加されます。

### Step 3

データプールアイテムの名前を`MyStringList`に変更します。

### Step 4

値列を選択し、 ボタンをクリックします。

エディターが開きます。

#### Step 4.1

追加をクリックします。

新しいエントリーが表に追加されます。

#### Step 4.2

ValueテキストボックスにOneと入力します。

#### Step 4.3

Two、Three、Four、およびFiveの値を`MyStringList`データプールアイテムに追加します。

#### Step 4.4

承認をクリックします。

`String list`タイプのデータプールアイテムが追加されました。データプールアイテムには5つのエントリーが含まれています。

リストのコンテンツは、Valueプロパティの横に表示されます。



## ウィジェットの追加

### Prerequisite:

- 前のセクションの手順を完了していること。

### Step 1

ビューにウィジェットを追加するために、コンテンツエリアでビューステートをダブルクリックします。

ビューがコンテンツエリアに表示されます。

### Step 2

ナビゲーションコンポーネントでビューステートとビューを展開します。

### Step 3

ツールボックスからインスタシエータをビューにドラッグしてドロップします。インスタシエータの名前をMyInstantiatorに変更します。

### Step 4

ツールボックスから四角形をドラッグし、インスタシエータにドロップします。四角形の名前をMyRectangleに変更します。

### Step 5

ツールボックスからラベルをドラッグし、四角形にドロップします。ラベルの名前をMyLabelに変更します。

ウィジェットの階層が、次のように変わります。

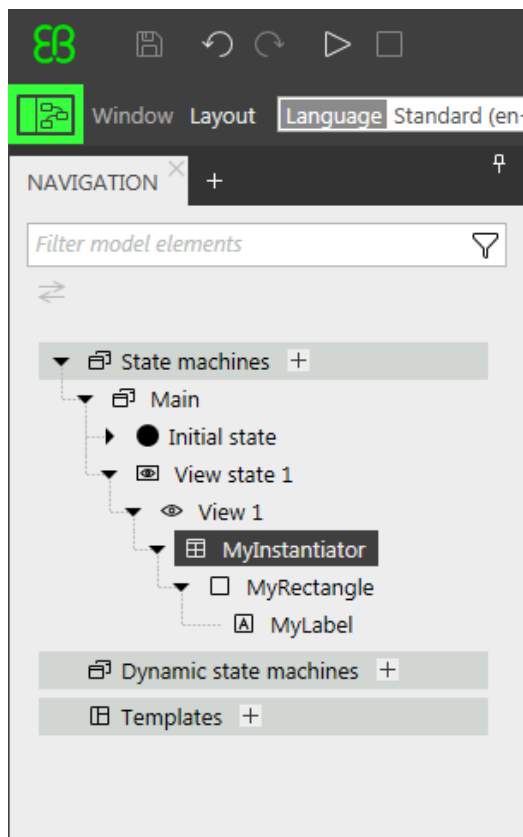


Figure 12.5. インスタンスエータを含むウィジェット階層



## インスタンスエータの設定

Prerequisite:

- 前のセクションの手順を完了していること。

### Step 1

MyInstantiatorのプロパティを変更するには、インスタンスエータを選択し、プロパティコンポーネントに移動します。

### Step 2

widthテキストボックスとheightテキストボックスに300と入力します。

### Step 3

xテキストボックスに250と入力します。

### Step 4

yテキストボックスに150と入力します。

### Step 5

リストの長さを動的に計算するために、条件スクリプトを追加します。

ユーザー定義プロパティカテゴリで、+をクリックします。

メニューが展開されます。

Step 5.1

メニューから条件スクリプトをクリックします。

Step 5.2

プロパティの名前をcalculateNumItemsに変更します。

Step 5.3

プロパティの横にある値列を選択し、{} をクリックします。

EB GUIDEスクリプトエディターが開きます。

Step 5.4

トリガーの下に、dp:MyStringListと入力します。

Step 5.5

**On Trigger**スクリプトに次のように入力します。

```
function (v:arg0::bool)
{
  v:this.numItems = length dp:MyStringList;
  false
}
```

MyStringListのコンテンツに応じてリストエントリー数を自動的に変更するスクリプトが追加されました。

Step 6

インスタンスエータ内のラベルをすべて整列させるために、レイアウトを追加します。

ウィジェット機能プロパティカテゴリで追加/削除をクリックします。

ウィジェット機能ダイアログが表示されます。

Step 6.1

使用可能なウィジェット機能の下からレイアウトカテゴリを展開し、ボックスレイアウトウィジェット機能を選択してラベルを横に整列させます。

関連するウィジェット機能プロパティがインスタンスエータに追加され、プロパティコンポーネントに表示されます。

Step 6.2

承認をクリックします。

Step 6.3

gapテキストボックスに5と入力して、各リスト要素の間隔を5ピクセルに設定します。

Step 6.4

layoutDirectionドロップダウンリストボックスからvertical (=1)を選択して、ラベルを整列させます。

リストの外観を定義し、リストアイテム数を動的に適用するインスタンスエータが設定されました。



## リスト要素テキストの設定

Prerequisite:

- 前のセクションの手順を完了していること。

### Step 1

ラベルの外観を変更するには、MyLabelを選択し、プロパティコンポーネントに移動します。

### Step 2

xテキストボックスとyテキストボックスに0と入力します。

### Step 3

ラベルのwidth プロパティから四角形のwidthプロパティへのリンクを追加。

#### Step 3.1

widthプロパティの横にある■ ボタンをクリックします。

メニューが展開されます。

#### Step 3.2

メニューのウィジェットプロパティへのリンクを追加をクリックします。

ダイアログが開きます。

#### Step 3.3

ダイアログで、四角形に移動し、そのwidthプロパティを選択します。

#### Step 3.4

承認をクリックします。

ダイアログが閉じられます。■ ボタンがwidthプロパティの横に表示されます。

### Step 4

ラベルのheight プロパティから四角形のheightプロパティへのリンクを追加。

### Step 5

horizontalAlignプロパティの横にあるcenter (1) を選択します。

ラベルの外観が変更され、四角の中心に表示されました。



## リスト要素の設定

Prerequisite:

- 前のセクションの手順を完了していること。

### Step 1

四角形の外観を変更するために、四角形を選択してプロパティコンポーネントに移動します。

### Step 2

リスト要素で利用可能な幅が必ず使用されるように、四角形のwidthプロパティからインスタシエータのwidthプロパティへのリンクを追加します。

### Step 3

heightテキストボックスに50と入力します。

### Step 4

リストの各ラインに対して一意の位置を定義するために、ラインインデックスウィジェット機能を追加します。

#### Step 4.1

ウィジェット機能プロパティカテゴリで追加/削除をクリックします。

ウィジェット機能ダイアログが表示されます。

#### Step 4.2

使用可能なウィジェット機能からリスト管理カテゴリを展開し、ラインインデックスウィジェット機能を選択します。

lineIndexプロパティが四角形のプロパティに追加されます。

### Step 5

リストのラベルにMyStringListのコンテンツを入れるため、条件スクリプトを追加します。

#### Step 5.1

ユーザー定義プロパティカテゴリの横にある+をクリックします。

メニューが展開されます。

#### Step 5.2

メニューから条件スクリプトをクリックします。

#### Step 5.3

プロパティの名前をsetTextに変更します。

#### Step 5.4

setTextプロパティの横にある値列を選択し、{} をクリックします。

EB GUIDEスクリプトエディターが開きます。

#### Step 5.5

トリガーの下に、v:this.lineIndexおよびdp:MyStringListと入力します。

#### Step 5.6

On Triggerスクリプトに次のように入力します。

```
function (v:arg0::bool)
{
    v:this->MyLabel.text=dp:MyStringList[v:this.lineIndex];
    false
}
```

四角形の表示が変更されました。setTextプロパティにより、MyStringListのコンテンツが自動的にMyStringListのラベルに入るようになりました。



## EB GUIDEモデルのテスト

### Prerequisite:

- 前のセクションの手順を完了していること。

### Step 1

シミュレーションを開始するには、コマンドエリアで▶をクリックします。

### 結果:

`MyStringList`にはデータプールアイテムが5つ含まれるため、OneからFiveのラベルが付いた5つの四角形が、縦方向に並んで表示されます。

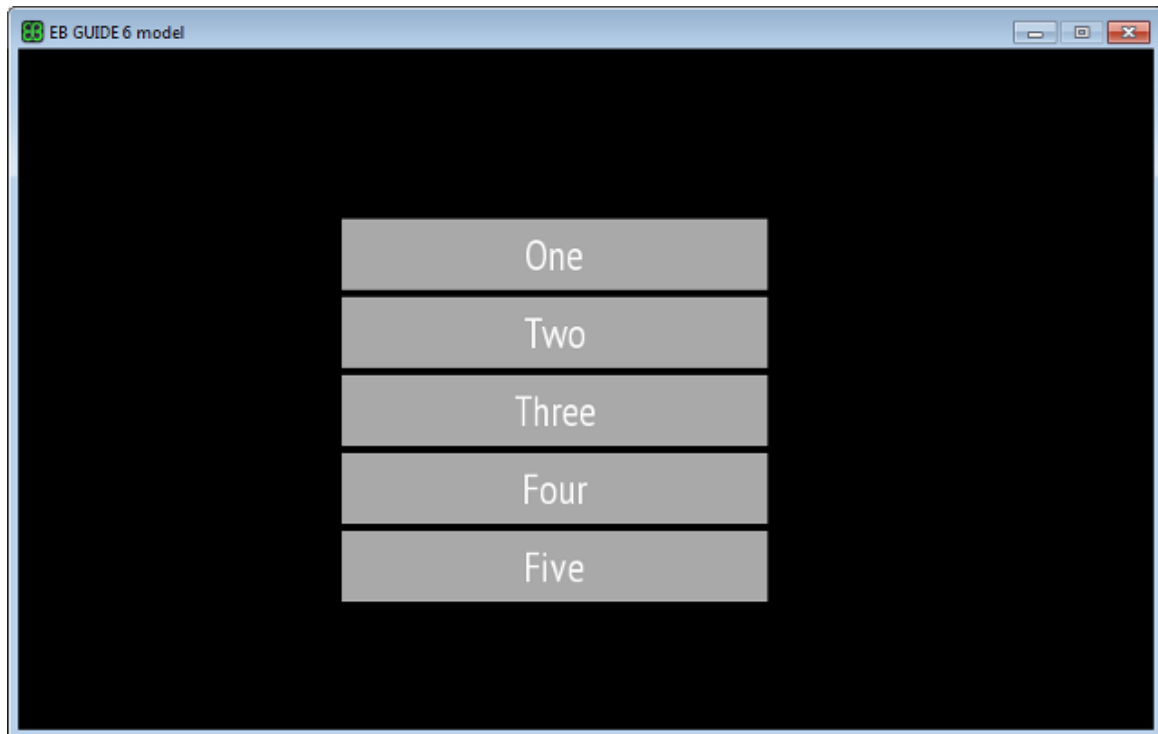


Figure 12.6. インスタンスエータで作成されたリスト



## 12.5. チュートリアル: 画面内で楕円を移動する

### NOTE



#### デフォルトのウィンドウレイアウト

すべての手順およびスクリーンショットでは、デフォルトのウィンドウレイアウトが使用されています。手順に従う場合は、EB GUIDE Studioウィンドウをレイアウト > **Reset to default layout**を選択することによって、デフォルトレイアウトにリセットすることをお勧めします。

このセクションでは、楕円をアニメーション化し、シミュレーションを開始したとき画面内で楕円が移動を繰り返すようにする方法を説明します。

所要時間: 5分



### ウィジェットを追加

以下の手順で、ビューに3つのウィジェットを追加し、ウィジェットの階層を編成します。

Prerequisite:

- コンテンツエリアに、メインステートマシンが表示されていること。
- メインステートマシンに、初期ステートとビューステートが含まれていること。
- 初期ステートにビューステートへの遷移があること。

#### Step 1

コンテンツエリアでビューステートをダブルクリックします。

ビューがコンテンツエリアに表示されます。

#### Step 2

ツールボックスから楕円をドラッグし、ビューにドロップします。

#### Step 3

ツールボックスからアニメーションをドラッグし、楕円にドロップします。

#### Step 4

ナビゲーションコンポーネントでアニメーションをクリックし、**F2**キーを押します。アニメーションの名前をMyAnimationに変更します。

ここでシミュレーションを開始すると、ビューに楕円が表示されますが、まだ楕円は動きません。



### タイプのユーザー定義プロパティの追加 `Conditional script`

以下の手順では、楕円にユーザー定義プロパティを追加します。条件スクリプトのプロパティを使用し、シミュレーションの際にアニメーションが始まると楕円を描画するようにします。

Prerequisite:

- 前のセクションの手順を完了していること。

#### Step 1

楕円を選択します。

#### Step 2

プロパティコンポーネントでユーザー定義プロパティカテゴリに移動し、**+**をクリックします。

メニューが展開されます。

#### Step 3

メニューでConditional scriptをクリックします。

Conditional scriptタイプのユーザー定義プロパティが楕円に追加されます。

#### Step 4

プロパティの名前をstartAnimationに変更します。

#### Step 5

startAnimationプロパティの横にある値列を選択し、**{ }** をクリックします。

EB GUIDEスクリプトエディターが開きます。

#### Step 6

次のEB GUIDEスクリプトを入力します。

```
function(v:arg0::bool)
{
  f:animation_play(v:this->^->MyAnimation)
}
```



### アニメーションの可視化

このセクションでは、アニメーションを可視化する手順を説明します。

Prerequisite:

- 前のセクションの手順を完了していること。
- コンテンツエリアにView 1ビューが表示されます。

#### Step 1

アニメーションエディターに移動します。アニメーションプロパティの横にある**+**をクリックして、View 1を選択します。

メニューが展開されます。

#### Step 2

Ellipse 1の下でxプロパティ、リニア補間曲線の順に選択します。

#### Step 3

承認をクリックします。

■ボタンがtargetプロパティの横に表示されます。

#### Step 4

endプロパティを、ビューのwidthプロパティにリンクします。

これらを設定すると、アニメーションの開始時に、楕円のxプロパティがゼロからビューの幅の値に変化します。したがって、楕円がビューの左の境界線から右の境界線まで移動します。

#### Step 5

アニメーションを無限に繰り返して実行するには、repeatプロパティに0を入力します。

#### Step 6

プロジェクトを保存します。

#### Step 7

シミュレーションを開始するには、コマンドエリアで▶をクリックします。

結果:

楕円がビューの左端から右端へ 移動を繰り返します。

## 12.6. チュートリアル: データプールアイテムに言語依存テキストを追加する

### NOTE



#### デフォルトのウィンドウレイアウト

すべての手順およびスクリーンショットでは、デフォルトのウィンドウレイアウトが使用されています。手順に従う場合は、EB GUIDE Studioウィンドウをレイアウト > **Reset to default layout**を選択することによって、デフォルトレイアウトにリセットすることをお勧めします。

EB GUIDEでは、テキストをユーザーの選択した言語で表示することができます。このセクションでは、英語、フランス語、ドイツ語のヒューマンマシンインターフェースで変化するラベルをモデル化する手順について説明します。

所要時間: 15分

## NOTE



### 言語依存関係の前提条件

データプールアイテムに言語サポートを追加するには、次の操作を行います。

- ▶ Valueプロパティが別のデータプールアイテムまたはウィジェットプロパティにリンクされている場合は、そのリンクを削除します。
- ▶ Valueプロパティがスクリプト値である場合は、プロパティをプレーン値に変換します。



### ウィジェットプロパティとデータプールアイテムのリンク設定

このセクションでは、ラベルのtextプロパティをデータプールアイテムにリンクする手順について説明します。ランタイム中に、表示テキストがデータプールアイテムから提供されます。

Prerequisite:

- EB GUIDEモデルに3つの言語、英語、ドイツ語、フランス語が追加されていること。言語1の名前がGermanに、言語2の名前がFrenchに設定されていること。
- メインステートマシンに、初期ステートとビューステートが含まれていること。
- 初期ステートにビューステートへの遷移があること。
- コンテンツエリアにビューが表示されます。
- ビューステートにラベルが含まれていること。

#### Step 1

ラベルをクリックします。

#### Step 2

プロパティコンポーネントでtextプロパティに移動し、プロパティの横にある  ボタンをクリックします。

#### Step 3

メニューのデータプールアイテムへのリンクを追加をクリックします。

ダイアログが開きます。

#### Step 4

新しいデータプールアイテムを追加するには、テキストボックスにWelcome\_textと入力します。

#### Step 5

データプールアイテムを追加をクリックします。

#### Step 6

承認をクリックします。

データプールアイテムWelcome\_textが追加されます。

コンテンツエリアで、ラベルにテキストが何も表示されなくなります。



## データプールアイテムに言語依存テキストを入力する

このセクションでは、データプールアイテムに言語依存テキストを追加する手順について説明します。言語ごとに、Valueプロパティに異なるテキストが入ります。

Prerequisite:

- 前のセクションの手順を完了していること。

### Step 1

データプールコンポーネントで、Welcome\_textデータプールアイテムをクリックします。

### Step 2

■ ボタンをクリックします。

### Step 3

メニューの**Add language support**をクリックします。

プロパティコンポーネントに言語プロパティが表示されます。

### Step 4

データプールコンポーネントで、ValueテキストボックスにWelcomeと入力します。

コンテンツエリアで、ラベルにWelcomeと表示されます。

### Step 5

プロパティコンポーネントに移動します。

### Step 6

Germanテキストボックスに、Willkommenと入力します。

左上隅のLanguageボックスで、言語をGermanに変更します。

コンテンツエリアで、ラベルにWillkommenと表示されます。

### Step 7

Frenchテキストボックスに、Bienvenueと入力します。

左上隅のLanguageボックスで、言語をFrenchに変更します。

コンテンツエリアで、ラベルにBienvenueと表示されます。

英語、ドイツ語、およびフランス語の言語サポートを追加し、言語依存のテキストラベルを定義しました。



## ランタイム処理中の言語の変更

このセクションでは、ランタイム中に言語を変更するためのスクリプトを作成する手順について説明します。ユーザーがラベルをクリックするたびに表示言語が変更されます。

Prerequisite:

- 前のセクションの手順を完了していること。

Step 1

データプールコンポーネントで、+をクリックします。

メニューが展開されます。

Step 2

メニューでIntegerをクリックします。

Integerタイプのデータプールアイテムが追加されます。

Step 3

データプールアイテムの名前をSelectedLanguageに変更します。

Step 4

ナビゲーションコンポーネントで、Label 1ラベルをクリックします。

Step 5

プロパティコンポーネントでウィジェット機能プロパティに移動し、追加/削除をクリックします。

ウィジェット機能ダイアログが表示されます。

Step 6

使用可能なウィジェット機能で、入力処理カテゴリを展開してタッチ押下ウィジェット機能を選択します。

Step 7

承認をクリックします。

関連するウィジェット機能プロパティがラベルに追加され、プロパティコンポーネントに表示されます。

Step 8

touchPressedプロパティの横にある値列を選択し、{} をクリックします。

EB GUIDEスクリプトエディターが開きます。

Step 9

既存のEB GUIDEスクリプトを次のコードで置き換えます。

```
function(v:touchId::int, v:x::int, v:y::int, v:fingerId::int)
{
    if (dp:SelectedLanguage == 0) // Standard selected
    {
        f:language(1:German)
        dp:SelectedLanguage = 1
    }
    else if (dp:SelectedLanguage == 1) // German selected
    {
        f:language(1:French)
        dp:SelectedLanguage = 2
    }
}
```

```
else if (dp:SelectedLanguage == 2) // French selected
{
    f:language(1:Standard)
    dp:SelectedLanguage = 0
}
false
}
```

#### Step 10

承認をクリックします。

ラベルを設定し、ランタイム中にラベルの言語を変更するEB GUIDEスクリプトを作成しました。

結果:

String型のデータプールアイテムが、EB GUIDEモデルに追加されました。データプールアイテムには、言語ごとに違う値が入っています。英語の値はWelcome、ドイツ語の値はWillkommen、フランス語の値はBienvenueです。このデータプールアイテムは、ラベルのtextプロパティにリンクされています。EB GUIDEモデルの言語を変更するたびに、ラベルのテキストも変わります。

## 12.7. チュートリアル: 3Dグラフィックの操作

### NOTE



#### デフォルトのウィンドウレイアウト

すべての手順およびスクリーンショットでは、デフォルトのウィンドウレイアウトが使用されています。手順に従う場合は、EB GUIDE Studioウィンドウをレイアウト > **Reset to default layout**を選択することによって、デフォルトレイアウトにリセットすることをお勧めします。

EB GUIDE Studioでは、EB GUIDEモデルに3Dグラフィックを使用できます。

このセクションでは、EB GUIDEモデルに3Dグラフィックを追加する手順について説明します。扱うのは3Dグラフィックをインポートする方法と、インポートした3Dグラフィックの外観をウィジェット機能で変更する方法です。最高の結果を得られるよう、ここで説明する順番どおりに操作してください。

## NOTE



### 3Dグラフィック

3Dグラフィックファイルを作成するには、サードパーティの3Dモデリングソフトウェアを使用します。

OpenGL ES 2.0以降のレンダラーに限り、3Dグラフィックを表示できます。グラフィックドライバがレンダラーのバージョンと互換性があることを確認してください。サポートされている3Dグラフィック形式は、COLLADA (.dae)およびFilmbox (.fbx)です。失敗を防ぐため、Filmbox形式を使用します。

メッシュにテクスチャを適用するには、3Dオブジェクトにテクスチャ座標を設定する必要があります。テクスチャ座標の追加には、サードパーティの3Dモデリングソフトウェアを使用してください。

所要時間: 15分



## 3Dグラフィックのインポート

このセクションでは、EB GUIDEプロジェクトに3Dグラフィックファイルをインポートする手順について説明します。

Prerequisite:

- コンテンツエリアに、メインステートマシンが表示されていること。
- メインステートマシンに、初期ステートとビューステートが含まれていること。
- 初期ステートにビューステートへの遷移があること。
- 3Dグラフィックファイルが使用可能になっていること。ファイルに、カメラ、光源、およびメッシュと少なくとも1つの材質を含む1つのオブジェクトが含まれていること。

### Step 1

コンテンツエリアでビューステートをダブルクリックします。

ビューがコンテンツエリアに表示されます。

### Step 2

ツールボックスからシーングラフをビューにドラッグしてドロップします。

ビューに空の矩形が表示されます。

### Step 3

シーングラフの名前をMy3DGraphicに変更します。

### Step 4

プロパティコンポーネントでファイルのインポートをクリックします。

ダイアログが開きます。

### Step 5

3Dグラフィックファイルが格納されているディレクトリに移動します。



#### Step 6

3Dグラフィックファイルを選択します。

#### Step 7

開くをクリックします。

インポートが開始します。インポートに成功しましたダイアログが表示されます。ここでインポートログファイルを確認することもできます。

#### Step 8

OKをクリックします。

ビューに3Dグラフィックが表示されます。インポートしたウィジェットツリーが、シーングラフを親ノードにしてナビゲーションコンポーネントに表示されます。My3DGraphic には、3Dグラフィックファイルのコンテンツに応じて少なくとも材質、カメラ、そして複数の子ウィジェットを持つメッシュ1つRootNodeが含まれます。



### ウィジェットの追加

このセクションでは、3Dグラフィックに別の光源を追加する手順について説明します。

Prerequisite:

- 前のセクションの手順を完了していること。

#### Step 1

ナビゲーションコンポーネントでRootNodeを展開します。

#### Step 2

ツールボックスから指向性ライトをドラッグし、RootNodeにドロップします。

My3DGraphicに指向性ライトが追加されました。この指向性ライトは、RootNodeの変形プロパティを使用して操作および変換できます。

#### Step 3

光源を追加してRootNodeシーングラフ以外のデフォルトのウィジェットプロパティで配置するには、次のようにします。

##### Step 3.1

ツールボックスからシーングラフノードをドラッグし、RootNodeにドロップします。

##### Step 3.2

シーングラフノードの名前をMyLightに変更します。

##### Step 3.3

ツールボックスから指向性ライトをドラッグし、MyLightにドロップします。

My3DGraphicに指向性ライトが追加されました。指向性ライトの配置を変更するには、MyLightのプロパティを変更します。



## メッシュの変更

### Prerequisite:

- 前のセクションの手順を完了していること。
- `$GUIDE_PROJECT_PATH/<project name>/resources/<3D graphic name>`ディレクトリに、追加の `.ebmesh`ファイルが含まれていること。

### Step 1

ナビゲーションコンポーネントで `Mesh 1` をクリックし、プロパティコンポーネントに移動します。

### Step 2

`mesh`コンボボックスで、前述のリソースフォルダーから `.ebmesh`ファイルを選択します。

ビューに表示されているシーングラフのメッシュが変更されます。

### Step 3

または、アセットコンポーネントから `.ebmesh`ファイルをドラッグして、`mesh`ドロップダウンリストボックスにドロップします。

ビューに表示されているシーングラフのメッシュが変更されます。



## テクスチャの変更

ここからは、3Dグラフィックにテクスチャを追加し、変更する手順について説明します。

### Prerequisite:

- 前のセクションの手順を完了していること。
- `$GUIDE_PROJECT_PATH/<project name>/resources/<3D graphic name>`ディレクトリに、`.png`または `.jpg`イメージファイルが含まれていること。

### Step 1

ナビゲーションコンポーネントで材質をクリックし、プロパティコンポーネントに移動します。

### Step 2

ウィジェット機能プロパティカテゴリで追加/削除をクリックします。

ウィジェット機能ダイアログが表示されます。

### Step 3

使用可能なウィジェット機能で **3D**カテゴリを展開して、例えばディフューズテクスチャなどのテクスチャウィジェット機能を選択します。

### Step 4

承認をクリックします。

関連するウィジェット機能プロパティが材質に追加され、プロパティコンポーネントに表示されます。

#### Step 5

プロパティコンポーネントで、diffuseTextureコンボボックスからイメージを選択します。

ビューに表示されているシーングラフのテクスチャが変更されます。

#### NOTE



#### 3Dウィジェット機能の使用

この説明は、カテゴリ3Dにある以下のウィジェット機能に該当します。

- ▶ アンビエントテクスチャ
- ▶ ディフューズテクスチャ
- ▶ エミッシブテクスチャ
- ▶ ライトマップテクスチャ
- ▶ ノーマルマップテクスチャ
- ▶ 不透明テクスチャ
- ▶ リフレクションテクスチャ
- ▶ スペキュラテクスチャ



#### 3Dオブジェクトの複数回表示

このセクションでは、3Dグラフィックの3Dオブジェクトを複数回表示するために別のカメラを追加する手順について説明します。同じオブジェクトを別の視点から移すことが可能になります。

Prerequisite:

- 前のセクションの手順を完了していること。

#### Step 1

ナビゲーションコンポーネントでMy3DGraphicをクリックし、プロパティコンポーネントに移動します。

#### Step 2

widthテキストボックスに800、heightテキストボックスに480と入力します。

My3DGraphicシーングラフにビューのサイズが設定されます。

#### Step 3

ナビゲーションコンポーネントでRootNodeおよびCamera001を展開します。

#### Step 4

Camera 1をクリックし、プロパティコンポーネントに移動します。

#### Step 5

ウィジェット機能プロパティカテゴリで追加/削除をクリックします。

ウィジェット機能ダイアログが表示されます。

#### Step 6

使用可能なウィジェット機能で、**3D**カテゴリを展開してカメラビューポートを選択します。

#### Step 7

承認をクリックします。

関連するウィジェット機能プロパティがCamera 1に追加され、プロパティコンポーネントに表示されます。

#### Step 8

ツールボックスからカメラをドラッグし、シーングラフノードCamera001にドロップします。

2つ目のカメラが追加されました。

#### Step 9

Camera 2をクリックし、プロパティコンポーネントに移動します。

#### Step 10

nearPlane、farPlane、fieldOfViewテキストボックスに、Camera 1と同じ値を入力します。

Camera 1とCamera 2の視点の位置が同じになりました。

#### Step 11

ウィジェット機能プロパティカテゴリで追加/削除をクリックします。

ウィジェット機能ダイアログが表示されます。

#### Step 12

使用可能なウィジェット機能で、**3D**カテゴリを展開してカメラビューポートを選択します。

#### Step 13

承認をクリックします。

関連するウィジェット機能プロパティがCamera 2に追加され、プロパティコンポーネントに表示されます。

#### Step 14

プロパティコンポーネントで、viewportXおよびviewportYテキストボックスに100と入力します。

ビューで3Dオブジェクトが2回、x座標とy座標を変えて表示されます。

## 12.8. チュートリアル: ガンマの正しいレンダリング

### NOTE



#### デフォルトのウィンドウレイアウト

すべての手順およびスクリーンショットでは、デフォルトのウィンドウレイアウトが使用されています。手順に従う場合は、EB GUIDE Studioウィンドウをレイアウト > **Reset to default layout**を選択することによって、デフォルトレイアウトにリセットすることをお勧めします。

EB GUIDE Studioでは、以下に対するガンマ補正を実行できます。

- ▶ ディスプレイ

- ▶ イメージ
- ▶ テクスチャ

このセクションでは、ガンマ補正の設定方法について説明します。最高の結果を得られるよう、ここで説明する順番どおりに操作してください。

所要時間: 15分




## ディスプレイのガンマエンコーディングの設定

このセクションでは、ガンマエンコーディングされた値をディスプレイに出力するようEB GUIDE Studioを設定する手順を説明します。

Prerequisite:

- コンテンツエリアに、メインステートマシンが表示されていること。
- メインステートマシンに、初期ステートとビューステートが含まれていること。
- 初期ステートにビューステートへの遷移があること。

### Step 1

をクリックします。

プロジェクトセンターが表示されます。

### Step 2

ナビゲーションエリアで設定 > プロファイルの順にクリックします。

### Step 3

sRGBを認識できる各プロファイルについて、以下のように設定します。

#### Step 3.1

コンテンツエリアで、シーントabをクリックします。

#### Step 3.2

colorModeドロップダウンリストボックスから、以下を選択します。

- ▶ レンダリングパイプラインがグラフィックス処理ユニットのハードウェアsRGBサポートを使用する場合は、次のものを選択します。32-bit sRGB (=4)
- ▶ レンダリングハードウェアがsRGBをサポートしていない場合は、32-bit sRGB (=5)を選択します。

編集モードで変更を適用するには、EB GUIDE Studioを再起動します。

## NOTE



### ハードウェアのレンダリング

OpenGL 3レンダラーには必ずハードウェアsRGBサポートがあります。ハードウェアが適切なOpenGL ES拡張機能によってハードウェアsRGBをサポートしている場合、OpenGL ES 2.0 APIを使用するOpenGLレンダラーはハードウェアsRGBサポートのみを使用します。これは自動的に検出されます。OpenGL ES 2.0ハードウェアがsRGBをサポートしていない場合、レンダラーは変換にフラグメントシェーダーを使用する32-bit sRGB (Emulated) (=5)に自動的に戻ります。システムがOpenGL ES 3.0をサポートしていても、ハードウェアsRGBモードはすべてのシステムで機能するわけではないことに注意してください。その場合は32-bit sRGB (Emulated) (=5)に切り替えます。



## イメージのガンマエンコーディングの設定

### Prerequisite:

- 前のセクションの手順を完了していること。
- イメージファイルはリソースディレクトリで利用可能です。

#### Step 1

プロジェクトエディターに移動して、ビューをダブルクリックします。ツールボックスからイメージをビューにドラッグしてドロップします。

#### Step 2

プロパティコンポーネントで、imageドロップダウンリストボックスからイメージファイルを選択します。

たいていの場合、イメージは非常に明るく表示されます。

これはシーンプロパティで設定されたsRGBカラーモードが原因です。すでにガンマ補正されているイメージにガンマ補正が適用されています。

#### Step 3

ガンマエンコーディングされるようにイメージを設定するには、プロパティコンポーネントでsRGBを選択します。

これで、イメージがブレンド操作で正しく表示および処理されます。



## テクスチャのガンマエンコーディングの設定

### Prerequisite:

- 前のセクションの手順を完了していること。
- 少なくとも1つのテクスチャ3Dオブジェクトを含む3Dファイル(.fbxファイルなど)がディフューズテクスチャで使用できること。

#### Step 1

ツールボックスからシーングラフをビューにドラッグしてドロップします。

#### Step 2

プロパティコンポーネントで、ファイルのインポートをクリックし、3Dファイルを選択します。

#### Step 3

ナビゲーションコンポーネントに、インポートされたシーングラフを配置します。シーングラフ構造内で、ディフューズテクスチャを使用する材質ウィジェットを選択します。

#### Step 4

プロパティコンポーネントのウィジェット機能プロパティで、追加/削除をクリックします。

#### Step 5

3Dカテゴリから、ディフューズテクスチャを選択します。

#### Step 6

diffuseSRGBプロパティを選択します。

このテクスチャはガンマエンコーディングされたイメージとして扱われ、ライトの計算で使用される前にリニアライズされます。

## 12.9. チュートリアル: 表示遷移アニメーションの使用

### NOTE



#### デフォルトのウィンドウレイアウト

すべての手順およびスクリーンショットでは、デフォルトのウィンドウレイアウトが使用されています。手順に従う場合は、EB GUIDE Studioウィンドウをレイアウト > **Reset to default layout**を選択することによって、デフォルトレイアウトにリセットすることをお勧めします。

表示遷移アニメーション(VTA)は、あるビューから別のビューへの移動時に実行されるアニメーションです。以下では、こうしたアニメーションを作成する手順を説明します。ビューの変更時に再生されるビューやアニメーションによってモデルを作成することになります。以下の要素を作成します。

- ▶ 2つのビューステート
- ▶ ナビゲーション要素(ボタン、ラベルなど)
- ▶ ビューステートの変化をトリガーするイベント
- ▶ 別のステートへの遷移時に再生されるアニメーション

所要時間: 30分



#### 最初のビューステートの作成

最初のビューとボタンを作成します。

Prerequisite:

- コンテンツエリアに、メインステートマシンが表示されていること。
- メインステートマシンに、初期ステートとビューステートが含まれていること。
- 初期ステートにビューステートへの遷移があること。

Step 1

ナビゲーションコンポーネントで、ビューステートの名前をFirstStateに、ビューの名前をFirstViewにそれぞれ変更します。

Step 2

FirstViewを開きます。

Step 3

ツールボックスコンポーネントから、四角形をFirstView内にドラッグし、その名前をRectNextViewに変更します。

この四角形は、遷移をトリガーするボタンのためのものです。

Step 4

プロパティコンポーネントでウィジェット機能プロパティカテゴリに移動し、追加/削除をクリックします。ウィジェット機能ダイアログが表示されます。

Step 5

使用可能なウィジェット機能の下で、エフェクトカテゴリを展開し、枠を選択します。

Step 6

承認をクリックします。

関連するウィジェット機能プロパティがプロパティコンポーネントに表示されます。

Step 7

プロパティコンポーネントで、次のように入力します。

- ▶ widthテキストボックスに、220と入力します。
- ▶ heightテキストボックスに、70と入力します。
- ▶ xテキストボックスに、290と入力します。
- ▶ yテキストボックスに、150と入力します。
- ▶ fillColorを黒に設定します。
- ▶ borderThicknessテキストボックスに、2と入力します。
- ▶ borderColorを白に設定します。

