



Elektrobit

# EB GUIDE Studio

사용 설명서  
버전 6.7.3.150087



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.

# 차례

1. 이 문서 정보 .....	16
1.1. 타깃 오디언스: 모델러 .....	16
1.2. 사용자 문서 구조 .....	16
1.3. 입력 체계와 스타일 규칙 .....	17
1.4. 명명 규칙 .....	19
1.5. 경로 규칙 .....	19
2. 안전하고 올바른 사용법 .....	20
2.1. 용도 .....	20
2.2. 오용 가능성 .....	20
3. 지원 .....	21
4. EB GUIDE 소개 .....	22
4.1. EB GUIDE product line의 소프트웨어 요소 .....	22
4.2. EB GUIDE Studio .....	22
4.2.1. HMI 동작 모델링 .....	22
4.2.2. HMI 모양 모델링 .....	23
4.2.3. 데이터 처리 .....	23
4.2.4. EB GUIDE 모델 시뮬레이션 .....	23
4.2.5. EB GUIDE 모델 내보내기 .....	24
4.3. EB GUIDE TF .....	24
5. 자습서: 시작하기 .....	26
5.1. EB GUIDE 시작하기 .....	26
5.2. 프로젝트 만들기 .....	27
5.3. HMI 동작 모델링 .....	28
5.4. HMI 모양 모델링 .....	31
5.5. 시뮬레이션 시작 .....	34
6. 배경 정보 .....	35
6.1. 3D 그래픽 .....	35
6.1.1. 지원되는 3D 그래픽 형식 .....	35
6.1.2. 3D 그래픽 파일의 설정 .....	35
6.1.3. 3D 그래픽 파일 가져오기 .....	36
6.2. 애니메이션 .....	37
6.2.1. 위젯 애니메이션 .....	38
6.2.2. 보기 전이 애니메이션 .....	38
6.3. 응용 프로그램과 모델 간의 응용 프로그래밍 인터페이스 .....	39
6.4. 통신 컨텍스트 .....	40
6.5. 그래픽 사용자 인터페이스의 구성 요소 .....	40
6.5.1. 프로젝트 센터 .....	40
6.5.1.1. 내비게이션 영역 .....	41
6.5.1.2. 콘텐츠 영역 .....	41

6.5.2. 프로젝트 편집자 .....	41
6.5.2.1. 내비게이션 구성 요소 .....	43
6.5.2.2. 개요 구성 요소 .....	44
6.5.2.3. 도구 상자 구성 요소 .....	44
6.5.2.4. 속성 구성 요소 .....	45
6.5.2.5. 콘텐츠 영역 .....	45
6.5.2.6. 이벤트 구성 요소 .....	47
6.5.2.7. 데이터폴 구성 요소 .....	47
6.5.2.8. 자산 구성 요소 .....	47
6.5.2.9. 명령 영역 .....	48
6.5.2.10. 문제 구성 요소 .....	49
6.5.2.11. VTA 구성 요소 .....	49
6.5.3. 도킹할 수 있는 구성 요소 .....	49
6.5.4. EB GUIDE Monitor .....	50
6.6. 데이터폴 .....	52
6.6.1. 개념 .....	52
6.6.2. 데이터폴 항목 .....	52
6.6.3. 창 모드 목록 .....	53
6.7. EB GUIDE 모델 및 EB GUIDE 프로젝트 .....	53
6.8. 이벤트 처리 .....	54
6.8.1. 이벤트 시스템 .....	54
6.8.2. 이벤트 .....	54
6.9. 확장 .....	57
6.9.1. EB GUIDE Studio 확장 .....	57
6.9.2. EB GUIDE GTF 확장 .....	57
6.10. 감마 정정 렌더링 .....	57
6.10.1. 개념 .....	57
6.10.2. EB GUIDE Studio에서의 감마 정정 .....	58
6.11. 이미지 기반 조명 .....	59
6.11.1. 개념 .....	59
6.11.2. 제한 .....	59
6.12. 언어 .....	60
6.12.1. EB GUIDE Studio의 표시 언어 .....	60
6.12.2. EB GUIDE 모델의 언어 .....	60
6.12.3. 언어별로 다른 텍스트 내보내기 및 가져오기 .....	61
6.13. Photoshop 파일 형식 지원 .....	61
6.14. 리소스 관리 .....	61
6.14.1. 글꼴 .....	62
6.14.1.1. 비트맵 글꼴 .....	62
6.14.2. 3D 그래픽을 위한 이미지 기반 조명 .....	62
6.14.3. 이미지 .....	63
6.14.3.1. 나인패치 이미지 .....	63

6.14.4. 3D 그래픽의 메시 .....	64
6.15. 스크립팅 언어 EB GUIDE 스크립트 .....	64
6.15.1. 응용 프로그램의 기능과 영역 .....	64
6.15.2. 네임스페이스 및 식별자 .....	65
6.15.3. 주석 .....	66
6.15.4. 유형 .....	66
6.15.5. 식 .....	67
6.15.6. 상수 및 참조 .....	67
6.15.7. 산술 식과 논리 식 .....	68
6.15.8. l 값과 r 값 .....	69
6.15.9. 로컬 변수 .....	69
6.15.10. while 루프 .....	70
6.15.11. if-then-else .....	70
6.15.12. 외래 함수 호출 .....	72
6.15.13. 데이터폴 액세스 .....	72
6.15.14. 위젯 속성 .....	73
6.15.15. 목록 .....	74
6.15.16. 이벤트 .....	75
6.15.17. 문자열 서식 지정 .....	77
6.15.18. 표준 라이브러리 .....	77
6.16. 스크립팅된 값 .....	77
6.17. 바로가기, 버튼 및 아이콘 .....	79
6.17.1. 바로가기 .....	79
6.17.2. 명령줄 옵션 .....	80
6.17.2.1. Studio.Console.exe의 명령줄 옵션 .....	80
6.17.2.2. Monitor.Console.exe의 명령줄 옵션 .....	81
6.17.3. 버튼 .....	82
6.17.4. 아이콘 .....	83
6.18. 스킨 .....	83
6.19. 상태 머신 및 상태 .....	84
6.19.1. 상태 머신 .....	84
6.19.1.1. 햅틱 상태 머신 .....	84
6.19.1.2. 논리적 상태 머신 .....	84
6.19.1.3. 동적 상태 머신 .....	84
6.19.2. 상태 .....	85
6.19.2.1. 복합 상태 .....	85
6.19.2.2. 보기 상태 .....	86
6.19.2.3. 초기 상태 .....	86
6.19.2.4. 최종 상태 .....	87
6.19.2.5. 선택 상태 .....	88
6.19.2.6. 이력 상태 .....	89
6.19.3. 전이 .....	92

6.19.4. 상태 기계 실행 .....	96
6.19.5. UML 표기법과 비교한 EB GUIDE 표기법 .....	100
6.19.5.1. 지원되는 요소 .....	101
6.19.5.2. 지원되지 않는 요소 .....	101
6.19.5.3. 차이점 .....	101
6.20. 터치 입력 .....	102
6.20.1. 비경로 제스처 .....	102
6.20.2. 경로 제스처 .....	102
6.20.3. 입력 처리 및 제스처 .....	103
6.20.4. 멀티 터치 입력 .....	103
6.21. 위젯 .....	104
6.21.1. 보기 .....	104
6.21.2. 위젯 범주 .....	105
6.21.3. 위젯 속성 .....	107
6.21.4. 위젯 템플릿 .....	108
6.21.5. 위젯 피쳐 .....	109
6.21.5.1. 포커스 위젯 피쳐 범주 .....	110
6.21.5.2. 목록 관리 위젯 피쳐 범주 .....	111
7. HMI 동작 모델링 .....	113
7.1. 상태 머신 모델링 .....	113
7.1.1. 상태 머신 추가 .....	113
7.1.2. 동적 상태 머신 추가 .....	113
7.1.3. 상태 머신에 대해 진입 작업 정의 .....	114
7.1.4. 상태 머신에 대해 종료 작업 정의 .....	114
7.1.5. 상태 머신 삭제 .....	115
7.2. 상태 모델링 .....	115
7.2.1. 상태 추가 .....	115
7.2.2. 복합 상태에 상태 추가 .....	116
7.2.3. 선택 상태 추가 .....	117
7.2.4. 상태에 대해 진입 작업 정의 .....	118
7.2.5. 상태에 대해 종료 작업 정의 .....	119
7.2.6. 상태 머신에서 모델 요소 삭제 .....	120
7.3. 전이를 통해 상태 연결 .....	120
7.3.1. 두 상태 간에 전이 추가 .....	120
7.3.2. 전이 이동 .....	121
7.3.3. 전이에 대해 트리거 정의 .....	122
7.3.4. 전이에 조건 추가 .....	123
7.3.5. 전이에 작업 추가 .....	124
7.3.6. 상태에 내부 전이 추가 .....	126
8. HMI 모양 모델링 .....	127
8.1. 위젯으로 작업 .....	127
8.1.1. 보기 추가 .....	127

8.1.2. 보기에 기본 위젯 추가 .....	128
8.1.2.1. 사각형 추가 .....	128
8.1.2.2. 타원 추가 .....	128
8.1.2.2.1. 타원 편집 .....	128
8.1.2.3. 이미지 추가 .....	129
8.1.2.4. 레이블 추가 .....	131
8.1.2.4.1. 레이블의 글꼴 변경 .....	132
8.1.2.5. 컨테이너 추가 .....	132
8.1.2.6. 인스턴스화기 추가 .....	133
8.1.2.7. 애니메이션 추가 .....	135
8.1.2.8. 알파 마스크 추가 .....	136
8.1.3. 보기에 3D 위젯 추가 .....	137
8.1.3.1. 보기에 장면 그래프 추가 .....	137
8.1.4. 보기에 .psd 파일 추가 .....	138
8.1.5. 보기에서 위젯 삭제 .....	139
8.2. 위젯 속성으로 작업 .....	140
8.2.1. 위젯 배치 .....	140
8.2.2. 위젯 크기 조정 .....	141
8.2.3. 위젯 속성을 서로 연결 .....	142
8.2.4. 데이터폴 항목에 위젯 속성 연결 .....	144
8.2.5. 위젯에 사용자 정의 속성 추가 .....	145
8.2.5.1. Function (): bool 유형의 사용자 정의 속성 추가 .....	146
8.2.6. 사용자 정의 속성 이름 바꾸기 .....	147
8.3. 위젯 피쳐로 위젯 확장 .....	148
8.3.1. 위젯 피쳐 추가 .....	148
8.3.2. 위젯 피쳐 제거 .....	150
8.4. EB GUIDE 모델에 언어 추가 .....	151
8.4.1. 언어 추가 .....	152
8.4.2. 언어 삭제 .....	153
8.5. 스킨 지원으로 작업 .....	153
8.5.1. EB GUIDE 모델에 스킨 추가 .....	154
8.5.2. 데이터폴 항목에 스킨 지원 추가 .....	154
8.5.3. 스킨 간 전환 .....	155
8.5.4. 스킨 삭제 .....	156
8.6. 관점 전환 애니메이션 생성 .....	156
8.6.1. 진입 애니메이션 추가 .....	156
8.6.2. 변경 애니메이션 추가 .....	157
8.6.3. 애니메이션 재배열 .....	158
8.7. 위젯 다시 사용 .....	158
8.7.1. 템플릿 추가 .....	158
8.7.2. 템플릿 인터페이스 정의 .....	159
8.7.3. 템플릿 사용 .....	160

8.7.4. 템플릿 삭제 .....	161
9. 데이터 처리 .....	162
9.1. 이벤트 추가 .....	162
9.2. 이벤트에 매개변수 추가 .....	162
9.3. 이벤트 주소 지정 .....	163
9.4. 이벤트 삭제 .....	164
9.5. 데이터폴 항목 추가 .....	164
9.6. 목록 유형의 데이터폴 항목 편집 .....	165
9.7. 속성을 스크립팅된 값으로 변환 .....	165
9.8. 외부 통신 설정 .....	166
9.9. 데이터폴 항목을 서로 연결 .....	167
9.10. 데이터폴 항목 삭제 .....	169
10. 프로젝트 처리 .....	170
10.1. 프로젝트 만들기 .....	170
10.2. 프로젝트 열기 .....	170
10.2.1. 파일 탐색기에서 프로젝트 열기 .....	170
10.2.2. EB GUIDE Studio 내에서 프로젝트 열기 .....	171
10.3. 모델 요소, 데이터폴 항목 및 이벤트 이름 바꾸기 .....	171
10.4. EB GUIDE 모델 유효성 검사 및 시뮬레이션 .....	172
10.4.1. EB GUIDE 모델 유효성 검사 .....	173
10.4.1.1. EB GUIDE Studio를 사용하여 EB GUIDE 모델 유효성 검사 .....	173
10.4.1.2. 명령줄을 사용하여 EB GUIDE 모델 유효성 검사 .....	174
10.4.2. 시뮬레이션 시작 및 중지 .....	174
10.5. EB GUIDE 모델 내보내기 .....	174
10.5.1. EB GUIDE Studio를 사용하여 EB GUIDE 모델 내보내기 .....	175
10.5.2. 명령줄을 사용하여 EB GUIDE 모델 내보내기 .....	175
10.6. EB GUIDE Studio의 표시 언어 변경 .....	176
10.7. 프로필 구성 .....	176
10.7.1. 프로필 복제 .....	176
10.7.2. 라이브러리 추가 .....	177
10.7.3. 장면 구성 .....	179
10.8. 언어별로 다른 텍스트 내보내기 및 가져오기 .....	180
10.8.1. 언어별로 다른 텍스트 내보내기 .....	180
10.8.2. 언어별로 다른 텍스트 가져오기 .....	181
10.8.2.1. EB GUIDE Studio를 사용하여 언어별로 다른 텍스트 가져오기 .....	182
10.8.2.2. 명령줄을 사용하여 언어별로 다른 텍스트 가져오기 .....	182
10.9. EB GUIDE Monitor로 작업 .....	183
10.9.1. EB GUIDE Monitor에서 이벤트를 실행합니다. ....	183
10.9.2. EB GUIDE Monitor로 데이터폴 항목의 값 변경 .....	184
10.9.3. EB GUIDE Monitor에서 스크립트 시작 .....	185
10.9.3.1. EB GUIDE Monitor에 대한 스크립트 파일 쓰기 .....	186
10.9.4. EB GUIDE Monitor를 독립 실행형 응용 프로그램으로 시작 .....	189



10.9.5. 구성을 EB GUIDE Monitor로 로드 .....	190
10.9.6. 감시 목록 내보내기 및 가져오기 .....	191
11. 자습서 .....	194
11.1. 자습서: 동적 상태 머신 추가 .....	194
11.2. 자습서: EB GUIDE 스크립트를 사용하여 버튼 동작 모델링 .....	202
11.3. 자습서: 경로 제스처 모델링 .....	209
11.4. 자습서: 동적 콘텐츠로 목록 만들기 .....	212
11.5. 자습서: 화면을 가로지르는 타원 이동 생성 .....	219
11.6. 자습서: 데이터폴 항목에 언어별로 다른 텍스트 추가 .....	222
11.7. 자습서: 3D 그래픽으로 작업 .....	225
11.8. 자습서: 감마를 올바르게 렌더링하기 .....	230
11.9. 자습서: 이미지 기반 조명 사용 .....	233
12. 참조 .....	236
12.1. Android 이벤트 .....	236
12.2. 데이터폴 항목 .....	237
12.3. 데이터 유형 .....	237
12.3.1. 부울 .....	237
12.3.2. 색 .....	238
12.3.3. 조건부 스크립트 .....	238
12.3.4. 부동 .....	239
12.3.5. 글꼴 .....	239
12.3.6. Function () : bool .....	240
12.3.7. lbl .....	240
12.3.8. 이미지 .....	240
12.3.9. 정수 .....	240
12.3.10. 메시 .....	241
12.3.11. 문자열 .....	241
12.3.12. 목록 .....	242
12.4. EB GUIDE 스크립트 .....	243
12.4.1. EB GUIDE 스크립트 키워드 .....	243
12.4.2. EB GUIDE 스크립트 연산자 우선순위 .....	244
12.4.3. EB GUIDE 스크립트 표준 라이브러리 .....	244
12.4.3.1. EB GUIDE 스크립트 함수 A - B .....	245
12.4.3.1.1. abs .....	245
12.4.3.1.2. absf .....	245
12.4.3.1.3. acosf .....	245
12.4.3.1.4. animation_before .....	245
12.4.3.1.5. animation_beyond .....	246
12.4.3.1.6. animation_cancel .....	246
12.4.3.1.7. animation_cancel_end .....	246
12.4.3.1.8. animation_cancel_reset .....	246
12.4.3.1.9. animation_pause .....	247

12.4.3.1.10. animation_play .....	247
12.4.3.1.11. animation_reverse .....	247
12.4.3.1.12. animation_running .....	247
12.4.3.1.13. animation_set_time .....	248
12.4.3.1.14. asinf .....	248
12.4.3.1.15. atan2f .....	248
12.4.3.1.16. atan2i .....	249
12.4.3.1.17. atanf .....	249
12.4.3.1.18. bool2string .....	249
12.4.3.2. EB GUIDE 스크립트 함수 C - H .....	249
12.4.3.2.1. ceil .....	249
12.4.3.2.2. changeDynamicStateMachinePriority .....	250
12.4.3.2.3. character2unicode .....	250
12.4.3.2.4. character2unicode .....	250
12.4.3.2.5. clampf .....	251
12.4.3.2.6. clampi .....	251
12.4.3.2.7. clearAllDynamicStateMachines .....	251
12.4.3.2.8. color2string .....	251
12.4.3.2.9. cosf .....	252
12.4.3.2.10. deg2rad .....	252
12.4.3.2.11. expf .....	252
12.4.3.2.12. float2string .....	253
12.4.3.2.13. floor .....	253
12.4.3.2.14. fmod .....	253
12.4.3.2.15. focusMoveTo .....	253
12.4.3.2.16. focusNext .....	254
12.4.3.2.17. focusPrevious .....	254
12.4.3.2.18. format_float .....	254
12.4.3.2.19. format_int .....	255
12.4.3.2.20. frac .....	256
12.4.3.2.21. getConfigItem .....	256
12.4.3.2.22. getFontAscender .....	256
12.4.3.2.23. getFontDescender .....	257
12.4.3.2.24. getFontLineGap .....	257
12.4.3.2.25. getImageHeight .....	257
12.4.3.2.26. getImageWidth .....	257
12.4.3.2.27. getLabelTextHeight .....	258
12.4.3.2.28. getLabelTextWidth .....	258
12.4.3.2.29. getLineCount .....	258
12.4.3.2.30. getLineHeight .....	258

12.4.3.2.31. getProductString .....	259
12.4.3.2.32. getTextHeight .....	259
12.4.3.2.33. getLength .....	259
12.4.3.2.34. getTextWidth .....	260
12.4.3.2.35. getVersionString .....	260
12.4.3.2.36. has_list_window .....	260
12.4.3.2.37. hsba2color .....	261
12.4.3.3. EB GUIDE 스크립트 함수 I - R .....	261
12.4.3.3.1. int2float .....	261
12.4.3.3.2. int2string .....	261
12.4.3.3.3. isDynamicStateMachineActive .....	262
12.4.3.3.4. isWidgetOnActiveStatemachine .....	262
12.4.3.3.5. language .....	262
12.4.3.3.6. lerp .....	262
12.4.3.3.7. localtime_day .....	263
12.4.3.3.8. localtime_hour .....	263
12.4.3.3.9. localtime_minute .....	263
12.4.3.3.10. localtime_month .....	264
12.4.3.3.11. localtime_second .....	264
12.4.3.3.12. localtime_weekday .....	264
12.4.3.3.13. localtime_year .....	264
12.4.3.3.14. log10f .....	265
12.4.3.3.15. logf .....	265
12.4.3.3.16. maxf .....	265
12.4.3.3.17. maxi .....	265
12.4.3.3.18. minf .....	266
12.4.3.3.19. mini .....	266
12.4.3.3.20. nearbyint .....	266
12.4.3.3.21. popDynamicStateMachine .....	266
12.4.3.3.22. powf .....	267
12.4.3.3.23. pushDynamicStateMachine .....	267
12.4.3.3.24. rad2deg .....	267
12.4.3.3.25. rand .....	268
12.4.3.3.26. rgba2color .....	268
12.4.3.3.27. round .....	268
12.4.3.4. EB GUIDE 스크립트 함수 S - W .....	268
12.4.3.4.1. saturate .....	268
12.4.3.4.2. seed_rand .....	269
12.4.3.4.3. shutdown .....	269
12.4.3.4.4. sinf .....	269

12.4.3.4.5. skin .....	269
12.4.3.4.6. smoothstep .....	270
12.4.3.4.7. sqrtf .....	270
12.4.3.4.8. string2float .....	270
12.4.3.4.9. string2int .....	271
12.4.3.4.10. string2string .....	271
12.4.3.4.11. substring .....	271
12.4.3.4.12. system_time .....	272
12.4.3.4.13. system_time_ms .....	272
12.4.3.4.14. tanf .....	272
12.4.3.4.15. trace_dp .....	272
12.4.3.4.16. trace_string .....	273
12.4.3.4.17. transformToScreenX .....	273
12.4.3.4.18. transformToScreenY .....	273
12.4.3.4.19. transformToWidgetX .....	274
12.4.3.4.20. transformToWidgetY .....	274
12.4.3.4.21. trunc .....	274
12.4.3.4.22. widgetGetChildCount .....	275
12.5. 이벤트 .....	275
12.5.1. 핵심 이벤트를 위한 십진 코드 .....	275
12.6. model.json 구성 파일 .....	277
12.6.1. EB GUIDE Studio의 예제 model.json .....	284
12.7. OpenGL ES 확장 .....	285
12.8. platform.json 구성 파일 .....	287
12.8.1. EB GUIDE Studio의 예제 platform.json .....	289
12.9. 장면 .....	290
12.10. EB GUIDE GTF에서 지원하는 터치 스크린 유형 .....	292
12.11. tracing.json 구성 파일 .....	293
12.11.1. 예제 tracing.json .....	295
12.12. 위젯 .....	295
12.12.1. 보기 .....	295
12.12.2. 기본 위젯 .....	296
12.12.2.1. 알파 마스크 .....	297
12.12.2.2. 애니메이션 .....	298
12.12.2.2.1. 상수 커브 .....	299
12.12.2.2.2. 빠른 시작 커브 .....	299
12.12.2.2.3. 느린 시작 커브 .....	299
12.12.2.2.4. 정방향 커브 .....	300
12.12.2.2.5. 공동 커브 .....	300
12.12.2.2.6. 스크립트 커브 .....	301
12.12.2.2.7. 선형 커브 .....	302

12.12.2.2.8. 선형 보간 커브 .....	302
12.12.2.3. 컨테이너 .....	303
12.12.2.4. 타원 .....	303
12.12.2.5. 이미지 .....	304
12.12.2.6. 인스턴스화기 .....	304
12.12.2.7. 레이블 .....	305
12.12.2.8. 사각형 .....	305
12.12.3. 애니메이션 .....	306
12.12.3.1. 애니메이션 .....	306
12.12.3.2. 상수 커브 .....	307
12.12.3.3. 빠른 시작 커브 .....	307
12.12.3.4. 느린 시작 커브 .....	308
12.12.3.5. 정방향 커브 .....	308
12.12.3.6. 공동 커브 .....	309
12.12.3.7. 스크립트 커브 .....	310
12.12.3.8. 선형 커브 .....	310
12.12.3.9. 선형 보간 커브 .....	310
12.12.4. 3D 위젯 .....	311
12.12.4.1. 주변 조명 .....	311
12.12.4.2. 카메라 .....	312
12.12.4.3. 방향 조명 .....	312
12.12.4.4. 이미지 기반 조명 .....	312
12.12.4.5. 재료 .....	313
12.12.4.6. 메시 .....	313
12.12.4.7. PBR GGX 재료 .....	313
12.12.4.8. PBR Phong 재료 .....	314
12.12.4.9. 포인트 조명 .....	315
12.12.4.10. 장면 그래프 .....	316
12.12.4.11. 장면 그래프 노드 .....	316
12.12.4.12. 스포트 조명 .....	317
12.13. 위젯 피쳐 .....	317
12.13.1. 일반 .....	317
12.13.1.1. 하위 가시성 선택 .....	317
12.13.1.2. 활성화됨 .....	318
12.13.1.3. 포커스됨 .....	318
12.13.1.4. 여러 줄 .....	318
12.13.1.5. 누름 .....	319
12.13.1.6. 선택됨 .....	319
12.13.1.7. 선택 그룹 .....	320
12.13.1.8. 스피닝 .....	320
12.13.1.9. 텍스트 잘림 .....	321
12.13.1.10. 터치함 .....	321

12.13.2. 효과 .....	322
12.13.2.1. 테두리 .....	322
12.13.2.2. 천연색 .....	323
12.13.2.3. 스트로크 .....	323
12.13.3. 포커스 .....	324
12.13.3.1. 자동 포커스 .....	324
12.13.3.2. 사용자 정의 포커스 .....	324
12.13.4. 제스처 .....	325
12.13.4.1. 굿기 제스처 .....	325
12.13.4.2. 누르기 제스처 .....	326
12.13.4.3. 길게 누르기 제스처 .....	326
12.13.4.4. 경로 제스처 .....	327
12.13.4.4.1. 제스처 ID .....	328
12.13.4.5. 모으기 제스처 .....	329
12.13.4.6. 회전 제스처 .....	329
12.13.5. 입력 처리 .....	330
12.13.5.1. 제스처 .....	330
12.13.5.2. 키 누름 .....	330
12.13.5.3. 키 놓음 .....	331
12.13.5.4. 키 상태 변경됨 .....	331
12.13.5.5. 키 유니코드 .....	332
12.13.5.6. 안으로 이동 .....	332
12.13.5.7. 밖으로 이동 .....	332
12.13.5.8. 위로 이동 .....	333
12.13.5.9. 이동 가능 .....	333
12.13.5.10. 회전식 .....	334
12.13.5.11. 터치 잡기 놓침 .....	334
12.13.5.12. 터치 이동 .....	335
12.13.5.13. 터치 누름 .....	335
12.13.5.14. 터치 놓음 .....	336
12.13.5.15. 터치 상태 변경됨 .....	336
12.13.6. 레이아웃 .....	337
12.13.6.1. 절대 레이아웃 .....	337
12.13.6.2. 상자 레이아웃 .....	338
12.13.6.3. 흐름 레이아웃 .....	338
12.13.6.4. 눈금 레이아웃 .....	339
12.13.6.5. 레이아웃 여백 .....	340
12.13.6.6. 목록 레이아웃 .....	340
12.13.6.7. 크기 조정 모드 .....	341
12.13.7. 목록 관리 .....	342
12.13.7.1. 줄 인덱스 .....	342
12.13.7.2. 목록 인덱스 .....	342

12.13.7.3. 템플릿 인덱스 .....	343
12.13.7.4. 뷰포트 .....	343
12.13.8. 3D .....	343
12.13.8.1. 카메라 뷰포트 .....	343
12.13.8.2. 주변 질감 .....	344
12.13.8.3. 확산 질감 .....	345
12.13.8.4. 방사 질감 .....	346
12.13.8.5. 조명 지도 질감 .....	346
12.13.8.6. 금속 질감 .....	347
12.13.8.7. 일반 지도 질감 .....	348
12.13.8.8. 불투명 질감 .....	349
12.13.8.9. 반사 질감 .....	350
12.13.8.10. 거친 질감 .....	351
12.13.8.11. 광택 질감 .....	352
12.13.8.12. 거울 질감 .....	353
12.13.8.13. 톤 매핑 .....	354
12.13.9. 변환 .....	355
12.13.9.1. 피벗 .....	355
12.13.9.2. 회전 .....	355
12.13.9.3. 크기 조정 .....	356
12.13.9.4. 전단 .....	356
12.13.9.5. 변환 .....	356
13. EB GUIDE Studio의 설치 .....	358
13.1. 배경 정보 .....	358
13.1.1. 제한 .....	358
13.1.2. 시스템 요구사항 .....	358
13.2. EB GUIDE 다운로드 중 .....	359
13.3. EB GUIDE 설치하기 .....	359
13.4. 제거 EB GUIDE .....	360
용어해설 .....	362
인덱스 .....	366

# 1. 이 문서 정보

## 1.1. 타겟 오디언스: 모델러

모델러는 EB GUIDE Studio를 사용하여 HMI(휴먼 기계 인터페이스)를 만듭니다. EB GUIDE에서 HMI의 명칭은 EB GUIDE 모델입니다. 응용 프로그램과의 통신은 이벤트 메커니즘을 사용하는 확인된 이벤트, 데이터풀을 사용하는 데이터풀 항목 및 사용자 관련 EB GUIDE 스크립트 함수를 통해 수행됩니다.

모델러는 다음과 같은 작업을 수행합니다.

- ▶ 위젯 아키텍처와 보기를 사용하여 디스플레이의 그래픽 요소 지정
- ▶ 디자이너 및 유용성 전문가와의 의견 교환을 통해 사용자 인터페이스 최적화
- ▶ 상태 기계 기능을 사용하여 그래픽 요소가 표시되는 시기 지정
- ▶ 제어판, 터치 스크린 등의 기기에서 요소가 입력에 반응하는 방법 정의
- ▶ 내비게이션 장치 등의 서비스를 제공하는 하드웨어 또는 소프트웨어 응용 프로그램에서 요소가 정보를 수신하는 방법 정의
- ▶ 모델 요소와 입력 및 출력 기기 간의 인터페이스 정의

모델러는 다음 항목에 대해 숙지하고 있어야 합니다.

- ▶ EB GUIDE Studio 기능
- ▶ UML 상태 머신 개념
- ▶ 도메인의 사양과 요구사항
- ▶ 교환되는 데이터 및 EB GUIDE GTF 통신 메커니즘
- ▶ 3D 그래픽이 프로젝트에 사용되는 경우 3D 그래픽의 사양

## 1.2. 사용자 문서 구조

정보는 다음과 같은 구조로 구성됩니다.

- ▶ 배경 정보

배경 정보에서는 특정 항목과 중요한 사실을 소개합니다. 이 정보를 참조하여 관련 지침을 수행할 수 있습니다.

- ▶ 사용 지침



지침에서는 특정 작업을 단계별로 수행하는 방법을 안내하고 EB GUIDE 사용 방법을 제시합니다. 지침의 제목은 예를 들어 EB GUIDE Studio 시작하기와 같이 '하기' 형식으로 되어 있습니다.

▶ 자습서

자습서는 사용 지침이 확장된 버전으로, 복잡한 작업을 수행하는 방법을 안내합니다. 자습서의 헤더라인은 자습서:로 시작됩니다. 예를 들어 자습서: 버튼 만들기과 같습니다.

▶ 참조

참조는 상세한 기술 매개변수와 테이블을 제공합니다.

▶ 데모

데모에서는 응용 프로그램 작성 방법 및 상호 작용 순서를 설명합니다. 데모는 EB GUIDE GTF SDK의 요소입니다.

## 1.3. 입력 체계와 스타일 규칙

중요한 정보를 나타내기 위해 이 문서에서 다음의 상형 문자 및 신호 문자가 사용됩니다.

신호 문자 WARNING은 구성 성공에 필수적인 정보를 나타냅니다.

**주의!**



**소스 및 문제의 종류**

소프트웨어는 어떻게 됩니까?

문제의 결과는 무엇입니까?

사용자가 문제를 어떻게 피할 수 있습니까?

신호 문자 NOTE는 주제에 관한 중요한 정보를 나타냅니다.

**[참고]**

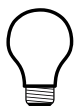


**중요한 정보**

주제에 관한 중요한 정보를 제공합니다.

신호 문자 TIP은 유용한 힌트, 팁 및 바로가기를 제공합니다.

**[작은 정보]**



**유용한 힌트**

유용한 힌트를 제공합니다.

문서 전체에서 굵게 또는 기울임꼴 또는 고정 폭 글꼴로 표시되는 단어와 구를 찾을 수 있습니다.



이러한 규칙의 의미를 알아보려면 다음 예제를 참조하십시오.

모든 기본 텍스트는 Arial Regular 또는 MS Malgun Gothic 글꼴로 작성됩니다.

글꼴	설명	예제
맑은 고딕	파일, 디렉터리, 챕터 이름	스크립트를 function_name/abcdirectory에 입력합니다.
맑은 고딕	사용자 입력, 코드 및 파일 디렉터리	<pre>CC_FILES_TO_BUILD =(PROJECT_PATH)/source/network/can_node.c CC_FILES_TO_BUILD += \$(PROJECT_PATH)/source/network/can_config.c</pre> <p>모듈이 BswM_Dcm_RequestSessionMode() 기능을 호출합니다.</p> <p>프로젝트 이름에는 Project_Test를 입력합니다.</p>
대괄호 []	선택적 매개변수 또는 선택적 매개변수가 있는 커맨드 구문 또는 GUI 요소를 나타냅니다.	<ol style="list-style-type: none"> <li>insertBefore [&lt;opt&gt;]</li> <li><b>Enter</b> 키를 누릅니다.</li> </ol>
중괄호 {}	필수 매개변수 또는 필수 매개변수가 있는 커맨드 구문을 나타냅니다.	insertBefore {<file>}
줄임표(...)	추가 매개변수 또는 여러 매개변수가 있는 커맨드 구문을 나타냅니다.	insertBefore [<opt>...]
수직 막대	사용 가능한 모든 매개변수 또는 사용 가능한 매개변수 중 선택한 하나의 매개변수가 있는 커맨드 구문을 나타냅니다.	allowinvalidmarkup {on off}



단계별 지침입니다.

단계를 따를 때 보이는 표시줄에 단계별 지침 또는 방법이 표시됩니다.

선수학습:

- 이 줄은 지침의 필수 구성 요소를 나타냅니다.

단계 1

작업 완료를 위한 지침입니다.

단계 2

작업 완료를 위한 지침입니다.

단계 3

작업 완료를 위한 지침입니다.

## 1.4. 명명 규칙

EB GUIDE 문서에서 사용되는 디렉터리 이름은 다음과 같습니다.

- ▶ EB GUIDE를 설치하는 디렉터리는 \$GUIDE\_INSTALL\_PATH입니다.

예:

C:/Program Files/Elektrobit/EB GUIDE Studio 6.7

- ▶ EB GUIDE SDK 플랫폼의 디렉터리는 \$GTF\_INSTALL\_PATH입니다. 이름 패턴은 \$GTF\_INSTALL\_PATH/platform/<platform name>입니다.

예:

C:/Program Files/Elektrobit/EB GUIDE Studio 6.7/platform/win32

- ▶ EB GUIDE 프로젝트를 저장하는 디렉터리는 \$GUIDE\_PROJECT\_PATH입니다.

예:

C:/Users/[user name]/Documents/EB GUIDE 6.7/projects/

- ▶ EB GUIDE 프로젝트를 저장하는 디렉터리는 \$EXPORT\_PATH입니다.

## 1.5. 경로 규칙

EB GUIDE Studio는 Windows 10에서 260자 이상의 경로 이름 처리를 지원합니다. 전체 경로 이름에는 260자 이상이 포함될 수 있지만 경로 내의 단일 파일 이름 또는 디렉터리 이름은 248자로 제한됩니다.

### [참고]



### Windows 7의 긴 경로 이름

Windows 7은 긴 경로 이름 처리를 지원하지 않습니다. 긴 경로 이름을 사용하려면 Windows 10에서 EB GUIDE Studio를 실행하십시오. Windows 10에서 긴 경로 이름을 활성화하는 방법에 대한 자세한 내용은 Windows 10 문서를 참조하십시오.

## 2. 안전하고 올바른 사용법

### 2.1. 용도

- ▶ EB GUIDE Studio 및 EB GUIDE GTF는 인포테인먼트 헤드 장치, 클러스터 장비 및 일부 업계 응용 프로그램용 사용자 인터페이스 프로젝트에서 사용됩니다.
- ▶ 주요 사용 사례는 라이선스 범위에 따라 대량 생산, 사양 및 원형 생성 용도입니다.

### 2.2. 오용 가능성

#### 주의!



#### 오용 가능성 및 책임

이 소프트웨어는 적합한 사용법에 따라 해당하는 라이선스 약관 및 계약에서 허용하는 대로만 사용할 수 있습니다. Elektrobit Automotive GmbH에서는 해당하는 라이선스 약관 및 계약을 준수하지 않는 소프트웨어 사용에 대해 책임을 지지 않으며 그에 대해 Elektrobit에 책임을 물을 수 없습니다.

- ▶ ISO 26262/A-SIL에 정의된 안전 관련 시스템에서 휴먼 기계 인터페이스를 구현하기 위한 용도로 Elektrobit Automotive GmbH에서 제공하는 EB GUIDE product line 제품을 사용하지 마십시오.
- ▶ EB GUIDE product line 제품은 DO-178B, SIL, A-SIL 등의 특정 인증이 필요한 안전 관련 시스템에서는 사용할 수 없습니다.

이러한 환경에서는 EB GUIDE GTF를 사용할 수 없습니다. 특정 응용 프로그램의 사용법에 대해 모르는 경우 [장 3. "지원"](#)에서 Elektrobit Automotive GmbH에 문의하여 확인하십시오.

## 3. 지원

EB GUIDE 지원은 다음과 같은 방식으로 제공됩니다.

▶ 커뮤니티 버전:

Elektrobit의 문서, 블로그 및 포럼에서 포괄적인 정보를 확인할 수 있습니다.

▶ 엔터프라이즈 버전:

지원 계약에 따라 Elektrobit에 문의할 수 있습니다.

지원 요청 시에는 설치한 EB GUIDE의 버전 번호를 준비하십시오. 버전 번호를 확인하려면 프로젝트 센터로 이동한 다음 **도움말**을 클릭합니다. 버전 번호는 대화 상자 오른쪽 아래에 있습니다.

## 4. EB GUIDE 소개

EB GUIDE에서는 사용자의 HMI(휴먼 기계 인터페이스) 개발 프로세스를 지원합니다. EB GUIDE 제품 라인은 그래픽 또는 음성 사용자 인터페이스용 도구와 플랫폼을 제공합니다. EB GUIDE 제품 라인은 인포테인먼트 헤드 장치, 클러스터 장비 및 일부 업계 응용 프로그램용 프로젝트에서 사용됩니다. 주요 사용 사례는 대량 생산, 사양 및 원형 생성입니다.

### 4.1. EB GUIDE product line의 소프트웨어 요소

EB GUIDE product line에는 다음과 같은 소프트웨어 요소가 포함되어 있습니다.

- ▶ EB GUIDE Studio
- ▶ EB GUIDE TF

EB GUIDE Studio는 PC의 모델링 도구입니다. EB GUIDE Studio를 통해 사용자가 함수에 액세스하는 데 사용할 수 있는 중앙 제어 요소로 전체 HMI 기능을 모델링합니다.

EB GUIDE TF는 EB GUIDE Studio에서 만든 EB GUIDE 모델을 실행합니다. EB GUIDE TF는 개발 PC 및 여러 포함된 플랫폼에서 사용할 수 있습니다.

EB GUIDE Studio를 사용하여 만든 EB GUIDE 모델 및 EB GUIDE TF에서 실행되는 내보낸 EB GUIDE 모델은 전혀 별개의 모델입니다. 이러한 모델은 상호 작용하기는 하지만 서로 차단할 수는 없습니다.

### 4.2. EB GUIDE Studio

#### 4.2.1. HMI 동작 모델링

상태 기계에서 여러 상태를 결합하고 상태를 배치하여 EB GUIDE 모델의 동적 동작을 지정합니다.

##### 상태 기계

상태 기계는 시스템의 동적 동작을 설명하는 명확한 유한 자동화 기능입니다. 예를 들어 EB GUIDE Studio에서는 햅틱 상태 기계 등 여러 유형의 상태 머신을 사용할 수 있습니다. 햅틱 상태 기계에서는 그래픽 사용자 인터페이스를 지정할 수 있습니다.

##### 상태

상태는 전이를 통해 연결됩니다. 전이는 상태 및 트리거 상태 변경 간의 연결입니다.

## 4.2.2. HMI 모양 모델링

EB GUIDE Studio에서 그래픽 사용자 인터페이스와 EB GUIDE 모델의 음성 사용자 인터페이스를 정의합니다.

### 위젯

EB GUIDE Studio에서는 그래픽 사용자 인터페이스를 만들 수 있도록 위젯을 제공합니다. 모양을 정의하는 모델 요소인 위젯은 예를 들어 주로 텍스트 레이블, 이미지 등의 정보를 표시하는 데 사용됩니다. 또한 사용자는 위젯을 통해 예를 들어 버튼, 슬라이더 등의 시스템 동작을 제어할 수 있습니다. 여러 위젯이 조합되어 보기라는 구조를 형성합니다.

### 스피젯

EB GUIDE Studio에서는 음성 사용자 인터페이스를 만들 수 있도록 스피젯을 제공합니다. 스피젯은 음성 대화 상자의 기본적인 요소를 지정하는 데 사용됩니다. 음성 인식이 사용자 입력으로 사용되고 음성 합성이 시스템 출력으로 사용됩니다. 프롬프트 스피젯을 사용하면 TTS(텍스트 음성 변환) 신시사이저를 통해 재생되는 텍스트를 모델링할 수 있습니다. 명령 스피젯을 사용하면 음성 인식기가 이해하는 내용을 설명하는 문법을 모델링할 수 있습니다. 관련 스피젯은 모델 요소를 통해 함께 그룹화됩니다. 이러한 그룹을 토크라고 합니다.

## 4.2.3. 데이터 처리

HMI와 응용 프로그램 간의 통신은 데이터풀 및 이벤트 시스템을 통해 구현됩니다.

### 데이터풀

데이터풀은 표시해야 하는 모든 데이터 및 추가 내부 정보가 저장되어 있는 포함된 데이터베이스입니다. 데이터풀 항목은 데이터를 저장하고 교환합니다.

### 이벤트 시스템

이벤트는 임시 트리거이며, 특정 상황이 발생했음을 나타내기 위해 HMI와 응용 프로그램 양쪽에 모두 전송할 수 있습니다.

응용 프로그램 소프트웨어는 API를 통해 이벤트 및 데이터풀에 액세스할 수 있습니다.

## 4.2.4. EB GUIDE 모델 시뮬레이션

EB GUIDE Studio를 사용하여 시뮬레이션 중에 EB GUIDE 모델의 기능을 테스트할 수 있습니다. 마우스를 클릭하여 시뮬레이션을 시작하고 EB GUIDE 모델의 모양과 느낌을 바로 체험할 수 있습니다.

마우스, 키보드 또는 터치스크린 등의 입력 장치를 사용하여 시뮬레이션과 상호 작용할 수 있습니다.

EB GUIDE Monitor를 통해 EB GUIDE 모델을 제어하고 다음을 수행할 수도 있습니다.

- ▶ 데이터풀 항목의 값을 변경하여 표시된 데이터 변경

- ▶ 이벤트를 실행하여 사용자 입력 시뮬레이션
- ▶ 로그에서 모든 변경 사항 추적
- ▶ 스크립트 시작

EB GUIDE Monitor를 독립 실행형 응용 프로그램으로 사용할 수도 있습니다.

## 4.2.5. EB GUIDE 모델 내보내기

타깃 기기에서 EB GUIDE 모델을 사용하려면 EB GUIDE Studio에서 EB GUIDE 모델을 내보낸 다음 타깃 기기가 인식할 수 있는 형식으로 변환해야 합니다. 내보내기 중에는 모든 관련 데이터가 ASCII 파일 집합으로 내보내집니다.

## 4.3. EB GUIDE TF

EB GUIDE TF는 EB GUIDE 모델을 실행하는 데 필요한 라이브러리 집합 및 GtfStartup 실행 파일로 구성됩니다.

EB GUIDE Studio에서 선택한 프로젝트 유형에 따라 다음을 실행합니다.

- ▶ EB GUIDE GTF

EB GUIDE Graphics Target Framework는 그래픽 HMI를 실행하는 런타임 환경입니다.

- ▶ EB GUIDE STF

EB GUIDE Speech Target Framework는 HMI에서 음성 기능을 실행하는 런타임 환경입니다.

EB GUIDE TF의 대다수 프로그램 코드는 플랫폼에 따라 달라지지 않습니다. 따라서 새 시스템으로 매우 쉽게 코드를 이식할 수 있습니다.

EB GUIDE 모델 파일만 교환하면 전체 HMI를 교환할 수 있습니다. 따라서 EB GUIDE TF를 다시 컴파일할 필요가 없습니다. 변경된 EB GUIDE 모델을 EB GUIDE Studio에서 다시 내보내기만 하면 됩니다.

EB GUIDE TF에서는 다음과 같은 플랫폼 추상화를 사용합니다.

- ▶ OS 추상화

OS(운영 체제)의 플랫폼 종속성이 GtfOSAL(운영 체제 추상화 계층)에 캡슐화됩니다. 예를 들어 EB GUIDE TF에서는 파일 시스템 또는 TCP 소켓과 같은 운영 체제의 기능을 사용합니다.

- ▶ GL 추상화

그래픽 하위 시스템의 플랫폼 종속성은 렌더러에 의해 캡슐화됩니다. EB GUIDE 모델은 기하 도형, 조명 등의 요소 속성을 포함합니다. 내보낸 EB GUIDE 모델에 포함되어 있는 데이터는 처리 후 디지털 이미지



로 출력할 수 있도록 렌더러로 전달됩니다. 렌더러는 하드웨어의 실제 그래픽 시스템이 추상화된 것입니다. EB GUIDE TF는 각 플랫폼에 대해 다양한 렌더러를 지원합니다.

▶ 오디오 추상화

음성 사용자 인터페이스는 오디오 하드웨어에 액세스할 수 있어야 합니다. 오디오 추상화에서는 마이크와 스피커 액세스 기능을 제공합니다. EB GUIDE STF는 음성 인식 및 텍스트 음성 합성을 구현합니다. 이 구현을 위해 EB GUIDE STF에는 서드파티 음성 엔진이 통합되어 있습니다.

## 5. 자습서: 시작하기

### [참고]



### 기본 창 레이아웃

이 사용자 설명서의 모든 지침 및 스크린샷이 기본 창 레이아웃을 사용합니다. 지침을 따르는 경우 EB GUIDE Studio 창을 기본 레이아웃으로 설정하는 것이 좋습니다. **레이아웃 > 기본 레이아웃으로 다시 설정**을 선택하여 설정할 수 있습니다.

다음 섹션에서는 EB GUIDE Studio를 사용한 HMI 모델링에 대한 간략한 배경 정보를 설명합니다. 여기서는 EB GUIDE Studio를 시작하고, 프로젝트를 만들고, EB GUIDE 모델의 동작과 모양을 모델링하고, EB GUIDE 모델을 시뮬레이션하는 방법을 설명합니다.

대략적인 소요 시간: 20분

### 5.1. EB GUIDE 시작하기



#### EB GUIDE 시작하기

선수학습:

- EB GUIDE가 설치되어 있어야 합니다.

#### 단계 1

Windows **시작** 메뉴에서 **모든 프로그램**을 클릭합니다.

#### 단계 2

**Elektrobit** 메뉴에서 시작할 버전을 클릭합니다.

EB GUIDE Studio가 시작됩니다. 프로젝트 센터가 표시됩니다.

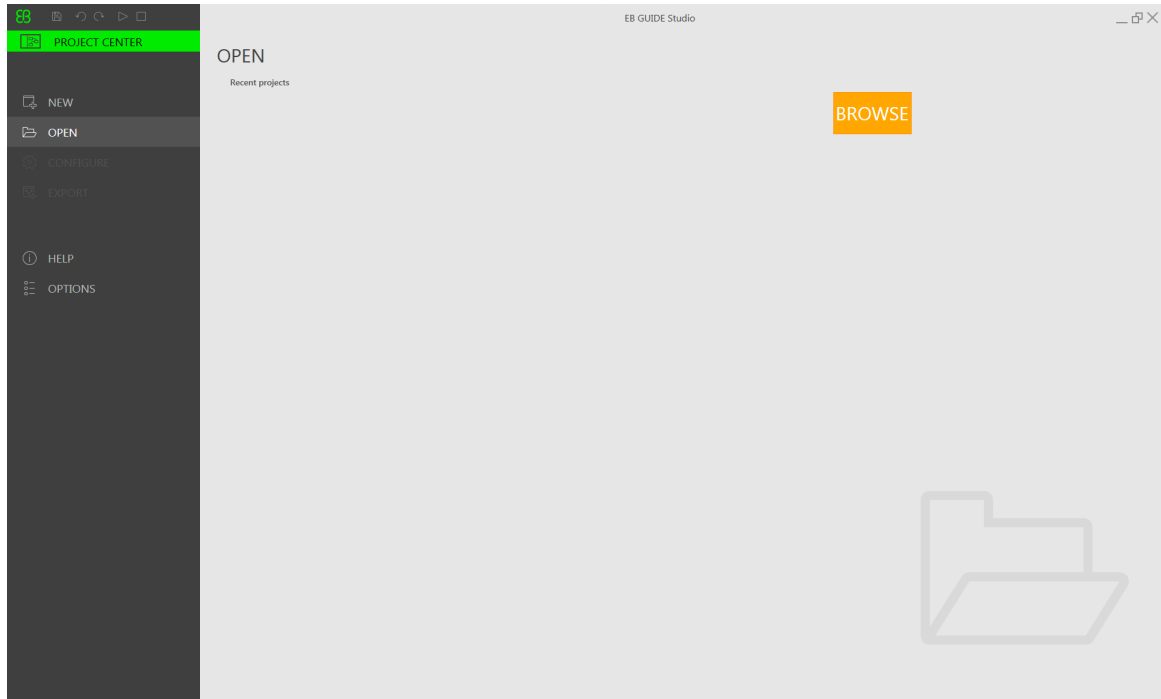


그림 5.1. 프로젝트 센터

## 5.2. 프로젝트 만들기



### 프로젝트 만들기

선수학습:

- EB GUIDE Studio를 시작해야 합니다.
- C:/temp 디렉터리를 만들어야 합니다.

#### 단계 1

프로젝트 센터의 내비게이션 영역에서 **새로 만들기**를 클릭합니다.

#### 단계 2

콘텐츠 영역에서 C:/temp 디렉터리를 **위치**로 선택합니다.

#### 단계 3

프로젝트 이름으로 MyProject를 입력합니다.

#### 단계 4

**만들기**를 클릭합니다.

프로젝트가 만들어집니다. 프로젝트 편집자가 열리고 빈 프로젝트가 표시됩니다.

기본 상태 기계가 기본적으로 추가되어 콘텐츠 영역에 표시됩니다.

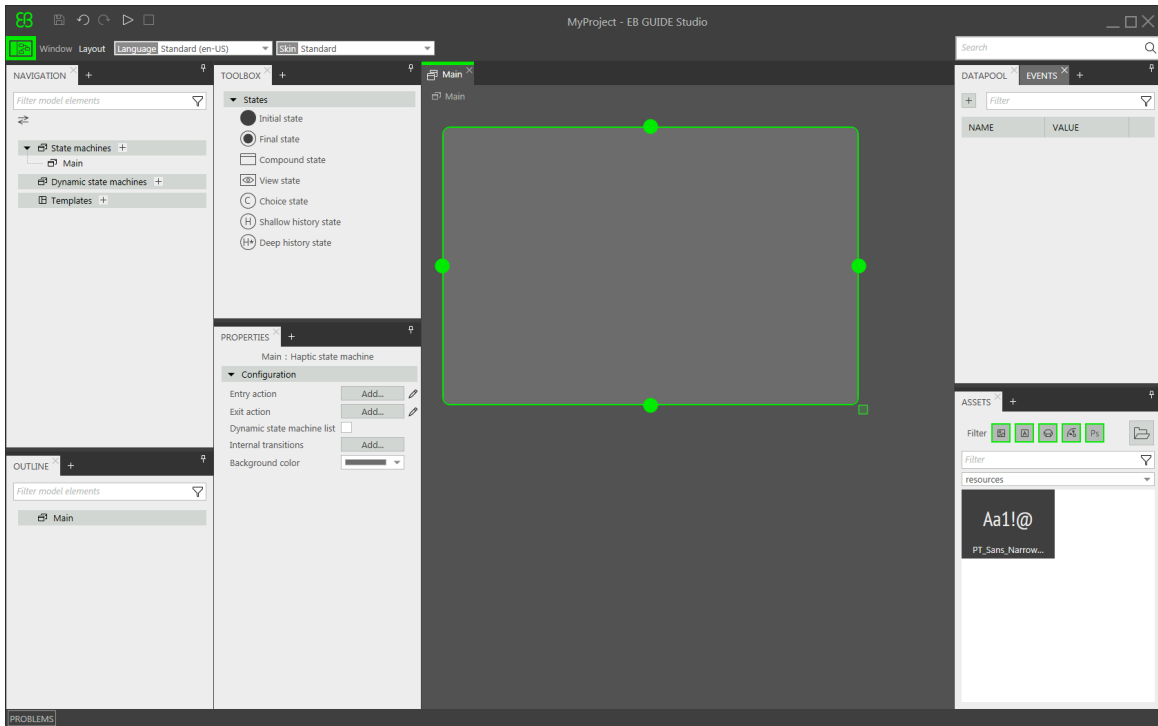



그림 5.2. 기본 상태 기계가 포함된 프로젝트 편집자

### 5.3. HMI 동작 모델링

EB GUIDE 모델의 동작은 상태 기계에 의해 정의됩니다. EB GUIDE에서는 UML과 비슷한 구문을 사용하여 모델의 동작을 정의합니다.

다음 섹션에서는 시작 시에는 정의된 보기를 표시하고 버튼을 누르면 다른 보기로 변경되는 상태 머신을 모델링하는 방법에 대해 알아봅니다.

 상태 기계에 상태 추가하기

EB GUIDE에서는 여러 가지 상태를 제공합니다. 다음 섹션에서는 3가지 상태에 대해 살펴봅니다. 초기 상태는 상태 기계의 시작 포인트를 정의합니다. 보기 상태는 기본적으로 보기를 표시합니다. 그리고 상태 기계의 최종 상태는 상태 기계를 종료합니다.

선수학습:

- MyProject 프로젝트가 만들어집니다.
- 콘텐츠 영역에 **기본** 상태 머신이 표시되어 있어야 합니다.

단계 1

보기 상태를 도구 상자에서 상태 기계로 끕니다.

View state 1과 함께 보기 하나가 EB GUIDE 모델에 추가됩니다.

단계 2

1단계를 반복합니다.

View state 2 이(가) 추가됩니다.

단계 3

초기 상태를 도구 상자에서 상태 기계로 끕니다.

단계 4

최종 상태를 도구 상자에서 상태 기계로 끕니다.

기본 상태 기계에 추가한 4개 상태는 콘텐츠 영역에 상태 차트로 표시되며 **내비게이션** 구성 요소에 계층 트리 보기로도 표시됩니다.

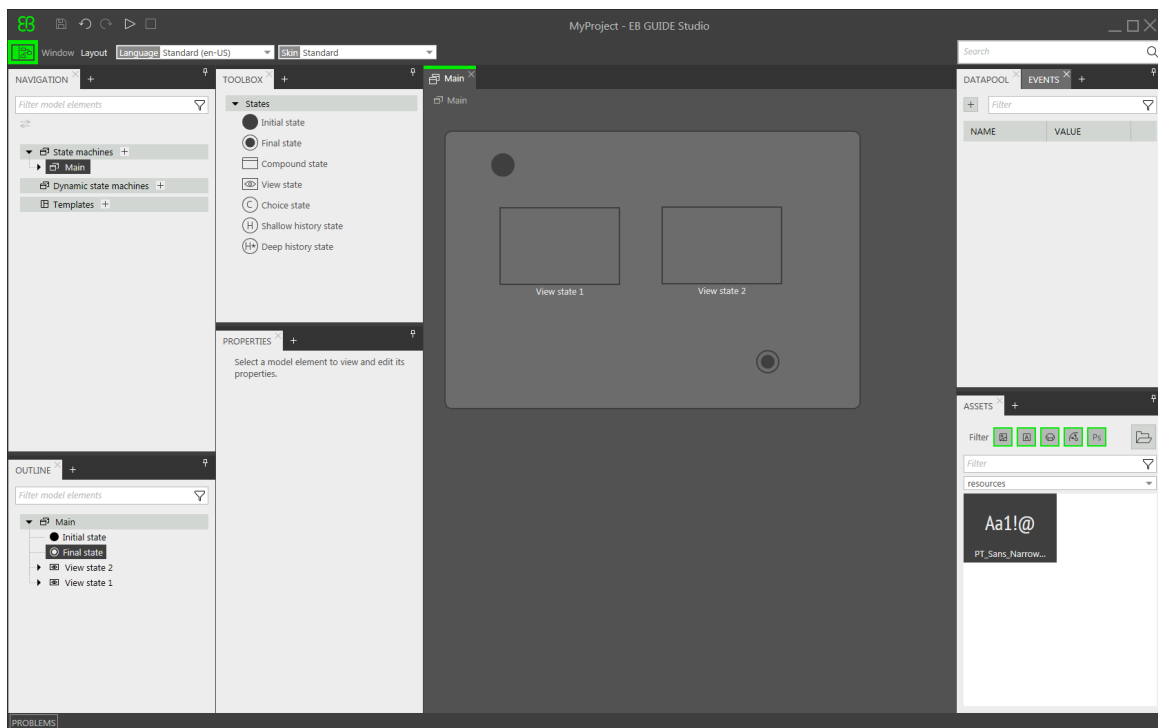


그림 5.3. 상태가 포함된 프로젝트 편집자



전이 추가하기

전이는 상태 간의 연결이며 상태 변경을 트리거합니다. 전이에는 서로 다른 여러 유형이 있습니다. 다음 섹션에서는 기본 전이와 이벤트 트리거 전이에 대해 살펴봅니다.

선수학습:

- 콘텐츠 영역에 **기본** 상태 머신이 표시되어 있어야 합니다.
- **기본** 상태 머신에 초기 상태, 보기 상태 2개 및 최종 상태가 포함되어 있어야 합니다.

단계 1

전이의 원본 상태로 초기 상태를 선택합니다.

단계 2

녹색 끌기 포인트를 클릭한 다음 마우스 버튼을 누른 상태로 유지합니다.

단계 3

마우스를 타깃 상태, View state 1로 끕니다.

단계 4

타깃 상태가 녹색으로 강조 표시되면 마우스 버튼을 놓습니다.

전이가 만들어져 녹색 화살표로 표시됩니다.

단계 5

View state 1과 View state 2 사이에 전이를 추가합니다.

View state 1을 선택하고 타깃 상태로서의 View state 2에 대해 2 ~ 4단계를 반복합니다.

단계 6

View state 1 및 View state 2 사이의 전이를 선택합니다.

다음 단계에서는 전이를 이벤트에 연결합니다.

단계 7

**속성** 구성 요소로 이동한 다음 **트리거** 콤보 박스에 Event 1을 입력하고 **이벤트 추가**를 클릭합니다.

Event 1이라는 이벤트가 만들어져 전이 트리거로 추가됩니다. Event 1이 발생할 때마다 전이가 실행됩니다.

단계 8

View state 2와 최종 상태 간에 전이를 추가합니다.

View state 2을 선택하고 타깃 상태로서의 최종 상태에 대해 2 ~ 4단계를 반복합니다.

새 이벤트인 Event 2를 트리거로 추가합니다.

이때 상태 기계는 다음 그림과 같습니다.