Step 8

ツールボックスコンポーネントから、ラベルをナビゲーションコンポーネント内にドラッグし、FirstViewの子ウィジェットとして追加します。

Step 9

ラベルの名前をLabelNextViewに変更します。



#### Step 10

プロパティコンポーネントで、次のように入力します。

- ▶ textテキストボックスに、Go to the next viewと入力します。
- ▶ fontテキストボックスに、25と入力します。
- ▶ horizontalAlignを中央に設定します。

#### Step 11

ラベルの寸法を四角形の寸法にリンクします。次のプロパティをリンクします。

- ▶ LabelNextViewのwidthプロパティをRectNextViewのwidthにリンクします。
- ▶ LabelNextViewのheightプロパティをRectNextViewのheightにリンクします。
- ▶ LabelNextViewのxプロパティをRectNextViewのxにリンクします。
- ▶ LabelNextViewのyプロパティをRectNextViewのyにリンクします。

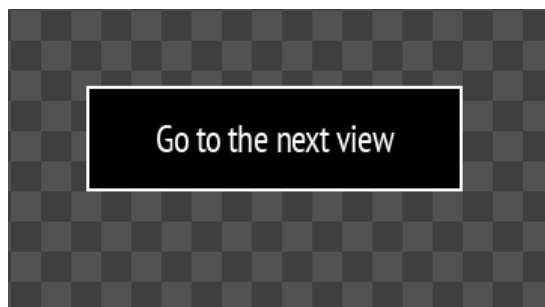


Figure 12.7. ボタン付きのFirstView



### 2番目のビューステートの作成

2番目のビューにはボタンが含まれています。このビューを作成するには、すでに作成した要素をコピーして名前を変更します。

Prerequisite:

- 前のセクションの手順を完了していること。

#### Step 1

メインタブを選択します。

#### Step 2

FirstStateステートをコピーして貼り付けます。

#### Step 3

ナビゲーションコンポーネントで、作成した新しいステートを探し、次のウィジェットの名前を変更します。

- ▶ ステートの名前をSecondStateに変更します。
- ▶ ビューの名前をSecondViewに変更します。
- ▶ RectNextViewという名前をRectGoBackに変更します。
- ▶ LabelNextViewという名前をLabelGoBackに変更します。

#### Step 4

LabelGoBackをダブルクリックし、textテキストボックスにGo backと入力します。



Figure 12.8. ボタン付きのSecondView



### 遷移とイベントの作成

Prerequisite:

- 前のセクションの手順を完了していること。

#### Step 1

メインステートマシンをダブルクリックします。

#### Step 2

メインステートマシンの端から双方のビューステートへの遷移を作成します。

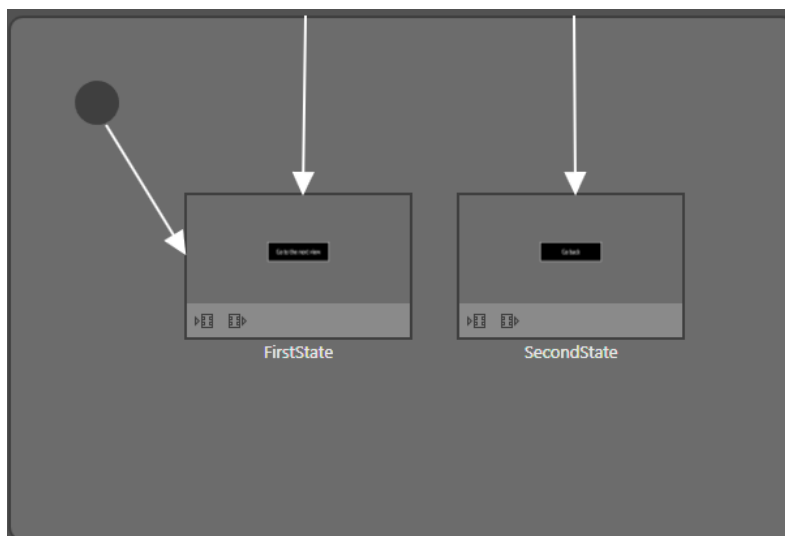


Figure 12.9. 遷移付きのMainステートマシン

### Step 3

メインステートマシンからFirstStateへの遷移を選択します。

### Step 4

プロパティコンポーネントで、トリガーコンボボックスにgoToFirstStateと入力し、イベントの追加をクリックします。

新しいイベントが作成されます。

### Step 5

SecondStateへの遷移を選択します。

### Step 6

プロパティコンポーネントで、トリガーコンボボックスにgoToSecondStateと入力し、イベントの追加をクリックします。

新しいイベントが作成されます。



## ボタンとイベントの接続

ここで、ボタンのクリック時に別のステートへの遷移がトリガーされるという動作を定義します。そのためには、EB GUIDEスクリプトを使用します。

Prerequisite:

- 前のセクションの手順を完了していること。

### Step 1

ナビゲーションコンポーネントで、LabelNextViewをダブルクリックします。

#### Step 1.1

プロパティコンポーネントでウィジェット機能プロパティカテゴリに移動し、追加/削除をクリックします。ウィジェット機能ダイアログが表示されます。

#### Step 1.2

使用可能なウィジェット機能の下で、入力処理カテゴリを展開し、タッチリリースを選択します。

#### Step 1.3

承認をクリックします。

関連するウィジェット機能のプロパティがプロパティコンポーネントに追加されます。

#### Step 1.4

touchShortReleasedプロパティの横にある値列を選択し、`{}` をクリックします。

EB GUIDEスクリプトエディターが開きます。

#### Step 1.5

ボタンのクリック時にgoToSecondStateイベントを発行する次のEB GUIDEスクリプトを入力します。

```
function(v:touchId::int, v:x::int, v:y::int, v:fingerId::int)
{
    fire ev:goToSecondState()
    false
}
```

#### Step 1.6

承認をクリックします。

### Step 2

ナビゲーションコンポーネントで、LabelGoBackをダブルクリックします。

#### Step 2.1

プロパティコンポーネントでウィジェット機能プロパティカテゴリに移動し、追加/削除をクリックします。ウィジェット機能ダイアログが表示されます。

#### Step 2.2

使用可能なウィジェット機能の下で、入力処理カテゴリを展開し、タッチリリースを選択します。

#### Step 2.3

承認をクリックします。

関連するウィジェット機能のプロパティがプロパティコンポーネントに追加されます。

#### Step 2.4

touchShortReleasedプロパティの横にある値列を選択し、`{}` をクリックします。

EB GUIDEスクリプトエディターが開きます。

#### Step 2.5

ボタンのクリック時にgoToFirstStateイベントを発行する次のEB GUIDEスクリプトを入力します。

```
function(v:touchId::int, v:x::int, v:y::int, v:fingerId::int)
{
    fire ev:goToFirstState()
    false
}
```

#### Step 2.6

承認をクリックします。



### FirstView開始アニメーションの作成

VTAコンポーネントを有効にし、ボタンが右側から入ってくるアニメーションを作成します。アニメーションを作成するには、アニメーション化されるプロパティ、アニメーションの継続時間、その開始と終了の場所を定義する必要があります。このチュートリアルでは、xプロパティのみを使用します。

Prerequisite:

- 前のセクションの手順を完了していること。

#### Step 1

VTAコンポーネントは、デフォルトレイアウトには表示されません。有効にする必要があります。

コマンドエリアでレイアウト>VTAの順にクリックします。

VTAコンポーネントが表示されます。

#### Step 2

ナビゲーションコンポーネントで、FirstStateをダブルクリックします。

#### Step 3

VTAコンポーネントで、+をクリックし、開始アニメーションを選択します。

開始アニメーションの表が表示されます。

アニメーションエディターがコンテンツエリアの下に表示されます。

#### Step 4

+をクリックし、終了アニメーションを選択します。

終了アニメーションの表が表示されます。

#### Step 5

アニメーションエディターのドロップダウンリストで、Entry animation 1を選択します。

#### Step 6

+をクリックし、遷移先:FirstViewを選択します。アニメーションプロパティダイアログが表示されます。

#### Step 7

RectNextView、x、高速開始曲線、および承認の順にクリックします。

新しいアニメーションがアニメーション化プロパティリストに追加されます。

#### Step 8

プロパティコンポーネントで、次のように入力します。

- ▶ startテキストボックスに、900と入力します。
- ▶ endテキストボックスに、290と入力します。

シミュレーションを開始すると、ボタンが移動してくることを確認できます。



### FirstView終了アニメーションの作成

ボタンが右側に出て行くアニメーションを作成します。

Prerequisite:

- 前のセクションの手順を完了していること。

#### Step 1

アニメーションエディターのドロップダウンリストで、Exit animation 1を選択します。

#### Step 2

十をクリックし、遷移元:FirstViewを選択します。アニメーションプロパティダイアログが表示されます。

#### Step 3

RectNextView、x、高速開始曲線、および承認の順にクリックします。

新しいアニメーションがアニメーション化プロパティリストに追加されます。

#### Step 4

プロパティコンポーネントで、次のように入力します。

- ▶ durationテキストボックスに、500と入力します。
- ▶ startテキストボックスに、290と入力します。
- ▶ endテキストボックスに、800と入力します。



### SecondView開始アニメーションの作成

ボタンが右側から入ってくるアニメーションを作成します。

Prerequisite:

- 前のセクションの手順を完了していること。

#### Step 1

ナビゲーションコンポーネントで、SecondViewをダブルクリックします。

SecondView がコンテンツエリアに表示されます。

#### Step 2

VTAタブで、以下をクリックします。+

#### Step 3

開始アニメーションと終了アニメーションを追加します。

アニメーションエディターがコンテンツエリアの下に表示されます。

#### Step 4

アニメーションエディターのドロップダウンリストで、Entry animation 2を選択します。

#### Step 5

十をクリックし、遷移先:SecondViewを選択します。アニメーションプロパティダイアログが表示されます。

#### Step 6

RectGoBack、x、高速開始曲線、および承認の順にクリックします。

新しいアニメーションがアニメーション化プロパティリストに追加されます。

#### Step 7

プロパティコンポーネントで、次のように入力します。

- ▶ startテキストボックスに、900と入力します。
- ▶ endテキストボックスに、290と入力します。



### SecondView終了アニメーションの作成

ボタンが左側に出て行くアニメーションを作成します。

Prerequisite:

- 前のセクションの手順を完了していること。

#### Step 1

アニメーションエディターのドロップダウンリストで、Exit animation 2を選択します。

#### Step 2

十をクリックし、遷移先:SecondViewを選択します。アニメーションプロパティダイアログが表示されます。

#### Step 3

RectGoBack、x、高速開始曲線、および承認の順にクリックします。

新しいアニメーションがアニメーション化プロパティリストに追加されます。

#### Step 4

プロパティコンポーネントで、次のように入力します。

- ▶ durationテキストボックスに、500と入力します。
- ▶ startテキストボックスに、290と入力します。

- ▶ endテキストボックスに、800と入力します。




#### EB GUIDEモデルの保存およびテスト


Prerequisite:

- 前のセクションの手順を完了していること。

#### Step 1

プロジェクトを保存するには、コマンドエリアで をクリックします。

#### Step 2

シミュレーションを開始するには、コマンドエリアで をクリックします。

Go to the next viewをクリックします。ビューが変化し、アニメーションが再生されます。

Go backをクリックします。ビューが最初のビューに再び変化し、アニメーションが再生されます。

## 12.10. チュートリアル: アニメーションでのスクリプト曲線の使用

### NOTE



#### デフォルトのウィンドウレイアウト

すべての手順およびスクリーンショットでは、デフォルトのウィンドウレイアウトが使用されています。手順に従う場合は、EB GUIDE Studioウィンドウをレイアウト > **Reset to default layout**を選択することによって、デフォルトレイアウトにリセットすることをお勧めします。

アニメーションで独自の曲線を定義する場合は、スクリプト曲線を使用します。その他のアニメーション曲線では適さない場合や、カスタム曲線を定義したい場合には、独自の曲線の定義が必要になることがあります。このチュートリアルでは、2つのアニメーション用に2つのスクリプト曲線を持つ簡単なモデルを作成します。以下の要素を作成します。

- ▶ ビューステート
- ▶ 2つの四角形ウィジェット
- ▶ スクリプト曲線によって四角形ウィジェットの位置をアニメーション化する2つのアニメーションウィジェット

結果として、2つの四角形を持つモデルが生成されます。一方の四角形は下に移動します。他方は横に移動します。



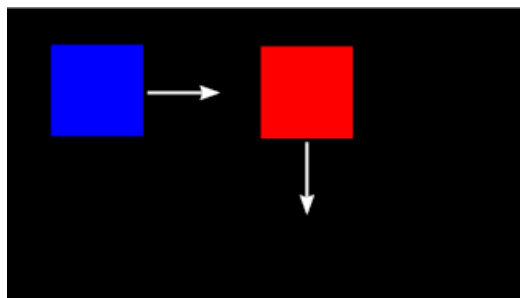


Figure 12.10. 移動方向が指定された四角形

所要時間: 15分



### 最初のスクリプト曲線の作成

Prerequisite:

- コンテンツエリアに、メインステートマシンが表示されていること。
- メインステートマシンに、初期ステートとFirstViewというビューステートが含まれていること。
- 初期ステートに次への遷移があること。FirstView

#### Step 1

ツールボックスコンポーネントからこのビューに四角形をドラッグし、その名前を次のように変更します。

BlueRectangle

#### Step 2

プロパティコンポーネントで、fillColorを青に設定します。

#### Step 3

ツールボックスコンポーネントからこのビューにアニメーションをドラッグし、その名前をMoveAnimationに変更します。

#### Step 4

データプールコンポーネントで、Float型のデータプールアイテムを追加し、その名前をxFloatに変更します。

#### Step 5

ナビゲーションコンポーネントでBlueRectangleを選択します。

#### Step 6

プロパティコンポーネントでユーザー定義プロパティカテゴリに移動し、+をクリックします。

メニューが開きます。

#### Step 7


メニューから条件スクリプトを選択します。

Conditional script 1 がユーザー定義プロパティに追加されます。

#### Step 8

Conditional script 1という名前をStartBlueAnimationに変更します。

#### Step 9

StartBlueAnimationの横にある  をクリックします。

EB GUIDEスクリプトエディターが開きます。

#### Step 10


次のスクリプトを入力します。

```
function(v:arg0::bool)
{
    f:animation_play(v:this->^-"MoveAnimation")
}
```

#### Step 11

ナビゲーションコンポーネントでFirstViewを選択します。

#### Step 12

アニメーションエディターで、アニメーション化プロパティの横にある  をクリックし、FirstViewを選択します。

アニメーションプロパティダイアログが開きます。

#### Step 13

BlueRectangleの下で、xプロパティを選択し、さらに次のものを選択します。Script curve

#### Step 14


承認をクリックします。

Script curve 1 がアニメーションエディターに追加されます。

#### Step 15

Script curve 1という名前をBlueCurveに変更します。

#### Step 16

プロパティコンポーネントで、curveプロパティの横にある  をクリックします。

EB GUIDEスクリプトエディターが開きます。

#### Step 17

次のスクリプトを入力します。

```
function(v:diff::int, v:t_anim::int)
{
    dp:xFloat+=0.2
    f:floor(dp:xFloat*dp:xFloat)
}
```



## 2番目のスクリプト曲線の作成

Prerequisite:

- 前のセクションの手順を完了していること。

### Step 1

ツールボックスコンポーネントから**FirstView**内に四角形をドラッグし、その名前をRedRectangleに変更します。

### Step 2

プロパティコンポーネントで、fillColorプロパティを赤に設定します。

### Step 3

データプールコンポーネントで、Integer型のデータプールアイテムを追加し、その名前を1\_diffに変更します。

### Step 4

Integer型のデータプールアイテムをもう1つ追加し、その名前を2t\_animに変更します。

### Step 5

RedRectangleを選択します。

### Step 6

プロパティコンポーネントでユーザー定義プロパティカテゴリに移動し、+をクリックします。

Conditional script 2 が追加されます。

### Step 7

Conditional script 2という名前をStartRedAnimationに変更します。

### Step 8

StartRedAnimationの横にある{ } をクリックします。

EB GUIDEスクリプトエディターが開きます。

### Step 9

次のスクリプトを入力します。

```
function(v:arg0::bool)
{
    f:animation_play(v:this->^->"MoveAnimation")
}
```

### Step 10

ナビゲーションコンポーネントでFirstViewを選択します。

### Step 11

アニメーションエディターで、アニメーション化プロパティの横にある+をクリックし、FirstViewを選択します。

アニメーションプロパティダイアログが開きます。

### Step 12

RedRectangleの下で、yプロパティ、Script curveの順に選択します。

#### Step 13

承認をクリックします。

Script curve 2 がアニメーションエディターに追加されます。

#### Step 14

Script curve 2という名前をRedCurveに変更します。

#### Step 15

プロパティコンポーネントで、curveプロパティの横にある  をクリックします。

EB GUIDEスクリプトエディターが開きます。

#### Step 16

次のスクリプトを入力します。

```
function(v:diff::int, v:t_anim::int)
{
  dp:"1_diff"=v:diff
  dp:"2t_anim"=v:t_anim
  v:t_anim/2::int
}
```




### EB GUIDEモデルの保存およびテスト


Prerequisite:

- 前のセクションの手順を完了していること。

#### Step 1

プロジェクトを保存するには、コマンドエリアで  をクリックします。

#### Step 2

シミュレーションを開始するには、コマンドエリアで  をクリックします。

シミュレーションの開始時にアニメーションが再生されます。

## 13. リファレンス

この章では、パラメータ、プロパティ、識別子などのリストと表を示します。

### 13.1. Androidイベント

AndroidイベントはSystemNotificationsイベントグループに属し、イベントグループIDは13です。

表13.1 Androidイベント

イベントID	名前	説明
1	RendererEnabled	Androidライフサイクル管理がレンダラーを停止または起動するとき、アプリケーションによって送信されます  パラメータ: <ul style="list-style-type: none"><li>▶ enabled: trueである場合、レンダラーは有効になります。falseである場合、レンダラーはスリープモードに設定されます。</li></ul>
2	setKeyboardVisibility	仮想キーボードが表示されるように設計されている場合、EB GUIDEモデルによって送信されます  パラメータ: <ul style="list-style-type: none"><li>▶ visibility: trueである場合、仮想キーボードが表示されるようになります。falseである場合、表示されません。</li></ul>
3	onKeyboardVisibilityChanged	仮想キーボードが表示されている場合、アプリケーションによって送信されます  パラメータ: <ul style="list-style-type: none"><li>▶ visibility: trueである場合、仮想キーボードが表示されます。falseである場合、表示されません。</li></ul>
4	onLayoutChanged	画面の表示領域が変更されたとき、アプリケーションによって送信されます

イベントID	名前	説明
		パラメータ(ピクセル単位): <ul style="list-style-type: none"><li>▶ x: 表示画面領域の左上隅のX座標</li><li>▶ y: 表示画面領域の左上隅のY座標</li><li>▶ width: 表示画面領域の幅</li><li>▶ height: 表示画面領域の高さ</li></ul>

## 13.2. データプールアイテム

表13.2 データプールアイテムのプロパティ

プロパティ名	説明
Value	データプールアイテムの初期値

## 13.3. データタイプ

このセクションでは、EB GUIDEのデータタイプについて説明します。以下に示すタイプのユーザー定義プロパティおよびデータプールアイテムを追加できます。

### 13.3.1. ブール値

ブール値プロパティは値trueおよびfalseを持つことができます。

使用可能な演算子は次のとおりです。

- ▶ 等しい(==)
- ▶ 等しくない(!=)
- ▶ 否定(!)
- ▶ AND (&&)
- ▶ OR (||)
- ▶ 代入(書き込み可能なプロパティ)(=)

ブール値プロパティをリストに格納できます。リストの詳細については、[13.3.12「リスト」](#)をご覧ください。

## 13.3.2. 色

色はRGBA8888フォーマットで格納されます。

使用例: 透過性がない赤色は(255, 0, 0, 255)です。

使用可能な演算子は次のとおりです。

- ▶ 等しい(==)
- ▶ 等しくない(!=)
- ▶ 代入(書き込み可能なプロパティ)(=)

色プロパティをリストに格納できます。リストの詳細については、[13.3.12「リスト」](#)をご覧ください。

## 13.3.3. 条件スクリプト

条件スクリプトは初期化時およびトリガー時の反応に使用されます。条件スクリプトを編集するとき、コンテンツエリアは次のセクションに分割されます。

- ▶ **Trigger**セクションには、**On trigger**スクリプトの実行をトリガーするイベント、データプールアイテム、およびウィジェットプロパティを追加できます。
- ▶ **On trigger**セクションには、初期化時、イベントトリガー時、またはデータプールアイテムやウィジェットプロパティの値を更新した後に呼び出されるEB GUIDEスクリプトを追加できます。

**On trigger** EB GUIDEスクリプトのパラメータは、スクリプトを実行する要因を表します。

`arg0`は、EB GUIDEスクリプトが初期化時やトリガーによって実行されるかどうかを参照します。注意事項を以下に示します。

- ▶ 初期化時にEB GUIDEスクリプトが実行される場合、`arg0`はtrueになります。
- ▶ トリガーによってEB GUIDEスクリプトが実行される場合、`arg0`はfalseになります。

**On trigger** EB GUIDEスクリプトの戻り値によって、プロパティの変更通知が次のように制御されます。

**On trigger** EB GUIDEスクリプトの戻り値によって、EB GUIDEスクリプトで通知を生成する必要があるかどうかを制御します。注意事項を以下に示します。

- ▶ 戻り値がtrueの場合、通知が生成されます。
- ▶ 戻り値がfalseの場合、通知は生成されません。

On triggerスクリプトを実行可能にするには、以下の場合に条件を満たす必要があります。

- ▶ 初期化時。例えば、データプールアイテムの場合はEB GUIDEモデル起動時、ウィジェットプロパティの場合はビュー作成時。
- ▶ トリガースクリプトからイベントを処理するとき。イベントが一致するたびにEB GUIDEスクリプトが実行されます。
- ▶ トリガースクリプトから1つ以上のアイテムのデータプール通知を処理するとき。複数の通知を一度に処理できません。
- ▶ トリガースクリプトから1つ以上のウィジェットプロパティの通知を処理するとき。複数の通知を一度に処理できません。

### 13.3.4. 浮動小数点数

浮動小数点数データタイプは単精度32ビットIEEE 754値を表します。

使用可能な演算子は次のとおりです。

- ▶ 等しい(==)
- ▶ 等しくない(!=)
- ▶ より大きい(>)
- ▶ 以上(>=)
- ▶ 未満(<)
- ▶ 以下(<=)
- ▶ 加算(+)
- ▶ 減算(-)
- ▶ 乗算(\*)
- ▶ 除算(/)
- ▶ 代入(書き込み可能なプロパティ)(=)

浮動小数点数プロパティをリストに格納できます。リストの詳細については、[13.3.12「リスト」](#)をご覧ください。

### 13.3.5. フォント

EB GUIDEプロジェクトにフォントを追加するには、フォントファイルを次のディレクトリにコピーします。 `$GUIDE_PROJECT_PATH/<project name>/resources`

使用可能な演算子は次のとおりです。



- ▶ 代入(書き込み可能なプロパティ)(=)

フォントプロパティをリストに格納できます。リストの詳細については、[13.3.12「リスト」](#)をご覧ください。

### 13.3.6. Function () : bool

Function () : boolを使って、独自の関数を作成できます。

このデータタイプに使用できる操作は、すべてのプロパティの読み取り/実行操作です。

### 13.3.7. IBL

IBLは、IBLGeneratorによって生成されたライティング情報を格納するデータフォーマットです。

EB GUIDEプロジェクトにIBLを追加するには、.ebiblファイルを次のディレクトリにコピーします。 `$GUIDE_PROJECT_PATH/<project name>/resources`

使用可能な演算子は次のとおりです。

- ▶ 代入(書き込み可能なプロパティ)(=)

IBLプロパティをリストに格納できます。リストの詳細については、[13.3.12「リスト」](#)をご覧ください。

### 13.3.8. イメージ

EB GUIDEプロジェクトにイメージを追加するには、イメージファイルを次のディレクトリにコピーします。 `$GUIDE_PROJECT_PATH/<project name>/resources`

使用可能な演算子は次のとおりです。

- ▶ 代入(書き込み可能なプロパティ)(=)

イメージプロパティをリストに格納できます。リストの詳細については、[13.3.12「リスト」](#)をご覧ください。

### 13.3.9. 整数

EB GUIDEは符号付き32ビット整数をサポートしています。

使用可能な演算子は次のとおりです。

- ▶ 等しい(==)
- ▶ 等しくない(!=)
- ▶ より大きい(>)
- ▶ 以上(>=)
- ▶ 未満(<)
- ▶ 以下(<=)
- ▶ 加算(+)
- ▶ 減算(-)
- ▶ 乗算(\*)
- ▶ 除算(/)
- ▶ 剰余(%)
- ▶ 代入(書き込み可能なプロパティ)(=)

整数プロパティをリストに格納できます。リストの詳細については、[13.3.12「リスト」](#)をご覧ください。

## 13.3.10. メッシュ

メッシュは3Dオブジェクトの形状を定義します。

EB GUIDEプロジェクトにメッシュを追加するには、`.ebmesh`ファイルを次のディレクトリにコピーします。`$GUIDE_PROJECT_PATH/<project name>/resources`

使用可能な演算子は次のとおりです。

- ▶ 代入(書き込み可能なプロパティ)(=)

メッシュプロパティをリストに格納できます。リストの詳細については、[13.3.12「リスト」](#)をご覧ください。

## 13.3.11. 文字列

EB GUIDEは、`Hello world`などの文字列をサポートしています。

使用可能な演算子は次のとおりです。

- ▶ 等しい(大文字と小文字を区別) (==)
- ▶ 等しくない(大文字と小文字を区別) (!=)
- ▶ 等しい(大文字と小文字を区別、ASCII範囲内のみ) (=Aa=)
- ▶ より大きい(>)
- ▶ 以上(>=)
- ▶ 未満(<)
- ▶ 以下(<=)
- ▶ 連結(+)
- ▶ 代入(書き込み可能なプロパティ) (=)

文字列プロパティをリストに格納できます。リストの詳細については、[13.3.12「リスト」](#)をご覧ください。

## 13.3.12. リスト

EB GUIDEは同じデータタイプを持つ値のリストをサポートしています。

以下のリストタイプがあります。

- ▶ ブール値リスト
- ▶ 色リスト
- ▶ 浮動小数点数リスト
- ▶ フォントリスト
- ▶ IBLリスト
- ▶ イメージリスト
- ▶ 整数リスト
- ▶ メッシュリスト
- ▶ 文字列リスト

以下のタイプはリストで使用できません。

- ▶ リスト
- ▶ プロパティの参照
- ▶ リスト要素の参照

使用可能な演算子は次のとおりです。

- ▶ 長さ: (長さ)

- ▶ 要素アクセサ: ([])

## 13.4. EB GUIDEスクリプト

### 13.4.1. EB GUIDEスクリプトのキーワード

以下はEB GUIDEスクリプトの予約されているキーワードの一覧です。スクリプト内でこれらの単語を識別子として使用する場合は、単語を引用符で囲む必要があります。

キーワード	説明
cancel_fire	fire_delayedによって発行されるイベントをキャンセルします。
color:	{0,255,255}などの色パラメータが後に続きます。
dp:	データプールアイテムが後に続きます。
l:	言語が後に続きます。f:languageで使用します。
else	if条件部が完了しています。代わりに後続のブロックが実行されます。
ev:	イベントが後に続きます。
f:	ユーザー定義関数が後に続きます。
false	ブールリテラル値
fire	イベントを発行します
fire_delayed	指定の時間後にイベントが発行されます。時間はミリ秒単位で指定します。
if	ブール式をテストする文が後に続きます。式がtrueである場合、文が実行されます。
in	ローカル変数宣言と変数の使用スコープの間のセパレータです match_eventおよびletとともに使用されます。
function	関数を宣言します
length	プロパティの長さ
let	スコープ内でアクセス可能なローカル変数を宣言します
list	整数リストなどのリストタイプを宣言します
match_event	現在のイベントが想定されるイベントに対応しているかどうかを確認し、などの変数を宣言します let
popup_stack	動的ステートマシンの優先順位を定義する動的ステートマシンリスト
s:	スキンが後に続きます。f:skinで使用します。
sm:	ステートマシンが後に続きます

キーワード	説明
true	ブールリテラル値
unit	voidタイプの値
v:	ローカル変数が後に続きます。
while	条件がtrueである限り、文を繰り返します

## 13.4.2. EB GUIDEスクリプト 演算子の優先順位

以下は、EB GUIDEスクリプトの演算子とそれらの優先順位および結合性の一覧です。演算子は上から下へ優先順位が高いものから低いものの順に示されています。

表 13.3 EB GUIDEスクリプト 演算子の優先順位

演算子	結合性
(()), ({}), ([])	なし
([])	なし
(->)	左
(.)	なし
(::)	左
長さ	なし
(&)	右
(!), (-) 単項マイナス	右
(*), (/), (%)	左
(+), (-)	左
(<), (>), (<=), (>=)	左
(!=), (==), (=Aa=)	左
(&&)	左
(  )	左
(=), (+=), (-=), (=>)	右
(,)	右
(;)	左

## 13.4.3. EB GUIDEスクリプト標準ライブラリ

この章では、EB GUIDEスクリプトのすべての関数の説明を示します。

### 13.4.3.1. EB GUIDEスクリプトの関数A～B

#### 13.4.3.1.1. abs

この関数は整数 $x$ の絶対値を返します。

表13.4 のパラメータ abs

パラメータ	タイプ	説明
$x$	整数	絶対値を返す数
<return>	整数	戻り値

#### 13.4.3.1.2. absf

この関数は浮動小数点数 $x$ の絶対値を返します。

表13.5 のパラメータ absf

パラメータ	タイプ	説明
$x$	浮動小数点数	絶対値を返す数
<return>	浮動小数点数	戻り値

#### 13.4.3.1.3. acosf

この関数は $x$ の逆余弦の主値を返します。

表13.6 のパラメータ acosf

パラメータ	タイプ	説明
$x$	浮動小数点数	逆余弦を返す数
<return>	浮動小数点数	戻り値

#### 13.4.3.1.4. animation\_before

この関数は、実行中のアニメーションが時間軸上の指定された点を過ぎたかどうかを確認します。

表13.7 のパラメータ animation\_before

パラメータ	タイプ	説明
animation	GtfTypeRecord	操作するアニメーション

パラメータ	タイプ	説明
time	整数	時間軸上の点
<return>	ブール値	trueである場合、アニメーションはまだ時間軸上の点を過ぎていません。

#### 13.4.3.1.5. animation\_beyond

この関数は、実行中のアニメーションが時間軸上の指定された点を過ぎたかどうかを確認します。

表13.8 のパラメータ animation\_beyond

パラメータ	タイプ	説明
animation	GtfTypeRecord	操作するアニメーション
time	整数	時間軸上の点
<return>	ブール値	trueである場合、アニメーションは時間軸上の点を過ぎています。

#### 13.4.3.1.6. animation\_cancel

この関数はアニメーションをキャンセルし、編集したプロパティを現在のステートのままにします。

表13.9 のパラメータ animation\_cancel

パラメータ	タイプ	説明
animation	GtfTypeRecord	操作するアニメーション
<return>	ブール値	trueである場合、関数が成功しました。

#### 13.4.3.1.7. animation\_cancel\_end

この関数はアニメーションをキャンセルし、可能であれば、編集したプロパティを終了ステートに設定します。

表13.10 のパラメータ animation\_cancel\_end

パラメータ	タイプ	説明
animation	GtfTypeRecord	操作するアニメーション
<return>	ブール値	trueである場合、関数が成功しました。

#### 13.4.3.1.8. animation\_cancel\_reset

この関数はアニメーションをキャンセルし、可能であれば、編集したプロパティを初期ステートにリセットします。

表13.11 のパラメータ animation\_cancel\_reset

パラメータ	タイプ	説明
animation	GtfTypeRecord	操作するアニメーション
<return>	ブール値	trueである場合、関数が成功しました。

#### 13.4.3.1.9. animation\_pause

この関数はアニメーションを一時停止します。

表13.12 のパラメータ animation\_pause

パラメータ	タイプ	説明
animation	GtfTypeRecord	操作するアニメーション
<return>	ブール値	trueである場合、関数が成功しました。

#### 13.4.3.1.10. animation\_play

この関数はアニメーションを開始または続行します。

表13.13 のパラメータ animation\_play

パラメータ	タイプ	説明
animation	GtfTypeRecord	操作するアニメーション
<return>	ブール値	trueである場合、アニメーションはまだ実行していません。

#### 13.4.3.1.11. animation\_reverse

この関数はアニメーションを逆方向に再生します。

表13.14 のパラメータ animation\_reverse

パラメータ	タイプ	説明
animation	GtfTypeRecord	操作するアニメーション
<return>	ブール値	trueである場合、アニメーションはまだ実行していません。

#### 13.4.3.1.12. animation\_running

この関数はアニメーションが現在実行中かどうかを確認します。



表13.15 のパラメータ animation\_running

パラメータ	タイプ	説明
animation	GtfTypeRecord	操作するアニメーション
<return>	ブール値	trueである場合、アニメーションは実行中です。

#### 13.4.3.1.13. animation\_set\_time

この関数はアニメーションの現在の時間を設定します。アニメーションをスキップまたは再生するために使用できません。

表13.16 のパラメータ animation\_set\_time

パラメータ	タイプ	説明
animation	GtfTypeRecord	操作するアニメーション
time	整数	時間
<return>	ブール値	trueである場合、関数が成功しました。

#### 13.4.3.1.14. asinf

この関数は $x$ の逆正弦の主値を計算します。

表13.17 のパラメータ asinf

パラメータ	タイプ	説明
$x$	浮動小数点数	逆正弦を返す数
<return>	浮動小数点数	戻り値

#### 13.4.3.1.15. atan2f

この関数は、2つの引数の符号を使用して結果の象限を決定し、 $y/x$ の逆正接の主値を計算します。

表13.18 のパラメータ atan2f

パラメータ	タイプ	説明
$y$	浮動小数点数	引数 $y$
$x$	浮動小数点数	引数 $x$
<return>	浮動小数点数	戻り値

#### 13.4.3.1.16. atan2i

この関数は、2つの引数の符号を使用して結果の象限を決定し、 $y/x$ の逆正接の主値を計算します。

表13.19 のパラメータ atan2i

パラメータ	タイプ	説明
y	整数	引数 y
x	整数	引数 x
<return>	浮動小数点数	戻り値

#### 13.4.3.1.17. atanf

この関数はxの逆正接の主値を計算します。

表13.20 のパラメータ atanf

パラメータ	タイプ	説明
x	浮動小数点数	逆正接を返す数
<return>	浮動小数点数	戻り値

#### 13.4.3.1.18. bool2string

この関数は、ブール変数を文字列trueまたはfalseのいずれかに変換します。

表13.21 のパラメータ bool2string

パラメータ	タイプ	説明
x	ブール	文字列に変換する値
<return>	文字列	true xがtrueの場合、その他はfalse

### 13.4.3.2. EB GUIDEスクリプトの関数C~H

#### 13.4.3.2.1. ceil

この関数は引数以上の最小整数値を返します。

表13.22 のパラメータ `ceil`

パラメータ	タイプ	説明
<code>value</code>	浮動小数点数	丸める値
<code>&lt;return&gt;</code>	整数	丸めた値

#### 13.4.3.2.2. `changeDynamicStateMachinePriority`

この関数は動的ステートマシンの優先順位を変更します。

表13.23 のパラメータ `changeDynamicStateMachinePriority`

パラメータ	タイプ	説明
<code>state</code>		動的ステートマシンリストを持つステート
<code>sm</code>	整数	動的ステートマシン
<code>priority</code>	整数	リスト内の動的ステートマシンの優先順位