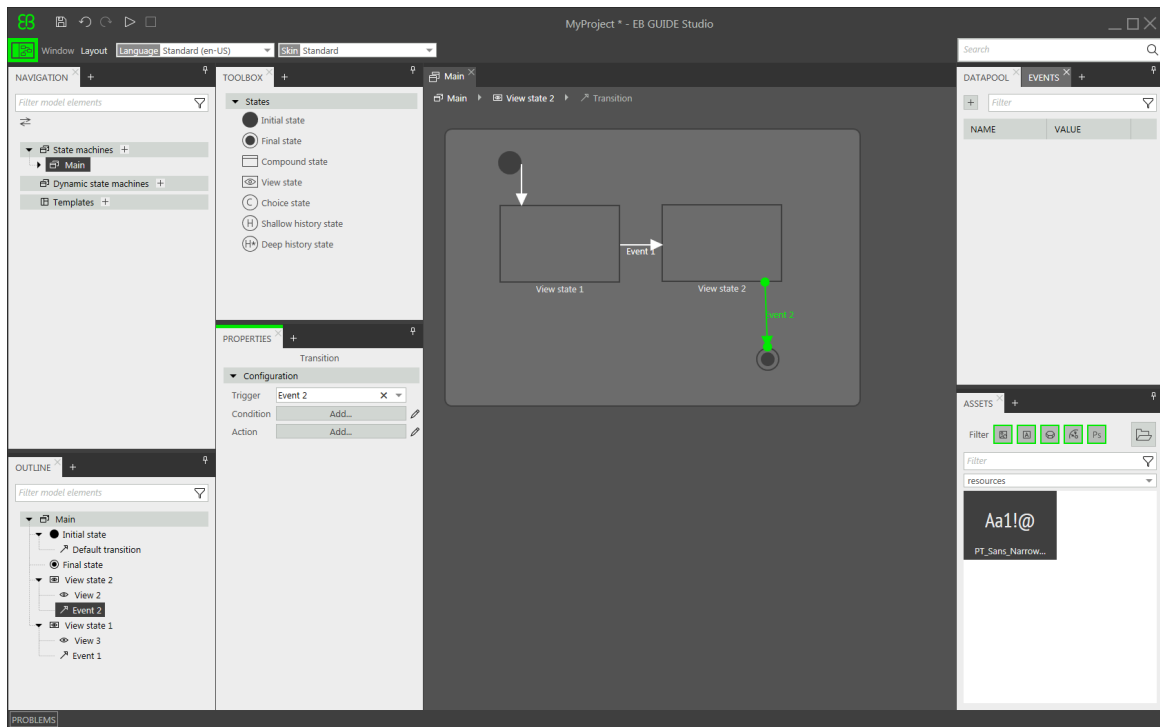


그림 5.4. 전이를 통해 이벤트와 연결된 상태

기본 상태 기계의 동작을 정의했습니다.

## 5.4. HMI 모양 모델링

위의 섹션에서 만든 상태 기계에는 보기 상태 2개가 포함되어 있습니다. 다음 섹션에서는 보기를 모델링하는 방법에 대해 알아봅니다.



### 보기 열기

선수학습:

- View state 1 이(가) 모델에 추가됩니다.

#### 단계 1

View state 1을 더블 클릭합니다.

콘텐츠 영역에 View 1이 표시됩니다.



## 보기에 버튼 추가

EB GUIDE Studio에서는 여러 옵션을 사용하여 보기의 모양을 모델링할 수 있습니다.

이러한 옵션의 한 가지 예로, 다음 섹션에서는 보기에 사각형을 추가하는 방법을 설명합니다. 사각형은 사용자 입력에 반응하므로 버튼 기능을 합니다.

선수학습:

- 콘텐츠 영역에 View 1이 표시되어 있어야 합니다.

### 단계 1

사각형을 **도구 상자**에서 보기로 끕니다.

### 단계 2

**속성** 구성 요소에서 **위젯 피쳐 속성** 범주로 이동한 다음 **추가/제거**를 클릭합니다.

**위젯 피쳐** 대화 상자가 표시됩니다.

### 단계 3

**사용 가능한 위젯 피쳐** 아래에서 **입력 처리** 범주를 확장하고 **터치 놓음**을 선택합니다.

**수락**을 클릭합니다.

관련 위젯 피쳐 속성이 **속성** 구성 요소에 추가됩니다.

### 단계 4

**속성** 구성 요소의 touchPolicy 드롭다운 목록 상자에서 Press then react를 선택합니다.

사각형이 시뮬레이션 모드에서의 터치 입력에 반응합니다.

### 단계 5

touchShortReleased 속성으로 이동하여 **편집**을 클릭합니다.

### 단계 6

다음 EB GUIDE 스크립트를 입력합니다.

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

사각형을 시뮬레이션 모드에서 터치하면 500밀리초 후에 Event 1이 발생합니다.

### 단계 7

**수락**을 클릭합니다.

### 단계 8

**속성** 구성 요소에서 fillColor 속성에 대해 빨강을 선택합니다.



단계 9

**내비게이션** 구성 요소에서 View 2를 더블 클릭합니다.

콘텐츠 영역에 View 2가 표시되어 있어야 합니다.

단계 10

1~5단계를 반복합니다.

단계 11

다음 EB GUIDE 스크립트를 입력합니다.

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

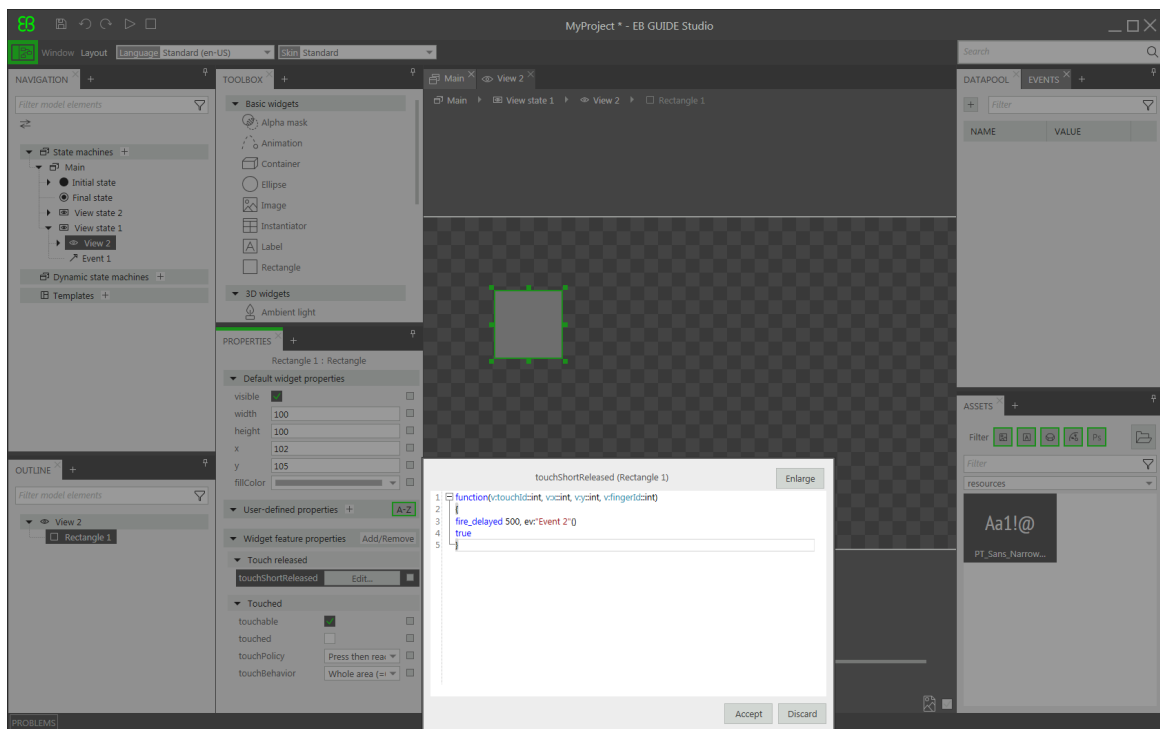


그림 5.5. EB GUIDE 스크립트가 포함된 위젯 속성

단계 12

**수락**을 클릭합니다.

사각형을 시뮬레이션 모드에서 터치하면 500밀리초 후에 Event 2가 발생합니다.

단계 13

**속성** 구성 요소에서 fillColor 속성에 대해 파란을 선택합니다.


## 5.5. 시뮬레이션 시작

EB GUIDE에서는 모델을 타킷 기기로 내보내기 전에 PC에서 모델 시뮬레이션을 진행할 수 있습니다.



### 시뮬레이션 시작

#### 단계 1


프로젝트를 저장하려면 명령 영역에서  을 클릭합니다.

#### 단계 2

명령 영역에서  을 클릭합니다.

EB GUIDE 모델이 시작되고 모델링한 동작과 모양이 표시됩니다.

우선 View 1이 표시됩니다. 빨간색 사각형을 클릭하면 화면이 View 2로 변경됩니다. 이는 클릭으로 인해 Event 1이 발생하고 Event 1이 View state 1에서 View state 2로의 전이를 실행하기 때문입니다.

그런 다음 View 2가 표시됩니다. View 2의 파란색 사각형을 클릭하면 상태 기계가 종료됩니다. 클릭으로 인해 Event 2가 발생하며, Event 2는 View state 2에서 최종 상태로의 전이를 실행하기 때문입니다. 시뮬레이션 창은 계속 열려 있습니다. 시뮬레이션을 중지하려면  를 클릭합니다.

## 6. 배경 정보

이 장의 항목은 사전순으로 정렬되어 있습니다.

### [참고]



#### 기본 창 레이아웃

이 사용자 설명서의 모든 지침 및 스크린샷이 기본 창 레이아웃을 사용합니다. 지침을 따르는 경우 EB GUIDE Studio 창을 기본 레이아웃으로 설정하는 것이 좋습니다. **레이아웃 > 기본 레이아웃으로 다시 설정**을 선택하여 설정할 수 있습니다.

### 6.1. 3D 그래픽

EB GUIDE Studio에서는 EB GUIDE 프로젝트에 3D 그래픽을 사용할 수 있습니다.

#### 6.1.1. 지원되는 3D 그래픽 형식

OpenGL ES 버전 2.0 이상 및 DirectX 11 렌더러만 3D 그래픽을 표시할 수 있습니다. 지원되는 3D 그래픽 형식은 COLLADA(.dae) 및 Filmbox(.fbx)입니다. 최적의 결과를 얻으려면 Filmbox 형식을 사용하십시오.

#### 6.1.2. 3D 그래픽 파일의 설정

EB GUIDE Studio에서 보기에 3D 객체를 표시하려면 다음 옵션을 사용하여 3D 그래픽 파일을 만들어야 합니다.

- ▶ 퍼스펙티브 카메라
- ▶ 메시와 하나 이상의 재료를 포함하는 하나 이상의 개체
- ▶ 하나 이상의 광원

3D 그래픽 파일을 만들려면 서드파티 3D 모델링 소프트웨어를 사용하십시오.

3D 그래픽 파일은 아래에 나와 있는 다양한 추가 콘텐츠를 지원합니다.

- ▶ 위치, 법선, 중법선, 접선 및 질감 채널 1개가 포함된 3D 객체
- ▶ 방향 광원
- ▶ 이미지 기반 광원
- ▶ 주변 광원

- ▶ 상수, 선형, 정방향 및 입방형 감쇠가 적용되는 포인트 광원
- ▶ 원뿔 각도, 상수, 선형, 정방향 및 입방형 감쇠가 적용되는 스포트 광원
- ▶ 퍼스펙티브 카메라의 시야, 근거리 평면 및 원거리 평면 지원
- ▶ 질감: 방사, 확산, 거울, 일반 지도, 불투명도, 반사 큐브 및 조명 지도

**[작은 정보] 3D 그래픽 파일 설정**



불투명 지도의 경우 유효한 알파 채널이 필요합니다.

### 6.1.3. 3D 그래픽 파일 가져오기

3D 그래픽을 보기에 추가하려면 장면 그래프를 사용하여 3D 그래픽 파일을 가져와야 합니다. 가져오는 도중 EB GUIDE Studio가 3D 그래픽 파일을 장면 그래프가 있는 부모 노드로서 위젯 트리로 변환합니다. 카메라, 재료, 메시, EB GUIDE Studio 등과 같은 3D 그래픽 파일의 콘텐츠인 경우 각각의 위젯을 생성합니다. 가져온 3D 그래픽 파일의 3D 장면애 애니메이션이 있으면 EB GUIDE Studio에서는 선형 키값 보간 정수 커브 및 선형 키값 보간 부동 커브를 사용하여 이러한 애니메이션을 가져옵니다.

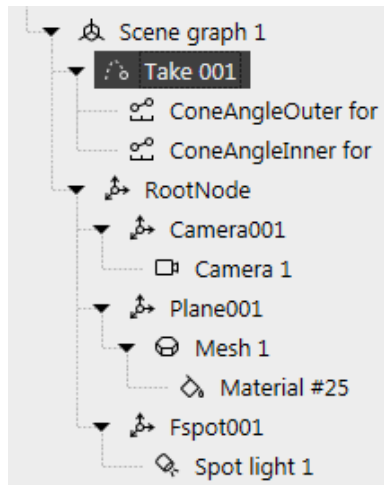


그림 6.1. 내비게이션 구성 요소에 표시된 장면 그래프 예제

**[참고]****제한**

EB GUIDE Studio에서 메시당 하나의 재료만 허용됩니다. 3D 그래픽에 메시당 둘 이상의 재료가 있는 경우 EB GUIDE Studio를 가져오는 중에 각 추가 재료에 대한 추가 메시가 생성됩니다.

.fbx 파일을 가져오는 중에는 기본 재료 위젯만 생성됩니다. 3D 모델에 다른 유형의 재료가 있는 경우 EB GUIDE Studio가 기본 재료만을 추가하며 그 속성은 기본 값으로 설정됩니다. EB GUIDE Studio에서 PBR Phong 재료 및 PBR GGX 재료 위젯을 이용해 다른 유형의 재료를 추가할 수 있습니다.

3D 그래픽 파일을 가져오고 나면 \$GUIDE\_PROJECT\_PATH/<project name>/resources 디렉터리에 하위 디렉터리가 생성됩니다. 하위 디렉터리의 이름은 .fbx 파일을 가져온 후에 지정됩니다. 또한 만든 날짜와 시간이 하위 디렉터리 이름에 추가됩니다.

**예 6.1.****가져오기 디렉터리 이름**

3D 그래픽 파일의 이름은 car.fbx입니다. EB GUIDE Studio의 3D 그래픽 파일을 가져오고 나면 \$GUIDE\_PROJECT\_PATH/<project name>/resources에서 이름이 car\_20160102\_103029인 하위 디렉터를 찾을 수 있습니다.

하위 디렉터리에는 다음이 포함되어 있습니다.

- ▶ .ebmesh 파일로 메시
- ▶ .png 또는 .jpg 파일로 질감

3D 그래픽에 대해 추가 질감을 사용하려면 질감을 \$GUIDE\_PROJECT\_PATH/<project name>/resources에 복사합니다. 질감은 .png 또는 .jpg 이미지를 사용합니다.

하나의 장면 그래프 내의 여러 3D 그래픽을 가져올 수 있습니다.

가져오고 나면 3D 위젯을 추가, 수정 또는 삭제할 수 있습니다.

자세한 내용은 [섹션 6.21, "위젯"](#), [섹션 12.12.4, "3D 위젯"](#) 및 [섹션 12.13.8, "3D"](#)를 참조하십시오.

관련 지침은 [섹션 8.1.3.1, "보기에 장면 그래프 추가"](#) 및 [섹션 11.7, "자습서: 3D 그래픽으로 작업"](#)를 참조하십시오.

## 6.2. 애니메이션

애니메이션은 사용자 EB GUIDE 모델에 모션과 시각 효과를 가져다줍니다. EB GUIDE에서 서로 다른 사용 사례에 애니메이션을 사용할 수 있습니다. 보기 내에서의 위젯 애니메이션과 하나의 보기에서 다른 보기로의 전이 애니메이션을 생성할 수 있습니다.

## 6.2.1. 위젯 애니메이션

위젯 애니메이션을 생성하는 것은 보기에 따라서 위젯을 이동시키는 것을 의미합니다. 이동은 커브를 통해 정의됩니다. 그러므로 **도구 상자의 기본 위젯**은 애니메이션이라 불리는 위젯을 포함합니다. 모든 애니메이션에 상수 커브, 선형 보간 커브 또는 공동 커브 등의 커브 집합을 추가할 수 있습니다. 커브는 target 위젯 속성을 포함하며 target 속성의 시간 기반 변경을 설명합니다.

각 애니메이션에는 커브가 하나 이상 연결되어 있습니다.

위젯 애니메이션은 다음을 비롯한 여러 작업을 수행할 수 있습니다.

- ▶ 보기 내에서 위젯 이동
- ▶ 위젯의 크기 변경
- ▶ 점진적으로 위젯의 색 변경

EB GUIDE 스크립트 함수 f:animation\_play, f:animation\_pause, f:animation\_cancel 등을 통해 애니메이션을 제어합니다.

### [작은 정보] 동시 애니메이션



EB GUIDE의 애니메이션은 동시 애니메이션이며 커브는 병렬로 실행됩니다. 즉, 여러 애니메이션의 커브가 같은 위젯 속성을 타겟으로 사용하는 경우 커브는 해당 target 속성의 값을 동시에 덮어씁니다.

애니메이션 및 커브 속성은 [섹션 12.12.2.2. "애니메이션"](#) 항목을 참조하십시오.

관련 지침은 [섹션 8.1.2.7. "애니메이션 추가"](#) 항목을 참조하십시오.

## 6.2.2. 보기 전이 애니메이션

관점 전환 애니메이션을 생성한다는 것은 보기 상태 진입 또는 종료를 위해 애니메이션이 이동하거나 희미해지도록 정의하는 것을 의미합니다. 보기가 변하면 그러한 애니메이션이 생성됩니다.

보기 상태 및 보기 템플릿을 위한 관점 전환 애니메이션을 정의하세요. 보기 템플릿을 재사용할 때마다 인스턴스가 템플릿의 관점 전환 애니메이션을 상속합니다.

다양한 관점 전환 애니메이션 유형이 있습니다.

표 6.1. 애니메이션 유형

애니메이션 유형	설명
진입 애니메이션	애니메이션이 있는 보기 상태에 들어가면 애니메이션이 재생됩니다. 애니메이션은 위젯 속성과 추가된 보기의 위젯 피처 속성만을 조작할 수 있습니다.

애니메이션 유형	설명
종료 애니메이션	애니메이션이 있는 보기 상태가 종료되면 애니메이션이 재생됩니다. 애니메이션은 위젯 속성과 추가된 보기의 위젯 피쳐 속성만을 조작할 수 있습니다.
애니메이션 변경	애니메이션은 보기 상태 변경 시 재생됩니다. 애니메이션은 소스 속성과 목적지 보기를 조작할 수 있습니다. 소스 보기는 애니메이션이 추가되는 보기 상태입니다. 목적지 보기는 다른 보기 상태 또는 보기 템플릿이 될 수 있습니다.
애니메이션 팝업 켜기	보기 템플릿 및 동적 상태 기계 보기에 대해서만 사용할 수 있습니다. 각각의 동적 상태 기계가 활성화(동적 상태 기계를 누름)된 경우에만 애니메이션이 재생됩니다. 애니메이션은 추가된 보기의 속성과 위젯 속성을 조작할 수 있습니다.
애니메이션 팝업 끄기	보기 템플릿 및 동적 상태 기계 보기에 대해서만 사용할 수 있습니다. 각각의 동적 상태 기계가 종료(동적 상태 기계가 팝업됨)된 경우에만 애니메이션이 재생됩니다. 애니메이션은 추가된 보기의 속성과 위젯 속성을 조작할 수 있습니다.

보기 템플릿에서의 애니메이션 속성은 [섹션 12.12.1, "보기"](#) 항목을 참조하십시오.

관련 지침은 [섹션 8.6, "관점 전환 애니메이션 생성"](#) 항목을 참조하십시오.

## 6.3. 응용 프로그램과 모델 간의 응용 프로그래밍 인터페이스

EB GUIDE에서는 API(응용 프로그래밍 인터페이스)에서 EB GUIDE TF 및 응용 프로그램 간의 모든 통신 데이터를 추상화합니다. 예를 들어 응용 프로그램의 미디어 플레이어, 내비게이션 등이 있습니다.

API는 데이터풀 항목 및 이벤트를 통해 정의됩니다. 이벤트는 HMI와 응용 프로그램 간에 전송됩니다.



### 예 6.2. API의 콘텐츠

- ▶ 응용 프로그램으로 전송되며 재생해야 하는 트랙 번호에 해당하는 track 매개변수를 포함하는 이벤트 START\_TRACK
- ▶ 재생한 트랙이 종료되면 응용 프로그램에서 HMI로 전송되는 이벤트 TRACK\_STOPPED
- ▶ 응용 프로그램이 기록하는 동적 데이터풀 항목 MEDIA\_CURRENT\_TRACK
- ▶ 재생 속도를 정의하며 HMI에서 사용자가 설정하는 동적 데이터풀 항목 MEDIA\_PLAY\_SPEED

## 6.4. 통신 컨텍스트

통신 컨텍스트는 통신이 수행되는 환경을 설명합니다. 통신 컨텍스트의 예로는 HMI 모델과 통신하는 미디어 또는 내비게이션 응용 프로그램이 있습니다. 특정 통신 컨텍스트의 변경 내용은 기록기 응용 프로그램에 의해 게시되어 독자 응용 프로그램에 의해 업데이트될 때까지는 다른 통신 컨텍스트에 표시되지 않습니다.

프로젝트 구성의 고유한 이름 및 숫자 ID(0~255)로 통신 컨텍스트를 식별합니다.

관련 지침은 [섹션 9.8. "외부 통신 설정"](#) 항목을 참조하십시오.

## 6.5. 그래픽 사용자 인터페이스의 구성 요소

EB GUIDE Studio의 그래픽 사용자 인터페이스는 두 가지 구성 요소, 즉 프로젝트 센터와 프로젝트 편집자로 구분됩니다. 프로젝트 센터에서는 EB GUIDE 프로젝트를 관리하고, 옵션을 구성하고, 타겟 기기로 복사하기 위해 EB GUIDE 모델을 내보냅니다. 프로젝트 편집자에서는 HMI 모양 및 동작을 모델링합니다.

### 6.5.1. 프로젝트 센터

EB GUIDE Studio를 시작하고 나면 처음으로 표시되는 화면이 프로젝트 센터입니다. 모든 프로젝트 관련 기능은 프로젝트 센터에 포함되어 있습니다. 프로젝트 센터는 내비게이션 영역과 콘텐츠 영역의 두 요소로 구성됩니다.



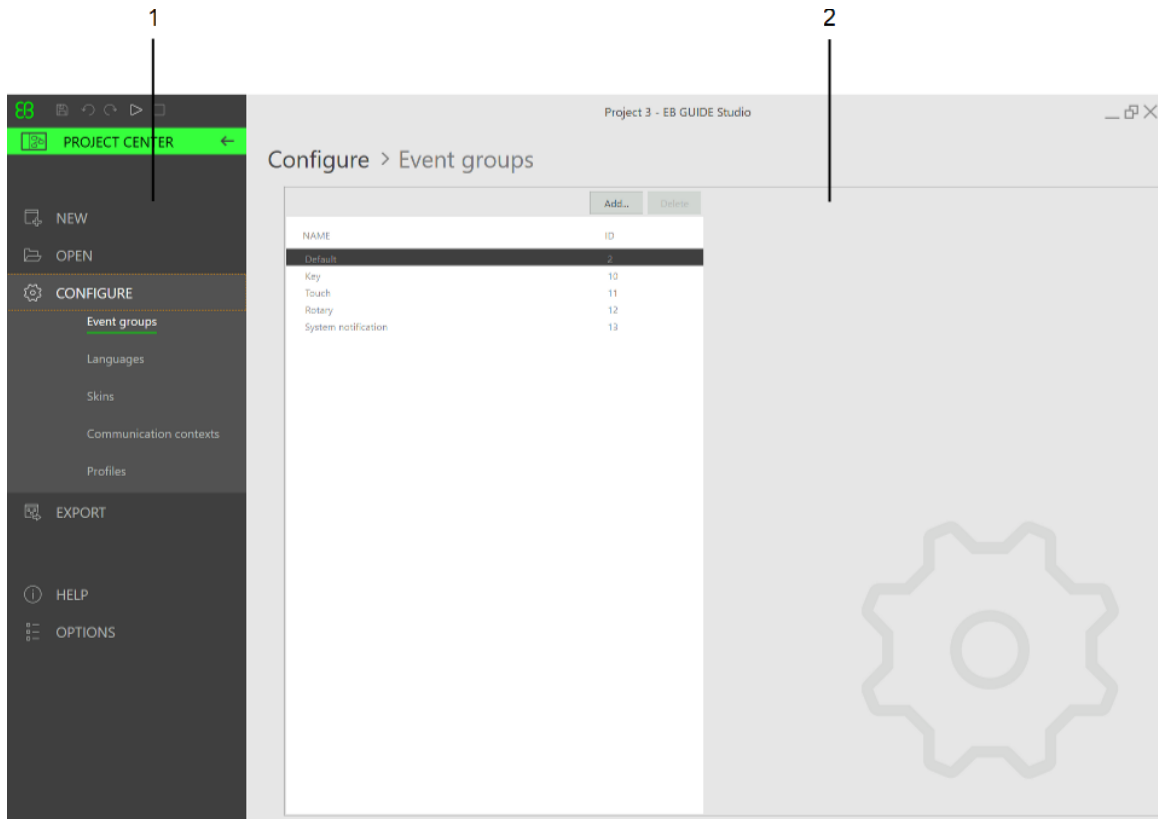


그림 6.2. 내비게이션 영역(1) 및 콘텐츠 영역(2)이 있는 프로젝트 센터

### 6.5.1.1. 내비게이션 영역

프로젝트 센터의 내비게이션 영역은 구성, 내보내기 등의 기능 탭으로 구성됩니다. 내비게이션 영역에서 탭을 클릭하면 콘텐츠 영역에 해당하는 기능과 설정이 표시됩니다.

### 6.5.1.2. 콘텐츠 영역

프로젝트 센터의 콘텐츠 영역에서는 프로젝트 관리 및 구성을 수행합니다. 예를 들어 프로젝트를 저장할 디렉토리를 선택하거나 EB GUIDE 모델의 시작 동작을 정의합니다. 콘텐츠 영역의 모양은 내비게이션 영역에서 선택한 탭에 따라 달라집니다.

## 6.5.2. 프로젝트 편집자

프로젝트를 생성하고 나면 프로젝트 편집자가 표시됩니다. 프로젝트 편집자에서는 HMI의 모양과 동작을 모델링합니다. 즉, 상태 머신을 모델링하고 보기를 생성하고 이벤트와 데이터풀을 관리합니다. 프로젝트 편집자는

다음 영역과 구성 요소로 구성됩니다. 프로젝트 편집자의 모든 구성 요소는 도킹되거나 부동으로 배치될 수 있으며 콘텐츠 영역을 제외한 프로젝트 편집자의 모든 위치에 배치될 수 있습니다.

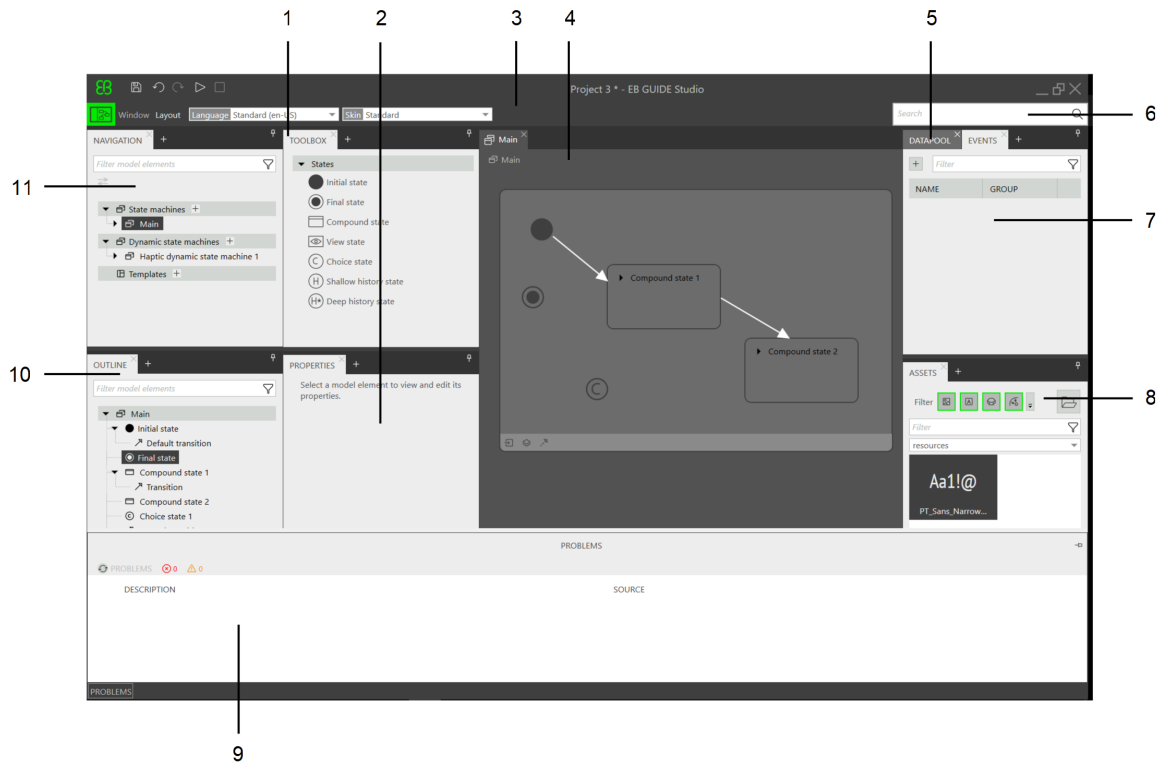


그림 6.3. 영역 및 구성 요소가 포함된 프로젝트 편집자

1. 도구 상자 구성 요소
2. 속성 구성 요소
3. 명령 영역
4. 콘텐츠 영역
5. 데이터풀 구성 요소
6. 검색 상자
7. 이벤트 구성 요소
8. 자산 구성 요소
9. 문제 영역
10. 개요 구성 요소
11. 내비게이션 구성 요소

### 6.5.2.1. 내비게이션 구성 요소

**내비게이션** 구성 요소에서는 상태, 보기, 애니메이션 및 EB GUIDE 모델의 전환과 같은 모델 요소가 계층적 구조로 표시되며 모든 요소로 이동할 수 있습니다. 모델 요소를 두 번 클릭하면 콘텐츠 영역에 해당 모델 요소가 표시됩니다.

상태 머신 계층이 반영되는 **내비게이션** 구성 요소 탭에는 EB GUIDE 모델의 모든 그래픽 요소와 그래픽 이외의 요소가 대략적으로 표시됩니다.

상태 머신, 동적 상태 머신 및 템플릿과 같은 EB GUIDE 모델에 요소를 추가할 수도 있습니다. **도구 상자**에서 끌어 놓기 작업을 이용하여 위젯 및 애니메이션 등의 요소를 추가할 수 있습니다.

**[참고]**



**필터 상자**

구성 요소 상단에는 구성 요소 내의 모든 요소를 검색할 수 있는 필터 상자가 있습니다.

구성 요소의 요소를 클릭하고 F3 키를 누르면 다음과 같은 참조 검색이 시작됩니다. 검색 결과 창이 열리고 EB GUIDE 모델에서 선택된 요소의 모든 항목이 나열됩니다.

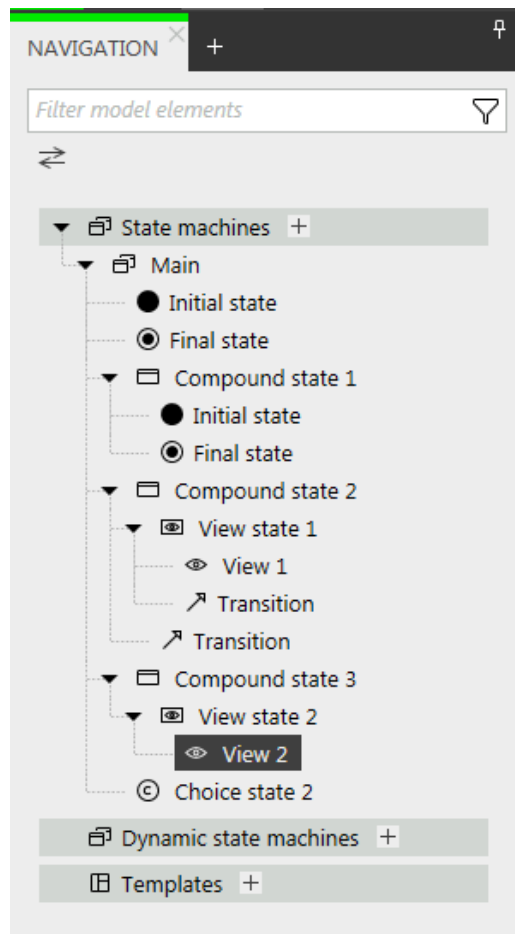


그림 6.4. 프로젝트 편집자의 내비게이션 구성 요소

### 6.5.2.2. 개요 구성 요소

내비게이션 구성 요소 또는 콘텐츠 영역의 현재 표시된 편집자 구성 요소에서 선택한 트리 부분에 포함된 구조 및 모델 요소만 표시합니다.

**[참고]**



**필터 상자**

구성 요소 상단에는 구성 요소 내의 모든 요소를 검색할 수 있는 필터 상자가 있습니다.

구성 요소의 요소를 클릭하고 F3 키를 누르면 다음과 같은 참조 검색이 시작됩니다. 검색 결과 창이 열리고 EB GUIDE 모델에서 선택된 요소의 모든 항목이 나열됩니다.

### 6.5.2.3. 도구 상자 구성 요소

모델링에 필요한 모든 도구는 **도구 상자**라고도 하는 **도구 상자** 구성 요소에서 사용할 수 있습니다. 콘텐츠 영역에 표시되는 요소에 따라 **도구 상자**는 콘텐츠 영역이나 **내비게이션** 구성 요소로 끌어올 수 있는 다른 도구 집합을 제공합니다. 예를 들어 **도구 상자**에는 다음 도구가 포함될 수 있습니다.

- ▶ 콘텐츠 영역에 상태 머신이 표시되는 경우 **도구 상자**에는 상태 머신에 추가할 수 있는 상태가 포함됩니다.
- ▶ 콘텐츠 영역에 보기가 표시되는 경우 **도구 상자**에는 보기에서 정렬할 수 있는 위젯이 포함됩니다.
- ▶ 콘텐츠 영역에 스크립팅된 값 속성이 표시되는 경우 **도구 상자**에는 삽입 가능한 EB GUIDE 스크립트 함수가 포함됩니다.

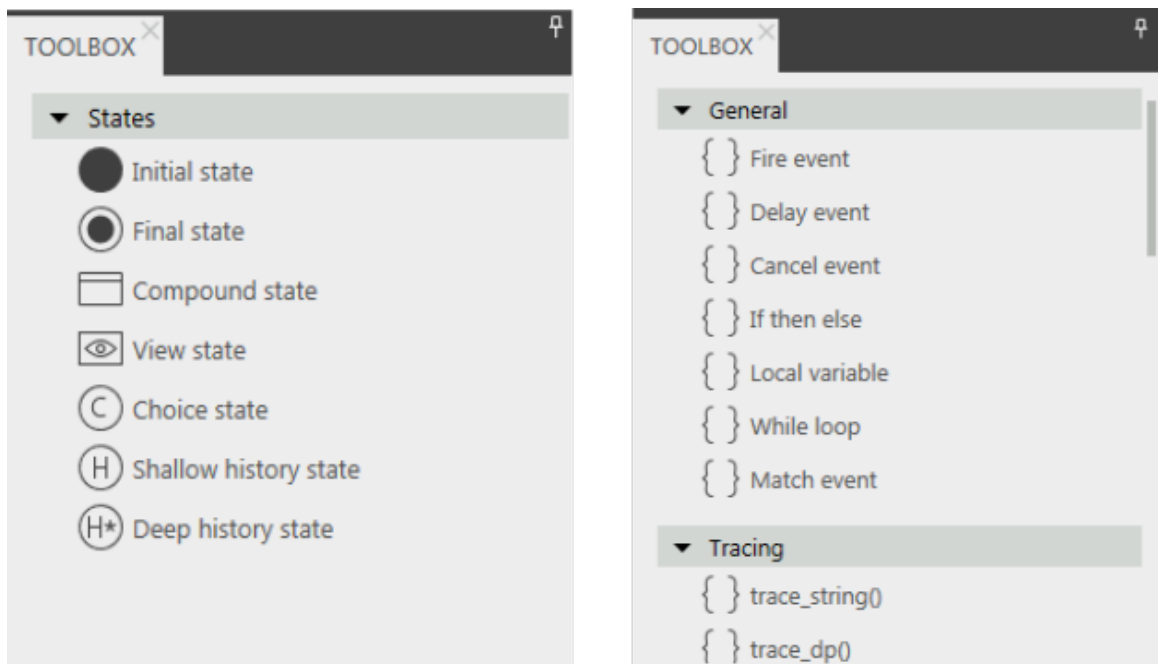


그림 6.5. 프로젝트 편집자의 도구 상자

### 6.5.2.4. 속성 구성 요소

속성 구성 요소에는 예를 들어 위젯이나 상태 등 선택한 모델 요소의 속성이 표시됩니다. 속성은 범주별로 그룹화되며 속성 구성 요소에서 편집할 수 있습니다.

속성을 클릭하고 F3 키를 눌러 참조 검색을 시작합니다. 검색 결과 창이 열리고 EB GUIDE 모델에서 선택된 속성의 모든 항목이 나열됩니다.

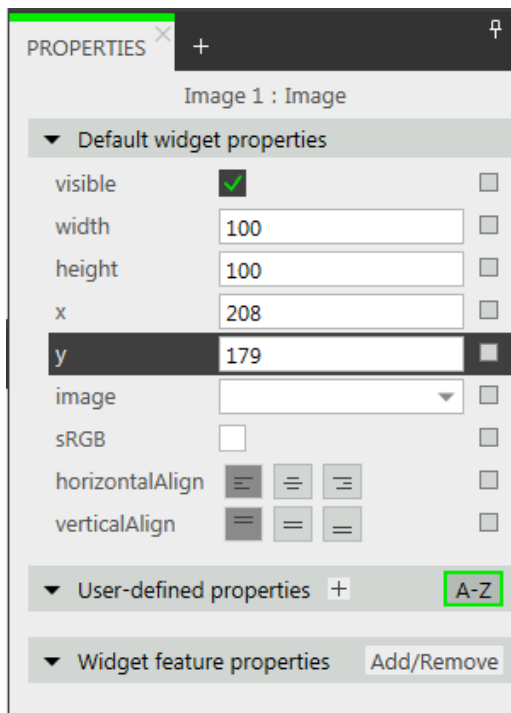


그림 6.6. 위젯의 속성이 표시된 속성 구성 요소

### 6.5.2.5. 콘텐츠 영역

콘텐츠 영역에 표시되는 내용은 **내비게이션** 구성 요소에서 선택한 항목에 따라 달라집니다. 모델 요소를 편집하려면 **내비게이션** 구성 요소에서 모델 요소를 두 번 클릭합니다. 그러면 콘텐츠 영역에 해당 요소가 표시됩니다. 예를 들어 콘텐츠 영역에서는 상태 머신의 상태를 모델링하거나 보기에서 위젯을 정렬하거나 EB GUIDE 스크립트를 편집합니다.

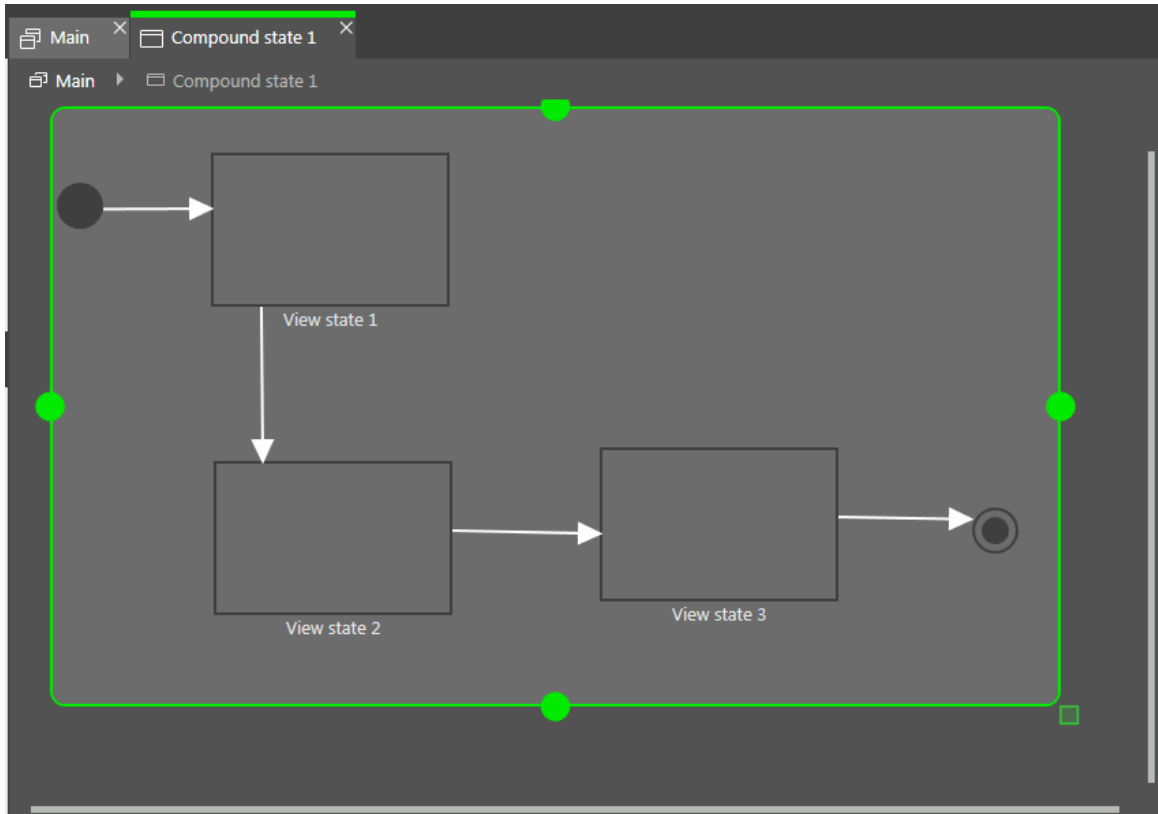


그림 6.7. 프로젝트 편집자의 콘텐츠 영역

콘텐츠 영역에 열린 보기가 있고 보기에 애니메이션이 포함된 경우 **애니메이션** 편집기가 열립니다. **애니메이션** 편집기에서 위젯 속성에 커브를 추가할 수 있습니다. 미리 보기에서 핸들을 움직여 커브의 delay 및 duration 속성을 편집할 수도 있습니다.

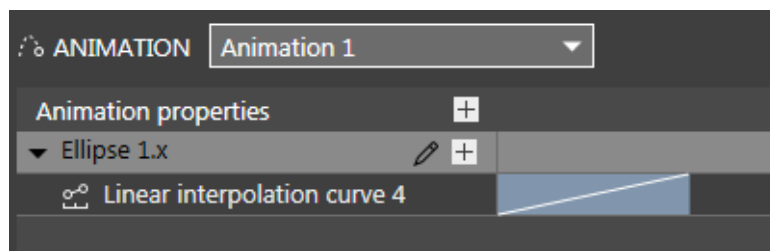


그림 6.8. 애니메이션 편집기

참조 검색을 시작하려면 콘텐츠 영역의 상태 또는 위젯을 클릭하고 F3 키를 누릅니다. 검색 결과 창이 열리고 EB GUIDE 모델에서 선택된 상태 또는 위젯의 모든 항목이 나열됩니다.

**예 6.3.**

### 가져오기 디렉터리 이름

3D 그래픽 파일의 이름은 car.fbx입니다. EB GUIDE Studio의 3D 그래픽 파일을 가져오고 나면 \$GUIDE\_PROJECT\_PATH/<project name>/resources에서 이름이 car\_20160102\_103029인 하위 디렉터리를 찾을 수 있습니다.

#### 6.5.2.6. 이벤트 구성 요소

여기서 모델에 이벤트를 추가하고 이벤트 테이블의 **이름**, **그룹**, **유형** 및 **매개변수 이름**과 같은 특성을 편집할 수 있습니다.

##### [참고]



##### 필터 상자

구성 요소 상단에는 구성 요소 내의 모든 요소를 검색할 수 있는 필터 상자가 있습니다.

구성 요소의 요소를 클릭하고 F3 키를 누르면 다음과 같은 참조 검색이 시작됩니다. 검색 결과 창이 열리고 EB GUIDE 모델에서 선택된 요소의 모든 항목이 나열됩니다.

#### 6.5.2.7. 데이터폴 구성 요소

여기서 **데이터폴** 항목을 추가하고 **이름** 및 **값**과 같은 속성을 편집할 수 있습니다. 데이터폴 항목에 대한 링크를 추가하고 값을 스크립트로 변환하고 언어 및 스킨 지원을 추가할 수도 있습니다.

##### [참고]



##### 필터 상자

구성 요소 상단에는 구성 요소 내의 모든 요소를 검색할 수 있는 필터 상자가 있습니다.

구성 요소의 요소를 클릭하고 F3 키를 누르면 다음과 같은 참조 검색이 시작됩니다. 검색 결과 창이 열리고 EB GUIDE 모델에서 선택된 요소의 모든 항목이 나열됩니다.

#### 6.5.2.8. 자산 구성 요소

여기서 이미지, 글꼴, .ebmesh 및 .psd 파일과 같은 리소스를 추가할 수 있습니다. \$GUIDE\_PROJECT\_PATH/<project name>/resources 디렉터리 및 그 하위 디렉터리에 있는 모든 리소스 파일은 구성 요소의 미리 보기 영역에 표시됩니다. 끌어서 놓기 작업을 이용하여 리소스를 모델에 추가할 수 있습니다.

**[참고]**




**필터 상자**

구성 요소 상단에는 구성 요소 내의 모든 요소를 검색할 수 있는 필터 상자가 있습니다.

**6.5.2.9. 명령 영역**

명령 영역에는 프로젝트 센터를 열 수 있는

- ▶  버튼을 찾을 수 있습니다.
- ▶ 모델의 요소를 검색하고 해당 요소로 이동할 수 있는 검색 상자
- ▶ 추가 메뉴

**검색 상자**

모델 요소는 검색 상자의 도움말을 통해 찾을 수 있습니다. 다음과 같이 검색 상자를 사용합니다.

- ▶ 검색 상자로 이동하려면 검색 상자를 클릭하거나 Ctrl+F 단축키를 사용합니다. 검색될 모델 요소의 이름을 입력합니다.  
  
대안적으로 모델 요소를 선택하고 F3을 누를 수 있습니다. 검색 결과 창이 열리고 검색 결과가 표시 됩니다.
- ▶ 대상 목록의 모델 요소를 두 번 클릭하여 이동합니다.

검색 결과 창의 나머지 요소는 범주별로 그룹화된 모델 요소를 나열합니다. 범주를 표시하거나 숨기려면 위의 필터 버튼을 사용합니다. 미리 보기를 받거나 읽기 전용 모드에서 모델 요소의 속성을 보려면 모델 요소를 선택합니다.

검색 결과 창을 닫으면 마지막으로 검색한 용어, 필터 설정, 해당 대상 목록이 저장되고 검색 결과 창을 다시 열 때 표시됩니다. 모델 요소가 중간에 변경되는 경우 검색이 다시 실행되어야 합니다.

검색은 대/소문자가 구분되지 않습니다.

와일드카드에 대한 별표(\*)를 사용하는 경우 다음과 같은 적용 규칙을 검색합니다.

- ▶ 검색 항목 t는 t를 포함하는 모든 요소 이름을 반환합니다.
- ▶ 검색 항목 \*t는 t로 끝나는 모든 요소 이름을 반환합니다.
- ▶ 검색 항목 t\*는 t로 시작하는 모든 요소 이름을 반환합니다.

다음과 같은 모델 요소 범주를 검색할 수 있습니다.

표 6.2. 검색 상자의 범주

범주	설명
상태	대상 목록에는 검색된 상태의 하위 요소도 표시됩니다.



범주	설명
보기	대상 목록에는 검색된 보기의 하위 요소도 표시됩니다.
템플릿	대상 목록에는 검색된 템플릿의 하위 요소도 표시됩니다.
이벤트	미리 보기에 이벤트의 속성이 표시됩니다.
데이터풀 항목	미리 보기에 데이터풀 항목의 속성이 표시됩니다.
스크립트	미리 보기에 텍스트를 포함하는 스크립트의 콘텐츠가 표시됩니다. 검색된 텍스트가 강조 표시됩니다.
속성	미리 보기에 속성이 속하는 위젯이 표시됩니다.

### 6.5.2.10. 문제 구성 요소

**문제** 구성 요소에서 모델이 유효한지 확인할 수 있습니다. 현재 열려 있는 EB GUIDE 모델에서 발생할 수 있는 오류와 경고가 표시됩니다. 문제가 발생한 부분으로 바로 이동하려면 설명을 더블 클릭합니다.

### 6.5.2.11. VTA 구성 요소

**VTA**(관점 전환 애니메이션) 구성 요소에서 보기 상태 또는 보기 템플릿의 관점 전환 애니메이션을 편집할 수 있습니다. 다른 애니메이션 유형을 선택할 수 있습니다. 상태가 true인 각 애니메이션 유형의 첫 번째 항목 등 모든 관련 애니메이션 유형이 동시에 시작됩니다.

**VTA** 구성 요소는 기본 레이아웃에 표시되지 않습니다. **VTA** 구성 요소를 열려면 **레이아웃** 메뉴에서 **VTA(관점 전환 애니메이션)**를 선택합니다.

## 6.5.3. 도킹할 수 있는 구성 요소

프로젝트의 모든 구성 요소를 탭으로 도킹하거나 부동 구성 요소로 도킹 해제할 수 있습니다. 구성 요소를 부동 구성 요소로서 콘텐츠 영역을 제외한 프로젝트 센터의 모든 부분으로 끌 수 있습니다.

도킹 제어의 화살표를 사용하여 도킹 위치를 선택할 수 있으며 실시간 미리 보기에서 레이아웃이 어떻게 보이는지 확인할 수 있습니다.

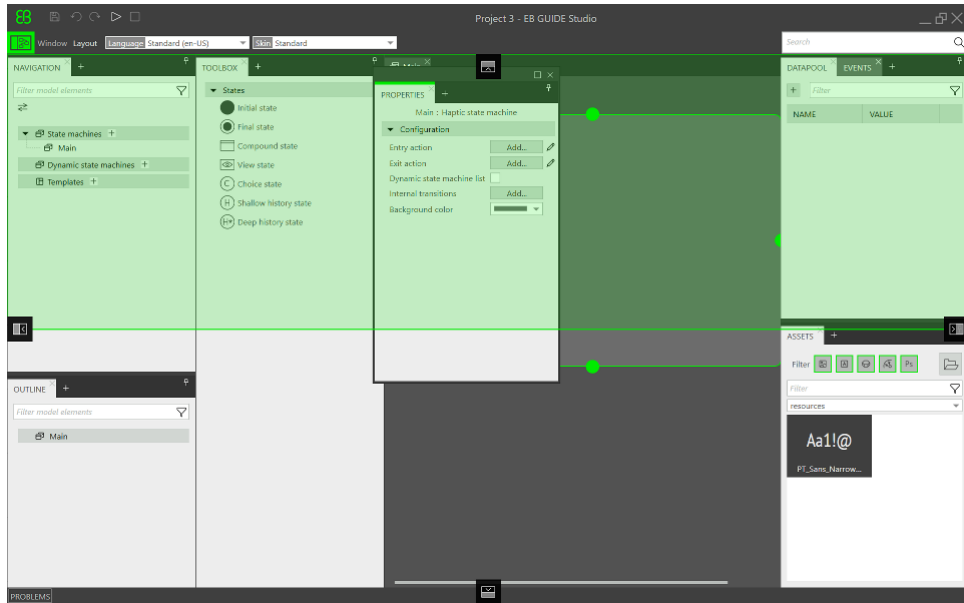


그림 6.9. 도킹 제어 및 실시간 미리 보기

**[참고]**



**기본 레이아웃**

기본 레이아웃으로 복원하려면 명령 영역으로 이동하여 **레이아웃 > 기본 레이아웃으로 복원**을 선택합니다.

**[참고]**



**자동 숨김**

프로젝트 편집자에 추가 공간을 확보하려면 구성 요소를 숨길 수 있습니다.

- ▶ 구성 요소 또는 구성 요소 그룹을 숨기려면 핀 기호를 클릭합니다.
- ▶ 숨겨진 구성 요소를 표시하려면 탭 위에 마우스를 놓고 핀 기호를 다시 클릭합니다.

### 6.5.4. EB GUIDE Monitor

EB GUIDE는 시뮬레이션하는 동안 EB GUIDE 모델을 관찰하고 제어할 수 있는 도구 EB GUIDE Monitor를 제공합니다. EB GUIDE Monitor는 EB GUIDE 모델의 상태 머신, 데이터풀 및 이벤트 시스템과 통신하기 위한 메커니즘을 포함합니다.

EB GUIDE Monitor는 EB GUIDE 모델 시뮬레이션 동안 EB GUIDE Studio에서 자동으로 시작합니다. EB GUIDE Monitor를 독립 실행형 응용 프로그램으로 사용할 수도 있습니다.

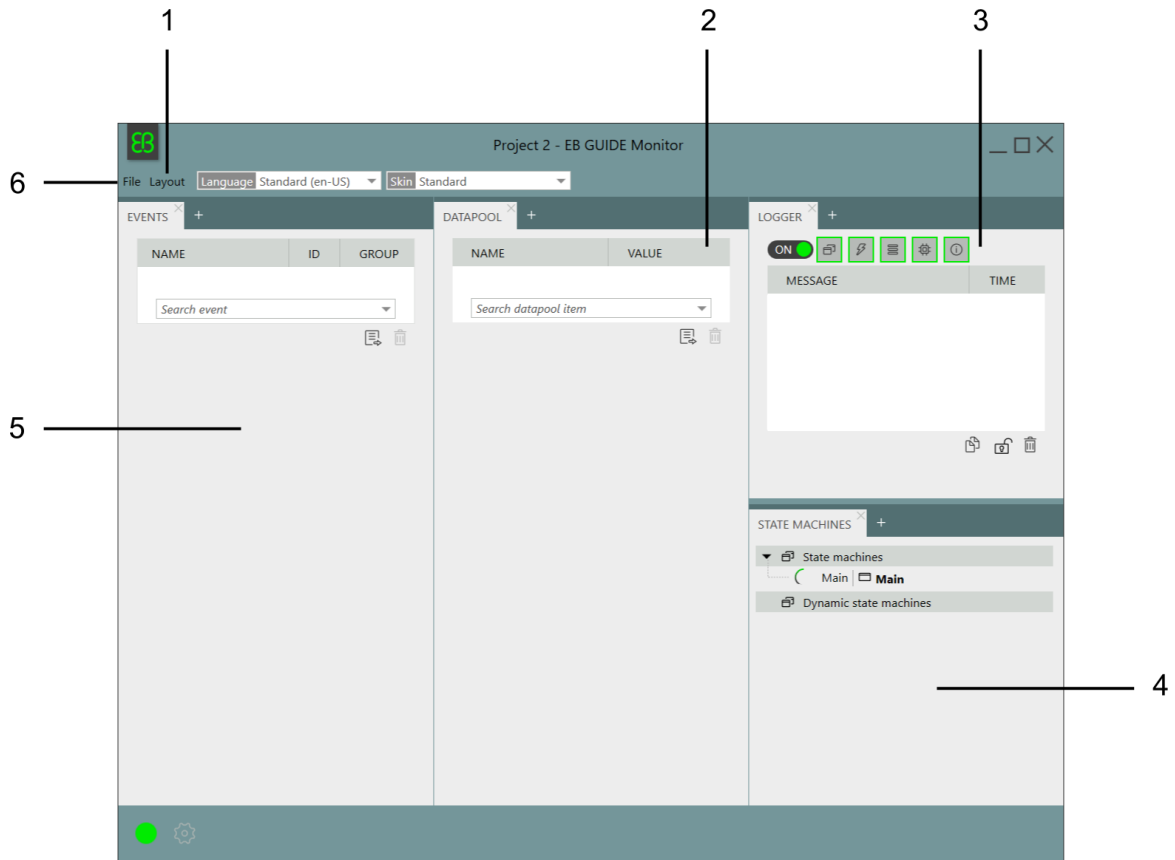


그림 6.10. 기본 레이아웃으로 EB GUIDE Monitor

1 레이아웃 메뉴

2. 데이터풀 구성 요소

3 로거 구성 요소

4 상태 머신 구성 요소

5. 이벤트 구성 요소

6 파일 메뉴




EB GUIDE Monitor는 다음 구성 요소를 포함합니다.

- ▶ **이벤트** 구성 요소에서 이벤트를 검색하고 실행할 수 있습니다. 이벤트에 매개변수가 있는 경우 매개변수를 변경한 다음 이 이벤트를 실행할 수 있습니다.
- ▶ **데이터풀** 구성 요소에서 데이터풀 항목을 검색하고 값을 변경할 수 있습니다.
- ▶ **로거** 구성 요소에서 모든 변경, 정보 메시지, 오류 및 경고가 추적됩니다. 구성 요소 상단에는 구성 요소 내의 항목을 필터링할 수 있는 필터 버튼이 있습니다. 자동 스크롤 기능을 활성화 또는 비활성화하려면 구성 요소 하단의 자동 스크롤 확인란을 선택하거나 선택 취소합니다.

- ▶ **상태 머신** 구성 요소에 현재 활성화인 상태와 상태 머신이 표시됩니다.
- ▶ **스크립팅** 구성 요소에서 스크립트를 시작하고 결과 스크립트 메시지를 볼 수 있습니다. **스크립팅** 구성 요소는 기본 레이아웃이 아닙니다. 구성 요소를 추가하려면 **레이아웃 > 스크립팅**을 클릭합니다.

구성 요소를 재정렬하고 프로젝트에 필요한 새 구성 요소를 추가할 수 있습니다. EB GUIDE Monitor 창 내의 구성 요소를 도킹하거나 도킹 해제할 수도 있습니다.

EB GUIDE Monitor 창의 왼쪽 하단 모서리에서 연결 상태에 대한 다음 버튼을 찾을 수 있습니다.

버튼	상태
	EB GUIDE Monitor이(가) 연결되었습니다. 버튼을 클릭하면 EB GUIDE Monitor의 연결이 해제됩니다.
	EB GUIDE Monitor이(가) 연결 해제되었습니다. 버튼을 클릭하면 EB GUIDE Monitor가 연결됩니다.
	EB GUIDE Monitor이(가) 연결 해제되었습니다. 버튼을 클릭하면 EB GUIDE Monitor의 연결 설정을 구성할 수 있습니다.

명령 영역의 드롭다운 상자를 사용하여 언어 및 스킨을 변경할 수도 있습니다.

관련 지침은 [섹션 10.9. "EB GUIDE Monitor로 작업"](#) 항목을 참조하십시오.

EB GUIDE Monitor API는 \$GUIDE\_INSTALL\_PATH/doc/monitor/monitor\_api.chm을 참조하십시오.

## 6.6. 데이터풀

### 6.6.1. 개념

모델은 실행 중에 여러 응용 프로그램과 통신합니다. 통신을 활성화하려면 EB GUIDE 모델이 인터페이스를 제공해야 합니다. 데이터 교환을 위해 데이터풀 항목 액세스를 허용하는 인터페이스가 데이터풀입니다. 데이터풀 항목은 값을 저장하며 HMI와 응용 프로그램 간에 교환됩니다. EB GUIDE 모델에서 데이터풀 항목을 정의합니다.

### 6.6.2. 데이터풀 항목

데이터풀 항목은 다음 작업을 수행하는 데 사용됩니다.

- ▶ 응용 프로그램에서 HMI로 데이터 전송
- ▶ HMI에서 응용 프로그램으로 데이터 전송
- ▶ HMI 또는 응용 프로그램 중 하나에서만 사용되는 데이터 저장

관련 지침은 [섹션 9.5. "데이터폴 항목 추가"](#) 항목을 참조하십시오.

채널 통신에는 기록기 및 독자 응용 프로그램을 사용합니다.

내부 통신은 데이터를 저장하는 데 사용됩니다. 서로 다른 두 통신 응용 프로그램을 사용하면 외부 통신이 설정됩니다.

관련 지침은 [섹션 9.8. "외부 통신 설정"](#) 항목을 참조하십시오.

### 6.6.3. 창 모드 목록

EB GUIDE product line은 창 모드 목록의 개념을 지원합니다. 창 모드 목록 작동 모드는 예를 들어 디렉터리의 모든 MP3 제목과 같은 큰 목록을 표시할 때 메모리 사용량을 줄이는 데 사용되는 경우가 많습니다. 이러한 목록은 일반적으로 미디어 응용 프로그램과 같은 단일 응용 프로그램에 의해 제공되며, 예를 들어 HMI 등의 다른 응용 프로그램에서는 부분적으로만 표시됩니다.