#### 13.4.3.2.3. `character2unicode`

この関数は文字列内の最初の文字のUnicode値を返します。

表13.24 のパラメータ `character2unicode`

パラメータ	タイプ	説明
<code>str</code>	文字列	入力文字列
<code>&lt;return&gt;</code>	整数	Unicode値としての文字 エラーの場合は0

#### 13.4.3.2.4. `clampf`

この関数は、浮動小数点値を定義済みの範囲 `[xmin, xmax]` にクランプします。つまり、この関数は `max (xmin, min (xmax, x))` を計算します。

表13.25 のパラメータ `clampf`

パラメータ	タイプ	説明
<code>x</code>	浮動小数点数	クランプする値

パラメータ	タイプ	説明
xmin	浮動小数点数	最小範囲
xmax	浮動小数点数	最大範囲
<return>	浮動小数点数	[xmin, xmax] の範囲にクランプされた値x

#### 13.4.3.2.5. clampi

この関数は、整数値を定義済みの範囲 [xmin, xmax] にクランプします。つまり、この関数は  $\max(\text{xmin}, \min(\text{xmax}, x))$  を計算します。

表13.26 のパラメータ clampi

パラメータ	タイプ	説明
x	整数	クランプする値
xmin	整数	最小範囲
xmax	整数	最大範囲
<return>	整数	[xmin, xmax] の範囲にクランプされた値x

#### 13.4.3.2.6. clearAllDynamicStateMachines

この関数は動的ステートマシンリストから動的ステートマシンをすべて削除します。

表13.27 のパラメータ clearAllDynamicStateMachines

パラメータ	タイプ	説明
state		動的ステートマシンリストを持つステート

#### 13.4.3.2.7. color2string

この関数は色を8桁の16進数値に変換します。

表13.28 のパラメータ color2string

パラメータ	タイプ	説明
value	色	文字列に変換する色
<return>	文字列	プレフィックスとして#が付いた16進数の文字列としてフォーマットされた色

注記



フォーマットの例

返される文字列のフォーマットは#RRGGBBAAで、赤、緑、青、アルファの各色チャンネルを表す2桁の数が含まれます。

例えば、不透明の純赤色は#ff0000ffに変換され、半透明の純緑色は#00ff007fに変換されます。

### 13.4.3.2.8. cosf

この関数はxの余弦を返します。このxはラジアン単位で指定します。

表13.29 のパラメータ cosf

パラメータ	タイプ	説明
x	浮動小数点数	余弦を返す数
<return>	浮動小数点数	戻り値

### 13.4.3.2.9. deg2rad

この関数は角度を度数からラジアンに変換します。

表13.30 のパラメータ deg2rad

パラメータ	タイプ	説明
x	浮動小数点数	度数からラジアンに変換する角度
<return>	浮動小数点数	戻り値

### 13.4.3.2.10. expf

この関数はe(自然対数の底)のx乗の値を返します。

表13.31 のパラメータ expf

パラメータ	タイプ	説明
x	浮動小数点数	指数
<return>	浮動小数点数	戻り値

### 13.4.3.2.11. float2string

この関数は単純浮動小数点数を文字列に変換します。

表13.32 のパラメータ float2string

パラメータ	タイプ	説明
value	浮動小数点数	文字列に変換する値
<return>	文字列	文字列としてフォーマットされた浮動小数点値

#### 13.4.3.2.12. floor

この関数はパラメータ値以下の最大整数値を返します。

表13.33 のパラメータ floor

パラメータ	タイプ	説明
value	浮動小数点数	丸める値
<return>	整数	丸めた値

#### 13.4.3.2.13. fmod

この関数は、浮動小数点除算 $x/y$ の剰余を計算します。

表13.34 のパラメータ fmod

パラメータ	タイプ	説明
x	浮動小数点数	浮動小数点分子
y	浮動小数点数	浮動小数点分母
<return>	浮動小数点数	除算の剰余 $x/y$

#### 13.4.3.2.14. focusMoveTo

この関数は、フォーカスマネージャーのフォーカスを専用のフォーカス可能な要素へ強制的に進めます。

表13.35 のパラメータ focusMoveTo

パラメータ	タイプ	説明
widget	ウィジェット	フォーカスを移動するウィジェット
<return>	void	

#### 13.4.3.2.15. focusNext

この関数はフォーカスマネージャーのフォーカスを次のフォーカス可能な要素へ強制的に進めます。

表13.36 のパラメータ `focusNext`

パラメータ	タイプ	説明
<return>	void	

#### 13.4.3.2.16. `focusPrevious`

この関数はフォーカスマネージャーのフォーカスを前のフォーカス可能な要素へ強制的に戻します。

表13.37 のパラメータ `focusPrevious`

パラメータ	タイプ	説明
<return>	void	

#### 13.4.3.2.17. `format_float`

この関数は浮動小数点値をフォーマットします。

表13.38 のパラメータ `format_float`

パラメータ	タイプ	説明
<code>format</code>	文字列	次の構造の文字列:  <code>%[フラグ] [幅] [.精度]タイプ</code> <ul style="list-style-type: none"> <li>▶ フラグ: 出力配置、および符号、空白、先行ゼロ、小数点、8進数と16進数のプレフィックスの出力を制御するオプションの1文字または複数文字。</li> <li>▶ 幅: 出力される最小文字数を指定するオプションの10進数。</li> <li>▶ 精度: 小数点文字の後の有効桁数または桁数を指定するオプションの10進数。</li> <li>▶ タイプ: 関連引数が文字、文字列、整数、または浮動小数点数として解釈されるかどうかを判別する必須の変換指定子文字。</li> </ul>
<code>useDotAsDelimiter</code>	ブール値	区切り記号を定義します。  使用可能な値: <ul style="list-style-type: none"> <li>▶ <code>true</code>: ドットを区切り記号として使用します。</li> <li>▶ <code>false</code>: カンマを区切り記号として使用します。</li> </ul>
<code>value</code>	浮動小数点数	フォーマットする数

警告



**C++のprintf仕様の順守**

formatパラメータは、C++のprintf仕様に従って定義されます。

この仕様に準拠しない値を使用すると、予期しない動作発生する可能性があります。

例えば、format\_floatに許可されるタイプは、f、a、g、およびeであり、1文字のタイプしか許可されません。

### 13.4.3.2.18. format\_int

この関数は整数値をフォーマットします。

表13.39 のパラメータ format\_int

パラメータ	タイプ	説明
format	文字列	次の構造の文字列: %[フラグ] [幅] [.精度]タイプ <ul style="list-style-type: none"><li>▶ フラグ: 出力配置、および符号、空白、先行ゼロ、小数点、8進数と16進数のプレフィックスの出力を制御するオプションの1文字または複数文字。</li><li>▶ 幅: 出力される最小文字数を指定するオプションの10進数。</li><li>▶ 精度: 出力される最小桁数を指定するオプションの10進数。</li><li>▶ タイプ: 関連引数が文字、文字列、整数、または浮動小数点数として解釈されるかどうかを判別する必須の変換指定子文字。</li></ul>
value	整数	フォーマットする数

警告



**C++のprintf仕様の順守**

formatパラメータは、C++のprintf仕様に従って定義されます。

この仕様に準拠しない値を使用すると、予期しない動作発生する可能性があります。

例えば、format\_intに許可されるタイプは、d、i、o、x、およびuであり、1文字のタイプしか許可されません。

### 13.4.3.2.19. frac

この関数は、浮動小数点値の端数部分を計算します。戻り値は、[0, 1]の範囲になります。例えば、この関数はパラメータ値がx=1.5またはx=-1.5の場合、0.5を返します。



表13.40 のパラメータ frac

パラメータ	タイプ	説明
x	浮動小数点数	浮動小数点値
<return>	浮動小数点数	浮動小数点値の端数部分

#### 13.4.3.2.20. getConfigItem

この関数は、設定アイテム値でデータプールアイテムを埋めます。

表13.41 のパラメータ getConfigItem

パラメータ	タイプ	説明
itemId	dp_id	設定アイテムが格納されているデータプールID
name	文字列	設定アイテム名
<return>	ブール値	データプールアイテムが設定アイテム値で正常に埋められている場合はtrueになります。

#### 13.4.3.2.21. getFontAscender

この関数はパラメータとして渡されたフォントのアセンダーを返します。

表13.42 のパラメータ getFontAscender

パラメータ	タイプ	説明
x	フォント	評価されるフォント  マルチフォントサポートを追加している場合、デフォルトフォントのみが評価されることに注意してください。
<return>	整数	フォントのアセンダー

#### 13.4.3.2.22. getFontDescender

この関数はパラメータとして渡されたフォントのディセンダーを返します。

表13.43 のパラメータ getFontDescender

パラメータ	タイプ	説明
x	フォント	評価されるフォント  マルチフォントサポートを追加している場合、デフォルトフォントのみが評価されることに注意してください。

パラメータ	タイプ	説明
<return>	整数	フォントのディセンダー

#### 13.4.3.2.23. getFontLineGap

この関数はパラメータとして渡されたフォントの行間を返します。

表13.44 のパラメータ getFontLineGap

パラメータ	タイプ	説明
x	フォント	評価されるフォント  マルチフォントサポートを追加している場合、デフォルトフォントのみが評価されることに注意してください。
<return>	整数	フォントの行間

#### 13.4.3.2.24. getImageHeight

この関数は、パラメータとして渡されたイメージの高さをピクセル単位で返します。

表13.45 のパラメータ getImageHeight

パラメータ	タイプ	説明
x	イメージウィジェット	評価するウィジェット
<return>	整数	ピクセル単位のイメージの高さ

#### 13.4.3.2.25. getImageWidth

この関数は、パラメータとして渡されたイメージの幅をピクセル単位で返します。

表13.46 のパラメータ getImageWidth

パラメータ	タイプ	説明
x	イメージウィジェット	評価されるウィジェット
<return>	整数	ピクセル単位のイメージの幅

#### 13.4.3.2.26. getLabelTextHeight

この関数はラベルテキストの全高をピクセル単位で返します。全高は次の式で計算されます。

$total\_height = line\_height * line\_count + line\_gap * (line\_count - 1)$

line\_gapは、フォントの行間と[複数行]ウィジェット機能のlineGapプロパティの合計として計算されます。フォントの行間とlineGapプロパティの両方とも負にすることができます。

表13.47 のパラメータ getLabelTextHeight

パラメータ	タイプ	説明
widget	ラベルウィジェット	評価されるウィジェット
<return>	整数	ピクセル単位のテキストの高さ

#### 13.4.3.2.27. getLabelTextWidth

この関数はラベルテキストの最長行の幅をピクセル単位で返します。

表13.48 のパラメータ getLabelTextWidth

パラメータ	タイプ	説明
widget	ラベルウィジェット	評価するウィジェット
<return>	整数	ピクセル単位のテキストの最長行の幅

#### 13.4.3.2.28. getLineCount

この関数はラベルテキストの行数を返します。

表13.49 のパラメータ getLineCount

パラメータ	タイプ	説明
widget	ラベルウィジェット	評価されるウィジェット
<return>	整数	テキストの行数

#### 13.4.3.2.29. getLineHeight

この関数は、パラメータとして渡されたフォントで書かれた行の高さを返します。

表13.50 のパラメータ getLineHeight

パラメータ	タイプ	説明
x	フォント	<p>評価されるフォント</p> <p>マルチフォントサポートを追加している場合、デフォルトフォントのみが評価されることに注意してください。</p>

パラメータ	タイプ	説明
<return>	整数	指定のフォントで書かれた行の高さ

#### 13.4.3.2.30. getProductString

この関数はEB GUIDE GTFの製品名を含む文字列を返します。

表13.51 のパラメータ getProductString

パラメータ	タイプ	説明
<return>	文字列	製品名

#### 13.4.3.2.31. getTextHeight

この関数はテキストのフォントリソースに関する高さを返します。高さはフォントのアセンダーとディセンダーの合計を表します。

表13.52 のパラメータ getTextHeight

パラメータ	タイプ	説明
text	文字列	評価するテキスト
font	フォント	評価するフォント
<return>	整数	テキストの高さ  フォントのサイズが 0 または負の値の場合、この関数は 0 を返します。

#### 注記



#### getTextHeight

この関数は常にテキストを1行と想定して高さの値を計算します。

#### 13.4.3.2.32. getTextLength

この関数はテキスト内の文字数を返します。

表13.53 のパラメータ getTextLength

パラメータ	タイプ	説明
text	文字列	評価するテキスト

パラメータ	タイプ	説明
<return>	整数	テキスト内の文字数

**注記**



**エスケープシーケンス**

EB GUIDEスクリプトは、\nなどのエスケープシーケンスを解決せず、すべての文字をカウントします。例えば、Label\nというテキストの場合、getTextLength関数は7を返します。

### 13.4.3.2.33. getTextWidth

この関数はテキストのフォントリソースに関する幅を返します。

表13.54 のパラメータ getTextWidth

パラメータ	タイプ	説明
text	文字列	評価するテキスト
font	フォント	評価するフォント
<return>	整数	テキストの幅  フォントのサイズが 0 または負の値の場合、この関数は 0 を返します。

**注記**



この関数は常にテキストを1行と想定して幅の値を計算します。

### 13.4.3.2.34. getVersionString

この関数はEB GUIDE GTFのバージョン番号を含む文字列を返します。

表13.55 のパラメータ getVersionString

パラメータ	タイプ	説明
<return>	文字列	バージョン文字列

### 13.4.3.2.35. has\_list\_window

この関数はタイプリストのデータプールアイテムでインデックスが有効かどうかを確認します。ウィンドウ表示リストでは、インデックスが少なくとも1つのウィンドウ内にあるかどうかを確認します。

表13.56 のパラメータ has\_list\_window

パラメータ	タイプ	説明
itemId	dp_id	タイプリストのデータプールアイテムのID
index	整数	データプールアイテム内のインデックス
<return>	ブール値	trueである場合、データプールアイテム内のインデックスは有効であり、少なくとも1つのウィンドウ内にあります。

### 13.4.3.2.36. hsba2color

この関数はHSB/HSV色をEB GUIDE GTF色に変換します。

表13.57 のパラメータ hsba2color

パラメータ	タイプ	説明
hue	整数	0~360の色の値(度単位)
saturation	整数	彩度(パーセント単位)
brightness	整数	明度(パーセント単位)
alpha	整数	0 (完全透明)~255 (不透明)の間のアルファ値
<return>	色	アルファ値が適用された結果のEB GUIDE GTF色

### 13.4.3.3. EB GUIDEスクリプトの関数I~R

#### 13.4.3.3.1. int2float

この関数は浮動小数点値に変換された整数値を返します。

表13.58 のパラメータ int2float

パラメータ	タイプ	説明
value	整数	浮動小数点数に変換する値
<return>	浮動小数点数	浮動小数点数に変換された整数値

#### 13.4.3.3.2. int2string

この関数は単純整数を文字列に変換します。

表13.59 のパラメータ int2string

パラメータ	タイプ	説明
value	整数	文字列に変換する値
<return>	文字列	文字列に変換された整数値(10進表記)

#### 13.4.3.3.3. isDynamicStateMachineActive

この関数は動的ステートマシンリストを持つステートがアクティブかどうかを確認します。

表13.60 のパラメータ isDynamicStateMachineActive

パラメータ	タイプ	説明
state		動的ステートマシンリストを持つステート
sm	整数	動的ステートマシン

#### 13.4.3.3.4. isWidgetOnActiveStatemachine

この関数は、ウィジェットがアクティブなステートマシンに属しているかどうかを確認します。

表13.61 のパラメータ isWidgetOnActiveStatemachine

パラメータ	タイプ	説明
widget	ウィジェット	評価されるウィジェット
<return>	ブール値	ウィジェットがアクティブなステートマシンに属している場合はtrueになります。

#### 13.4.3.3.5. language

この関数はすべてのデータプールアイテムの言語を切り替えます。この動作は非同期で実行されます。

表13.62 のパラメータ language

パラメータ	タイプ	説明
language	languageType	切り替える言語(など) f: language (l: German)
<return>	void	

#### 13.4.3.3.6. lerp

この関数は、次の式を使用して2つの値xおよびyのリニア補間を計算します。  $(1-s) * x + s * y$

表13.63 のパラメータ lerp

パラメータ	タイプ	説明
x	浮動小数点数	最初の値
y	浮動小数点数	2番目の値
s	浮動小数点数	値xおよびyの間をリニア補間する値
<return>	浮動小数点数	次のリニア補間を返します。 $(1-s) * x + s * y$

#### 13.4.3.3.7. localtime\_day

この関数はシステム時刻値からローカル時刻の日[1:31]を抽出します。

表13.64 のパラメータ localtime\_day

パラメータ	タイプ	説明
time	整数	システム時刻が返すタイムスタンプ
<return>	整数	抽出された日

#### 13.4.3.3.8. localtime\_hour

この関数はシステム時刻値のローカル時刻から時を抽出します。

表13.65 のパラメータ localtime\_hour

パラメータ	タイプ	説明
time	整数	システム時刻が返すタイムスタンプ
<return>	整数	抽出された時

#### 13.4.3.3.9. localtime\_minute

この関数はシステム時刻値のローカル時刻から分を抽出します。

表13.66 のパラメータ localtime\_minute

パラメータ	タイプ	説明
time	整数	システム時刻が返すタイムスタンプ
<return>	整数	抽出された分



#### 13.4.3.3.10. localtime\_month

この関数はシステム時刻値のローカル時刻から月[0:11]を抽出します。

表13.67 のパラメータ localtime\_month

パラメータ	タイプ	説明
time	整数	システム時刻が返すタイムスタンプ
<return>	整数	抽出された月

#### 13.4.3.3.11. localtime\_second

システム時刻値からローカル時刻の秒を抽出します。

表13.68 のパラメータ localtime\_second

パラメータ	タイプ	説明
time	整数	システム時刻が返すタイムスタンプ
<return>	整数	抽出された秒

#### 13.4.3.3.12. localtime\_weekday

この関数はシステム時刻値のローカル時刻から曜日[0:6]を抽出します。0は日曜日です。

表13.69 のパラメータ localtime\_weekday

パラメータ	タイプ	説明
time	整数	システム時刻が返すタイムスタンプ
<return>	整数	抽出された曜日

#### 13.4.3.3.13. localtime\_year

システム時刻値からローカル時刻の年を抽出します。

表13.70 のパラメータ localtime\_year

パラメータ	タイプ	説明
time	整数	システム時刻が返すタイムスタンプ

パラメータ	タイプ	説明
<return>	整数	抽出された年

#### 13.4.3.3.14. log10f

この関数は $x$ の10を底とする対数を返します。

表13.71 のパラメータ log10f

パラメータ	タイプ	説明
$x$	浮動小数点数	引数
<return>	浮動小数点数	戻り値

#### 13.4.3.3.15. logf

この関数は $x$ の自然対数を返します。

表13.72 のパラメータ logf

パラメータ	タイプ	説明
$x$	浮動小数点数	引数
<return>	浮動小数点数	戻り値

#### 13.4.3.3.16. maxf

この関数は2つの浮動小数点値の最大値を計算します。

表13.73 のパラメータ maxf

パラメータ	タイプ	説明
$x$	浮動小数点数	最初の値
$y$	浮動小数点数	2番目の値
<return>	浮動小数点数	$x$ および $y$ の最大値

#### 13.4.3.3.17. maxi

この関数は2つの整数値の最大値を計算します。

表13.74 のパラメータ `maxi`

パラメータ	タイプ	説明
<code>x</code>	整数	最初の値
<code>y</code>	整数	2番目の値
<return>	整数	※および <code>y</code> の最大値 <code>y</code>

#### 13.4.3.3.18. `minf`

この関数は2つの浮動小数点値の最小値を計算します。

表13.75 のパラメータ `minf`

パラメータ	タイプ	説明
<code>x</code>	浮動小数点数	最初の値
<code>y</code>	浮動小数点数	2番目の値
<return>	浮動小数点数	次の最小値: ※および <code>y</code>

#### 13.4.3.3.19. `mini`

この関数は2つの整数値の最小値を計算します。

表13.76 のパラメータ `mini`

パラメータ	タイプ	説明
<code>x</code>	整数	最初の値
<code>y</code>	整数	2番目の値
<return>	整数	次の最小値: ※および <code>y</code>

#### 13.4.3.3.20. `nearbyint`

この関数は最も近い整数に丸めます。

表13.77 のパラメータ `nearbyint`

パラメータ	タイプ	説明
<code>value</code>	浮動小数点数	丸める値

パラメータ	タイプ	説明
<return>	整数	丸めた値

#### 13.4.3.3.21. popDynamicStateMachine

この関数は、指定された動的ステートマシンを優先順位キューから削除します。

表13.78 のパラメータ popDynamicStateMachine

パラメータ	タイプ	説明
state		動的ステートマシンリストを持つステート
sm	文字列	動的ステートマシン

#### 13.4.3.3.22. powf

この関数は $x$ の $y$ 乗の値を返します。

表13.79 のパラメータ powf

パラメータ	タイプ	説明
$x$	浮動小数点数	引数 $x$
$y$	浮動小数点数	引数 $y$
<return>	浮動小数点数	戻り値

#### 13.4.3.3.23. pushDynamicStateMachine

この関数は優先順位キューに動的ステートマシンを挿入します。

表13.80 のパラメータ pushDynamicStateMachine

パラメータ	タイプ	説明
state		動的ステートマシンリストを持つステート
sm	文字列	動的ステートマシン
priority	整数	リスト内の動的ステートマシンの優先順位。0は1よりも高い優先度で扱われることに注意してください。

#### 13.4.3.3.24. rad2deg

この関数は角度をラジアンから度数に変換します。

表13.81 のパラメータ rad2deg

パラメータ	タイプ	説明
x	浮動小数点数	引数
<return>	浮動小数点数	戻り値

#### 13.4.3.3.25. rand

この関数は $0 \sim 2^{31} - 1$ の間の乱数値を取得します。

表13.82 のパラメータ rand

パラメータ	タイプ	説明
<return>	整数	$0 \sim 2^{31} - 1$ の間の乱数

#### 13.4.3.3.26. rgba2color

この関数はRGB色空間からEB GUIDE GTF色に変換します。

表13.83 のパラメータ rgba2color

パラメータ	タイプ	説明
red	整数	0～255の範囲の赤色座標
green	整数	0～255の範囲の緑色座標
blue	整数	0～255の範囲の青色座標
alpha	整数	0 (完全透明)～255 (不透明)の範囲のアルファ値
<return>	色	RGB色空間からEB GUIDE GTF色に変換され、アルファ値が適用された色

#### 13.4.3.3.27. round

この関数は最も近い整数に丸めます。ただし、中間の場合はゼロから遠ざかるように丸めます。

表13.84 のパラメータ round

パラメータ	タイプ	説明
value	浮動小数点数	丸める値

パラメータ	タイプ	説明
<return>	整数	丸めた値

### 13.4.3.4. EB GUIDEスクリプトの関数S~W

#### 13.4.3.4.1. saturate

この関数は、浮動小数点値を[0, 1]の範囲にクランプします。つまり、この関数は $\max(0, \min(1, x))$ を計算し、`clampf(0, 1, x)`の略式表記としての役割を果たします。`clampf(0, 1, x)`

表13.85 のパラメータ saturate

パラメータ	タイプ	説明
x	浮動小数点数	クランプする値
<return>	浮動小数点数	[0, 1]の範囲にクランプされた値×

#### 13.4.3.4.2. seed\_rand

この関数は乱数生成のシードを設定します。

表13.86 のパラメータ seed\_rand

パラメータ	タイプ	説明
seed	整数	乱数生成のシードとなる値
<return>	void	

#### 13.4.3.4.3. shutdown

この関数は、プログラムをシャットダウンするようにフレームワークに要求します。

#### 13.4.3.4.4. sinf

この関数はxの正弦を返します。このxはラジアン単位で指定します。

表13.87 のパラメータ sinf

パラメータ	タイプ	説明
x	浮動小数点数	引数

パラメータ	タイプ	説明
<return>	浮動小数点数	戻り値

#### 13.4.3.4.5. skin

この関数はすべてのデータプールアイテムのスキンを切り替えます。この動作は非同期で実行されます。

表13.88 のパラメータ skin

パラメータ	タイプ	説明
skin	skinType	切り替えるスキン(など) f:skin (s:Standard)
<return>	void	

#### 13.4.3.4.6. smoothstep

この関数は、値が  $[x_{\min}, x_{\max}]$  の範囲内にある場合、滑らかなエルミート補間  $3z^2 - 2z^3$  with  $z = (x - x_{\min}) / (x_{\max} - x_{\min})$  を計算します。この範囲内にはない場合は0を返します。この関数は  $[0, 1]$  の範囲の値を返します。

表13.89 のパラメータ smoothstep

パラメータ	タイプ	説明
xmin	浮動小数点数	xmin値
xmax	浮動小数点数	xmax値
x	浮動小数点数	補間される値
<return>	浮動小数点数	次のエルミート補間を返します。 $3z^2 - 2z^3$ with $z = (x - x_{\min}) / (x_{\max} - x_{\min})$

#### 13.4.3.4.7. sqrtf

この関数はxの負ではない平方根を返します。

表13.90 のパラメータ sqrtf

パラメータ	タイプ	説明
x	浮動小数点数	引数
<return>	浮動小数点数	戻り値

#### 13.4.3.4.8. string2float

この関数は文字列の最初の部分を浮動小数点数に変換します。

文字列の最初の部分の想定される形式は次のとおりです。

1. 先頭の空白(省略可能)
2. プラス(「+」)またはマイナス(「-」)符号(省略可能)
3. 次のいずれか:
  - ▶ 10進数
  - ▶ 16進数
  - ▶ 無限大
  - ▶ NAN (非数)

表13.91 のパラメータ string2float

パラメータ	タイプ	説明
str	文字列	文字列値
<return>	浮動小数点数	戻り値

#### 13.4.3.4.9. string2int

この関数は文字列の最初の部分を整数に変換します。入力が2147483647~-2147483648の範囲を超える場合、結果はこの範囲に切り捨てられます。文字列の先頭が数字ではない場合、関数は0を返します。

表13.92 のパラメータ string2int

パラメータ	タイプ	説明
str	文字列	文字列値
<return>	整数	戻り値

#### 13.4.3.4.10. string2string

この関数は、文字列の切り捨てによって指定の文字数のみを残すために使用します。

表13.93 のパラメータ string2string

パラメータ	タイプ	説明
str	文字列	切り捨て前の文字列
len	整数	文字列の最大長



パラメータ	タイプ	説明
<return>	文字列	切り捨て後の文字列

#### 13.4.3.4.11. substring

この関数は文字列の部分文字列コピーを作成します。負の終了インデックスを使用できます。

例:

- ▶ `substring("abc", 0, -1)` abcを返します。
- ▶ `substring("abc", 0, -2)` abを返します。
- ▶ `substring ("abcd", 1, 3)` bcを返します。

表13.94 のパラメータ `substring`

パラメータ	タイプ	説明
<code>str</code>	文字列	入力文字列
<code>startIndex</code>	整数	結果文字列の最初の文字インデックス
<code>endIndex</code>	整数	結果に含まれない最初の文字インデックス
<return>	文字列	言語の文字列

#### 13.4.3.4.12. system\_time

この関数は現在のシステム時刻を秒数で取得します。結果は、`localtime_*`関数に渡されることを想定したものになっています。

表13.95 のパラメータ `system_time`

パラメータ	タイプ	説明
<return>	整数	秒数のシステム時刻

#### 13.4.3.4.13. system\_time\_ms

この関数は現在のシステム時刻をミリ秒数で取得します。

表13.96 のパラメータ `system_time_ms`

パラメータ	タイプ	説明
<return>	整数	ミリ秒数のシステム時刻

#### 13.4.3.4.14. tanf

この関数はxの正接を返します。このxはラジアン単位で指定します。

表13.97 のパラメータ tanf

パラメータ	タイプ	説明
x	浮動小数点数	引数
<return>	浮動小数点数	戻り値

#### 13.4.3.4.15. trace\_dp

この関数はデータプールアイテムに関するデバッグ情報を、トレースログと接続ログに書き出します。

表13.98 のパラメータ trace\_dp

パラメータ	タイプ	説明
itemId	dp_id	デバッグ情報をトレースするアイテムのデータプールID
<return>	void	

#### 13.4.3.4.16. trace\_string

この関数は文字列をトレースログと接続ログに書き出します。

表13.99 のパラメータ trace\_string

パラメータ	タイプ	説明
str	文字列	トレースするテキスト
<return>	void	

#### 13.4.3.4.17. transformToScreenX

この関数はウィジェットとローカル座標を取得し、画面を基準にしたワールド座標系のx位置を返します。

表13.100 のパラメータ transformToScreenX

パラメータ	タイプ	説明
widget	ウィジェット	座標の基準にするウィジェット
localX	整数	ローカル座標のx位置

パラメータ	タイプ	説明
localY	整数	ローカル座標のY位置
<return>	整数	画面座標のX位置

#### 13.4.3.4.18. transformToScreenY

この関数はウィジェットとローカル座標を取得し、画面を基準にしたワールド座標系の位置のY位置を返します。

表13.101 のパラメータ transformToScreenY

パラメータ	タイプ	説明
widget	ウィジェット	座標の基準にするウィジェット
localX	整数	ローカル座標のX位置
localY	整数	ローカル座標のY位置
<return>	整数	画面座標のY位置

#### 13.4.3.4.19. transformToWidgetX

この関数はタッチ反応に提供されるウィジェットと画面座標を取得し、ウィジェットを基準にしたローカル座標系のX位置を返します。

表13.102 のパラメータ transformToWidgetX

パラメータ	タイプ	説明
widget	ウィジェット	座標の基準にするウィジェット
screenX	整数	画面座標のX位置
screenY	整数	画面座標のY位置
<return>	整数	ローカル座標のX位置

#### 13.4.3.4.20. transformToWidgetY

この関数はタッチ反応に提供されるウィジェットと画面座標を取得し、ウィジェットを基準にしたローカル座標系のY位置を返します。

表13.103 のパラメータ transformToWidgetY

パラメータ	タイプ	説明
widget	ウィジェット	座標の基準にするウィジェット

パラメータ	タイプ	説明
screenX	整数	画面座標のX位置
screenY	整数	画面座標のY位置
<return>	整数	ローカル座標のY位置

#### 13.4.3.4.21. trunc

この関数は常にゼロに近づくように最も近い整数に丸めます。

表13.104 のパラメータ trunc

パラメータ	タイプ	説明
value	浮動小数点数	丸める値
<return>	整数	丸めた値

#### 13.4.3.4.22. widgetGetChildCount

この関数は指定されたウィジェットの子ウィジェットの数を取得します。

表13.105 のパラメータ widgetGetChildCount

パラメータ	タイプ	説明
widget	ウィジェット	子ウィジェットの数を取得するウィジェット
<return>	整数	子ウィジェットの数

## 13.5. イベント

Table 13.106. イベントのプロパティ

プロパティ名	説明
Name	イベントの名前
Event ID	EB GUIDE TFがイベントを送受信するために使用する数値
Event group	イベントグループの名前  イベントグループには、EB GUIDE TFがイベントを送受信するために使用するIDがあります。

### 13.5.1. キーイベントの10進コード

Table 13.107. テンキーの10進コード

テンキー	10進コード
0	5
1	6
2	7
3	8
4	9
5	10
6	11
7	12
8	13
9	14

Table 13.108. ファンクションキーの10進コード

ファンクションキー	10進コード
F1キー	18
F2キー	19
F3キー	20
F4キー	21
F5キー	22
F6キー	23
F7キー	24
F8キー	25
F9キー	26
F10キー	27
F11キー	28
F12キー	29

Table 13.109. ASCIIキーの10進コード

ASCIIキー	10進コード
スペース	32
a	97
b	98

ASCIIキー	10進コード
c	99
d	100
e	101
f	102
g	103
h	104
i	105
j	106
k	107
l	108
m	109
n	110
o	111
p	112
q	113
r	114
s	115
t	116
u	117
v	118
w	119
x	120
y	121
z	122

## 13.6. model.json設定ファイル

model.jsonは、単一のEB GUIDEモデルに関連する設定アイテムが含まれているEB GUIDE TF設定ファイルです。

model.jsonファイルはエクスポートされたEB GUIDEモデルの一部です。

以下の表は、すべてのデフォルト設定パラメータをまとめたものです。

**NOTE**



**JSONオブジェクト表記**

EB GUIDE Studioで`model.json`を設定する場合は、JSONオブジェクト表記を使用します。

例えば、[13.6.1「EB GUIDE Studioでのサンプルmodel.json」](#)をご覧ください。

JSON形式の詳細については、<http://www.json.org>をご覧ください。

Table 13.110. 共通

設定アイテム	タイプ	説明	デフォルト値
<code>gtf.eventsystem.maxQueue</code>	整数	イベントキューの最大サイズ	0
<code>gtf.model.traces</code>	ブール値	<code>f:trace_string</code> スクリプト関数のトレースを有効にします	true
<code>gtf.model.identifier</code>	文字列	EB GUIDEモデルの一意的識別子(EB GUIDE StudioプロジェクトのUUIDと同等)	空
<code>gtf.model.identifier.short</code>	整数	EB GUIDEモデルの短い識別子	0xdeadbeaf
<code>gtf.model.initialLanguage.id</code>	文字列	EB GUIDEモデルの起動後にアクティブ化される言語の一意的識別子(EB GUIDE Studioプロジェクト内の言語のUUIDに等しい)。include <code>\export\aspects\languages.h</code> でのEB GUIDE Studioプロジェクトエクスポートの一環として、言語のUUID識別子を検索します。	未定義
<code>gtf.model.initialLanguage.force</code>	ブール値	trueの場合、すでに実行中のEB GUIDEモデルのアクティブな言語よりも <code>gtf.model.initialLanguage.id</code> の値が優先されず。それ以外の場合	false

設定アイテム	タイプ	説明	デフォルト値
		は、すでに実行中のEB GUIDEモデルのアクティブな言語の優先度のほうが高くなります。	
gtf.model.initialSkin.id	文字列	EB GUIDEモデルの起動後にアクティブ化されるスキンの一意識別子(EB GUIDE Studioプロジェクト内のスキンのUUIDに等しい)。include \export\aspects \skins.hでのEB GUIDE Studioプロジェクトエクスポートの一環として、スキンのUUID識別子を検索します。	未定義
gtf.model.initialSkin.force	ブール値	trueの場合、すでに実行中のEB GUIDEモデルのアクティブなスキンよりもgtf.model.initialSkin.idの値が優先されます。それ以外の場合は、すでに実行中のEB GUIDEモデルのアクティブなスキンの優先度のほうが高くなります。	false

Table 13.111. ファイルとパス

設定アイテム	タイプ	説明	デフォルト値
gtf.model.path	文字列	EB GUIDEモデルへのパス	None
gtf.model.config	文字列	EB GUIDEモデル設定へのフルパス	<gtf.model.path>/model.json
gtf.datapool.descriptionFile	文字列	データプール記述ファイルの名前	datapool.gtf
gtf.model.files.sm	文字列	ステートマシン記述ファイルの名前	model.bin



設定アイテム	タイプ	説明	デフォルト値
gtf.model.files.rm	文字列	リソース記述ファイルの名前	resources.bin
gtf.model.files.views	文字列	ビュー記述ファイルの名前	views.bin
gtf.model.files.types	文字列	タイプ記述ファイルの名前	types.bin
gtf.model.pluginstoload	文字列リスト	読み込むEB GUIDEモデルプラグインの名前	empty string list
gtf.eventsystem.mapFile	文字列	イベントシステムマッピングファイルの名前	eventMap.gtf

オプションgtf.model.coreNamesは、設定されているすべてのコアの名前が含まれている文字列リストです。以下の表には、すべてのコアの設定アイテムが記されています。

Table 13.112. コア

設定アイテム	タイプ	説明	デフォルト値
gtf.model.cores.<corename>.ownThread	ブール値	独自のスレッドを使用してコアを実行するかどうかを指定します	false
gtf.model.cores.<corename>.id	整数	コアコンテキスト識別子を指定します	0

オプションgtf.model.sceneNamesは、設定されているすべてのシーンの名前が含まれている文字列リストです。すべてのシーンには、次の表の設定アイテムがあります。

Table 13.113. シーン

設定アイテム	タイプ	説明	デフォルト値
gtf.model.scenes.<scenename>.visible	ブール値	シーンの可視性を指定します	true
gtf.model.scenes.<scenename>.width	整数	シーンの幅	800
gtf.model.scenes.<scenename>.height	整数	シーンの高さ	480
gtf.model.scenes.<scenename>.x	整数	シーンの始点の座標	0
gtf.model.scenes.<scenename>.y	整数	シーンの始点の座標	0
gtf.model.scenes.<scenename>.projectName	文字列	操作しているプロジェクトの名前	
gtf.model.scenes.<scenename>.windowCaption	文字列	表示されるウィンドウ名のテキスト	

設定アイテム	タイプ	説明	デフォルト値
gtf.model.scenes.<scenename>.-sceneId	整数	シーンの識別子	0
gtf.model.scenes.<scenename>.-maxFPS	整数	再描画率(FPS = フレーム/秒)。再描画率を無制限にするには、0に設定します。	60
gtf.model.scenes.<scenename>.-hwLayerId	整数	コアコンテキスト識別子を指定します	0
gtf.model.scenes.<scenename>.-colorMode	整数	色モードを指定します。 ▶ 1: 32ビット (RGBA8888) ▶ 2: 16ビット (RGB565) ▶ 3: 24ビット (RGB888) ▶ 4: 32ビットsRGB ▶ 5: 32ビットsRGB(エミュレーション)	1
gtf.model.scenes.<scenename>.-multisampling	整数	シーンのマルチサンプリングを指定します ▶ 0: マルチサンプリングなし ▶ 1: 2xマルチサンプリング ▶ 2: 4xマルチサンプリング	0
gtf.model.scenes.<scenename>.-enableRemoteFramebuffer	ブール値	trueである場合、シミュレーションウィンドウへの画面外のバッファの転送が有効になります	false
gtf.model.scenes.<scenename>.-showWindowFrame	ブール値	レンダラーウィンドウフレームを表示するかどうかを指定します	true
gtf.model.scenes.<scenename>.-showWindow	ブール値	trueである場合、Windowsベースのシ	true

設定アイテム	タイプ	説明	デフォルト値
		STEMでシミュレーション用の追加ウィンドウが開きます	
gtf.model.scenes.<scenename>.-disableVsync	ブール値	trueである場合、レンダラーの垂直同期が無効になります。	false
gtf.model.scenes.<scenename>.-showFPS	整数	使用可能な値: ▶ 0: FPS を表示しない ▶ 1: 画面にFPSを表示する ▶ 2: コンソールにFPSを表示する ▶ 3: 画面およびコンソールにFPSを表示する ▶ 4: コンソールに拡大テキストでFPSを表示する ▶ 5: 画面およびコンソールに拡大テキストでFPSを表示する	0
gtf.model.scenes.<scenename>.-renderer	文字列	使用するレンダラーの名前: OpenGLRendererまたは OpenGL3Renderer	

Table 13.114. レンダリング共通

設定アイテム	タイプ	説明	デフォルト値
gtf.model.fontCache.width	整数	フォントキャッシュアトラステクスチャの幅	512
gtf.model.fontCache.height	整数	フォントキャッシュアトラステクスチャの高さ	512
gtf.model.fontCache.age	整数	フォントキャッシュのリフレッシュ操作の実行が	100

設定アイテム	タイプ	説明	デフォルト値
		必要となるまでの最大許容時間	
gtf.model.traversalStackSize	整数	レンダラートラバーススタックサイズ(バイト単位)	32768

次の表の各設定アイテムは相互に対応しています。つまり、レンダラーは、3つのリストすべてに同じ数のアイテムがあることを想定しています。1つのリストのあるインデックスのエントリーは、他のリストの同じインデックスのエントリーに対応しています。

Table 13.115. レンダラー表示拡張機能

設定アイテム	タイプ	説明	デフォルト値
gtf.model.displayId	整数リスト	シーンの識別子	
gtf.model.maxCacheSize	整数リスト	シーンの最大テクスチャキャッシュ	
gtf.model.driverName	文字列リスト	OS シーンの固有のドライバ名。例えば、/dev/fb0	

以下の表の設定アイテムは、TextEngineコンポーネントの設定に使用します。TextEngine は、サードパーティライブラリFreeTypeに基づいています。以下のパラメータはFreeType実装に渡されます。FreeTypeの詳細については、[https://www.freetype.org/freetype2/docs/reference/ft2-cache\\_subsystem.html](https://www.freetype.org/freetype2/docs/reference/ft2-cache_subsystem.html)をご覧ください。

EB GUIDE TFのフォントサイズ処理方法により、ft\_sizeオブジェクトがft\_faceオブジェクトと別にキャッシュされることはありません。max\_sizesの値がターゲットプラットフォームのハードウェアによって制限される可能性があることを考慮してください。

Table 13.116. TextEngine 設定アイテム

設定アイテム	タイプ	説明	デフォルト値
gtf.model.textengine.-replacementGlyph	整数	専用のフォント文字が現在のフォントに見つからない場合に使用されるUnicode文字	0xffffd
gtf.model.textengine.maxFaces	整数	キャッシュされるフォントフェイスの最大数	0
gtf.model.textengine.maxSizes	整数	キャッシュされるフォントサイズの最大数	0
gtf.model.textengine.maxBytes	整数	キャッシュに使用できるメモリの最大バイト数	0
gtf.model.textengine.-fontCacheThreshold	整数	キャッシュされるフォントの好ましい数	2

設定アイテム	タイプ	説明	デフォルト値
gtf.model.textengine.- enablePlainFileStream	ブール値	フォントへのアクセスに関する設定を決定します。trueの場合、プレーンファイルへのI/Oアクセスが使用されます。falseの場合、ROMマップ済みファイルへのアクセスが使用されます。	false

**NOTE**



**ビットマップフォントの設定アイテム**

.fntビットマップフォントで使用できるのは、replacementGlyph設定アイテムのみです。[table 13.116. “TextEngine 設定アイテム”](#)のその他の設定アイテムは、ビットマップフォントでは使用できません。

**NOTE**



**ROMマップ済みファイルアプローチとプレーンファイルI/Oアプローチの比較**

一般的に、ROMマップ済みファイルアプローチのほうがパフォーマンスが優れています。ただしQNXなど一部のシステムでは、プレーンファイルI/Oアプローチよりメモリを多く消費してしまいます。プレーンファイルI/Oアプローチは、概してROMマップ済みファイルアプローチよりもメモリの消費量が抑えられます。ただし、パフォーマンスは低くなる場合があります。

オプションgtf.model.touchDevicesNamesは、設定されているすべてのタッチデバイスの名前が含まれている文字列リストです。すべてのタッチデバイスで、次の表に示されている設定アイテムを使用できます。

Table 13.117. タッチデバイス

設定アイテム	タイプ	説明	デフォルト値
gtf.model.- touchDevices.<deviceName>.- touchscreenType	整数	タッチデバイスタイプを定義します  ▶ 0: Galaxy ▶ 1: imx WVGA ▶ 2: Mouse ▶ 3: General ▶ 4: Lilliput_889GL ▶ 5: General Multitouch ▶ 6: Lilliput with	3

設定アイテム	タイプ	説明	デフォルト値
		automatic calibration ▶ 7: Generic Touch Configuration	
gtf.model.- touchDevices.<deviceName>.- displayManagerId	整数	デバイスが有効なシーンIDを指定します	0
gtf.model.- touchDevices.<deviceName>.touchId	整数	デバイスのIDを指定します	0
gtf.model.- touchDevices.<deviceName>.- minimalDistanceToMove	整数	タッチ位置の変更に反応するしきい値	0
gtf.model.- touchDevices.<deviceName>.- touchMoveRepeatTimeout	整数	タッチ位置の変更通知と次の変更通知の間の遅延	0
gtf.model.- touchDevices.<deviceName>.width	整数	タッチ可能なデバイス領域の幅	0
gtf.model.- touchDevices.<deviceName>.height	整数	タッチ可能なデバイス領域の高さ	0
gtf.model.- touchDevices.<deviceName>.x_high	整数	タッチ可能なデバイス領域の最大水平方向解像度の限度	0
gtf.model.- touchDevices.<deviceName>.y_high	整数	タッチ可能なデバイス領域の最大垂直方向解像度の限度	0
gtf.model.- touchDevices.<deviceName>.x_low	整数	タッチ可能なデバイス領域の最小水平方向解像度の限度	0
gtf.model.- touchDevices.<deviceName>.y_low	整数	タッチ可能なデバイス領域の最小垂直方向解像度の限度	0
gtf.model.- touchDevices.<deviceName>.- devicePath	文字列	タッチに使用されるドライバの名前。例えば、/dev/input0	

以下の表の設定アイテムは、レンダラーのバイナリシェーダーキャッシュの設定に使用されます。

Table 13.118. シェーダー

設定アイテム	タイプ	説明	デフォルト値
gtf.model.binShadersLocation	文字列	ライブラリシェーダーを配置する必要があるEB GUIDE GTFバイナリディレクトリ配下のフォルダーの名前。絶対パスにすることもできます。	
gtf.model.readBinShaders	ブール値	前回の実行時に書き込まれたバイナリシェーダーをディスクから読み取るかどうかを決定します。	false
gtf.model.writeBinShaders	ブール値	バイナリシェーダーを今後の実行で再利用されるディスクに書き込むかどうかを決定します。	false

### 13.6.1. EB GUIDE Studioでのサンプルmodel.json



**Example 13.1.**  
EB GUIDE Studioでのmodel.json

```
{
  "gtf": {
    "model": {
      "coreNames": [
        "HMI"
      ],
      "cores": {
        "HMI": {
          "id": 0,
          "ownThread": false
        }
      },
      "sceneNames": [
        "Main"
      ],
      "scenes": {
        "Main": {
```

```
        "name": "Main",
        "visible": true,
        "width": 800,
        "height": 480,
        "x": 0,
        "y": 0,
        "projectName": "project",
        "windowCaption": "EB GUIDE 6 model",
        "sceneId": 0,
        "maxFPS": 60,
        "hwLayerId": 0,
        "colorMode": 1,
        "multisampling": 0,
        "enableRemoteFramebuffer": false,
        "showWindowFrame": true,
        "showWindow": true,
        "disableVSync": false,
        "showFPS": 0,
        "renderer": "OpenGL3Renderer",
        "context": 0
    }
},
"identifier": "29691ce7-cb4c-4337-8852-93c90c62e624",
"pluginstoload": [
    "GtfGui",
    "GtfGuiOpenGLES3"
]
}
}
```

## 13.7. OpenGL ES拡張機能

OpenGLおよびOpenGL3レンダラーでは、以下に示すさまざまなOpenGL ES拡張機能を使用します。

拡張機能を利用できない場合は、レンダリングに制限があります。



**NOTE**



以下の表に示す複数の拡張機能

ORを使用して表示されている複数の拡張機能: 表示されている拡張機能のいずれかを使用できない場合、制限が適用されます。

ANDを使用して表示されている複数の拡張機能: 表示されているすべての拡張機能を使用できない場合、制限が適用されます。

Table 13.119. OpenGL拡張機能

OpenGL拡張機能	OpenGLレンダラーで使用できない場合の制限	OpenGL3レンダラーで使用できない場合の制限
GL_OES_get_program_binary	バイナリシェーダーキャッシュはサポートされません。	なし
GL_EXT_texture_compression_s3tc AND GL_EXT_texture_compression_dxt1	DXT1テクスチャの圧縮はサポートされません。	DXT1テクスチャの圧縮はサポートされません。
GL_EXT_texture_compression_s3tc	DXT3およびDXT5テクスチャの圧縮はサポートされません。	DXT3およびDXT5テクスチャの圧縮はサポートされません。
GL_IMG_texture_compression_pvrtc	PVRTCテクスチャの圧縮はサポートされません。	PVRTCテクスチャの圧縮はサポートされません。
GL_IMG_texture_compression_pvrtc2	PVRTC2テクスチャの圧縮はサポートされません。	PVRTC2テクスチャの圧縮はサポートされません。
GL_KHR_texture_compression_astc_ldr	ASTCテクスチャの圧縮はサポートされません。	ASTCテクスチャの圧縮はサポートされません。
GL_OES_depth32 AND GL_OES_depth24	<ul style="list-style-type: none"> <li>▶ シーングラフは低い深さ精度でレンダリングされます。</li> <li>▶ ウィジェットの数は1,600万ではなく65534に制限されます。</li> <li>▶ EB GUIDE SDKインターフェースでは、フレームバッファを16ビット以上の深さ精度で作成することはできません。</li> </ul>	なし
GL_EXT_shader_texture_lod OR GL_EXT_texture_rg	イメージベースドライティングはサポートされません。	なし

OpenGL拡張機能	OpenGLレンダラーで使用できない場合の制限	OpenGL3レンダラーで使用できない場合の制限
<p>OR</p> <p>GL_OES_texture_float</p> <p>OR</p> <p>GL_OES_texture_half_float</p>		
<p>GL_EXT_multisampled_render_to_texture</p> <p>AND</p> <p>GL_IMG_multisampled_render_to_texture</p> <p>AND</p> <p>(GL_NV_framebuffer_blitORGL_NV_framebuffer_multisample)</p> <p>AND</p> <p>(GL_ANGLE_framebuffer_blitORGL_ANGLE_framebuffer_multisample)</p>	シーングラフおよびEB GUIDE SDK オフスクリーンフレームバッファのマルチサンプリングはサポートされません。	なし
GL_OES_texture_float	シーングラフのトーンマッピングはサポートされません。	なし
<p>GL_EXT_texture_border_clamp</p> <p>AND</p> <p>GL_OES_texture_border_clamp</p> <p>AND</p> <p>GL_NV_texture_border_clamp</p>	アルファマスクウィジェット以外のエリアに視覚的な不具合が生じる可能性があります。	アルファマスクウィジェット以外のエリアで視覚的な不具合が生じる可能性があります。
GL_OES_element_index_uint	機能の制限はありませんが、パフォーマンスが低下する可能性があります。	なし

OpenGL拡張機能	OpenGLレンダラーで使用できない場合の制限	OpenGL3レンダラーで使用できない場合の制限
GL_OES_packed_depth_stencil	機能の制限はありませんが、パフォーマンスが低下する可能性があります。	なし
GL_OES_vertex_half_float	機能の制限はありませんが、パフォーマンスが低下し、グラフィックスメモリの消費量が増加する可能性があります。	なし
GL_OES_vertex_type_10_10_2_10_2	機能の制限はありませんが、パフォーマンスが低下し、グラフィックスメモリの消費量が増加する可能性があります。	なし
GL_EXT_sRGB	機能の制限はありませんが、sRGBカラーモードでのシーングラフまたはsRGBテクスチャのパフォーマンスが低下する可能性があります。	なし

## 13.8. platform.json設定ファイル

platform.jsonは、共通アイテムおよびプラットフォーム依存アイテムが含まれているEB GUIDE TF設定ファイルです。

platform.jsonファイルはエクスポートされたEB GUIDEモデルの一部です。

以下の表は、すべてのデフォルト設定パラメータをまとめたものです。

### NOTE



#### JSONオブジェクト表記

EB GUIDE Studio内でplatform.jsonを設定する場合は、JSONオブジェクト表記を使用します。

例えば、[13.8.1「EB GUIDE Studioでのサンプルplatform.json」](#)をご覧ください。

JSON形式の詳細については、<http://www.json.org>をご覧ください。

Table 13.120. プラットフォームの設定

設定アイテム	タイプ	説明	デフォルト値
gtf.servicemapper.port	整数	サービスの接続ポート(例: EB GUIDE Monitor)	60000

設定アイテム	タイプ	説明	デフォルト値
gtf.core.pluginstoload	文字列リスト	読み込むコアプラグインのリスト(バイナリフォルダの相対パス、または絶対パス)	None
gtf.launcher.editmode	ブール値	EB GUIDE TFがEB GUIDE Studioで実行されるかどうかを定義します。これは読み取り専用アイテムです。	false
gtf.platform.config	文字列	platform.json ファイルへのフルパス。これは読み取り専用アイテムです。	<model_folder>/platform.json
gtf.framework.path	文字列	GtfStartup実行可能ファイルへのパス。これは読み取り専用アイテムです。	<binary_folder>
gtf.diagnostic.memory.interval	整数	メモリー診断の間隔を指定します。値が0の場合、診断は非アクティブ化されます。	0
gtf.ipc.role	文字列	IPCノードの役割。使用可能な値はserverまたはclient	server
gtf.ipc.discovery.network	文字列	サーバークライアント検出メカニズムで使用されるIPv4ネットワークアドレス。直接接続の場合、これはサーバーのネットワークアドレスを表します。	255.255.255.255
gtf.ipc.discovery.port	整数	サーバークライアント検出メカニズムで使用されるネットワークポート。直接接続の場合、これはサーバー設定のアイテムgtf.-servicemapper.-	4711

設定アイテム	タイプ	説明	デフォルト値
		portと等しい必要があります。	
gtf.ipc.datapool.config	文字列	プロセス間通信の一部となるデータプールアイテムが含まれている設定ファイル	ipc_datapool.gtf
gtf.ipc.discovery.mode	文字列	サーバーとクライアントを接続するために使用される検出モード。使用可能なオプションは、"broadcast"、"multicast"および"direct"です。	broadcast
gtf.ipc.client.timeout	整数	サーバーへのクライアント接続の再試行時間(ミリ秒単位)。	5000
gtf.osal.threading.pool.-threadCount	整数	スレッドプールで作成および使用されるスレッドの数。有効な値の範囲は1~32です。他のすべての値はクランプされます。	1
gtf.resourcesystem.residentformats	文字列リスト	リソースに一度付加されたリソースシステム形式は、そのリソースから削除できません。前の設定形式を置き換える唯一の方法は、そのリソースに新しい設定形式を付加することです。	デフォルト値については、 <a href="#">13.8.1「EB GUIDE Studioでのサンプルplatform.-json」</a> の"residentformats"をご覧ください。
gtf.resourcesystem.defaultdecoding	オブジェクトリスト	実際のデコード要求の前に、リソースシステムで事前にデコードを行うために使用される入力および出力形式のペア。	デフォルト値については、 <a href="#">13.8.1「EB GUIDE Studioでのサンプルplatform.-json」</a> の"defaultdecoding"をご覧ください。
gtf.resourcesystem.cache.softlimit	整数	リソースシステムキャッシュの上限サイズをMB	-1

設定アイテム	タイプ	説明	デフォルト値
		単位で指定します。上限を超えた場合は、クリーンアップアクションがキューに追加されます。	
gtf.resourcesystem.cache.hardlimit	整数	リソースシステムキャッシュの上限サイズをMB単位で指定します。上限を超えた場合は、クリーンアップアクションが実行されます。softlimitの場合とは異なり、クリーンアップアクションは即座に実行されます。	-1

### 13.8.1. EB GUIDE Studioでのサンプルplatform.json



**Example 13.2.**  
EB GUIDE Studioでのplatform.json

```
{
  "gtf": {
    "core": {
      "pluginstoload": [
        "GtfService",
        "GtfRuntime"
      ]
    },
    "servicemapper": {
      "port": 60000
    },
    "resourcesystem": {
      "defaultdecoding": [
        {
          "inputformat": "gtf::decoder::ImagePath",
          "outputformat": "gtf::decoder::EncodedMemImage"
        },
        {
          "inputformat": "gtf::decoder::MeshPath",
          "outputformat": "gtf::decoder::EncodedMeshData"
        }
      ]
    }
  }
}
```

```
{
  "inputformat": "gtf::decoder::IBLPath",
  "outputformat": "gtf::decoder::EncodedIBLData"
},
{
  "inputformat": "gtf::decoder::EncodedMemImage",
  "outputformat": "gtf::decoder::MemImage"
},
{
  "inputformat": "gtf::decoder::DescriptorResource::font",
  "outputformat": "gtf::decoder::FontData"
},
{
  "inputformat": "gtf::decoder::DescriptorResource::image",
  "outputformat": "gtf::decoder::ImagePath"
},
{
  "inputformat": "gtf::decoder::DescriptorResource::ibl",
  "outputformat": "gtf::decoder::IBLPath"
},
{
  "inputformat": "gtf::decoder::DescriptorResource::mesh",
  "outputformat": "gtf::decoder::MeshPath"
},
{
  "inputformat": "gtf::decoder::MemImage",
  "outputformat": "gtf::scdr::decoder::NinePatchImage"
},
{
  "inputformat": "gtf::scdr::decoder::NinePatchImage",
  "outputformat": "gtf::scdr::decoder::DefaultMemImage"
}
],
"residentformats": [
  "gtf::decoder::FontData",
  "gtf::decoder::IBLPath",
  "gtf::decoder::ImagePath",
  "gtf::decoder::MeshPath"
]
}
}
```

## 13.9. シーン

Table 13.121. シーンのプロパティ

プロパティ名	説明
height	ハプティックステートマシンのビューがあるエリアの高さは、対象デバイス上でレンダリングされます。
width	ハプティックステートマシンのビューがあるエリアの幅は、対象デバイス上でレンダリングされます。
x	ハプティックステートマシンのビューがあるエリアのXオフセットは、対象デバイス上でレンダリングされます。
y	ハプティックステートマシンのビューがあるエリアのYオフセットは、対象デバイス上でレンダリングされます。
visible	trueである場合、ステートマシンおよび子ウィジェットが表示されます。
projectName	EB GUIDEプロジェクトの名前
windowCaption	ウィンドウフレームに表示されるテキスト
sceneID	入力処理などに使用できる一意のシーン識別子
maxFPS	再描画率(FPS = フレーム/秒)  再描画率を無制限にするには、0に設定します。
hwLayerID	現在のステートマシンにマップされる対象デバイスの表示上のハードウェアレイヤーのID
colorMode	<p>使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ 32-bit (=1): RGBA8888キー</li> <li>▶ 16-bit (=2): RGB565キー</li> <li>▶ 24-bit (=3): RGB888キー</li> <li>▶ 32-bit sRGB (=4):</li> </ul> <p>この値は、グラフィックス処理ユニットのハードウェアサポートを使用します。</p> <p>イメージウィジェットまたはディフューズテクスチャウィジェット機能でsRGBサポートが必要な場合は、この値を使用します。</p> <ul style="list-style-type: none"> <li>▶ 32-bit sRGB (Emulated) (=5):</li> </ul> <p>32-bit sRGBで適切な結果が得られない場合に限り、この値を使用します。</p>
multisampling	<p>使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ Off (= 0): マルチサンプリングなし</li> <li>▶ 2x (=1): 2xマルチサンプリング</li> </ul>



プロパティ名	説明
	<ul style="list-style-type: none"> <li>▶ 4x (=2): 4xマルチサンプリング</li> </ul> <p><a href="#">“マルチサンプリングの設定”</a>もご覧ください。</p>
enableRemoteFramebuffer	trueである場合、シミュレーションウィンドウへの画面外のバッファの転送が有効になります
showWindowFrame	trueである場合、シミュレーションウィンドウにフレームが表示されます。このフレームを使用すると、ウィンドウをグラブして移動できます。
showWindow	trueである場合、Windowsベースのシステムでシミュレーション用の追加ウィンドウが開きます。
disableVSync	trueである場合、レンダラーの垂直同期が無効になります。
showFPS	<p>使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ Off (=0): FPS を表示しない</li> <li>▶ On screen (=1): 画面にFPSを表示する</li> <li>▶ Console (=2): コンソールにFPSを表示する</li> <li>▶ Console &amp; on screen (=3): 画面およびコンソールにFPSを表示する</li> <li>▶ On screen (large text) (=4)</li> <li>▶ Console &amp; on screen (large text) (=5)</li> </ul>
Renderer	<p>シーンのレンダラーを定義します。</p> <p>使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ OpenGLRenderer</li> <li>▶ OpenGL3Renderer</li> </ul>

**TIP**



**マルチサンプリングの設定**

マルチサンプリングの解像度を高くするほど、レンダリング結果の画質はよくなります。ただし、マルチサンプリングを行うと、特に対象デバイスで、レンダリングパフォーマンスが低下することに注意してください。高解像度の小型ディスプレイでは、マルチサンプリングの効果はほとんどありません。

マルチサンプリングなしで開始し、パフォーマンスが良好であれば、2xまたは4xのマルチサンプリングを試してください。マルチサンプリングを高くしても大きな差がない場合は、低いほうの設定を使用してください。

**TIP**



**マルチサンプリングの設定はハードウェア依存**

必要なマルチサンプリング設定がハードウェア側で使用できない場合は、それに関する情報をログファイルで参照できます。

**NOTE**



**シーン設定でのsceneIDの使用**

シーン設定で同じsceneIDを使用している場合、複数のステートマシンが入力処理に同時に反応します。

これを避け、1つのステートマシンのみが入力処理に反応するようにするには、シーン設定で各ステートマシンに異なるsceneID値を割り当てます。

## 13.10. がサポートするタッチスクリーンタイプ EB GUIDE GTF

サポートされるタイプは、対象デバイスによって異なります。

表13.122 がサポートするタッチスクリーンタイプ EB GUIDE GTF

値	説明	プラットフォーム
0	Galaxy	Linux
1	IMX WVGA	Linux
2	マウスインターフェイスに接続されたタッチスクリーン	すべて
3	一般的なプラットフォーム依存のタッチスクリーンインターフェイス	すべて
4	Lilliput 889GL	QNX

値	説明	プラットフォーム
5	一般的なプラットフォーム依存のマルチタッチ タッチスクリーンインターフェイス	Linux

## 13.11. tracing.json設定ファイル

tracing.jsonは、Tracing依存アイテムが含まれているEB GUIDE TF設定ファイルです。

tracing.jsonファイルは、エクスポートされたEB GUIDEモデルまたは標準デリバリの一部ではありません。そのため、お使いのPCでtracing.jsonファイルを作成する必要があります。

### NOTE

#### JSONオブジェクト表記



例えば、[13.11.2「tracing.jsonの例」](#)をご覧ください。

JSON形式の詳細については、<http://www.json.org>をご覧ください。

以下の表にデフォルトの設定パラメータを示します。

Table 13.123. 設定アイテムのトレース

アイテム	タイプ	説明	デフォルト値
gtf.tracing.Plugins	文字列リスト	出力プラグインのトレース	空リスト。StdErr/ StdErr/StdOut (Android以外のすべての ハードウェア)やLogCat (Android)など統合型出力 プラグインが使用されます。
gtf.tracing.- ChannelSeverities	文字列リスト	指定されたチャンネル用に個 別設定された重大度	
gtf.tracing.- DefaultSeverity	文字列	デフォルトのトレース重大度  重大度レベルの詳細につ いては、 <a href="#">13.11.1「重大度レ ベル」</a> をご覧ください。	Notice
gtf.tracing.- output.path	ブール値	トレース出力ファイ ルlog.txtをエクスポート する場所	tracing.jsonが格納さ れているのと同じディレクト リです。
gtf.tracing.- output.enabled	ブール値	トレース出力ファイルのエク スポートを有効にします。	true
gtf.tracing.- config.path	文字列	tracing.jsonファイルへ のパス	

アイテム	タイプ	説明	デフォルト値
gtf.tracing.- gtffileoutput.- severity	文字列	GtfFileOutputプラグインの重大度	
gtf.tracing.- gtfkernel eventoutput.- severity	文字列	GtfKernelEvent Output プラグインの重大度	
gtf.tracing.- gtfvsdebugoutput.- severity	文字列	GtfVsDebugOutputプラグインの重大度	
gtf.tracing.- stdoutoutput.severity	文字列	重大度: StdErrOutput	
gtf.tracing.- logcatoutput.- severity	文字列	重大度: LogCatOutput	
gtf.tracing.- output.- printChannelname	ブール値	トレース通知内のチャンネル名の印刷を有効にします。設定アイテム名のoutputはトレース出力名 (gtffileoutput、gtfkernelevent output、stdoutoutput、gtfvsdebugoutput など)と置き換える必要があります。	true
gtf.tracing.- output.- printTimestamp	ブール値	トレース通知内のタイムスタンプの印刷を有効にします。設定アイテム名のoutputはトレース出力名 (gtffileoutput、gtfkernelevent output、stdoutoutput、gtfvsdebugoutput など)と置き換える必要があります。	true
gtf.tracing.- output.- printSeverity	ブール値	トレース通知内の重大度の印刷を有効にします。設定アイテム名のoutputはトレース出力	true

アイテム	タイプ	説明	デフォルト値
		名 (gtffileoutput、gtfkernelevent output、stdoutoutput、gtfvsdebugoutput など)と置き換える必要があります。	
gtf.tracing.- PeriodicOutputFlush	整数	ミリ秒単位の周期的な出力フラッシュの値。  値0は各トレースの後にフラッシュが実行されることを意味します。0より小さい値はフラッシュが実行されないことを意味します。	-1

### 13.11.1. 重大度レベル

以下の表に使用可能なトレース重大度レベルを示します。

Table 13.124. 重大度レベル

重大度レベル	説明
None	このレベルは、トレースが無効になっていて作成されないことを示します。
Fatal	このレベルは、アプリケーションの停止につながる可能性がある重大なエラーを示します。
Error	このレベルは、アプリケーションが実行をまだ継続できる可能性があるエラーイベントを示します。
Warning	このレベルは、問題が起こりうる状況、または一部のインターフェースが想定どおりに使用されていないことを示します。
Notice	このレベルは、アプリケーションの進捗を高レベルで説明する情報メッセージを示します。
Info	このレベルは、アプリケーションの進捗を大まかに説明する情報メッセージを示します。
Debug	このレベルは、アプリケーションの進捗を詳細に説明する情報メッセージを示します。このレベルはアプリケーションのデバッグに役立ちます。

**NOTE**



含まれるレベル

[table 13.124](#), “[重大度レベル](#)”に表示されるレベルには、使用しているものよりも上のレベルの情報が自動的に含まれます(ただし、Noneレベルを除きます)。つまり、Error重大度レベルを使用している場合は、重大度がFatalの情報も含まれます。また、Debug重大度レベルを選択すると、上記の重大度のうちNoneを除くすべての情報が含まれます。

## 13.11.2. tracing.jsonの例



**Example 13.3.**  
tracing.json

```
{
  "gtf":
  {
    "tracing":
    {
      "Plugins": ["GtfFileoutput"],
      "ChannelSeverities":
      ["GTF_Launcher", "Warning",
      "GTF_PluginLoader", "Notice"],
      "DefaultSeverity": "Info"
    }
  }
}
```

トレース使用の別の例については、EB GUIDE SDKの例をご覧ください。すべての EB GUIDE SDKサンプルが付属するEB\_GUIDE\_Examples.zipアーカイブは、<https://www.elektrobit.com/ebguide/learn/resources/>からダウンロードできます。なお、EB GUIDE SDKサンプルの操作方法については、EB GUIDE Studio Howto Using examples in EB GUIDE Studio.pdfファイル(上記の.zipアーカイブに収録)をご覧ください。

## 13.12. ウィジェット

### 13.12.1. ビュー

表13.125 ビューのプロパティ

プロパティ名	説明
height	ピクセル単位のウィジェットの高さ

プロパティ名	説明
width	ピクセル単位のウィジェットの幅
visible	trueである場合、ウィジェットおよび子ウィジェットが表示されます。
x	ウィジェットのX座標
y	ウィジェットのY座標

ビューステートとビューテンプレートには、View Transition Animation用に追加のプロパティがあります。View Transition Animationは、開始アニメーション、終了アニメーション、変更アニメーション、ポップアップオンアニメーション、およびポップアップオフアニメーションに適用されます。

表 13.126 ビュー遷移アニメーションのプロパティ

プロパティ名	説明
enabled	アニメーションを実行するかどうかを定義します。
repeat	繰り返しの回数。0は無無限数を表します。
alternating	trueの場合、アニメーションは前後(双方向)に繰り返し実行されます。 falseの場合、アニメーションは1つの方向のみ(単一方向)に繰り返し実行されます。 繰り返し回数はrepeatプロパティで定義されます。
scale	アニメーション時間を乗じる際の係数。
onPlay	アニメーションが開始または続行された場合に実行される反応。パラメータ: 開始時間および再生の方向(trueの場合は順方向、falseの場合は逆方向)。
onPause	アニメーションが一時停止した場合に実行される反応。パラメータ: 現在のアニメーション時間。
onTerminate	アニメーションが完了した場合に実行される反応。最初のパラメータ: アニメーション時間。2番目のパラメータ: 終了の理由。次のようにエンコードされています。 <ul style="list-style-type: none"> <li>▶ 0: アニメーションは完了します。</li> <li>▶ 1: アニメーションはキャンセルされます。これは、によってトリガーされます。 f:animation_cancel</li> <li>▶ 2: ビュー遷移が原因でウィジェットが破棄されます。</li> <li>▶ 3: アニメーションは最後のステップにジャンプします。これは、によってトリガーされます。f:animation_cancel_end</li> <li>▶ 4: アニメーションは最初のステップにジャンプしてからキャンセルされます。これは、によってトリガーされます。f:animation_cancel_reset</li> </ul>