기록기 응용 프로그램은 가상 목록 길이 및 창의 수를 정의합니다. 이러한 창에는 목록의 일부분만 포함될 수 있습니다. 독자 응용 프로그램은 창에 포함되는 위치에서만 데이터를 읽습니다. 다른 위치로부터의 읽기는 실패합니다. 이러한 사용 사례에서 독자 응용 프로그램은 현재 필요한 목록 부분을 기록기 응용 프로그램에 알려야 합니다. 예를 들어 HMI는 전체 목록 내의 현재 커서 위치를 제공하는 응용 프로그램 호출을 수행할 수 있습니다.



#### 예 6.4. 창 모드 목록

오디오 플레이어 기기의 MP3 타이틀 목록에 요소가 100만 개 포함되어 있다고 가정해 보겠습니다. HMI는 서로 다른 3개 디스플레이(헤드 장치 디스플레이, 클러스터 장비 디스플레이, 헤드업 디스플레이)에 이 목록을 표시해야 합니다.

각 디스플레이는 개별적으로 제어되고, 디스플레이 줄 수가 서로 다르며, 전체 목록 내에서의 커서 위치도 각기 다릅니다.

3개 커서 중 하나가 이동하면 HMI는 이벤트를 통해 새 위치를 비동기 방식으로 미디어 응용 프로그램에 전송합니다. 미디어 응용 프로그램은 3개 창이 있는 목록을 제공합니다. 3개 창은 각각 3개 디스플레이 중 하나와 연결됩니다. 커서가 이동한 후 창 업데이트가 약간 연기됩니다. 따라서 특정 디스플레이에 표시되는 줄 주위의 확장된 범위를 포함하는 창 크기와 창 위치를 사용하는 것이 좋습니다.

## 6.7. EB GUIDE 모델 및 EB GUIDE 프로젝트

EB GUIDE 모델에는 HMI의 모양과 동작을 설명하는 모든 요소가 포함되어 있습니다. 이 모델은 EB GUIDE Studio에서 완전히 작성됩니다. PC에서 EB GUIDE 모델을 시뮬레이션할 수 있습니다.

타깃 기기에서 EB GUIDE 모델을 실행하려면 EB GUIDE 모델을 내보낸 다음 생성된 이진 파일을 타깃 기기에 복사합니다.

EB GUIDE 프로젝트는 EB GUIDE 모델과 모델링에 필요한 설정으로 구성됩니다. 이러한 설정에는 프로젝트 관련 옵션, 확장, 리소스, 그리고 그래픽 프로젝트의 경우 햅틱 대화 상자의 설명이 포함됩니다.

EB GUIDE 프로젝트는 EB GUIDE 모델 내에서 구성 및 연결되는 개체를 포함합니다. 이러한 개체를 EB GUIDE 모델 요소라고 합니다. EB GUIDE 모델 요소의 예는 다음과 같습니다.

- ▶ 데이터폴 항목
- ▶ 이벤트
- ▶ 상태
- ▶ 상태 기계
- ▶ 위젯
- ▶ 리소스
- ▶ 언어

## 6.8. 이벤트 처리

### 6.8.1. 이벤트 시스템

이벤트 시스템은 응용 프로그램 내에서 또는 응용 프로그램 간에 수행되는 통신을 위한 비동기 메커니즘입니다.

EB GUIDE 이벤트 시스템은 모든 이벤트를 정확히 전송된 순서로 제공합니다. 각 구독자에게 이벤트를 제공하는 미리 정의된 순서는 없습니다.

### 6.8.2. 이벤트

EB GUIDE의 이벤트는 고유한 이벤트 ID를 포함하며 이벤트 그룹에 속하는 모델 요소입니다. 이벤트 ID는 EB GUIDE GTF에서 이벤트를 송수신하는 데 사용됩니다.



#### 예 6.5. 이벤트 사용

HMI에는 로터리 버튼 또는 왼쪽, 오른쪽, 위, 아래, enter 등의 하드 키 세트가 있으며 사용자가 다음에 어떤 요소와 상호 작용해야 하는지 분명하지 않을 수 있습니다. 따라서 이러한 시스템은 일반적으로 현재



활성 디스플레이 요소를 강조 표시합니다. 예를 들어 YES 버튼은 색이 있는 경계선으로 활성 상태로 표시됩니다. EB GUIDE Studio에서 **포커스됨** 위젯 피처를 이용해 이 강조 표시 피처를 모델링합니다. 이 요소는 현재 포커스됨 상태이므로, 즉 focused 속성이 true로 설정되어 있으므로 이 요소도 활성 상태입니다. 또한 포커스 경로를 형성하는 이 요소의 상위로 활성 상태입니다. 포커스됨 요소가 키 또는 로터리 입력을 처리할 수 없는 경우 해당 입력은 반대 방향으로, 즉 루트 요소를 향한 포커스 경로를 따라 처리됩니다. 포커스 경로의 요소 중 하나가 입력을 처리하면 해당 포커스가 처리된 것으로 간주됩니다.

터치 입력이 있는 HMI에서 상호 작용은 특정 위치의 요소를 통해 이루어집니다. 예를 들어 터치스크린의 YES 버튼을 누를 경우 입력이 모호하지 않습니다. 그 이유는 디스플레이 상의 눌린 위치를 토대로 시스템이 YES 버튼과의 상호 작용이 수행되었음을 인지하기 때문입니다.

이벤트 그룹 ID 0~65535는 EB GUIDE product line 내에서 내부용으로 예약되어 있습니다. 단, 다음 표에 나와 있는 이벤트 그룹의 경우는 예외입니다.

표 6.3. 허용되는 이벤트 그룹 및 ID

이벤트 그룹	ID	세부 정보									
기본값	2	내부 그룹, 즉 장면이 실행되는 코어가 이벤트를 수신합니다.									
키 입력 이벤트	10	다음 매개변수를 구성할 수 있습니다.									
		<table border="1"> <thead> <tr> <th>매개변수</th> <th>유형</th> <th>세부 정보</th> </tr> </thead> <tbody> <tr> <td>display</td> <td>정수</td> <td>입력 이벤트를 수신해야 하는 장면 ID</td> </tr> <tr> <td>status</td> <td>정수</td> <td>0: 키 누름 1: 키 놓음 2: 키 유니코드</td> </tr> </tbody> </table>	매개변수	유형	세부 정보	display	정수	입력 이벤트를 수신해야 하는 장면 ID	status	정수	0: 키 누름 1: 키 놓음 2: 키 유니코드
		매개변수	유형	세부 정보							
display	정수	입력 이벤트를 수신해야 하는 장면 ID									
status	정수	0: 키 누름 1: 키 놓음 2: 키 유니코드									
터치 입력 이벤트	11	다음 매개변수를 구성할 수 있습니다.									
		<table border="1"> <thead> <tr> <th>매개변수</th> <th>유형</th> <th>세부 정보</th> </tr> </thead> <tbody> <tr> <td>display</td> <td>정수</td> <td>입력 이벤트를 수신해야 하는 장면 ID</td> </tr> <tr> <td>status</td> <td>정수</td> <td>0: 터치 누름 1: 터치 이동 2: 터치 놓음 3: 근접 이동 4: 새 항목 터치 5: 터치 이동됨</td> </tr> </tbody> </table>	매개변수	유형	세부 정보	display	정수	입력 이벤트를 수신해야 하는 장면 ID	status	정수	0: 터치 누름 1: 터치 이동 2: 터치 놓음 3: 근접 이동 4: 새 항목 터치 5: 터치 이동됨
		매개변수	유형	세부 정보							
display	정수	입력 이벤트를 수신해야 하는 장면 ID									
status	정수	0: 터치 누름 1: 터치 이동 2: 터치 놓음 3: 근접 이동 4: 새 항목 터치 5: 터치 이동됨									

이벤트 그룹	ID	세부 정보		
		매개변수	유형	세부 정보
		x	정수	터치 이벤트의 x 좌표입니다.
		y	정수	터치 이벤트의 y 좌표입니다.
		fingerId	정수	멀티 터치 지원을 위해 여러 병렬 터치 위치를 추적하는 숫자
회전식 입력 이벤트	12	다음 매개변수를 구성할 수 있습니다.		
		display	정수	입력 이벤트를 수신해야 하는 장면 ID
		increment	정수	증분 값
시스템 알림 이벤트	13	Android에서의 화면 회전 등의 시스템 이벤트 또는 끄기 등의 라이프 사이클 관리에 사용됩니다.		

다음 그림은 EB GUIDE Studio에서 터치, 키, 로터리 이벤트를 모델링하는 방법을 보여줍니다.

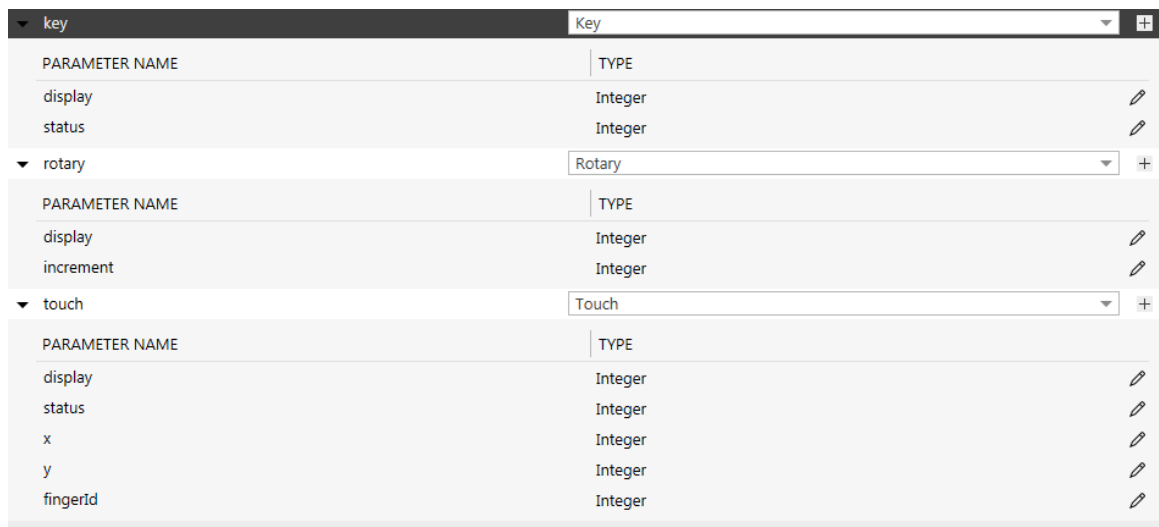


그림 6.11. 이벤트 그룹 및 이벤트 ID의 예시

그룹 ID의 나머지 범위를 고객 관련 응용 프로그램에 사용할 수 있습니다.

지침은 다음을 참조하십시오.

- ▶ [섹션 9.1. "이벤트 추가"](#)
- ▶ [섹션 9.3. "이벤트 주소 지정"](#)

참조 자료는 [섹션 12.5. "이벤트"](#)에서 확인하십시오.



## 6.9. 확장

### 6.9.1. EB GUIDE Studio 확장

EB GUIDE Studio 확장은 EB GUIDE Studio에 대한 보충이며 모든 EB GUIDE 모델에 유효합니다. EB GUIDE Studio 확장은 EB GUIDE GTF와는 관련이 없습니다.

일반적인 EB GUIDE Studio 확장은 다음과 같습니다.

- ▶ 추가 도구 모음 버튼
- ▶ 추가 데이터 내보내기

### 6.9.2. EB GUIDE GTF 확장

EB GUIDE GTF 확장은 EB GUIDE Studio에서 추가 기능을 제공하는 EB GUIDE GTF에 대한 보충이지만 EB GUIDE 모델 하나에만 사용할 수 있습니다. EB GUIDE GTF 확장은 EB GUIDE GTF를 기반으로 합니다.

일반적인 EB GUIDE GTF 확장은 다음과 같습니다.

- ▶ 새 위젯 피쳐
- ▶ 새 EB GUIDE 스크립트 함수

EB GUIDE GTF 확장은 동적 연결 라이브러리(.dll) 또는 공유 객체(.so) 파일입니다.

EB GUIDE GTF 확장은 서드파티 라이브러리와 함께 다음 디렉터리에 배치합니다.

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

## 6.10. 감마 정정 렌더링

### 6.10.1. 개념

감마 정정은 렌더링 파이프라인에서 중요한 역할을 합니다. 감마 정정은 화면의 색 재현 및 이미지 색상 저장소에 영향을 미칩니다. 감마는 색상 값과 화면상의 인지된 강도 사이의 관계를 나타내며 감마의 예는 [그림 6.12. "감마의 예"](#)에서 확인해 보십시오.

HVS(인간 시각 시스템)도 유사한 동작을 보입니다. 밝은 이미지 영역 속의 휘도 차이보다는 어두운 이미지 영역 속의 휘도 차이에 더 민감합니다. 일반적인 8비트 이미지 형식(JPEG, PNG)이 이 사실을 보여주며 어두운 이미지 영역 내의 정밀도를 높이기 위해 비선형 전송 기능을 이용해 색상을 sRGB 색공간에 저장합니다. 3D 조

명 계산과 텍스처 처리된 알파 블렌딩은 선형 텍스처 색상 입력에 의존하므로 이에 영향을 미칩니다. 이러한 이유로 EB GUIDE에서는 이러한 효과에 대응할 수 있도록 [그림 6.13, "sRGB 텍스처의 예"](#)에서 볼 수 있듯이 감마 정정 렌더링을 이용합니다.

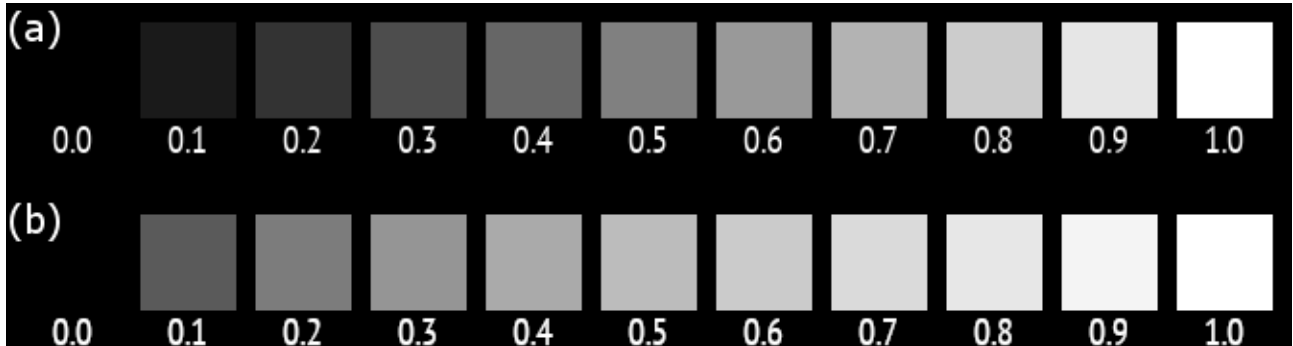


그림 6.12. 감마의 예

(a) 색상이 있는 정사각형 아래의 값은 회색 레벨을 나타냅니다. 색상 값과 표시된 휘도 사이의 비선형 관계로 인해 올바르게 보정된 모니터의 경우 50%의 밝기가 약 0.7의 회색 레벨에 달합니다. 화면에 대한 비선형 휘도 응답에도 불구하고 상대 편차는 영구적으로 일정합니다. (b) 표시 전에 감마가 인코딩된 색상 값입니다. 감마를 인코딩하면 디스플레이의 감마가 취소되므로 50% 밝기가 0.5 회색 레벨에 달하게 됩니다.

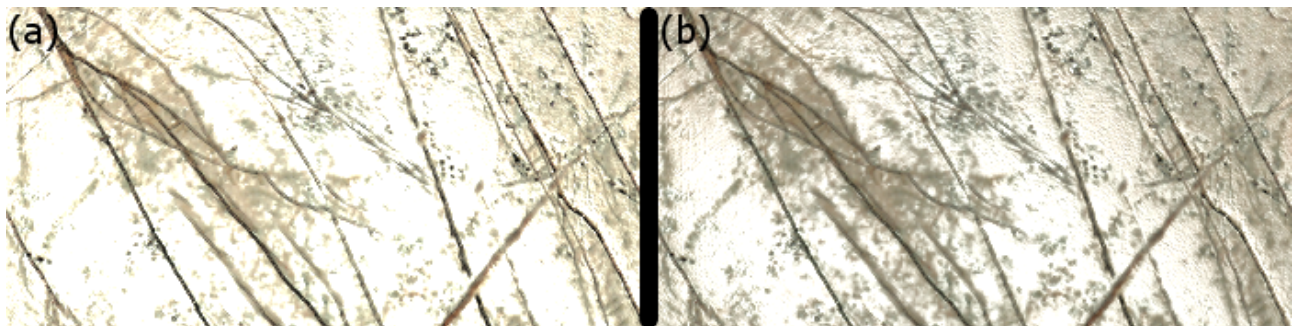


그림 6.13. sRGB 텍스처의 예

(a) sRGB 텍스처는 조명 계산에서 정정되지 않은 채로 남으며 출력은 감마 정정되지 않은 상태입니다. 조명이 과포화되고 상세한 부분이 날아가 버립니다. (b) sRGB 텍스처는 조명 전에 선형화되며 결과는 감마 정정됩니다. 상세한 부분과 표면 구조가 눈에 보이게 됩니다.

### 6.10.2. EB GUIDE Studio에서의 감마 정정

EB GUIDE Studio에서의 감마 정정 렌더링에는 다음과 같이 사용자가 렌더링 파이프라인을 구성하는 과정이 필요합니다.

- ▶ 출력 구성은 디스플레이 자체에 대한 감마 인코딩을 제어합니다.
- ▶ 입력 구성은 렌더링 파이프라인이 렌더링 작업을 위해 이미지와 텍스처 리소스를 적절하게 선형화할 수 있도록 어떤 것이 sRGB 이미지로 처리되어야 하는지 EB GUIDE Studio에 알려줍니다.

EB GUIDE Studio 렌더링 파이프라인의 출력을 감마 인코딩하는 경우 다음과 같은 이유로 인해 sRGB 형식이 사용됩니다.

- ▶ sRGB 형식은 최신 GPU에 의해 하드웨어에서 직접 지원됩니다.
- ▶ sRGB 인코딩은 디스플레이의 전송 기능과 거의 유사합니다.
- ▶ 인코딩은 정상적인 8비트 이미지의 인코딩과 동일합니다.

입력 인코딩을 구성하려면 사용되는 각 이미지 또는 텍스처에 대해 구성되어야 합니다. 이미지 형식 자체는 sRGB 인코딩에 관한 정보를 제공하지 않는다는 점에 유의하십시오. 그 전에 이 정보를 습득해야 합니다. 관련 지침은 [섹션 11.8. "자습서: 감마를 올바르게 렌더링하기"](#) 항목을 참조하십시오.

## 6.11. 이미지 기반 조명

### 6.11.1. 개념

이미지 기반 조명(IBL)은 실제 환경의 조명 정보를 이용해 장면 내의 가상 3D 개체를 비추는 기술입니다.

조명 정보를 저장하기 위해 높은 동적 범위의 이미지 데이터를 지원하는 이미지 형식이 필요합니다. 휴대용 유동 지도(.pfm)는 이러한 종류의 이미지를 지원하는 단순한 형식입니다. 정육면체, 구 또는 위도 경도 매개변수화로 환경을 나타낼 수 있습니다. 휴대용 유동 지도 파일은 IBLGenerator라 불리는 외부 명령줄 도구에서 입력으로 사용됩니다. IBLGenerator는 환경 조명 정보를 처리하고 EB GUIDE Studio에서 이미지 기반 조명으로 사용되는 데이터 세트를 생성합니다. 이 데이터 세트는 .ebibl 파일로 저장됩니다.

이미지 기반 조명 사용 방법에 대한 특정 지침은 [섹션 11.9. "자습서: 이미지 기반 조명 사용"](#)에서 참조하십시오.

ibl 데이터 유형에 대한 자세한 내용은 [섹션 12.3.7. "ibl"](#) 항목을 참조하십시오.

이미지 기반 조명 위젯에 대한 자세한 내용은 [섹션 6.21. "위젯"](#) 및 [섹션 12.12.4.4. "이미지 기반 조명"](#)에서 참조하십시오.

### 6.11.2. 제한

OpenGL 렌더러를 사용할 경우 OpenGL ES 2.0 드라이버가 다음 OpenGL 확장을 지원해야 합니다.

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

이러한 확장 중 하나가 지원되지 않는다면 이미지 기반 조명 위젯이 무시됩니다. OpenGL 3 렌더러 및 DirectX 렌더러는 언제나 이미지 기반 조명을 지원합니다.

EB GUIDE Studio에서는 장면 그래프당 하나의 이미지 기반 조명만 허용됩니다. 하나 이상을 추가하면 계층상 첫 번째에 해당하는 이미지 기반 조명이 장면에 영향을 미칩니다.

## 6.12. 언어

### 6.12.1. EB GUIDE Studio의 표시 언어

EB GUIDE Studio에서는 그래픽 사용자 인터페이스용으로 여러 표시 언어를 제공합니다. 프로젝트 센터의 **옵션** 탭에서 표시 언어를 선택합니다.

관련 지침은 [섹션 10.6. "EB GUIDE Studio의 표시 언어 변경"](#) 항목을 참조하십시오.

### 6.12.2. EB GUIDE 모델의 언어

대부분의 휴먼 기계 인터페이스에서는 사용자의 기본 설정 언어로 텍스트를 표시할 수 있습니다. EB GUIDE에서도 이러한 언어 관리 기능이 제공됩니다. 프로젝트 구성에서 EB GUIDE 모델의 언어를 추가합니다.

관련 지침은 [섹션 8.4.1. "언어 추가"](#) 항목을 참조하십시오.

#### [참고]



#### 사용할 수 있는 스킨 지원 없음

데이터풀 항목에 대한 언어 지원을 정의한 경우 동일한 항목에 스킨 지원을 추가할 수 없습니다.

데이터풀 항목이 언어에 따라 달라지도록 설정할 수 있습니다. 데이터풀 항목은 각 언어의 값을 정의합니다. 언어를 지원하려면 **언어 지원** 속성을 선택합니다.



#### 예 6.6.

#### 언어에 따라 달라지는 텍스트

프로젝트 구성에서 영어, 독일어, 프랑스어의 3개 언어를 추가했다고 가정해 보겠습니다. 이 경우 데이터풀 항목의 값은 Welcome(영어), Willkommen(독일어), Bienvenue(프랑스어)가 됩니다.

관련 지침은 [섹션 11.6. "자습서: 데이터풀 항목에 언어별로 다른 텍스트 추가"](#) 항목을 참조하십시오.

런타임 동안 내보낸 EB GUIDE 모델의 현재 언어를 설정할 수 있습니다.

### 6.12.3. 언어별로 다른 텍스트 내보내기 및 가져오기

EB GUIDE Studio의 내보내기 및 가져오기 기능을 사용하여 모든 언어별로 다른 텍스트를 내보내고 편집하고 가져옵니다. 텍스트를 .xliff 파일로 내보내고 파일을 번역자에게 전달합니다. .xliff(XML Localization Interchange 파일 형식)는 추출된 데이터를 저장하고 현지화 프로세스의 단계 간에 데이터를 이동하는 데 사용되는 XML 기반 형식입니다.

번역 후에는 번역된 .xliff 파일을 EB GUIDE Studio의 해당하는 언어로 가져옵니다.

관련 지침은 [섹션 10.8. "언어별로 다른 텍스트 내보내기 및 가져오기"](#) 항목을 참조하십시오.

## 6.13. Photoshop 파일 형식 지원

EB GUIDE Studio에서는 .psd 파일 형식을 지원합니다. EB GUIDE Studio에서 .psd파일을 가져온 후 위젯 트리가 생성됩니다. 위젯 트리는 .psd 파일의 레이어에서 가져오는 동안 생성된 이미지와 컨테이너로 구성됩니다. 다음 내용을 확인하십시오.

- ▶ .psd 파일의 레이어가 표시되지 않도록 설정된 경우 해당 컨테이너 또는 이미지의 visible 속성 옆에 있는 확인란이 표시 해제됩니다.
- ▶ .psd 파일의 레이어가 투명값으로 설정된 경우 가져온 후 **천연색** 위젯 피처가 해당 이미지가 추가됩니다. colorationColor 속성의 알파 채널은 .psd 파일과 동일한 투명값으로 설정됩니다.

관련 지침은 [섹션 8.1.4. "보기에 .psd 파일 추가"](#) 항목을 참조하십시오.

## 6.14. 리소스 관리

리소스는 EB GUIDE 내에서 만들어지지는 않지만 프로젝트에 필요한 콘텐츠입니다. resources 디렉터리에서 EB GUIDE Studio 프로젝트의 모든 리소스를 찾을 수 있습니다.

resources 디렉터리는 \$GUIDE\_PROJECT\_PATH/<project name>/resources에 있습니다.

EB GUIDE에서는 리소스 파일의 다음 유형을 지원합니다.

1. 글꼴
2. 3D 그래픽의 ibl
3. 이미지
4. 3D 그래픽의 메시
5. .psd 파일 형식

프로젝트에서 리소스를 사용하려면 \$GUIDE\_PROJECT\_PATH/<project name>/resources 디렉터리에 리소스 파일을 추가합니다.

## 6.14.1. 글꼴

프로젝트에서 글꼴을 사용하려면 \$GUIDE\_PROJECT\_PATH/<project name>/resources 디렉터리에 글꼴을 추가합니다.

지원되는 글꼴 유형은 트루타입 글꼴(\*.ttf, \*.ttc)과 오픈타입 글꼴(\*.otf) 및 비트맵 글꼴(\*.fnt)입니다.

관련 지침은 [섹션 8.1.2.4.1. "레이블의 글꼴 변경"](#) 항목을 참조하십시오.

### 6.14.1.1. 비트맵 글꼴

EB GUIDE Studio은 버전 3.0에서 Angelcode의 \*.fnt 비트맵 글꼴을 지원합니다. 비트맵 글꼴을 생성하려면 Angelcode 비트맵 글꼴 생성기 등의 서드파티 글꼴 생성기를 사용합니다. 자세한 내용은 <http://www.angelcode.com>을 참조하십시오.

생성된 글꼴은 다음과 같이 설정되어야 합니다.

- ▶ 원하는 글꼴 크기를 정의합니다.
- ▶ 문자 설정은 유니코드입니다.
- ▶ 글꼴 설명자는 이진수입니다.
- ▶ 질감은 8비트 .png 파일로 제공됩니다.

다음 내용을 확인하십시오.

- ▶ EB GUIDE Studio에서 레이블의 font 속성을 이용하여 비트맵 글꼴의 글꼴 크기를 변경할 수 없습니다. 이는 .fnt 글꼴을 생성할 때 크기를 정의해야 한다는 의미입니다.
- ▶ **스트로크** 위젯 피처는 비트맵 글꼴에 적용되지 않습니다. 글꼴에 특정 윤곽선이 필요한 경우 .fnt 글꼴을 생성할 때 정의하십시오.
- ▶ \$GUIDE\_PROJECT\_PATH/resources 디렉터리에서 서드파티 도구로 생성한 .fnt 비트맵 글꼴 및 .png 질감 파일을 위한 하위 디렉터리를 생성합니다. EB GUIDE Studio에서는 .png 파일을 .fnt 파일과 같은 디렉터리에서 찾을 수 있습니다.

여러 개의 비트맵 글꼴이 있는 경우 각 해당 글꼴을 위한 하위 디렉터리를 생성하십시오.

## 6.14.2. 3D 그래픽을 위한 이미지 기반 조명

EB GUIDE Studio에서 이미지 기반 조명을 이용할 수 있습니다. 외부 명령줄 도구 IBLGenerator에서는 .pfm 파일을 입력 데이터로 취해 IBL 리소스를 나타내는 .ebibl 파일을 생성합니다. IBL 리소스는 이미지 기반 조명 위젯의 ibl 속성에 의해 사용됩니다.

### 6.14.3. 이미지

프로젝트에서 이미지를 사용하려면 \$GUIDE\_PROJECT\_PATH/<project name>/resources 디렉터리에 이미지를 추가합니다. 다른 디렉터리에서 선택하는 이미지는 이 폴더에 복사됩니다.

지원되는 이미지 형식은 Portable Network Graphic(\*.png), JPEG(\*.jpg) 및 나인패치 이미지(\*.9.png)입니다.

관련 지침은 [섹션 8.1.2.3. "이미지 추가"](#) 항목을 참조하십시오.

#### 6.14.3.1. 나인패치 이미지

EB GUIDE Studio에서는 나인패치 방식에 따라 추가 메타 정보가 포함된 이미지를 지원합니다. 나인패치 이미지는 늘릴 수 있는 .png 이미지입니다. 나인패치 이미지에는 이미지 왼쪽과 위쪽에 각각 하나씩 두 개의 검은색 표식이 포함되어 있습니다. 표식이 없는 영역은 크기가 조정되지 않습니다. 표식이 있는 영역은 크기를 조정할 수 있습니다. 마커는 EB GUIDE Studio에 표시되지 않습니다.

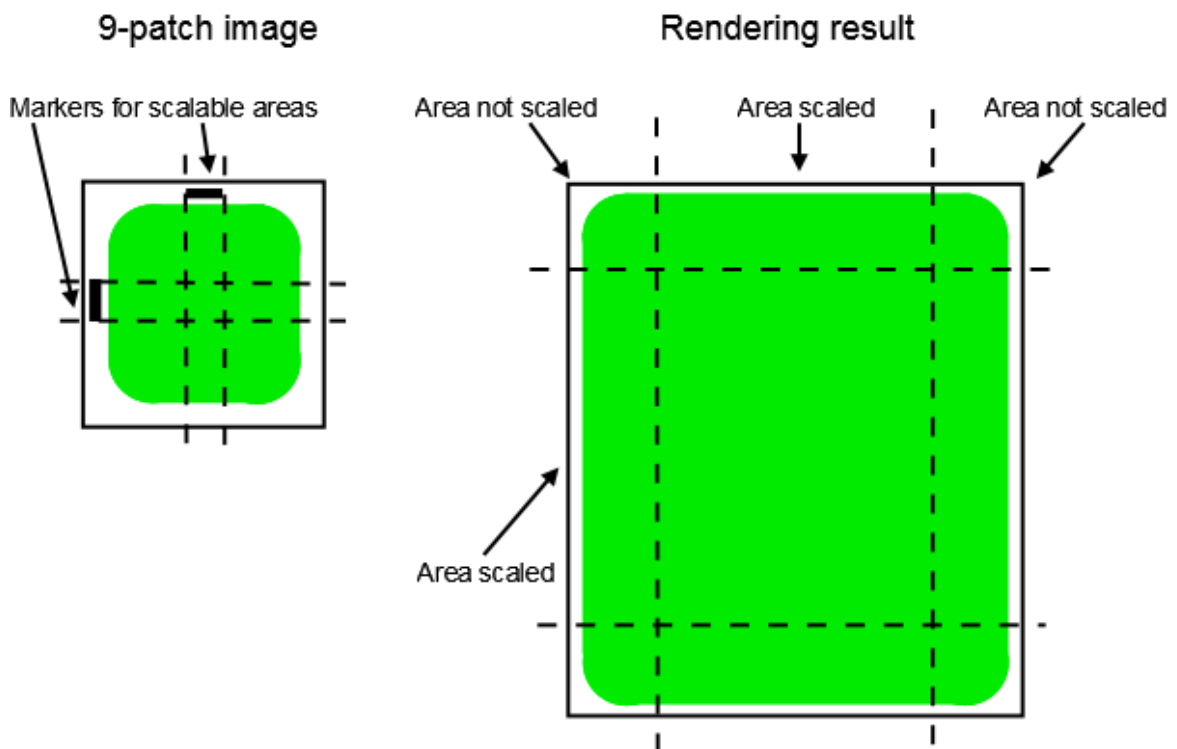


그림 6.14. 나인패치 예제

나인패치 이미지로 작업할 때는 다음 사항을 고려하십시오.

- ▶ 나인패치 프로세스는 OpenGL ES 버전 2.0 이상 및 DirectX 렌더러에서만 작동합니다.
- ▶ 나인패치 처리는 .png 이미지에서만 작동합니다.
- ▶ \*.9.png 확장자는 나인패치 이미지의 의무 사항입니다.
- ▶ 왼쪽과 위쪽에 마커를 지정하지 않을 수도 있고 하나 또는 두 개 이상 지정할 수도 있습니다. 나인패치 정의에는 이미지 아래쪽과 오른쪽의 텍스트 영역에 대한 표식도 포함됩니다. 이러한 마커는 EB GUIDE Studio에서 평가되지 않습니다.

관련 지침은 [섹션 8.1.2.3. "이미지 추가"](#) 항목을 참조하십시오.

#### 6.14.4. 3D 그래픽의 메시

EB GUIDE Studio에서는 3D 그래픽 파일을 가져올 수 있습니다. EB GUIDE Studio의 3D 그래픽 파일을 가져오고 나면 \$GUIDE\_PROJECT\_PATH/<project name>/resources에서 하위 디렉터리를 찾을 수 있습니다. 3D 그래픽 파일에 정의된 메시가 .ebmesh 파일로 가져오기됩니다. 자세한 내용은 [섹션 6.1.3. "3D 그래픽 파일 가져오기"](#) 항목을 참조하십시오.

관련 지침은 [섹션 8.1.3.1. "보기에 장면 그래프 추가"](#) 항목을 참조하십시오.

### 6.15. 스크립팅 언어 EB GUIDE 스크립트

EB GUIDE 스크립트는 EB GUIDE의 내장 스크립팅 언어입니다. 이 장에서는 EB GUIDE 스크립트 언어 기능, 구문 및 사용법에 대해 설명합니다.

#### 6.15.1. 응용 프로그램의 기능과 영역

프로젝트의 다음과 같은 여러 위치에서 EB GUIDE 스크립트를 사용할 수 있습니다. 예를 들어 다음과 같습니다.

- ▶ 위젯 속성
- ▶ 상태 기계(전이나 상태의 요소)
- ▶ 데이터폴 항목

모든 경우에 EB GUIDE 스크립트의 모든 기능을 사용할 수 있는 것은 아닙니다. 예를 들어 스크립트가 위젯의 요소인 경우에만 로컬 위젯 속성에 액세스할 수 있습니다. 반면 데이터폴에는 항상 액세스가 가능합니다.



EB GUIDE 스크립트를 사용하면 모델 요소를 직접 조작하여 예를 들어 다음과 같은 작업을 수행할 수 있습니다.

- ▶ 이벤트 실행
- ▶ 데이터폴 항목 쓰기
- ▶ 위젯 속성 수정

## 6.15.2. 네임스페이스 및 식별자

EB GUIDE에서는 여러 종류의 개체에 같은 이름을 지정할 수 있습니다. 예를 들어 이벤트와 데이터폴 항목 둘 다의 이름을 Napoleon으로 지정할 수 있습니다. EB GUIDE 스크립트 네임스페이스를 사용하면 이와 같은 동일 이름 지정이 가능합니다. EB GUIDE 스크립트의 모든 식별자(개체 이름)에는 네임스페이스와 콜론을 접두사로 추가해야 합니다.

EB GUIDE 스크립트에서 네임스페이스 집합은 고정되어 있으며 새 네임스페이스를 추가할 수는 없습니다. 포함되어 있는 네임스페이스는 다음과 같습니다.

- ▶ ev: 이벤트
- ▶ dp: 데이터폴 항목
- ▶ f: 사용자 정의 작업(외래 함수)
- ▶ v: 로컬 변수

예를 들어 ev:Napoleon은 이름이 Napoleon인 이벤트를 지정하고 dp:Napoleon은 이름이 Napoleon인 데이터폴 항목을 지정합니다.

명칭공간 접두사가 없는 식별자는 문자열 상수입니다.

EB GUIDE의 식별자에는 공백과 문장 부호를 비롯한 여러 문자가 포함됩니다. 그러므로 EB GUIDE 스크립트의 식별자를 따옴표로 묶어야 할 수 있습니다. 예를 들어 문자, 숫자 및 밑줄로만 구성되는 유효한 C 식별자와 같이 식별자에 특수 문자가 없는 경우에는 따옴표로 묶지 않아도 됩니다.



### 예 6.7.

#### EB GUIDE 스크립트의 식별자

```
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!";
```

### 6.15.3. 주석

EB GUIDE 스크립트에는 두 가지 종류의 주석, 즉 C 스타일 블록 주석과 C++ 스타일 줄 주석이 있습니다. 블록 주석은 중첩하면 안 됩니다.



예 6.8.

#### EB GUIDE 스크립트의 주석

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

"todo" 문자열이 포함된 모든 EB GUIDE 스크립트 주석의 경우 프로젝트 유효성을 검사할 때 EB GUIDE Studio에서 **문제** 구성 요소에 경고를 표시합니다. 이 기능을 사용하면 열려 있는 모든 작업에 표시를 추가하고 한눈에 확인 가능하도록 작업을 표시할 수 있습니다.

#### [참고]

#### 조건부 스크립트의 기본 주석



기본적으로 Conditional script 유형의 속성이나 데이터폴 항목에는 // todo: auto generated return value, please adapt 주석이 포함되어 있습니다. 경고가 표시되지 않도록 하려면 필요한 EB GUIDE 스크립트 코드를 입력한 후 주석에서 todo 문자열을 삭제하십시오.

### 6.15.4. 유형

EB GUIDE 스크립트는 강력한 정적 유형이 지정된 프로그래밍 언어입니다. 모든 식에는 적절하게 정의된 유형이 있습니다. 잘못된 유형을 입력하면 오류가 발생합니다.

EB GUIDE 스크립트에서 지원하는 유형은 다음과 같습니다.

- ▶ 정수
- ▶ 유니코드 문자열(문자열)
- ▶ 참조 계산이 포함된 개체
- ▶ 위에 나와 있는 유형 및 다음 항목에 대한 유형 정의
  - ▶ 색(32비트 RGBA 값에 해당하는 정수)
  - ▶ 부울
  - ▶ 데이터폴 항목, 보기, 상태 기계, 팝업 등 여러 모델 요소의 ID(모두 정수 유형)
- ▶ void(단위 유형이라고도 함). 이 유형에는 예를 들어 Haskell 등의 기능 프로그래밍에서와 같은 역할이 있습니다.
- ▶ 위젯 및 이벤트 참조. 이러한 참조는 레코드 유형이며, C 또는 Java에서와같이 dot 표기법을 사용하여 해당 필드에 액세스할 수 있습니다. 이러한 종류의 새 개체는 직접 만들 수는 없으며 해당하는 경우 자동으로 만들어집니다.

모든 유형 및 유형 정의는 서로 호환되지 않으며 유형 캐스팅도 없습니다. 이 기능을 사용하면 스크립트가 정상적으로 컴파일된 후 유형을 안전하게 유지할 수 있습니다.

## 6.15.5. 식

EB GUIDE 스크립트는 식을 기반으로 합니다. 모든 언어 구문은 식입니다. 여러 개의 작은 식을 연산자로 결합하여 큰 식을 만듭니다.

식을 평가할 때는 식을 값으로 바꿉니다.



### 예 6.9. 정수 값 평가

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

## 6.15.6. 상수 및 참조

기본 식은 정수, 색, 부울 및 문자열 상수와 모델 요소에 대한 참조입니다.

void 유형에도 값 상수가 있으며, 이 상수는 의미상으로는 동일하지만 형식은 다른 두 가지 방식으로 작성할 수 있습니다.

- ▶ 여는 중괄호와 닫는 중괄호({}) 사용
- ▶ unit 키워드 사용



### 예 6.10. 상수 사용법

```
"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
```

## 6.15.7. 산술 식과 논리 식

EB GUIDE 스크립트에서는 다음과 같은 산술 식을 지원합니다.

- ▶ 덧셈 (+), 뺄셈 (-), 곱셈 (\*), 나눗셈 (/), 모듈로 (%)는 정수 유형의 식에 적용할 수 있습니다.
- ▶ 논리 연산자 또는 (||), 및 (&&), 아님 (!)은 부울 유형의 식에 적용할 수 있습니다.
- ▶ 큼 (>), 작음 (<), 크거나 같음 (>=), 작거나 같음 (<=) 비교 연산자를 사용하여 정수와 문자열을 비교할 수 있습니다.
- ▶ 같음 연산자로 같음 (==) 및 같지 않음 (!=)을 사용하여 데이터 유형을 비교할 수 있습니다.

같음 연산자 (=Aa=)를 사용하여 대/소문자를 구분하지 않고 문자열을 비교할 수 있습니다.

### [참고]

#### 같음 연산자 사용 가능 여부



3D 그래픽, 글꼴, 이미지 등의 이벤트 및 리소스 데이터 유형은 같음 연산자 (==) 및 (!=)를 지원하지 않습니다.

- ▶ (+) 연산자를 사용하여 문자열을 연결할 수 있습니다.



### 예 6.11.

#### 산술 식과 논리 식

```
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
```

### 6.15.8. l 값과 r 값

EB GUIDE 스크립트에는 두 가지 종류의 식, 즉 l-values 및 r-values가 있습니다. l 값의 경우 주소를 포함하며 지정 왼쪽에 올 수 있습니다. r 값의 경우 주소를 포함하지 않으며 지정 왼쪽에 올 수 없습니다.

- ▶ l 값은 데이터풀 참조, 로컬 위젯 속성 및 로컬 변수입니다.
- ▶ r 값은 이벤트 매개변수 및 상수 식(예: 문자열 또는 정수 상수)입니다.

### 6.15.9. 로컬 변수

let 식에서는 로컬 변수를 사용합니다. 이 식은 변수 선언과 변수가 표시되는 in 식의 목록으로 구성됩니다. 변수는 l 값이므로 지정 왼쪽에 사용할 수 있습니다. 변수에는 네임스페이스 v:가 포함됩니다. let 식의 구문은 다음과 같습니다.

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

let 식의 유형 및 값은 in 식의 유형 및 값과 같습니다.

let 식은 중첩할 수 있으며 외부 let 식의 변수는 내부 식에도 표시됩니다.



#### 예 6.12. let 식의 사용법

```
// 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;
  }
}

```

## 6.15.10. while 루프

while EB GUIDE 스크립트의 루프 구문은 C 또는 Java의 구문과 비슷하며, 조건식과 수행식으로 구성됩니다. 해당 구문은 다음과 같습니다.

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

조건식이 false가 될 때까지 수행식을 반복적으로 평가합니다. condition expression은 boolean 유형이어야 하며 수행식은 void 유형이어야 합니다. while 식은 void 유형이며 지정의 왼쪽이나 오른쪽에 올 수 없습니다.



### 예 6.13. while 루프의 사용법

```

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

```

## 6.15.11. if-then-else

if-then-else EB GUIDE 스크립트의 는 C 및 Java의 3진 조건부 연산자 (?:)와 비슷하게 작동합니다.

if-then-else 식은 다음 하위 식으로 구성됩니다.

- ▶ 조건식
- ▶ then 식
- ▶ else 식

해당 구문은 다음과 같습니다.

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

if-then-else 는 다음과 같이 처리됩니다.

1. 먼저 조건식을 평가합니다. 조건식은 부울 유형이어야 합니다.
2. 조건이 참이면 then 식을 평가합니다.
3. 조건이 거짓이면 else 식을 평가합니다.

if-then-else 자체는 식입니다. 전체 식의 유형은 then 식과 else 식의 유형이며, 이 두 유형은 동일해야 합니다.

if-then-else 식의 값은 위의 규칙에 따라 then 식의 값이거나 else 식의 값입니다.

else 분기를 생략할 수 있는 특수한 if-then-else 형식이 있습니다. 이 특수 형식은 void 유형이며, 스크립트에서 값을 반환하는 데 사용할 수 없습니다.



#### 예 6.14. if-then-else의 사용법

```
// 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 {}
```

## 6.15.12. 외래 함수 호출

C에서 작성된 함수인 외래 함수를 사용하여 EB GUIDE 스크립트를 확장할 수 있습니다.

접두사 f:가 붙은 식별자가 외래 함수의 이름입니다. 외래 함수는 C에서와 마찬가지로 인수 목록과 반환 값을 포함합니다. 외래 함수 호출의 구문은 다음과 같습니다.

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



### 예 6.15. 외래 함수 호출

```
// 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);
```

## 6.15.13. 데이터풀 액세스



EB GUIDE 스크립트에서 작성한 스크립트는 데이터풀 항목을 읽고 쓸 수 있습니다. 네임스페이스 dp:가 접두사로 붙은 식별자를 데이터풀 항목 식이라고 합니다. 이 유형은 datapool item of type X이며, 여기서 X는 식이 참조하는 데이터풀 항목의 유형입니다.

X 유형의 데이터풀 항목이 지정 왼쪽에 오고 X 유형의 식이 지정 오른쪽에 오는 경우 데이터풀 항목의 값을 씁니다.

데이터풀 항목이 프로그램의 특정 위치에 있는데 지정의 왼쪽에는 없으면 데이터풀 항목의 값을 읽습니다.



#### 예 6.16. 데이터풀 값 지정

```
// 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 )
{
    // ...
}
```

다음 연산자를 데이터풀 항목에 적용할 수 있습니다.

- ▶ 데이터풀 항목에 참조 연산자(&)를 적용할 수 있습니다. 이 연산자는 데이터풀 항목의 값이 아닌 주소를 참조합니다. 참조 연산자는 외래 함수 호출에서 dp\_id 유형의 매개변수를 전달하는 데 사용됩니다.
- ▶ 리디렉션 링크 연산자(=>)는 데이터풀 항목의 링크 대상을 변경합니다. 이미 연결된 데이터풀 항목만 링크 소스가 될 수 있습니다.

## 6.15.14. 위젯 속성

위젯의 요소인 스크립트는 해당 위젯의 속성에 액세스할 수 있습니다. EB GUIDE 스크립트에서는 점 표기법을 사용하여 속성에 액세스할 수 있도록 v:this라는 변수를 만듭니다.

예를 들어 클릭 또는 버튼 누르기 등의 입력 반응으로 위젯 속성에 연결된 스크립트는 위젯의 요소가 됩니다.



#### 예 6.17.

### 위젯 속성 설정

```
// 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);
```

위젯의 요소인 스크립트는 위젯 트리 내 다른 위젯의 속성에도 액세스할 수 있습니다.

이동 연산자(->)를 사용하여 위젯 트리 내의 다른 위젯을 참조합니다. 해당 구문은 다음과 같습니다.

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

왼쪽의 식은 위젯을 참조해야 하고 오른쪽의 식은 문자열(하위 위젯의 이름)이어야 합니다. 상위 위젯으로 이동하려면 오른쪽의 ^ 기호를 사용합니다. 전체 이동 식은 위젯을 참조합니다.

위젯 트리를 이동하는 경우 런타임 성능에 영향을 줄 수 있습니다. 위젯은 여러 속성을 효율적으로 조작할 수 있도록 변수에 지정됩니다.



#### 예 6.18.

#### 위젯 속성에 액세스

```
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;
}
```

## 6.15.15. 목록

데이터풀 항목과 위젯 속성은 목록을 포함할 수 있습니다. 하위 스크립트 연산자([])는 목록 요소에 액세스합니다. 해당 구문은 다음과 같습니다.

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

첫 번째 식은 목록 유형으로, 두 번째 식은 정수 값으로 평가되어야 합니다. 목록이 list A 유형이면 전체 목록 하위 스크립트 식은 A 유형이어야 합니다.

목록 하위 스크립트 식이 지정 왼쪽에 오는 경우에는 참조되는 목록 요소의 값을 씁니다.

length 키워드는 목록의 요소 수를 반환합니다. 이 키워드를 목록 식 앞에 추가하는 경우 전체 식은 정수 유형이어야 합니다.



#### 예 6.19. 목록

```
// 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";
```

현재 EB GUIDE 스크립트에서는 요소를 목록에 추가하거나 목록에서 제거할 수 없습니다.

목록 끝을 벗어나 목록 요소에 액세스하려고 하면 스크립트 실행이 즉시 중지됩니다. 따라서 모든 목록은 범위 내에서 액세스해야 합니다.

## 6.15.16. 이벤트

EB GUIDE 스크립트에서는 이벤트 처리를 위한 다음 식을 제공합니다.

- ▶ fire 식은 이벤트를 전송합니다. 해당 구문은 다음과 같습니다.

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

이벤트는 매개변수를 포함할 수 있지만 반드시 포함할 필요는 없습니다. fire 식의 매개변수 목록은 실행되는 이벤트의 매개변수와 일치해야 합니다. 이벤트에 매개변수가 없으면 괄호를 비워 두어야 합니다.



#### 예 6.20. fire 식 사용

```
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 식은 지정된 연기 시간 이후 이벤트를 전송합니다. 해당 구문은 다음과 같습니다.

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



time 매개변수는 연기를 밀리초 단위로 지정하는 정수 값입니다.



**예 6.21.**  
**fire\_delayed 식 사용**

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

- ▶ cancel\_fire 식은 연기된 이벤트를 취소합니다. 해당 구문은 다음과 같습니다.

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

- ▶ match\_event 식은 이벤트에 의해 스크립트 실행이 트리거되었는지 여부를 확인합니다. 해당 구문은 다음과 같습니다.

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

match\_event 식의 유형은 in 식과 else 식의 유형이며, 이 두 유형은 동일해야 합니다.

else 분기를 생략할 수 있는 특수한 match\_event 식의 형식이 있습니다. 이 특수 형식은 void 유형이며, 스크립트에서 값을 반환하는 데 사용할 수 없습니다.



**예 6.22.**  
**match\_event 식 사용**

```
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 {}
```

매개변수가 포함된 이벤트에 의해 스크립트가 트리거된 경우에는 match\_event 식의 in 식에서 매개변수에 액세스할 수 있습니다. C 구조의 필드에 액세스할 때와 같이 점 표기법을 사용하여 매개변수를 읽습니다. else 식에서는 이벤트 매개변수를 사용할 수 없습니다.



**예 6.23.**

### 이벤트 매개변수

```
// 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;
}
```

## 6.15.17. 문자열 서식 지정

EB GUIDE 스크립트에서 문자열 서식을 지정할 때는 문자열에서 연결 연산자(+)와 데이터를 문자열로 변환하는 여러 함수를 함께 사용합니다. EB GUIDE 스크립트 표준 라이브러리에는 단순 정수에서 문자열 변환을 위한 `int2string` 함수가 포함되어 있습니다.



### 예 6.24. 문자열 서식 지정

```
// 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);
```

## 6.15.18. 표준 라이브러리

EB GUIDE 스크립트에는 외래 함수 집합으로 구성된 예를 들어 음과 같은 표준 라이브러리가 포함되어 있습니다.

- ▶ 문자열 서식 지정
- ▶ 언어 관리
- ▶ 추적
- ▶ 시간 및 날짜
- ▶ 난수 생성

자세한 내용은 [섹션 12.4.3, "EB GUIDE 스크립트 표준 라이브러리"](#) 항목을 참조하십시오.

## 6.16. 스크립팅된 값

스크립팅된 값은 위젯 속성이나 데이터폴 항목 값의 대체 표기법입니다. 위젯 또는 데이터폴 항목의 이러한 속성은 다른 모델 요소를 사용하여 자체 값을 평가하거나 이벤트 또는 속성 업데이트에 반응합니다. 스크립팅된 값은 EB GUIDE 스크립트 스크립팅 언어로 작성됩니다.

EB GUIDE의 속성을 스크립팅된 값으로 변환했다가 다시 일반 값으로 변환할 수 있습니다.

관련 지침은 [섹션 9.7. "속성을 스크립팅된 값으로 변환"](#) 항목을 참조하십시오.

EB GUIDE Studio에는 스크립팅된 값을 편집할 수 있는 EB GUIDE 스크립트 편집기가 포함되어 있으며, 이 편집기는 여러 범주로 구분됩니다.



그림 6.15. EB GUIDE Studio의 EB GUIDE 스크립트 편집기

- ▶ 스크립팅된 값 속성을 읽을 때는 **Read** 스크립트가 호출됩니다. 속성이 목록 유형이면 매개변수에는 목록 인덱스가 포함됩니다.

**Read** 스크립트의 반환 값은 속성의 현재 값을 나타냅니다.

- ▶ 스크립팅된 값 속성을 쓸 때는 **Write** 스크립트가 호출됩니다.

새 속성 값은 **Write** 스크립트의 매개변수입니다. 속성이 목록 유형이면 매개변수에는 목록 인덱스가 포함됩니다.

**Write** 스크립트의 반환 값은 속성의 변경 알림을 제어합니다.

- ▶ 값이 true이면 변경 알림이 트리거됩니다.



- ▶ 값이 false이면 변경 알림이 트리거되지 않습니다.
- ▶ **트리거** 목록에는 **On trigger** 스크립트의 실행을 트리거하는 위젯 속성, 데이터폴 항목 및 이벤트 목록이 포함됩니다.
- ▶ **On trigger** 스크립트는 초기화 시, 이벤트 트리거 이후 또는 속성 업데이트 이후에 호출됩니다.

**On trigger** 스크립트의 매개변수는 스크립트가 실행된 이유를 나타냅니다. **트리거** 목록의 트리거 중 하나 또는 초기화로 인해 스크립트가 실행될 수 있습니다.

**On trigger** 스크립트의 반환 값은 속성의 변경 알림을 제어합니다.

- ▶ 값이 true이면 변경 알림이 트리거됩니다.
- ▶ 값이 false이면 변경 알림이 트리거되지 않습니다.
- ▶ **Length** 스크립트는 목록 유형의 속성에만 사용할 수 있습니다.

**Length** 스크립트의 반환 값은 목록의 현재 길이를 나타냅니다.

## 6.17. 바로가기, 버튼 및 아이콘

### 6.17.1. 바로가기

다음 표에는 EB GUIDE Studio에서 제공되는 바로가기 및 해당 의미에 대한 설명이 나와 있습니다.

표 6.4. 바로가기

바로가기	설명
Ctrl+C	선택 항목 복사
Ctrl+F	검색 상자로 이동
Ctrl+S	저장
Ctrl+V	복사한 선택 항목 붙여 넣기
Ctrl+Y	다시 실행
Ctrl+Z	실행 취소
Alt+F4	활성 창 닫기
ShiftF1	EB GUIDE TF에 대한 사용자 문서 열기
F1	EB GUIDE Studio에 대한 사용자 문서 열기
ShiftF2	<b>데이터폴</b> 또는 <b>이벤트</b> 구성 요소 및 선택한 요소가 사용되는 EB GUIDE 스크립트와 같은 모든 위치에서 선택한 요소의 이름을 바꿉니다. 데이터 폴 항목 및 이벤트에 적용할 수 있습니다.

바로가기	설명
F2	선택한 요소 이름 바꾸기
F3	EB GUIDE 모델에서 선택된 요소의 모든 항목 찾기
F5	시뮬레이션 시작
F6	유효성
Delete 키	콘텐츠 영역 또는 구성 요소에서 선택한 요소 삭제
-	내비게이션 또는 개요 구성 요소에서 선택한 요소 축소
* 및 +	내비게이션 또는 개요 구성 요소에서 선택한 모델 요소 확장
위쪽/아래쪽/왼쪽/오른쪽 화살표 키	콘텐츠 영역에서 선택한 상태 또는 위젯을 한 픽셀 위/아래/왼쪽/오른쪽으로 이동

## 6.17.2. 명령줄 옵션

### 6.17.2.1. Studio.Console.exe의 명령줄 옵션

다음 테이블에는 Studio.Console.exe에 대해 EB GUIDE Studio에서 제공되는 명령줄 옵션 및 해당 의미에 대한 설명이 나와 있습니다. 정의되지 않은 명령줄 옵션은 무시됩니다.

명령줄의 일반 구문은 다음과 같습니다.

Studio.Console.exe <option> "project\_name.ebguide"

표 6.5. Studio.Console.exe의 명령줄 옵션

옵션	설명
-c <logfile dir>	EB GUIDE 모델의 유효성을 검증하고 로그 파일을 logfile dir로 지정된 디렉터리에 기록
-e <destination dir>	EB GUIDE 모델을 대상 디렉터리 destination dir에 내보내기 명령줄 옵션 -p을 사용하려면 아래 예제를 참조하십시오.
-h	이 도움말 메시지를 표시합니다.
-l <language file>	language file(.xliff)로 저장된 하나의 언어 파일을 EB GUIDE 모델로 가져오고 로그 파일 생성
-m	프로젝트의 마이그레이션 허용
-o	프로젝트 파일을 엽니다.



옵션	설명
-p <profile>	내보내는 중에 profile로 지정된 프로파일 사용



**예 6.25.**  
명령줄 옵션

명령줄 Studio.Console.exe -e "C:/temp/exported\_project" -p "target\_profile" -o "project\_name.-ebguide"는 프로파일 target\_profile을 사용하여 project\_name.ebguide를 지정된 대상 디렉터리 C:/temp/exported\_project에 내보냅니다.

지침은 다음을 참조하십시오.

- ▶ [섹션 10.4.1.2, "명령줄을 사용하여 EB GUIDE 모델 유효성 검사"](#)
- ▶ [섹션 10.5.2, "명령줄을 사용하여 EB GUIDE 모델 내보내기"](#)
- ▶ [섹션 10.8.2.2, "명령줄을 사용하여 언어별로 다른 텍스트 가져오기"](#)

### 6.17.2.2. Monitor.Console.exe의 명령줄 옵션

다음 테이블에는 Monitor.Console.exe에 대해 EB GUIDE Monitor에서 제공되는 명령줄 옵션 및 해당 의미에 대한 설명이 나와 있습니다. 정의되지 않은 명령줄 옵션은 무시됩니다.

명령줄의 일반 구문은 다음과 같습니다.

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

표 6.6. Monitor.Console.exe의 명령줄 옵션

옵션	설명
-c <host:port>	EB GUIDE 모델을 실행 중인 EB GUIDE GTF 프로세스에 연결합니다.
-h	이 도움말 메시지를 표시합니다.
-l <language>	EB GUIDE Monitor의 언어를 다음 중 하나로 설정합니다. 영어는 en으로, 일본어는 ja로, 한국어는 ko로, 중국어는 zh-cn으로 설정합니다.
-o	monitor.cfg 구성 파일을 엽니다.
-s	정의된 스크립트의 모든 메서드를 실행합니다.



**예 6.26.**

**명령줄 옵션**

명령줄 Monitor.Console.exe -l ko는 EB GUIDE Monitor의 언어를 한국어로 설정합니다.

EB GUIDE Monitor를 사용하는 방법에 대한 지침은 [섹션 10.9. “EB GUIDE Monitor로 작업”](#) 항목을 참조하십시오.

### 6.17.3. 버튼

다음 표에는 EB GUIDE Studio 및 EB GUIDE Monitor에서 사용되는 버튼 및 해당 의미에 대한 설명이 나와 있습니다.











표 6.7. EB GUIDE Studio의 버튼

버튼	설명
	실행 취소
	다시 실행
	저장
	프로젝트 유효성 검사
	시뮬레이션 시작
	시뮬레이션 중지
	프로젝트 센터 열기
	추가 편집자 열기
	콘텐츠 영역과 <b>내비게이션</b> 구성 요소 동기화
	이벤트, 데이터풀 항목 또는 상태 기계를 추가합니다.
	속성 관련 컨텍스트 메뉴 열기 아래 열거된 버튼 색상은 다음을 나타냅니다. <ul style="list-style-type: none"> <li> 로컬 속성입니다.</li> <li> 다른 속성에 연결된 속성입니다.</li> <li> 데이터풀 항목에 연결된 속성입니다.</li> <li> 속성 값과 템플릿 값이 같습니다.</li> </ul>
	이벤트 실행

### 6.17.4. 아이콘

다음 표에는 EB GUIDE Studio에서 사용되는 아이콘 및 해당 의미에 대한 설명이 나와 있습니다.

표 6.8. EB GUIDE Studio의 아이콘

아이콘	설명
	보기 템플릿의 종료 애니메이션을 나타냅니다.
	보기 템플릿의 진입 애니메이션을 나타냅니다.
	상태 기계 또는 상태의 진입 작업을 나타냅니다.
	상태 기계 또는 상태의 종료 작업을 나타냅니다.
	컨텍스트 메뉴를 열어서 진입 또는 종료 작업을 삭제합니다.
	동적 상태 기계 목록이 활성화됨을 나타냅니다.
	템플릿을 나타냅니다.
	전이를 나타냅니다.
	내부 전이를 나타냅니다.
	위젯 템플릿: 속성이 위젯 템플릿 인터페이스에 추가되었음을 나타냅니다.

### 6.18. 스킨

스킨을 사용하면 동일한 EB GUIDE 모델에 다른 데이터폴 값을 정의하여 다른 사용자 인터페이스를 정의할 수 있습니다. 이 방법으로 야간 및 주간 모드 스킨 등의 예로 동일한 HMI에 대해 다양한 모양을 정의할 수 있습니다.

런타임 동안 스킨 간을 전환하여 다른 데이터폴 값의 효과를 확인할 수 있습니다.

스킨 지원은 일반 데이터폴 값에만 사용할 수 있으며 스크립팅된 값이나 연결된 데이터폴 항목에는 사용할 수 없습니다.

**[참고]**



**사용할 수 있는 언어 지원 없음**

데이터폴 항목에 대한 스킨 지원을 정의한 경우 동일한 항목에 언어 지원을 추가할 수 없습니다.

관련 지침은 [섹션 8.5. "스킨 지원으로 작업"](#) 항목을 참조하십시오.

## 6.19. 상태 머신 및 상태

### [참고]



### 상태 머신 및 상태의 배경 색상 변경

EB GUIDE Studio에서 다음의 배경 색상을 변경할 수 있습니다.

- ▶ 메인 상태 머신
- ▶ 보기 상태
- ▶ 복합 상태

배경 색상을 변경하려면 **속성** 구성 요소의 Background color 드롭다운 목록 상자에서 색상을 선택합니다.

### 6.19.1. 상태 머신

상태 머신은 시스템의 동적 동작을 설명하는 명확한 유한 자동화 기능입니다. EB GUIDE에서 상태 머신은 계층적으로 순서가 지정된 임의의 수의 상태 및 상태 간 전이로 구성됩니다.

EB GUIDE에서는 다음과 같은 유형의 상태 머신을 만들 수 있습니다.

#### 6.19.1.1. 햅틱 상태 머신

햅틱 상태 머신을 통해 GUI를 지정할 수 있습니다.

#### 6.19.1.2. 논리적 상태 머신

논리적 상태 머신을 통해 GUI가 없는 일부 논리를 지정할 수 있습니다.

#### 6.19.1.3. 동적 상태 머신

동적 상태 머신은 다른 상태 머신과 병렬로 실행됩니다.

동적 상태 머신은 시스템 시작 시 자동으로 시작되지 않습니다. 동적 상태 머신의 시작과 종지는 다른 상태 머신에 의해 시작됩니다.

동적 상태 머신에는 다음의 두 가지 종류가 있습니다.

- ▶ 햅틱 동적 상태 머신

▶ 논리적 동적 상태 머신

관련 지침은 [섹션 11.1, "자습서: 동적 상태 머신 추가"](#) 항목을 참조하십시오.

## 6.19.2. 상태

EB GUIDE에서는 상태 개념을 사용합니다. 상태에 따라 상태 머신의 동작과 상태가 결정됩니다. 상태는 전이를 통해 연결됩니다. 상태 간의 연결인 전이는 원본 상태에서 대상 상태로의 상태 변경을 정의합니다.

상태에는 다음 속성이 포함되어 있습니다.

- ▶ 진입 작업
- ▶ 종료 작업
- ▶ 내부 전이

### 6.19.2.1. 복합 상태

복합 상태는 다른 상태를 내부에 하위 상태로 포함할 수 있습니다. 복합 상태 구조는 계층형이며, 포함 가능한 하위 상태의 수는 임의로 결정됩니다. 모든 유형의 상태를 복합 상태에 중첩할 수 있습니다.



그림 6.16. 복합 상태

내비게이션 구성 요소에서 상태 계층은 트리 구조로 표시됩니다.

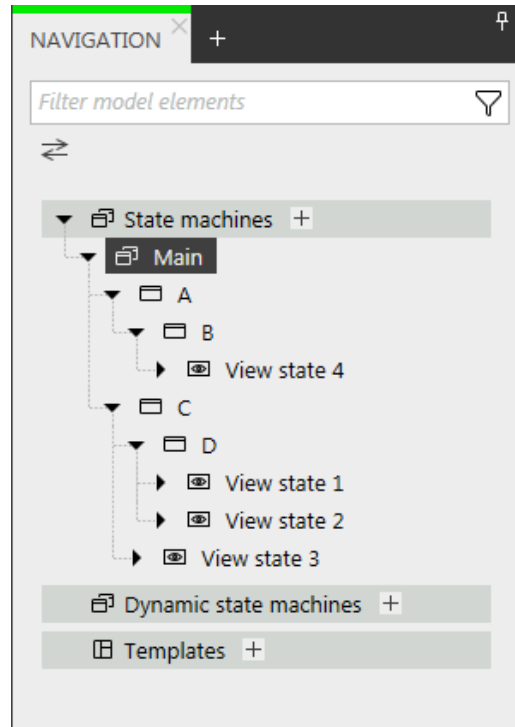


그림 6.17. 트리로 표시되는 상태 계층

복합 상태는 임의의 수의 들어오는 전이와 나가는 전이 및 내부 전이를 포함할 수 있습니다. 하위 상태는 상위 상태의 전이를 상속합니다.

### 6.19.2.2. 보기 상태

보기 상태는 보기를 포함합니다. 보기는 프로젝트 관련 HMI 화면을 나타냅니다. 해당하는 보기 상태가 활성화 상태인 동안 보기가 표시됩니다. 보기는 사용자와 시스템 간의 인터페이스인 위젯으로 구성됩니다.

### 6.19.2.3. 초기 상태

초기 상태는 상태 머신의 시작 포인트를 정의합니다. 초기 상태에는 첫 번째 상태를 가리키는 나가는 기본 전이가 있으며 들어오는 전이는 없습니다.

초기 상태는 복합 상태의 시작 포인트로 사용할 수도 있고, 다음과 같은 방식으로 복합 상태에 진입하기 위해 사용할 수도 있습니다.

- ▶ 복합 상태로의 전이 사용(초기 상태는 필수 항목임)
- ▶ 복합 상태 하위 상태로의 전이 사용

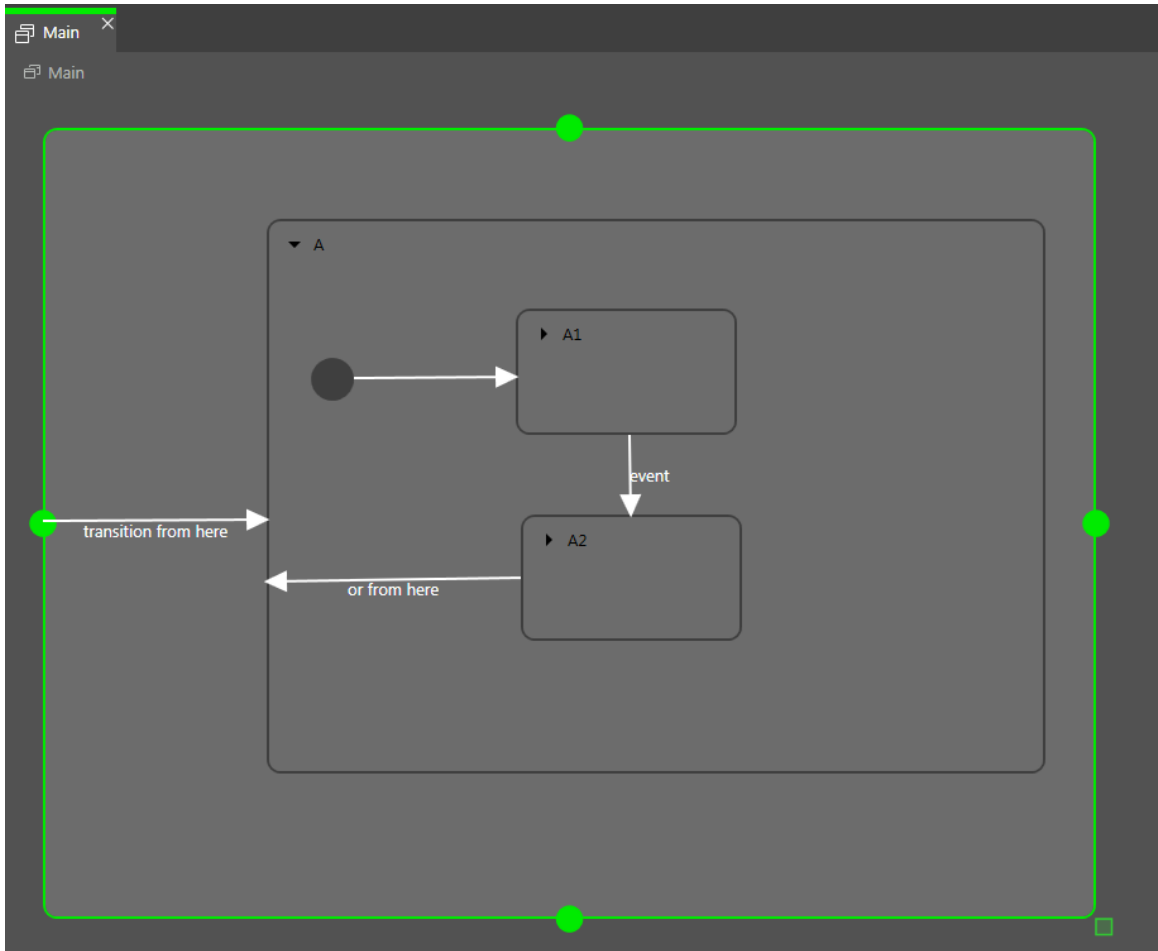


그림 6.18. 초기 상태의 예

#### 6.19.2.4. 최종 상태

최종 상태는 복합 상태를 종료하는 데 사용됩니다. 상태 머신의 최종 상태에 진입하면 상태 머신이 종료됩니다. 그러면 복합 상태 내의 이력 상태가 다시 설정됩니다. 최종 상태에는 나가는 전이가 없습니다.

복합 상태는 최종 상태를 하나만 포함할 수 있습니다. 최종 상태는 다음 작업을 통해 트리거됩니다.

- ▶ 하위 상태에서 복합 상태 외부로의 전이(이벤트 z를 통한 전이)
- ▶ 복합 상태에서 나가는 전이(이벤트 y를 통한 전이)
- ▶ 복합 상태 내 최종 상태로의 전이(이벤트 x를 통한 전이)

최종 상태를 포함하는 복합 상태에는 나가는 전이가 있어야 합니다.

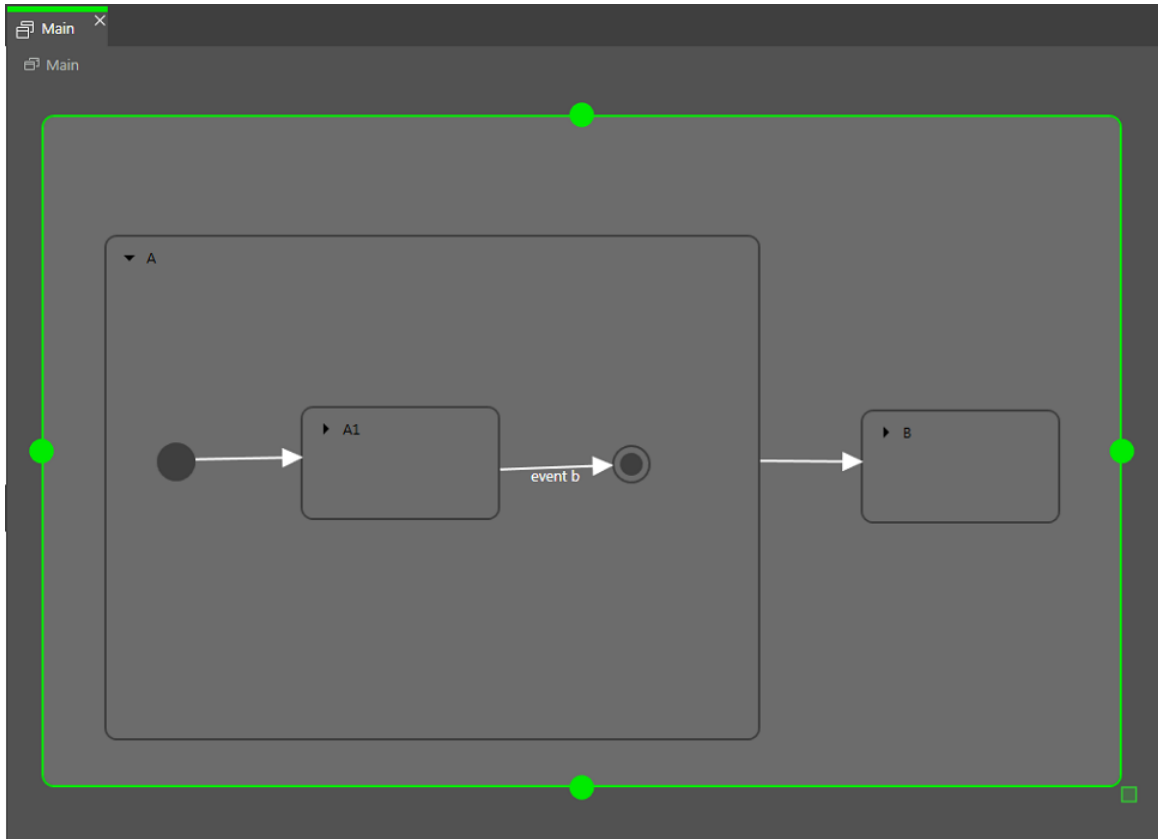


그림 6.19. 복합 상태 내의 최종 상태 사용법

### 6.19.2.5. 선택 상태

선택 상태는 동적 조건부 분기를 인식하며 조건에 따라 이벤트를 실행할 때 사용됩니다. 선택 상태는 원본 상태와 대상 상태 간의 연결입니다. 선택 상태에는 들어오는 전이와 나가는 전이가 여러 개 있을 수 있습니다. 모든 나가는 전이에는 조건이 지정되며, 해당 조건이 true로 평가되어야 나가는 전이가 실행됩니다. 나가는 전이 하나만 있는 else 전이입니다. 이 전이는 기타 모든 조건이 false로 평가되면 실행됩니다. else 전이는 필수 항목입니다.

여러 나가는 전이가 true일 수도 있으므로 나가는 전이를 평가할 순서를 정의해야 합니다.



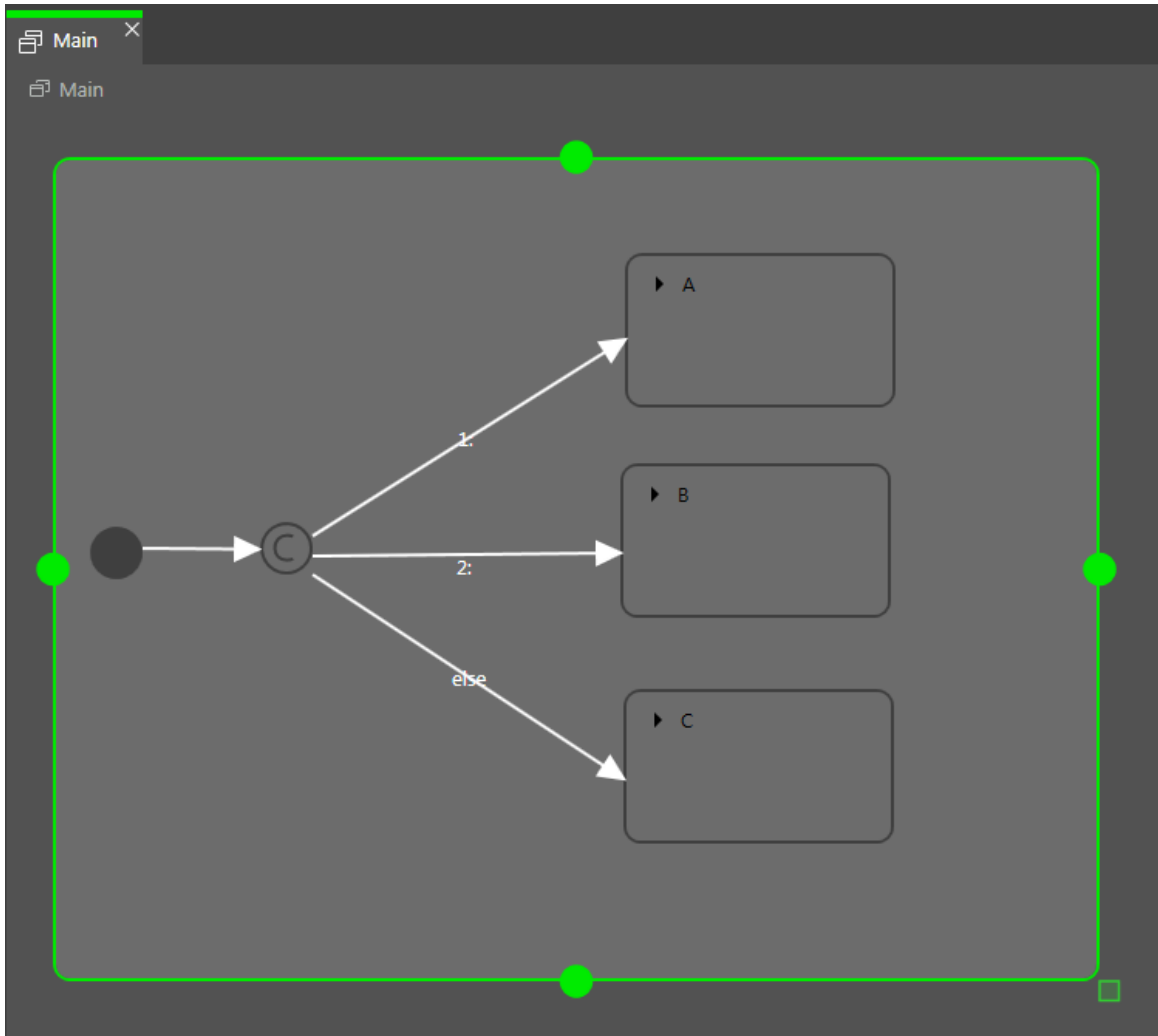


그림 6.20. 들어오는 전이와 나가는 전이가 포함된 선택 상태

### 6.19.2.6. 이력 상태

EB GUIDE에서는 두 가지 유형의 이력 상태가 지원됩니다.

- ▶ 얇은 이력 상태에는 가장 최근의 활성 하위 상태(복합 상태가 종료되기 직전에 활성이었던 하위 상태)가 저장됩니다.
- ▶ 깊은 이력 상태에는 종료되기 직전의 복합 상태와 이 상태의 전체 하위 계층이 저장됩니다.

이력 상태의 상위 상태에 처음으로 진입하면 마지막 활성 하위 상태가 복원됩니다.

얇은 이력 상태에는 복합 상태가 종료되기 전에 마지막으로 활성이었던 상태만 저장됩니다. 이 상태에 계층을 저장할 수는 없습니다.

얇은 이력 상태는 복합 상태 내에 기록된 마지막 활성 상태를 복원합니다. 이 상태에는 조건이 없는 나가는 기본 전이 하나가 포함되며, 들어오는 전이는 여러 개 포함될 수 있습니다.

복합 상태에 처음으로 진입할 때 얇은 이력 상태는 비어 있습니다. 비어 있는 얇은 이력 상태에 진입하면 얇은 이력 상태 기본 전이에 따라 다음 상태가 결정됩니다.



**예 6.27.**  
**얇은 이력 상태**

얇은 이력 상태는 다음과 같이 사용할 수 있습니다.

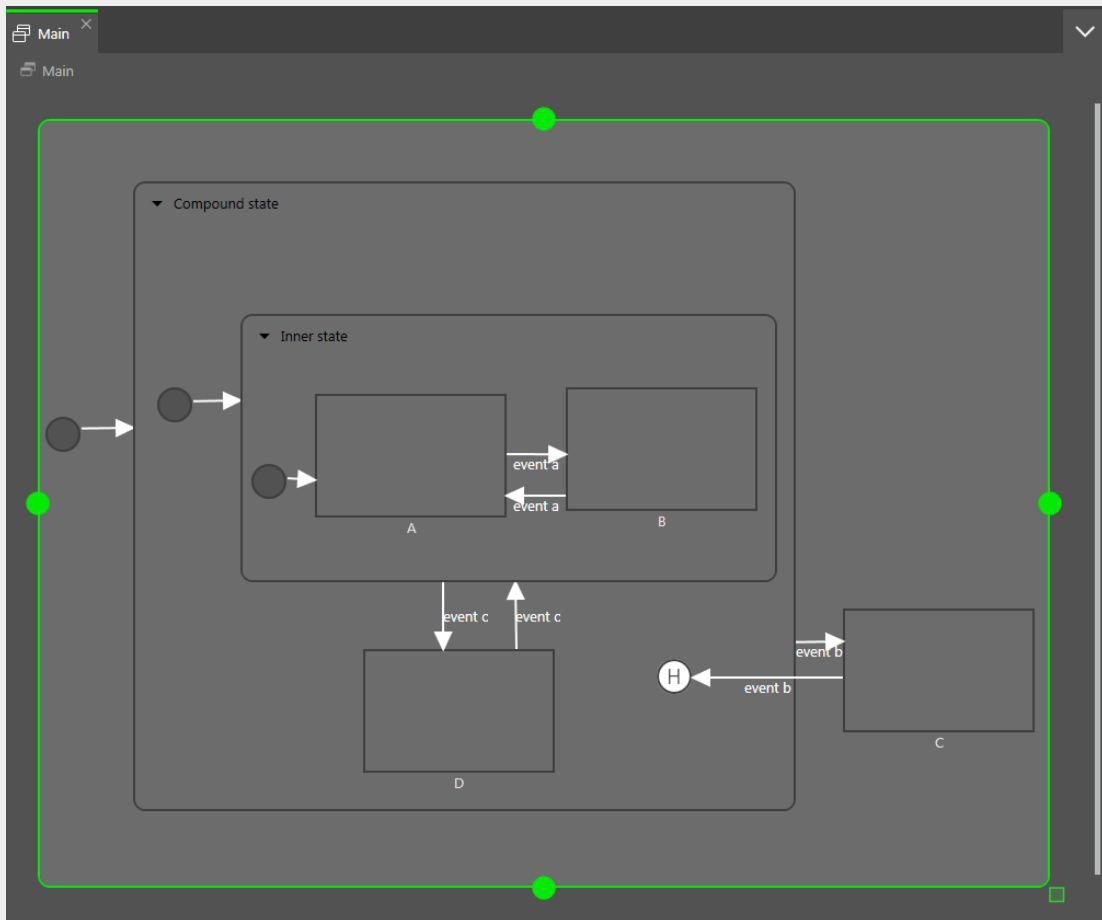


그림 6.21. 얇은 이력 상태

- ▶ 사례 1: 활성 상태가 D입니다.
  1. event b 가 발생하여 C 상태에 진입합니다.
  2. event b 가 다시 발생하여 얇은 이력 상태에 진입합니다.
  3. 얇은 이력 상태에서 상태 머신은 D 상태에 진입합니다. D 상태가 Compound State의 마지막 활성 상태였기 때문입니다.
- ▶ 사례 2: 활성 상태가 B입니다.

1. event b 가 발생하여 C 상태에 진입합니다.
2. event b 가 다시 발생하여 얕은 이력 상태에 진입합니다.
3. 얕은 이력 상태에서 상태 머신은 Inner state 상태에 진입합니다. 얕은 이력 상태는 마지막 활성 상태는 저장하지만 계층은 저장할 수 없기 때문입니다.
4. Inner state에 진입하면 상태는 A로 설정됩니다.

깊은 이력 상태에서는 계층 이력을 저장할 수 있습니다.



**예 6.28.**  
**깊은 이력 상태**

깊은 이력 상태는 다음과 같이 사용할 수 있습니다.

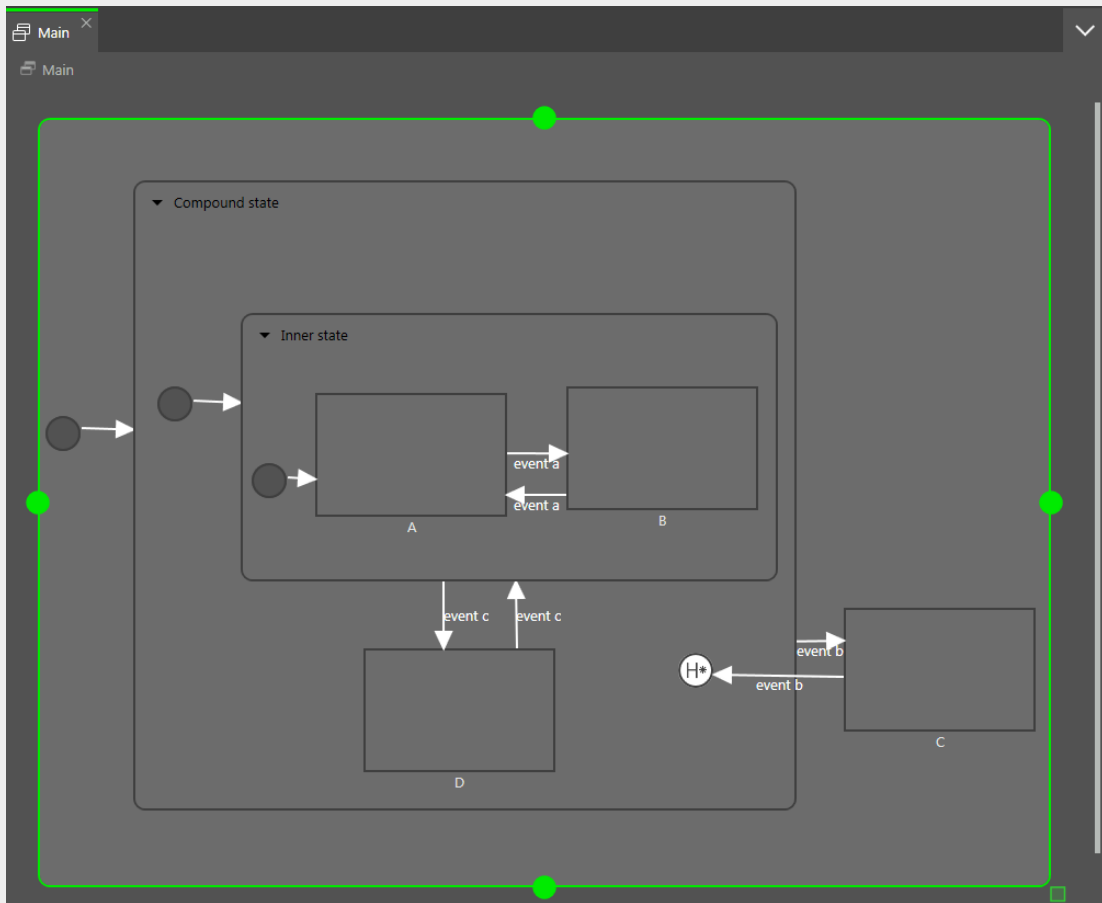


그림 6.22. 깊은 이력 상태

▶ 사례 1: 활성 상태가 D입니다.

1. event b 가 발생하여 C 상태에 진입합니다.
2. event b 가 다시 발생하여 깊은 이력 상태에 진입합니다.

3. 깊은 이력 상태에서 상태 머신은 D 상태에 진입합니다. D 상태가 Compound State의 마지막 활성 상태였기 때문입니다.

▶ 사례 2: 활성 상태가 B입니다.

1. event b가 발생하여 C 상태에 진입합니다.

2. event b가 다시 발생하여 깊은 이력 상태에 진입합니다.

3. 깊은 이력 상태에서 상태 머신은 B 상태에 진입합니다. 마지막 활성 상태가 B 상태였으며, 깊은 이력 상태는 상태 계층을 저장하기 때문입니다.

상태는 깊은 이력 상태 또는 얕은 이력 상태일 수 있습니다. 상위 상태와 하위 상태에 각기 다른 이력 상태를 적용할 수 있습니다.

### 6.19.3. 전이

전이는 원본 상태와 타깃 상태 간의 방향이 지정된 관계이며, 상태 기계의 상태를 변경합니다. 전이에는 다음 속성이 포함되어 있습니다.

▶ 전이를 실행하는 트리거

트리거는 데이터폴 항목 변경 또는 이벤트일 수 있습니다.

▶ 전이를 실행하려면 true로 평가되어야 하는 조건

▶ 전이와 함께 실행되는 작업

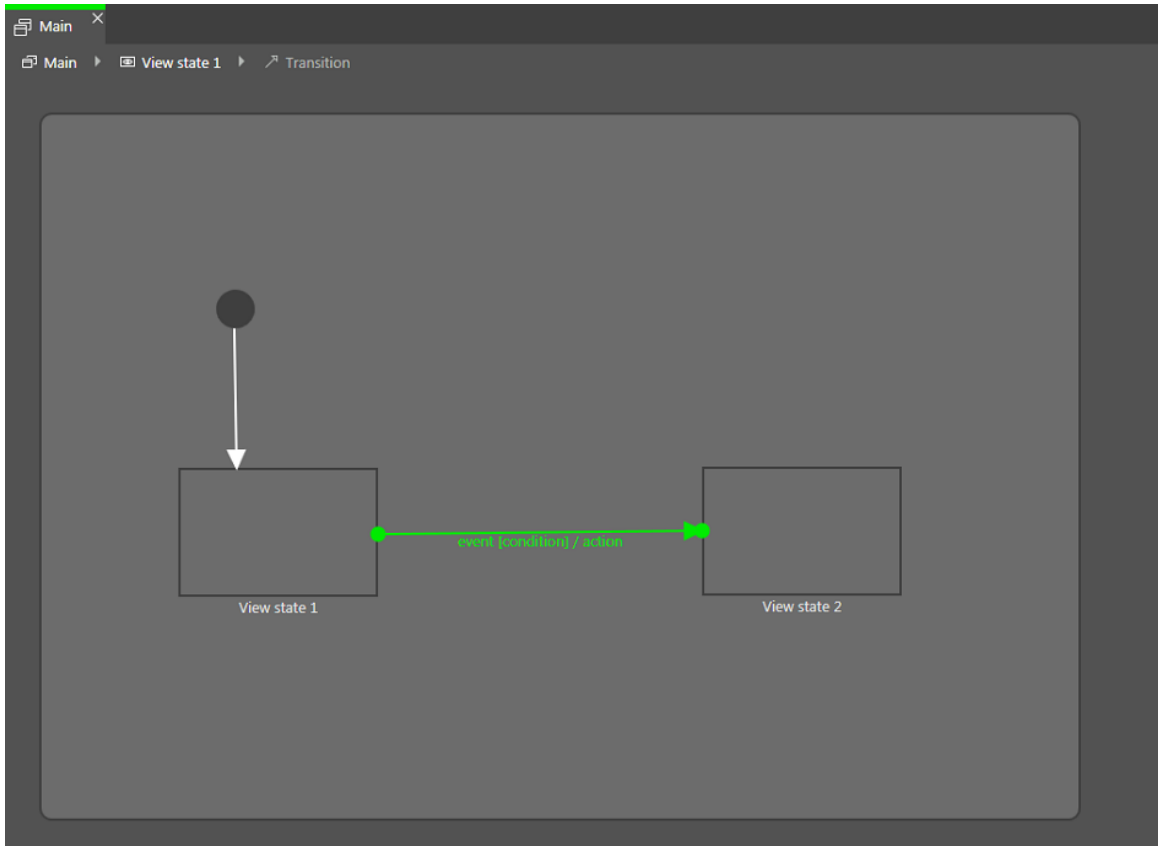


그림 6.23. 전이

**[참고]**

**전이의 명확성 유지**



조건이 다르더라도 같은 이벤트에 대해 특정 원본 상태의 전이를 둘 이상 사용할 수는 없습니다. 상태 기계가 조건에 따라 각기 다른 대상 상태로 전환되어야 하는 경우에는 선택 상태를 사용합니다.

상태는 상위 상태에서 모든 전이를 상속합니다. 여러 상태가 다른 상태와 같은 전이를 공유하는 경우 이러한 상태를 포함하는 복합 상태를 사용하여 전이를 번들로 묶으면 조건의 수를 줄일 수 있습니다.



**예 6.29.**

전이 상속

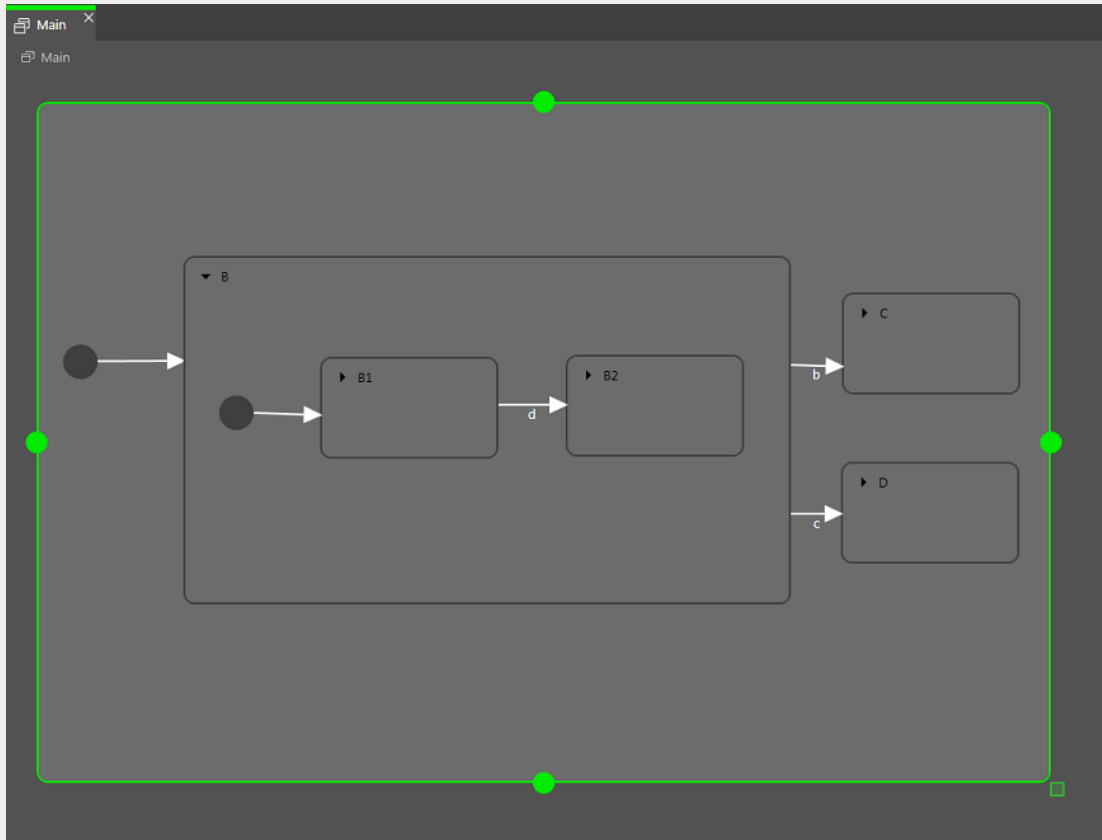


그림 6.24. 전이 상속

상태 기계가 State B1 상태일 때 b 이벤트가 발생하면 State C로의 전이가 실행됩니다. State B1 및 State B2 하위 상태는 State B 상태의 전이를 상속하기 때문입니다.

하위 상태의 내부 전이가 상위 상태의 외부 전이와 같은 이벤트를 사용하는 경우에는 전이 상속이 재정의됩니다.



예 6.30.

전이 재정의

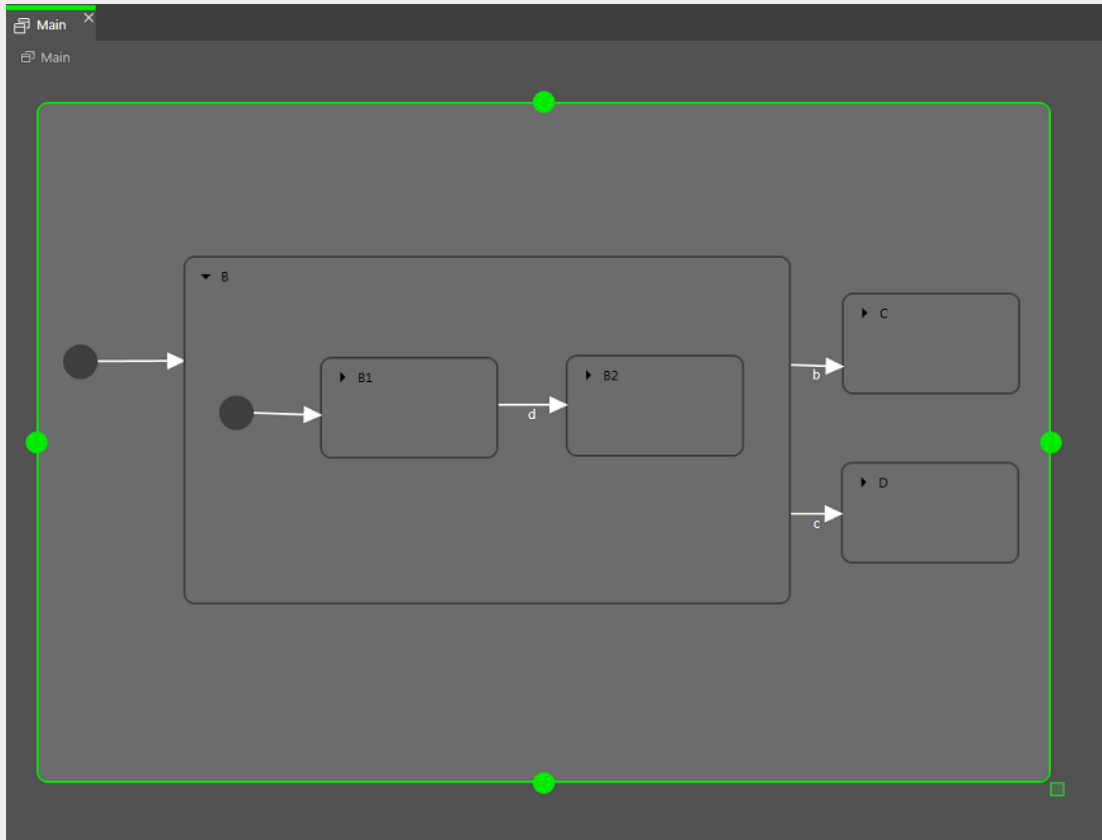


그림 6.25. 전이 재정의

상태 기계가 State B 상태일 때 d 이벤트가 발생하면 State C로의 전이가 실행됩니다.

상태 기계가 State B1 상태일 때 d 이벤트가 발생하면 State C가 아닌 State B2로의 전이가 실행됩니다. 두 전이는 이름이 같으므로 내부 전이가 외부 전이를 재정의합니다.

[참고]

실행 계층



상태 기계에서 같은 이벤트를 사용하는 전이 실행을 위한 계층은 항상 내부에서 외부 방향입니다. 즉, 내부 전이는 외부 전이보다 우선합니다.

다음과 같은 다양한 유형의 전이가 있습니다.

▶ 기본 전이

기본 전이는 이벤트 또는 데이터풀 항목 업데이트에 의해 트리거되지 않으며 자동으로 트리거됩니다. 기본 전이에는 조건은 없지만 작업은 있을 수 있습니다. 기본 전이는 초기 상태, 최종 상태, 선택 상태 및 이력 상태에서 사용됩니다.

▶ 선택 전이

선택 전이는 조건이 지정된 나가는 전이입니다. 이 전이의 원본 상태는 선택 상태입니다. 선택 전이는 조건 평가를 통해 트리거됩니다. 이러한 전이가 실행되면 작업이 수행됩니다. 조건이 true인 첫 번째 선택 전이가 실행됩니다.

▶ else 전이

else 전이는 선택 전이에 상응하는 필수 전이입니다. 모든 선택 상태에는 else 전이 하나가 포함되어야 합니다. 이 전이는 모든 선택 전이의 조건이 false로 평가되면 실행됩니다.

▶ 내부 전이

내부 전이는 타깃 상태가 없어 활성 상태를 변경하지 않는 전이입니다. 내부 전이는 현재 상태를 그대로 유지하면서 이벤트에 반응하는 데 사용됩니다. 이 전이는 조건을 포함할 수 있으며, 실행 시에는 작업이 수행됩니다.

특정 상태의 동일 이벤트에 대해 여러 내부 전이를 포함할 수 있습니다. 이 경우 실행 순서가 정의됩니다.

▶ 자체 전이

자체 전이는 원본 상태 및 타깃 상태와 동일한 상태의 전이입니다. 내부 전이와 달리 자체 전이는 상태를 종료했다가 해당 상태에 다시 진입하므로, 진입 작업과 종료 작업을 실행합니다.

## 6.19.4. 상태 기계 실행

상태 기계는 실행 시 항상 정확히 하나의 활성 상태로 설정됩니다. 상태 기계는 이벤트를 기반으로 합니다.

상태 기계 주기는 다음과 같습니다.

1. 상태 기계가 초기 상태에 진입하여 시작됩니다.
2. 상태 기계가 들어오는 이벤트를 대기합니다.
  - a. 내부 전이를 찾습니다.
    - i. 상태 기계가 현재 상태에서 시작된 다음, 현재 이벤트에 의해 트리거되며 조건이 true인 첫 번째 내부 전이를 검색합니다. 이러한 전이는 발견되면 실행됩니다.
    - ii. 전이가 발견되지 않으면 상태 기계가 상위 상태로 이동한 다음, 현재 이벤트에 의해 트리거되며 조건이 true인 첫 번째 내부 전이를 검색합니다.
    - iii. 전이가 발견되지 않으면 최상위 상태에 도달할 때까지 위의 단계를 반복합니다.
  - b. 내부 전이를 처리합니다.
 

내부 전이가 실행되면 내부 전이에 연결된 작업만 트리거됩니다. 이 경우 상태는 종료되지 않으며, 해당 상태에 다시 진입하게 됩니다.
  - c. 전이를 찾습니다.



- i. 상태 기계가 현재 상태에서 시작된 다음, 현재 이벤트에 의해 트리거되며 조건이 true인 전이를 검색합니다. 이러한 전이는 발견되면 실행됩니다.
- ii. 전이가 없으면 상위 상태로 이동하여 전이를 검색합니다.
- iii. 조건에 맞는 첫 번째 전이가 발견될 때까지 위의 단계를 반복합니다.
- d. 전이를 처리합니다.

전이가 실행되면 상태 기계의 상태가 변경됩니다. 따라서 원본 상태가 종료되고 대상 상태에 진입하게 됩니다.

해당하는 이벤트가 발생하고 조건이 true로 평가될 때만 전이가 실행됩니다.

이 경우 상태 계층의 여러 복합 상태 간에 전이가 종료되고 진입할 수 있습니다. 종료 연속과 진입 연속 간에 전이의 작업이 실행됩니다.

특정 상태에 진입할 때는 후속 전이가 필요할 수 있습니다. 예를 들어 복합 상태에 진입하려면 초기 상태의 전이가 후속 전이로 실행되어야 합니다. 이 경우 여러 후속 전이의 체인을 사용할 수 있습니다.

- 3. 상태 기계는 최종 상태에 도달하면 중지됩니다.

전이에서 상태 계층의 여러 상태가 설정되는 경우에는 종료 작업과 진입 작업이 연속으로 실행됩니다.



#### 예 6.31.

전이 실행

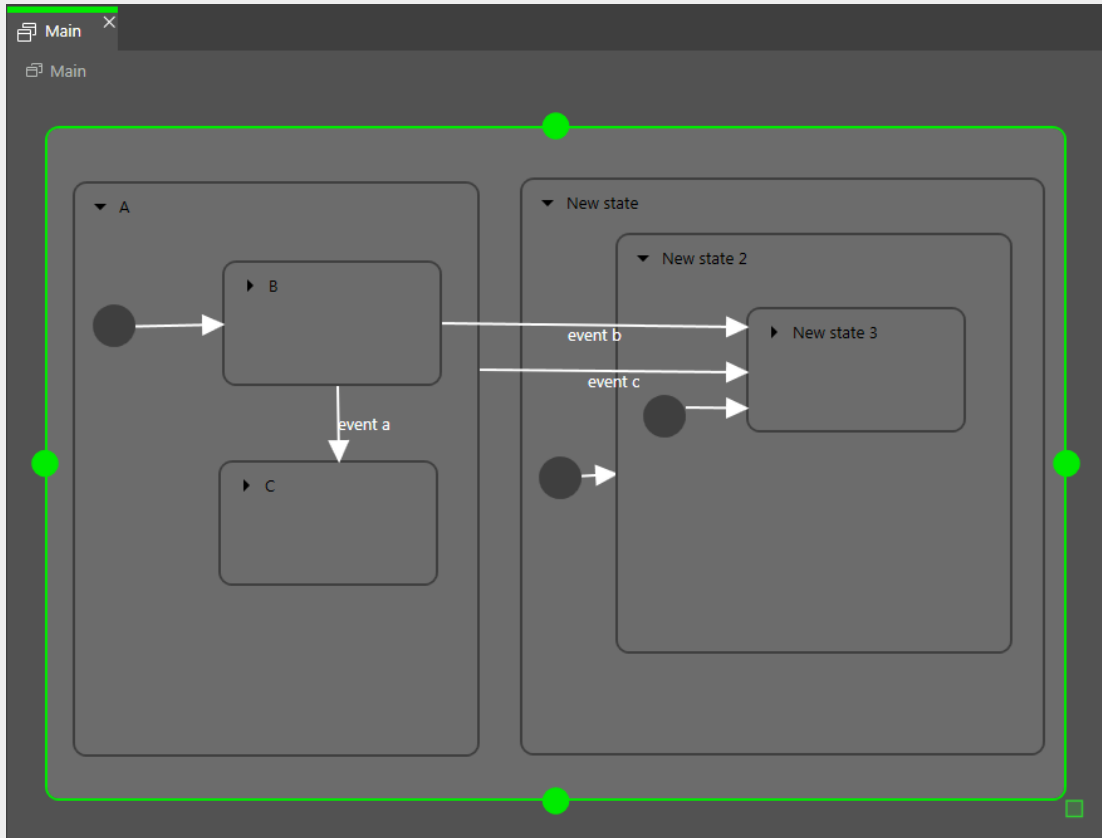


그림 6.26. 전이 실행

event a가 발생하는 경우의 결과는 다음과 같습니다.

1. B 상태가 종료됩니다.
2. C 상태에 진입합니다.

event b가 발생하는 경우의 결과는 다음과 같습니다.

1. B 상태가 종료됩니다.
2. A 상태가 종료됩니다.
3. New state 상태에 진입합니다.
4. New state 2 상태에 진입합니다.
5. New state 3 상태에 진입합니다.

event c가 발생하는 경우의 결과는 다음과 같습니다.

1. B 상태 또는 C 상태가 활성이면 B 상태 또는 C 상태가 종료됩니다.

2. A 상태가 종료됩니다.
3. New state 상태에 진입합니다.
4. New state 2 상태에 진입합니다.
5. New state 3 상태에 진입합니다.



**예 6.32.**  
전이 실행

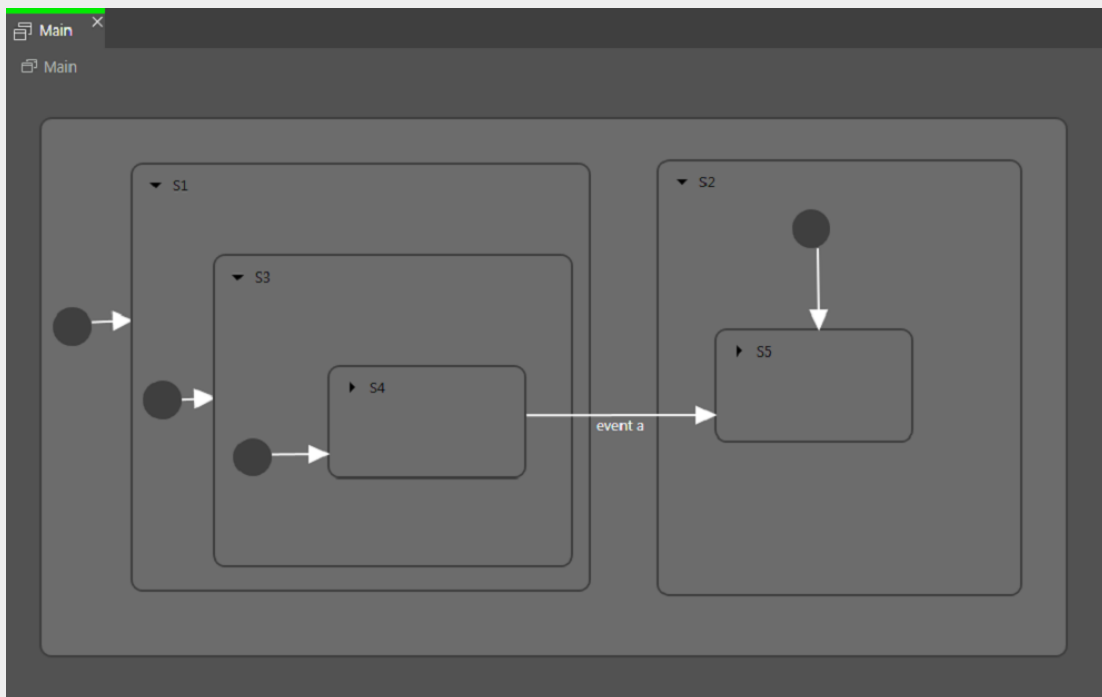


그림 6.27. 전이 실행

event a가 전이를 트리거하는 경우의 결과는 다음과 같습니다.

1. S4 상태가 종료됩니다.
2. S3 상태가 종료됩니다.
3. S1 상태가 종료됩니다.
4. S2 상태에 진입합니다.
5. S5 상태에 진입합니다.



**예 6.33.**

전이 실행

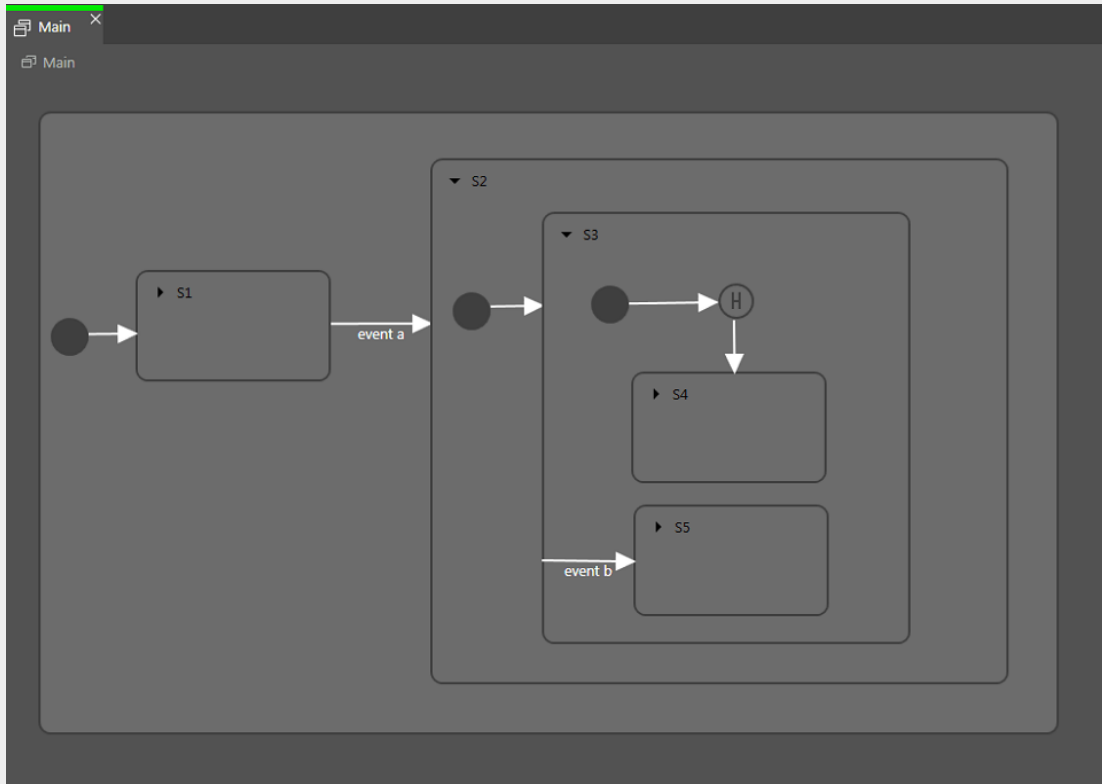


그림 6.28. 전이 실행

event a에 의해 전이가 트리거되면 다음 전이 시퀀스가 실행됩니다.

1. 상태 기계가 S2 상태로 설정됩니다.
2. 기본 전이가 실행되어 S3 상태가 설정됩니다.
3. 다음 기본 전이는 얇은 이력 상태로 진입합니다.
4. 얇은 이력 상태는 S3 상태의 마지막 활성 상태(S4 또는 S5 상태)를 복원합니다.

각 단계에서는 진입/종료/연속이 개별적으로 실행됩니다.

### 6.19.5. UML 표기법과 비교한 EB GUIDE 표기법

이 섹션에서는 EB GUIDE 표기법을 UML(Unified Modeling Language) 2.5 표기법과 비교하여 설명합니다.

### 6.19.5.1. 지원되는 요소

다음 표에는 EB GUIDE에서 지원하는 모든 UML 2.5 요소가 나와 있습니다. 일부 요소의 이름은 UML 2.5의 명명 규칙과 다르지만 이러한 요소가 제공하는 기능은 동일하게 유지됩니다.

EB GUIDE의 이름	UML 2.5의 이름
초기 상태	초기(의사 상태)
최종 상태	최종 상태
복합 상태	상태
선택 상태	선택(의사 상태)
깊은 이력 상태	DeepHistory(의사 상태)
얕은 이력 상태	ShallowHistory(의사 상태)
내부 전이	내부 전이
전이	외부/로컬 전이 <sup>a</sup>

<sup>a</sup>EB GUIDE에서는 외부 전이와 로컬 전이가 구분되지 않습니다.

### 6.19.5.2. 지원되지 않는 요소

EB GUIDE에서 지원되지 않는 UML 2.5 요소는 다음과 같습니다.

- ▶ 조인
- ▶ 포크
- ▶ 접합
- ▶ 진입 포인트
- ▶ 종료 포인트
- ▶ 종료

### 6.19.5.3. 차이점

UML 2.5 표기법의 일부 요소는 EB GUIDE에서 구현되지 않습니다. 그러나 EB GUIDE 개념을 통해 이러한 요소의 기능을 모델링할 수 있습니다.

UML 2.5의 개념	EB GUIDE의 해결 방법
병렬 상태	동적 상태 기계를 사용하여 개념을 구현합니다.
전이당 트리거 수	데이터폴 항목 또는 보기에서 EB GUIDE 스크립트를 사용하여 개념을 구현합니다.

UML 2.5의 개념	EB GUIDE의 해결 방법
전이의 시간 트리거	상태 기계, 데이터풀 항목, 전이 또는 보기에서 EB GUIDE 스크립트 (fire_delayed)를 사용하여 개념을 구현합니다.

## 6.20. 터치 입력

EB GUIDE에서는 두 가지 유형의 터치 입력, 즉 터치 제스처와 멀티 터치 입력이 지원됩니다.

각 터치 제스처는 EB GUIDE Studio에서 위젯 피처로 표시됩니다. 위젯 피처를 활성화하면 위젯에 속성 집합이 추가됩니다.

제스처는 두 가지 기본 유형으로 구분됩니다.

- ▶ 비경로 제스처
- ▶ 경로 제스처

### 6.20.1. 비경로 제스처

EB GUIDE에서는 다음과 같은 비경로 제스처를 구현합니다.

- ▶ 굿기
- ▶ 모으기
- ▶ 회전
- ▶ 누르기
- ▶ 길게 누르기

비경로 제스처에는 멀티 터치 및 단일 터치 제스처가 포함됩니다. 멀티 터치 제스처를 사용하려면 멀티 터치 입력을 지원하는 입력 기기가 필요합니다. 단일 터치 제스처는 지원되는 모든 입력 기기에서 작동합니다.

각 제스처는 서로 독립적으로 반응합니다. 여러 제스처가 활성화되어 있으면 모델러는 EB GUIDE 모델이 일관된 방식으로 작동하는지를 확인합니다.

### 6.20.2. 경로 제스처

경로 제스처는 터치 스크린에 손가락으로 그리거나 다른 입력 기기를 통해 입력하는 도형입니다. 위젯에서 위젯 피처가 활성화되어 있으면 사용자는 위젯에서 시작하는 도형을 입력할 수 있습니다. 도형이 구성 가능한 최소 경계 상자를 초과해야 경로 제스처 인식이 해당 도형을 고려합니다. 이 도형과 알려진 도형 집합의 일치 여부를 확인한 다음 일치하는 항목이 있으면 제스처가 인식됩니다.

관련 지침은 [섹션 11.3, "자습서: 경로 제스처 모델링"](#) 항목을 참조하십시오.

### 6.20.3. 입력 처리 및 제스처

제스처 인식은 일반 입력 처리와 병렬로 실행됩니다. 각 제스처는 제스처에 포함된 접촉부를 일반 입력 처리에서 제거하도록 요청할 수 있습니다. 제스처가 접촉부 제거를 요청하는 시기는 실제 제스처에 따라 다르며, 이 시기를 구성할 수 있는 제스처도 있습니다.

손가락을 사용하는 제스처의 경우에만 접촉부를 제거할 수 있습니다. 제거된 접촉부는 해당 접촉부에 대한 해제 이벤트가 수신될 때까지 일반 입력 처리에서 무시됩니다. 따라서 근접 기능이 지원되지 않는 터치 스크린의 경우에는 제거된 접촉부가 추가 터치 반응을 트리거하지 않습니다.

#### [작은 정보] 일반 입력 처리에서 접촉부 제거



창에 제스처용 위젯 피처와 버튼이 포함되어 있다고 가정해 보겠습니다. 이 창에서는 버튼을 사용하는 동안 제스처에 사용되는 접촉부를 해제하더라도 이 접촉부로 인해 버튼과 연관된 작업이 트리거되어서는 안 됩니다.

### 6.20.4. 멀티 터치 입력

호환 멀티 터치 입력 기기를 사용하는 경우 EB GUIDE에서는 멀티 터치 입력을 처리할 수 있습니다.

멀티 터치는 입력 기기 표면에서 접촉 포인트를 둘 이상 인식하고 추적할 수 있는 기능입니다. 일반적으로는 여러 손가락으로 터치 스크린을 터치하는 경우가 멀티 터치에 해당합니다.

#### ▶ 멀티 터치 이벤트 처리

멀티 터치 이벤트는 마우스 및 단일 터치 방식 터치 화면의 이벤트가 발송되는 것과 같은 방식으로 터치 이벤트용 메커니즘을 사용하여 발송됩니다. 멀티 터치 이벤트의 경우 각 접촉점이 모두 독립적으로 터치 반응을 트리거한다는 점만 다릅니다. 개별 접촉점을 식별할 수 있도록 각 터치 반응에는 fingerid 매개변수가 제공됩니다.

#### ▶ 손가락 ID

입력 기기에서 추적하는 각 접촉부에는 해당 접촉부를 식별하는 숫자가 지정됩니다. fingerid라는 이 식별자는 입력 기기별로 고유합니다. 그러나 나중에 특정 값이 더 이상 사용되지 않으면 다른 접촉부에 동일 값을 지정할 수 있습니다.

멀티 터치 입력이 활성화된 경우 최종 사용자가 수행할 수 있는 추가 터치 상호 작용 시퀀스에는 다음이 포함됩니다.

- ▶ 예를 들어 최종 사용자가 목록을 스크롤하면서 버튼을 누르는 등 인터페이스의 여러 요소와 동시에 상호 작용할 수 있습니다.

- ▶ 최종 사용자가 여러 손가락으로 위젯 하나를 터치할 수 있습니다.

이러한 방식을 확인할 수 있는 두 가지 일반적인 상황이 스크롤과 끌기입니다. fingerid를 통해 이러한 작업을 올바르게 처리할 수 있습니다. 필요한 동작에 따라 사용할 수 있는 방법은 다음과 같습니다.

- ▶ 위젯을 처음 누른 손가락만 스크롤 및/또는 끌기를 수행할 수 있도록 합니다.
- ▶ 항상 위젯을 마지막으로 터치한 손가락을 사용하여 스크롤 및/또는 끌기를 수행합니다. 위의 방식을 약간만 수정하면 이 방법을 쉽게 적용할 수 있습니다.

## 6.21. 위젯

위젯은 EB GUIDE 모델의 구성 요인 기본 그래픽 요소입니다.

위젯을 사용자 정의할 수 있습니다. 위젯의 속성을 편집하여 개별 요구사항에 맞게 위젯을 조정할 수 있습니다. 다음 요소는 터치하거나 이동될 경우의 예시 속성입니다.

- ▶ 크기
- ▶ 색
- ▶ 레이아웃
- ▶ 동작

위젯을 결합할 수 있습니다. 작은 구성 요소로 복합 구조를 만들 수 있습니다. 예를 들어 다음 요소로 구성된 버튼을 만들 수 있습니다.

- ▶ 타원
- ▶ 이미지
- ▶ 레이블
- ▶ 사각형

위젯을 중첩할 수 있습니다. 위젯 계층에서 이러한 중첩된 위젯을 하위 위젯이라고 하고 상위의 위젯을 상위 위젯이라고 합니다.

### 6.21.1. 보기

보기는 각 장면의 최상위 위젯입니다. 모델링 중에 다음 요소가 보기에 배치됩니다.

- ▶ 기본 위젯
- ▶ 3D 위젯
- ▶ 애니메이션



▶ 위젯 템플릿

각 보기는 정확히 하나의 보기 상태와 연결됩니다. 보기에는 반드시 보기 상태가 있어야 합니다.

**[참고]**

**보기의 크기 변경**




EB GUIDE Studio에서 클로즈업 보기를 확인하거나 더 자세하게 보려면 보기의 크기를 늘리거나 줄일 수 있습니다. 확대 및 축소하려면 슬라이더를 사용하거나 보기 하단의 텍스트 상자를 클릭합니다. 기본 확대 수준은 100%입니다. 또는 Ctrl+로 확대, Ctrl-으로 축소, Ctrl0으로 확대 수준을 100%로 다시 설정할 수 있습니다.

**[참고]**

**마스터 이미지에 요소 정렬**



EB GUIDE Studio에서 기본 위젯 및 3D 위젯 등의 요소를 고르게 정렬할 수 있도록 보기에 마스터 이미지를 추가할 수 있습니다. 보기에 마스터 이미지를 추가하려면 보기 하단에서  를 클릭하십시오. 마스터 이미지를 숨기려면 확인란을 선택하거나 선택 취소합니다. 보기를 닫으면 마스터 이미지를 다시 추가해야 합니다.

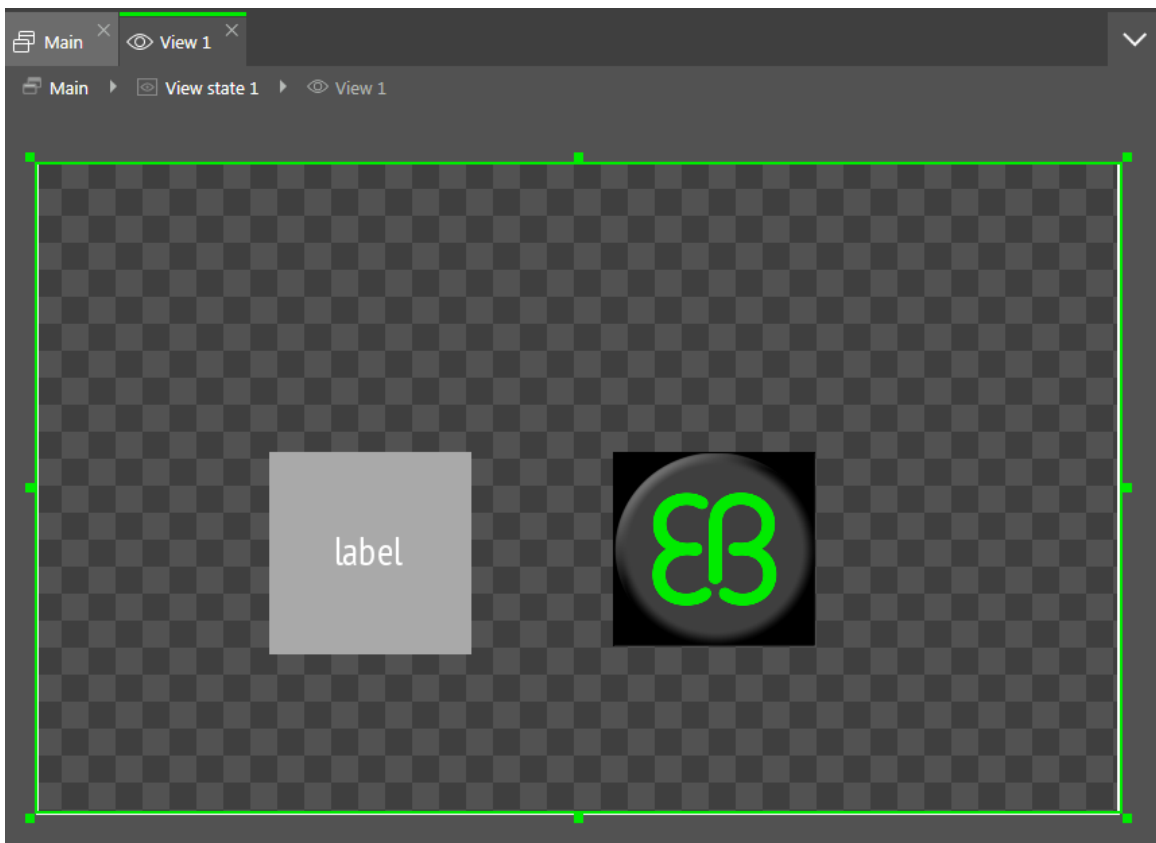


그림 6.29. 사각형, 레이블 및 이미지를 포함하는 보기

## 6.21.2. 위젯 범주

도구 상자에서 위젯은 범주별로 그룹화됩니다. 사용 가능한 범주는 다음과 같습니다.

▶ 기본 위젯

다음 요소가 기본 위젯입니다.

- ▶ 알파 마스크
- ▶ 애니메이션
- ▶ 컨테이너
- ▶ 타원
- ▶ 이미지
- ▶ 인스턴스화기
- ▶ 레이블
- ▶ 사각형

▶ 3D 위젯

3D 그래픽을 표시하려면 **3D 위젯** 범주에 포함된 위젯을 사용하십시오. **3D 위젯**은 다음 요소입니다.

- ▶ 주변 조명
- ▶ 카메라
- ▶ 방향 조명
- ▶ 이미지 기반 조명
- ▶ 재료
- ▶ 메시
- ▶ PBR GGX 재료
- ▶ PBR Phong 재료
- ▶ 포인트 조명
- ▶ 장면 그래프
- ▶ 장면 그래프 노드
- ▶ 스포트 조명

**[참고]**



**지원되는 렌더러**

3D 그래픽을 표시하려면 OpenGL ES 버전 2.0 이상 또는 DirectX 11 렌더러가 필요합니다. 그래픽 드라이버가 렌더러의 버전과 호환되는지 확인합니다.

▶ 위젯 템플릿

**템플릿** 범주에는 위젯 템플릿이 포함되어 있습니다. 이 범주는 위젯 템플릿을 정의해야 표시됩니다.

▶ 사용자 지정 위젯

**사용자 지정 위젯** 범주에는 사용자 지정된 위젯이 포함되어 있으므로 사용자 지정된 위젯을 프로젝트에 추가하는 경우에만 표시됩니다. 자세한 내용은 웹 사이트 <https://www.elektrobit.com/ebguide/learn/resources/>를 참조하십시오.

관련 지침은 [섹션 8.1. "위젯으로 작업"](#) 항목을 참조하십시오.

### 6.21.3. 위젯 속성

위젯은 해당 모양과 동작을 지정하는 속성 집합에 의해 정의됩니다. 현재 포커스된 위젯의 속성이 표시되는 속성 구성 요소에서 속성을 편집할 수 있습니다.

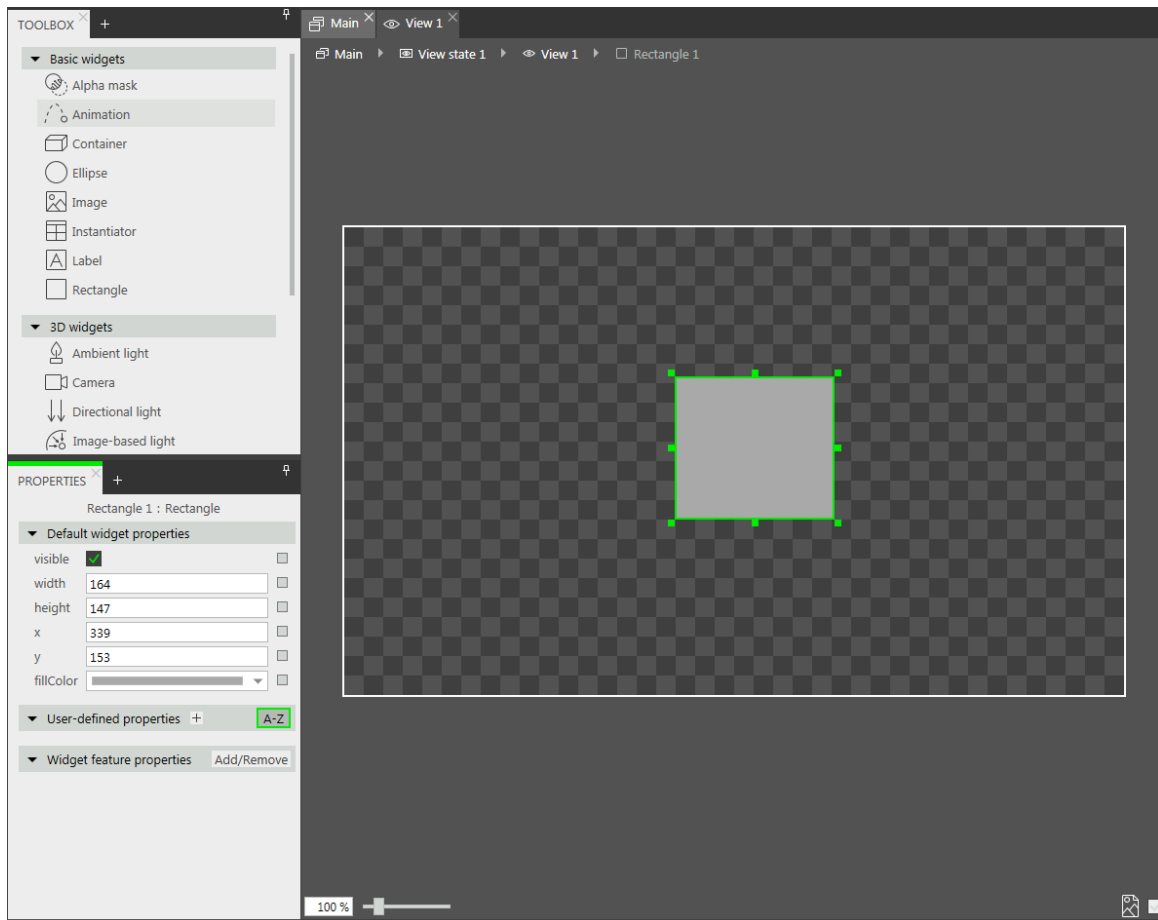


그림 6.30. 사각형 및 해당 속성

위젯 속성에는 다음과 같은 세 가지 유형이 있습니다.

- ▶ 기본 위젯 속성은 각 위젯 인스턴스와 함께 만들어집니다. 모든 위젯의 기본 속성 목록은 [섹션 12.12. "위젯"](#) 항목을 참조하십시오.

- ▶ 사용자 정의 위젯 속성은 기본 속성과 함께 모델러에서 만들어집니다.
- ▶ 모델러가 위젯에 위젯 피처를 추가하면 EB GUIDE Studio에서 위젯 피처 속성을 만듭니다. 위젯 피처 속성은 범주별로 그룹화됩니다. 위젯 피처는 위젯 표시 및 동작과 관련된 기능을 추가로 제공합니다.



**예 6.34.**

**터치함 위젯 피처**

**터치함** 위젯 피처는 위젯이 터치에 반응하는지 여부 및 반응 방법을 정의합니다. 이 위젯 피처는 4개 속성을 추가합니다. Boolean 속성 `touchable`에 따라 위젯이 터치 입력에 반응하는지 여부가 결정됩니다. Boolean 속성 `touched`는 현재 위젯을 터치하고 있는 경우 EB GUIDE에서 런타임 중에 설정하는 속성입니다. 그리고 두 가지 정수 속성(`touchPolicy` 및 `touchBehavior`)에 따라 위젯이 터치 입력에 반응하는 방법이 결정됩니다.

### 6.21.4. 위젯 템플릿

위젯 템플릿을 사용하면 EB GUIDE 모델에서 여러 번 사용할 수 있는 사용자 지정된 위젯을 정의할 수 있습니다. 기존 위젯을 기준으로 템플릿을 정의하거나 기존 템플릿에서 새 템플릿을 파생시킬 수 있습니다. 새 템플릿을 만든 다음 요구사항에 따라 템플릿을 수정(예: 속성 또는 위젯 피처 추가)할 수 있습니다. 따라서 위젯 템플릿을 사용하면 복합 위젯의 라이브러리를 구축할 수 있습니다.

위젯 템플릿에는 템플릿 인터페이스가 있습니다. 템플릿 인터페이스는 위젯 인스턴스에 표시되고 액세스 가능한 템플릿 속성을 포함합니다. 따라서 위젯 인스턴스는 템플릿 인터페이스의 속성을 상속합니다. 상속되는 속성을 템플릿 속성이라고 합니다. 템플릿 속성은 ■ 버튼으로 표시됩니다.

템플릿 속성의 값을 변경하면 속성이 로컬 속성으로 바뀝니다. 로컬 속성은 ■ 버튼으로 표시됩니다.



**예 6.35.**

**위젯 템플릿의 속성과 해당 인스턴스의 관계**

Square 위젯 템플릿을 EB GUIDE 모델에 추가합니다. Square에 `color` 속성을 포함합니다. `color`가 템플릿 인터페이스에 추가되었습니다. `color` 값을 `red`로 지정합니다.

Square 위젯 템플릿 인스턴스를 보기에 추가합니다. 인스턴스 이름을 `BlueSquare`로 지정합니다.

- ▶ `BlueSquare` 는 값이 `red`인 `color`를 상속합니다.
- ▶ Square 템플릿에서 `color` 값을 `green`으로 변경합니다.  
=> `BlueSquare`에서도 `color` 값이 `green`으로 변경됩니다.
- ▶ `BlueSquare`에서 `color` 값을 `blue`로 변경합니다.

Square 템플릿에서 `color` 값을 `yellow`으로 변경합니다.

=> `BlueSquare`에서 `color` 값은 `blue`로 유지됩니다.

관련 지침은 [섹션 8.7. "위젯 다시 사용"](#) 항목을 참조하십시오.

## 6.21.5. 위젯 피처

위젯 피처를 이용해 위젯 및 위젯 템플릿의 기능을 확장할 수 있습니다. 위젯 피처에는 미리 정의된 위젯 속성이 있습니다. 위젯 피처는 범주별로 그룹화됩니다.

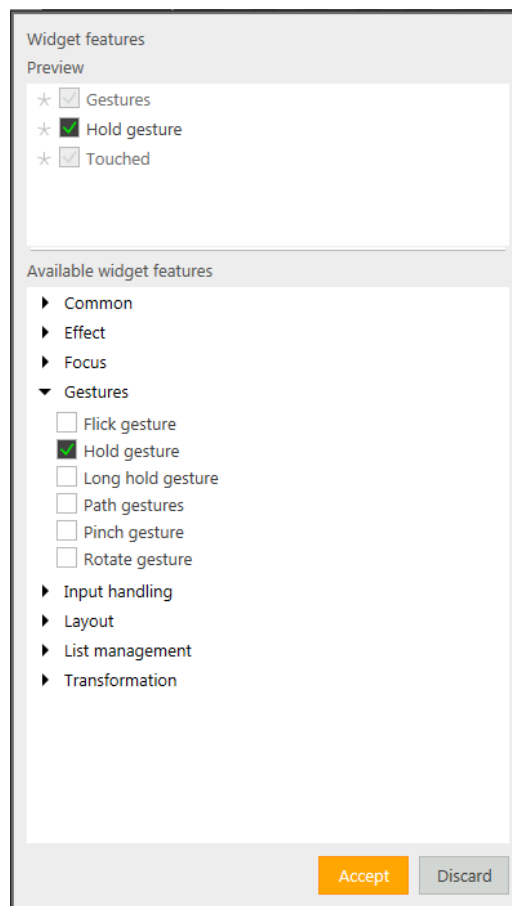


그림 6.31. 위젯 피처

위젯 템플릿에 위젯 피처를 추가하면 생성된 모든 위젯 템플릿 인스턴스가 추가된 위젯 피처를 상속합니다. 위젯 템플릿 인스턴스 또는 템플릿에서 작성된 템플릿에는 위젯 피처를 추가할 수 없습니다.

위젯 피처 사용에 대한 제한 사항은 다음과 같습니다.

- ▶ 위젯 피처에는 상속 계층 구조가 없습니다.
- ▶ 위젯당 두 번 이상 위젯 피처를 추가할 수 없습니다.

- ▶ 일부 위젯 피처는 상호의존적입니다. 즉, 하나의 위젯 피처를 추가하려면 다른 피처를 추가해야 합니다. 그렇지 않으면 위젯 피처가 서로 제외될 수 있습니다.
- ▶ 위젯 피처가 특정 유형의 위젯을 제한할 수 있습니다.
- ▶ 런타임 중에 위젯 피처를 활성화 또는 비활성화할 수 없습니다.

기본적으로 모든 위젯 피처는 비활성화되어 있습니다. 특정 위젯 피처가 필요하다면 위젯에 추가해야 합니다.

관련 지침은 [섹션 8.3. "위젯 피처로 위젯 확장"](#) 항목을 참조하십시오. 전체 위젯 피처 목록은 [섹션 12.13. "위젯 피처"](#) 항목을 참조하십시오.

### 6.21.5.1. 포커스 위젯 피처 범주

EB GUIDE Studio에서는 **포커스** 위젯 피처를 사용하여 위젯의 포커스 관리를 모델링합니다. **자동 포커스** 및 **사용자 정의 포커스**

다음 두 가지 포커스 방향을 사용할 수 있습니다.

1. 순방향: 다음 포커스 가능한 위젯이 포커스됩니다.
2. 역방향: 이전의 포커스 가능한 위젯이 포커스됩니다.

**자동 포커스** 및 **사용자 정의 포커스** 위젯 피처는 순방향에 대해 포커스가 처리되는 방법의 구성을 제공합니다. 역방향의 경우 포커스 순서는 동일하지만 반대 방향으로만 사용됩니다.

**포커스** 위젯 피처에는 다음과 같은 특징이 있습니다.

#### 자동 포커스

이 정책에서 포커스 가능한 위젯의 맨 위 줄부터 왼쪽에서 오른쪽으로 포커스가 분산됩니다. 순서는 위젯 트리의 구조를 통해 정의됩니다.

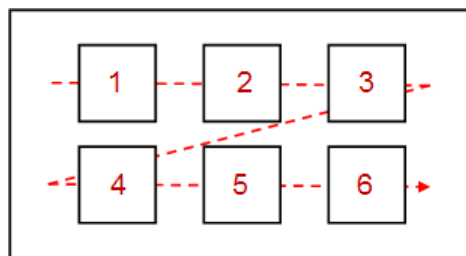


그림 6.32. 자동 포커스 위젯 피처의 정책

포커스 가능한 하위 위젯은 건너뛴 수 없습니다. 표시되지 않는 위젯, focused 속성을 사용할 수 없는 위젯 및 **포커스된** 위젯 피처가 없는 위젯은 유효한 포커스 가능한 위젯으로 인식되지 않습니다. 따라서 현재 포커스된 위젯이 결정되면 위 위젯은 건너뛰게 됩니다.

#### 사용자 정의 포커스

보기의 복잡성으로 인해 자동 포커스 정책을 통한 포커스 시퀀싱이 상당히 어려울 수도 있습니다. 이 경우 사용자 정의 포커스 순서를 결정하는 것이 좋습니다.

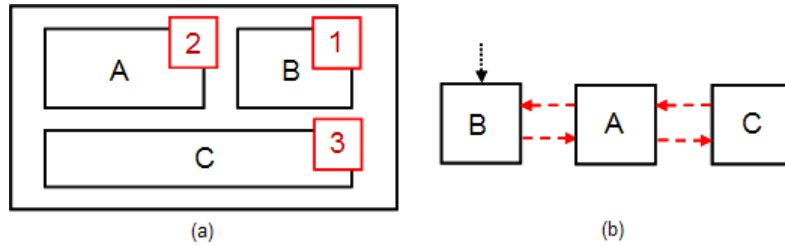


그림 6.33. 사용자 정의 포커스 위젯 피처의 정책

그림 6.33. “[사용자 정의 포커스 위젯 피처의 정책](#)”에서 (a)는 보기를 나타내고 (b)는 포커스 순서를 나타냅니다. 포커스 변경이 처리되는 순서는 위젯 트리 구조와 다를 수 있습니다.

위젯 계층 내의 위젯이 포커스 가능하다고 표시되면 포커스 계층의 일부입니다. 이 포커스 계층은 포커스 가능한 위젯, 포커스 정책, **자동 포커스** 위젯 피처 또는 계층 내에서 포커스가 처리되는 방법을 정의하는 **사용자 정의 포커스** 위젯 피처로 구성됩니다. 포커스 계층을 중첩할 수 있습니다.

### 6.21.5.2. 목록 관리 위젯 피처 범주

**줄 인덱스** 및 **템플릿 인덱스** 위젯 피처를 사용하면 이미지, 노래 제목과 같은 데이터를 인스턴스화기의 동적으로 생성된 줄 템플릿에 연결할 수 있습니다.

#### 줄 인덱스

**줄 인덱스** 위젯 피처는 인스턴스화기 위젯의 줄 템플릿을 사용자 지정하는 데 사용됩니다. **줄 인덱스** 위젯 피처는 목록 또는 테이블의 각 줄의 고유한 위치를 지정합니다.



#### 예 6.36. 줄 인덱스 위젯 피처

목록을 모델링하려면 목록의 각 항목에 목록 속성의 항목을 반영하는 특정 값이 있어야 합니다. 목록의 특정 항목에 액세스하려면 줄 템플릿의 인스턴스가 인스턴스화기의 하위 항목이 무엇인지 알아야 합니다. **줄 인덱스** 위젯 피처는 `lineIndex` 속성을 추가합니다. 인스턴스화기가 줄 템플릿의 인스턴스를 생성하는 동안 `lineIndex`를 다음 값으로 채웁니다. 인덱스는 첫 번째 인스턴스의 경우 0으로 시작합니다. 인스턴스화기에 두 가지 요소가 있는 경우 두 번째 요소는 `lineIndex` 값 1을 받습니다.

관련 지침은 [섹션 11.4, “자습서: 동적 콘텐츠로 목록 만들기”](#) 항목을 참조하십시오.

#### 템플릿 인덱스

**템플릿 인덱스** 위젯 피처를 사용하면 복잡한 데이터 추상화가 가능합니다. 매우 복잡한 목록이나 표의 경우 항목 또는 일련의 항목을 시각화하기 위해 하나 이상의 데이터 목록이 필요합니다. 예를 들어, 이미지와 텍스트 콘텐츠가 혼합된 테이블의 경우 이미지 목록과 문자열 목록이 필요합니다. 이러한 복잡한 경우를 다루기 위해 **템플릿 인덱스** 위젯 피처는 `lineTemplateIndex` 속성을 제공합니다.



#### 예 6.37.

**템플릿 인덱스 위젯 피쳐**

lineMapping 속성을 0으로 설정하고 numItems 속성을 5로 설정한 인스턴스화기를 사용하여 목록을 모델링하면 lineTemplateIndex는 0이1|1|2가 됩니다.



## 7. HMI 동작 모델링

### [참고]



#### 기본 창 레이아웃

이 사용자 설명서의 모든 지침 및 스크린샷이 기본 창 레이아웃을 사용합니다. 지침을 따르는 경우 EB GUIDE Studio 창을 기본 레이아웃으로 설정하는 것이 좋습니다. **레이아웃 > 기본 레이아웃으로 다시 설정**을 선택하여 설정할 수 있습니다.

## 7.1. 상태 머신 모델링

### 7.1.1. 상태 머신 추가



#### 상태 머신 추가

##### 단계 1

내비게이션 구성 요소에서 **상태 머신**으로 이동한 다음 **+**를 클릭합니다.

메뉴가 확장됩니다.

##### 단계 2

상태 머신의 유형을 선택합니다.

선택한 유형의 새 상태 머신이 추가됩니다.

##### 단계 3

상태 머신의 이름을 바꿉니다.

### 7.1.2. 동적 상태 머신 추가

동적 상태 머신은 다른 상태 머신과 병렬로 실행되며, 런타임에 시작(푸시)하고 중지(팝)할 수 있습니다.



#### 동적 상태 머신 추가

예를 들어 동적 상태 머신을 사용하여 일반 화면에 겹쳐지는 오류 메시지를 표시하는 등의 작업을 수행할 수 있습니다.

선수학습:

- 상태 머신, 보기 상태 또는 복합 상태가 EB GUIDE 모델에 추가되어 있어야 합니다.

#### 단계 1

**내비게이션** 구성 요소에서 **동적 상태 머신**으로 이동한 다음 **+**를 클릭합니다.

메뉴가 확장됩니다.

#### 단계 2

동적 상태 머신의 유형을 선택합니다.

선택한 유형의 새 동적 상태 머신이 추가됩니다.

#### 단계 3

**내비게이션** 구성 요소에서 동적 상태 머신과 병렬로 실행할 상태 머신, 보기 상태 또는 복합 상태를 클릭합니다.

#### 단계 4

**속성** 구성 요소에서 Dynamic state machine list 확인란을 선택합니다.

이러한 단계를 완료한 후 동적 상태 머신과 관련된 EB GUIDE 스크립트 함수를 사용합니다.

자세한 내용은 [섹션 11.1, "자습서: 동적 상태 머신 추가"](#) 항목을 참조하십시오.

## 7.1.3. 상태 머신에 대해 진입 작업 정의



### 상태 머신에 대해 진입 작업 정의

#### 단계 1

상태 머신을 선택합니다.

#### 단계 2

**속성** 구성 요소에서 **진입 작업** 속성으로 이동한 다음 **추가**를 클릭합니다.

#### 단계 3

EB GUIDE 스크립트를 사용하여 작업을 입력합니다.

배경 정보는 [섹션 6.15, "스크립팅 언어 EB GUIDE 스크립트"](#) 항목을 참조하십시오.

#### 단계 4

**수락**을 클릭합니다.

상태 머신에 대해 진입 작업을 정의했습니다.

## 7.1.4. 상태 머신에 대해 종료 작업 정의



### 상태 머신에 대해 종료 작업 정의

#### 단계 1

상태 머신을 선택합니다.

#### 단계 2

속성 구성 요소에서 Exit action 속성으로 이동한 다음 **추가**를 클릭합니다.

#### 단계 3

EB GUIDE 스크립트를 사용하여 작업을 입력합니다.

배경 정보는 [섹션 6.15. "스크립팅 언어 EB GUIDE 스크립트"](#) 항목을 참조하십시오.

#### 단계 4

**수락**을 클릭합니다.

상태 머신에 대해 종료 작업을 정의했습니다.

## 7.1.5. 상태 머신 삭제



### 상태 머신 삭제

#### 단계 1

**내비게이션** 구성 요소에서 상태 머신을 오른쪽 마우스 클릭합니다.

#### 단계 2

컨텍스트 메뉴에서 **삭제**를 클릭합니다.

상태 머신이 삭제됩니다.

## 7.2. 상태 모델링

### 7.2.1. 상태 추가



### 상태 추가

선수학습:

- 콘텐츠 영역에 상태 머신이 표시되어 있어야 합니다.

단계 1

상태를 **도구 상자**에서 상태 머신으로 끕니다.

상태가 상태 머신에 추가됩니다.

**[참고]**

**초기 상태, 최종 상태 및 이력 상태가 고유합니다.**



초기 상태, 최종 상태 및 이력 상태는 복합 상태당 한 번씩만 삽입할 수 있습니다.

**[작은 정보]**

**상태 복사 및 찾기**



컨텍스트 메뉴 또는 Ctrl+C 및 Ctrl+V를 사용하여 기존 상태를 복사하고 붙여 넣을 수도 있습니다.

EB GUIDE 모델 내의 특정 상태를 찾으려면 검색 상자에 상태 이름을 입력하거나 Ctrl+F를 사용합니다. 한 상태로 건너 뛰려면 결과 목록에서 더블 클릭합니다.

## 7.2.2. 복합 상태에 상태 추가



### 복합 상태에 상태 추가

상태 계층을 만들려면 다른 상태에 대한 하위 항목으로 상태를 만듭니다. 이렇게 하려면 복합 상태에 상태를 추가합니다.

선수학습:

- 콘텐츠 영역에 상태 머신이 표시되어 있어야 합니다.
- 상태 머신에 복합 상태가 포함되어 있어야 합니다.

단계 1

**내비게이션** 구성 요소에서 복합 상태를 더블 클릭합니다.

복합 상태가 콘텐츠 영역으로 확장됩니다.

단계 2

상태를 **도구 상자**에서 복합 상태로 끕니다.

상태가 복합 상태에 하위 상태로 추가됩니다.

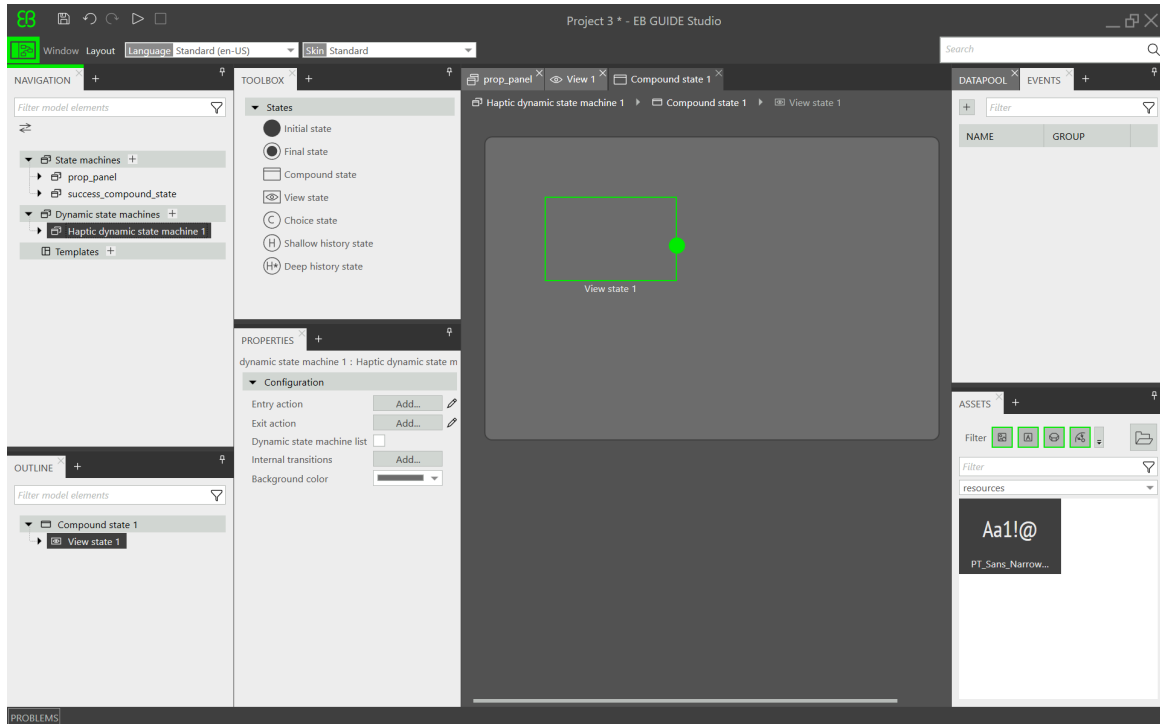


그림 7.1. 보기 상태가 중첩되어 있는 복합 상태

### 7.2.3. 선택 상태 추가



#### 선택 상태 추가

선수학습:

- 콘텐츠 영역에 상태 머신이 표시되어 있어야 합니다.
- 상태 머신에 상태가 2개 이상 포함되어 있어야 합니다.

#### 단계 1

선택 상태를 도구 상자에서 상태 머신으로 끕니다.

#### 단계 2

선택 상태에서 나가는 전이를 추가합니다.

#### 단계 3

나가는 전이에 조건을 추가합니다. 자세한 내용은 [섹션 7.3.4. "전이에 조건 추가"](#) 항목을 참조하십시오.

조건에 우선순위 1이 지정됩니다. 상태 머신이 선택 상태에 진입하면 우선순위가 1인 조건을 먼저 평가합니다.

#### 단계 4

선택 전이를 더 추가하려면 위의 두 단계를 반복합니다.

새 선택 전이에는 먼저 만든 전이보다 낮은 우선순위가 지정됩니다.

단계 5

선택 상태에서 나가는 전이를 추가합니다.

단계 6

**내비게이션** 구성 요소에서 전이를 오른쪽 마우스 클릭합니다. 컨텍스트 메뉴에서 **else로 변환**을 클릭합니다.

else 전이가 추가되었습니다. else 전이는 나가는 선택 전이에 지정된 모든 조건이 false로 평가되면 실행됩니다.

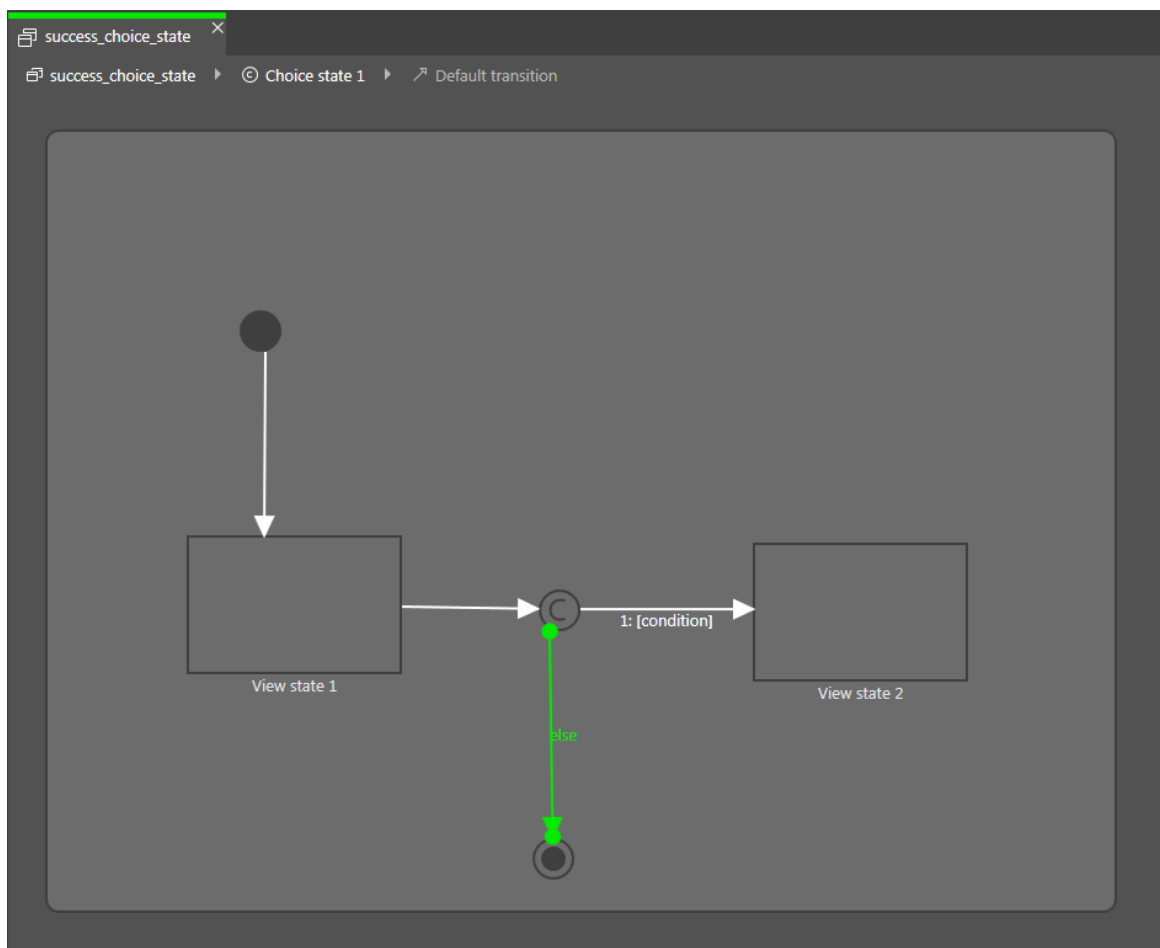


그림 7.2. 선택 전이가 포함된 선택 상태

## 7.2.4. 상태에 대해 진입 작업 정의



### 상태에 대해 진입 작업 정의

보기 상태 및 복합 상태에 대해 진입 작업을 정의할 수 있습니다. 해당 상태에 진입할 때마다 진입 작업이 실행됩니다.

선수학습:

- 상태 머신에 보기 상태 또는 복합 상태가 포함되어 있어야 합니다.

#### 단계 1

상태를 선택합니다.

#### 단계 2

**속성** 구성 요소에서 Entry action 속성으로 이동한 다음 **추가**를 클릭합니다.

#### 단계 3

EB GUIDE 스크립트를 사용하여 작업을 입력합니다.

배경 정보는 [섹션 6.15. "스크립팅 언어 EB GUIDE 스크립트"](#) 항목을 참조하십시오.

#### 단계 4

**수락**을 클릭합니다.

## 7.2.5. 상태에 대해 종료 작업 정의



### 상태에 대해 종료 작업 정의

보기 상태 및 복합 상태에 대해 종료 작업을 정의할 수 있습니다. 해당 상태가 종료될 때마다 종료 작업이 실행됩니다.

선수학습:

- 상태 머신에 보기 상태 또는 복합 상태가 포함되어 있어야 합니다.

#### 단계 1

상태를 선택합니다.

#### 단계 2

**속성** 구성 요소에서 Exit action 속성으로 이동한 다음 **추가**를 클릭합니다.

#### 단계 3

EB GUIDE 스크립트를 사용하여 작업을 입력합니다.

배경 정보는 [섹션 6.15. "스크립팅 언어 EB GUIDE 스크립트"](#) 항목을 참조하십시오.

단계 4  
수락을 클릭합니다.

## 7.2.6. 상태 머신에서 모델 요소 삭제



### 상태 머신에서 모델 요소 삭제

선수학습:

- 상태 머신에 모델 요소가 하나 이상 포함되어 있어야 합니다.

단계 1  
내비게이션 구성 요소에서 모델 요소를 오른쪽 마우스 클릭합니다.

단계 2  
컨텍스트 메뉴에서 **삭제**를 클릭합니다.

모델 요소가 삭제됩니다.

## 7.3. 전이를 통해 상태 연결

### 7.3.1. 두 상태 간에 전이 추가



### 두 상태 간에 전이 추가

전이를 사용하면 원본 상태를 타겟 상태에 연결할 수 있습니다.

선수학습:

- 콘텐츠 영역에 상태 머신이 표시되어 있어야 합니다.
- 상태 머신에 상태가 2개 이상 포함되어 있어야 합니다.

단계 1  
전이의 원본 상태로 사용할 상태를 선택합니다.

단계 2  
녹색 끌기 포인트를 클릭한 다음 마우스 버튼을 누른 상태로 유지합니다.



단계 3  
마우스를 타깃 상태로 끕니다.

단계 4  
타깃 상태가 녹색으로 강조 표시되면 마우스 버튼을 놓습니다.

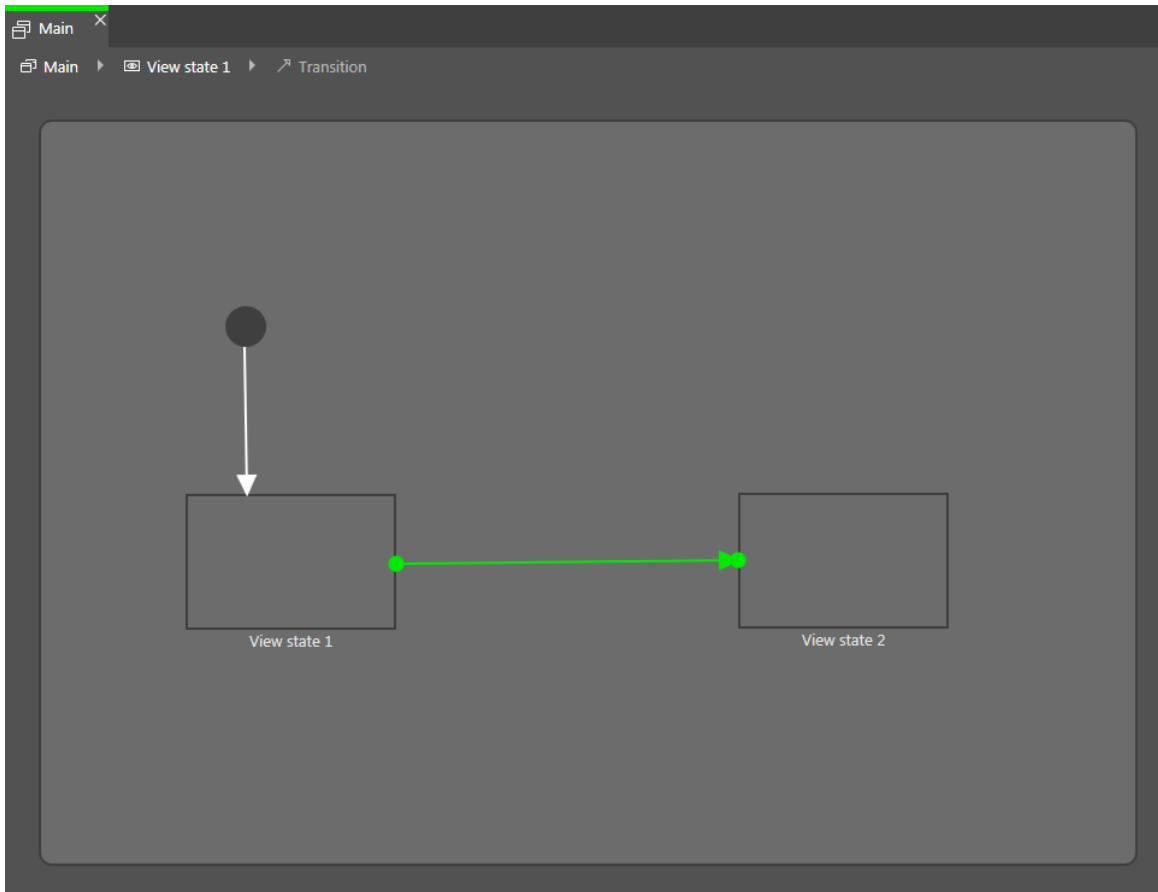


그림 7.3. 전이

전이 추가되어 녹색 화살표로 표시됩니다.

**[작은 정보]** 상태 머신에 전이 연결



상태 머신은 최상위 복합 상태입니다. 따라서 상태 머신 테두리에서 들어오고 나가는 전이를 만들 수 있습니다. 상태 머신의 모든 상태는 이러한 전이를 상속합니다.

### 7.3.2. 전이 이동



### 전이 이동

전이의 끝 포인트 중 하나를 움직여 전이를 이동합니다.

선수학습:

- 콘텐츠 영역에 상태 머신이 표시되어 있어야 합니다.
- 상태 머신에 상태가 2개 이상 포함되어 있어야 합니다.
- 상태가 전이로 연결되어 있어야 합니다.

#### 단계 1

콘텐츠 영역에서 전이를 클릭합니다.

녹색 끌기 포인트 2개가 표시됩니다.

#### 단계 2

이동할 끌기 포인트를 클릭하고 마우스 버튼을 누른 상태로 유지합니다.

#### 단계 3

마우스를 다른 상태로 끕니다.

#### 단계 4

상태가 녹색으로 강조 표시되면 마우스 버튼을 놓습니다.

전이가 이동됩니다.

## 7.3.3. 전이에 대해 트리거 정의



### 전이에 대해 트리거 정의

전이에 대해 해당 전이를 트리거하는 이벤트를 정의할 수 있습니다.

선수학습:

- 상태 머신에 상태가 2개 이상 포함되어 있어야 합니다.
- 상태가 전이로 연결되어 있어야 합니다.

#### 단계 1

전이를 선택합니다.

#### 단계 2

속성 구성 요소에서 **트리거** 콤보 박스를 확장합니다.

단계 3  
이벤트를 선택합니다.

단계 4  
새 이벤트를 만들려면 트리거 콤보 박스에 이름을 입력하고 **이벤트 추가**를 클릭합니다.

이벤트가 전이 트리거로 추가됩니다.

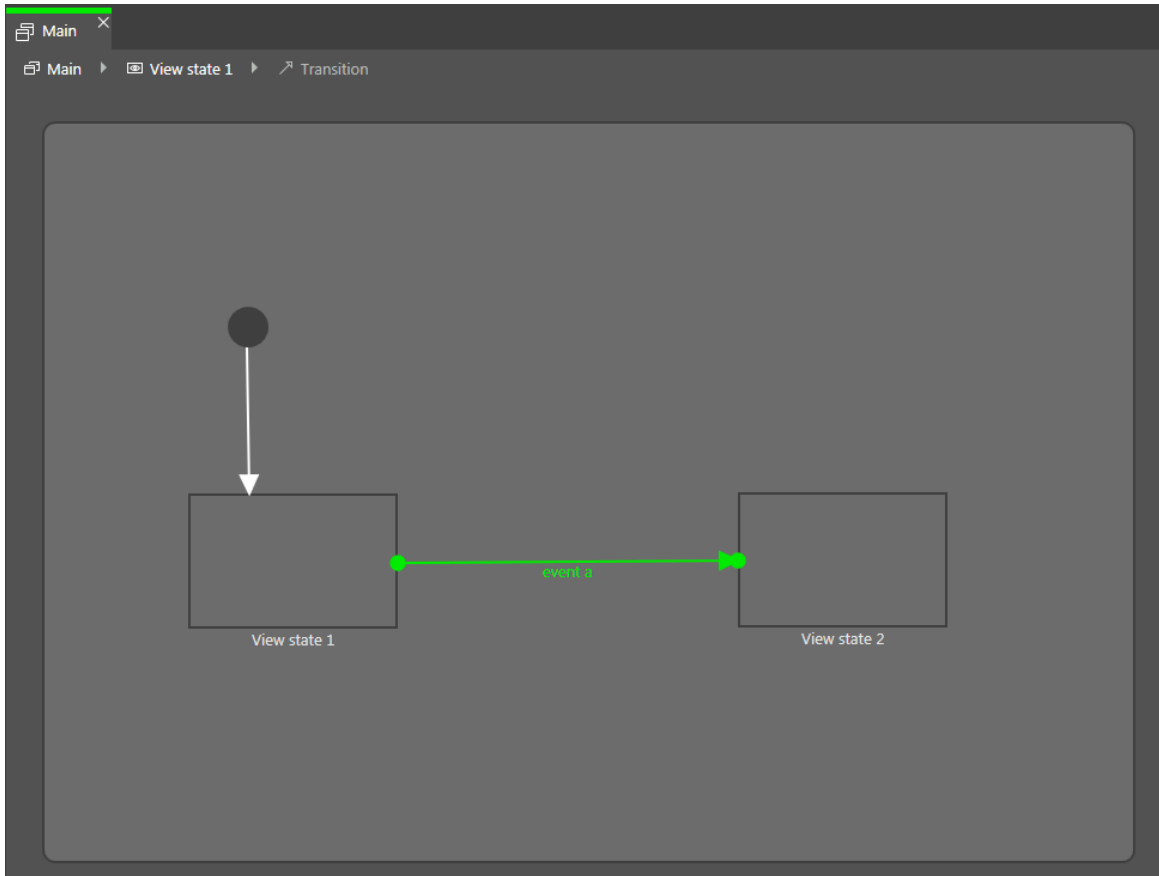


그림 7.4. 트리거가 포함된 전이

### 7.3.4. 전이에 조건 추가



#### 전이에 조건 추가

모든 전이에 대해 전이를 실행하려면 충족되어야 하는 조건을 정의할 수 있습니다.

선수학습:

- 상태 머신에 상태가 2개 이상 포함되어 있어야 합니다.

- 상태가 전이로 연결되어 있어야 합니다.

단계 1

전이를 선택합니다.

단계 2

전이에 조건을 추가하려면 **속성** 구성 요소로 이동합니다. Condition 속성 옆의 **추가**를 클릭합니다.

단계 3

EB GUIDE 스크립트를 사용하여 조건을 입력합니다.

배경 정보는 [섹션 6.15. "스크립팅 언어 EB GUIDE 스크립트"](#) 항목을 참조하십시오.

단계 4

**수락**을 클릭합니다.

조건이 전이에 추가됩니다.

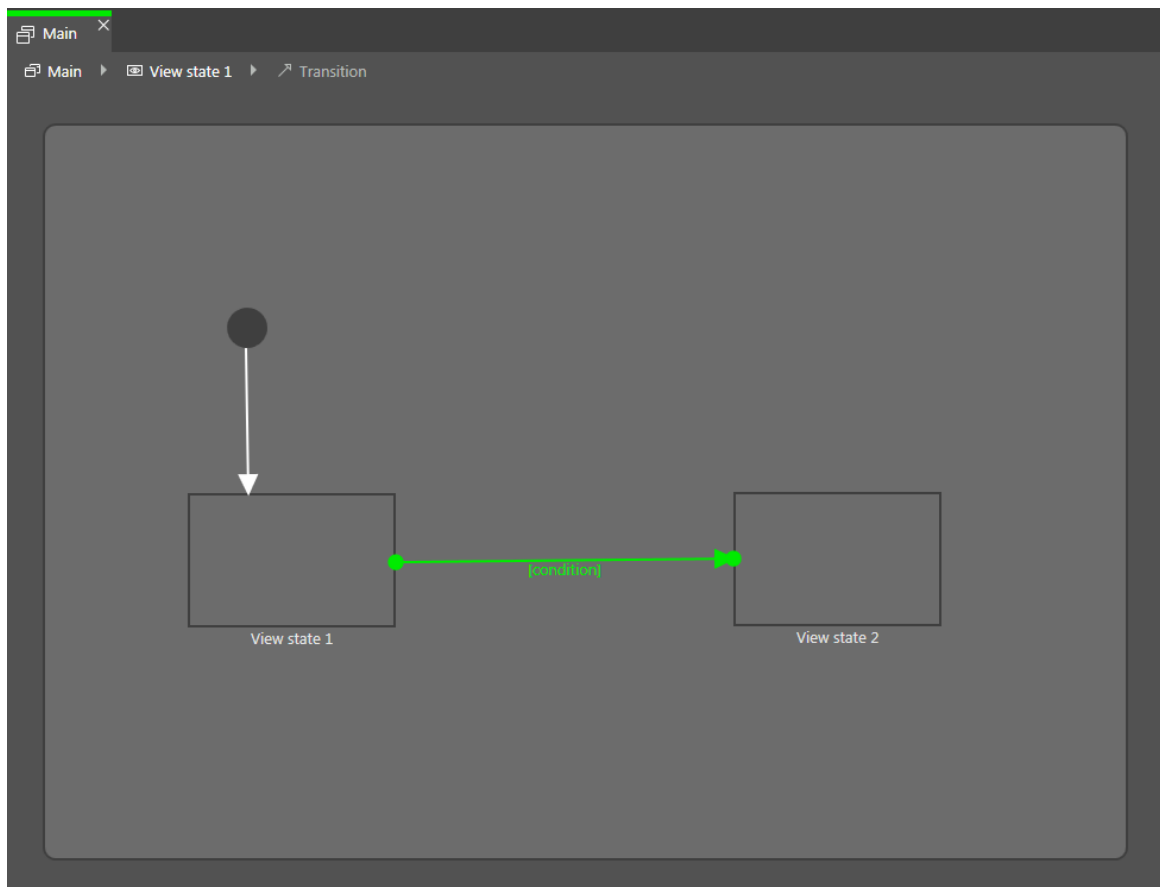


그림 7.5. 조건이 포함된 전이

### 7.3.5. 전이에 작업 추가



## 전이에 작업 추가

모든 전이에 대해 전이와 함께 실행되는 작업을 정의할 수 있습니다.

선수학습:

- 상태 머신에 상태가 2개 이상 포함되어 있어야 합니다.
- 상태가 전이로 연결되어 있어야 합니다.

### 단계 1

전이를 선택합니다.

### 단계 2

전이에 동작을 추가하려면 **속성** 구성 요소로 이동합니다. Action 속성 옆의 **추가**를 클릭합니다.

### 단계 3

EB GUIDE 스크립트를 사용하여 작업을 입력합니다.

배경 정보는 [섹션 6.15. "스크립팅 언어 EB GUIDE 스크립트"](#) 항목을 참조하십시오.

### 단계 4

**수락**을 클릭합니다.

작업이 전이에 추가됩니다.

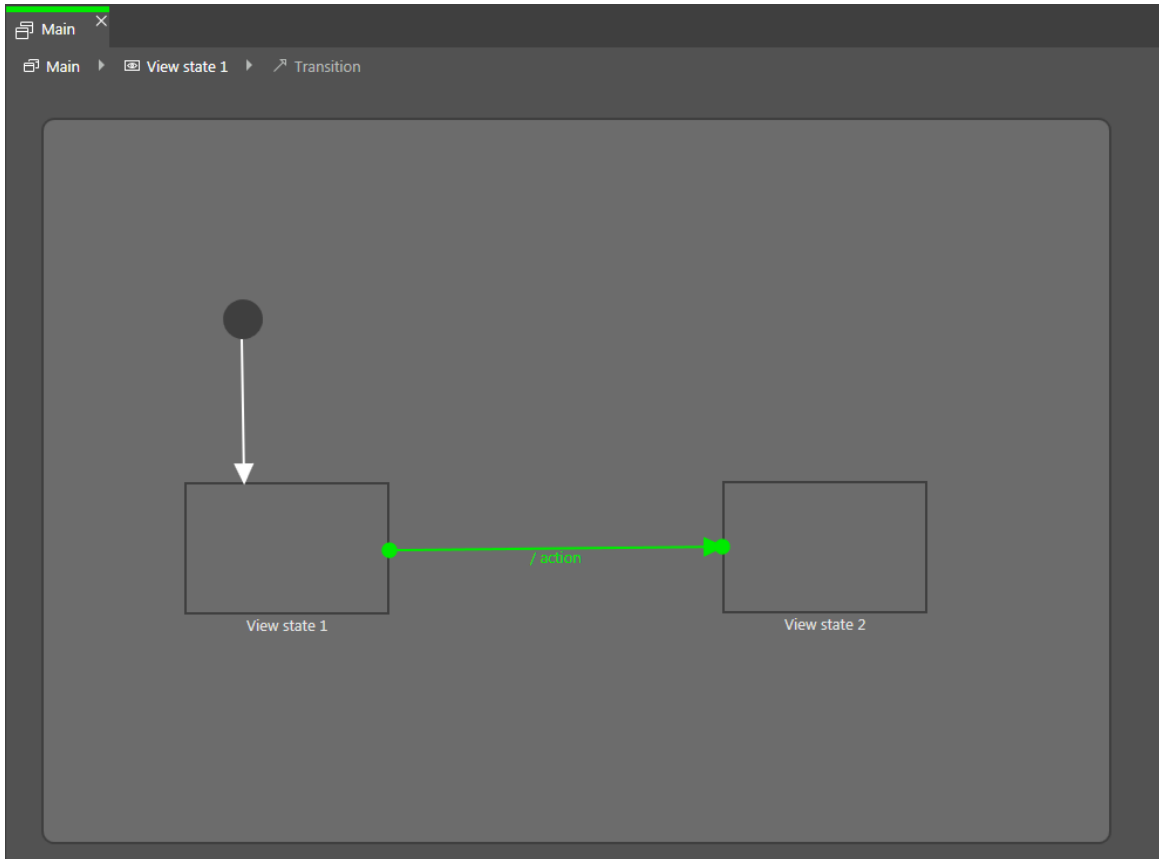


그림 7.6. 작업이 포함된 전이

### 7.3.6. 상태에 내부 전이 추가



#### 상태에 내부 전이 추가

선수학습:

- 상태 머신에 상태가 포함되어 있어야 합니다.

#### 단계 1

상태를 선택합니다.

#### 단계 2

속성 구성 요소에서 **내부 전이**로 이동한 다음 **추가**를 클릭합니다.

내부 전이가 상태에 추가됩니다. 내부 전이는 **내비게이션** 구성 요소에 표시됩니다.

## 8. HMI 모양 모델링

### [참고]



#### 기본 창 레이아웃

이 사용자 설명서의 모든 지침 및 스크린샷이 기본 창 레이아웃을 사용합니다. 지침을 따르는 경우 EB GUIDE Studio 창을 기본 레이아웃으로 설정하는 것이 좋습니다. **레이아웃 > 기본 레이아웃**으로 다시 설정을 선택하여 설정할 수 있습니다.

## 8.1. 위젯으로 작업

### [작은 정보]



#### 보기와 위젯 복사 및 찾기

컨텍스트 메뉴 또는 Ctrl+C 및 Ctrl+V를 사용하여 기존 보기 또는 위젯을 복사하고 붙여 넣을 수도 있습니다.

EB GUIDE 모델 내의 특정 보기 또는 위젯을 찾으려면 검색 상자에 보기 또는 위젯 이름을 입력하거나 Ctrl+F를 사용합니다. 보기 또는 위젯으로 건너 뛰려면 결과 목록에서 더블 클릭합니다.

### 8.1.1. 보기 추가



#### 보기 추가

선수학습:

- 콘텐츠 영역에 상태 머신이 표시되어 있어야 합니다.

#### 단계 1

보기 상태를 **도구 상자**에서 상태 머신으로 끕니다.

보기 상태와 함께 보기 하나가 모델에 추가됩니다.

#### 단계 2

**내비게이션** 구성 요소에서 보기를 클릭합니다.

#### 단계 3

F2 키를 누르고 보기의 이름을 바꿉니다.

#### 단계 4

콘텐츠 영역에서 보기 상태를 더블 클릭합니다.

콘텐츠 영역에 새 보기가 표시됩니다.

## 8.1.2. 보기에 기본 위젯 추가

기본 위젯에 대한 자세한 내용은 [섹션 12.12.2. "기본 위젯"](#) 항목을 참조하십시오.

### 8.1.2.1. 사각형 추가



#### 사각형 추가

선수학습:

- 콘텐츠 영역에 보기가 표시되어 있어야 합니다.

단계 1

사각형을 **도구 상자**에서 보기로 끕니다.

사각형이 보기에 추가됩니다.

### 8.1.2.2. 타원 추가



#### 타원 추가

선수학습:

- 콘텐츠 영역에 보기가 표시되어 있어야 합니다.

단계 1

타원을 **도구 상자**에서 보기로 끕니다.

위젯이 보기에 추가됩니다.

#### 8.1.2.2.1. 타원 편집

타원의 부채꼴만 그리거나 타원의 호를 변경할 수도 있습니다.





### 부채꼴 생성

선수학습:

- 보기에 타원이 포함되어 있어야 합니다.

#### 단계 1

타원을 클릭하고 **속성** 구성 요소로 이동합니다.

#### 단계 2

centralAngle 텍스트 상자에서 부채꼴의 각도를 입력합니다.

#### 단계 3

sectorRotation 텍스트 상자에서 부채꼴의 방향을 입력합니다.

부채꼴을 생성했습니다.



### 원호 생성

선수학습:

- 보기에 타원이 포함되어 있어야 합니다.

#### 단계 1

타원을 클릭하고 **속성** 구성 요소로 이동합니다.

#### 단계 2

arcWidth 텍스트 상자에 호의 너비를 입력합니다.

원호를 생성했습니다.

## 8.1.2.3. 이미지 추가



### 도구 상자를 사용하여 이미지 추가

선수학습:

- 이미지 파일이 \$GUIDE\_PROJECT\_PATH/<project name>/resources 디렉터리에 있어야 합니다. 지원되는 파일 유형은 [섹션 6.14.3. "이미지"](#) 항목을 참조하십시오.
- 콘텐츠 영역에 보기가 표시되어 있어야 합니다.

#### 단계 1

이미지를 **도구 상자**에서 보기로 끕니다.

단계 2

**속성** 구성 요소의 image 콤보 박스에서 이미지를 선택합니다. 또는 다른 이미지를 **자산** 구성 요소에서 image 드롭다운 목록 상자로 끕니다.

보기에 이미지가 표시됩니다.



자산 구성 요소를 사용하여 이미지 추가

선수학습:

- 이미지 파일이 \$GUIDE\_PROJECT\_PATH/<project name>/resources 디렉터리에 있어야 합니다. 지원되는 파일 유형은 [섹션 6.14.3, "이미지"](#) 항목을 참조하십시오.
- 콘텐츠 영역에 보기가 표시되어 있어야 합니다.

단계 1

이미지를 **자산** 구성 요소에서 보기로 끕니다.

보기에 이미지가 표시됩니다.

단계 2

이미지 파일을 변경하려면 **속성** 구성 요소로 이동한 다음 image 콤보 박스에서 이미지를 선택합니다. 또는 다른 이미지를 **자산** 구성 요소에서 image 콤보 박스로 끕니다.

보기에 이미지가 표시됩니다.



나인패치 이미지 추가

선수학습:

- 나인패치 이미지 파일이 \$GUIDE\_PROJECT\_PATH/<project name>/resources 디렉터리에 있어야 합니다. 나인패치 이미지에 대한 배경 정보는 [섹션 6.14.3.1, "나인패치 이미지"](#) 항목을 참조하십시오.
- 콘텐츠 영역에 보기가 표시되어 있어야 합니다.
- 이미지가 EB GUIDE 모델에 추가되어 있어야 합니다.

단계 1

이미지를 선택하고 **속성** 구성 요소로 이동합니다.

단계 2

image 콤보 박스에서 나인패치 이미지를 선택합니다.

단계 3

**위젯 피쳐 속성**으로 이동하여 **추가/제거**를 클릭합니다.

**위젯 피쳐** 대화 상자가 표시됩니다.

단계 4

사용 가능한 위젯 피처에서 레이아웃 범주를 확장하고 크기 조정 모드를 선택합니다.

단계 5

수락을 클릭합니다.

관련 위젯 속성이 이미지에 추가되고 속성 구성 요소에 표시됩니다.

단계 6

속성 구성 요소에서 scaleMode 속성에 대해 fit to Size (=1)를 선택합니다.

[참고]



**나인패치 이미지 추가**

크기 조정 모드 위젯 피처를 추가하지 않거나 scaleMode 속성에 original Size (=0) 또는 keep aspect ratio (=2)를 선택한 경우 나인패치 이미지는 일반 .png 이미지로 크기가 조정됩니다.

### 8.1.2.4. 레이블 추가



#### 도구 상자를 사용하여 레이블 추가

선수학습:

- 콘텐츠 영역에 보기가 표시되어 있어야 합니다.

단계 1

레이블을 도구 상자에서 보기로 끕니다.

레이블이 보기에 추가됩니다. 레이블에는 기본 글꼴인 PT\_Sans\_Narrow.ttf가 있습니다.



#### 자산 구성 요소를 사용하여 레이블 추가

선수학습:

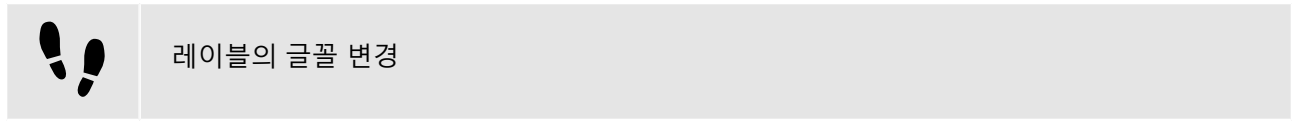
- 글꼴 파일은 \$GUIDE\_PROJECT\_PATH/<project name>/resources 디렉터리에 있습니다. 지원되는 파일 유형은 [섹션 6.14.1. "글꼴"](#) 항목을 참조하십시오.
- 콘텐츠 영역에 보기가 표시되어 있어야 합니다.

단계 1

글꼴 파일을 자산 구성 요소에서 보기로 끕니다.

보기에 선택한 글꼴이 적용된 레이블이 표시됩니다.

### 8.1.2.4.1. 레이블의 글꼴 변경



선수학습:

- 글꼴 파일은 \$GUIDE\_PROJECT\_PATH/<project name>/resources 디렉터리에 있습니다. 지원되는 파일 유형은 [섹션 6.14.1, "글꼴"](#) 항목을 참조하십시오.
- EB GUIDE 모델에 보기 상태가 포함되어 있어야 합니다.
- 보기에 레이블이 포함되어 있어야 합니다.

#### 단계 1

보기에서 레이블을 선택합니다.

#### 단계 2

속성 구성 요소의 font 콤보 박스에서 글꼴을 선택합니다.

또는 글꼴 파일을 **자산** 구성 요소에서 font 콤보 박스로 끕니다.

보기에 새 글꼴이 적용된 레이블이 표시됩니다. .fnt 비트맵 글꼴을 선택한 경우 글꼴 크기가 고정되며 레이블의 font 속성에서 이를 변경할 수 없다는 점에 유의하십시오.

### [참고]



#### 텍스트 높이 및 줄 간격 계산

다음 그림에서는 EB GUIDE Studio에서 텍스트 높이, 줄 높이 및 줄 간격을 계산하는 방법을 보여 줍니다. 레이블의 글꼴 스타일, 크기 또는 줄 간격을 변경할 때 고려하시기 바랍니다.

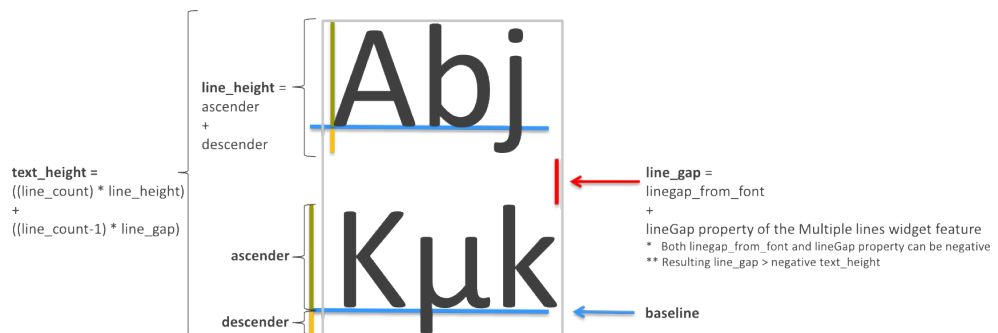


그림 8.1. 텍스트 높이, 줄 높이 및 줄 간격 계산

### 8.1.2.5. 컨테이너 추가



### 컨테이너 추가

컨테이너에서는 위젯을 그룹화할 수 있습니다.

선수학습:

- 콘텐츠 영역에 보기가 표시되어 있어야 합니다.

#### 단계 1

컨테이너를 **도구 상자**에서 보기로 끕니다.

#### 단계 2

콘텐츠 영역에서 컨테이너의 모서리 중 하나를 끌어 컨테이너를 확대합니다.

#### 단계 3

위젯을 두 개 이상 **도구 상자**에서 컨테이너로 끕니다.

위젯이 컨테이너의 하위 위젯으로 모델링됩니다. 컨테이너를 이동하면 하위 위젯도 함께 이동합니다.

## 8.1.2.6. 인스턴스화기 추가



### 인스턴스화기 추가

선수학습:

- 콘텐츠 영역에 보기가 표시되어 있어야 합니다.

#### 단계 1

인스턴스화기를 **도구 상자**에서 보기로 끕니다.

#### 단계 2

위젯을 **도구 상자**에서 인스턴스화기로 끕니다.

위젯은 줄 템플릿으로 사용됩니다.

#### 단계 3

인스턴스화기를 선택하고 **속성** 구성 요소로 이동합니다.

##### 단계 3.1

numItems 속성에 대해 1보다 큰 값을 입력합니다.

##### 단계 3.2

다음 위젯 피처 중 하나를 인스턴스화기에 추가합니다.

- ▶ 상자 레이아웃
- ▶ 흐름 레이아웃

- ▶ **눈금 레이아웃**
- ▶ **목록 레이아웃**

자세한 내용은 [섹션 8.3.1. "위젯 피처 추가"](#) 항목을 참조하십시오.

보기에서 하위 위젯이 인스턴스화기에 대해 위젯 피처가 지정한 레이아웃에 numItems 속성으로 지정된 횟수만큼 표시됩니다.

#### 단계 4

위젯을 **도구 상자**에서 인스턴스화기로 끕니다.

두 번째 줄 템플릿으로 사용하는 두 번째 하위 위젯이 추가됩니다.

#### 단계 5

인스턴스화기를 선택하고 **속성** 구성 요소로 이동합니다.

##### 단계 5.1

lineMapping 옆의  을 클릭합니다.

##### 단계 5.2

**추가** 버튼을 클릭합니다.

새 항목이 테이블에 추가됩니다.

##### 단계 5.3

Value 텍스트 상자에 0을 입력합니다.

##### 단계 5.4

**추가** 버튼을 클릭합니다.

새 항목이 테이블에 추가됩니다.

##### 단계 5.5

Value 텍스트 상자에 1을 입력합니다.

줄 템플릿이 인스턴스화되는 순서가 정의됩니다.



### 예 8.1. 인스턴스화 순서

lineMapping 속성이 인스턴스화의 순서를 정의합니다. 예를 들어, 1|0 값을 입력하면 인스턴스화기에서 줄 템플릿 1을 첫 번째 하위 위젯 템플릿으로 인스턴스화하며 줄 템플릿 0을 두 번째 하위 위젯으로 인스턴스화합니다.

lineMapping 속성이 반복적으로 적용됩니다. numItems 속성의 경우 10을 입력하면 결과는 1|0|1|0|1|0|1|0|1|0 순서가 됩니다.

인스턴스화기를 사용하는 방법에 대한 상세 예제는 [섹션 11.4. "자습서: 동적 콘텐츠로 목록 만들기"](#) 항목을 참조하십시오.

[참고]



**줄 템플릿의 속성 연결**

다음은 연결 규칙입니다.

- ▶ 줄 템플릿 간의 속성을 연결할 수 없습니다.
- ▶ 인스턴스화기 외부에서 해당 줄 템플릿으로 연결할 수 없습니다.
- ▶ 줄 템플릿에서 해당 인스턴스화기로 연결할 수 없습니다.

### 8.1.2.7. 애니메이션 추가



#### 애니메이션 추가

커브에 대한 자세한 내용과 커브 속성에 대한 설명은 [섹션 12.12.2.2. "애니메이션"](#) 항목을 참조하십시오.

선수학습:

- 콘텐츠 영역에 보기가 표시되어 있어야 합니다.

단계 1

기본 위젯 중 하나를 **도구 상자**에서 보기로 끕니다.

단계 2

애니메이션을 **도구 상자**에서 추가한 위젯으로 끕니다.

단계 3

**애니메이션** 편집기로 이동하여 **애니메이션 속성** 옆에서 **+**를 클릭합니다.

메뉴가 확장됩니다.

단계 4

**애니메이션 속성**에서 애니메이션을 생성할 속성을 선택하고 **애니메이션 커브**에서 각 커브를 선택합니다.

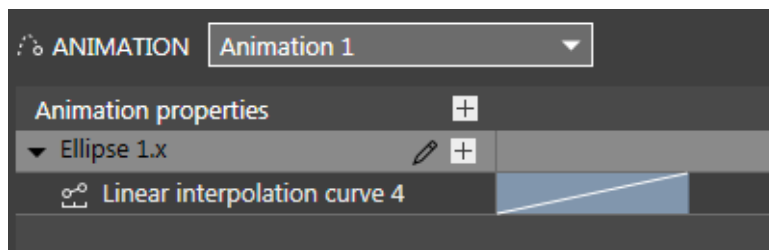


그림 8.2. 예제 커브가 포함된 애니메이션 편집기

단계 5

기본 위젯을 선택하고 Conditional script 유형의 사용자 정의 속성을 추가합니다. 자세한 내용은 [섹션 8.2.5. "위젯에 사용자 정의 속성 추가"](#) 항목을 참조하십시오.

#### 단계 6

속성 이름 옆의 속성 구성 요소에서 **편집**을 클릭합니다.

EB GUIDE 스크립트 편집기가 열립니다.

#### 단계 7

다음 EB GUIDE 스크립트를 입력합니다.

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

**Animation 1**은 처음으로 추가하는 애니메이션의 기본 이름입니다. 2단계에서 추가한 애니메이션의 이름이 다른 경우 **On trigger** 스크립트의 이름을 바꾸십시오.

#### 단계 8

시뮬레이션을 시작합니다.

위젯의 연결된 속성이 추가한 커브에 의해 지정된 대로 서서히 변경됩니다.

후속 단계로 애니메이션 또는 커브의 속성을 변경할 수 있습니다.

#### 단계 9

커브 동작을 변경하려면 속성 구성 요소에서 애니메이션 또는 커브 속성을 편집합니다.

**애니메이션** 편집기에서 커브 동작을 미리 볼 수 있습니다.

구체적인 애니메이션 예제는 [섹션 11.5. "자습서: 화면을 가로지르는 타원 이동 생성"](#) 항목을 참조하십시오.

### 8.1.2.8. 알파 마스크 추가



#### 알파 마스크 추가

알파 마스크에 대한 자세한 내용은 [섹션 12.12.2.1. "알파 마스크"](#) 항목을 참조하십시오.

선수학습:

- \$GUIDE\_PROJECT\_PATH/<project name>/resources 디렉터리에 이미지가 있습니다.
- 콘텐츠 영역에 보기가 표시되어 있어야 합니다.

#### 단계 1

알파 마스크를 **도구 상자**에서 보기로 끕니다.

#### 단계 2

**속성** 구성 요소로 이동하여 image 드롭다운 목록 상자에서 이미지를 선택합니다.



[참고]



**알파 마스크에 지원되는 이미지 파일 유형**

사용 가능한 이미지 형식은 렌더러 구현에 따라 달라집니다. DirectX 11 및 OpenGL ES 버전 2.0 이상에서는 .png 파일과 .jpg 파일을 지원합니다. RGB 이미지는 알파 마스크로 사용되기 전에 흑백 이미지로 변환됩니다. 흑백 이미지는 그대로 사용됩니다. 이미지의 알파 채널은 무시됩니다.

나인패치 이미지로는 알파 마스크를 사용할 수 없습니다.

단계 3

도구 상자의 기본 위젯 중 하나를 알파 마스크에 하위 위젯으로 추가합니다.

알파 채널(즉, 하위 위젯의 투명도)은 알파 마스크로 제어됩니다.

## 8.1.3. 보기에 3D 위젯 추가

### 8.1.3.1. 보기에 장면 그래프 추가



보기에 장면 그래프 추가

제한 및 권장 사항은 [섹션 6.1.2, "3D 그래픽 파일의 설정"](#) 항목을 참조하십시오.

선수학습:

- 3D 그래픽 파일을 사용할 수 있습니다. 이 파일에는 카메라, 광원 및 메시와 하나 이상의 재료가 있는 개체 하나가 포함되어 있어야 합니다. 지원되는 3D 그래픽 파일 형식은 [섹션 6.1.1, "지원되는 3D 그래픽 형식"](#) 항목을 참조하십시오.
- 콘텐츠 영역에 보기가 표시되어 있어야 합니다.

단계 1

장면 그래프를 도구 상자에서 보기로 끕니다.

보기에 비어 있는 경계 상자가 표시됩니다.

단계 2

속성 구성 요소에서 **파일 가져오기**를 클릭합니다.

대화 상자가 열립니다.

단계 3

3D 그래픽 파일이 저장되어 있는 디렉터리로 이동합니다.

단계 4

3D 그래픽 파일을 선택합니다.

단계 5

**열기**를 클릭합니다.

가져오기가 시작됩니다. 대화 상자가 열립니다.

단계 6

**확인**을 클릭합니다.

보기에 3D 그래픽이 표시됩니다. **내비게이션** 구성 요소에 장면 그래프를 사용하여 가져온 위젯 트리가 상위 노드로 표시됩니다. 가져온 3D 장면애 애니메이션이 있는 경우, 선형 키값 보간 정수 또는 선형 키값 보간 부동 커브가 추가됩니다. 이러한 커브의 근본적인 키값 쌍은 EB GUIDE Studio에서 수정할 수 있습니다.

**[작은 정보] 여러 항목 가져오기**



하나의 장면 그래프 내의 여러 3D 그래픽을 가져올 수 있습니다.

가져오고 나면 여러 3D 그래픽이 각 항목의 맨 위에 렌더링됩니다. 3D 객체를 별도로 표시하려면 RootNode의 visible 속성을 사용합니다.

## 8.1.4. 보기에 .psd 파일 추가



### 보기에 .psd 파일 추가

선수학습:

- .psd 파일은 \$GUIDE\_PROJECT\_PATH/<project name>/resources에서 사용할 수 있습니다. 배경 정보는 [섹션 6.13, "Photoshop 파일 형식 지원"](#) 항목을 참조하십시오.
- 콘텐츠 영역에 보기가 표시되어 있어야 합니다.

단계 1

**자산** 구성 요소에서 resource 폴더를 선택합니다.

단계 2

미리 보기 영역에서 .psd 파일을 콘텐츠 영역으로 끕니다.

가져오기 상태 메시지가 표시됩니다.

단계 3

**확인**을 클릭합니다.

가져오기가 성공하면 **내비게이션** 구성 요소에 .psd 파일에서 생성된 위젯 트리가 표시됩니다. 위젯 트리는 컨테이너 및 이미지로 구성되며 .psd 파일 구조를 반영합니다. \$GUIDE\_PROJECT\_PATH/<project name>/resources 디렉터리에서 추출된 모든 이미지가 있는 하위 디렉터리가 생성됩니다.



### .psd 파일에서 이미지 추출

이미지를 가져오지 않고 .psd 파일에서 추출할 수 있습니다. 이는 위젯 트리가 생성되지 않는다는 의미입니다.

선수학습:

- .psd 파일은 \$GUIDE\_PROJECT\_PATH/<project name>/resources에서 사용할 수 있습니다. 배경 정보는 [섹션 6.13, "Photoshop 파일 형식 지원"](#) 항목을 참조하십시오.
- 콘텐츠 영역에 보기가 표시되어 있어야 합니다.

단계 1

**자산** 구성 요소에서 .psd 파일을 오른쪽 마우스 클릭하고 **.psd 파일에서 이미지 생성**을 선택합니다.

\$GUIDE\_PROJECT\_PATH/<project name>/resources 디렉터리에서 추출된 모든 이미지가 있는 하위 디렉터리가 생성됩니다.

## 8.1.5. 보기에서 위젯 삭제



### 보기에서 위젯 삭제

선수학습:

- EB GUIDE 모델에 위젯이 포함되어 있어야 합니다.

단계 1

**내비게이션** 구성 요소에서 위젯을 오른쪽 마우스 클릭합니다.

단계 2

컨텍스트 메뉴에서 **삭제**를 클릭합니다.

위젯이 삭제됩니다.

[작은 정보] 콘텐츠 영역에서 위젯 삭제



콘텐츠 영역에서 위젯을 선택한 다음 Delete 키를 눌러 삭제할 수도 있습니다.

## 8.2. 위젯 속성으로 작업

### 8.2.1. 위젯 배치



#### 위젯 배치

위젯의 x 및 y 속성을 조정하여 위젯을 배치합니다. 상위 위젯의 왼쪽 상단에 x 및 y 값이 모두 0인 원점이 있습니다.

선수학습:

- 콘텐츠 영역에 보기가 표시되어 있어야 합니다.
- 보기에 위젯이 포함되어 있어야 합니다.

#### 단계 1

위젯을 선택합니다.

**속성** 구성 요소에 선택한 위젯의 속성이 표시됩니다.

#### 단계 2

위젯의 x 좌표를 정의하려면 x 텍스트 상자에 값을 입력합니다.

#### 단계 3

위젯의 y 좌표를 정의하려면 y 텍스트 상자에 값을 입력합니다.

#### 단계 4

텍스트 상자 바깥쪽을 클릭합니다.

콘텐츠 영역에서 위젯이 입력한 위치에 표시됩니다.

**[작은 정보] 대체 방식**



육안으로 확인하여 위젯을 배치하려면 콘텐츠 영역에서 위젯을 선택하여 마우스로 이동합니다.

## 8.2.2. 위젯 크기 조정



### 위젯 크기 조정

선수학습:

- 콘텐츠 영역에 보기가 표시되어 있어야 합니다.
- 보기에 위젯이 포함되어 있어야 합니다.

단계 1

위젯을 선택합니다.

속성 구성 요소에 선택한 위젯의 속성이 표시됩니다.

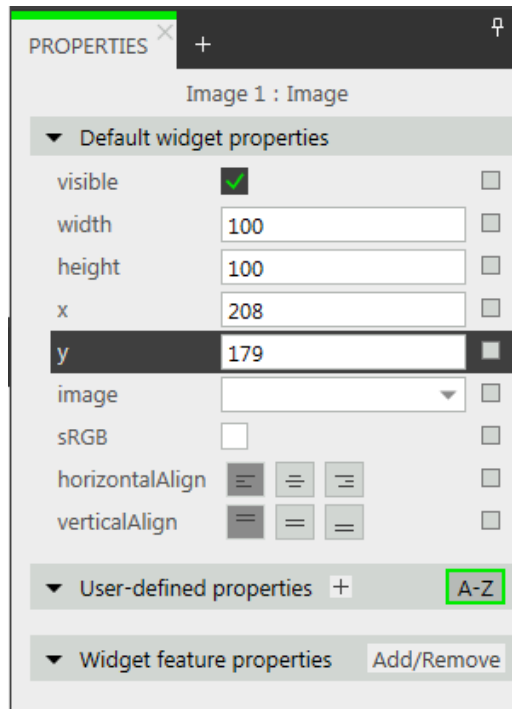


그림 8.3. 이미지의 속성

단계 2

위젯의 높이를 정의하려면 height 텍스트 상자에 값을 입력합니다.

단계 3

위젯의 너비를 정의하려면 width 텍스트 상자에 값을 입력합니다.

단계 4

텍스트 상자 바깥쪽을 클릭합니다.

콘텐츠 영역에서 위젯이 입력한 크기로 표시됩니다.

**[참고]**

**음수**



height 및 width 속성에 음수를 입력하지 마십시오. EB GUIDE Studio는 음수를 0으로 처리하므로 해당 위젯이 표시되지 않습니다.

**[작은 정보]**

**대체 방식**



육안으로 확인하여 위젯의 크기를 조정하려면 콘텐츠 영역에서 위젯을 선택하고 그중 모서리 하나를 마우스로 끕니다.

### 8.2.3. 위젯 속성을 서로 연결



#### 위젯 속성을 서로 연결

두 위젯 속성의 값이 항상 같도록 하려는 경우 두 위젯 속성을 연결할 수 있습니다. 예를 들어 다음 지침에서는 사각형의 width 속성을 보기의 width 속성에 연결하는 방법을 설명합니다.

위젯 속성은 같은 보기 내에서만 연결할 수 있음

인스턴스화기의 하위 위젯 속성에는 연결할 수 없습니다.

선수학습:

- EB GUIDE 모델에 보기 상태가 포함되어 있어야 합니다.
- 보기에 사각형이 포함되어 있어야 합니다.
- 사각형의 width 속성은 스크립팅된 값이 아닙니다.

단계 1

사각형을 클릭합니다.

속성 구성 요소에 사각형의 속성이 표시됩니다.

단계 2

속성 구성 요소에서 width 속성으로 이동한 다음 속성 옆의  버튼을 클릭합니다.

메뉴가 확장됩니다.

단계 3

메뉴에서 위젯 속성에 대한 링크 추가를 클릭합니다.

대화 상자가 열립니다.

단계 4

대화 상자에서 보기로 이동한 다음 해당 width 속성을 선택합니다.

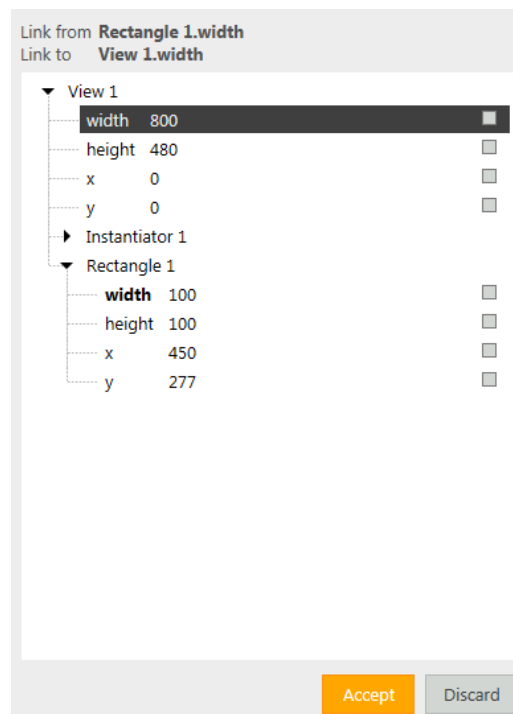


그림 8.4. 위젯 속성을 서로 연결

단계 5

수락을 클릭합니다.

대화 상자가 닫힙니다.  버튼이 width 속성 옆에 표시됩니다. 이 버튼은 사각형의 width 속성이 이제 보기의 width 속성에 연결되었음을 나타냅니다. 그러면 보기나 직사각형 중 하나의 너비를 변경할 때마다 다른 하나의 너비도 변경됩니다.

[참고]



링크 원본 및 링크 대상

■ 버튼만 링크 원본 옆에 표시됩니다. 링크 대상의 경우에는 이 버튼이 표시되지 않습니다.

[작은 정보]



링크 제거

링크를 제거하려면 ■ 버튼을 다시 클릭합니다. 메뉴가 열리면 **링크 제거**를 클릭합니다.

## 8.2.4. 데이터폴 항목에 위젯 속성 연결



### 데이터폴 항목에 위젯 속성 연결

위젯 속성과 데이터폴 항목의 값이 항상 같도록 하려는 경우 데이터폴 항목에 위젯 속성을 연결할 수 있습니다. 예를 들어 다음 지침에서는 이미지의 image 속성을 새 데이터폴 항목에 연결하는 방법을 설명합니다.

선수학습:

- EB GUIDE 모델에 보기 상태가 포함되어 있어야 합니다.
- 보기에 이미지가 포함되어 있어야 합니다.
- 이미지의 image 속성은 스크립팅된 값이 아닙니다.

단계 1

이미지를 클릭합니다.

속성 구성 요소에 이미지의 속성이 표시됩니다.

단계 2

속성 구성 요소에서 image 속성으로 이동한 다음 속성 옆의 ■ 버튼을 클릭합니다.

메뉴가 확장됩니다.

단계 3

메뉴에서 **데이터폴 항목에 대한 링크 추가**를 클릭합니다.

대화 상자가 열립니다.

단계 4

새 데이터폴 항목을 추가하려면 텍스트 상자에 이름을 입력합니다.

단계 5

**데이터폴 항목 추가**를 클릭합니다.



단계 6  
수락을 클릭합니다.

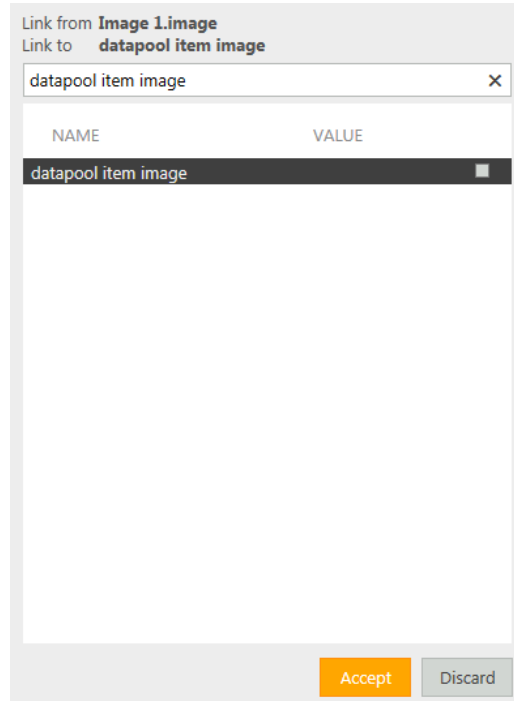


그림 8.5. 데이터풀 항목에 연결

새 데이터풀 항목이 추가됩니다.

단계 7  
대화 상자가 닫힙니다. ■ 버튼이 image 속성 옆에 표시됩니다. 이 버튼은 image 속성이 이제 데이터풀 항목에 연결되었음을 나타냅니다. 그러면 이미지 또는 데이터풀 항목 중 하나를 변경할 때마다 다른 하나도 변경됩니다.

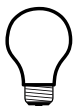
[참고]



링크 원본 및 링크 대상

■ 버튼만 링크 원본 옆에 표시됩니다. 링크 대상의 경우에는 이 버튼이 표시되지 않습니다.

[작은 정보]



링크 제거

링크를 제거하려면 ■ 버튼을 다시 클릭합니다. 메뉴가 열리면 링크 제거를 클릭합니다.

## 8.2.5. 위젯에 사용자 정의 속성 추가



### 위젯에 사용자 정의 속성 추가

선수학습:

- EB GUIDE 모델에 보기 상태가 포함되어 있어야 합니다.
- 보기에 위젯이 포함되어 있어야 합니다.

#### 단계 1

위젯을 선택합니다.

속성 구성 요소에 선택한 위젯의 속성이 표시됩니다.

#### 단계 2

속성 구성 요소에서 **사용자 정의 속성** 범주로 이동한 다음 **+**를 클릭합니다.

메뉴가 확장됩니다.

#### 단계 3

메뉴에서 사용자 정의 속성의 유형을 클릭합니다.

선택한 유형의 새 위젯 속성이 위젯에 추가됩니다.

#### 단계 4

속성 이름을 바꿉니다.

### 8.2.5.1. Function (): bool 유형의 사용자 정의 속성 추가



### Function (): bool 유형의 사용자 정의 속성 추가

Function (): bool 유형의 속성은 매개변수를 포함하지 않으며 부울 값을 반환하는 함수입니다. 위젯 속성과 인 수 목록의 주소를 차례로 지정하는 방식으로 EB GUIDE 스크립트에서 이 함수를 호출합니다.

선수학습:

- EB GUIDE 모델에 보기 상태가 포함되어 있어야 합니다.
- 보기에 위젯이 포함되어 있어야 합니다.

#### 단계 1

위젯을 선택합니다.

속성 구성 요소에 선택한 위젯의 속성이 표시됩니다.

#### 단계 2

속성 구성 요소에서 **사용자 정의 속성** 범주로 이동한 다음 **+**를 클릭합니다.

메뉴가 확장됩니다.

단계 3

메뉴에서 Function (): bool를 클릭합니다.

Function (): bool 유형의 새 위젯 속성이 위젯에 추가됩니다.

단계 4

속성 이름을 바꿉니다.

단계 5

속성 옆의 **편집**을 클릭합니다.

EB GUIDE 스크립트 편집기가 열립니다.

단계 6

EB GUIDE 스크립트를 사용하여 새 함수의 동작을 정의합니다.

단계 7

**수락**을 클릭합니다.



**예 8.2.**

**Function (): bool 유형의 속성 호출**

EB GUIDE 모델에 Background color라는 사각형이 있어야 합니다. Function (): bool 유형의 속성을 모델에 추가해야 합니다. 이 속성의 이름은 change입니다.

EB GUIDE 모델의 모든 EB GUIDE 스크립트 코드에서 다음과 같이 속성의 스크립트를 호출할 수 있습니다.