## 13.12.2. 基本ウィジェット

基本ウィジェットは8つあります。

- ▶ アルファマスク
- ▶ アニメーション
- ▶ コンテナー
- ▶ 楕円
- ▶ イメージ
- ▶ インスタンスエータ
- ▶ ラベル
- ▶ 四角形

次のセクションでは、基本ウィジェットのプロパティをリストします。

注記



一意の名前

同じ親ウィジェットを持つ2つのウィジェットには一意の名前を使用してください。

注記



負の値

heightおよびwidthプロパティに負の値を使用しないでください。EB GUIDE Studioは負の値を0として扱うため、各ウィジェットが描出されなくなってしまう。

### 13.12.2.1. アルファマスク

アルファマスクとは、子ウィジェットのアルファチャネル(オパシティ)をあるイメージによって制御するコンテナーウィジェットです。

表13.127 アルファマスクのプロパティ

プロパティ名	説明
visible	trueである場合、ウィジェットおよび子ウィジェットが表示されます。
width	ピクセル単位のウィジェットの幅
height	ピクセル単位のウィジェットの高さ
x	親ウィジェットを基準にしたウィジェットのX座標
y	親ウィジェットを基準にしたウィジェットのY座標
enabled	trueの場合、アルファマスクは子ウィジェットに適用されます。
image	アルファチャネル(子ウィジェットのオパシティ)を制御するイメージ
horizontalAlign	ウィジェットの境界内のイメージファイルの水平方向の位置揃え



プロパティ名	説明
verticalAlign	ウィジェットの境界内のイメージファイルの垂直方向の位置揃え
scaleMode	イメージの拡大縮小モード。使用可能な値: ▶ original size (0) ▶ fit to size (1) ▶ keep aspect ratio (2)

注記



サポートされているアルファマスク用イメージファイル形式

使用可能なイメージファイル形式は、レンダラーの実装によって異なります。レンダラー (OpenGL ESバージョン2.0以降向け)では、.pngファイルおよび .jpgファイルがサポートされます。RGBイメージは、グレイスケールイメージに変換されてから アルファマスクとして使用されます。グレイスケールイメージはそのまま使用されます。イメージのアルファチャンネルは無視されます。

アルファマスク機能は9-patchイメージには適用されません。9-patchイメージは PNGおよび JPEGファイル形式と同じ方法で処理されます。

### 13.12.2.2. アニメーション

アニメーションは、ビューに従ってウィジェットの動きを定義します。アニメーションの外観を定義するには、[アニメーション]エディターで曲線を追加します。

表13.128 アニメーションのプロパティ

プロパティ名	説明
enabled	アニメーションを実行するかどうかを定義します。
repeat	繰り返しの回数。0は無限数を表します
alternating	trueの場合、アニメーションは前後/双方向に繰り返し実行されます。 falseの場合、アニメーションは1つの方向のみ/単一方向に繰り返し実行されます。 繰り返し回数はrepeatで定義されます。
scale	アニメーション時間を乗じる際の係数
onPlay	アニメーションが開始または続行された場合に実行される反応。パラメータ: 開始時間および再生の方向(trueの場合は順方向、falseの場合は逆方向)
onPause	アニメーションが一時停止した場合に実行される反応。パラメータ: 現在のアニメーション時間。
onTerminate	アニメーションが完了した場合に実行される反応。最初のパラメータ: アニメーション時間。2番目のパラメータ: 終了の理由。次のようにエンコードされています。

プロパティ名	説明
	<ul style="list-style-type: none"> <li>▶ 0: アニメーションは完了します。</li> <li>▶ 1: アニメーションはキャンセルされます。これは、によってトリガーされます。 f:animation_cancel</li> <li>▶ 2: ビュー遷移が原因でウィジェットが破棄されます。</li> <li>▶ 3: アニメーションは最後のステップにジャンプします。これは、によってトリガーされます。f:animation_cancel_end</li> <li>▶ 4: アニメーションは最初のステップにジャンプしてからキャンセルされます。これは、によってトリガーされます。f:animation_cancel_reset</li> </ul>

### 13.12.2.2.1. コンスタント曲線

コンスタント曲線は、定義された遅延時間の経過後にターゲット値を設定します。コンスタント曲線は、整数型、ブール型、浮動小数点数型、および色型に対して使用できます。

表13.129 コンスタント曲線のプロパティ

プロパティ名	説明
enabled	アニメーションを実行するかどうかを定義します。
delay	アニメーションの開始を基準にしたミリ秒単位の遅延
duration	曲線セグメントのミリ秒単位の時間
repeat	繰り返しの回数。0はいつまでも繰り返されることを表します
alternating	アニメーションを繰り返し実行するかどうかを定義します。
relative	初期値に更新を適用するかどうかを定義します。
value	結果のコンスタント値
target	結果値が適用されるターゲットプロパティ

### 13.12.2.2.2. 高速開始曲線

高速開始曲線は、値を定期的に設定します。最初の値は高速ですが、設定のたびに一定のペースで減速します。高速開始曲線は、整数型、ブール型、浮動小数点数型、および色型に対して使用できます。

表13.130 高速開始曲線のプロパティ

プロパティ名	説明
enabled	アニメーションを実行するかどうかを定義します。
delay	アニメーションの開始を基準にしたミリ秒単位の遅延

プロパティ名	説明
duration	曲線セグメントのミリ秒単位の時間
repeat	繰り返しの回数。0はいつまでも繰り返されることを表します
alternating	アニメーションを繰り返し実行するかどうかを定義します。
relative	初期値に更新を適用するかどうかを定義します。
start	初期値
end	最終値
target	結果値が適用されるターゲットプロパティ

### 13.12.2.2.3. 低速開始曲線

低速開始曲線は、値を定期的に設定します。最初の値は低速ですが、設定のたびに一定のペースで加速します。低速開始曲線は、整数型、ブール型、浮動小数点数型、および色型に対して使用できます。

表13.131 低速開始曲線のプロパティ

プロパティ名	説明
enabled	アニメーションを実行するかどうかを定義します。
delay	アニメーションの開始を基準にしたミリ秒単位の遅延
duration	曲線セグメントのミリ秒単位の時間
repeat	繰り返しの回数。0はいつまでも繰り返されることを表します
alternating	アニメーションを繰り返し実行するかどうかを定義します。
relative	初期値に更新を適用するかどうかを定義します。
start	初期値
end	最終値
target	結果値が適用されるターゲットプロパティ

### 13.12.2.2.4. 二次曲線

二次曲線は、二次関数曲線を使って値を定期的に設定します。二次曲線は、整数型、ブール型、浮動小数点数型、および色型に対して使用できます。

表13.132 二次曲線のプロパティ

プロパティ名	説明
enabled	アニメーションを実行するかどうかを定義します。
delay	アニメーションの開始を基準にしたミリ秒単位の遅延

プロパティ名	説明
duration	曲線セグメントのミリ秒単位の時間
repeat	繰り返しの回数。0はいつまでも繰り返されることを表します
alternating	アニメーションを繰り返し実行するかどうかを定義します。
relative	初期値に更新を適用するかどうかを定義します。
acceleration	曲線の加速
velocity	結果を計算するための速度
constant	結果を計算するためのコンスタント値
target	結果値が適用されるターゲットプロパティ

### 13.12.2.2.5. 正弦曲線

正弦曲線は、正弦関数曲線を使って値を定期的に設定します。正弦曲線は、整数型、ブール型、浮動小数点数型、および色型に対して使用できます。

表13.133 正弦曲線のプロパティ

プロパティ名	説明
enabled	アニメーションを実行するかどうかを定義します。
delay	アニメーションの開始を基準にしたミリ秒単位の遅延
duration	曲線セグメントのミリ秒単位の時間
repeat	繰り返しの回数。0はいつまでも繰り返されることを表します
alternating	アニメーションを繰り返し実行するかどうかを定義します。
relative	初期値に更新を適用するかどうかを定義します。
amplitude	正弦曲線の振幅
constant	結果を計算するためのコンスタント値
frequency	ヘルツ単位の曲線の周波数
phase	ラジアン単位の角位変換
target	結果値が適用されるターゲットプロパティ

### 13.12.2.2.6. スクリプト曲線

スクリプト曲線は、EB GUIDEスクリプトによって独自に定義できる曲線です。その他の曲線では使用できないアニメーションや独自のカスタムアニメーションを使用したい場合には、スクリプト曲線を使用します。この曲線が特に役立つのは、カスタマイズされた軌跡をウィジェットの移動で使用したい場合です。スクリプト曲線は、整数型、ブール型、浮動小数点数型、および色型に対して使用できます。

表13.134 スクリプト曲線のプロパティ

プロパティ名	説明
enabled	アニメーションを実行するかどうかを定義します。
delay	アニメーションの開始を基準にしたミリ秒単位の遅延
duration	曲線セグメントのミリ秒単位の時間
repeat	繰り返しの回数。0はいつまでも繰り返されることを表します
alternating	アニメーションを繰り返し実行するかどうかを定義します。
relative	初期値に更新を適用するかどうかを定義します。
curve	EB GUIDEスクリプトで曲線の関数を定義します。次の2つのパラメータを指定します。 ▶ diff: 前回の実行からのミリ秒単位の経過時間です。アニメーションの開始時点ではdiffが0になっています。 ▶ t_anim: アニメーション開始以降のミリ秒単位の経過時間です。
target	結果値が適用されるターゲットプロパティ

### 13.12.2.2.7. リニア曲線

リニア曲線は、リニアプログレッション曲線を使って値を定期的に設定します。リニア曲線は、整数型、ブール型、浮動小数点数型、および色型に対して使用できます。

表13.135 リニア曲線のプロパティ

プロパティ名	説明
enabled	アニメーションを実行するかどうかを定義します。
delay	アニメーションの開始を基準にしたミリ秒単位の遅延
duration	曲線セグメントのミリ秒単位の時間
repeat	繰り返しの回数。0はいつまでも繰り返されることを表します
alternating	アニメーションを繰り返し実行するかどうかを定義します。
relative	初期値に更新を適用するかどうかを定義します。
velocity	結果を計算するための速度
constant	結果を計算するためのコンスタント値
target	結果値が適用されるターゲットプロパティ

### 13.12.2.2.8. リニア補間曲線

リニア補間曲線ウィジェットは、リニア補間曲線を使って値を定期的に設定します。リニア補間曲線は、整数型、ブール型、浮動小数点数型、および色型に対して使用できます。

注記



リニアキー値補間曲線

3Dグラフィックファイルのインポート中に、インポートされる3Dシーンにアニメーションがある場合、リニアキー値補間整数曲線およびリニアキー値補間浮動小数点数曲線が作成されます。これらの曲線の基礎となるキー値ペアは、EB GUIDE Studioで変更できません。

表13.136 リニア補間曲線のプロパティ

プロパティ名	説明
enabled	アニメーションを実行するかどうかを定義します。
delay	アニメーションの開始を基準にしたミリ秒単位の遅延
duration	曲線セグメントのミリ秒単位の時間
repeat	繰り返しの回数。0はいつまでも繰り返されることを表します
alternating	アニメーションを繰り返し実行するかどうかを定義します。
relative	初期値に更新を適用するかどうかを定義します。
target	結果値が適用されるターゲットプロパティ

### 13.12.2.3. コンテナ

コンテナは、複数のウィジェットを子ウィジェットとして格納し、それらをグループ化します。

表13.137 コンテナのプロパティ

プロパティ名	説明
height	ピクセル単位のウィジェットの高さ
width	ピクセル単位のウィジェットの幅
visible	trueである場合、ウィジェットおよび子ウィジェットが表示されます。
x	親ウィジェットを基準にしたウィジェットのX座標
y	親ウィジェットを基準にしたウィジェットのY座標

### 13.12.2.4. 楕円

楕円は、色の付いた楕円をウィジェットの寸法と座標でビューに描画します。このウィジェットは、扇形または弧を描出するために使用することもできます。

表13.138 楕円のプロパティ

プロパティ名	説明
height	ピクセル単位のウィジェットの高さ
width	ピクセル単位のウィジェットの幅

プロパティ名	説明
visible	trueである場合、ウィジェットおよび子ウィジェットが表示されます。
x	親ウィジェットを基準にしたウィジェットのX座標
y	親ウィジェットを基準にしたウィジェットのY座標
fillColor	楕円を塗りつぶす色
arcWidth	楕円の弧の幅
centralAngle	楕円の扇型を定義する角度
sectorRotation	楕円の扇型の回転を指定する角度

### 13.12.2.5. イメージ

イメージは、画像をビューに配置します。

表13.139 イメージのプロパティ

プロパティ名	説明
height	ピクセル単位のウィジェットの高さ
width	ピクセル単位のウィジェットの幅
visible	trueである場合、ウィジェットおよび子ウィジェットが表示されます。
x	親ウィジェットを基準にしたウィジェットのX座標
y	親ウィジェットを基準にしたウィジェットのY座標
image	ウィジェットが表示するイメージ
sRGB	このプロパティが有効になっている場合、imageで選択されているイメージはsRGB色空間を使用してレンダリングされます。  sRGB機能を使用するには、プロジェクトセンターの[設定] > [プロファイル]にあるcolorModeプロパティで、32-bit sRGB (=4)または32-bit sRGB (Emulated) (=5)を選択します。
horizontalAlign	ウィジェットの境界内のイメージファイルの水平方向の位置揃え
verticalAlign	ウィジェットの境界内のイメージファイルの垂直方向の位置揃え

**注記**



サポートされているイメージファイル形式

使用可能なイメージファイル形式は、レンダラーの実装によって異なります。OpenGL ES 2.0以降のレンダラーは、.pngファイルと.jpgファイルをサポートしています。

### 13.12.2.6. インスタンスエータ

インスタンスエータは、ランタイムにウィジェットインスタンスを作成します。インスタンスエータを使用して、動的コンテンツまたは静的コンテンツを持つリストまたは表をモデル化できます。インスタンスエータの子ウィジェットは、ランタイムに作成されるリストまたは表のラインテンプレートとして利用されます。デフォルトでは、インスタンスエータは最初のラインテンプレートのみをインスタンス化します。

表13.140 インスタンスエータのプロパティ

プロパティ名	説明
height	ピクセル単位のウィジェットの高さ
width	ピクセル単位のウィジェットの幅
visible	trueである場合、ウィジェットおよび子ウィジェットが表示されます。
x	親ウィジェットを基準にしたウィジェットのX座標
y	親ウィジェットを基準にしたウィジェットのY座標
numItems	インスタンス化された子ウィジェットの数。numItemsが0の場合、子ウィジェットは作成されません。
lineMapping	子ウィジェットとラインテンプレートのラインの関連付けを定義します。つまり、インスタンス化の順序を定義します。

### 13.12.2.7. ラベル

ラベルは、テキストをビューに配置します。

表13.141 ラベルのプロパティ

プロパティ名	説明
height	ピクセル単位のウィジェットの高さ
width	ピクセル単位のウィジェットの幅
visible	trueである場合、ウィジェットおよび子ウィジェットが表示されます。
x	親ウィジェットを基準にしたウィジェットのX座標
y	親ウィジェットを基準にしたウィジェットのY座標
text	ラベルに表示されるテキスト。テキストは、ウィジェットエリアに収まらない場合、デフォルトでは末尾で切り捨てられます。
textColor	テキストが表示される色
font	テキストが表示されるフォント
horizontalAlign	ラベルの境界内のテキストの水平方向の位置揃え
verticalAlign	ラベルの境界内のテキストの垂直方向の位置揃え

### 13.12.2.8. 四角形



四角形は、色の付いた四角形をウィジェットの寸法と座標でビューに描画します。

表13.142 四角形のプロパティ

プロパティ名	説明
height	ピクセル単位のウィジェットの高さ
width	ピクセル単位のウィジェットの幅
visible	trueである場合、ウィジェットおよび子ウィジェットが表示されます。
x	親ウィジェットを基準にしたウィジェットのx座標
y	親ウィジェットを基準にしたウィジェットのy座標
fillColor	四角形を塗りつぶす色

### 13.12.3. アニメーション

次のセクションでは、[アニメーション]カテゴリ内のウィジェットのプロパティをリストします。

#### 13.12.3.1. アニメーション

アニメーションは、その親ウィジェットに作用します。アニメーションには、子ウィジェットとして最低でも1つの曲線が必要です。

表13.143 アニメーションのプロパティ

プロパティ名	説明
name	アニメーションの名前
alternating	アニメーションを繰り返し実行するかどうかを定義します。
repeat	繰り返しの回数。0は無限数を表します
enabled	アニメーションを実行するかどうかを定義します。
scale	アニメーション時間を乗じる際の係数
onPause	アニメーションが一時停止した場合に実行される反応。パラメータ: 現在のアニメーション時間。
onPlay	アニメーションが開始または続行された場合に実行される反応。パラメータ: 開始時間および再生の方向(trueの場合は順方向、falseの場合は逆方向)
onTerminate	アニメーションが完了した場合に実行される反応。最初のパラメータ: アニメーション時間。2番目のパラメータ: 終了の理由。次のようにエンコードされています。 ▶ 0: アニメーションは完了します。

プロパティ名	説明
	<ul style="list-style-type: none"> <li>▶ 1: アニメーションはキャンセルされます。これは、によってトリガーされます。 f:animation_cancel</li> <li>▶ 2: ビュー遷移が原因でウィジェットが破棄されます。</li> <li>▶ 3: アニメーションは最後のステップにジャンプします。これは、によってトリガーされます。f:animation_cancel_end</li> <li>▶ 4: アニメーションは最初のステップにジャンプしてからキャンセルされます。これは、によってトリガーされます。f:animation_cancel_reset</li> </ul>

### 13.12.3.2. コンスタント曲線

コンスタント曲線は、アニメーションの子ウィジェットです。コンスタント曲線は、定義された遅延時間の経過後にターゲット値を設定します。コンスタント曲線は、整数型、ブール型、浮動小数点数型、および色型に対して使用できません。

表13.144 コンスタント曲線のプロパティ

プロパティ名	説明
name	曲線の名前
delay	アニメーションの開始を基準にしたミリ秒単位の遅延
duration	曲線セグメントのミリ秒単位の時間
enabled	アニメーションを実行するかどうかを定義します。
alternating	アニメーションを繰り返し実行するかどうかを定義します。
relative	初期値に更新を適用するかどうかを定義します。
repeat	繰り返しの回数
target	結果値が適用されるターゲットプロパティ
value	結果のコンスタント値

### 13.12.3.3. 高速開始曲線

高速開始曲線は、アニメーションの子ウィジェットです。高速開始曲線は、値を定期的に設定します。最初の値は高速ですが、設定のたびに一定のペースで減速します。高速開始曲線は、整数型、ブール型、浮動小数点数型、および色型に対して使用できます。

表13.145 高速開始曲線のプロパティ

プロパティ名	説明
name	曲線の名前

プロパティ名	説明
delay	アニメーションの開始を基準にしたミリ秒単位の遅延
duration	曲線セグメントのミリ秒単位の時間
enabled	アニメーションを実行するかどうかを定義します。
alternating	アニメーションを繰り返し実行するかどうかを定義します。
relative	初期値に更新を適用するかどうかを定義します。
repeat	繰り返しの回数
target	結果値が適用されるターゲットプロパティ
start	初期値
end	最終値

#### 13.12.3.4. 低速開始曲線

低速開始曲線は、アニメーションの子ウィジェットです。低速開始曲線は、値を定期的に設定します。最初の値は低速ですが、設定のたびに一定のペースで加速します。低速開始曲線は、整数型、ブール型、浮動小数点数型、および色型に対して使用できます。

表13.146 低速開始曲線のプロパティ

プロパティ名	説明
name	曲線の名前
delay	アニメーションの開始を基準にしたミリ秒単位の遅延
duration	曲線セグメントのミリ秒単位の時間
enabled	アニメーションを実行するかどうかを定義します。
alternating	アニメーションを繰り返し実行するかどうかを定義します。
relative	初期値に更新を適用するかどうかを定義します。
repeat	繰り返しの回数
target	結果値が適用されるターゲットプロパティ
start	初期値
end	最終値

#### 13.12.3.5. 二次曲線

二次曲線は、アニメーションの子ウィジェットです。二次曲線は、二次関数曲線を使って値を定期的に設定します。二次曲線は、整数型、ブール型、浮動小数点数型、および色型に対して使用できます。

表13.147 二次曲線のプロパティ

プロパティ名	説明
name	曲線の名前
delay	アニメーションの開始を基準にしたミリ秒単位の遅延
duration	曲線セグメントのミリ秒単位の時間
enabled	アニメーションを実行するかどうかを定義します。
alternating	アニメーションを繰り返し実行するかどうかを定義します。
relative	初期値に更新を適用するかどうかを定義します。
repeat	繰り返しの回数
target	結果値が適用されるターゲットプロパティ
velocity	結果を計算するための速度
acceleration	曲線の加速
constant	結果を計算するためのコンスタント値

### 13.12.3.6. 正弦曲線

正弦曲線は、アニメーションの子ウィジェットです。正弦曲線は、正弦関数曲線を使って値を定期的に設定します。正弦曲線は、整数型、ブール型、浮動小数点数型、および色型に対して使用できます。

表13.148 正弦曲線のプロパティ

プロパティ名	説明
name	曲線の名前
delay	アニメーションの開始を基準にしたミリ秒単位の遅延
duration	曲線セグメントのミリ秒単位の時間
enabled	アニメーションを実行するかどうかを定義します。
alternating	アニメーションを繰り返し実行するかどうかを定義します。
relative	初期値に更新を適用するかどうかを定義します。
repeat	繰り返しの回数
target	結果値が適用されるターゲットプロパティ
amplitude	正弦曲線の振幅
constant	結果を計算するためのコンスタント値
phase	ラジアン単位の角位変換
frequency	ヘルツ単位の曲線の周波数

### 13.12.3.7. スクリプト曲線

スクリプト曲線は、アニメーションの子ウィジェットです。スクリプト曲線は、EB GUIDEスクリプトで記述された曲線を使って値を設定します。スクリプト曲線は、整数型、ブール型、浮動小数点数型、および色型に対して使用できます。

表13.149 スクリプト曲線のプロパティ

プロパティ名	説明
name	曲線の名前
delay	アニメーションの開始を基準にしたミリ秒単位の遅延
duration	曲線セグメントのミリ秒単位の時間
enabled	アニメーションを実行するかどうかを定義します。
alternating	アニメーションを繰り返し実行するかどうかを定義します。
relative	初期値に更新を適用するかどうかを定義します。
repeat	繰り返しの回数
target	結果値が適用されるターゲットプロパティ
curve	結果の曲線関数

### 13.12.3.8. リニア曲線

リニア曲線は、アニメーションの子ウィジェットです。リニア曲線は、リニアプログレッション曲線を使って値を定期的を設定します。リニア曲線は、整数型、ブール型、浮動小数点数型、および色型に対して使用できます。

表13.150 リニア曲線のプロパティ

プロパティ名	説明
name	曲線の名前
delay	アニメーションの開始を基準にしたミリ秒単位の遅延
duration	曲線セグメントのミリ秒単位の時間
enabled	アニメーションを実行するかどうかを定義します。
alternating	アニメーションを繰り返し実行するかどうかを定義します。
relative	初期値に更新を適用するかどうかを定義します。
repeat	繰り返しの回数
target	結果値が適用されるターゲットプロパティ
velocity	結果を計算するための速度

### 13.12.3.9. リニア補間曲線

リニア補間曲線は、アニメーションの子ウィジェットです。リニア補間曲線ウィジェットは、リニア補間曲線を使って値を定期的に設定します。リニア補間曲線は、整数型、ブール型、浮動小数点数型、および色型に対して使用できます。

注記



リニアキー値補間曲線

3Dグラフィックファイルのインポート中に、インポートされる3Dシーンにアニメーションがある場合、リニアキー値補間整数曲線およびリニアキー値補間浮動小数点数曲線が作成されます。これらの曲線の基礎となるキー値ペアは、EB GUIDE Studioで変更できません。

表13.151 リニア補間曲線のプロパティ

プロパティ名	説明
name	曲線の名前
delay	アニメーションの開始を基準にしたミリ秒単位の遅延
duration	曲線セグメントのミリ秒単位の時間
enabled	アニメーションを実行するかどうかを定義します。
alternating	アニメーションを繰り返し実行するかどうかを定義します。
relative	初期値に更新を適用するかどうかを定義します。
repeat	繰り返しの回数
target	結果値が適用されるターゲットプロパティ
start	初期値
end	最終値

## 13.12.4. 3Dウィジェット

### 13.12.4.1. 環境光

環境光とは、シーンを均一に照らす光です。環境光は、材質、PBR GGX材質、およびPBR Phong材質のambient色プロパティに影響します。

表13.152 環境光のプロパティ

プロパティ名	説明
enabled	trueである場合、ウィジェットが有効になります。
color	光の色です。
intensity	光の強度です。下限値である0.0は、環境光がないことを示します。

### 13.12.4.2. カメラ

カメラは、特定の視点からのシーンのビューを定義します。複数のカメラを使用すると、複数の視点からシーンを表示できます。

表13.153 カメラのプロパティ

プロパティ名	説明
enabled	trueである場合、ウィジェットが有効になります。
nearPlane	視線方向からシーンが見え始める、カメラからの最短距離。測定単位は、サードパーティの3Dモデリングソフトウェアで3Dモデルを作成する際に定義されます。
farPlane	視線方向からシーンが消えずにいる、カメラからの最長距離。測定単位は、サードパーティの3Dモデリングソフトウェアで3Dモデルを作成する際に定義されます。
fieldOfView	カメラの垂直方向の視野角度。最大値は180
projectionType	カメラの投影タイプを定義します。オブジェクトは、perspective (=0) または orthographic (=1) のどちらかの投影法でレンダリングされます。  投影タイプが直交の場合、ビューボリュームはfieldOfView角を使用して計算されます。

### 13.12.4.3. 指向性ライト

指向性ライトは、一方向からシーンを照らします。

表13.154 指向性ライトのプロパティ

プロパティ名	説明
enabled	trueである場合、ウィジェットが有効になります。
color	ライトの色
intensity	ライトの強度です。下限値である0.0は、指向性ライトがないことを示します

### 13.12.4.4. イメージベースドライフト

イメージベースドライティングは、.pfmまたは.hdrファイルに保存された現実世界のライティング情報に基づいてシーンを照らすライトです。.pfmまたは.hdrファイルは、.ebiblファイルを作成する際にIBLGeneratorの入力データとして利用されます。

表13.155 イメージベースドライフトのプロパティ

プロパティ名	説明
enabled	trueである場合、ウィジェットが有効になります。

プロパティ名	説明
ibl	手動で作成されたIBLファイル.ebibl
intensity	ライトの強度です。0.0はイメージベースライトがないことを示します。

### 13.12.4.5. 材質

材質は、Phong反射モデルを使用してメッシュ表面の外観を定義します。

表13.156 材質のプロパティ

プロパティ名	説明
ambient	環境光で照らされている場合にオブジェクトが反射する色親シーングラフに環境光が追加されない場合、このプロパティに効力はありません。
diffuse	白色光で照らされている場合にオブジェクトが全方向に均一に反射する色。[ディフューズテクスチャ]ウィジェット機能が追加されている場合、このプロパティは無効です。
emissive	オブジェクトの自己発光色。[エミッシブテクスチャ]ウィジェット機能が追加されている場合、このプロパティは無効です。
shininess	光沢要素 有効なのは0.0と1.0の間の値(0.3など)のみであるという点に注意してください。 [光沢テクスチャ]ウィジェット機能が使用されている場合、shininessプロパティは無視されます。
specular	表面に光沢があるオブジェクトが反射する色。[スペキュラテクスチャ]ウィジェット機能が追加されている場合、またはshininessプロパティが0.0に設定されている場合、specularプロパティは無効です。
opacity	オパシティ値 有効なのは0.0と1.0の間の値(0.3など)のみであるという点に注意してください。

### 13.12.4.6. メッシュ

メッシュは3Dオブジェクトの形状を定義します。

表13.157 メッシュのプロパティ

プロパティ名	説明
visible	trueである場合、ウィジェットおよび子ウィジェットが表示されます。



プロパティ名	説明
mesh	自動的に作成されるメッシュファイル *.ebmesh
culling	メッシュから三角を集めない(0)か、表向きの三角のみを集める(1)か、裏向きの三角のみを集める(2)かを定義します。

### 13.12.4.7. PBR GGX材質

PBR GGX材質は、物理的に正しいCook-Torranceモデルを使用してメッシュ表面の外観を定義します。

表13.158 PBR GGX材質のプロパティ

プロパティ名	説明
ambient	環境光で照らされている場合にオブジェクトが反射する色[アンビエントテクスチャ]ウィジェット機能が追加されている場合、このプロパティは無効です。
diffuse	白色光で照らされている場合にオブジェクトが全方向に均一に反射する色。[ディフューズテクスチャ]ウィジェット機能が追加されている場合、このプロパティは無効です。
emissive	オブジェクトの自己発光色。[エミッシブテクスチャ]ウィジェット機能が追加されている場合、このプロパティは無効です。
specular	表面に光沢があるオブジェクトが反射する色。[スペキュラテクスチャ]ウィジェット機能が追加されている場合、またはshininessプロパティが0.0に設定されている場合、specularプロパティは無効です。
metallic	表面の質感を金属のようにする値 この値は、ディフューズの影響とスペキュラの影響の間を補間します。 0と1の間の値(0.3など)のみが有効です。
roughness	表面の質感をざらざらにする値 この値は、表面の微細構造を制御します。 0と1の間の値(0.3など)のみが有効です。
opacity	オパシティ値 0と1の間の値(0.3など)のみが有効です。

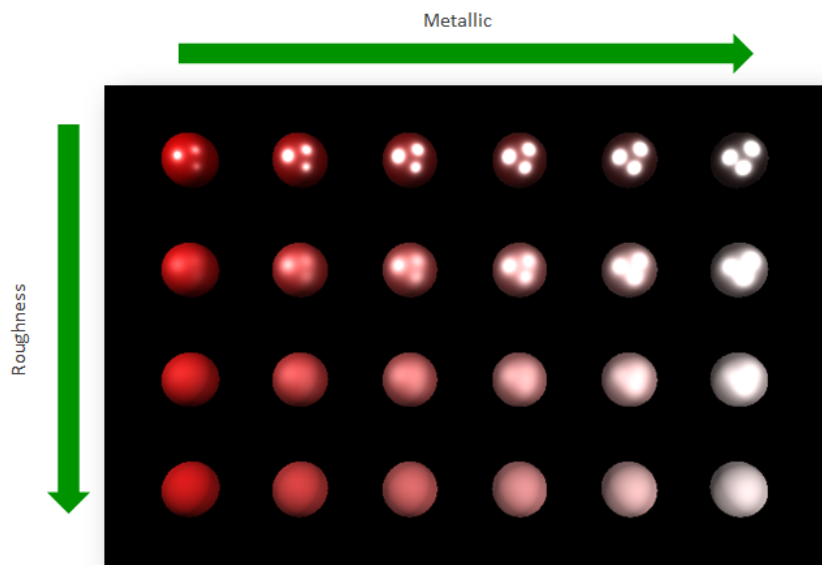


図13.1 物理ベース材質のサンプル

### 13.12.4.8. PBR Phong材質

PBR Phong材質は、物理的に正しいPhong反射モデルを使用してメッシュ表面の外観を定義します。

表13.159 PBR Phong材質のプロパティ

プロパティ名	説明
ambient	環境光で照らされている場合にオブジェクトが反射する色[アンビエントテクスチャ]ウィジェット機能が追加されている場合、このプロパティは無効です。
diffuse	白色光で照らされている場合にオブジェクトが全方向に均一に反射する色。[ディフューズテクスチャ]ウィジェット機能が追加されている場合、このプロパティは無効です。
emissive	オブジェクトの自己発光色。[エミッシブテクスチャ]ウィジェット機能が追加されている場合、このプロパティは無効です。
shininess	光沢要素
specular	表面に光沢があるオブジェクトが反射する色。[スペキュラテクスチャ]ウィジェット機能が追加されている場合、またはshininessプロパティが0.0に設定されている場合、specularプロパティは無効です。
metallic	表面の質感を金属のようにする値  この値は、ディフューズの影響とスペキュラの影響の間を補間します。  0と1の間の値(0.3など)のみが有効です。

プロパティ名	説明
opacity	オパシティ値  0と1の間の値(0.3など)のみが有効です。

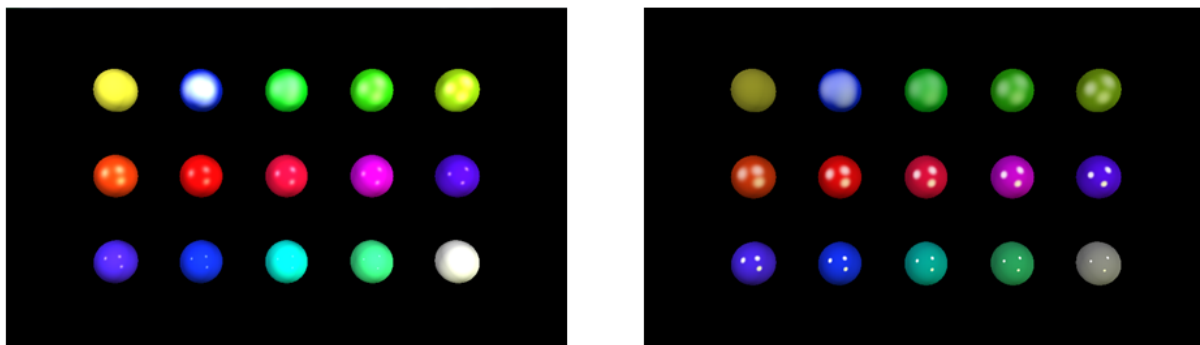


図13.2 正規化されていない材質(左)と正規化された材質(右)のサンプル

### 13.12.4.9. 点ライト

点ライトは、電球のように全方向に光を放つライトをシーンに追加します。

表13.160 点ライトのプロパティ

プロパティ名	説明
enabled	trueである場合、ウィジェットが有効になります。
color	ライトの色
intensity	ライトの強度です。下限値0.0は点ライトがないことを示し、上限値は減衰係数によって異なります。
attenuationConstant	距離が離れるにつれライトの強度を減退させる定数係数。値が0.0の場合、この係数は使用されません。
attenuationLinear	距離が離れるにつれライトの強度を減退させる線形係数。値が0.0の場合、この係数は使用されません。
attenuationQuadratic	距離が離れるにつれライトの強度を減退させる二次係数。値が0.0の場合、この係数は使用されません。

### 13.12.4.10. シーングラフ

シーングラフは、3Dオブジェクトをビュー内に配置します。

表13.161 シーングラフのプロパティ

プロパティ名	説明
visible	trueである場合、ウィジェットおよび子ウィジェットが表示されます。
width	ピクセル単位のウィジェットの幅
height	ピクセル単位のウィジェットの高さ
x	親ウィジェットを基準にしたウィジェットのX座標
y	親ウィジェットを基準にしたウィジェットのY座標
gamma	シーングラフの輝度出力を補正します。デフォルト値は2.2に設定されています。

### 13.12.4.11. シーングラフノード

シーングラフノードは子ノードであり、シーングラフや別のシーングラフノードに追加されます。シーングラフノードは、変形プロパティを持つ3Dウィジェットを3Dシーンに配置するために使用します。次の3Dウィジェットをシーングラフノードに追加できます。