```
"Background color".change()
```

## 8.2.6. 사용자 정의 속성 이름 바꾸기



### 사용자 정의 속성 이름 바꾸기

선수학습:

- EB GUIDE 모델에 사용자 정의 속성이 포함된 위젯이 있어야 합니다.

단계 1

**내비게이션** 구성 요소에서 사용자 정의 속성이 있는 위젯을 선택합니다.

단계 2

**속성** 구성 요소에서 속성 이름을 오른쪽 마우스 클릭합니다.

메뉴가 확장됩니다.

단계 3

메뉴에서 **이름 바꾸기**를 클릭합니다.

단계 4

속성의 이름을 입력합니다.

단계 5

Enter 키를 누릅니다.

## 8.3. 위젯 피처로 위젯 확장

위젯 피처는 위젯 표시 및 동작과 관련된 기능을 추가로 제공합니다. 위젯 피처를 위젯에 추가하면 위젯 속성이 하나 이상 추가됩니다. 제공된 위젯 피처는 위젯의 유형에 따라 다릅니다.

### 8.3.1. 위젯 피처 추가



#### 위젯 피처 추가

선수학습:

- EB GUIDE 모델에 위젯이 포함되어 있어야 합니다.

단계 1

**내비게이션** 구성 요소에서 위젯을 클릭합니다.

**속성** 구성 요소에 선택한 위젯의 속성이 표시됩니다.

단계 2

**속성** 구성 요소에서 **위젯 피처 속성** 범주로 이동한 다음 **추가/제거**를 클릭합니다.

**위젯 피처** 대화 상자가 표시됩니다.

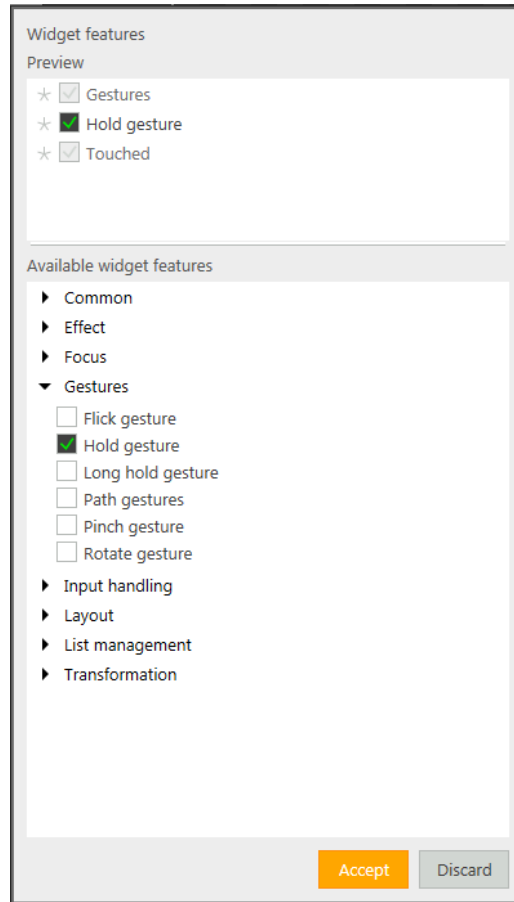


그림 8.6. 위젯 피쳐 대화 상자

단계 3

**사용 가능한 위젯 피쳐**에서 범주를 확장한 다음 추가할 위젯 피쳐를 선택합니다.

선택한 위젯 피쳐 및 해당 위젯 피쳐와 함께 자동으로 활성화되는 종속 위젯 피쳐가 **미리보기** 아래에 나열됩니다.

수락을 클릭합니다.

**[작은 정보]** 위젯 피쳐 간의 종속성



다른 위젯 피쳐가 있어야 사용 가능한 위젯 피쳐도 있습니다. 따라서 경우에 따라 위젯 피쳐를 선택하면 다른 위젯 피쳐가 자동으로 선택됩니다.

예를 들어 **이동 가능** 위젯 피쳐를 추가하려고 할 수 있습니다. 이 경우 **터치함** 및 **터치 이동** 위젯 피쳐도 자동으로 선택됩니다.

범주별로 그룹화된 위젯 피쳐 목록은 [섹션 12.13. "위젯 피쳐"](#) 항목을 참조하십시오.

자습서는 다음을 참조하십시오.

- ▶ [섹션 11.3. "자습서: 경로 제스처 모델링"](#)
- ▶ [섹션 11.4. "자습서: 동적 콘텐츠로 목록 만들기"](#)
- ▶ [섹션 11.2. "자습서: EB GUIDE 스크립트를 사용하여 버튼 동작 모델링"](#)

## 8.3.2. 위젯 피쳐 제거



### 위젯 피쳐 제거

선수학습:

- EB GUIDE 모델에 위젯이 포함되어 있어야 합니다.
- 위젯 피쳐를 위젯에 하나 이상 추가해야 합니다.

단계 1

**내비게이션** 구성 요소에서 위젯을 클릭합니다.

**속성** 구성 요소에 선택한 위젯의 속성이 표시됩니다.

단계 2

**속성** 구성 요소에서 **위젯 피쳐 속성** 범주로 이동한 다음 **추가/제거**를 클릭합니다.

**위젯 피쳐** 대화 상자가 표시됩니다.

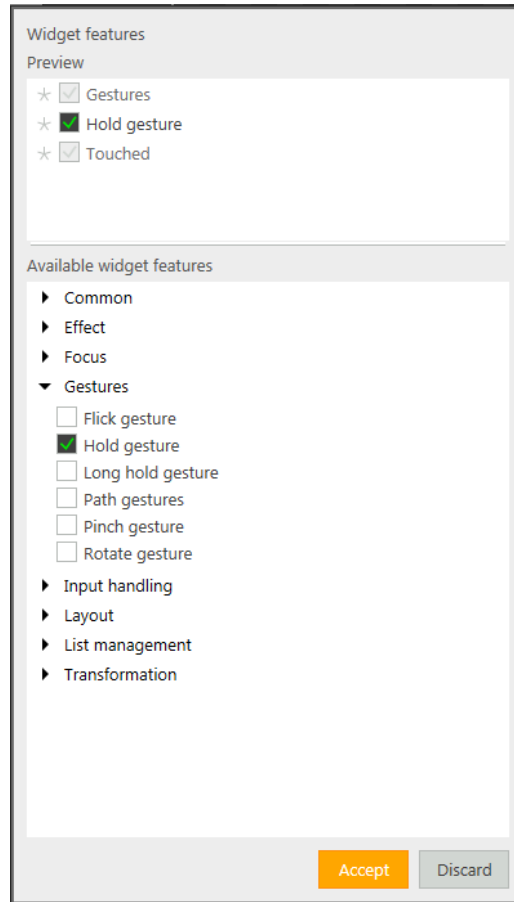


그림 8.7. 위젯 피쳐 대화 상자

### 단계 3

미리보기에서 제거할 위젯 피쳐의 선택을 취소합니다.

수락을 클릭합니다.

관련 위젯 피쳐 속성이 속성 구성 요소에서 제거됩니다.

### [참고]



#### 종속성이 있는 위젯 피쳐 제거

종속성으로 인해 자동으로 추가된 위젯 피쳐는 자동으로 삭제되지 않습니다. 이 위젯 피쳐는 바로 제거할 수 없습니다. 즉, 하위 위젯 피쳐를 지우기 전에 상위 위젯 피쳐를 지워야 합니다.

## 8.4. EB GUIDE 모델에 언어 추가

런타임 동안 언어 지원을 활성화하려면 EB GUIDE 모델에 언어를 추가합니다.

## 8.4.1. 언어 추가

### [참고]



### 사용할 수 있는 스킨 지원 없음


데이터폴 항목에 대한 언어 지원을 정의한 경우 동일한 항목에 스킨 지원을 추가할 수 없습니다.



### 언어 추가

목록의 첫 번째 언어는 항상 기본 언어이며 삭제할 수 없습니다. 언어를 추가하는 경우 표준 언어 설정이 초기 값으로 사용됩니다.

#### 단계 1

를 클릭합니다.

프로젝트 센터가 열립니다.

#### 단계 2

내비게이션 영역에서 **구성 > 언어**를 클릭합니다.

사용 가능한 언어가 표시됩니다.

#### 단계 3

콘텐츠 영역에서 **추가**를 클릭합니다.

언어가 테이블에 추가됩니다.

#### 단계 4

언어의 이름을 입력합니다.

#### 단계 5

**언어** 드롭다운 목록 상자에서 언어를 선택합니다.

#### 단계 6

**국가** 드롭다운 목록 상자에서 국가를 선택합니다.

언어가 추가되었습니다.

런타임 중에 언어를 변경하는 방법에 대한 지침은 [섹션 11.6, "자습서: 데이터폴 항목에 언어별로 다른 텍스트 추가"](#) 항목을 참조하십시오.



## 8.4.2. 언어 삭제




### 언어 삭제

선수학습:

- 언어 2개 이상이 EB GUIDE 모델에 추가되어 있어야 합니다.

#### 단계 1

를 클릭합니다.

프로젝트 센터가 열립니다.

#### 단계 2

내비게이션 영역에서 **구성 > 언어**를 클릭합니다.

사용 가능한 언어가 표시됩니다.

#### 단계 3

콘텐츠 영역에서 언어를 선택합니다.

#### 단계 4

콘텐츠 영역에서 **삭제**를 클릭합니다.

언어가 테이블에서 삭제됩니다.

## 8.5. 스킨 지원으로 작업

스킨 지원을 통해 사용자 모델에 다른 데이터폴 값을 지정할 수 있습니다. 이 방법으로 야간 및 주간 모드 등의 예로 동일한 모델에 대해 다른 모양을 정의할 수 있습니다.

스킨에 대한 자세한 내용은 [섹션 6.18. "스킨"](#) 항목을 참조하십시오.

## 8.5.1. EB GUIDE 모델에 스킨 추가

### [참고]




### 사용할 수 있는 언어 지원 없음

데이터풀 항목에 대한 스킨 지원을 정의한 경우 동일한 항목에 언어 지원을 추가할 수 없습니다.



### EB GUIDE 모델에 스킨 추가

#### 단계 1

를 클릭합니다.

프로젝트 센터가 열립니다.

#### 단계 2

내비게이션 영역에서 **구성 > 스킨**을 클릭합니다.

기본적으로 각 모델에 표준 스킨이 추가됩니다.

#### 단계 3

콘텐츠 영역에서 **추가**를 클릭합니다.

스킨이 테이블에 추가됩니다.

#### 단계 4

스킨의 이름을 입력합니다.

새 스킨이 EB GUIDE 모델에 추가됩니다. 프로젝트 편집자에서, 명령 영역의 **스킨** 드롭다운 목록 상자에서 새 스킨을 선택할 수 있습니다.

## 8.5.2. 데이터풀 항목에 스킨 지원 추가



### 데이터풀 항목에 스킨 지원 추가

다른 데이터풀 값을 정의하여 EB GUIDE 모델에 다양한 모양을 정의하려면, 먼저 데이터풀 항목에 스킨 지원을 추가해야 합니다.


선수학습:

- EB GUIDE 모델에 데이터풀 항목이 포함되어 있어야 합니다.
- 스킨이 모델에 추가됩니다.

단계 1

프로젝트 편집자에서 **데이터폴** 구성 요소로 이동합니다.


단계 2

데이터폴 항목의 **값** 속성 옆에 있는  버튼을 클릭합니다.

메뉴가 확장됩니다.

단계 3

메뉴에서 **스킨 지원 추가**를 클릭합니다.

대화 상자가 닫힙니다. **값** 속성 옆에  버튼이 표시됩니다. 이는 이 데이터폴 항목에 스킨 지원이 추가되었으며 이제 각 스킨에 다른 값을 정의할 수 있음을 나타냅니다.

단계 4

데이터폴 항목에 다른 값을 정의하려면 **데이터폴** 구성 요소의 데이터폴을 선택합니다.

**속성** 구성 요소에는 EB GUIDE 모델에서 사용할 수 있는 모든 스킨이 있는 테이블이 표시됩니다.

단계 5

테이블의 각 스킨에 값을 정의합니다.

### 8.5.3. 스킨 간 전환



#### 스킨 간 전환

선수학습:

- EB GUIDE 모델에 데이터폴 항목이 포함되어 있어야 합니다.
- 스킨이 모델에 추가됩니다.

단계 1

프로젝트 편집자에서 명령 영역으로 이동합니다.

단계 2

드롭다운 목록 상자에서 스킨을 선택합니다.

콘텐츠 영역에는 해당 스킨에 유효한 데이터폴 값이 있는 모델이 표시됩니다. 또한 시뮬레이션 모드에는 특정 스킨 값이 있는 모델이 표시됩니다.

## 8.5.4. 스킨 삭제




### 스킨 삭제

선수학습:

- 스킨이 모델에 추가됩니다.

#### 단계 1

를 클릭합니다.

프로젝트 센터가 열립니다.

#### 단계 2

내비게이션 영역에서 **구성 > 스킨**을 클릭합니다.

현재 프로젝트의 모든 스킨이 나열됩니다.

#### 단계 3

삭제할 스킨을 선택하고 **삭제**를 클릭합니다.

스킨이 테이블에서 삭제됩니다.

## 8.6. 관점 전환 애니메이션 생성

### 8.6.1. 진입 애니메이션 추가



### 진입 애니메이션 추가

다음 지침에서는 보기 상태에 진입 애니메이션을 추가하는 프로세스를 안내합니다. 또한 이 지침은 종료 애니메이션, 애니메이션 팝업 켜기, 애니메이션 팝업 끄기에도 적용됩니다. 자세한 내용은 [섹션 6.2.2. "보기 전의 애니메이션"](#) 및 [섹션 12.12.1. "보기"](#) 항목을 참조하십시오.

선수학습:

- 보기 상태와 함께 보기 하나가 EB GUIDE 모델에 추가됩니다.
- VTA** 구성 요소가 열립니다.

#### 단계 1

**VTA** 구성 요소에서 **+**를 클릭합니다.

단계 2

컨텍스트 메뉴에서 **진입 애니메이션**을 선택합니다.

**애니메이션** 편집기가 열립니다.

**속성** 구성 요소에 추가된 진입 애니메이션 속성이 표시됩니다.

단계 3

종속 보기에 이용 가능한 모든 위젯 속성이 애니메이션됩니다.

**애니메이션** 편집기에서 진입 애니메이션을 정의합니다.

## 8.6.2. 변경 애니메이션 추가



### 변경 애니메이션 추가

다음 지침은 보기 상태 또는 보기 템플릿에 변경 애니메이션을 추가하는 프로세스를 안내합니다.

선수학습:

- **VTA** 구성 요소가 열립니다.
- **메인** 상태 머신에 2개의 보기 상태가 포함되어 있습니다.

단계 1

**메인** 상태 머신에서 View state 1을 선택합니다.

단계 2

**VTA** 구성 요소에서 **+**를 클릭합니다.

단계 3

컨텍스트 메뉴에서 **변경 애니메이션**을 선택합니다.

대화 상자가 열립니다.

단계 4

View 2를 선택합니다.

단계 5

**수락**을 클릭합니다.


목적지 보기의 이름은 변경 애니메이션 이름 옆에 표시됩니다.

**애니메이션** 편집기가 열립니다.

**속성** 구성 요소에 추가된 변경 애니메이션 속성이 표시됩니다.

시작 보기는 프로젝트 편집자에 표시됩니다.

#### 단계 6

목적지 보기를 편집하려면 **VTA** 구성 요소에서  를 클릭합니다.

#### 단계 7

애니메이션 속성을 추가하려면 **애니메이션** 편집자에서 **+**를 눌러 각 범주를 추가합니다.

종속 보기에 이용 가능한 모든 위젯 속성이 애니메이션됩니다.

### 8.6.3. 애니메이션 재배열



#### 애니메이션 재배열

다음 지침은 보기 상태 또는 보기 템플릿의 애니메이션을 재배열하는 프로세스를 안내합니다. 또한 이 지침은 종료 애니메이션, 변경 애니메이션, 애니메이션 팝업 켜기, 애니메이션 팝업 끄기에도 적용됩니다.

선수학습:

- **VTA** 구성 요소가 열립니다.
- 여러 진입 애니메이션을 추가했습니다.

#### 단계 1

**VTA** 구성 요소에서 가장 먼저 시작하려는 진입 애니메이션 옆의 우선 순위 텍스트 상자를 선택합니다.

#### 단계 2

진입 애니메이션의 값을 0으로 변경합니다.

진입 애니메이션의 순서가 변경됩니다. 편집된 진입 애니메이션이 가장 먼저 시작하며 이어지는 모든 항목은 1씩 증가합니다.

값 0은 해당 애니메이션이 처음에 재생된다는 의미입니다. 후속 값은 해당 애니메이션이 후속 순서로 재생됨을 의미합니다.

## 8.7. 위젯 다시 사용

템플릿에 대한 자세한 내용은 [섹션 6.21.4, "위젯 템플릿"](#) 항목을 참조하십시오.

### 8.7.1. 템플릿 추가



## 템플릿 추가

### 단계 1

**내비게이션** 구성 요소에서 **템플릿**으로 이동한 다음 **+**를 클릭합니다.

메뉴가 확장됩니다.

### 단계 2

메뉴에서 템플릿의 유형을 클릭합니다.

선택한 유형의 새 템플릿이 추가됩니다. 콘텐츠 영역에 템플릿이 표시됩니다.

### 단계 3

템플릿의 이름을 바꿉니다.

### 단계 4

**속성** 구성 요소에서 템플릿의 속성을 편집하고 템플릿 인터페이스를 정의합니다.

#### [작은 정보] 템플릿의 템플릿



템플릿의 유형은 기존 템플릿일 수 있습니다. 따라서 EB GUIDE를 통해 템플릿에서 템플릿을 만들 수 있습니다.

#### [작은 정보] 템플릿 복사 및 찾기



컨텍스트 메뉴 또는 Ctrl+C 및 Ctrl+V를 사용하여 기존 템플릿을 복사하고 붙여 넣을 수도 있습니다.

EB GUIDE 모델 내의 특정 템플릿을 찾으려면 검색 상자에 템플릿 이름을 입력하거나 CtrlF를 사용합니다. 한 템플릿으로 건너 뛰려면 결과 목록에서 더블 클릭합니다.

## 8.7.2. 템플릿 인터페이스 정의



## 템플릿 인터페이스 정의

선수학습:

- EB GUIDE 모델에 템플릿이 포함되어 있어야 합니다.

### 단계 1

템플릿을 선택합니다.

단계 2

템플릿 인터페이스에 속성을 추가하려면 속성 구성 요소에서 속성 옆의 버튼을 클릭합니다. 메뉴에서 **템플릿 인터페이스에 추가**를 클릭합니다.

속성 옆에 아이콘이 표시됩니다.

단계 3

템플릿 인터페이스에서 속성을 제거하려면 속성 옆의 버튼을 클릭합니다. 메뉴에서 **템플릿 인터페이스에서 제거**를 클릭합니다.

속성 옆에 아이콘이 더 이상 표시되지 않습니다.

[참고]



**인스턴스화기 템플릿**

인스턴스화기의 템플릿에서는 템플릿 인터페이스에 인스턴스화기 하위 위젯의 속성을 추가할 수 없습니다.

### 8.7.3. 템플릿 사용



#### 템플릿 사용

선수학습:

- 콘텐츠 영역에 보기가 표시되어 있어야 합니다.
- **도구 상자**에서 위젯 템플릿을 사용할 수 있어야 합니다.
- 위젯 템플릿의 템플릿 인터페이스에 속성이 하나 이상 있습니다.

단계 1

위젯 템플릿을 **도구 상자**에서 보기로 끕니다.

템플릿의 인스턴스가 보기에 추가됩니다. 템플릿 인터페이스에 속하는 속성이 **속성** 구성 요소에 표시됩니다.

[작은 정보]

**템플릿 인터페이스 정의**



**속성** 구성 요소에 템플릿 인스턴스의 속성이 표시되지 않는 경우에는 템플릿 인터페이스에 속성이 추가되지 않은 것입니다. 속성이 표시되도록 하려면 템플릿 인터페이스를 정의합니다.

단계 2

**속성** 구성 요소에서 템플릿 인스턴스의 속성을 편집합니다.

속성을 편집하고 나면 버튼이 버튼으로 변경됩니다.



단계 3

속성 값을 템플릿의 값으로 다시 설정하려면 속성 옆의 ■ 버튼을 클릭합니다. 메뉴에서 **템플릿 값으로 다시 설정**을 클릭합니다.

## 8.7.4. 템플릿 삭제



### 템플릿 삭제

단계 1

**내비게이션** 구성 요소에서 보기 템플릿을 오른쪽 마우스 클릭합니다.

단계 2

컨텍스트 메뉴에서 **삭제**를 클릭합니다.

템플릿이 삭제됩니다.

## 9. 데이터 처리

### [참고]



#### 기본 창 레이아웃

이 사용자 설명서의 모든 지침 및 스크린샷이 기본 창 레이아웃을 사용합니다. 지침을 따르는 경우 EB GUIDE Studio 창을 기본 레이아웃으로 설정하는 것이 좋습니다. **레이아웃 > 기본 레이아웃**으로 다시 설정을 선택하여 설정할 수 있습니다.

### 9.1. 이벤트 추가



#### 이벤트 추가

##### 단계 1

이벤트 구성 요소에서 **+**를 클릭합니다.

이벤트가 테이블에 추가됩니다.

##### 단계 2

이벤트의 이름을 바꿉니다.

### [작은 정보]



#### 이벤트 복사 및 찾기

또는 컨텍스트 메뉴 또는 **CtrlC** 및 **CtrlV**를 이용해 기존 이벤트를 복사해서 붙여 넣을 수도 있습니다. 중복을 방지하려면 붙여 넣은 이벤트에 복사한 이벤트와 다른 이벤트 ID를 설정합니다.

EB GUIDE 모델 내의 특정 이벤트를 찾으려면 검색 상자에 이벤트 이름을 입력하거나 **CtrlF**를 사용합니다. 한 이벤트로 건너 뛰려면 결과 목록에서 더블 클릭합니다.

### 9.2. 이벤트에 매개변수 추가



#### 이벤트에 매개변수 추가

선수학습:

- EB GUIDE 모델에 이벤트가 추가됩니다.

단계 1

**이벤트** 구성 요소에서 이벤트를 클릭합니다.

단계 2

이벤트 테이블에서 이벤트 옆의 **+**을 클릭합니다.

단계 3

드롭다운 목록 상자에서 매개변수의 유형을 선택합니다.

선택한 유형의 매개변수가 이벤트에 추가됩니다.

단계 4

매개변수의 이름을 바꿉니다.

## 9.3. 이벤트 주소 지정

이벤트 ID와 이벤트 그룹 ID를 사용하여 이벤트의 주소를 지정합니다. EB GUIDE TF에서는 ID를 사용하여 런타임에 이벤트를 보내고 받습니다.



### 이벤트 그룹 추가

단계 1

**IS**를 클릭합니다.

프로젝트 센터가 열립니다.

단계 2

내비게이션 영역에서 **구성 > 이벤트 그룹**을 클릭합니다.

단계 3

콘텐츠 영역에서 **추가**를 클릭합니다.

이벤트 그룹이 테이블에 추가됩니다.

단계 4

이벤트 그룹의 이름을 바꿉니다.

단계 5

이벤트 그룹 ID를 변경하려면 **ID**를 더블 클릭하고 숫자를 입력합니다.



### EB GUIDE TF에 대해 이벤트 주소 지정

선수학습:

- 이벤트 그룹이 추가되어 있어야 합니다.
- EB GUIDE 모델에 이벤트가 추가됩니다.

단계 1

**이벤트** 구성 요소에서 이벤트를 클릭합니다.

**속성** 구성 요소에 선택한 이벤트의 속성이 표시됩니다.

단계 2

Event ID 텍스트 상자에 ID를 입력합니다.

단계 3

**이벤트** 구성 요소로 이동하고 Group 드롭다운 목록 상자에서 이벤트 그룹을 선택합니다.

## 9.4. 이벤트 삭제



### 이벤트 삭제

선수학습:

- EB GUIDE 모델에 이벤트가 추가됩니다.

단계 1

**이벤트** 구성 요소에서 이벤트를 오른쪽 마우스 클릭합니다.

단계 2

컨텍스트 메뉴에서 **삭제**를 클릭합니다.

이벤트가 삭제됩니다.

## 9.5. 데이터폴 항목 추가



### 데이터폴 항목 추가

단계 1

**데이터폴** 구성 요소에서 **+**를 클릭합니다.

메뉴가 확장됩니다.

단계 2

메뉴에서 데이터폴 항목의 유형을 클릭합니다.

선택한 유형의 새 데이터폴 항목이 추가됩니다. 데이터폴 항목이 내부용으로 사용 가능하도록 준비됩니다.

단계 3

데이터폴 항목의 이름을 바꿉니다.

### [작은 정보] 데이터풀 항목 복사 및 찾기



컨텍스트 메뉴 또는 Ctrl+C 및 Ctrl+V를 사용하여 기존 데이터풀 항목을 복사하고 붙여 넣을 수도 있습니다.

EB GUIDE 모델 내의 특정 데이터풀 항목을 찾으려면 검색 상자에 데이터풀 항목 이름을 입력하거나 Ctrl+F를 사용합니다. 한 데이터풀 항목으로 건너 뛰려면 결과 목록에서 더블 클릭합니다.

## 9.6. 목록 유형의 데이터풀 항목 편집



### 목록 유형의 데이터풀 항목 편집

선수학습:

- 목록 유형의 데이터풀 항목이 추가되어 있어야 합니다.

#### 단계 1

데이터풀 구성 요소에서 목록 유형의 데이터풀 항목을 클릭합니다.

#### 단계 2

Value 열에서  을 클릭합니다.

편집기가 열립니다.

#### 단계 3

목록 데이터풀 항목에 항목을 추가하려면 **추가**를 클릭합니다.

새 항목이 테이블에 추가됩니다.

#### 단계 4

Value 텍스트 상자에 새 항목의 값을 입력하거나 콤보 박스에서 값을 선택합니다.

#### 단계 5

3~4단계를 반복하여 목록에 항목을 더 추가합니다.

#### 단계 6

**수락**을 클릭합니다.

목록의 콘텐츠가 Value 열에 표시됩니다.

## 9.7. 속성을 스크립팅된 값으로 변환




### 속성을 스크립팅된 값으로 변환

데이터폴 항목과 위젯의 속성을 스크립팅된 값으로 변환했다가 다시 일반 값으로 변환할 수 있습니다. 다음 지침에서는 데이터폴 항목 값을 변환하는 절차를 설명합니다. 위젯 속성의 경우에도 절차는 동일합니다.

선수학습:

- 데이터폴 항목이 추가되어 있어야 합니다.
- 데이터폴 항목은 언어에 따라 달라지지 않습니다.
- 데이터폴 항목은 연결되어 있지 않습니다.

#### 단계 1

**데이터폴** 구성 요소에서 데이터폴 항목을 클릭하고  버튼을 클릭합니다.

메뉴가 확장됩니다.

#### 단계 2

메뉴에서 **스크립트로 변환**을 클릭합니다.

데이터폴 항목이 스크립팅된 값으로 변환됩니다.

#### 단계 3


Value 열에서 **편집**을 클릭합니다.

콘텐츠 영역에 EB GUIDE 스크립트 편집기가 열립니다.

#### 단계 4

EB GUIDE 스크립트를 편집합니다.

#### 단계 5

데이터폴 항목을 다시 일반 값으로 변환하려면  버튼을 클릭합니다.

메뉴가 확장됩니다.

#### 단계 6

메뉴에서 **일반 값으로 변환**을 클릭합니다.

데이터폴 항목이 일반 값으로 변환됩니다.

## 9.8. 외부 통신 설정

예를 들어 EB GUIDE 모델과 응용 프로그램 간의 통신과 같은 외부 통신을 설정하려면 EB GUIDE 모델에 통신 컨텍스트를 추가합니다.



### 통신 컨텍스트 추가

통신 컨텍스트를 사용하면 통신 채널을 설정할 수 있습니다.

#### 단계 1

를 클릭합니다.

프로젝트 센터가 열립니다.

#### 단계 2

내비게이션 영역에서 **구성 > 통신 컨텍스트**를 클릭합니다.

#### 단계 3

콘텐츠 영역에서 **추가**를 클릭합니다.

통신 컨텍스트가 테이블에 추가됩니다.

#### 단계 4

통신 컨텍스트의 이름을 예를 들어 Media 등으로 바꿉니다.

#### 단계 5

자체 스레드에서 통신 컨텍스트를 실행하려면 **자체 스레드 사용**을 선택합니다.

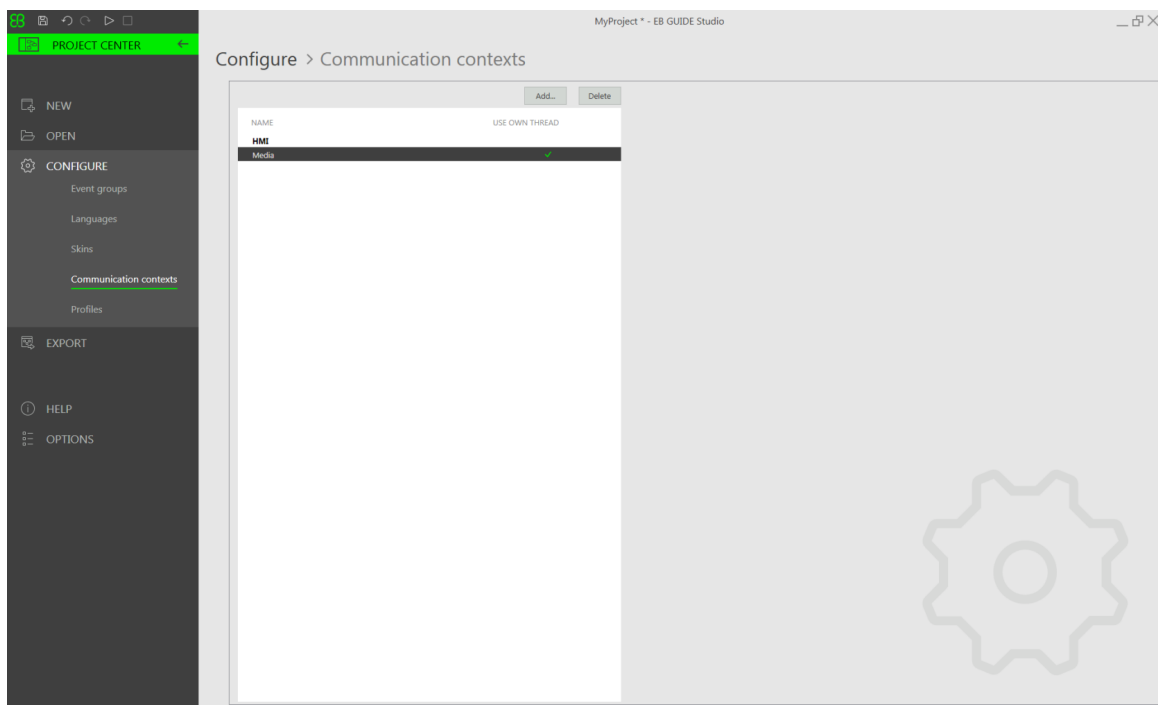


그림 9.1. 통신 컨텍스트 Media

## 9.9. 데이터풀 항목을 서로 연결



## 데이터폴 항목을 서로 연결

선수학습:

- 데이터폴 항목이 추가되어 있어야 합니다.
- 데이터폴 항목은 언어에 따라 달라지지 않습니다.
- 데이터폴 항목이 스크립팅된 값으로 변환되지 않습니다.

단계 1

**데이터폴** 구성 요소에서 데이터폴 항목을 클릭합니다.

단계 2

■ 버튼을 클릭합니다.

메뉴가 확장됩니다.

단계 3

메뉴에서 **데이터폴 항목에 대한 링크 추가**를 클릭합니다.

대화 상자가 열립니다.

단계 4

새 데이터폴 항목을 추가하려면 텍스트 상자에 이름을 입력합니다.

단계 5

**데이터폴 항목 추가**를 클릭합니다.

단계 6

수락을 클릭합니다.



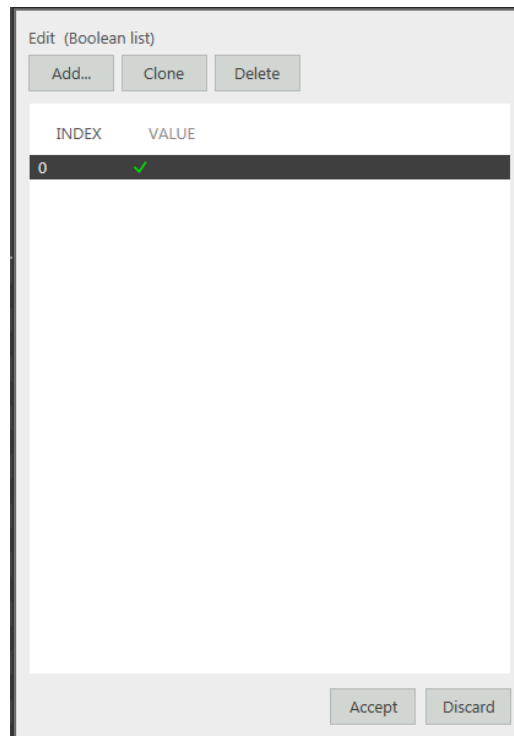



그림 9.2. 데이터폴 항목을 서로 연결

대화 상자가 닫힙니다. Value 속성 옆의  버튼이 표시됩니다. 이 버튼은 Value 속성이 데이터폴 항목에 연결되었음을 나타냅니다. 데이터폴 항목 중 하나의 값이 변경되면 다른 항목의 값도 변경됩니다.

## 9.10. 데이터폴 항목 삭제



### 데이터폴 항목 삭제

선수학습:

- 데이터폴 항목이 추가되어 있어야 합니다.

#### 단계 1

**데이터폴** 구성 요소에서 데이터폴 항목을 오른쪽 마우스 클릭합니다.

#### 단계 2

컨텍스트 메뉴에서 **삭제**를 클릭합니다.

데이터폴 항목이 삭제됩니다.

## 10. 프로젝트 처리

### [참고]



### 기본 창 레이아웃


이 사용자 설명서의 모든 지침 및 스크린샷이 기본 창 레이아웃을 사용합니다. 지침을 따르는 경우 EB GUIDE Studio 창을 기본 레이아웃으로 설정하는 것이 좋습니다. **레이아웃 > 기본 레이아웃**으로 다시 설정을 선택하여 설정할 수 있습니다.

### 10.1. 프로젝트 만들기



#### 프로젝트 만들기

##### 단계 1

를 클릭합니다.

프로젝트 센터가 열립니다.

##### 단계 2

내비게이션 영역에서 **새로 만들기**를 클릭합니다.

##### 단계 3

프로젝트 이름을 입력하고 위치를 선택합니다.

##### 단계 4

**만들기**를 클릭합니다.

프로젝트가 만들어집니다. 프로젝트 편집자가 열리고 새 프로젝트가 표시됩니다.

### 10.2. 프로젝트 열기

#### 10.2.1. 파일 탐색기에서 프로젝트 열기



#### 파일 탐색기에서 프로젝트 열기

선수학습:

- EB GUIDE Studio 프로젝트를 만들어야 합니다.

단계 1

파일 탐색기를 연 다음 열리는 EB GUIDE Studio 프로젝트 파일을 선택합니다. EB GUIDE Studio 프로젝트 파일의 파일 확장명은 .ebguide입니다.

단계 2

EB GUIDE Studio 프로젝트 파일을 더블 클릭합니다.

프로젝트가 EB GUIDE Studio에서 열립니다.

## 10.2.2. EB GUIDE Studio 내에서 프로젝트 열기




### EB GUIDE Studio 내에서 프로젝트 열기

선수학습:

- EB GUIDE Studio 프로젝트를 만들어야 합니다.

단계 1

를 클릭합니다.

프로젝트 센터가 열립니다.

단계 2

내비게이션 영역에서 **열기** 탭을 클릭합니다.

단계 3

**최근 프로젝트** 아래에 나열되어 있는 프로젝트를 선택하거나, **찾아보기**를 클릭하고 열리는 EB GUIDE Studio 프로젝트 파일을 선택합니다. EB GUIDE Studio 프로젝트 파일의 파일 확장명은 .ebguide입니다.

프로젝트가 EB GUIDE Studio에서 열립니다.

## 10.3. 모델 요소, 데이터폴 항목 및 이벤트 이름 바꾸기



### 모델 요소 이름 바꾸기

다음 지침에서는 상태, 위젯, 전이 및 애니메이션과 같은 모델 요소의 이름 바꾸기 프로세스를 안내합니다.

선수학습:

- 모델 요소가 EB GUIDE 모델에 추가됩니다.

단계 1

**내비게이션** 구성 요소에서 모델 요소를 오른쪽 마우스 클릭합니다.

컨텍스트 메뉴를 엽니다.

단계 2

컨텍스트 메뉴에서 **이름 바꾸기**를 클릭하고 Enter 키를 눌러 확인합니다.

변경된 이름이 표시됩니다.



데이터폴 항목 및 이벤트 이름 바꾸기

다음 지침에서는 데이터폴 항목의 이름 바꾸기 프로세스를 안내합니다.

**[참고]**



**이벤트 이름 바꾸기**

**이벤트** 구성 요소의 이벤트 이름 바꾸기에도 동일한 절차가 적용됩니다.

선수학습:

- 데이터폴 항목이 EB GUIDE 모델에 추가됩니다.

단계 1

**데이터폴** 구성 요소에서 데이터폴 항목을 오른쪽 마우스 클릭합니다.

컨텍스트 메뉴를 엽니다.

단계 2

컨텍스트 메뉴에서 다음 중 하나를 클릭합니다.

- ▶ **이름 바꾸기**를 클릭하여 선택한 데이터폴 항목의 이름만 바꿉니다.
- ▶ **전역 이름 바꾸기**를 클릭하여 선택한 데이터폴 항목과 EB GUIDE 모델의 항목(예: EB GUIDE 스크립트)의 이름을 바꿉니다.

## 10.4. EB GUIDE 모델 유효성 검사 및 시뮬레이션

EB GUIDE 모델을 타깃 기기로 내보내기 전에 PC에서 오류를 해결하고 모델을 시뮬레이션해야 합니다.

## 10.4.1. EB GUIDE 모델 유효성 검사

### 10.4.1.1. EB GUIDE Studio를 사용하여 EB GUIDE 모델 유효성 검사



#### EB GUIDE Studio를 사용하여 EB GUIDE 모델 유효성 검사

**문제** 구성 요소에서 EB GUIDE에 다음이 표시됩니다.

- ▶ ❌ 오류
- ▶ ⚠️ 경고

#### 단계 1

**문제** 구성 요소에서 를 클릭합니다.

오류와 경고의 수가 표시됩니다.

#### 단계 2

**문제**를 클릭하여 **문제** 구성 요소를 확장합니다.

오류와 경고의 목록이 표시됩니다.

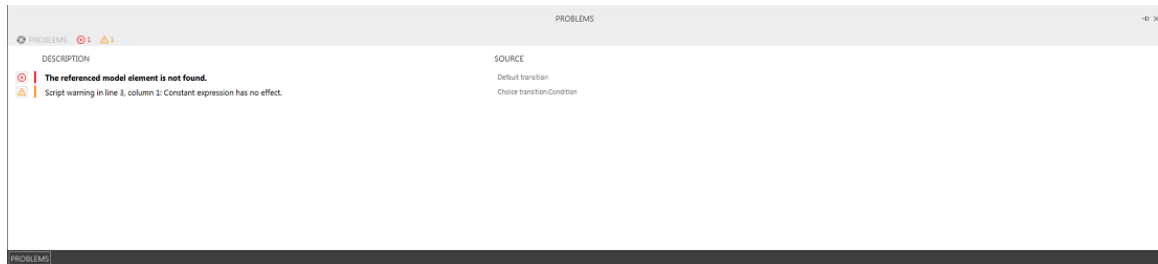


그림 10.1. 문제 구성 요소

#### 단계 3

문제가 발생한 위치로 이동하려면 해당 줄을 두 번 클릭합니다.

문제의 원인이 된 요소가 강조 표시됩니다.

#### 단계 4

문제를 해결합니다.

#### 단계 5

를 클릭합니다.

해결한 문제는 더 이상 **문제** 구성 요소에 나열되지 않습니다.

단계 6

**문제** 구성 요소를 축소하려면 **문제**를 다시 클릭합니다.

오류가 없는 경우 EB GUIDE 모델은 유효합니다. 약간의 경고가 있는 경우에도 EB GUIDE 모델은 유효합니다.

### 10.4.1.2. 명령줄을 사용하여 EB GUIDE 모델 유효성 검사



#### 명령줄을 사용하여 EB GUIDE 모델 유효성 검사

단계 1

명령줄을 이용하여 \$GUIDE\_INSTALL\_PATH/Studio로 이동합니다.

단계 2

Studio.Console.exe -c "<logfile dir>/log.txt" -o "\$GUIDE\_PROJECT\_PATH/project\_name.ebguide"를 입력합니다.

EB GUIDE 모델의 유효성 검사가 완료되었으며 결과는 지정된 위치 <logfile dir>에 있는 로그 파일에 저장됩니다.

### 10.4.2. 시뮬레이션 시작 및 중지



#### 시뮬레이션 시작 및 중지

단계 1

시뮬레이션을 시작하려면 명령 영역에서 ▶을 클릭합니다.

시뮬레이션 및 EB GUIDE Monitor가 시작됩니다. 시뮬레이션은 자체 구성으로 시작됩니다.

구성을 변경하려면 프로젝트 센터로 이동한 다음 **구성 > 프로필**을 클릭합니다.

단계 2

시뮬레이션을 중지하려면 명령 영역에서 □을 클릭합니다.

시뮬레이션 및 EB GUIDE Monitor가 중지됩니다.

## 10.5. EB GUIDE 모델 내보내기

## 10.5.1. EB GUIDE Studio를 사용하여 EB GUIDE 모델 내보내기




### EB GUIDE Studio를 사용하여 EB GUIDE 모델 내보내기

EB GUIDE 모델을 타깃 기기에 복사하려면 EB GUIDE Studio에서 해당 모델을 내보내야 합니다.

EB GUIDE 모델을 내보낼 때는 항상 프로필을 선택합니다.

#### 단계 1

를 클릭합니다.

프로젝트 센터가 열립니다.

#### 단계 2

내비게이션 영역에서 **내보내기** 탭을 클릭합니다.

#### 단계 3

Profile 드롭다운 목록 상자에서 프로필을 선택합니다.

#### 단계 4

**찾아보기**를 클릭하고 이진 파일을 내보낼 위치를 선택합니다.

#### 단계 5

**폴더 선택**을 클릭합니다.

#### 단계 6

**내보내기**를 클릭합니다.

선택한 위치로 이진 파일이 내보내집니다.

## 10.5.2. 명령줄을 사용하여 EB GUIDE 모델 내보내기



### 명령줄을 사용하여 EB GUIDE 모델 내보내기

선수학습:

- EB GUIDE 모델에서 오류와 경고가 발생하지 않아야 합니다.

#### 단계 1

명령줄을 이용하여 \$GUIDE\_INSTALL\_PATH/Studio로 이동합니다.

#### 단계 2

Studio.Console.exe -e <destination dir> -p <profile> -o "\$GUIDE\_PROJECT\_PATH/project\_name.-ebguide"를 입력합니다.


지정한 프로파일 <profile>와 함께 선택한 위치 <destination dir>로 EB GUIDE 모델이 내보내집니다.

## 10.6. EB GUIDE Studio의 표시 언어 변경



### EB GUIDE Studio의 표시 언어 변경

#### 단계 1

를 클릭합니다.

프로젝트 센터가 열립니다.

#### 단계 2

내비게이션 영역에서 **옵션** 탭을 클릭합니다.

#### 단계 3

**표시 언어** 드롭다운 목록 상자에서 언어를 선택합니다.

#### 단계 4

EB GUIDE Studio를 다시 시작합니다.

다시 시작하면 그래픽 사용자 인터페이스가 선택한 언어로 표시됩니다.

## 10.7. 프로파일 구성

EB GUIDE Studio에서는 EB GUIDE 모델에 대한 다른 프로파일을 만들 수 있습니다.

다음을 수행하는 경우 프로파일을 사용합니다.

- ▶ 메시지 전송
- ▶ 내부 및 사용자 정의 라이브러리를 구성하여 로드
- ▶ 장면 구성
- ▶ 렌더러 구성

기본 프로파일은 두 가지로, **편집** 및 **시뮬레이션**이 있습니다.

### 10.7.1. 프로파일 복제





### 프로필 복제

선수학습:

- EB GUIDE Studio 프로젝트가 열립니다.
- 프로젝트 센터가 표시됩니다.

#### 단계 1

내비게이션 영역에서 **구성 > 프로필**을 클릭합니다.

#### 단계 2

콘텐츠 영역에서 **시뮬레이션** 프로필을 선택합니다.

#### 단계 3

**복제**를 클릭합니다.

프로필이 테이블에 추가됩니다. 이 프로필이 기본 프로필 **시뮬레이션**의 복제본입니다.

#### 단계 4

테이블을 두 번 클릭하고 프로필 이름으로 MySimulation으로 바꿉니다.

#### 단계 5

**시뮬레이션용으로 사용**라디오 버튼을 선택합니다.

MySimulation 프로필이 PC에서 시뮬레이션용으로 사용됩니다.

## 10.7.2. 라이브러리 추가

기본 제공되는 EB GUIDE TF은(는) 예를 들어 Windows 10, Linux 또는 QNX 등의 공유 라이브러리를 지원하는 운영 체제에서 실행됩니다. EB GUIDE TF은(는) 기본 고객 프로젝트에 최적화되는 실행 파일과 라이브러리 집합으로 구성되어 있습니다.

다음 작업에는 EB GUIDE 모델과 상호 작용하고 추가 기능을 제공하는 사용자 정의 라이브러리를 추가하는 방법이 나와 있습니다.



### 라이브러리 추가: 플랫폼

이 작업에서는 라이브러리 또는 현재 플랫폼의 모든 EB GUIDE 모델에서 사용할 수 있는 여러 라이브러리를 추가하는 방법이 나와 있습니다.

선수학습:

- EB GUIDE Studio 프로젝트가 열립니다.
- 프로젝트 센터가 표시됩니다.

- 내비게이션 영역에서 **구성 > 프로필** 탭이 선택됩니다.
- MySimulation 프로필이 추가됩니다.
- 라이브러리 MyLibraryA 및 MyLibraryB는 \$GTF\_INSTALL\_PATH/platform/<platform name>에서 사용할 수 있습니다.

단계 1

콘텐츠 영역에서 MySimulation 프로필을 선택합니다.

단계 2

**플랫폼** 탭을 클릭합니다.

단계 3

다음 코드를 입력합니다.

```
{  
  "gtf":  
  {  
    "core":  
    {  
      "pluginstoload": ["MyLibraryA", "MyLibraryB"]  
    }  
  }  
}
```

라이브러리 MyLibraryA 및 MyLibraryB를 시작 코드에 추가했습니다.

**[참고]**

**JSON 객체 표기법**



EB GUIDE Studio내에서 platform.json을 구성할 경우 JSON 객체 표기법을 사용하십시오.

예는 [섹션 12.8.1, "EB GUIDE Studio의 예제 platform.json"](#)를 참조하십시오.

JSON 형식에 대한 자세한 내용은 <http://www.json.org>를 참조하십시오.



라이브러리 추가: 모델

이 작업에서는 현재 EB GUIDE 모델에서만 사용할 수 있는 여러 라이브러리를 추가하는 방법이 나와 있습니다.

선수학습:

- EB GUIDE Studio 프로젝트가 열립니다.
- 프로젝트 센터가 표시됩니다.
- 내비게이션 영역에서 **구성 > 프로필** 탭이 선택됩니다.
- MySimulation 프로필이 추가됩니다.

- 라이브러리 MyLibraryA 및 MyLibraryB는 \$GUIDE\_PROJECT\_PATH/<project name>/resources에서 사용할 수 있습니다.

단계 1

콘텐츠 영역에서 MySimulation 프로필을 선택합니다.

단계 2

**모델** 탭을 클릭합니다.

단계 3

다음 코드를 입력합니다.