- ▶ カメラ
- ▶ 指向性ライト
- ▶ イメージベースドライト
- ▶ メッシュ
- ▶ 点ライト
- ▶ スポットライト

表13.162 シーングラフノードのプロパティ

プロパティ名	説明
visible	trueである場合、ウィジェットおよび子ウィジェットが表示されます。
rotationX	X軸での回転
rotationY	Y軸での回転
rotationZ	Z軸での回転
scalingX	X軸方向の拡大縮小
scalingY	Y軸方向の拡大縮小
scalingZ	Z軸方向の拡大縮小
translationX	X軸での変換
translationY	Y軸での変換
translationZ	Z軸での変換

### 13.12.4.12. スポットライト

スポットライトは、光の影響範囲を円錐状に制限したライトを追加します。

表13.163 スポットライトのプロパティ

プロパティ名	説明
enabled	trueである場合、ウィジェットが有効になります。
color	ライトの色
intensity	ライトの強度です。下限値0.0はスポットライトがないことを示し、上限値は減衰係数によって異なります。
attenuationConstant	距離が離れるにつれライトの強度を減退させる定数係数
attenuationLinear	距離が離れるにつれライトの強度を減退させる線形係数
attenuationQuadratic	距離が離れるにつれライトの強度を減退させる二次係数
coneAngleInner	ライトの円錐の内側の角度(度単位)。最大値は180
coneAngleOuter	ライトの円錐の外側の角度(度単位)。最大値は180

## 13.13. ウィジェット機能

次のリストには、実装されているすべてのウィジェット機能の説明と、これらをEB GUIDEモデルで使用方法に関する簡単な説明が含まれます。

### 13.13.1. 共通

#### 13.13.1.1. 子の可視性の選択

[子の可視性の選択]ウィジェット機能は、子ウィジェットの可視性を処理します。表示する1つのウィジェットを定義したり、同時に表示する子ウィジェットのグループを定義したりできます。グループを定義するには、子ウィジェットのインデックスを同じグループ値にマッピングします。

表13.164 [子の可視性の選択]ウィジェット機能のプロパティ

プロパティ名	説明
containerIndex	子ウィジェットの可視性を制御します。  containerMappingが設定されていない場合、containerIndexによって1つの子ウィジェットが表示されます。表示される子ウィジェットは、ウィジェットツリーにおけ

プロパティ名	説明
	<p>るそのウィジェットの順序によって識別されます。最上位の子のcontainerIndexが0、次のcontainerIndexが1というように続きます。</p> <p>containerMappingが設定されている場合、containerIndexは子ウィジェットのグループを参照します。containerMappingでグループを定義します。</p>
containerMapping	<p>子ウィジェットのグループを作成するには、このプロパティを使用します。[インデックス]列は子ウィジェットを識別します。[値]列はグループを定義します。</p> <p>行の数は、子ウィジェットの数と一致している必要があります。そうでない場合、マッピングは使用されません。</p>

### 13.13.1.2. 有効

[有効]ウィジェット機能は、ウィジェットにenabledプロパティを追加します。

表 13.165 [有効]ウィジェット機能のプロパティ

プロパティ名	説明
enabled	trueである場合、ウィジェットはタッチ入力および押下入力に反応します。

### 13.13.1.3. フォーカス

[フォーカス]ウィジェット機能は、ウィジェットに入力フォーカスを設定できるようにします。

表 13.166 [フォーカス]ウィジェット機能のプロパティ

プロパティ名	説明
focusable	<p>ウィジェットにフォーカスするかどうかを定義します。使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ not focusable (=0)</li> <li>▶ only by touch (=1)</li> <li>▶ only by key (=2)</li> <li>▶ focusable (=3)</li> </ul>
focused	trueである場合、ウィジェットがフォーカスされます。

### 13.13.1.4. 複数行

[複数行]ウィジェット機能を使用すると、改行が可能になります。

制限:

- ▶ [複数行]ウィジェット機能は、ラベルウィジェットにのみ使用できます。

表13.167 [複数行]ウィジェット機能のプロパティ

プロパティ名	説明
lineGap	行間のサイズ。負の値の場合は行間が狭くなり、正の値の場合は行間が広くなります。  line gapが小さ過ぎる(大きい負の値)場合、効果はなくなり、テキストは1行でレンダリングされます。これは、例えば、フォントスタイルがPT_Sans_Narrow、サイズが30に設定されていて、line gapが-50と定義されている場合に発生します。
maxLineCount	表示行の最大数。0 = 制限なし

#### ティップ



使用されている行数

スクリプト関数getLineCountを使用すると、テキストの行数を取得できます。

詳しくは、[13.4.3.2.28「getLineCount」](#)をご覧ください。

#### 注記



文字置換

'\ '¥'のシーケンスは'¥¥'で置き換えられます。'¥' 'n'のシーケンスは'¥n'で置き換えられます。

ラベルのサイズを大きくして1行でテキストを十分に表示できるようにすると、'¥n'が' 'で置き換えられます。

### 13.13.1.5. 押下

[押下]ウィジェット機能は、ウィジェットが押下可能かどうかを定義します。

制限:

- ▶ [押下]ウィジェット機能を追加すると、[フォーカス]ウィジェット機能が自動的に追加されます。

表13.168 [押下]ウィジェット機能のプロパティ

プロパティ名	説明
pressed	trueである場合、ウィジェットにフォーカスがあるときにキーを押せます。

[タッチ]ウィジェット機能を[タッチ押下]ウィジェット機能に組み合わせると、押ボタンをモデル化できます。

### 13.13.1.6. 選択

[選択]ウィジェット機能は、ウィジェットにselectedプロパティを追加します。これは通常、アプリケーションまたはヒューマンマシンインターフェースモデラーによって設定されます。フレームワークの他のコンポーネントによって変更されることはありません。

表 13.169 [選択]ウィジェット機能のプロパティ

プロパティ名	説明
selected	trueである場合、ウィジェットが選択されます。

### 13.13.1.7. 選択グループ

[選択グループ]ウィジェット機能は、オプションボタンのリストをモデル化するために使用されます。リスト内では、すべてのオプションボタンが[選択グループ]ウィジェット機能と一意のボタンIDを持ちます。

buttonValueプロパティにはデータプールアイテムを使用します。オプションボタン内のすべてのウィジェットにデータプールアイテムを割り当てます。

ボタングループ内のウィジェットの選択および選択解除を行うには、buttonValueプロパティを設定するアプリケーションを使用します。タッチまたはキー入力や、ボタン値を設定する条件の追加で変更をトリガーすることもできます。

制限:

- ▶ [選択グループ]ウィジェット機能を追加すると、[選択]ウィジェット機能が自動的に追加されます。

表 13.170 [選択グループ]ウィジェット機能のプロパティ

プロパティ名	説明
buttonId	ボタングループ内のボタンを識別するID
buttonValue	ボタンの現在の値。この値がbuttonIdと一致する場合、ボタンが選択されます。
selected	buttonIDとbuttonValueが同じであるかどうかを評価します。trueである場合、ボタンが選択されます。

### 13.13.1.8. スピン

[スピン]ウィジェット機能は、ウィジェットを回転ボタンに変換します。[スピン]ウィジェット機能を持つウィジェットは、内部値を変更することによって増加イベントおよび減少イベントに反応します。[スピン]ウィジェット機能は、プレビュー値を持つスケール、プログレスバー、またはウィジェットの作成に使用できます。

表 13.171 [スピン]ウィジェット機能のプロパティ

プロパティ名	説明
currentValue	現在の回転値
maxValue	currentValueプロパティの最大値
minValue	currentValueプロパティの最小値
incValueTrigger	trueである場合、currentValueプロパティが1増加します
incValueReaction	currentValueプロパティの増加に対する反応
decValueTrigger	trueである場合、現在の値が1減少します。



プロパティ名	説明
decValueReaction	currentValueプロパティの減少に対する反応
steps	currentValueプロパティの増加または減少を計算するステップの数
valueWrapAround	<p>使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ true: minValueまたはmaxValueを超えた場合、currentValueプロパティは逆側に進みます。</li> <li>▶ false: minValueまたはmaxValueを超えた場合、currentValueプロパティは増加/減少しません。</li> </ul>

### 13.13.1.9. テキストの切り捨て

[テキストの切り捨て]ウィジェット機能は、textプロパティのコンテンツがウィジェットエリアに収まらない場合にそのコンテンツを切り捨てます。ウィジェット機能を使用すると、デフォルト設定のtrailingとは異なる切り捨てが可能です。

制限:

- ▶ [テキストの切り捨て]ウィジェット機能は、ラベルウィジェットにのみ使用できます。

表 13.172 [テキストの切り捨て]ウィジェット機能のプロパティ

プロパティ名	説明
truncationPolicy	<p>1行のテキストの場合、truncationPolicyプロパティは切り捨ての位置を定義します。使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ leading (=0): テキストはテキストの先頭で置き換えられます。</li> <li>▶ trailing (=1): テキストはテキストの末尾で置き換えられます。</li> </ul> <p>複数行のテキストの場合、truncationPolicyプロパティは、テキストを置き換える位置を定義します。使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ leading (=0): 先頭にある行が置き換えられ、最初に表示される行のテキストがテキストの先頭で切り捨てられます。</li> <li>▶ trailing (=1) 末尾にある行が置き換えられ、最後に表示される行のテキストがテキストの末尾で切り捨てられます。</li> </ul>
truncationSymbol	置き換えられたテキスト部分の代わりに表示される文字列

### 13.13.1.10. タッチ

[タッチ]ウィジェット機能は、ウィジェットがタッチ入力に反応できるようにします。

表13.173 [タッチ]ウィジェット機能のプロパティ

プロパティ名	説明
touchable	trueである場合、ウィジェットはタッチ入力に反応します。
touched	trueである場合、ウィジェットは現在タッチされています。
touchPolicy	<p>ウィジェットの境界を超えるタッチおよび動きを処理する方法を定義します。使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ Press then react (=0): 最初に押下すると、ウィジェットが反応します。移動および解放の通知はウィジェットエリア内でのみアクティブです。</li> <li>▶ Press and grab (=1): 押すとコンタクトをグラブできます。コンタクトは、ウィジェットエリアの外部に移動しても、グラブされたままです。</li> <li>▶ Press then react on contact (=2): コンタクトがウィジェットの境界の外部で押された状態になっても、それ以降の移動イベントおよび解放イベントはウィジェットに伝達されます。</li> </ul>
touchBehavior	<p>タッチ評価を定義します。使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ Whole area (=0): タッチされたウィジェットを識別するために、レンダラーはウィジェットのクリッピング四角形を評価します。</li> <li>▶ Visible pixels (=1): タッチされたウィジェットを識別するために、レンダラーはタッチされたピクセルが属するウィジェットを評価します。</li> </ul> <p>アルファ透過性を持つ透過ピクセル、またはOまたはAなどの文字内のピクセルにはタッチできません。</p> <p>Visible pixels値はラベルに対して無効であるという点に注意してください。</p>

[タッチ]ウィジェット機能を[押下]ウィジェット機能に組み合わせると、押ボタンをモデル化できます。

#### ティップ



#### パフォーマンスに関する推奨事項

プロジェクトにおいてパフォーマンスが重要な問題である場合、touchBehaviorプロパティにWhole area (=0)を設定します。EB GUIDE GTFでは、Visible pixels (=1)より速くWhole area (=0)が評価されます。

## 13.13.2. 効果

### 13.13.2.1. 枠

[枠]ウィジェット機能は、設定可能な枠をウィジェットに追加します。枠はウィジェットの境界から始まり、ウィジェット内に配置されます。

制限:

- ▶ このウィジェット機能は、四角形に対して使用できます。

表13.174 [枠]ウィジェット機能のプロパティ

プロパティ名	説明
borderThickness	ピクセル単位の枠の厚さ
borderColor	枠のレンダリングに使用する色
borderStyle	枠のレンダリングに使用するスタイル

### 13.13.2.2. 配色

[配色]ウィジェット機能は、ウィジェットおよびウィジェットサブツリーに色を付けます。また、アルファ値が不透明でない場合は透過性にも影響します。



#### 例13.4 配色ウィジェット機能の使用方法

RGBA要素が0.0~1.0であるすべての色について、[配色]ウィジェット機能のアルゴリズムが、ウィジェットの現在の色値にcolorationColorプロパティ値を掛けます。掛け算はピクセル単位で要素に対して行われます。

半透明のグレーに不透明な青を掛けると、次のように半透明の暗い青になります。

$$(0.5, 0.5, 0.5, 0.5) * (0.0, 0.0, 1.0, 1.0) = (0.0, 0.0, 0.5, 0.5)$$

表13.175 [配色]ウィジェット機能のプロパティ

プロパティ名	説明
colorationEnabled	trueである場合、配色が使用されます。
colorationColor	使用する配色使用可能な値: <ul style="list-style-type: none"> <li>▶ 純粋</li> <li>▶ 不透明</li> <li>▶ 白</li> </ul>

### 13.13.3. フォーカス

[フォーカス]ウィジェット機能カテゴリは、フォーカス管理に関連するウィジェット機能を提供します。

### 13.13.3.1. 自動フォーカス

[自動フォーカス]ウィジェット機能を使用すると、子ウィジェットがフォーカスされる順序が事前定義済みになります。[自動フォーカス]ウィジェット機能は、`focusable`プロパティを持つ子ウィジェットのウィジェットサブツリーをチェックします。

フォーカスの順序の計算には、レイアウト内のウィジェットの順序が使用されます。レイアウトの方向に応じて、アルゴリズムは左上または右上から開始されます。

制限:

- ▶ [自動フォーカス]ウィジェット機能では、[フォーカス]ウィジェット機能が自動的に追加されます。

表 13.176 [自動フォーカス]ウィジェット機能のプロパティ

プロパティ名	説明
<code>focusNext</code>	フォーカスインデックスが増加する条件
<code>focusPrev</code>	フォーカスインデックスが減少する条件
<code>focusFlow</code>	フォーカスの動作は階層内で変化します。使用可能な値: <ul style="list-style-type: none"> <li>▶ <code>stop at hierarchy (=0)</code></li> <li>▶ <code>wrap within hierarchy level (=1)</code></li> <li>▶ <code>step up in hierarchy (=2)</code></li> </ul>
<code>focusedIndex</code>	フォーカス可能な <i>n</i> 番目の子ウィジェットとして現在フォーカスされている子ウィジェットのインデックス
<code>initFocus</code>	このインデックスは、初期化時にフォーカスされる子ウィジェットを定義します。ウィジェットがフォーカス不可能である場合、次にフォーカス可能な子が使用されます。

### 13.13.3.2. ユーザー定義フォーカス

[ユーザー定義フォーカス]ウィジェット機能を使用すると、ウィジェットにフォーカス機能を追加できます。この機能を使用するウィジェットは、ウィジェットサブツリーのローカルフォーカス階層を管理します。

制限:

- ▶ [ユーザー定義フォーカス]ウィジェット機能では、[フォーカス]ウィジェット機能が自動的に追加されます。

表 13.177 [ユーザー定義フォーカス]ウィジェット機能のプロパティ

プロパティ名	説明
<code>focusNext</code>	フォーカスを次の子ウィジェットに割り当てるトリガー
<code>focusOrder</code>	<code>focusOrder</code> プロパティを使用すると、フォーカスを割り当てる際に子ウィジェットをスキップできます。子ウィジェットのIDは、サブツリー内の位置に対応しています。フォー

プロパティ名	説明
	<p>カスタム可能な子ウィジェットはデフォルトでスキップされます。子ウィジェットがフォーカスされる順序は、次のとおりです。</p> <ul style="list-style-type: none"> <li>▶ 定義済み: ユーザー定義のウィジェット順序を使用します。</li> <li>▶ 未定義: デフォルトのウィジェット順序を代わりに使用します。</li> </ul> <p>各子ウィジェットには[フォーカス]ウィジェット機能が必要です。この機能がない場合、ウィジェットはフォーカス管理で無視されます。例: focusOrder=1 0 2は、まず2番目のウィジェットにフォーカスし、次に1番目のウィジェット、最後に3番目のウィジェットにフォーカスすることを意味します。</p>
focusPrev	フォーカスを前の子に割り当てるトリガー
focusFlow	<p>フォーカスの動作は階層内で変化します。使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ stop at hierarchy level (=0)</li> <li>▶ wrap within hierarchy level (=1)</li> <li>▶ step up in hierarchy (=2)</li> </ul>
focusedIndex	インデックスは、focusOrderリスト内の子ウィジェットの位置を定義します。ウィジェットがフォーカス不可能である場合、リスト内の次の子が使用されます。
initFocus	初期化時にフォーカスされる子ウィジェットのインデックス

## 13.13.4. ジェスチャー

### 13.13.4.1. フリックジェスチャー

表面を素早くなでるコンタクト

制限:

- ▶ [フリックジェスチャー]ウィジェット機能を追加すると、[ジェスチャー]および[タッチ]ウィジェット機能が自動的に追加されます。

表13.178 [フリックジェスチャー]ウィジェット機能のプロパティ

プロパティ名	説明
onGestureFlick	<p>ジェスチャーが認識されたときにトリガーされる反応</p> <p>反応引数は、次のとおりです。</p> <ul style="list-style-type: none"> <li>▶ speed: フリックジェスチャーの相対速度</li> </ul>

プロパティ名	説明
	<p>ピクセル/ミリ秒の速度をflickMinLength/flickMaxTimeで割った値</p> <ul style="list-style-type: none"> <li>▶ directionX: ジェスチャーの方向ベクトルのX部分</li> <li>▶ directionY: ジェスチャーの方向ベクトルのY部分</li> </ul>
flickMaxTime	ジェスチャーがフリックジェスチャーとして認識されるためにコンタクトが所定の位置に留まることが認められるミリ秒単位の最大時間
flickMinLength	コンタクトがフリックジェスチャーとして認識されるために表面上で移動する必要があるピクセル単位の最小距離

### 13.13.4.2. ホールドジェスチャー

動きのないホールドジェスチャー

制限:

- ▶ [ホールドジェスチャー]ウィジェット機能を追加すると、[ジェスチャー]および[タッチ]ウィジェット機能が自動的に追加されます。
- ▶ [ホールドジェスチャー]ウィジェット機能は、[タッチの喪失]ウィジェット機能をトリガーしません。

表13.179 [ホールドジェスチャー]ウィジェット機能のプロパティ

プロパティ名	説明
onGestureHold	<p>ジェスチャーが認識されたときにトリガーされる反応反応はコンタクトごとに1回のみトリガーされます。つまり、holdDurationが期限切れになったときに、コンタクトがまだ最初のタッチ位置の周囲の小さな境界ボックス内にある場合です。</p> <p>反応引数は、次のとおりです。</p> <ul style="list-style-type: none"> <li>▶ x: コンタクト位置のX座標</li> <li>▶ y: コンタクト位置のY座標</li> </ul>
holdDuration	ジェスチャーがホールドジェスチャーとして認識されるためにコンタクトが所定の位置に留まる必要があるミリ秒単位の最小時間

### 13.13.4.3. ロングホールドジェスチャー

動きのないロングホールドジェスチャー

制限:

- ▶ [ロングホールドジェスチャー]ウィジェット機能を追加すると、[ジェスチャー]および[タッチ]ウィジェット機能が自動的に追加されます。
- ▶ [ロングホールドジェスチャー]ウィジェット機能は、[タッチの喪失]ウィジェット機能をトリガーしません。

表13.180 [ロングホールドジェスチャー]ウィジェット機能のプロパティ

プロパティ名	説明
onGestureLongHold	ジェスチャーが認識されたときにトリガーされる反応はコンタクトごとに1回のみトリガーされます。つまり、longHoldDurationが期限切れになったときに、コンタクトがまだ最初のタッチ位置の周囲の小さな境界ボックス内にある場合です。  反応引数は、次のとおりです。 <ul style="list-style-type: none"> <li>▶ x: コンタクト位置のx座標</li> <li>▶ y: コンタクト位置のy座標</li> </ul>
longHoldDuration	ジェスチャーがロングホールドジェスチャーとして認識されるためにコンタクトが所定の位置に留まる必要があるミリ秒単位の最小時間

#### 13.13.4.4. パスジェスチャー

1つのコンタクトが描画した形状を、既知の形状集合とマッチングします。

制限:

- ▶ [パスジェスチャー]ウィジェット機能を追加すると、[ジェスチャー]および[タッチ]ウィジェット機能が自動的に追加されます。

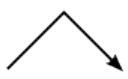
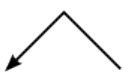
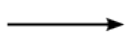
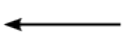



表13.181 [パスジェスチャー]ウィジェット機能のプロパティ

プロパティ名	説明
onPath	入力した形状が一致した場合にトリガーされる反応がトリガーされるのは、onPathStartがすでにトリガーされている場合のみです。反応引数は、次のとおりです。 <ul style="list-style-type: none"> <li>▶ gestureId: マッチングされたパスのID</li> </ul>
onPathStart	コンタクトが最小ボックス(pathMinXBox、pathMinYBox)の外に移動したときにトリガーされる反応。
onPathNotRecognized	入力した形状が一致しない場合にトリガーされる反応がトリガーされるのは、onPathStartがすでにトリガーされている場合のみです。
pathMinXBox	パスジェスチャー認識機能が入力を始めるためにコンタクトが移動する必要があるピクセル単位の最小距離のx座標
pathMinYBox	パスジェスチャー認識機能が入力を始めるためにコンタクトが移動する必要があるピクセル単位の最小距離のy座標

### 13.13.4.4.1. ジェスチャーID

ジェスチャー識別子は、パスジェスチャー認識機能の設定によって決まります。次の表は、EB GUIDEに含まれるサンプル設定を示しています。

表13.182 EB GUIDEに含まれるパスジェスチャーの設定の例

ID	形状	説明
0		左から右への屋根形状
1		右から左への屋根形状
2		左から右への水平線
3		右から左への水平線
4		チェックマーク
5		左から右への波形状
6		右から左への波形状



### 13.13.4.5. ピンチジェスチャー

2つのコンタクトが近づくまたは離れる動き

制限:

- ▶ [ピンチジェスチャー]ウィジェット機能を追加すると、[ジェスチャー]および[タッチ]ウィジェット機能が自動的に追加されます。

表13.183 [ピンチジェスチャー]ウィジェット機能のプロパティ

プロパティ名	説明
onGesturePinchStart	ジェスチャーの開始が認識されたときにトリガーされる反応。反応引数は、次のとおりです。 <ul style="list-style-type: none"> <li>▶ ratio: 最初のコンタクトの距離に対する現在のコンタクトの距離の比率</li> <li>▶ centerX: 2つのコンタクト間の現在の中心点のX座標</li> <li>▶ centerY: 2つのコンタクト間の現在の中心点のY座標</li> </ul>
onGesturePinchUpdate	ピンチ率または中心点が変更されたときにトリガーされる反応。反応引数は、次のとおりです。 <ul style="list-style-type: none"> <li>▶ ratio: 最初のコンタクトの距離に対する現在のコンタクトの距離の比率</li> <li>▶ centerX: 2つのコンタクト間の現在の中心点のX座標</li> <li>▶ centerY: 2つのコンタクト間の現在の中心点のY座標</li> </ul>
onGesturePinchEnd	ジェスチャーが終了したときにトリガーされる反応。反応引数は、次のとおりです。 <ul style="list-style-type: none"> <li>▶ ratio: 最初のコンタクトの距離に対する現在のコンタクトの距離の比率</li> <li>▶ centerX: 2つのコンタクト間の現在の中心点のX座標</li> <li>▶ centerY: 2つのコンタクト間の現在の中心点のY座標</li> </ul>
pinchThreshold	ジェスチャーが認識されるために各コンタクトが最初の位置から移動する必要があるピクセル単位の最小距離

### 13.13.4.6. 回転ジェスチャー

2つのコンタクトの円に沿った動き

制限:

- ▶ [回転ジェスチャー]ウィジェット機能を追加すると、[ジェスチャー]および[タッチ]ウィジェット機能が自動的に追加されます。

表13.184 [回転ジェスチャー]ウィジェット機能のプロパティ

プロパティ名	説明
onGestureRotateStart	ジェスチャーの開始が認識されたときにトリガーされる反応
onGestureRotateUpdate	認識された角度または中心点に変更されたときにトリガーされる反応
onGestureRotateEnd	ジェスチャーが終了したときにトリガーされる反応
rotateThreshold	ジェスチャーの開始が認識されるために各コンタクトが最初の位置から移動する必要があるピクセル単位の最小距離

onGestureRotateEnd、onGestureRotateStart、およびonGestureRotateUpdateの反応引数:

- ▶ angle: 2つの関連コンタクトの最初の位置によって指定される線と、2つのコンタクトの現在の位置によって指定される線の間の角度。角度は反時計回りに測定されます。
- ▶ centerX: 2つのコンタクト間の現在の中心点のX座標
- ▶ centerY: 2つのコンタクト間の現在の中心点のY座標

## 13.13.5. 入力処理

### 13.13.5.1. ジェスチャー

[ジェスチャー]ウィジェット機能を使用すると、ウィジェットがタッチジェスチャーに反応できるようになります。

制限:

- ▶ [ジェスチャー]ウィジェット機能を追加すると、[タッチ]ウィジェット機能が自動的に追加されます。
- ▶ [ジェスチャー]ウィジェット機能には、追加プロパティはありません。

### 13.13.5.2. キー押下

[キー押下]ウィジェット機能を使用すると、ウィジェットがキー押下に反応できるようになります。

制限:

- ▶ [キー押下]ウィジェット機能を追加すると、[押下]および[フォーカス]ウィジェット機能が自動的に追加されます。

表 13.185 [キー押下]ウィジェット機能のプロパティ

プロパティ名	説明
keyPressed	キー押下に対するウィジェットの反応 反応引数は、次のとおりです。 ▶ keyId: 処理されるキーのID

### 13.13.5.3. キーリリース

[キーリリース]ウィジェット機能を使用すると、ウィジェットがキーリリースに反応できるようになります。

制限:

- ▶ [キーリリース]ウィジェット機能を追加すると、[押下]および[フォーカス]ウィジェット機能が自動的に追加されます。

表 13.186 [キーリリース]ウィジェット機能のプロパティ

プロパティ名	説明
keyShortReleased	キーリリースに対するウィジェットの反応 反応引数は、次のとおりです。 ▶ keyId: 処理されるキーのID

### 13.13.5.4. キーのステータス変更

[キーのステータス変更]ウィジェット機能を使用すると、ウィジェットがキー押下またはキーリリースに反応できるようになります。これは、[ショート押下]、[ロング]、[超ロング]、および[連続]などのキー入力に対する反応を定義します。

制限:

- ▶ [キーのステータス変更]ウィジェット機能を追加すると、[押下]および[フォーカス]ウィジェット機能が自動的に追加されます。

表 13.187 [キーのステータス変更]ウィジェット機能のプロパティ

プロパティ名	説明
keyStatusChanged	キー押下またはキーリリースに対するウィジェットの反応 反応引数は、次のとおりです。

プロパティ名	説明
	<ul style="list-style-type: none"> <li>▶ keyId: 処理されるキーのID</li> <li>▶ status: ステータス変更の数値ID</li> </ul>

### 13.13.5.5. キーUnicode

[キーUnicode]ウィジェット機能を使用すると、ウィジェットがUnicodeキー入力に反応できるようになります。

制限:

- ▶ [キーUnicode]ウィジェット機能を追加すると、[押下]および[フォーカス]ウィジェット機能が自動的に追加されません。

表13.188 [キーUnicode]ウィジェット機能のプロパティ

プロパティ名	説明
keyUnicode	<p>Unicodeキー入力に対するウィジェットの反応</p> <p>反応引数は、次のとおりです。</p> <ul style="list-style-type: none"> <li>▶ keyId: 処理されるキーのID</li> </ul>

### 13.13.5.6. 内部へ移動

[内部へ移動]ウィジェット機能を使用すると、ウィジェットが境界内への動きに反応できるようになります。

制限:

- ▶ [内部に移動]ウィジェット機能を追加すると、[タッチ]ウィジェット機能が自動的に追加されます。

表13.189 [内部へ移動]ウィジェット機能のプロパティ

プロパティ名	説明
moveIn	<p>境界内への動きに対するウィジェットの反応</p> <p>反応引数は、次のとおりです。</p> <ul style="list-style-type: none"> <li>▶ touchId: ユーザーがクリックまたはリリースしたタッチスクリーンのID</li> <li>▶ x: X座標</li> <li>▶ y: Y座標</li> <li>▶ fingerId: ウィジェット内を移動するコンタクトのID</li> </ul>

### 13.13.5.7. 外部へ移動

[外部へ移動]ウィジェット機能を使用すると、ウィジェットが境界外への動きに反応できるようになります。

制限:

- ▶ [外部へ移動]ウィジェット機能を追加すると、[タッチ]ウィジェット機能が自動的に追加されます。

表 13.190 [外部へ移動]ウィジェット機能のプロパティ

プロパティ名	説明
moveOut	境界外への動きに対するウィジェットの反応 反応引数は、次のとおりです。 <ul style="list-style-type: none"><li>▶ touchId: ユーザーがクリックまたはリリースしたタッチスクリーンのID</li><li>▶ x: X座標</li><li>▶ y: Y座標</li><li>▶ fingerId: ウィジェット内を移動するコンタクトのID</li></ul>

### 13.13.5.8. 内部で移動

[内部で移動]ウィジェット機能を使用すると、ウィジェットが境界内の動きに反応できるようになります。

制限:

- ▶ [内部で移動]ウィジェット機能を追加すると、[タッチ]ウィジェット機能が自動的に追加されます。

表 13.191 [内部で移動]ウィジェット機能のプロパティ

プロパティ名	説明
moveOver	境界内の動きに対するウィジェットの反応 反応引数は、次のとおりです。 <ul style="list-style-type: none"><li>▶ touchId: ユーザーがクリックまたはリリースしたタッチスクリーンのID</li><li>▶ x: X座標</li><li>▶ y: Y座標</li><li>▶ fingerId: ウィジェット内を移動するコンタクトのID</li></ul>

### 13.13.5.9. 可動

[可動]ウィジェット機能を使用すると、ウィジェットをタッチによって移動できるようになります。

制限:

- ▶ [可動]ウィジェット機能を追加すると、[タッチ]および[タッチ移動]ウィジェット機能が自動的に追加されます。

表 13.192 [可動]ウィジェット機能のプロパティ

プロパティ名	説明
moveDirection	ウィジェットが移動する方向使用可能な値: <ul style="list-style-type: none"><li>▶ horizontal (=0)</li><li>▶ vertical (=1)</li><li>▶ free (=2)</li></ul>

### 13.13.5.10. 回転

[回転]ウィジェット機能を使用すると、ウィジェットが回転に反応できるようになります。

制限:

- ▶ [回転]ウィジェット機能を追加すると、[フォーカス]ウィジェット機能が自動的に追加されます。

表 13.193 [回転]ウィジェット機能のプロパティ

プロパティ名	説明
rotaryReaction	回転に対するウィジェットの反応を定義します。trueである場合、ウィジェットは入力された回転イベントに反応します。  反応引数は、次のとおりです。 <ul style="list-style-type: none"><li>▶ rotaryId: 整数ID</li><li>▶ increment: 入力されたイベントが送信されたときに回転入力移動する単位数</li></ul>

### 13.13.5.11. タッチの喪失

[タッチの喪失]ウィジェット機能を使用すると、ウィジェットがタッチコンタクトの喪失に反応できるようになります。

コンタクトがジェスチャーの一部である場合や、リリースなしでタッチスクリーンを離れた場合、コンタクトは消える可能性があります。この場合、touchShortReleased反応は実行されません。

制限:

- ▶ [タッチの喪失]ウィジェット機能を追加すると、[タッチ]ウィジェット機能が自動的に追加されます。
- ▶ [タッチの喪失]を追加する場合、[タッチ]ウィジェット機能のtouchPolicyドロップダウンリストボックスでPress and grabを選択します。

[タッチの喪失]は、他のタッチポリシーでは機能しません。

表13.194 [タッチの喪失]ウィジェット機能のプロパティ

プロパティ名	説明
onTouchGrabLost	タッチコンタクトの喪失に対するウィジェットの反応 反応引数は、次のとおりです。 <ul style="list-style-type: none"><li>▶ touchId: ユーザーがクリックまたはリリースしたタッチスクリーンのID</li><li>▶ x: X座標</li><li>▶ y: Y座標</li><li>▶ fingerId: ウィジェット内を移動するコンタクトのID</li></ul>

### 13.13.5.12. タッチ移動

[タッチ移動]ウィジェット機能を使用すると、タッチでの移動に反応できるようになります。

制限:

- ▶ [タッチ移動]ウィジェット機能を追加すると、[タッチ]ウィジェット機能が自動的に追加されます。

表13.195 [タッチ移動]ウィジェット機能のプロパティ

プロパティ名	説明
touchMoved	タッチでの移動に対するウィジェットの反応 反応引数は、次のとおりです。 <ul style="list-style-type: none"><li>▶ touchId: ユーザーがクリックまたはリリースしたタッチスクリーンのID</li><li>▶ x: X座標</li><li>▶ y: Y座標</li><li>▶ fingerId: ウィジェット内を移動するコンタクトのID</li></ul>

### 13.13.5.13. タッチ押下

[タッチ押下]ウィジェット機能を使用すると、ウィジェットが押下に反応できるようになります。

制限:

- ▶ [タッチ押下]ウィジェット機能を追加すると、[タッチ]ウィジェット機能が自動的に追加されます。

表13.196 [タッチ押下]ウィジェット機能のプロパティ

プロパティ名	説明
touchPressed	押下に対するウィジェットの反応  反応引数は、次のとおりです。 <ul style="list-style-type: none"><li>▶ touchId: ユーザーがクリックまたはリリースしたタッチスクリーンのID</li><li>▶ x: X座標</li><li>▶ y: Y座標</li><li>▶ fingerId: ウィジェット内を移動するコンタクトのID</li></ul>

#### 13.13.5.14. タッチリリース

[タッチリリース]ウィジェット機能を使用すると、ウィジェットがリリースに反応できるようになります。

制限:

- ▶ [タッチリリース]ウィジェット機能を追加すると、[タッチ]ウィジェット機能が自動的に追加されます。

表13.197 [タッチリリース]ウィジェット機能のプロパティ

プロパティ名	説明
touchShortReleased	リリースに対するウィジェットの反応  反応引数は、次のとおりです。 <ul style="list-style-type: none"><li>▶ touchId: ユーザーがクリックまたはリリースしたタッチスクリーンのID</li><li>▶ x: X座標</li><li>▶ y: Y座標</li><li>▶ fingerId: ウィジェット内を移動するコンタクトのID</li></ul>

#### 13.13.5.15. タッチのステータス変更

[タッチのステータス変更]ウィジェット機能を使用すると、ウィジェットがタッチのステータス変更に反応できるようになります。

制限:



- ▶ [タッチのステータス変更]ウィジェット機能を追加すると、[タッチ]ウィジェット機能が自動的に追加されます。

表13.198 [タッチのステータス変更]ウィジェット機能のプロパティ

プロパティ名	説明
touchStatusChanged	<p>タッチのステータス変更に対するウィジェットの反応</p> <p>反応引数は、次のとおりです。</p> <ul style="list-style-type: none"> <li>▶ touchId: ユーザーがクリックまたはリリースしたタッチスクリーンのID</li> <li>▶ x: X座標</li> <li>▶ y: Y座標</li> <li>▶ touchStatus: タッチのタイプのID</li> </ul> <p>使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ 0: 新しいコンタクト</li> <li>▶ 1: タッチ押下</li> <li>▶ 2: タッチ移動</li> <li>▶ 3: タッチリリース</li> <li>▶ 4: タッチなしの移動</li> <li>▶ 5: タッチ終了</li> </ul> <li>▶ fingerId: ウィジェット内を移動するコンタクトのID</li>

## 13.13.6. レイアウト

### 13.13.6.1. 絶対レイアウト

親ウィジェットの[絶対レイアウト]ウィジェット機能は、子ウィジェットの位置およびサイズを定義します。非表示になっている子ウィジェットは無視されます。追加されたウィジェット機能プロパティは整数リストで構成されます。各リスト要素は1つの子ウィジェットにマップされます。

制限:

- ▶ [絶対レイアウト]ウィジェット機能では、次のウィジェット機能は除外されます。
  - ▶ [ボックスレイアウト]
  - ▶ [フローレイアウト]
  - ▶ [グリッドレイアウト]

▶ [リストレイアウト]

表13.199 [絶対レイアウト]ウィジェット機能のプロパティ

プロパティ名	説明
itemLeftOffset	子ウィジェットの左枠からのオフセットを格納する整数リスト。各リスト要素は1つの子ウィジェットにマップされます。
itemTopOffset	子ウィジェットの上枠からのオフセットを格納する整数リスト。各リスト要素は1つの子ウィジェットにマップされます。
itemRightOffset	子ウィジェットの右枠からのオフセットを格納する整数リスト。各リスト要素は1つの子ウィジェットにマップされます。
itemBottomOffset	子ウィジェットの下の枠からのオフセットを格納する整数リスト。各リスト要素は1つの子ウィジェットにマップされます。

### 13.13.6.2. ボックスレイアウト

[ボックスレイアウト]ウィジェット機能は、各子ウィジェットの位置およびサイズを定義します。

子ウィジェットの位置プロパティおよびサイズプロパティは、親ウィジェットによって設定されます。非表示になっている子ウィジェットは計算では無視されます。

制限:

- ▶ [ボックスレイアウト]ウィジェット機能では、次のウィジェット機能は除外されます。
  - ▶ [絶対レイアウト]
  - ▶ [フローレイアウト]
  - ▶ [グリッドレイアウト]
  - ▶ [リストレイアウト]

表13.200 [ボックスレイアウト]ウィジェット機能のプロパティ

プロパティ名	説明
gap	レイアウトの方向に応じた2つの子ウィジェット間のスペース
layoutDirection	リスト要素(子ウィジェット)が配置される方向使用可能な値: <ul style="list-style-type: none"> <li>▶ horizontal (=0)</li> <li>▶ vertical (=1)</li> </ul>

### 13.13.6.3. フローレイアウト

[フローレイアウト]ウィジェット機能は、各子ウィジェットの位置およびサイズを定義します。

子ウィジェットの位置プロパティおよびサイズプロパティは、親ウィジェットによって設定されます。非表示になっている子ウィジェットは計算では無視されます。

制限:

- ▶ [フローレイアウト]ウィジェット機能では、次のウィジェット機能は除外されます。
  - ▶ [絶対レイアウト]
  - ▶ [ボックスレイアウト]
  - ▶ [グリッドレイアウト]
  - ▶ [リストレイアウト]

表13.201 [フローレイアウト]ウィジェット機能のプロパティ

プロパティ名	説明
horizontalGap	2つの子ウィジェット間の水平方向のスペース
verticalGap	2つの子ウィジェット間の垂直方向のスペース
layoutDirection	リスト要素(子ウィジェット)が配置される方向使用可能な値: <ul style="list-style-type: none"><li>▶ horizontal (=0)</li><li>▶ vertical (=1)</li></ul>
horizontalChildAlign	子ウィジェットの水平方向の位置揃え使用可能な値: <ul style="list-style-type: none"><li>▶ leading (=0): 子ウィジェットは左側に配置されます。</li><li>▶ center (=1): 子ウィジェットは中央に配置されます。</li><li>▶ trailing (=2): 子ウィジェットは右側に配置されます。</li></ul>
verticalChildAlign	子ウィジェットの垂直方向の位置揃え使用可能な値: <ul style="list-style-type: none"><li>▶ center (=0): 子ウィジェットは中央に配置されます。</li><li>▶ top (=1): 子ウィジェットは上に配置されます。</li><li>▶ bottom (=2): 子ウィジェットは下に配置されます。</li></ul>

#### 13.13.6.4. グリッドレイアウト

[グリッドレイアウト]ウィジェット機能は、各子ウィジェットの位置およびサイズを定義します。

子ウィジェットの位置プロパティおよびサイズプロパティは、親ウィジェットによって設定されます。非表示になっている子ウィジェットは計算では無視されます。

制限:

- ▶ [グリッドレイアウト]ウィジェット機能では、次のウィジェット機能は除外されます。
  - ▶ [絶対レイアウト]
  - ▶ [ボックスレイアウト]
  - ▶ [フローレイアウト]
  - ▶ [リストレイアウト]

表13.202 [グリッドレイアウト]ウィジェット機能のプロパティ

プロパティ名	説明
horizontalGap	2つの子ウィジェット間の水平方向のスペース
verticalGap	2つの子ウィジェット間の垂直方向のスペース
numRows	行の数を定義します。
numColumns	列の数を定義します。

### 13.13.6.5. レイアウト余白

[レイアウト余白]ウィジェット機能は、[フローレイアウト]、[絶対レイアウト]、[ボックスレイアウト]、または[グリッドレイアウト]ウィジェット機能を使用するウィジェットに設定可能な余白を追加します。

表13.203 [レイアウト余白]ウィジェット機能のプロパティ

プロパティ名	説明
leftMargin	左枠の余白
topMargin	上枠の余白
rightMargin	右枠の余白
bottomMargin	下枠の余白

### 13.13.6.6. リストレイアウト

[リストレイアウト]ウィジェット機能は、ピクセル単位で各子ウィジェットの位置およびサイズを定義します。

子ウィジェットの位置プロパティおよび[リストインデックス]ウィジェット機能のlistIndexプロパティは、親ウィジェットによって設定されます。

子ウィジェットを作成するインスタシエータとともに使用するのが最も適しています。

[リストインデックス]ウィジェット機能の詳細については、[13.13.7.2「リストインデックス」](#)をご覧ください。

制限:

- ▶ [リストレイアウト]ウィジェット機能は、インスタシエータとともに使用するよう設計されています。
- ▶ [リストレイアウト]ウィジェット機能では、次のウィジェット機能は除外されます。
  - ▶ [絶対レイアウト]
  - ▶ [ボックスレイアウト]
  - ▶ [フローレイアウト]
  - ▶ [グリッドレイアウト]

表 13.204 [リストレイアウト]ウィジェット機能のプロパティ

プロパティ名	説明
layoutDirection	リスト要素(子ウィジェット)が配置される方向使用可能な値: ▶ horizontal (=0) ▶ vertical (=1)
scrollOffset	リストをスクロールするピクセルの数
scrollOffsetRebase	scrollOffsetRebaseプロパティが変更されると、現在のscrollOffsetがscrollIndexに変換されます。残りのオフセットはscrollOffsetプロパティに書き込まれます。
firstListIndex	ウィジェット機能によって定義される、最初に表示されるリスト要素のリストインデックス
scrollIndex	scrollOffsetプロパティが適用される基本リストインデックス。スクロールは、scrollIndexプロパティに指定されているリスト要素から開始されます。
scrollValue	現在のスクロール値(ピクセル単位)
scrollValueMax	リストの終了位置にマップされる最大スクロール値(ピクセル単位)
scrollValueMin	リストの開始位置にマップされる最小スクロール値(ピクセル単位)
bounceValue	scrollOffsetプロパティが有効なスクロール範囲内の位置にある限り、bounceValueプロパティはゼロです。この値は、スクロール位置がリストの開始位置を超える場合は正、スクロール位置がリストの終了位置を超える場合は負になります。bounceValueをscrollOffsetに追加すると、スクロール位置が範囲内に戻ります。
bounceValueMax	scrollOffsetが有効なスクロール範囲の外で移動できる最大値。scrollOffsetは、ユーザーがさらにスクロールしようとするとき切り捨てられます。
segments	水平レイアウト方向の場合: 行の数 垂直レイアウト方向の場合: 列の数
listLength	リスト要素の数
wrapAround	使用可能な値:

プロパティ名	説明
	<ul style="list-style-type: none"> <li>▶ <b>true:</b> scrollValueMinまたはscrollValueMaxを超えた場合、scrollValueプロパティは逆側に進みます。</li> <li>▶ <b>false:</b> scrollValueMinまたはscrollValueMaxを超えた場合、scrollValueプロパティは増加/減少しません。</li> </ul>

### 13.13.6.7. 拡大縮小モード

[拡大縮小モード]ウィジェット機能は、イメージのサイズがウィジェットのサイズと異なる場合にイメージを表示する方法を定義します。

制限:

- ▶ [拡大縮小モード]ウィジェット機能は、ウィジェットイメージにのみ使用できます。

表13.205 [拡大縮小モード]ウィジェット機能のプロパティ

プロパティ名	説明
scaleMode	<p>イメージの拡大縮小モード。使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ 0 = original size</li> <li>▶ 1 = fit to size</li> <li>▶ 2 = keep aspect ratio</li> </ul>

## 13.13.7. リスト管理

### 13.13.7.1. ラインインデックス

[ラインインデックス]ウィジェット機能は、リストまたは表の各ラインの一意の位置を定義します。

制限:

- ▶ [ラインインデックス]ウィジェット機能は、インスタシエータとともに使用するよう設計されています。

表13.206 [ラインインデックス]ウィジェット機能のプロパティ

プロパティ名	説明
lineIndex	表内の現在のラインのインデックス

### 13.13.7.2. リストインデックス

[リストインデックス]ウィジェット機能は、リスト内のウィジェットの一意の位置を定義します。

制限:

- ▶ [リストインデックス]ウィジェット機能は、[リストレイアウト]ウィジェット機能とともに使用するよう設計されています。

表13.207 [リストインデックス]ウィジェット機能のプロパティ

プロパティ名	説明
listIndex	リスト内の現在のウィジェットのインデックス

### 13.13.7.3. テンプレートインデックス

[テンプレートインデックス]ウィジェット機能は、使用されるラインテンプレートの一意の位置を定義します。

制限:

- ▶ [テンプレートインデックス]ウィジェット機能は、インスタンシエータとともに使用するよう設計されています。

表13.208 [テンプレートインデックス]ウィジェット機能のプロパティ

プロパティ名	説明
lineTemplateIndex	使用されているラインテンプレートのインデックス

### 13.13.7.4. ビューポート

[ビューポート]ウィジェット機能は、ウィジェットの境界にあるサイズ超過の要素をクリップします。

制限:

- ▶ [ビューポート]は、コンテナまたはリストとともに使用するよう設計されています。
- ▶ [ビューポート]ウィジェット機能は、次のモデル要素に対して有効です。
  - ▶ [ビューポート]を追加したウィジェットの子ウィジェットは、そのウィジェットの寸法内でクリップされます。
  - ▶ [ビューポート]を追加したウィジェットは、その親ビューの寸法内でクリップされます。

表13.209 [ビューポート]ウィジェット機能のプロパティ

プロパティ名	説明
xOffset	子ウィジェットの描画エリアにおける表示クリッピングの水平方向のオフセット
yOffset	子ウィジェットの描画エリアにおける表示クリッピングの垂直方向のオフセット

## 13.13.8. 3D

[3D]カテゴリのウィジェット機能は、3Dウィジェットに対してのみ使用することができます。

### 13.13.8.1. カメラビューポート

[カメラビューポート]ウィジェット機能は、シーングラフ内でのカメラの描画領域を定義します。

制限:

- ▶ [カメラビューポート]ウィジェット機能は、カメラに対して使用できます。

表13.210 [カメラビューポート]ウィジェット機能のプロパティ

プロパティ名	説明
viewportX	シーングラフ内のビューポートのX原点
viewportY	シーングラフ内のビューポートのY原点
viewportWidth	ビューポートの幅(ピクセル単位)
viewportHeight	ビューポートの高さ(ピクセル単位)

### 13.13.8.2. アンビエントテクスチャ

[アンビエントテクスチャ]ウィジェット機能は、材質に拡張設定値を追加します。

制限:

- ▶ [アンビエントテクスチャ]ウィジェット機能は、材質、PBR Phong材質、およびPBR GGX材質で使用できます。
- ▶ [アンビエントテクスチャ]が追加されている場合、ambientプロパティは無視されます。

表13.211 [アンビエントテクスチャ]ウィジェット機能のプロパティ

プロパティ名	説明
ambientTexture	テクスチャのファイル名
ambientTextureAddressModeU	u方向のテクスチャのアドレスモード。使用可能な値: <ul style="list-style-type: none"><li>▶ repeat (=0): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャが繰り返し返されます。ラップまたはタイルとも呼ばれます</li><li>▶ clamp (=1): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャの端のピクセルが使用されます。</li></ul>
ambientTextureAddressModeV	v方向のテクスチャのアドレスモード。使用可能な値:



プロパティ名	説明
	<ul style="list-style-type: none"> <li>▶ repeat (=0): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャが繰り返されます。ラップまたはタイルとも呼ばれます</li> <li>▶ clamp (=1): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャの端のピクセルが使用されます。</li> </ul>
ambientFilterMode	<p>テクスチャのフィルタリングモード。使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ point (=0): テクスチャはまったく平滑化されません。最もコストが低いが、テクスチャが最小化されたときにアーティファクトがぎざぎざになりやすくなります。</li> <li>▶ linear (=1): バイリニアフィルタリングとも呼ばれます。アーティファクトがぎざぎざにならないように最小化されたときにテクスチャを平滑化します。</li> <li>▶ trilinear (=2): 最もコストが高いが、リニアフィルタリングよりもよい結果となります。</li> </ul>
diffuseSRGB	<p>このプロパティが有効になっている場合、ambientTextureで選択されているテクスチャはsRGB色空間を使用してレンダリングされます。</p> <p>sRGB機能を使用するには、プロジェクトセンターを開き、[設定] &gt; [プロファイル]にあるcolorModeプロパティで、32-bit sRGB (=4)または32-bit sRGB (Emulated) (=5)を選択します。</p>

### 13.13.8.3. ディフューズテクスチャ

[ディフューズテクスチャ]ウィジェット機能は、材質に拡張設定値を追加します。

制限:

- ▶ [ディフューズテクスチャ]ウィジェット機能は、材質、PBR Phong材質、およびPBR GGX材質で使用できます。
- ▶ [ディフューズテクスチャ]が追加されている場合、diffuseプロパティは無視されます。

表13.212 [ディフューズテクスチャ]ウィジェット機能のプロパティ

プロパティ名	説明
diffuseTexture	テクスチャのファイル名
diffuseTextureAddressModeU	<p>u方向のテクスチャのアドレスモード。使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ repeat (=0): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャが繰り返されます。ラップまたはタイルとも呼ばれます</li> <li>▶ clamp (=1): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャの端のピクセルが使用されます。</li> </ul>

プロパティ名	説明
diffuseTextureAddressModeV	<p>V方向のテクスチャのアドレスモード。使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ repeat (=0): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャが繰り返されます。ラップまたはタイルとも呼ばれます</li> <li>▶ clamp (=1): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャの端のピクセルが使用されます。</li> </ul>
diffuseFilterMode	<p>テクスチャのフィルタリングモード。使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ point (=0): テクスチャはまったく平滑化されません。最もコストが低いが、テクスチャが最小化されたときにアーティファクトがぎざぎざになりやすくなります。</li> <li>▶ linear (=1): バイリニアフィルタリングとも呼ばれます。アーティファクトがぎざぎざにならないように最小化されたときにテクスチャを平滑化します。</li> <li>▶ trilinear (=2): 最もコストが高いが、リニアフィルタリングよりもよい結果となります。</li> </ul>
diffuseSRGB	<p>このプロパティが有効になっている場合、diffuseTextureで選択されているテクスチャはsRGB色空間を使用してレンダリングされます。</p> <p>sRGB機能を使用するには、プロジェクトセンターを開き、[設定] &gt; [プロファイル]にあるcolorModeプロパティで、32-bit sRGB (=4)または32-bit sRGB (Emulated) (=5)を選択します。</p>

#### 13.13.8.4. エミッシブテクスチャ

[エミッシブテクスチャ]ウィジェット機能は、材質に拡張設定値を追加します。

制限:

- ▶ [エミッシブテクスチャ]ウィジェット機能は、材質、PBR Phong材質、およびPBR GGX材質で使用できます。
- ▶ [エミッシブテクスチャ]が追加されている場合、emissiveプロパティは無視されます。

表13.213 [エミッシブテクスチャ]ウィジェット機能のプロパティ

プロパティ名	説明
emissiveTexture	テクスチャのファイル名
emissiveTextureAddressModeU	<p>U方向のテクスチャのアドレスモード。使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ repeat (=0): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャが繰り返されます。ラップまたはタイルとも呼ばれます</li> </ul>

プロパティ名	説明
	<ul style="list-style-type: none"> <li>▶ clamp (=1): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャの端のピクセルが使用されます。</li> </ul>
emissiveTextureAddress ModeV	<p>V方向のテクスチャのアドレスモード。使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ repeat (=0): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャが繰り返されます。ラップまたはタイルとも呼ばれます</li> <li>▶ clamp (=1): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャの端のピクセルが使用されます。</li> </ul>
emissiveFilterMode	<p>テクスチャのフィルタリングモード。使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ point (=0): テクスチャはまったく平滑化されません。最もコストが低いですが、テクスチャが最小化されたときにアーティファクトがぎざぎざになりやすくなります。</li> <li>▶ linear (=1): バイリニアフィルタリングとも呼ばれます。アーティファクトがぎざぎざにならないように最小化されたときにテクスチャを平滑化します。</li> <li>▶ trilinear (=2): 最もコストが高いが、リニアフィルタリングよりもよい結果となります。</li> </ul>
diffuseSRGB	<p>このプロパティが有効になっている場合、emissiveTextureで選択されているテクスチャはsRGB色空間を使用してレンダリングされます。</p> <p>sRGB機能を使用するには、プロジェクトセンターを開き、[設定] &gt; [プロファイル]にあるcolorModeプロパティで、32-bit sRGB (=4)または32-bit sRGB (Emulated) (=5)を選択します。</p>

### 13.13.8.5. ライトマップテクスチャ

[ライトマップテクスチャ]ウィジェット機能は、材質に拡張設定値を追加します。

制限:

- ▶ [ライトマップテクスチャ]ウィジェット機能は、材質、PBR Phong材質、およびPBR GGX材質で使用できます。

表13.214 [ライトマップテクスチャ]ウィジェット機能のプロパティ

プロパティ名	説明
lightMapTexture	テクスチャのファイル名
lightMapTextureAddress ModeU	<p>U方向のテクスチャのアドレスモード。使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ repeat (=0): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャが繰り返されます。ラップまたはタイルとも呼ばれます</li> </ul>

プロパティ名	説明
	<ul style="list-style-type: none"> <li>▶ <code>clamp (=1)</code>: テクスチャの制限範囲の外側からアクセスされた場合、テクスチャの端のピクセルが使用されます。</li> </ul>
<code>lightMapTextureAddressModeV</code>	<p>V方向のテクスチャのアドレスモード。使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ <code>repeat (=0)</code>: テクスチャの制限範囲の外側からアクセスされた場合、テクスチャが繰り返されます。ラップまたはタイルとも呼ばれます</li> <li>▶ <code>clamp (=1)</code>: テクスチャの制限範囲の外側からアクセスされた場合、テクスチャの端のピクセルが使用されます。</li> </ul>
<code>lightMapFilterMode</code>	<p>テクスチャのフィルタリングモード。使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ <code>point (=0)</code>: テクスチャはまったく平滑化されません。最もコストが低いですが、テクスチャが最小化されたときにアーティファクトがぎざぎざになりやすくなります。</li> <li>▶ <code>linear (=1)</code>: バイリニアフィルタリングとも呼ばれます。アーティファクトがぎざぎざにならないように最小化されたときにテクスチャを平滑化します。</li> <li>▶ <code>trilinear (=2)</code>: 最もコストが高いが、リニアフィルタリングよりもよい結果となります。</li> </ul>

### 13.13.8.6. 金属テクスチャ

[金属]ウィジェット機能は、材質に拡張設定値を追加します。このテクスチャは、PBR GGX材質とPBR Phong材質ウィジェットの金属パラメータを制御します。

制限:

- ▶ [金属テクスチャ]ウィジェット機能は、PBR GGX材質およびPBR Phong材質で使用できます。
- ▶ [金属テクスチャ]はグレースケールイメージです。RGBカラーイメージでは、赤のチャンネルのみが使用されます。
- ▶ [金属テクスチャ]が追加されている場合、`metallic`プロパティは無視されます。

表13.215 [金属テクスチャ]ウィジェット機能のプロパティ

プロパティ名	説明
<code>metallicTexture</code>	テクスチャのファイル名
<code>metallicMinFactor</code>	テクスチャ値を補間する浮動小数点数として最小の金属パラメータ
<code>metallicMaxFactor</code>	テクスチャ値を補間する浮動小数点数として最大の金属パラメータ
<code>metallicTextureAddressModeU</code>	<p>U方向のテクスチャのアドレスモード。使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ <code>repeat (=0)</code>: テクスチャの制限範囲の外側からアクセスされた場合、テクスチャが繰り返されます。ラップまたはタイルとも呼ばれます</li> </ul>

プロパティ名	説明
	<ul style="list-style-type: none"> <li>▶ clamp (=1): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャの端のピクセルが使用されます。</li> </ul>
metallicTextureAddress ModeV	<p>V方向のテクスチャのアドレスモード。使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ repeat (=0): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャが繰り返されます。ラップまたはタイルとも呼ばれます</li> <li>▶ clamp (=1): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャの端のピクセルが使用されます。</li> </ul>
metallicFilterMode	<p>テクスチャのフィルタリングモード。使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ point (=0): テクスチャはまったく平滑化されません。最もコストが低いですが、テクスチャが最小化されたときにアーティファクトがぎざぎざになりやすくなります。</li> <li>▶ linear (=1): バイリニアフィルタリングとも呼ばれます。アーティファクトがぎざぎざにならないように最小化されたときにテクスチャを平滑化します。</li> <li>▶ trilinear (=2): 最もコストが高いが、リニアフィルタリングよりもよい結果となります。</li> </ul>

### 13.13.8.7. ノーマルマップテクスチャ

[ノーマルマップ]ウィジェット機能は、材質に拡張設定値を追加します。

制限:

- ▶ [ノーマルマップテクスチャ]ウィジェット機能は、材質、PBR Phong材質、およびPBR GGX材質で使用できます。

表13.216 [ノーマルマップ]ウィジェット機能のプロパティ

プロパティ名	説明
normalMapTexture	テクスチャのファイル名
normalMapTextureAddress ModeU	<p>U方向のテクスチャのアドレスモード。使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ repeat (=0): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャが繰り返されます。ラップまたはタイルとも呼ばれます</li> <li>▶ clamp (=1): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャの端のピクセルが使用されます。</li> </ul>
normalMapTextureAddress ModeV	V方向のテクスチャのアドレスモード。使用可能な値:

プロパティ名	説明
	<ul style="list-style-type: none"> <li>▶ repeat (=0): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャが繰り返されます。ラップまたはタイルとも呼ばれます</li> <li>▶ clamp (=1): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャの端のピクセルが使用されます。</li> </ul>
normalMapFilterMode	<p>テクスチャのフィルタリングモード。使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ point (=0): テクスチャはまったく平滑化されません。最もコストが低いが、テクスチャが最小化されたときにアーティファクトがぎざぎざになりやすくなります。</li> <li>▶ linear (=1): バイリニアフィルタリングとも呼ばれます。アーティファクトがぎざぎざにならないように最小化されたときにテクスチャを平滑化します。</li> <li>▶ trilinear (=2): 最もコストが高いが、リニアフィルタリングよりもよい結果となります。</li> </ul>

### 13.13.8.8. 不透明テクスチャ

[不透明テクスチャ]ウィジェット機能は、材質に拡張設定値を追加します。

制限:

- ▶ [不透明テクスチャ]ウィジェット機能は、材質、PBR Phong材質、およびPBR GGX材質で使用できます。

表13.217 [不透明テクスチャ]ウィジェット機能のプロパティ

プロパティ名	説明
opaqueTexture	テクスチャのファイル名
opaqueTextureAddressModeU	<p>U方向のテクスチャのアドレスモード。使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ repeat (=0): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャが繰り返されます。ラップまたはタイルとも呼ばれます</li> <li>▶ clamp (=1): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャの端のピクセルが使用されます。</li> </ul>
opaqueTextureAddressModeV	<p>V方向のテクスチャのアドレスモード。使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ repeat (=0): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャが繰り返されます。ラップまたはタイルとも呼ばれます</li> <li>▶ clamp (=1): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャの端のピクセルが使用されます。</li> </ul>

プロパティ名	説明
opaqueFilterMode	<p>テクスチャのフィルタモード。使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ point (=0): テクスチャはまったく平滑化されません。最もコストが低いが、テクスチャが最小化されたときにアーティファクトがぎざぎざになりやすくなります。</li> <li>▶ linear (=1): バイリニアフィルタリングとも呼ばれます。アーティファクトがぎざぎざにならないように最小化されたときにテクスチャを平滑化します。</li> <li>▶ trilinear (=2): 最もコストが高いが、リニアフィルタリングよりもよい結果となります。</li> </ul>

### 13.13.8.9. リフレクションテクスチャ

[リフレクションテクスチャ]ウィジェット機能は、材質に拡張設定値を追加します。

制限:

- ▶ [リフレクションテクスチャ]ウィジェット機能は、材質、PBR Phong材質、およびPBR GGX材質で使用できます。

表13.218 [リフレクションテクスチャ]ウィジェット機能のプロパティ

プロパティ名	説明
reflectionTopTexture	テクスチャのファイル名
reflectionBottomTexture	テクスチャのファイル名
reflectionLeftTexture	テクスチャのファイル名
reflectionRightTexture	テクスチャのファイル名
reflectionFrontTexture	テクスチャのファイル名
reflectionBackTexture	テクスチャのファイル名
reflectionFilterMode	<p>テクスチャのフィルタリングモード。使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ point (=0): テクスチャはまったく平滑化されません。最もコストが低いが、テクスチャが最小化されたときにアーティファクトがぎざぎざになりやすくなります。</li> <li>▶ linear (=1): バイリニアフィルタリングとも呼ばれます。アーティファクトがぎざぎざにならないように最小化されたときにテクスチャを平滑化します。</li> <li>▶ trilinear (=2): 最もコストが高いが、リニアフィルタリングよりもよい結果となります。</li> </ul>

注記



[リフレクションテクスチャ]ウィジェット機能

EB GUIDE Studioは、イメージファイルが以下のプロパティすべてで選択されている場合にのみ、[リフレクションテクスチャ]ウィジェット機能を表示します。

- ▶ reflectionTopTexture
- ▶ reflectionBottomTexture
- ▶ reflectionLeftTexture
- ▶ reflectionRightTexture
- ▶ reflectionFrontTexture
- ▶ reflectionBackTexture

イメージファイルは、同一サイズで長方形のものでなければなりません。

### 13.13.8.10. ラフネス(粗さ)テクスチャ

[ラフネス(粗さ)テクスチャ]ウィジェット機能は、材質に拡張設定値を追加します。このテクスチャは、PBR GGX材質ウィジェットのラフネス(粗さ)パラメータを制御します。

制限:

- ▶ [ラフネス(粗さ)テクスチャ]ウィジェット機能は、PBR GGX材質に対して使用できます。
- ▶ [ラフネス(粗さ)テクスチャ]はグレースケールイメージです。RGBカラーイメージでは、赤のチャンネルのみが使用されます。
- ▶ [ラフネス(粗さ)テクスチャ]がアクティブの場合、roughnessプロパティは無視されます。

表13.219 [ラフネス(粗さ)テクスチャ]ウィジェット機能のプロパティ

プロパティ名	説明
roughnessTexture	テクスチャのファイル名
roughnessMinFactor	テクスチャ値を補間する浮動小数点数として最小のラフネス(粗さ)パラメータ
roughnessMaxFactor	テクスチャ値を補間する浮動小数点数として最大のラフネス(粗さ)パラメータ
roughnessTextureAddress ModeU	u方向のテクスチャのアドレスモード。使用可能な値: <ul style="list-style-type: none"> <li>▶ repeat (=0): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャが繰り返されます。ラップまたはタイルとも呼ばれます</li> <li>▶ clamp (=1): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャの端のピクセルが使用されます。</li> </ul>
roughnessTextureAddress ModeV	v方向のテクスチャのアドレスモード。使用可能な値:



プロパティ名	説明
	<ul style="list-style-type: none"> <li>▶ repeat (=0): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャが繰り返されます。ラップまたはタイルとも呼ばれます</li> <li>▶ clamp (=1): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャの端のピクセルが使用されます。</li> </ul>
roughnessFilterMode	<p>テクスチャのフィルタリングモード。使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ point (=0): テクスチャはまったく平滑化されません。最もコストが低いが、テクスチャが最小化されたときにアーティファクトがぎざぎざになりやすくなります。</li> <li>▶ linear (=1): バイリニアフィルタリングとも呼ばれます。アーティファクトがぎざぎざにならないように最小化されたときにテクスチャを平滑化します。</li> <li>▶ trilinear (=2): 最もコストが高いが、リニアフィルタリングよりもよい結果となります。</li> </ul>

### 13.13.8.11. 光沢テクスチャ

[光沢テクスチャ]ウィジェット機能は、材質に拡張設定値を追加します。このテクスチャは、スカラーshininessプロパティを持つテクスチャ値を増加させることで光沢強度を調整します。

制限:

- ▶ [光沢テクスチャ]ウィジェット機能は、材質およびPBR Phong材質で使用できます。
- ▶ [光沢テクスチャ]はグレースケールイメージです。RGBカラーイメージでは、赤のチャンネルのみが使用されます。
- ▶ [光沢テクスチャ]ウィジェット機能が使用されている場合、shininessプロパティは無視されます。

表13.220 [光沢テクスチャ]ウィジェット機能のプロパティ

プロパティ名	説明
shininessTexture	テクスチャのファイル名
shininessMinFactor	テクスチャ値を補間する浮動小数点数として最小の光沢パラメータ
shininessMaxFactor	テクスチャ値を補間する浮動小数点数として最大の光沢パラメータ
shininessTextureAddressModeU	<p>u方向のテクスチャのアドレスモード。使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ repeat (=0): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャが繰り返されます。ラップまたはタイルとも呼ばれます</li> <li>▶ clamp (=1): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャの端のピクセルが使用されます。</li> </ul>

プロパティ名	説明
shininessTextureAddress ModeV	V方向のテクスチャのアドレスモード。使用可能な値: <ul style="list-style-type: none"> <li>▶ repeat (=0): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャが繰り返されます。ラップまたはタイルとも呼ばれます</li> <li>▶ clamp (=1): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャの端のピクセルが使用されます。</li> </ul>
shininessFilterMode	テクスチャのフィルタリングモード。使用可能な値: <ul style="list-style-type: none"> <li>▶ point (=0): テクスチャはまったく平滑化されません。最もコストが低いですが、テクスチャが最小化されたときにアーティファクトがぎざぎざになりやすくなります。</li> <li>▶ linear (=1): バイリニアフィルタリングとも呼ばれます。アーティファクトがぎざぎざにならないように最小化されたときにテクスチャを平滑化します。</li> <li>▶ trilinear (=2): 最もコストが高いが、リニアフィルタリングよりもよい結果となります。</li> </ul>

### 13.13.8.12. スペキュラテクスチャ

[スペキュラテクスチャ]ウィジェット機能は、材質に拡張設定値を追加します。

制限:

- ▶ [スペキュラテクスチャ]ウィジェット機能は、材質、PBR Phong材質、およびPBR GGX材質で使用できます。
- ▶ [スペキュラテクスチャ]が追加されている場合、specularプロパティは無視されます。

表13.221 [スペキュラテクスチャ]ウィジェット機能のプロパティ

プロパティ名	説明
specularTexture	テクスチャのファイル名
specularTextureAddress ModeU	U方向のテクスチャのアドレスモード。使用可能な値: <ul style="list-style-type: none"> <li>▶ repeat (=0): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャが繰り返されます。ラップまたはタイルとも呼ばれます</li> <li>▶ clamp (=1): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャの端のピクセルが使用されます。</li> </ul>
specularTextureAddress ModeV	V方向のテクスチャのアドレスモード。使用可能な値: <ul style="list-style-type: none"> <li>▶ repeat (=0): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャが繰り返されます。ラップまたはタイルとも呼ばれます</li> </ul>

プロパティ名	説明
	<ul style="list-style-type: none"> <li>▶ clamp (=1): テクスチャの制限範囲の外側からアクセスされた場合、テクスチャの端のピクセルが使用されます。</li> </ul>
specularFilterMode	<p>テクスチャのフィルタリングモード。使用可能な値:</p> <ul style="list-style-type: none"> <li>▶ point (=0): テクスチャはまったく平滑化されません。最もコストが低いが、テクスチャが最小化されたときにアーティファクトがぎざぎざになりやすくなります。</li> <li>▶ linear (=1): バイリニアフィルタリングとも呼ばれます。アーティファクトがぎざぎざにならないように最小化されたときにテクスチャを平滑化します。</li> <li>▶ trilinear (=2): 最もコストが高いが、リニアフィルタリングよりもよい結果となります。</li> </ul>
diffuseSRGB	<p>このプロパティが有効になっている場合、specularTextureで選択されているテクスチャはsRGB色空間を使用してレンダリングされます。</p> <p>sRGB機能を使用するには、プロジェクトセンターを開き、[設定] &gt; [プロファイル]にあるcolorModeプロパティで、32-bit sRGB (=4)または32-bit sRGB (Emulated) (=5)を選択します。</p>

### 13.13.8.13. Texture coordinate transformation

[Texture coordinate transformation]ウィジェット機能を使用すると、材質テクスチャの座標を変更できます。この機能は、材質、PBR Phong材質、およびPBR GGX材質で使用できます。