```
{  
  "gtf":  
  {  
    "model":  
    {  
      "pluginstoload": ["resources/MyLibraryA", "resources/MyLibraryB"]  
    }  
  }  
}
```

라이브러리 MyLibraryA 및 MyLibraryB를 시작 코드에 추가했습니다.

**[참고]**

**JSON 객체 표기법**



EB GUIDE Studio에서 model.json을 구성할 경우 JSON 객체 표기법을 사용하십시오.

예는 [섹션 12.6.1, "EB GUIDE Studio의 예제 model.json"](#)를 참조하십시오.

JSON 형식에 대한 자세한 내용은 <http://www.json.org>를 참조하십시오.

### 10.7.3. 장면 구성

EB GUIDE Studio에서는 모든 상태 기계에 대한 장면을 구성할 수 있습니다.

다음 이유 중 하나에 해당하는 경우 프로젝트에 상태 기계를 둘 이상 포함할 수 있습니다.

- ▶ 모델의 논리를 다른 상태 기계로 분리하는 경우
- ▶ 디스플레이 또는 계층을 둘 이상 사용하는 경우



## 장면 구성

선수학습:

- EB GUIDE Studio 프로젝트가 열립니다.
- 프로젝트 센터가 표시됩니다.
- 내비게이션 영역에서 **구성 > 프로필** 탭이 선택됩니다.

### 단계 1

콘텐츠 영역에서 **장면** 탭을 클릭합니다.

### 단계 2

**상태 머신** 드롭다운 목록 상자에서 예를 들어 **기본**과 같은 기본 디스플레이의 상태 머신을 선택합니다.

### 단계 3

PC 데스크톱에서 창의 초기 위치를 설정하려면 x 및 y 값을 입력합니다.

### 단계 4

**렌더러** 드롭다운 목록 상자에서 렌더러를 선택합니다.

### 단계 5

추가 속성을 조정합니다. 각 속성에 대한 자세한 내용은 [섹션 12.9. "장면"](#) 항목을 참조하십시오.

## 10.8. 언어별로 다른 텍스트 내보내기 및 가져오기

### 10.8.1. 언어별로 다른 텍스트 내보내기

#### [작은 정보] EB GUIDE 모델 유효성 검사



텍스트를 내보내고 가져오는 동안 오류 발생을 방지하려면 시작하기 전에 EB GUIDE 모델 유효성을 검사합니다.




## 언어별로 다른 텍스트 내보내기

사용자의 기본 설정 언어로 텍스트를 제공하려면 데이터폴 항목의 언어별로 다른 텍스트를 모두 내보낸 다음 번역자에게 전달합니다.

선수학습:

- String 또는 String list 유형의 데이터폴 항목이 추가됩니다.
- 데이터폴 항목에는 언어 지원이 포함되어 있어야 합니다. 언어별로 다른 텍스트를 추가하는 방법에 대한 자세한 내용은 [섹션 11.6. "자습서: 데이터폴 항목에 언어별로 다른 텍스트 추가"](#) 항목을 참조하십시오.
- 언어 2개 이상이 EB GUIDE 모델에 추가되어 있어야 합니다.
- EB GUIDE 모델에서 오류와 경고가 발생하지 않아야 합니다.

단계 1

를 클릭합니다.

프로젝트 센터가 열립니다.

단계 2

내비게이션 영역에서 **구성 > 언어**를 클릭합니다.

단계 3

콘텐츠 영역에서 번역해야 하는 대상 언어를 선택합니다.

여러 파일을 선택할 수 있습니다.

단계 4

**내보내기**를 클릭합니다.

대화 상자가 열립니다.

단계 5

파일을 내보낼 디렉터리를 선택합니다.

단계 6

**폴더 선택**을 클릭합니다.

결과: 내보내기가 시작됩니다. 파일이 선택한 디렉터리에 저장됩니다. 이 파일은 언어별로 다른 머리글자어로 표시되며 형식은 .xliff입니다. 그리고 원본 언어에 해당하는 값과 대상 언어에 해당하는 값이 파일에 포함되어 있습니다.

**[참고]**



**언어당 파일을 하나씩 내보냄**

프로젝트 센터에서 선택하는 모든 언어에 대해 개별 파일을 내보냅니다.

## 10.8.2. 언어별로 다른 텍스트 가져오기

### 10.8.2.1. EB GUIDE Studio를 사용하여 언어별로 다른 텍스트 가져오기




#### EB GUIDE Studio를 사용하여 언어별로 다른 텍스트 가져오기

선수학습:

- String 또는 String list 유형의 데이터폴 항목이 추가됩니다.
- 데이터폴 항목에는 언어 지원이 포함되어 있어야 합니다. 언어별로 다른 텍스트를 추가하는 방법에 대한 자세한 내용은 [섹션 11.6, "자습서: 데이터폴 항목에 언어별로 다른 텍스트 추가"](#) 항목을 참조하십시오.
- 언어 2개 이상이 EB GUIDE 모델에 추가되어 있어야 합니다.
- EB GUIDE 모델에서 오류와 경고가 발생하지 않아야 합니다.
- 번역된 .xliff 파일을 하나 이상 사용할 수 있어야 합니다.

단계 1

를 클릭합니다.

프로젝트 센터가 열립니다.

단계 2

내비게이션 영역에서 **구성 > 언어**를 클릭합니다.

단계 3

**가져오기**를 클릭합니다.

대화 상자가 열립니다.

단계 4

번역된 .xliff 파일이 저장되어 있는 디렉토리를 선택합니다.

단계 5

번역된 .xliff 파일을 선택합니다.

여러 파일을 선택할 수 있습니다.

단계 6

**열기**를 클릭합니다.

가져오기가 시작됩니다. 대화 상자가 열립니다.

단계 7

**닫기**를 클릭합니다.

### 10.8.2.2. 명령줄을 사용하여 언어별로 다른 텍스트 가져오기



명령줄을 사용하여 언어별로 다른 텍스트 가져오기

선수학습:

- 언어 2개 이상이 EB GUIDE 모델에 추가되어 있어야 합니다.
- EB GUIDE 모델에서 오류와 경고가 발생하지 않아야 합니다.
- 번역된 하나의 .xliff 언어 파일을 사용할 수 있습니다.

단계 1

명령줄을 이용하여 \$GUIDE\_INSTALL\_PATH/Studio로 이동합니다.

단계 2

Studio.Console.exe -l <language file> -o "\$GUIDE\_PROJECT\_PATH/project\_name.ebguide"를 입력합니다.

가져오기에 성공하면 EB GUIDE 모델이 저장됩니다. 가져오기에 실패하면 EB GUIDE 모델이 저장되지 않습니다. 두 경우 모두 로그 파일이 생성됩니다. 날짜와 타임스탬프가 로그 파일 이름에 추가됩니다.

## 10.9. EB GUIDE Monitor로 작업

### 10.9.1. EB GUIDE Monitor에서 이벤트를 실행합니다.



EB GUIDE Monitor에서 이벤트를 실행합니다.

선수학습:

- EB GUIDE 모델에 이벤트가 포함되어 있어야 합니다.
- EB GUIDE 모델의 시뮬레이션이 시작됩니다.
- EB GUIDE Monitor가 시작됩니다.

단계 1


EB GUIDE Monitor의 **이벤트** 구성 요소에서 **이벤트 검색** 검색 상자에서 실행할 이벤트를 검색합니다.

단계 2

이벤트를 클릭합니다.

이벤트가 목록에 추가됩니다.


단계 3

이벤트를 실행하려면 이벤트 옆의 **이벤트** 구성 요소에서  을 클릭합니다.

이벤트가 발생합니다. **로거** 구성 요소에 로그 메시지가 표시됩니다.

#### 단계 4


##### 단계 4.1

이벤트에 매개변수가 있는 경우 매개변수를 확장하려면  를 클릭합니다.

##### 단계 4.2

**값** 열에서 매개변수를 변경합니다.

##### 단계 4.3

이벤트를 발생하려면 이벤트 옆의  을 클릭합니다.

변경된 매개변수로 이벤트가 발생합니다. **로거** 구성 요소에 로그 메시지가 표시됩니다.

## 10.9.2. EB GUIDE Monitor로 데이터폴 항목의 값 변경



### EB GUIDE Monitor에서 데이터폴 항목의 값 변경

선수학습:

- EB GUIDE 모델에 데이터폴 항목이 포함되어 있어야 합니다.
- EB GUIDE 모델의 시뮬레이션이 시작됩니다.
- EB GUIDE Monitor가 시작됩니다.

#### 단계 1

EB GUIDE Monitor의 **데이터폴** 구성 요소에서 **데이터폴 항목 검색** 검색 상자의 데이터폴 항목을 검색합니다.

#### 단계 2

데이터폴 항목을 클릭합니다.

데이터폴 항목이 목록에 추가됩니다.

#### 단계 3

**값** 열에서 데이터폴 항목의 값을 변경합니다.



**[참고]**



**지원되는 유형**

다음 데이터 유형의 데이터폴 항목을 변경할 수 있습니다.

- ▶ 부울
- ▶ 색
- ▶ 정수
- ▶ 부동
- ▶ 문자열

데이터폴 항목의 값이 변경됩니다. 로거 구성 요소에 로그 메시지가 표시됩니다.

### 10.9.3. EB GUIDE Monitor에서 스크립트 시작



#### EB GUIDE Monitor에서 스크립트 시작

선수학습:

- EB GUIDE 모델의 시뮬레이션이 시작됩니다.
- EB GUIDE Monitor가 시작됩니다.
- 컴퓨터에서 스크립트가 있는 .cs 또는 .dll 파일을 사용할 수 있습니다. 스크립트 예는 [섹션 10.9.3.1, “EB GUIDE Monitor에 대한 스크립트 파일 쓰기”](#)에서 참조하십시오.

단계 1

스크립팅 구성 요소를 열려면 **레이아웃 > 스크립팅**을 선택합니다.

스크립팅 구성 요소가 도킹된 구성 요소로 열립니다.

단계 2

스크립팅 구성 요소에서 **열기** 버튼을 클릭합니다.

파일 탐색기가 열립니다.

단계 3

.cs 또는 .dll 파일을 선택하고 **열기**를 클릭합니다.

파일에 포함된 적용할 수 있는 모든 메서드와 해당 클래스가 **스크립트** 테이블에 나열됩니다.

단계 4

메서드를 선택하고 시작 버튼을 클릭합니다.

스크립트가 시작됩니다. **스크립트 출력** 영역에 로그 메시지가 표시됩니다.

### 10.9.3.1. EB GUIDE Monitor에 대한 스크립트 파일 쓰기

스크립트 메시드에 대한 자세한 내용은 \$GUIDE\_INSTALL\_PATH/doc/monitor/monitor\_api.chm에서 EB GUIDE Monitor API를 참조하십시오.

다음은 기본 EB GUIDE Monitor 스크립트 함수에 대한 예입니다.

#### [참고]



#### 상태 및 상태 머신에 대한 메시드 사용

EB GUIDE 모델에 이름이 동일한 상태 또는 상태 머신이 여러 개 있는 경우 uint ID를 사용하십시오. \$EXPORT\_PATH/monitor.cfg에서 프로젝트와 관련된 uint ID를 찾으십시오.



#### 예 10.1. EB GUIDE Monitor의 예제 스크립트 파일

다음은 예제 스크립트 MonitorScriptSample.cs입니다.

```
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"); //❹
await monitor.Write(string.Format("State {0} left",
    leftState.Name));

var enteredState = await monitor.WaitForStateEnter
    ("Main", "State 2"); //❺
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"); //❻
    await monitor.Write(string.Format("State Machine {0} started",
        startedStateMachine.Name));

    var stoppedStateMachine = await monitor.WaitForStateMachineStop
        ("Dynamic state machine 1"); //❼
    await monitor.Write(string.Format("State Machine {0} stopped",
        stoppedStateMachine.Name));
}

public class Advanced
{
    public async Task CaptureScreenshot(IMonitorContext monitor) //❿
    {
        // 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) //⓫
    {
```

```
// 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");  
}  
}
```

- ❶ 메시지를 출력하는 메서드
- ❷ 이벤트를 실행하는 메서드
- ❸ 이벤트를 기다리는 메서드
- ❹ 데이터풀 값을 쓰는 메서드
- ❺ 데이터풀 값을 읽는 메서드
- ❻ 상태가 입력된 다음 보고할 때까지 기다리는 메서드
- ❼ 상태가 종료된 다음 보고할 때까지 기다리는 메서드
- ❽ 상태 머신이 시작된 다음 보고할 때까지 기다리는 메서드
- ❾ 상태 머신이 중지된 다음 보고할 때까지 기다리는 메서드
- ❿ 스크린샷을 캡처하는 메서드
- ⓫ 시간 초과인 이벤트를 기다리는 메서드

### 10.9.4. EB GUIDE Monitor를 독립 실행형 응용 프로그램으로 시작

EB GUIDE Monitor는 EB GUIDE 모델의 시뮬레이션 동안 EB GUIDE Studio에서 자동으로 시작합니다. 하지만 \$GUIDE\_INSTALL\_PATH/tools/monitor에서 독립 실행형 응용 프로그램으로 EB GUIDE Monitor를 시작하거나 명령줄을 사용할 수도 있습니다.



명령줄을 사용하여 EB GUIDE Monitor 시작

선수학습:

- EB GUIDE가 설치되어 있어야 합니다.

- EB GUIDE 모델을 \$EXPORT\_PATH로 내보냈습니다.

단계 1

파일 탐색기에서 \$GUIDE\_INSTALL\_PATH/tools/monitor로 이동합니다.

단계 2

명령줄을 열고 Monitor.exe를 입력하십시오.

EB GUIDE Monitor가 시작됩니다.



### EB GUIDE Monitor의 표시 언어 변경

선수학습:

- EB GUIDE Monitor는 독립 실행형 응용 프로그램으로 시작됩니다.

단계 1

**파일 > 표시 언어**에서 언어를 선택합니다.

단계 2

EB GUIDE Monitor를 다시 시작합니다.

다시 시작하면 그래픽 사용자 인터페이스가 선택한 언어로 표시됩니다.

**[참고]**



**EB GUIDE Monitor는 EB GUIDE Studio에서 언어를 상속합니다.**

EB GUIDE Studio에서 EB GUIDE Monitor가 시작되었다면 그래픽 사용자 인터페이스의 표시 언어를 변경할 수 없습니다. EB GUIDE Monitor의 표시 언어가 EB GUIDE Studio와 동일합니다.

## 10.9.5. 구성을 EB GUIDE Monitor로 로드



### EB GUIDE Monitor로 구성 파일 로드

선수학습:

- EB GUIDE Monitor는 독립 실행형 응용 프로그램으로 시작됩니다.
- EB GUIDE 모델을 \$EXPORT\_PATH로 내보냈습니다.

단계 1

**파일 > 구성 로드**를 선택합니다.

대화 상자가 열립니다.

단계 2

\$EXPORT\_PATH로 이동해 monitor.cfg 구성 파일을 선택합니다.

단계 3

열기를 클릭합니다.

프로젝트의 구성이 EB GUIDE Monitor에 로드됩니다.



EB GUIDE Monitor로 최근 구성 파일 로드

선수학습:

- EB GUIDE Monitor는 독립 실행형 응용 프로그램으로 시작됩니다.
- 최근 하나 이상의 구성이 사용되었습니다.

단계 1

파일 > 최근 구성 파일을 선택합니다.

대화 상자가 열립니다.

단계 2

각 위치로 이동해 구성 파일을 선택합니다.

구성이 EB GUIDE Monitor로 로드됩니다.

[참고]



**EB GUIDE Monitor를 EB GUIDE GTF에서 분리합니다.**

새 구성이 로드되기 전에 EB GUIDE Monitor가 현재의 EB GUIDE GTF에서 자동으로 분리됩니다.

EB GUIDE Monitor가 다시 연결해서 새 구성을 로드합니다.

## 10.9.6. 감시 목록 내보내기 및 가져오기

프로젝트를 위해 사용하는 이벤트 및 데이터풀 항목은 감시 목록에 저장됩니다. 다른 프로젝트의 항목을 이용하기 위해 감시 목록을 .xml 파일로 내보낸 후 나중에 새 프로젝트로 가져올 수 있습니다.



모든 감시 목록 내보내기

선수학습:

- EB GUIDE Monitor를 시작합니다.

- EB GUIDE 모델은 이미 **데이터폴** 또는 **이벤트** 탭에 저장된 항목으로 설정됩니다.

단계 1

모든 감시 목록을 내보내려면 **파일 > 모든 감시 목록 내보내기**를 선택합니다.

대화 상자가 열립니다.

단계 2

목적지 디렉터리를 선택하고 파일 이름을 입력합니다.

모든 데이터폴 항목과 이벤트가 내보내집니다.



### 단일 감시 목록 내보내기

선수학습:

- EB GUIDE Monitor를 시작합니다.
- EB GUIDE 모델은 이미 **데이터폴** 또는 **이벤트** 구성 요소에 저장된 항목으로 설정됩니다.

단계 1

내보내려는 항목이 포함된 **데이터폴** 또는 **이벤트** 구성 요소로 이동합니다.

단계 2

이 탭의 항목 목록을 저장하려면 **내보내기**를 클릭합니다.

대화 상자가 열립니다.

목적지 디렉터리를 선택하고 파일 이름을 입력합니다.

구성 요소의 데이터폴 항목 또는 이벤트가 내보내졌습니다.



### 감시 목록 가져오기

선수학습:

- EB GUIDE Monitor를 시작합니다.
- 내보내진 감시 목록은 이미 사용 가능합니다.

단계 1

감시 목록을 내보내려면 **파일 > 감시 목록 내보내기**를 선택합니다.

대화 상자가 열립니다.

단계 2

내보낼 감시 목록을 선택합니다.

데이터폴 항목 또는 이벤트는 새 **데이터폴** 또는 **이벤트** 구성 요소에서 열립니다.



[참고]



**레이아웃을 가져오지 않음**

레이아웃이 아닌 데이터폴 항목과 이벤트만 가져옵니다.

기본 레이아웃이 새롭게 열린 **데이터폴** 및 **이벤트** 구성 요소에 사용됩니다.

# 11. 자습서

## [참고]



### 기본 창 레이아웃

이 사용자 설명서의 모든 지침 및 스크린샷이 기본 창 레이아웃을 사용합니다. 지침을 따르는 경우 EB GUIDE Studio 창을 기본 레이아웃으로 설정하는 것이 좋습니다. **레이아웃 > 기본 레이아웃으로 다시 설정**을 선택하여 설정할 수 있습니다.

## 11.1. 자습서: 동적 상태 머신 추가

## [참고]



### 기본 창 레이아웃

이 사용자 설명서의 모든 지침 및 스크린샷이 기본 창 레이아웃을 사용합니다. 지침을 따르는 경우 EB GUIDE Studio 창을 기본 레이아웃으로 설정하는 것이 좋습니다. **레이아웃 > 기본 레이아웃으로 다시 설정**을 선택하여 설정할 수 있습니다.

동적 상태 머신에서는 런타임에 팝업을 표시할 수 있습니다. 예를 들어 동적 상태 머신을 사용하여 일반 디스플레이에 겹쳐지는 오류 메시지를 표시할 수 있습니다.

다음 지침에서는 동적 상태 머신을 만드는 프로세스를 안내합니다. 이 지침은 볼륨 제어용으로 동적 상태 머신을 모델링하는 방법을 제시합니다. 최적의 결과를 얻으려면 다음 단계를 아래에 나와 있는 순서대로 진행하십시오.

대략적인 소요 시간: 20분



### 이벤트 및 데이터풀 항목 추가

다음 지침에서는 이벤트 및 데이터풀 항목을 추가하는 프로세스를 안내합니다. 나중에 이러한 이벤트를 사용하여 볼륨을 변경합니다. 데이터풀 항목은 이후 섹션에서 그래픽 요소의 위치를 변경하는 데 사용됩니다.

#### 단계 1

**이벤트** 구성 요소로 이동한 다음 **+**을 클릭합니다.

이벤트가 테이블에 추가됩니다.

#### 단계 2

이벤트 이름을 Volume up으로 바꿉니다.

#### 단계 3

이벤트를 추가하고 이름을 Volume down으로 바꿉니다.

#### 단계 4

이벤트를 추가하고 이름을 Close volume control로 바꿉니다.

#### 단계 5

**데이터풀** 구성 요소로 이동한 다음 **+**을 클릭합니다.

메뉴가 확장됩니다.

#### 단계 6

메뉴에서 **정수**를 클릭합니다.

Integer 유형의 데이터폴 항목이 추가됩니다.

#### 단계 7

데이터폴 항목의 이름을 Volume indicator로 바꿉니다.

지금까지 이벤트 3개와 데이터폴 항목 1개를 추가했습니다.



### 동적 상태 머신 추가 및 동작 모델링

다음 지침에서는 동적 상태 머신을 추가하는 프로세스를 안내합니다. 여기서 모델링하는 햅틱 동적 상태 머신을 사용하여 볼륨을 제어합니다.

선수학습:

- 이 단계를 수행하려면 이전 지침의 단계를 완료한 상태여야 합니다.

#### 단계 1

**내비게이션** 구성 요소에서 **동적 상태 머신**로 이동한 다음 **+**를 클릭합니다.

메뉴가 확장됩니다.

#### 단계 2

메뉴에서 **햅틱 동적 상태 머신**을 클릭합니다.

햅틱 동적 상태 머신이 콘텐츠 영역에 추가되어 표시됩니다.

#### 단계 3

동적 상태 머신의 이름을 Volume control로 바꿉니다.

#### 단계 4

초기 상태를 **도구 상자**에서 동적 상태 머신으로 끕니다.

#### 단계 5

보기 상태를 **도구 상자**에서 동적 상태 머신으로 끕니다.

보기 상태와 함께 보기 하나가 EB GUIDE 모델에 추가됩니다.

#### 단계 6

**내비게이션** 구성 요소에서 보기 상태를 클릭합니다.

#### 단계 7

F2 키를 누르고 보기 상태의 이름을 Volume으로 바꿉니다.

#### 단계 8

초기 상태에서 Volume 보기 상태로의 전이를 추가합니다.



## 슬라이더 모델링

다음 지침에서는 가로 슬라이더 표시기를 모델링하는 프로세스를 안내합니다. 런타임 동안 슬라이더 표시기에 볼륨이 표시됩니다.

슬라이더 표시기는 사각형 2개로 구성됩니다. 사각형 중 하나는 슬라이더 배경을 나타내고 두 번째 사각형은 볼륨을 나타냅니다.

선수학습:

- 이 단계를 수행하려면 이전 지침의 단계를 완료한 상태여야 합니다.

### 단계 1

**내비게이션** 구성 요소에서 Volume 보기 상태를 확장합니다. 보기를 더블 클릭합니다.

콘텐츠 영역에 보기가 표시됩니다.

### 단계 2

사각형을 **도구 상자**에서 보기로 끕니다.

### 단계 3

**내비게이션** 구성 요소에서 사각형을 클릭하고 F2 키를 누릅니다.

### 단계 4

사각형 이름을 Slider background로 바꿉니다.

### 단계 5

Slider background의 모양을 변경하려면 사각형을 클릭하고 **속성** 구성 요소로 이동합니다.

#### 단계 5.1

width 텍스트 상자에 500을 입력합니다.

#### 단계 5.2

x 텍스트 상자에 125를 입력합니다.

#### 단계 5.3

y 텍스트 상자에 300을 입력합니다.

### 단계 6

사각형을 **도구 상자**에서 **내비게이션** 구성 요소의 Slider background로 끕니다.

사각형이 Slider background에 하위 위젯으로 추가됩니다.

### 단계 7

**내비게이션** 구성 요소에서 사각형을 클릭하고 F2 키를 누릅니다.

### 단계 8

사각형 이름을 Indicator로 바꿉니다.

### 단계 9

Indicator의 모양을 변경하려면 사각형을 클릭하고 **속성** 구성 요소로 이동합니다.

단계 9.1

width 텍스트 상자에 40을 입력합니다.

단계 9.2

height 텍스트 상자에 80을 입력합니다.

단계 9.3

x 속성 옆의  버튼을 클릭합니다.

메뉴가 확장됩니다.

단계 9.4

메뉴에서 **데이터폴 항목에 대한 링크 추가**를 클릭합니다.

대화 상자가 열립니다.

단계 9.5

목록에서 Volume indicator 데이터폴 항목을 선택합니다.

단계 9.6

**수락**을 클릭합니다.

대화 상자가 닫힙니다.  버튼이 x 속성 옆에 표시됩니다. 이제 x 및 Volume indicator의 값이 연결되었습니다.

단계 9.7

y 텍스트 상자에 10을 입력합니다.

단계 9.8

fillColor 속성에 대해 검정을 선택합니다.

사각형 2개가 보기에 추가되었습니다. 사각형의 모양을 변경했습니다.

단계 10

**데이터폴** 구성 요소에서 Volume indicator 데이터폴 항목을 클릭합니다.

단계 11

Value 텍스트 상자에 10을 입력합니다.

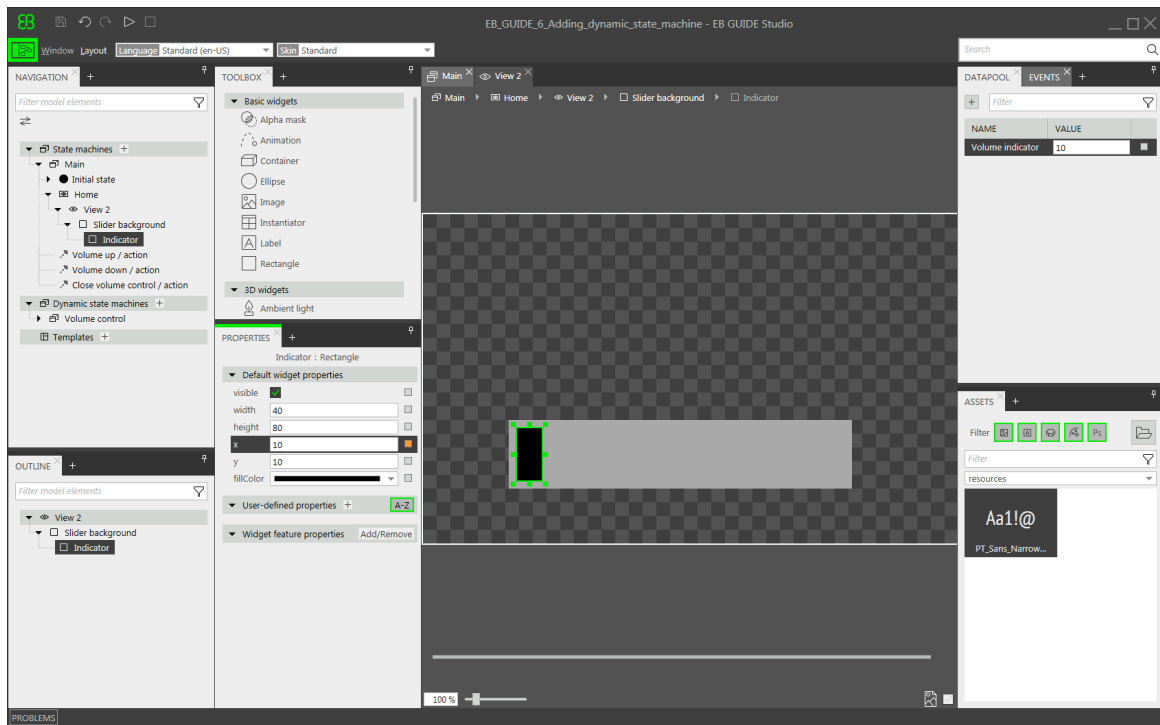


그림 11.1. 사각형 2개가 포함된 View 1의 모양

콘텐츠 영역에서 Indicator 사각형의 위치가 변경됩니다.

Volume indicator 데이터폴 항목은 Indicator 사각형의 x 위치를 제어합니다.



### 기본 상태 머신에 상태 추가하기

다음 지침에서는 **기본** 상태 머신에 초기 상태 및 보기 상태를 추가합니다. 보기 상태를 사용하여 다른 상태 머신과 병렬로 동적 상태 머신을 실행합니다.

선수학습:

- 이 단계를 수행하려면 이전 지침의 단계를 완료한 상태여야 합니다.

#### 단계 1

**내비게이션** 구성 요소에서 **기본**을 두 번 클릭합니다.

기본 상태 머신이 콘텐츠 영역에 표시됩니다.

#### 단계 2

초기 상태를 **도구 상자**에서 상태 머신으로 끕니다.

#### 단계 3

보기 상태를 **도구 상자**에서 상태 머신으로 끕니다.

보기 상태와 함께 보기 하나가 EB GUIDE 모델에 추가됩니다.

#### 단계 4

보기 상태의 이름을 Home으로 바꿉니다.

#### 단계 5

콘텐츠 영역에서 초기 상태를 클릭합니다.

#### 단계 6

초기 상태에서 Home 보기 상태로의 전이를 추가합니다.

#### 단계 7

**내비게이션** 구성 요소에서 **기본**을 클릭합니다.

#### 단계 8

**속성** 구성 요소에서 Dynamic state machine list 확인란을 선택합니다.

이러한 단계를 완료하면 동적 상태 머신과 관련된 EB GUIDE 스크립트 함수를 사용할 수 있습니다.

**기본** 상태 머신에 초기 상태 및 보기 상태를 추가했습니다. 햅틱 동적 상태 머신이 **기본** 상태 머신과 병렬로 실행됩니다.



### 기본 상태 머신에 내부 전이 추가하기

다음 지침에서는 내부 전이를 추가합니다. 내부 전이를 사용하여 런타임 동안 동적 상태 머신을 시작(푸시)하고 중지(팝)합니다.

선수학습:

- 이 단계를 수행하려면 이전 지침의 단계를 완료한 상태여야 합니다.

#### 단계 1

**내비게이션** 구성 요소에서 **기본** 상태 머신을 클릭합니다.

#### 단계 2

**속성** 구성 요소에서 **내부 전이**로 이동한 다음 **추가**를 클릭합니다.

내부 전이가 상태 머신에 추가됩니다. 내부 전이는 **내비게이션** 구성 요소에 표시됩니다.

#### 단계 3

내부 전이를 2개 더 추가합니다.

#### 단계 4

**내비게이션** 구성 요소에서 첫 번째 내부 전이를 클릭합니다.

##### 단계 4.1

**속성** 구성 요소로 이동합니다.

##### 단계 4.2

**트리거** 콤보 박스에서 Volume up을 선택합니다.

##### 단계 4.3

**작업** 속성 옆의 **추가**를 클릭합니다.

단계 4.4

다음 EB GUIDE 스크립트를 입력합니다.

```
function()
{
  dp:"Volume indicator" = dp:"Volume indicator" + 20
  f:pushDynamicStateMachine(popup_stack:Main, sm:"Volume control", 0)
}
```

단계 4.5

수락을 클릭합니다.

작업이 전이에 추가됩니다. **내비게이션** 구성 요소에서 내부 전이의 이름이 Volume up로 바뀝니다.

단계 5

**내비게이션** 구성 요소에서 두 번째 내부 전이를 클릭합니다.

단계 5.1

속성 구성 요소로 이동합니다.

단계 5.2

**트리거** 콤보 박스에서 Volume down을 선택합니다.

단계 5.3

작업 속성 옆의 **추가**를 클릭합니다.

단계 5.4

다음 EB GUIDE 스크립트를 입력합니다.

```
function()
{
  dp:"Volume indicator" = dp:"Volume indicator" - 20
  f:pushDynamicStateMachine(popup_stack:Main, sm:"Volume control", 0)
}
```

단계 5.5

수락을 클릭합니다.

작업이 전이에 추가됩니다. **내비게이션** 구성 요소에서 내부 전이의 이름이 Volume down로 바뀝니다.

단계 6

**내비게이션** 구성 요소에서 세 번째 내부 전이를 클릭합니다.

단계 6.1

속성 구성 요소로 이동합니다.

단계 6.2

**트리거** 콤보 박스에서 Close volume control을 선택합니다.

단계 6.3

작업 속성 옆의 **추가**를 클릭합니다.



#### 단계 6.4

다음 EB GUIDE 스크립트를 입력합니다.

```
function()
{
  f:popDynamicStateMachine(popup_stack:Main,sm:"Volume control")
}
```

#### 단계 6.5

수락을 클릭합니다.

작업이 전이에 추가됩니다. **내비게이션** 구성 요소에서 내부 전이의 이름이 Close volume control로 바뀝니다.

동적 상태 머신을 시작하고 중지하는 내부 전이 3개를 추가했습니다. 그리고 내부 전이 Volume up과 Volume down은 Indicator 사각형의 위치를 변경합니다.

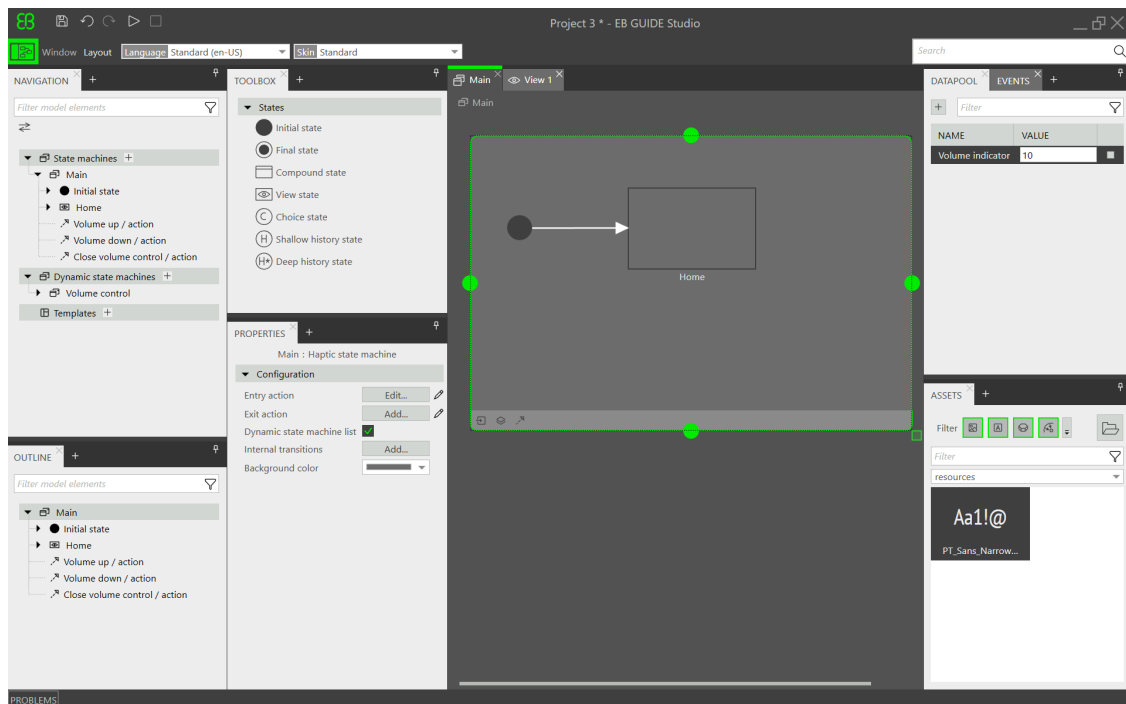


그림 11.2. 모든 모델 요소가 포함된 EB GUIDE 모델



### 시뮬레이션 시작하기 및 EB GUIDE 모델 테스트하기


#### 선수학습:

- 이 단계를 수행하려면 이전 지침의 단계를 완료한 상태여야 합니다.

시뮬레이션을 시작하려면 명령 영역에서 ▶ 을 클릭합니다.

시뮬레이션 및 EB GUIDE Monitor가 시작됩니다. EB GUIDE 모델에 Home 보기 상태가 표시됩니다.

단계 1

**이벤트** 구성 요소의 EB GUIDE Monitor에서 Volume up 이벤트를 선택한 다음  를 클릭하여 이벤트를 실행합니다.

동적 상태 머신이 시작되고 슬라이더 표시기가 표시됩니다. 동적 상태 머신이 Home 보기 상태에 겹쳐져서 표시됩니다.

Volume up 또는 Volume down 이벤트를 실행하면 검은색 Indicator 사각형이 이동합니다. Close volume control 이벤트를 실행하면 슬라이더가 보기에서 사라집니다.

**기본** 상태 머신에 상태를 더 추가하면 Volume control 동적 상태 기계가 다른 상태에도 겹쳐져서 표시됩니다.

## 11.2. 자습서: EB GUIDE 스크립트를 사용하여 버튼 동작 모델링

### [참고]



### 기본 창 레이아웃

이 사용자 설명서의 모든 지침 및 스크린샷이 기본 창 레이아웃을 사용합니다. 지침을 따르는 경우 EB GUIDE Studio 창을 기본 레이아웃으로 설정하는 것이 좋습니다. **레이아웃 > 기본 레이아웃으로 다시 설정**을 선택하여 설정할 수 있습니다.

EB GUIDE 스크립트를 사용하면 속성 값, 작업 또는 조건을 표시한 다음 런타임에 평가할 수 있습니다.

다음 지침에서는 EB GUIDE 스크립트를 사용하여 버튼의 동작을 모델링하는 프로세스를 안내합니다. 버튼은 클릭하면 커지고 정의된 최대 크기에 도달하면 원래 크기로 다시 작아집니다. 최적의 결과를 얻으려면 단계를 아래에 나와 있는 순서대로 진행하십시오.

대략적인 소요 시간: 10분



위젯 추가

선수학습:

- **기본** 상태 기계에 초기 상태와 보기 상태가 포함되어 있어야 합니다.
- 초기 상태에 보기 상태로의 전이가 있어야 합니다.
- 콘텐츠 영역에 보기가 표시됩니다.

단계 1

사각형을 **도구 상자**에서 보기로 끕니다.



#### 단계 2

**내비게이션** 구성 요소에서 사각형을 클릭하고 F2 키를 누른 다음 사각형의 이름을 Background로 바꿉니다.

#### 단계 3

사각형을 **도구 상자**에서 **내비게이션** 구성 요소로 끕니다. 사각형을 Background 사각형에 하위 위젯으로 배치합니다.

#### 단계 4

**내비게이션** 구성 요소에서 새 사각형을 클릭하고 F2 키를 누른 다음 사각형의 이름을 Button으로 바꿉니다.

#### 단계 5

레이블을 **도구 상자**에서 **내비게이션** 구성 요소로 끕니다. 레이블을 Button 사각형에 하위 위젯으로 배치합니다.

#### 단계 6

**내비게이션** 구성 요소에서 레이블을 클릭하고 F2 키를 누른 다음 레이블의 이름을 Button text로 바꿉니다.

이제 위젯 계층이 다음과 같이 표시됩니다.

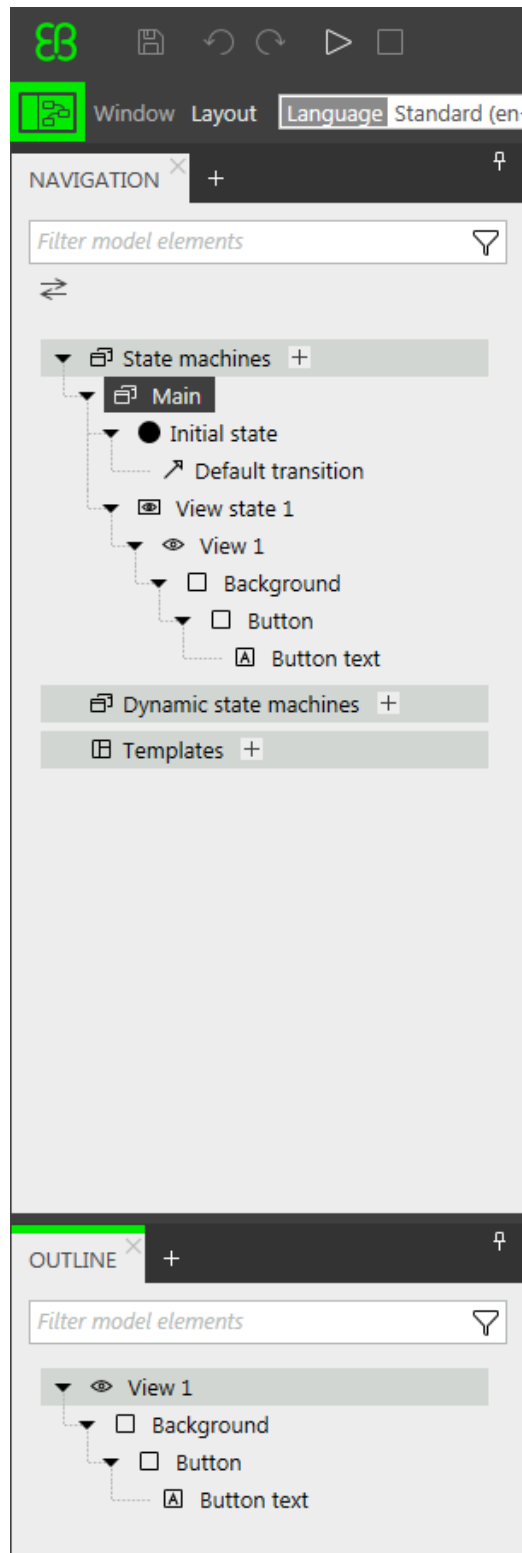


그림 11.3. 위젯 계층



## 배경 구성

선수학습:

- 이 단계를 수행하려면 이전 지침의 단계를 완료한 상태여야 합니다.

### 단계 1

**내비게이션** 구성 요소에서 Background 사각형을 클릭하고 **속성** 구성 요소로 이동합니다.

### 단계 2

width 속성 옆의  버튼을 클릭합니다.

메뉴가 확장됩니다.

### 단계 3

메뉴에서 **위젯 속성에 대한 링크 추가**를 클릭합니다.

대화 상자가 열립니다.

### 단계 4

대화 상자에서 보기로 이동한 다음 해당 width 속성을 선택합니다.

### 단계 5

**수락**을 클릭합니다.

대화 상자가 닫힙니다.  버튼이 width 속성 옆에 표시됩니다.

### 단계 6

Background 사각형의 height 속성을 보기의 height 속성에 연결합니다.

### 단계 7

Background 사각형의 x 속성을 보기의 x 속성에 연결합니다.

### 단계 8

Background 사각형의 y 속성을 보기의 y 속성에 연결합니다.

Background 사각형은 보기의 정확한 크기와 위치로 설정됩니다.



## 최대 버튼 너비 정의

데이터풀 항목에는 버튼의 최대 너비에 해당하는 값이 저장됩니다. 런타임 동안 이 값을 변경할 수 있습니다.

선수학습:

- 이 단계를 수행하려면 이전 지침의 단계를 완료한 상태여야 합니다.

### 단계 1

**데이터풀** 구성 요소에서 **+**를 클릭합니다.

메뉴가 확장됩니다.

#### 단계 2

메뉴에서 **정수**를 클릭합니다.

Integer의 새 데이터풀 항목이 추가됩니다.

#### 단계 3

데이터풀 항목의 이름을 Maximum width로 바꿉니다.

#### 단계 4

Value 텍스트 상자에 400을 입력합니다.



## 버튼 구성

선수학습:

- 이 단계를 수행하려면 이전 지침의 단계를 완료한 상태여야 합니다.

#### 단계 1

**내비게이션** 구성 요소에서 Button 사각형을 클릭하고 **속성** 구성 요소로 이동합니다.

##### 단계 1.1

height 텍스트 상자에 50을 입력합니다.

##### 단계 1.2

x 텍스트 상자에 350을 입력합니다.

##### 단계 1.3

y 텍스트 상자에 215를 입력합니다.

##### 단계 1.4

fillColor 속성에 대해 파란색을 선택합니다.

이제 버튼이 파란색으로 설정되었습니다.

#### 단계 2

**위젯 피쳐 속성** 범주에서 **추가/제거**를 클릭합니다.

**위젯 피쳐** 대화 상자가 표시됩니다.

#### 단계 3

**사용 가능한 위젯 피쳐** 아래에서 **입력 처리** 범주를 확장하고 **터치 누름** 위젯 피쳐를 선택합니다.

#### 단계 4

**수락**을 클릭합니다.

관련 위젯 피쳐 속성이 Button 사각형에 추가되고 **속성** 구성 요소에 표시됩니다.

#### 단계 5

touchPressed 속성 옆의 **편집**을 클릭합니다.

#### 단계 6

기존 EB GUIDE 스크립트를 다음 코드로 바꿉니다.