表 13.222 [Texture coordinate transformation]ウィジェット機能のプロパティ

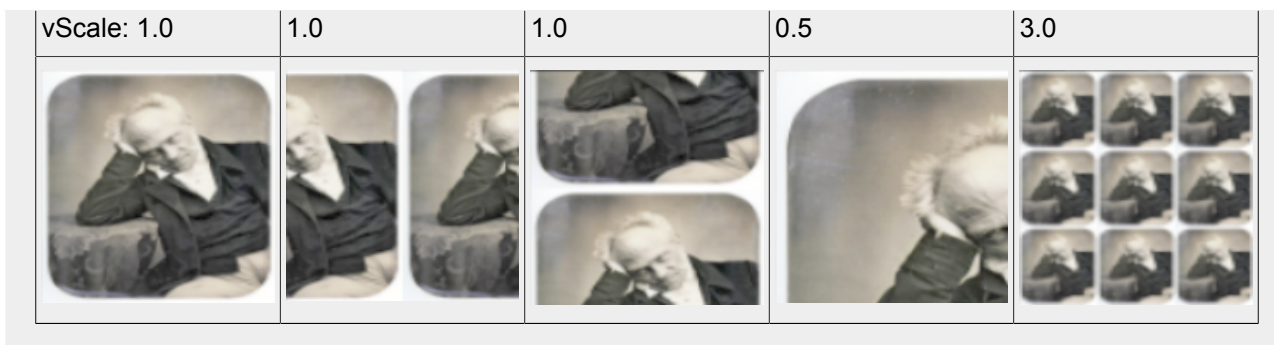
プロパティ名	説明
uOffset	テクスチャ座標のu方向のオフセットを定義します。
vOffset	テクスチャ座標のv方向のオフセットを定義します。
uScale	テクスチャ座標のu方向の拡大縮小を定義します。
vScale	テクスチャ座標のv方向の拡大縮小を定義します。



#### 例13.5

[Texture coordinate transformation]の例

uOffset: 1.0	0.5	1.0	1.0	1.0
vOffset: 1.0	1.0	0.5	1.0	1.0
uScale: 1.0	1.0	1.0	0.5	3.0



### 13.13.8.14. トーンマッピング

[トーンマッピング]ウィジェット機能は、トーンマッピングを有効にします。トーンマッピングは、シーングラフで輝度の値を限定された範囲にマッピングする技法です。

制限:

- ▶ [トーンマッピング]ウィジェット機能は、シーングラフで使用できます。

[トーンマッピング]ウィジェット機能は、Erik Reinhard氏らによるグローバルトーンマッピング演算子を実装しています。<sup>1</sup>

表13.223 [トーンマッピング]ウィジェット機能のプロパティ

プロパティ名	説明
pureWhiteLuminance	純白にマップされる最小の輝度値。ただし、0以上の値のみが有効です。

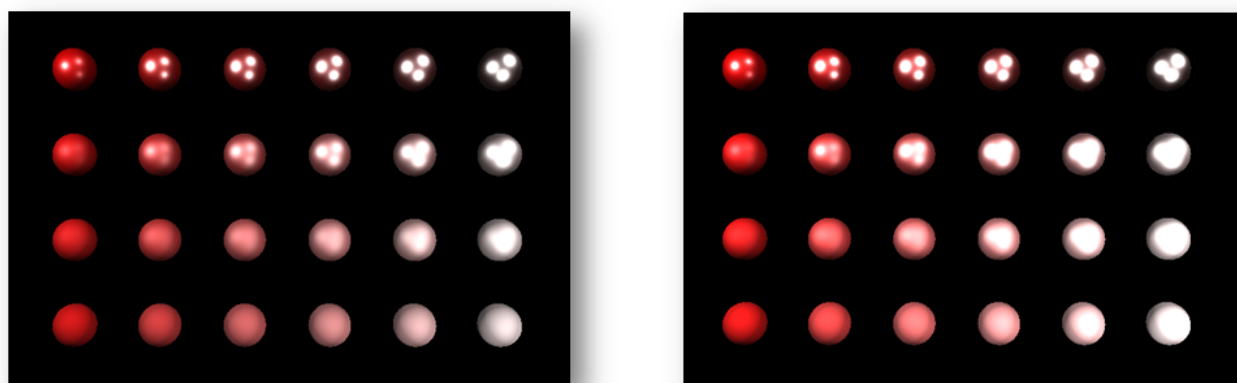


図13.3 トーンマッピングなしイメージ(左)とトーンマッピングありイメージ(右)のサンプル

<sup>1</sup>Photographic tone reproduction for digital images、Reinhard, Erik et al.、"Proceedings of the 29th annual conference on Computer graphics and interactive techniques"(2002年)、267~276ページ

## 13.13.9. 変換

[変形]カテゴリのウィジェット機能は、ウィジェットの位置、形式、およびサイズを変更します。

変形が実行される順序は、ウィジェットツリー内の順序と同じです。同じウィジェットツリーの階層レベルで複数の変形が1つのウィジェットに適用される場合、その順序は次のようになります。

1. 変換
2. せん断
3. 拡大縮小
4. z軸中心の回転
5. y軸中心の回転
6. x軸中心の回転

### 13.13.9.1. ピボット

[ピボット]ウィジェット機能は、ウィジェットに適用される変形のピボット点を定義します。ピボット点が設定されていない場合、デフォルトのピボット点は(0.0, 0.0, 0.0)になります。

制限:

- ▶ [ピボット]ウィジェット機能を追加すると、[回転]、[拡大縮小]、および[せん断]ウィジェット機能が自動的に追加されます。

表13.224 [ピボット]ウィジェット機能のプロパティ

プロパティ名	説明
pivotX	親ウィジェットを基準にしたx軸のピボット点
pivotY	親ウィジェットを基準にしたy軸のピボット点
pivotZ	ウィジェットがシーングラフである場合、親ウィジェットを基準にしたz軸のピボット点

### 13.13.9.2. 回転

[回転]ウィジェット機能は、ウィジェットおよびサブツリーを回転させるために使用します。

表13.225 [回転]ウィジェット機能のプロパティ

プロパティ名	説明
rotationEnabled	回転を使用するかどうかを定義します。

プロパティ名	説明
rotationAngleX	X軸での回転角度。このプロパティはシーングラフにのみ影響します。
rotationAngleY	Y軸での回転角度。このプロパティはシーングラフにのみ影響します。
rotationAngleZ	Z軸での回転角度。

### 13.13.9.3. 拡大縮小

[拡大縮小]ウィジェット機能は、ウィジェットおよびサブツリーを拡大縮小するために使用します。

表13.226 [拡大縮小]ウィジェット機能のプロパティ

プロパティ名	説明
scalingEnabled	拡大縮小を使用するかどうかを定義します。
scalingX	X軸でのパーセント単位の拡大縮小
scalingY	Y軸でのパーセント単位の拡大縮小
scalingZ	ウィジェットがシーングラフである場合、z軸でのパーセント単位の拡大縮小

### 13.13.9.4. せん断

[せん断]ウィジェット機能は、ウィジェットとそのサブツリーを変形するために使用します。

表13.227 [せん断]ウィジェット機能のプロパティ

プロパティ名	説明
shearingEnabled	せん断を使用するかどうかを定義します。
shearingXbyY	Y軸によるX軸のせん断
shearingXbyZ	ウィジェットがシーングラフである場合、z軸によるX軸のせん断
shearingYbyX	X軸によるY軸のせん断
shearingYbyZ	ウィジェットがシーングラフである場合、z軸によるY軸のせん断
shearingZbyX	ウィジェットがシーングラフである場合、x軸によるZ軸のせん断
shearingZbyY	ウィジェットがシーングラフである場合、y軸によるZ軸のせん断

### 13.13.9.5. 変換

[変換]ウィジェット機能は、ウィジェットおよびサブツリーを変換するために使用します。これにより、ウィジェットはx、y、およびz方向に移動します。

表13.228 [変換]ウィジェット機能のプロパティ

プロパティ名	説明
translationEnabled	変換を使用するかどうかを定義します。
translationX	X軸での変換
translationY	Y軸での変換
translationZ	ウィジェットがシーングラフである場合、z軸での変換

## 14. EB GUIDE Studioのインストール

### 14.1. バックグラウンド情報

#### 14.1.1. 制限

**NOTE**



**互換性**

EB GUIDE product line 6は、以前のメジャーバージョンと一切互換性がありません。

**NOTE**



**EB GUIDE Speech Extension**

EB GUIDE Speech Extensionは、アドオン製品であり、購入しなければ使用できません。

**NOTE**



**ユーザーの権限**

EB GUIDEをWindows 7 またはWindows 10システムにインストールするには、管理者権限が必要です。

#### 14.1.2. システム要件

以下の構成を考慮してください。

Table 14.1. EB GUIDE Studioの推奨構成

ハードウェア	クアッドコアCPU(最低2 GHz)および8 GB RAMを搭載するPC
オペレーティングシステム	Windows 7(64ビット)、Windows 10(64ビット)
ディスプレイ解像度	1920x1080ピクセル以上 モニターは2台の使用を推奨
ソフトウェア	Microsoft .NET Framework 4.7



Table 14.2. EB GUIDE SDKの推奨構成

ソフトウェア開発キット	Microsoft Visual Studio 2013以降
ファイル統合	CMake

## 14.2. EB GUIDEのダウンロード

EB GUIDEのコミュニティエディションをダウンロードするには、<https://www.elektrobit.com/ebguide/try-eb-guide/>に移動して、手順に従います。

EB GUIDEのエンタープライズエディションをダウンロードするには、EB Commandに移動します。

### NOTE



アカウントをアクティブにする

製品を注文すると、営業担当者から電子メールが送られてきます。電子メールに埋め込まれたリンクをクリックします。電子メールとブラウザに記載された手順に従ってアカウントを作成してから、ログインに進みます。

EB Commandは、EB GUIDE product lineソフトウェアのダウンロード元となるサーバーです。EB Commandからダウンロードする手順については、<https://www.elektrobit.com/support/downloading-from-eb-command/>をご覧ください。

## 14.3. EB GUIDEのインストール



### EB GUIDEのインストール

Prerequisite:

- セットアップファイルstudio\_setup.exeのダウンロードが完了しています。
- オペレーティングシステムの管理者権限を持っていること。

#### Step 1

セットアップファイルstudio\_setup.exeをダブルクリックします。

ダイアログが開きます。

#### Step 2

Yesをクリックします。

Setup - EB GUIDE Studioダイアログが開きます。

### Step 3

ライセンス使用許諾契約書に同意し、**Next**をクリックします。

### Step 4

インストール先のディレクトリを選択します。

デフォルトのインストールディレクトリは、C:/Program Files/Elektrobit/EB GUIDE <version>です。

### Step 5

**Next**をクリックします。

要約ダイアログにすべての選択したインストール設定が表示されます。

### Step 6

表示されている設定でインストールを実行するには、**Install**をクリックします。

インストールが開始されます。

### Step 7

セットアップを終了するには、**Finish**をクリックします。

EB GUIDEのインストールはこれで完了です。

#### TIP



複数のインストール  
複数のEB GUIDEバージョンをインストールすることが可能です。

## 14.4. アンインストールする EB GUIDE



### アンインストールする EB GUIDE

#### 注記



#### EB GUIDEの永久的な削除

以下の手順に従うと、EB GUIDEがPCから永久的に削除されます。

前提条件:

- EB GUIDE がインストールされていること。
- オペレーティングシステムの管理者権限を持っていること。

#### ステップ 1

Windowsの[スタート]メニューで、[すべてのプログラム]をクリックします。



ステップ 2

[Elektrobit]メニューで、アンインストールするバージョンをクリックします。

ステップ 3

サブメニューで、[Uninstall]をクリックします。

# 用語集

## #

**3Dグラフィック** 3Dグラフィックは、3Dシーンの仮想画像です。3Dシーンは、3Dモデル(メッシュや形状)、材質、光源、カメラをまとめたものです。材質は色やテクスチャ、仮想ライト効果下での動作などで3Dモデルの外観を定義します。カメラは、3Dシーンの仮想画像を撮影する視点となります。

## A

**アプリケーションプログラミングインターフェース** アプリケーションプログラミングインターフェース

**アプリケーション** EB GUIDEのコンテキストでのアプリケーションとは、例えば、イベントシステムやデータプールなどを介して1つ以上のEB GUIDEモデルとのやり取りをEB GUIDE GTFのランタイムに行うコンピューターソフトウェアです。例えば、メディアプレーヤーのようなエンターテインメントソフトウェアや、電話のような通信ソフトウェアなどは、アプリケーションです。  
See Also アプリケーションプログラミングインターフェース。

**アスペクト** EB GUIDEでのアスペクトとは、EB GUIDE GTFのランタイムに適用される、EB GUIDEモデルの外観に関する変更です。EB GUIDEモデルに対してさまざまな外観を定義できるスキンと、言語という2つのタイプのアスペクトが存在します。

## C

**通信コンテキスト** 通信コンテキストは、通信が行われる環境を記述します。各通信コンテキストは、一意の数値IDによって識別されます。

## D

**データプール** データプールは、データプールアイテムへのアクセスをランタイムに提供するEB GUIDEモデル内のデータキャッシュです。アプリケーションとヒューマンマシンインターフェースとの間のデータ交換のために使用されます。

**データプールアイテム** データプールアイテムはデータを格納し、やり取りします。データプール内の各アイテムには通信方向があります。

## E

EB GUIDE GTF	EB GUIDE GTFは、EB GUIDE product lineのグラフィックターゲットフレームワークであり、EB GUIDE TFの一部です。EB GUIDE GTFは、対象デバイスでEB GUIDEモデルを実行するためのランタイム環境を表しています。
EB GUIDE GTF SDK	EB GUIDE GTF SDKはEB GUIDE GTFに含まれているSDKです。EB GUIDE SDKのサブセットにあたります。もう一つのサブセットがEB GUIDE Studio SDKです。
EB GUIDEモデル	EB GUIDEモデルは、EB GUIDE Studioで作成されたヒューマンマシンインターフェースの記述です。
EB GUIDE product line	EB GUIDE product lineは、ヒューマンマシンインターフェースモデルを記述したり、そのモデルを組み込み環境のシステムで動作するグラフィカルユーザーインターフェースに変換したりするために必要なソフトウェアライブラリおよびツールの集まりです。
EB GUIDEスクリプト	EB GUIDEスクリプトはEB GUIDE product lineのスクリプト記述言語です。EB GUIDEスクリプトを使用すると、データプール、モデル要素(ウィジェット、ステートマシンなど)、システムイベントにアクセスできます。
EB GUIDE SDK	EB GUIDE SDKは、EB GUIDEの製品コンポーネントであり、EB GUIDE product lineのソフトウェア開発キットです。EB GUIDE Studio SDKとEB GUIDE GTF SDKが含まれています。
EB GUIDE Studio	EB GUIDE Studioは、グラフィカルユーザーインターフェースによってヒューマンマシンインターフェースのモデリングや記述を行うためのツールです。
EB GUIDE Studio SDK	EB GUIDE Studio SDKは、EB GUIDE Studioと通信するためのアプリケーションプログラミングインターフェース(API)です。EB GUIDE SDKのサブセットにあたります。もう一つのサブセットがEB GUIDE GTF SDKです。
EB GUIDE TF	EB GUIDE TFは、EB GUIDEのランタイム環境であり、EB GUIDE GTFとEB GUIDE STFが含まれています。EB GUIDEモデルを実行するために必要です。
拡張機能	EB GUIDEでの拡張機能とは、任意のEB GUIDE製品に対する追加要素です。拡張機能は、特定の機能をEB GUIDE Studio、EB GUIDE GTF、またはEB GUIDE Monitorに追加するライブラリ(.dllまたは.soファイル)の形をとるプラグインです。そのような機能としては、例えば、データエクスポーターや追加のウィジェット機能が考えられます。

## G

GL	グラフィカルライブラリ
GUI	グラフィカルユーザーインターフェース

## H

HMI ヒューマンマシンインターフェース

## I

IBL イメージベースドライティング

IBLGenerator IBLGeneratorは、環境ライティング情報を処理するツールです。

## L

ライブラリ ライブラリとは、EB GUIDEで使用されるプリコンパイルされたソフトウェア部品、サブルーチン、またはプログラムのコレクションです。EB GUIDEプロジェクトに必要なライブラリは、プロジェクトセンターで定義されます。サポートされるファイルタイプとしては、.dllおよび.soの2種類があります。

## M

モデル要素 モデル要素とは、EB GUIDEモデル内のオブジェクト(例えば、ステート、ウィジェット、データプールアイテム)です。  
See Also EB GUIDEモデル.

マルチフォントサポート 異なる文字範囲に対する複数のフォントの集約であり、1つのフォントとして機能します。

MVC モデルビューコントローラ

## N

名前空間 EB GUIDE Studioでは、名前空間を使用して、データプールアイテムやイベントのようなモデル要素のグループを作成します。こうしたグループには通常、定義された機能があります。それぞれの名前空間では、異なる名前空間のモデル要素に同じ名前を持たせることができるように、モデル要素のネーミングスコープを作成しています。

## O

OS オペレーティングシステム

## P

PBR	物理ベースレンダリング
プロファイル	プロジェクトセンターにおいて、プロファイルは仕様の集まりです。プロファイルでプロジェクトのライブラリ、メッセージ、シーンを定義します。EB GUIDEモデルをエクスポートする際、プロファイルのデータはmodel.json設定ファイルに書き込まれます。
プロジェクトセンター	プロジェクトセンターには、プロファイルや言語など、すべてのプロジェクト関連機能があります。
プロジェクトエディター	プロジェクトエディターでは、ヒューマンマシンインターフェースの動作と外観をモデリングします。

## R

リソース	リソースとは、EB GUIDEプロジェクトの一部となるデータパッケージです。例えば、フォント、イメージ、メッシュなどがあります。オペレーティングシステムによっては、リソースがEB GUIDEモデルの外部(例えば、ファイル)に保存されています。
RomFS	読み取り専用メモリファイルシステム

## S

共有ライブラリ	共有ライブラリは静的ライブラリと異なり、実行用プログラムの準備中に読み込むことができます。Windowsプラットフォームでは、共有ライブラリはダイナミックリンクライブラリと呼ばれ、ファイル拡張子は.dllとなります。Unixシステムでは、共有ライブラリは共有オブジェクトと呼ばれ、ファイル拡張子は.soとなります。
ステート	ステートは、ステートマシンのステータスを定義したものです。ステートやステート遷移は、ステートチャートでモデリングされます。
ステートマシン	Sステートマシンとは、ステート、ステート間の遷移、動作の集合です。ステートマシンは、システムのダイナミックな動作を記述します。

## T

遷移	遷移は、あるステートから別のステートへの変化を定義したものです。通常、遷移はイベントによってトリガーされます。
----	---

## U

UI	ヒューマンマシンインターフェース
----	------------------

## V

**ビュー** ビューは、プロジェクト特有のヒューマンマシンインターフェース画面のグラフィカル表現であり、ステートマシンの特定のステートに関連しています。ビューは、ウィジェットのツリーで構成されます。

**VTA** ビュー遷移アニメーション

## W

**ウィジェット** ウィジェットは基本的なグラフィカル要素です。ウィジェットは、グラフィカルユーザーインターフェースとの対話処理のために使用されます。



# インデックス

## シンボル

.psdファイル形式, 153  
 3Dウィジェット, 37, 74, 76, 119  
     リファレンス, 358  
 3Dオブジェクト, 37  
 3Dグラフィック, 37, 74, 76, 255, 412  
     イメージベースドライティング, 74  
     インポート, 255  
     サポートされている形式, 37, 74, 76  
     メッシュ, 76  
     追加, 152  
 せん断  
     リファレンス, 406  
 アイコン  
     ヒューマンマシンインターフェース, 91  
 アクション  
     エントリーアクション, 127  
     終了アクション, 128  
     遷移, 138  
 アスペクト, 412  
 アニメーション, 40, 118, 148, 179, 249, 272  
     リファレンス, 345, 353  
     変更アニメーション, 41, 179  
     終了アニメーション, 41, 179  
     開始アニメーション, 40, 179  
 アニメーションの再配置, 180  
 アプリケーション, 412  
 アプリケーションプログラミングインターフェース, 42, 412  
 (see アプリケーションプログラミングインターフェース)  
 アルファマスク, 151  
     リファレンス, 344  
 アンビエントテクスチャ  
     リファレンス, 392  
 イベント, 62, 87  
     イベントID, 62  
     イベントグループ, 62  
     コピー, 185  
     リファレンス, 316  
     発行, 211

    貼り付け, 185  
     追加, 185  
 イベントシステム, 61  
 イメージ  
     9-patch, 75  
     サポートされている形式, 74  
     データタイプ, 281  
     リファレンス, 351  
     追加, 142  
 イメージベースドライティング, 74  
     レンダリング, 67  
 イメージベースドライト  
     リファレンス, 359  
 インスタンスエータ, 241  
     ラインテンプレート, 146, 351  
     リファレンス, 351  
     追加, 146  
 インポート  
     言語依存テキスト, 209  
 ウィジェット, 117, 416  
     3Dウィジェット, 118  
     アニメーション, 118  
     グループ, 145  
     サイズの変更, 158  
     削除, 156  
     基本, 118  
     追加, 141  
     配置, 157  
 ウィジェットテンプレート, 121, 181, 183  
 ウィジェットテンプレートインターフェース, 121  
 ウィジェットプロパティ, 120  
     EB GUIDEスクリプト, 85  
     ウィジェットテンプレート, 121  
     ウィジェットプロパティへのリンク, 159  
     ウィジェット機能プロパティ, 121  
     デフォルトプロパティ, 121  
     データプールアイテムへのリンク, 160  
     ユーザー定義プロパティ, 121, 162  
     追加, 162  
 ウィジェット機能, 120, 121, 122  
     パスジェスチャー, 238  
     削除, 172

- 追加, 170
- ウィンドウ表示リスト
  - データプールアイテム, 57
- エクスポート, 201
  - 言語依存テキスト, 207
- エミッシブテキストチャ
  - リファレンス, 394
- エントリーアクション, 132
  - ステートマシン, 127
- カメラ
  - リファレンス, 358
- カメラビューポート
  - リファレンス, 392
- ガンマ補正
  - レンダリング, 66
- キーUnicode
  - リファレンス, 380
- キーのステータス変更
  - リファレンス, 379
- キーリリース
  - リファレンス, 379
- キー押下
  - リファレンス, 378
- グリッドレイアウト
  - リファレンス, 387
- グローバルな名前変更, 199
- コピー
  - イベント, 185
  - データプールアイテム, 189
- コマンドエリア
  - プロジェクトエディター, 51
- コマンドライン, 91, 201, 202, 210
- コンスタント曲線
  - リファレンス, 346, 354
- コンソール (see コマンドライン)
- コンテナー
  - リファレンス, 350
  - 追加, 145
- コンテンツエリア
  - プロジェクトエディター, 48
  - プロジェクトセンター, 44
- コンポーネント
  - ドッキング, 53
  - ドッキング解除, 53
- シミュレーション, 201
- ショートカット
  - ヒューマンマシンインターフェース, 91
- シーングラフ, 37, 74, 76, 152, 255
  - テキストチャ, 255
  - リファレンス, 363
  - 追加, 152
- シーングラフノード
  - リファレンス, 364
- シーン設定
  - リファレンス, 335
- ジェスチャー, 115
  - パスジェスチャー, 115
  - リファレンス, 373, 378
  - 非パスジェスチャー, 115
- ジェスチャーID
  - リファレンス, 376
- スキン
  - サポート, 96
  - 切り替え, 176
  - 削除, 176
  - 追加, 176
- スクリプト値, 89, 190, 222
- スクリプト曲線, 41, 272, 348, 357
- ステート, 97, 128, 129, 227, 415
  - エントリーアクション, 132
  - ビューステート, 99
  - 初期ステート, 99
  - 履歴ステート, 102
  - 最終ステート, 100
  - 混合ステート, 97
  - 終了アクション, 132
  - 遷移, 133
  - 選択ステート, 101
- ステートマシン, 96, 415
  - UML 2.5記法, 113
  - UMLとの比較, 113
  - インクルードステートマシン, 96, 114
  - ステート, 97
  - ステートマシンの実行, 109

- ハプティックステートマシン, 96
- ロジックステートマシン, 96
- 削除, 128
- 動的ステートマシン, 96
- 追加, 126
- 遷移, 105
- スピン
  - リファレンス, 368
- スペキュラテクスチャ
  - リファレンス, 402
- スポットライト
  - リファレンス, 365
- タッチ
  - リファレンス, 369
- タッチのステータス変更
  - リファレンス, 384
- タッチの喪失
  - リファレンス, 382
- タッチジェスチャー (see ジェスチャー)
- タッチリリース
  - リファレンス, 384
- タッチ入力 (see ジェスチャー)
- タッチ押下
  - リファレンス, 383
- タッチ移動
  - リファレンス, 383
- ツールボックス
  - プロジェクトエディター, 47
- ツールボックスコンポーネント
  - プロジェクトエディター, 47
- テキストの切り捨て
  - リファレンス, 369
- テンプレート
  - 作成, 181
  - 使用, 183
  - 削除, 183
- テンプレートインターフェース, 182
  - プロパティの削除, 182
  - プロパティの追加, 182
- テンプレートインデックス
  - リファレンス, 391
- ディスプレイ
  - 設定, 206
- ディフューズテクスチャ
  - リファレンス, 393
- データタイプ
  - IBL, 281
  - イメージ, 281
  - フォント, 280
  - ブール値, 278
  - ブール値リスト, 278
  - メッシュ, 282
  - メッシュリスト, 282
  - リスト, 283
  - 整数, 281
  - 文字列, 282
  - 条件スクリプト, 279
  - 浮動小数点数, 280
  - 色, 279
- データプール, 57, 412
- データプールアイテム, 57, 190, 412
  - インポート, 209
  - ウィンドウ表示リスト, 57
  - エクスポート, 207
  - コピー, 189
  - リスト, 189
  - リファレンス, 278
  - リンク, 192
  - 変更, 212
  - 言語サポート, 251
  - 貼り付け, 189
  - 追加, 189
- トリガー
  - 遷移, 135
- トーンマッピング
  - リファレンス, 404
- ドッキング
  - コンポーネント, 53
- ドッキング解除
  - コンポーネント, 53
- ナビゲーションエリア
  - プロジェクトセンター, 44
- ナビゲーションコンポーネント
  - プロジェクトエディター, 45

- ノーマルマップテキストチャ  
リファレンス, 397
- パスジェスチャー, 238  
リファレンス, 375, 376
- ビュー, 117, 416  
リファレンス, 342  
追加, 140
- ビューステート  
リファレンス, 343
- ビューテンプレート  
リファレンス, 342, 343
- ビューポート  
リファレンス, 391
- ビュー遷移のアニメーション化, 179
- ピボット  
リファレンス, 405
- ピンチジェスチャー  
リファレンス, 377
- フォント, 72  
OpenTypeフォント, 72  
TrueTypeフォント, 72  
データタイプ, 280  
ビットマップフォント, 72, 73  
マルチフォントサポート, 73
- フォーカス  
リファレンス, 366
- フリックジェスチャー  
リファレンス, 373
- フローレイアウト  
リファレンス, 386
- ブール値  
データタイプ, 278
- ブール値リスト  
データタイプ, 279
- プラグイン (see 拡張機能)
- プロジェクトエディター, 44, 414  
VTAコンポーネント, 53  
コマンドエリア, 51  
コンテンツエリア, 48  
ツールボックス, 47  
ツールボックスコンポーネント, 47  
ナビゲーションコンポーネント, 45  
問題検出コンポーネント, 52
- プロジェクトセンター, 43, 414  
コンテンツエリア, 44  
ナビゲーションエリア, 44
- プロパティコンポーネント, 222  
コマンドエリア, 47  
プロジェクトエディター, 47
- プロファイル, 203, 414  
複製, 203  
追加, 203
- ベストプラクティス, 222
- ホールドジェスチャー  
リファレンス, 374
- ボタン  
ヒューマンマシンインターフェース, 91
- ボックスレイアウト  
リファレンス, 386
- ポップアップオフアニメーション  
リファレンス, 343
- ポップアップオンアニメーション  
リファレンス, 343
- マルチサンプリング, 338
- マルチタッチ入力, 116
- マルチフォントサポート, 73, 166, 414
- メッシュ, 76  
データタイプ, 282  
リファレンス, 360
- メッシュリスト  
データタイプ, 282
- モデル要素, 58, 414  
削除, 133
- ユーザー定義フォーカス  
リファレンス, 372
- ユーザー定義プロパティ, 162
- ライターアプリケーション, 42
- ライトマップテキストチャ  
リファレンス, 395
- ライブラリ, 414 (see 拡張機能)  
(see also 拡張機能)  
追加, 204
- ラインインデックス  
リファレンス, 390

- ラフネス(粗さ)テクスチャ
  - リファレンス, 400
- ラベル, 144
  - フォント, 144, 145
  - リファレンス, 352
  - 追加, 144
- リスト, 189
  - データタイプ, 283
  - 作成, 241
- リストインデックス
  - リファレンス, 391
- リストレイアウト
  - リファレンス, 388
- リソース, 415
  - .psdファイル形式, 71
  - 3Dグラフィック, 74, 76
  - イメージ, 74
  - イメージベースドライティング, 74
  - フォント, 72
  - メッシュ, 76
- リソース管理, 72
- リニア曲線, 349, 357
- リニア補間整数, 249
- リニア補間曲線, 349, 357
- リフレクションテクスチャ
  - リファレンス, 399
- リンク
  - ウィジェットプロパティ, 159, 161
  - データプールアイテム, 192
- リーダーアプリケーション, 42
- レイアウト余白
  - リファレンス, 388
- レンダラー
  - 設定, 206
- ロングホールドジェスチャー
  - リファレンス, 374
- 不透明テクスチャ
  - リファレンス, 398
- 二次曲線
  - リファレンス, 347, 355
- 低速開始曲線
  - リファレンス, 347, 355
- 光沢テクスチャ
  - リファレンス, 401
- 共有ライブラリ, 415
- 内部で移動
  - リファレンス, 381
- 内部へ移動
  - リファレンス, 380
- 内部遷移, 139
- 効果
  - ウィジェット機能, 370
- 動的ステートマシン
  - 追加, 126, 223
- 可動
  - リファレンス, 381
- 可視性, 365
- 可視性グループ, 365
- 名前空間, 70, 194, 414
- 問題検出コンポーネント, 200
  - プロジェクトエディター, 52
- 四角形
  - リファレンス, 352
- 回転
  - リファレンス, 382, 405
- 回転ジェスチャー
  - リファレンス, 377
- 基本ウィジェット, 118
  - リファレンス, 343
- 変換
  - リファレンス, 406
- 変更アニメーション, 179
  - リファレンス, 343
- 変更アニメーションの追加, 179
- 外部へ移動
  - リファレンス, 381
- 子の可視性の選択
  - リファレンス, 365
- 押下
  - リファレンス, 367
- 拡大縮小
  - リファレンス, 406
- 拡大縮小モード
  - リファレンス, 390

- 拡張機能, 413
  - 指向性ライト
    - リファレンス, 359
  - 整数
    - データタイプ, 282
  - 文字列
    - データタイプ, 282
  - 有効
    - リファレンス, 366
  - 材質
    - PBR GGX材質, 361, 414
    - PBR Phong材質, 362, 414
    - リファレンス, 360, 361, 362
  - 条件
    - 遷移, 136
  - 条件スクリプト
    - データタイプ, 279
  - 枠
    - リファレンス, 370
  - 楕円
    - リファレンス, 350
  - 正弦曲線
    - リファレンス, 348, 356
  - 浮動小数点数
    - データタイプ, 280
  - 混合ステート, 129
  - 点ライト
    - リファレンス, 363
  - 環境光
    - リファレンス, 358
  - 終了アクション, 132
    - ステートマシン, 128
  - 終了アニメーション, 179
    - リファレンス, 343
  - 絶対レイアウト
    - リファレンス, 385
  - 自動フォーカス
    - リファレンス, 372
  - 自動非表示, 53
  - 色
    - データタイプ, 279
  - 表示遷移アニメーション, 263, 416
  - 複数行
    - リファレンス, 366
  - 言語
    - 変更, 251
    - 言語サポート, 173
    - 言語依存テキスト, 251
      - インポート, 209
      - エクスポート, 207
  - 設定
    - ディスプレイ, 206
    - 設定ファイル, 318, 331, 339
  - 貼り付け
    - イベント, 185
      - データプールアイテム, 189
    - 通信コンテキスト, 42, 191, 412
  - 遷移, 105, 133, 263, 415
    - アクション, 137
    - トリガー, 135
      - 内部, 139
      - 条件, 136
      - 移動, 134
      - 追加, 133
  - 選択
    - リファレンス, 367
  - 選択グループ
    - リファレンス, 368
  - 選択ステート, 130
  - 配色
    - リファレンス, 371
  - 金属テクスチャ
    - リファレンス, 396
  - 開始アニメーション, 179
    - リファレンス, 343
  - 開始アニメーションの追加, 179
  - 高速開始曲線
    - リファレンス, 346, 354
- ## E
- EB GUIDE GTF, 412
  - EB GUIDE GTF SDK, 413
  - EB GUIDE GTF拡張機能, 64 (see 拡張機能)

- EB GUIDE Monitor, 54, 201, 211, 211, 212, 213, 218, 219, 220
- monitor.cfg, 219
  - イベント, 211
  - イベントコンポーネント, 211
  - ウォッチリスト, 220
  - ウォッチリストをインポート, 220
  - ウォッチリストをエクスポート, 220
  - コマンドライン, 218
  - スクリプトコンポーネント, 213
  - スタンドアロン, 218
  - タブ, 54
  - データプールアイテム, 212
  - データプールコンポーネント, 212
  - 設定, 219
- EB GUIDE Monitor拡張機能, 64 (see 拡張機能)
- EB GUIDE product line, 413
- EB GUIDE SDK, 413
- EB GUIDE Studio, 413
- EB GUIDE Studio SDK, 413
- EB GUIDE Studio拡張機能 (see 拡張機能)
- EB GUIDE TF, 413
- EB GUIDEスクリプト, 76, 190, 222, 413
- if-then-else, 82
  - L値, 80
  - R値, 80
  - Whileループ, 82
  - イベント, 87
  - ウィジェットプロパティ, 85
  - コメント, 77
  - スクリプト値, 89
  - チュートリアル, 231
  - データプールアクセス, 84
  - データ型, 78
  - プレフィックス, 76
  - リスト, 86
  - ローカル変数, 81
  - 外部関数呼び出し, 83
  - 式, 78
  - 文字列の書式設定, 89
  - 標準ライブラリ, 89
  - 識別子, 77
- EB GUIDEプロジェクト, 58
- EB GUIDEモデル, 58, 413
- モデル要素, 58
- EB GUIDE拡張機能, 64
- ## F
- finger ID, 116
- ## G
- GL, 413
- GUI, 413
- ## H
- HMI, 414
- ## I
- IBL, 67, 414
- データタイプ, 281
- IBLGenerator, 414
- イメージベースドライティング, 67
- ## M
- model.json, 318
- ## O
- OS, 414
- ## P
- platform.json, 331
- PSD, 71, 153
- ## R
- RomFS, 415
- ## T
- todo
- EB GUIDEスクリプト, 77
- tracing.json, 339
- ## U
- UI, 415

## V

VTA, 179, 263, 416

VTAコンポーネント

プロジェクトエディター, 53