```
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
}
```

단계 7

수락을 클릭합니다.

Button 사각형을 구성하고 런타임에 Button 사각형의 크기를 변경하는 EB GUIDE 스크립트를 작성했습니다.



## 버튼 텍스트 구성

선수학습:

- 이 단계를 수행하려면 이전 지침의 단계를 완료한 상태여야 합니다.

단계 1

**내비게이션** 구성 요소에서 Button text 레이블을 클릭하고 **속성** 구성 요소로 이동합니다.

단계 2

text 텍스트 상자에 grow!를 입력합니다.

단계 3

Button text 레이블의 width 속성을 Button 사각형의 width 속성에 연결합니다.

단계 4

Button text 레이블의 height 속성을 Button 사각형의 height 속성에 연결합니다.

단계 5

x 텍스트 상자에 0를 입력합니다.

단계 6

y 텍스트 상자에 0을 입력합니다.

단계 7

horizontalAlign 속성 옆의 = 을 클릭합니다.

이제 Button text 레이블과 Button 사각형의 크기와 위치가 같아졌습니다.




## EB GUIDE 모델 저장 및 테스트


## 선수학습:

- 이 단계를 수행하려면 이전 지침의 단계를 완료한 상태여야 합니다.

단계 1

프로젝트를 저장하려면 명령 영역에서  을 클릭합니다.

단계 2

시뮬레이션을 시작하려면 명령 영역에서  을 클릭합니다.

## 결과:

앞에서 만든 EB GUIDE 모델이 시뮬레이션에서 시작됩니다. 이 모델은 다음과 같이 작동합니다.

1. 먼저 가운데에 파란색 버튼이 있는 회색 화면이 표시됩니다. 이 화면의 모양은 다음과 같습니다.





그림 11.4. 결과

2. 버튼은 클릭할 때마다 크기가 커지지만 위치는 화면 가운데로 유지됩니다.
3. 버튼은 너비가 Maximum width 데이터폴 항목의 값에 도달하는 즉시 원래 크기와 위치로 다시 작아집니다.

## 11.3. 자습서: 경로 제스처 모델링

### [참고]



### 기본 창 레이아웃

이 사용자 설명서의 모든 지침 및 스크린샷이 기본 창 레이아웃을 사용합니다. 지침을 따르는 경우 EB GUIDE Studio 창을 기본 레이아웃으로 설정하는 것이 좋습니다. **레이아웃 > 기본 레이아웃으로 다시 설정**을 선택하여 설정할 수 있습니다.

경로 제스처는 터치 스크린에 손가락으로 그리거나 다른 입력 기기를 통해 입력하는 도형입니다.

다음 지침에서는 경로 제스처를 모델링하는 프로세스를 안내합니다.

대략적인 소요 시간: 10분



### 위젯 추가 및 기본 위젯 속성 구성

선수학습:

- **기본** 상태 기계에 초기 상태와 보기 상태가 포함되어 있어야 합니다.
- 초기 상태에 보기 상태로의 전이가 있어야 합니다.
- 콘텐츠 영역에 보기가 표시되어 있어야 합니다.

#### 단계 1

사각형을 **도구 상자**에서 보기로 끕니다.

#### 단계 2

레이블을 **도구 상자**에서 사각형으로 끕니다.

레이블이 사각형에 하위 위젯으로 추가됩니다.

**속성** 구성 요소에 레이블의 속성이 표시됩니다.

#### 단계 3

**속성** 구성 요소에서 width 텍스트 상자에 500을 입력합니다.

#### 단계 4

사각형을 선택합니다.

**속성** 구성 요소에 사각형의 속성이 표시됩니다.

#### 단계 5

width 텍스트 상자에 500를 입력합니다.

#### 단계 6

**속성** 구성 요소에서 **fillColor**로 이동한 다음 빨강을 선택합니다.

위젯 2개를 추가하고 기본 위젯 속성을 구성했습니다.



### 사각형에 위젯 피쳐 추가

사용자가 위젯에서 시작하는 도형을 입력할 수 있도록 하려면 사각형에 **경로 제스처** 위젯 피쳐를 추가합니다. 이 도형과 알려진 도형 집합의 일치 여부를 확인한 다음 일치하는 항목이 있으면 제스처가 인식됩니다.

선수학습:

- 이 단계를 수행하려면 이전 지침의 단계를 완료한 상태여야 합니다.

#### 단계 1

사각형을 선택합니다.

**속성** 구성 요소에 사각형의 속성이 표시됩니다.

단계 2

**속성** 구성 요소에서 **위젯 피쳐 속성**으로 이동한 다음 **추가/제거**를 클릭합니다.

**위젯 피쳐** 대화 상자가 표시됩니다.

단계 3

**사용 가능한 위젯 피쳐** 아래에서 **제스처** 범주를 확장하고 Path gestures를 선택합니다.

**제스처** 위젯 피쳐에 필요한 **터치함위젯 피쳐**가 자동으로 선택됩니다.

단계 4

**수락**을 클릭합니다.

관련 위젯 피쳐 속성이 사각형에 추가되고 **속성** 구성 요소에 표시됩니다.

단계 5

**경로 제스처** 위젯 피쳐에 대해 다음 속성을 편집합니다.

단계 5.1

onPath 속성 옆의 **편집...**을 클릭합니다.

단계 5.2

다음 EB GUIDE 스크립트를 입력합니다.

```
function(v:gestureId:int)
{
    v.this->"Label 1".text = "recognized path gesture #"
    + f:int2string(v:gestureId);
}
```

단계 5.3

**수락**을 클릭합니다.

단계 5.4

onPathStart 속성 옆의 **편집...**을 클릭합니다.

단계 5.5

다음 EB GUIDE 스크립트를 입력합니다.

```
function()
{
    v.this->"Label 1".text = "path gesture start";
}
```

단계 5.6

**수락**을 클릭합니다.

단계 5.7

onPathNotRecognized 속성 옆의 **편집...**을 클릭합니다.

단계 5.8

다음 EB GUIDE 스크립트를 입력합니다.

```
function()
{
  v.this->"Label 1".text = "shape not recognized";
}
```

#### 단계 5.9

수락을 클릭합니다.

#### 단계 6

시뮬레이션을 시작하려면 명령 영역에서 ▶을 클릭합니다.

시뮬레이션 및 EB GUIDE Monitor가 시작됩니다. 반응을 확인하려면 사각형 안에서 마우스로 도형을 그립니다.

## 11.4. 자습서: 동적 콘텐츠로 목록 만들기

### [참고]



### 기본 창 레이아웃

이 사용자 설명서의 모든 지침 및 스크린샷이 기본 창 레이아웃을 사용합니다. 지침을 따르는 경우 EB GUIDE Studio 창을 기본 레이아웃으로 설정하는 것이 좋습니다. **레이아웃 > 기본 레이아웃으로 다시 설정**을 선택하여 설정할 수 있습니다.

인스턴스화기를 사용하면 런타임 동안 동적으로 목록을 만들 수 있습니다. 목록 유형의 데이터풀 항목에 따라 인스턴스화기는 미리 정의된 레이아웃에 모든 목록 요소를 표시합니다. 데이터풀 항목의 콘텐츠를 수정하면 인스턴스화기의 모양도 수정됩니다.

다음 지침에서는 동적 콘텐츠를 사용하여 목록을 만드는 프로세스를 안내합니다. 각 목록 요소는 레이블이 지정된 사각형으로 구성됩니다.

대략적인 소요 시간: 15분



### 데이터풀 항목 추가

다음 지침에서는 String list 유형의 데이터풀 항목을 추가하는 프로세스를 안내합니다. 데이터풀 항목은 인스턴스화기의 모든 목록 요소에 대해 값을 제공합니다. 데이터풀 항목의 콘텐츠를 수정하면 인스턴스화기의 모양도 수정됩니다.

선수학습:

- 기본 상태 머신에 초기 상태와 보기 상태가 포함되어 있어야 합니다.
- 초기 상태에 보기 상태로의 전이가 있어야 합니다.

단계 1

목록에서 콘텐츠를 표시하려면 String list. 유형의 데이터폴 항목을 추가합니다.

**데이터폴** 구성 요소에서 **+**를 클릭합니다.

메뉴가 확장됩니다.

단계 2


메뉴에서 **문자열 목록**을 클릭합니다.

String list의 새 데이터폴 항목이 추가됩니다.

단계 3

데이터폴 항목의 이름을 MyStringList로 바꿉니다.

단계 4

 버튼을 클릭합니다.

편집기가 열립니다.

단계 4.1

**추가**를 클릭합니다.

새 항목이 테이블에 추가됩니다.

단계 4.2

Value 텍스트 상자에 One를 입력합니다.

단계 4.3

Two, Three, Four, Five 값을 MyStringList 데이터폴 항목에 추가합니다.

단계 4.4

**수락**을 클릭합니다.

String list 유형의 데이터폴 항목을 추가했습니다. 데이터폴 항목에 5개 항목이 포함되어 있습니다.

목록의 콘텐츠가 Value 속성 옆에 표시됩니다.



위젯 추가

선수학습:

- 이 단계를 수행하려면 이전 지침의 단계를 완료한 상태여야 합니다.

단계 1

보기에 위젯을 추가하려면 콘텐츠 영역의 보기 상태를 더블 클릭합니다.

보기가 콘텐츠 영역에 표시됩니다.

단계 2

**내비게이션** 구성 요소에서 보기 상태 및 보기를 확장합니다.

단계 3

인스턴스화기를 **도구 상자**에서 보기로 끕니다. 인스턴스화기 이름을 MyInstantiator로 바꿉니다.

단계 4

사각형을 **도구 상자**에서 인스턴스화기로 끕니다. 사각형 이름을 MyRectangle로 바꿉니다.

단계 5

레이블을 **도구 상자**에서 사각형으로 끕니다. 레이블 이름을 MyLabel로 바꿉니다.

이제 위젯 계층이 다음과 같이 표시됩니다.

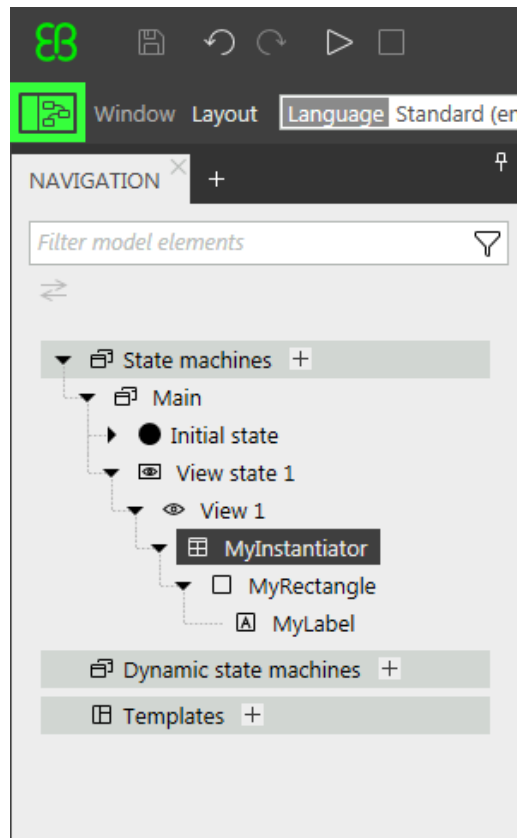


그림 11.5. 인스턴스화기가 포함된 위젯 계층



인스턴스화기 구성하기

선수학습:

- 이 단계를 수행하려면 이전 지침의 단계를 완료한 상태여야 합니다.

단계 1

MyInstantiator의 속성을 변경하려면 인스턴스화기를 선택하고 **속성** 구성 요소로 이동합니다.

단계 2

width 텍스트 상자와 height 텍스트 상자에 300을 입력합니다.

단계 3

x 텍스트 상자에 250을 입력합니다.

단계 4

y 텍스트 상자에 150을 입력합니다.

단계 5

목록의 길이를 동적으로 계산하려면 조건부 스크립트를 추가합니다.

**사용자 정의 속성** 범주에서 **+**를 클릭합니다.

메뉴가 확장됩니다.

단계 5.1

메뉴에서 **조건부 스크립트**를 클릭합니다.

단계 5.2

속성의 이름을 calculateNumItems로 바꿉니다.

단계 5.3

calculateNumItems 속성 옆의 **편집**을 클릭합니다.

콘텐츠 영역에 EB GUIDE 스크립트 편집기가 열립니다.

단계 5.4

MyStringList 데이터폴 항목을 **시발점** 목록에 추가합니다.

단계 5.5

다음 **On trigger** 스크립트를 입력합니다.

```
function(v:arg0::bool)
{
  v.this.numItems = length dp:MyStringList;
  false
}
```

MyStringList의 콘텐츠에 따라 목록 항목의 수를 자동으로 변경하는 스크립트를 추가했습니다.

단계 6

인스턴스화기 내의 모든 레이블을 정렬하려면 레이아웃을 인스턴스화기에 추가합니다.

**위젯 피쳐 속성** 범주에서 **추가/제거**를 클릭합니다.

**위젯 피쳐** 대화 상자가 표시됩니다.

단계 6.1

레이블을 나란히 정렬하려면 **사용 가능한 위젯 피쳐** 아래에서 **레이아웃** 범주를 확장하고 **상자 레이아웃** 위젯 피쳐를 선택합니다.

관련 위젯 피쳐 속성이 인스턴스화기에 추가되고 **속성** 구성 요소에 표시됩니다.

단계 6.2

**수락**을 클릭합니다.

단계 6.3

각 목록 요소 사이에 5px의 간격을 설정하려면 gap 텍스트 상자에 5를 입력합니다.

단계 6.4

각각의 레이블을 정렬하려면 **layoutDirection** 드롭다운 목록 상자에서 **세로(=1)**를 선택합니다.

목록의 시각적 모양을 정의하고 목록 항목의 수를 동적으로 조정하는 인스턴스화기를 구성했습니다.



## 목록 요소 텍스트 구성하기

선수학습:

- 이 단계를 수행하려면 이전 지침의 단계를 완료한 상태여야 합니다.

단계 1

레이블의 모양을 변경하려면 레이블을 선택하고 **속성** 구성 요소로 이동합니다.

단계 2

x 및 y 텍스트 상자에 0을 입력합니다.

단계 3

레이블 width 속성에서 사각형 width 속성으로의 링크를 추가합니다.

단계 3.1

width 속성 옆의  버튼을 클릭합니다.

메뉴가 확장됩니다.

단계 3.2

메뉴에서 **위젯 속성에 대한 링크 추가**를 클릭합니다.

대화 상자가 열립니다.

단계 3.3

대화 상자에서 사각형으로 이동한 다음 해당 width 속성을 선택합니다.

단계 3.4

**수락**을 클릭합니다.

대화 상자가 닫힙니다.  버튼이 width 속성 옆에 표시됩니다.

단계 4

레이블의 height 속성에서 사각형 height 속성으로의 링크를 추가합니다.

단계 5

horizontalAlign 속성 옆의  을 클릭합니다.

레이블의 모양을 변경했습니다. 이제 레이블이 사각형의 가운데에 배치됩니다.





## 목록 요소 구성

선수학습:

- 이 단계를 수행하려면 이전 지침의 단계를 완료한 상태여야 합니다.

### 단계 1

사각형의 모양을 변경하려면 사각형을 선택하고 **속성** 구성 요소로 이동합니다.

### 단계 2

목록 요소가 사용 가능한 모든 너비를 사용하도록 하려면 사각형 width 속성에서 인스턴스화기 width 속성으로의 링크를 추가합니다.

### 단계 3

height 텍스트 상자에 50를 입력합니다.

### 단계 4

목록의 각 줄에 대한 고유 위치를 정의하려면 **줄 인덱스** 위젯 피처를 추가합니다.

#### 단계 4.1

**위젯 피처 속성** 범주에서 **추가/제거**를 클릭합니다.

**위젯 피처** 대화 상자가 표시됩니다.

#### 단계 4.2

**사용 가능한 위젯 피처** 아래에서 **목록 관리** 범주를 확장하고 **줄 인덱스** 위젯 피처를 선택합니다.

lineIndex 속성이 사각형의 속성에 추가됩니다.

### 단계 5

MyStringList의 콘텐츠로 목록의 레이블을 채우려면 조건부 스크립트를 추가합니다.

#### 단계 5.1

**사용자 정의 속성** 범주 옆의 **+**를 클릭합니다.

메뉴가 확장됩니다.

#### 단계 5.2

메뉴에서 **조건부 스크립트**를 클릭합니다.

#### 단계 5.3

속성의 이름을 setText로 바꿉니다.

#### 단계 5.4

setText 속성 옆의 **편집**을 클릭합니다.

콘텐츠 영역에 EB GUIDE 스크립트 편집기가 열립니다.

#### 단계 5.5

사각형의 lineIndex 속성 및 MyStringList 데이터폴 항목을 **시발점** 목록에 추가합니다.

#### 단계 5.6

다음 **On trigger** 스크립트를 입력합니다.

```
function(v:arg0::bool)
{
  v.this->MyLabel.text=dp:MyStringList[v.this.lineIndex];
  false
}
```

사각형의 모양을 변경했습니다. setText 속성과 함께 MyStringList의 콘텐츠로 MyStringList의 레이블이 자동으로 채워집니다.



#### EB GUIDE 모델 테스트

선수학습:

- 이 단계를 수행하려면 이전 지침의 단계를 완료한 상태여야 합니다.

#### 단계 1

시뮬레이션을 시작하려면 명령 영역에서 ▶ 을 클릭합니다.

결과:

MyStringList에 5개의 데이터폴 항목이 포함되어 있으므로 레이블 1~5가 지정된 5개 사각형이 세로로 정렬되어 표시됩니다.

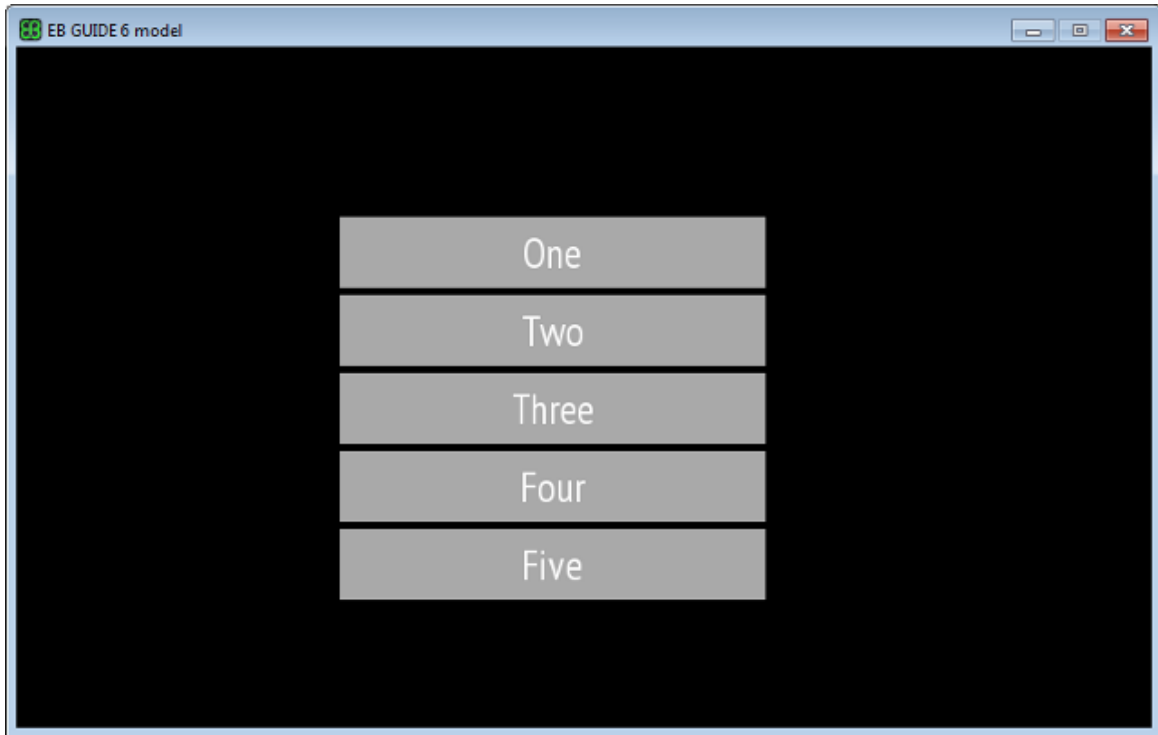


그림 11.6. 인스턴스화기를 사용하여 만든 목록

## 11.5. 자습서: 화면을 가로지르는 타원 이동 생성

**[참고]**



**기본 창 레이아웃**

이 사용자 설명서의 모든 지침 및 스크린샷이 기본 창 레이아웃을 사용합니다. 지침을 따르는 경우 EB GUIDE Studio 창을 기본 레이아웃으로 설정하는 것이 좋습니다. **레이아웃 > 기본 레이아웃으로 다시 설정**을 선택하여 설정할 수 있습니다.

다음 지침에서는 시뮬레이션이 시작될 때 타원이 화면을 가로질러 이동하도록 애니메이션을 생성하는 프로세스를 안내합니다.

대략적인 소요 시간: 5분

위젯 추가

다음 단계에서는 보기에 위젯 3개를 추가하고 위젯의 계층을 구성합니다.

선수학습:

- 콘텐츠 영역에 **기본** 상태 머신이 표시되어 있어야 합니다.

- **기본** 상태 머신에 초기 상태와 보기 상태가 포함되어 있어야 합니다.
- 초기 상태에 보기 상태로의 전이가 있어야 합니다.

#### 단계 1

콘텐츠 영역에서 보기 상태를 두 번 클릭합니다.

보기가 콘텐츠 영역에 표시됩니다.

#### 단계 2

타원을 **도구 상자**에서 보기로 끕니다.

#### 단계 3

애니메이션을 **도구 상자**에서 타원으로 끕니다.

#### 단계 4

**내비게이션** 구성 요소에서 애니메이션을 클릭하고 F2 키를 누릅니다. 애니메이션 이름을 MyAnimation으로 바꿉니다.

이제 시뮬레이션을 시작하면 타원이 보기에 표시됩니다. 하지만 타원이 아직 움직이지는 않습니다.



### Conditional script 유형의 사용자 정의 속성 추가

다음 단계에서는 타원에 사용자 정의 속성을 추가합니다. 조건부 스크립트 속성을 사용하는 경우 시뮬레이션 중에 타원을 렌더링하면 애니메이션이 시작됩니다.

선수학습:

- 이 단계를 수행하려면 이전 지침의 단계를 완료한 상태여야 합니다.

#### 단계 1

타원을 선택합니다.

#### 단계 2

**속성** 구성 요소에서 **사용자 정의 속성** 범주로 이동한 다음 **+**를 클릭합니다.

메뉴가 확장됩니다.

#### 단계 3

메뉴에서 Conditional script를 클릭합니다.

Conditional script 유형의 사용자 정의 속성이 타원에 추가됩니다.

#### 단계 4

속성의 이름을 startAnimation로 바꿉니다.

#### 단계 5

startAnimation 속성 옆의 **편집**을 클릭합니다.

콘텐츠 영역에 EB GUIDE 스크립트 편집기가 열립니다.

단계 6

다음 EB GUIDE 스크립트를 입력합니다.

```
function(v:arg0::bool)
{
  f:animation_play(v:this->MyAnimation)
}
```



## 애니메이션 표시

다음 지침에서는 애니메이션을 표시하는 프로세스를 안내합니다.

선수학습:

- 이 단계를 수행하려면 이전 지침의 단계를 완료한 상태여야 합니다.
- 콘텐츠 영역에 View 1 보기가 표시됩니다.

단계 1

**애니메이션 편집기**로 이동합니다. **애니메이션 속성** 옆의 **+**를 클릭하고 View 1을 선택합니다.


메뉴가 확장됩니다.

단계 2

Ellipse 1 아래에서 x 속성을 선택한 다음 **선형 보간 커브**를 선택합니다.

단계 3

**수락**을 클릭합니다.

 버튼이 target 속성 옆에 표시됩니다.

단계 4

end 속성을 보기의 width 속성에 연결합니다.

이러한 설정을 사용하는 경우 애니메이션이 시작될 때 타원의 x 속성이 0에서 보기의 너비로 변경됩니다. 따라서 타원이 보기의 왼쪽 경계에서 오른쪽 경계로 이동합니다.

단계 5

애니메이션이 무한히 반복 실행되도록 하려면 repeat 속성에 0을 입력합니다.

단계 6

프로젝트를 저장합니다.

단계 7

시뮬레이션을 시작하려면 명령 영역에서 ▶을 클릭합니다.

결과:

타원이 보기의 왼쪽에서 오른쪽으로 계속 이동합니다.

## 11.6. 자습서: 데이터폴 항목에 언어별로 다른 텍스트 추가

### [참고]



#### 기본 창 레이아웃

이 사용자 설명서의 모든 지침 및 스크린샷이 기본 창 레이아웃을 사용합니다. 지침을 따르는 경우 EB GUIDE Studio 창을 기본 레이아웃으로 설정하는 것이 좋습니다. **레이아웃 > 기본 레이아웃으로 다시 설정**을 선택하여 설정할 수 있습니다.

EB GUIDE에서는 사용자의 기본 설정 언어로 텍스트를 표시할 수 있습니다. 다음 지침에서는 영어, 프랑스어 및 독일어 사용자 인터페이스에서 서로 다르게 변경되는 레이블을 모델링하는 방법을 설명합니다.

대략적인 소요 시간: 15분

### [참고]



#### 언어 종속성의 전제조건

데이터폴 항목에 언어 지원을 추가하려면 다음 작업을 수행합니다.

- ▶ 데이터폴 항목의 Value 속성이 다른 데이터폴 항목 또는 위젯 속성과 연결되어 있으면 링크를 제거합니다.
- ▶ 데이터폴 항목의 Value 속성이 스크립팅된 값이면 속성을 일반 값으로 변환합니다.



#### 데이터폴 항목에 위젯 속성 연결

다음 지침에서는 레이블의 text 속성을 데이터폴 항목에 연결하는 프로세스를 안내합니다. 런타임에 표시되는 텍스트는 데이터폴 항목에서 제공됩니다.

선수학습:

- 언어 3개(영어, 독일어, 프랑스어)가 EB GUIDE 모델에 추가되었습니다. **언어 1**의 이름은 German으로, **언어 2**의 이름은 French로 설정되었습니다.
- 콘텐츠 영역에 **기본** 상태 머신이 표시되어 있어야 합니다.
- 기본 상태 머신에 초기 상태와 보기 상태가 포함되어 있어야 합니다.
- 콘텐츠 영역에 보기가 표시됩니다.
- 초기 상태에 보기 상태로의 전이가 있어야 합니다.
- 보기 상태에 레이블이 포함되어 있어야 합니다.

#### 단계 1

레이블을 클릭합니다.

#### 단계 2

속성 구성 요소에서 text 속성으로 이동한 다음 속성 옆의  버튼을 클릭합니다.

단계 3

메뉴에서 **데이터폴 항목에 대한 링크 추가**를 클릭합니다.

대화 상자가 열립니다.

단계 4

새 데이터폴 항목을 추가하려면 텍스트 상자에 Welcome\_text를 입력합니다.

단계 5

**데이터폴 항목 추가**를 클릭합니다.

단계 6

**수락**을 클릭합니다.

데이터폴 항목 Welcome\_text이 추가됩니다.

콘텐츠 영역에서 레이블에 더 이상 텍스트가 표시되지 않습니다.



데이터폴 항목에 언어별로 다른 텍스트 입력

다음 지침에서는 데이터폴 항목에 언어별로 다른 텍스트를 추가하는 프로세스를 안내합니다. 모든 언어에서 Value 속성에 서로 다른 텍스트가 포함되어 있어야 합니다.

선수학습:

- 이 단계를 수행하려면 이전 지침의 단계를 완료한 상태여야 합니다.

단계 1

**데이터폴** 구성 요소에서 Welcome\_text 데이터폴 항목을 클릭합니다.

단계 2

■ 버튼을 클릭합니다.

단계 3

메뉴에서 **언어 지원 추가**를 클릭합니다.

**속성** 구성 요소에서 언어 속성이 표시됩니다.

단계 4

**데이터폴** 구성 요소에서 Value 텍스트 상자에 Welcome을 입력합니다.

콘텐츠 영역에서 레이블에 Welcome가 표시됩니다.

단계 5

**속성** 구성 요소로 이동합니다.

단계 6

German 텍스트 상자에 Willkommen을 입력합니다.

왼쪽 위 모서리의 Language 상자에서 언어를 German으로 변경합니다.

콘텐츠 영역에서 레이블에 Willkommen가 표시됩니다.

단계 7

French 텍스트 상자에 Bienvenue를 입력합니다.

왼쪽 위 모서리의 Language 상자에서 언어를 French으로 변경합니다.

콘텐츠 영역에서 레이블에 Bienvenue가 표시됩니다.

영어, 독일어 및 프랑스어에 대한 언어 지원이 추가되었으며 언어별 텍스트 레이블이 정의되었습니다.



## 런타임 중에 언어 변경

다음 지침에서는 런타임 중에 언어를 변경하기 위해 스크립트를 만드는 프로세스를 안내합니다. 사용자가 레이블을 클릭할 때마다 표시 언어가 변경됩니다.

선수학습:

- 이 단계를 수행하려면 이전 지침의 단계를 완료한 상태여야 합니다.

단계 1

**데이터풀** 구성 요소에서 **+**를 클릭합니다.

메뉴가 확장됩니다.

단계 2

메뉴에서 Integer를 클릭합니다.

Integer 유형의 데이터풀 항목이 추가됩니다.

단계 3

데이터풀 항목의 이름을 SelectedLanguage로 바꿉니다.

단계 4

**내비게이션** 구성 요소에서 Label 1 레이블을 클릭합니다.

단계 5

**속성** 구성 요소에서 **위젯 피쳐 속성**으로 이동한 다음 **추가/제거**를 클릭합니다.

**위젯 피쳐** 대화 상자가 표시됩니다.

단계 6

**사용 가능한 위젯 피쳐** 아래에서 **입력 처리** 범주를 확장하고 **터치 누름** 위젯 피쳐를 선택합니다.

단계 7

**수락**을 클릭합니다.

관련 위젯 피쳐 속성이 레이블에 추가되고 **속성** 구성 요소에 표시됩니다.

단계 8

touchPressed 속성 옆의 **편집**을 클릭합니다.

단계 9

기존 EB GUIDE 스크립트를 다음 코드로 바꿉니다.



```
function(v:touchId::int, v:x::int, v:y::int, v:fingerId::int)
{
  if (dp:SelectedLanguage == 0) // Standard selected
  {
    f:language(!:German)
    dp:SelectedLanguage = 1
  }
  else if (dp:SelectedLanguage == 1) // German selected
  {
    f:language(!:French)
    dp:SelectedLanguage = 2
  }
  else if (dp:SelectedLanguage == 2) // French selected
  {
    f:language(!:Standard)
    dp:SelectedLanguage = 0
  }
  false
}
```

#### 단계 10

수락을 클릭합니다.

레이블을 구성하고 런타임에 레이블의 언어를 변경하는 EB GUIDE 스크립트를 작성했습니다.

결과:

String 유형의 데이터폴 항목을 EB GUIDE에 추가했습니다. 이 데이터폴 항목에는 여러 언어에 대해 각기 다른 값이 포함되어 있습니다. 영어의 값은 Welcome입니다. 독일어의 값은 Willkommen입니다. 프랑스어의 값은 Bienvenue입니다. 데이터폴 항목이 레이블의 text 속성에 연결되었습니다. EB GUIDE 모델의 언어를 변경할 때마다 레이블의 텍스트도 변경됩니다.

## 11.7. 자습서: 3D 그래픽으로 작업

### [참고]



### 기본 창 레이아웃

이 사용자 설명서의 모든 지침 및 스크린샷이 기본 창 레이아웃을 사용합니다. 지침을 따르는 경우 EB GUIDE Studio 창을 기본 레이아웃으로 설정하는 것이 좋습니다. **레이아웃 > 기본 레이아웃으로 다시 설정**을 선택하여 설정할 수 있습니다.

EB GUIDE Studio에서는 EB GUIDE 모델에 3D 그래픽을 사용할 수 있습니다.

다음 지침에서는 EB GUIDE 모델에 3D 그래픽을 추가하는 프로세스를 안내합니다. 이 지침에는 위젯 피처를 사용하여 3D 그래픽을 내보내는 방법 및 내보낸 3D 그래픽의 모양을 수정하는 방법이 나와 있습니다. 최적의 결과를 얻으려면 다음 단계를 아래에 나와 있는 순서대로 진행하십시오.

**[참고]****3D 그래픽**

3D 그래픽 파일을 만들려면 서드파티 3D 모델링 소프트웨어를 사용하십시오.

OpenGL ES 버전 2.0 이상 및 DirectX 11 렌더러만 3D 그래픽을 표시할 수 있습니다. 그래픽 드라이버가 렌더러의 버전과 호환되는지 확인합니다. 지원되는 3D 그래픽 형식은 COLLADA(.dae) 및 Filmbox(.fbx)입니다. 최적의 결과를 얻으려면 Filmbox 형식을 사용하십시오.

질감을 메시에 적용하도록 하려면 3D 객체에 질감 좌표가 있어야 합니다. 질감 좌표를 추가하려면 서드파티 3D 모델링 소프트웨어를 사용하십시오.

대략적인 소요 시간: 15분.

**3D 그래픽 가져오기**

다음 지침에서는 3D 그래픽 파일을 EB GUIDE Studio 프로젝트에 가져오는 프로세스를 안내합니다.

선수학습:

- 콘텐츠 영역에 **기본** 상태 머신이 표시되어 있어야 합니다.
- **기본** 상태 머신에 초기 상태와 보기 상태가 포함되어 있어야 합니다.
- 초기 상태에 보기 상태로의 전이가 있어야 합니다.
- 3D 그래픽 파일을 사용할 수 있습니다. 이 파일에는 카메라, 광원 및 메시와 하나 이상의 재료가 있는 개체 하나가 포함되어 있어야 합니다.

단계 1

콘텐츠 영역에서 보기 상태를 더블 클릭합니다.

보기가 콘텐츠 영역에 표시됩니다.

단계 2

장면 그래프를 **도구 상자**에서 보기로 끕니다.

보기에 비어 있는 경계 상자가 표시됩니다.

단계 3

장면 그래프 이름을 My3DGraphic으로 바꿉니다.

단계 4

**속성** 구성 요소에서 **파일 가져오기**를 클릭합니다.

대화 상자가 열립니다.

단계 5

3D 그래픽 파일이 저장되어 있는 디렉터리로 이동합니다.

단계 6

3D 그래픽 파일을 선택합니다.

단계 7

**열기**를 클릭합니다.

가져오기가 시작됩니다. **가져오기 성공** 대화 상자가 표시됩니다. 여기에서 가져오기 로그 파일을 확인할 수 있습니다.

단계 8

**확인**을 클릭합니다.

보기에 3D 그래픽이 표시됩니다. **내비게이션** 구성 요소에 장면 그래프를 사용하여 가져온 위젯 트리가 상위 노드로 표시됩니다. My3DGraphic에는 3D 그래픽 파일의 콘텐츠에 따라 재료, 카메라, 몇몇 기타 하위 위젯을 사용한 메시가 하나 이상 있는 RootNode가 포함되어 있습니다.



## 위젯 추가

다음 지침에서는 3D 그래픽에 추가 2차 광원을 추가하는 프로세스를 안내합니다.

선수학습:

- 이 단계를 수행하려면 이전 지침의 단계를 완료한 상태여야 합니다.

단계 1

**내비게이션** 구성 요소에서 RootNode를 확장합니다.

단계 2

**도구 상자**의 방향 조명을 RootNode로 끕니다.

방향 조명을 My3DGraphic에 추가했습니다. RootNode의 변환 속성으로 이 방향 조명을 조작 및 변환할 수 있습니다.

단계 3

광원을 추가하고 RootNode 장면 그래프와 다른 기본 위젯 속성으로 배치하려면 다음 작업을 수행합니다.

단계 3.1

**도구 상자**의 장면 그래프 노드를 RootNode로 끕니다.

단계 3.2

장면 그래프 노드 이름을 MyLight로 바꿉니다.

단계 3.3

**도구 상자**의 방향 조명을 MyLight로 끕니다.

방향 조명을 My3DGraphic에 추가했습니다. 방향 조명의 배치를 바꾸려면 MyLight의 속성을 바꿉니다.



## 메시 변경

선수학습:

- 이 단계를 수행하려면 이전 지침의 단계를 완료한 상태여야 합니다.
- \$GUIDE\_PROJECT\_PATH/<project name>/resources/<3D graphic name> 디렉터리에 추가 .ebmesh 파일이 포함되어 있습니다.

단계 1

**내비게이션** 구성 요소에서 Mesh 1을 클릭하고 **속성** 구성 요소로 이동합니다.

단계 2

mesh 콤보 박스에서 위에 언급된 리소스 폴더의 .ebmesh 파일을 선택합니다.

보기에 새 메시가 적용된 장면 그래프가 표시됩니다.

단계 3

또는 .ebmesh 파일을 **자산** 구성 요소에서 mesh 드롭다운 목록 상자로 끕니다.

보기에 새 메시가 적용된 장면 그래프가 표시됩니다.



## 질감 변경

다음 지침에서는 3D 그래픽의 질감을 추가 및 수정하는 프로세스를 안내합니다.

선수학습:

- 이 단계를 수행하려면 이전 지침의 단계를 완료한 상태여야 합니다.
- \$GUIDE\_PROJECT\_PATH/<project name>/resources/<3D graphic name> 디렉터리에 .png 또는 .jpg 이미지 파일이 포함되어 있습니다.

단계 1

**내비게이션** 구성 요소에서 재료를 클릭하고 **속성** 구성 요소로 이동합니다.

단계 2

**위젯 피쳐 속성** 범주에서 **추가/제거**를 클릭합니다.

**위젯 피쳐** 대화 상자가 표시됩니다.

단계 3

**사용 가능한 위젯 피쳐**에서 **3D** 범주를 확장하고 **확산 질감** 등의 질감 위젯 피쳐를 선택합니다.

단계 4

**수락**을 클릭합니다.

관련 위젯 피쳐 속성이 질감에 추가되고 **속성** 구성 요소에 표시됩니다.

**단계 5**

**속성** 구성 요소의 diffuseTexture 콤보 박스에서 이미지를 선택합니다.

보기에 새 질감이 적용된 장면 그래프가 표시됩니다.

**[참고]****3D 위젯 피쳐 사용법**

이러한 지침은 **3D** 범주의 다음과 같은 위젯 피쳐에 대해 유효합니다.

- ▶ 주변 질감
- ▶ 확산 질감
- ▶ 방사 질감
- ▶ 조명 지도 질감
- ▶ 일반 지도 질감
- ▶ 불투명 질감
- ▶ 반사 질감
- ▶ 거울 질감

**3D 객체 여러 번 표시**

다음 지침에서는 3D 그래픽의 3D 객체를 여러 번 표시하도록 할 추가 카메라를 추가하는 프로세스를 안내합니다. 동일한 객체에 대한 다양한 관점을 가질 수 있습니다.

선수학습:

- 이 단계를 수행하려면 이전 지침의 단계를 완료한 상태여야 합니다.

**단계 1**

**내비게이션** 구성 요소에서 My3DGraphic을 클릭하고 **속성** 구성 요소로 이동합니다.

**단계 2**

width 텍스트 상자에 800을, height 텍스트 상자에 480을 입력합니다.

My3DGraphic 장면 그래프에 보기 크기가 있습니다.

**단계 3**

**내비게이션** 구성 요소에서 RootNode 및 Camera001을 확장합니다.

**단계 4**

Camera 1를 클릭하고 **속성** 구성 요소로 이동합니다.

**단계 5**

**위젯 피쳐 속성** 범주에서 **추가/제거**를 클릭합니다.

**위젯 피쳐** 대화 상자가 표시됩니다.

**단계 6**

사용 가능한 위젯 피쳐 아래에서 **3D** 범주를 확장하고 **카메라 뷰포트**를 선택합니다.

**단계 7**

수락을 클릭합니다.

관련 위젯 피쳐 속성이 Camera 1에 추가되고 **속성** 구성 요소에 표시됩니다.

**단계 8**

**도구 상자**의 카메라를 Camera001.장면 그래프 노드로 끕니다.

보조 카메라를 추가했습니다.

**단계 9**

Camera 2를 클릭하고 **속성** 구성 요소로 이동합니다.

**단계 10**

nearPlane, farPlane 및fieldOfView 텍스트 상자에서 Camera 1의 값과 같은 값을 입력합니다.

Camera 1 및 Camera 2에 모두 같은 보기 위치가 있습니다.

**단계 11**

**위젯 피쳐 속성** 범주에서 **추가/제거**를 클릭합니다.

**위젯 피쳐** 대화 상자가 표시됩니다.

**단계 12**

사용 가능한 위젯 피쳐 아래에서 **3D** 범주를 확장하고 **카메라 뷰포트**를 선택합니다.

**단계 13**

수락을 클릭합니다.

관련 위젯 피쳐 속성이 Camera 2에 추가되고 **속성** 구성 요소에 표시됩니다.

**단계 14**

**속성** 구성 요소에서 viewportX 및 viewportY 텍스트 상자에 100을 입력합니다.

보기에서 서로 다른 x 좌표와 y 좌표를 사용해 3D 객체가 두 번 표시됩니다.

## 11.8. 자습서: 감마를 올바르게 렌더링하기

### [참고]



### 기본 창 레이아웃

이 사용자 설명서의 모든 지침 및 스크린샷이 기본 창 레이아웃을 사용합니다. 지침을 따르는 경우 EB GUIDE Studio 창을 기본 레이아웃으로 설정하는 것이 좋습니다. **레이아웃 > 기본 레이아웃**으로 다시 설정을 선택하여 설정할 수 있습니다.

EB GUIDE Studio에서는 다음에 대한 감마 정정을 수행할 수 있습니다.

- ▶ 디스플레이

- ▶ 이미지
- ▶ 질감

다음 지침은 감마 정정을 구성하는 방법을 안내합니다. 최상의 결과를 얻으려면 단계를 아래에 나와 있는 순서대로 진행하십시오.

대략적인 소요 시간: 15분




### 디스플레이용 감마 인코딩 구성

다음 단계는 감마 인코딩 값을 디스플레이에 출력할 수 있도록 EB GUIDE Studio를 설정하는 프로세스를 안내합니다.

선수학습:

- 콘텐츠 영역에 **기본** 상태 머신이 표시되어 있어야 합니다.
- **기본** 상태 머신에 초기 상태와 보기 상태가 포함되어 있어야 합니다.
- 초기 상태에 보기 상태로의 전이가 있어야 합니다.

#### 단계 1

를 클릭합니다. 프로젝트 센터가 열립니다.

#### 단계 2

내비게이션 영역에서 **구성 > 프로필**을 클릭합니다.

#### 단계 3

sRGB를 인식하는 각 프로필에 대해 다음을 구성하십시오.

##### 단계 3.1

콘텐츠 영역에서 **장면** 탭을 클릭합니다.

##### 단계 3.2

colorMode 드롭다운 목록 상자에서 다음을 선택합니다.

- ▶ 렌더링 파이프라인이 GPU의 하드웨어 sRGB 지원을 이용하는 경우 32-bit sRGB (=4)를 선택합니다.
- ▶ 렌더링 하드웨어가 sRGB를 지원하지 않는 경우 32-bit sRGB (=5)를 선택합니다.

편집 모드에서 변경 사항을 적용하려면 EB GUIDE Studio를 다시 시작합니다.

**[참고]**



**렌더링 하드웨어**

DirectX 렌더러 및 OpenGL 3 렌더러는 언제나 하드웨어 sRGB 지원을 받습니다. OpenGL ES 2.0 API를 사용하는 OpenGL 렌더러는 하드웨어가 적합한 OpenGL ES 확장을 통해 이를 지원하는 경우 하드웨어 sRGB 지원만을 이용합니다. 이는 자동으로 감지됩니다. OpenGL ES 2.0 하드웨어가 sRGB를 지원하지 않는 경우 렌더러가 변환을 위해 픽셀 셰이더를 사용하는 32-bit sRGB (Emulated)(=5)로 자동으로 되돌아갑니다. 하드웨어 sRGB 모드는 OpenGL ES 3.0을 지원한다고 하더라도 모든 시스템에서 작동하지는 않습니다. 그런 경우 32-bit sRGB (Emulated)(=5)로 전환합니다.



**이미지용 감마 인코딩 구성**

선수학습:

- 이 단계를 수행하려면 이전 지침의 단계를 완료한 상태여야 합니다.
- 리소스 디렉토리에 이용 가능한 이미지 파일이 있습니다.

단계 1

프로젝트 편집기로 이동해 보기를 더블 클릭합니다. 이미지를 **도구 상자**에서 보기로 끕니다.

단계 2

**속성** 구성 요소의 image 드롭다운 목록 상자에서 이미지 파일을 선택합니다.

이미지가 너무 밝아 보일 수 있습니다.

이는 장면 속성에서 구성되었던 sRGB 색상 모드에 의해 유발되며 이제 이미 감마 정정된 이미지에 감마 정정을 적용합니다.

단계 3

이미지가 감마 인코딩되도록 구성하려면 **속성** 구성 요소에서 sRGB를 선택합니다.

이제 이미지가 블렌딩 작업에서 올바르게 표시 및 처리됩니다.



**질감용 감마 인코딩 구성**

선수학습:

- 이전 지침의 단계를 완료하셨습니다.
- 최소한 1개의 질감 처리된 3D 개체가 포함된 3D 파일인 .fbx 파일은 확산 질감과 함께 사용 가능합니다.

단계 1

장면 그래프를 **도구 상자**에서 보기로 끕니다.



단계 2

**속성** 구성 요소에서 **파일 가져오기**를 클릭하고 3D 파일을 선택합니다.

단계 3

**내비게이션** 구성 요소에서 가져온 장면 그래프를 찾습니다. 장면 그래프 구조 내에서 확산 질감을 사용하는 재료 위젯을 선택합니다.

단계 4

**속성** 구성 요소의 위젯 **피쳐 속성**에서 **추가/제거**를 클릭합니다.

단계 5

**3D** 범주에서 **재료 확산**을 선택합니다.

단계 6

diffuseSRGB 속성을 선택합니다.

질감은 감마 인코딩된 이미지로 취급되며 조명 계산에 사용되기 전에 선형화됩니다.

## 11.9. 자습서: 이미지 기반 조명 사용

### [참고]



### 기본 창 레이아웃

이 사용자 설명서의 모든 지침 및 스크린샷이 기본 창 레이아웃을 사용합니다. 지침을 따르는 경우 EB GUIDE Studio 창을 기본 레이아웃으로 설정하는 것이 좋습니다. **레이아웃 > 기본 레이아웃**으로 다시 설정을 선택하여 설정할 수 있습니다.

다음 지침은 EB GUIDE Studio의 장면 그래프를 위해 이미지 기반 조명을 사용하는 것의 프로세스를 안내합니다.

대략적인 소요 시간: 10분



도움말 지침에 IBLGenerator 액세스하는 중

다음 단계에서 IBLGenerator을 시작합니다.

단계 1

명령줄의 도움으로 IBLGenerator.exe 실행 파일이 포함된 디렉터리로 이동하십시오.

예: C:/Program Files/Elektrobit/EB GUIDE Studio 6.7/platform/win32

단계 2

IBLGenerator.exe -h를 입력합니다.

IBLGenerator의 모든 옵션이 열거됩니다.



## .ebibl 파일 생성 중

선수학습:

- 이 단계를 수행하려면 이전 지침의 단계를 완료한 상태여야 합니다.
- 입력 데이터로 .pfm 파일이 있습니다.

단계 1

IBLGenerator 도움말에 따라 원하는 매개변수와 명령을 입력합니다.

예:

```
IBLGenerator.exe -i "MyInputPath\MyInputFile.pfm" -o
    "MyOutputPath\MyOutputFile.ebibl" -p cube -q 3
```

이 명령은 데이터를 정육면체 매개변수화에 저장하는 MyInputFile.pfm 입력 파일을 통해 IBLGenerator을 실행합니다. 품질 수준은 3으로 설정되어 있으며 결과는 MyOutputFile.ebibl에 저장됩니다.



## 이미지 기반 조명 적용

선수학습:

- 이 단계를 수행하려면 이전 지침의 단계를 완료한 상태여야 합니다.
- IBLGenerator을 사용하면 .ebibl 파일이 생성됩니다.
- .ebibl 파일은 현재 프로젝트의 리소스 디렉터리 내에 배치됩니다.
- 프로젝트가 EB GUIDE Studio에서 열립니다.
- 보기 상태가 추가됩니다.
- 장면 그래프가 추가되고 3D 파일이 가져오기됩니다.

단계 1

가져온 장면 그래프의 장면 그래프 노드에 이미지 기반 조명을 추가합니다.

단계 2

**속성** 구성 요소로 이동합니다.

단계 3

ibl 드롭다운 목록 상자에서 입력으로 MyOutputFile.ebibl을 선택합니다.

단계 4

최상의 결과를 위해 이미지 기반 조명의 속성을 조정하고 PBR GGX 재료 또는 PBR Phong 재료를 사용합니다.

이미지 기반 조명으로 장면만 비추기 위해 다른 모든 광원은 비활성화합니다.

장면 그래프에 하나 이상의 이미지 기반 조명이 추가된 경우 계층상 첫 번째 조명만 장면에 영향을 미칩니다.

**[참고]****OpenGL ES 2.0의 이미지 기반 조명**

이미지 기반 조명을 사용하기 위해서는 OpenGL 렌더러 사용 시 OpenGL ES 2.0 드라이버가 GL\_EXT\_shader\_texture\_lod, GL\_EXT\_texture\_rg, GL\_OES\_texture\_float 및 GL\_OES\_texture\_half\_float OpenGL 확장을 지원해야 합니다.

DirectX 렌더러 및 OpenGL 3 렌더러를 사용하는 경우 이미지 기반 조명이 언제나 지원됩니다.

# 12. 참조

다음 장에서는 예를 들어 매개변수, 속성, 식별자 등이 포함된 목록 및 테이블을 소개합니다.

## 12.1. Android 이벤트

Android 이벤트는 SystemNotifications 이벤트 그룹에 속하며, 이벤트 그룹 ID는 13입니다.

표 12.1. Android 이벤트

이벤트 ID	이름	설명
1	RendererEnabled	<p>Android 라이프 사이클 관리에서 렌더러를 시작하거나 중지하면 응용 프로그램에 의해 전송됩니다.</p> <p>매개변수:</p> <ul style="list-style-type: none"> <li>▶ enabled: true인 경우 렌더러가 활성화됩니다. false인 경우 렌더러가 절전 모드로 설정됩니다.</li> </ul>
2	setKeyboardVisibility	<p>가상 키보드를 표시해야 하는 경우 EB GUIDE 모델에 의해 전송됩니다.</p> <p>매개변수:</p> <ul style="list-style-type: none"> <li>▶ visibility: true인 경우 가상 키보드가 표시됩니다. false인 경우 가상 키보드가 표시되지 않습니다.</li> </ul>
3	onKeyboardVisibilityChanged	<p>가상 키보드가 표시되면 응용 프로그램에 의해 전송됩니다.</p> <p>매개변수:</p> <ul style="list-style-type: none"> <li>▶ visibility: true인 경우 가상 키보드가 표시됩니다. false인 경우 가상 키보드가 표시되지 않습니다.</li> </ul>
4	onLayoutChanged	<p>화면의 표시 영역이 변경되면 응용 프로그램에 의해 전송됩니다.</p> <p>매개변수(픽셀):</p> <ul style="list-style-type: none"> <li>▶ x: 표시되는 화면 영역 왼쪽 위 모서리의 x 좌표입니다.</li> </ul>

이벤트 ID	이름	설명
		<ul style="list-style-type: none"> <li>▶ y: 표시되는 화면 영역 왼쪽 위 모서리의 y 좌표입니다.</li> <li>▶ width: 표시되는 화면 영역의 너비입니다.</li> <li>▶ height: 표시되는 화면 영역의 높이입니다.</li> </ul>

## 12.2. 데이터폴 항목

표 12.2. 데이터폴 항목의 속성

속성 이름	설명
Value	데이터폴 항목의 초기 값입니다.

## 12.3. 데이터 유형

다음 섹션에서는 EB GUIDE의 데이터 유형에 대해 설명합니다. 아래에 나와 있는 유형에서 사용자 정의 속성 및 데이터폴 항목을 추가할 수 있습니다.

### 12.3.1. 부울

부울 속성의 값은 true 또는 false일 수 있습니다.

사용 가능한 작업은 다음과 같습니다.

- ▶ 같음(==)
- ▶ 같지 않음(!=)
- ▶ 부정(!)
- ▶ 및(&&)
- ▶ 또는(||)
- ▶ 지정(쓰기 가능 속성)(=)

목록에 부울 속성을 저장할 수 있습니다. 목록에 대한 자세한 내용은 [섹션 12.3.12, "목록"](#) 항목을 참조하십시오.

## 12.3.2. 색

색은 RGBA8888 형식으로 저장됩니다.

예: 투명도가 없는 빨간색은 (255, 0, 0, 255)입니다.

사용 가능한 작업은 다음과 같습니다.

- ▶ 같음(==)
- ▶ 같지 않음(!=)
- ▶ 지정(쓰기 가능 속성)(=)

목록에 색 속성을 저장할 수 있습니다. 목록에 대한 자세한 내용은 [섹션 12.3.12, "목록"](#) 항목을 참조하십시오.

## 12.3.3. 조건부 스크립트

조건부 스크립트는 초기화 및 트리거에 반응하는 데 사용됩니다. 조건부 스크립트를 편집할 때 콘텐츠 영역은 다음 섹션으로 구분됩니다.

- ▶ **트리거** 섹션에서 **On trigger** 스크립트의 실행을 트리거하는 이벤트, 데이터폴 항목 또는 위젯 속성을 추가할 수 있습니다.
- ▶ **On trigger** 섹션에서 초기화 시 이벤트 트리거 또는 데이터폴 항목이나 위젯 속성의 값이 업데이트된 이후에 호출되는 EB GUIDE 스크립트를 추가할 수 있습니다.

**On trigger**의 EB GUIDE 스크립트 매개변수는 스크립트가 실행된 이유를 나타냅니다.

arg0은 초기화 중 또는 트리거에 의해 EB GUIDE 스크립트가 실행되는지 여부를 참조합니다. 다음 사항을 고려하십시오.

- ▶ 초기화 중에 EB GUIDE 스크립트가 실행되면 arg0은 true입니다.
- ▶ EB GUIDE 스크립트가 트리거에 의해 실행되면 arg0은 false입니다.

**On trigger** EB GUIDE 스크립트의 반환 값은 속성의 변경 알림을 제어합니다.

**On trigger** EB GUIDE 스크립트의 반환 값은 EB GUIDE 스크립트가 알림을 작성해야 하는지 여부를 규정합니다. 다음 사항을 고려하십시오.

- ▶ 반환 값이 true인 경우 알림이 생성됩니다.

- ▶ 반환 값이 false인 경우 알림이 생성되지 않습니다.

**On trigger** 스크립트를 실행할 수 있으려면 다음 중에 조건이 충족되어야 합니다.

- ▶ 초기화 시, EB GUIDE 모델 시작 중의 데이터폴 항목이나 보기 작성 중의 위젯 속성을 예로 들 수 있습니다.
- ▶ 트리거 목록에서 이벤트를 처리 중입니다. EB GUIDE 스크립트는 일치하는 각 이벤트에 대해 한 번씩 실행됩니다.
- ▶ 트리거 목록에서 하나 이상의 데이터폴 알림을 처리 중입니다. 여러 알림이 동시에 처리될 수 있습니다.
- ▶ 트리거 목록에서 하나 이상의 위젯 속성의 알림을 처리 중입니다. 여러 알림이 동시에 처리될 수 있습니다.

### 12.3.4. 부동

부동 소수점 숫자 데이터 유형은 단정밀도 32비트 IEEE 754 값을 나타냅니다.

사용 가능한 작업은 다음과 같습니다.

- ▶ 같음(==)
- ▶ 같지 않음(!=)
- ▶ 큼(>)
- ▶ 크거나 같음(>=)
- ▶ 작음(<)
- ▶ 작거나 같음(<=)
- ▶ 더하기(+)
- ▶ 빼기(-)
- ▶ 곱하기(\*)
- ▶ 나누기(/)
- ▶ 지정(쓰기 가능 속성)(=)

목록에 부동 속성을 저장할 수 있습니다. 목록에 대한 자세한 내용은 [섹션 12.3.12. "목록"](#) 항목을 참조하십시오.

### 12.3.5. 글꼴

EB GUIDE 프로젝트에 글꼴을 추가하려면 다음 디렉터리에서 글꼴 파일을 복사하십시오. \$GUIDE\_PROJECT\_PATH/<project name>/resources

사용 가능한 작업은 다음과 같습니다.

- ▶ 지정(쓰기 가능 속성)(=)

목록에 글꼴 속성을 저장할 수 있습니다. 목록에 대한 자세한 내용은 [섹션 12.3.12. "목록"](#) 항목을 참조하십시오.

### 12.3.6. Function () : bool

Function () : bool을 이용해 자신만의 기능을 생성할 수 있습니다.

이 데이터 유형에 대해 사용할 수 있는 작업은 모든 속성의 작업의 읽기/실행입니다.

### 12.3.7. Ibl

Ibl은 IBLGenerator에 의해 생성된 조명 정보를 보관하는 데이터 형식입니다.

EB GUIDE 프로젝트에 Ibl을 추가하려면 \$GUIDE\_PROJECT\_PATH/<project name>/resources 디렉터리에서 .-ebibl 파일을 복사하십시오.

사용 가능한 작업은 다음과 같습니다.

- ▶ 지정(쓰기 가능 속성)(=)

목록에 Ibl 속성을 저장할 수 있습니다. 목록에 대한 자세한 내용은 [섹션 12.3.12. "목록"](#) 항목을 참조하십시오.

### 12.3.8. 이미지

EB GUIDE 프로젝트에 이미지를 추가하려면 다음 디렉터리에서 이미지 파일을 복사하십시오. \$GUIDE\_PROJECT\_PATH/<project name>/resources

사용 가능한 작업은 다음과 같습니다.

- ▶ 지정(쓰기 가능 속성)(=)

목록에 이미지 속성을 저장할 수 있습니다. 목록에 대한 자세한 내용은 [섹션 12.3.12. "목록"](#) 항목을 참조하십시오.

### 12.3.9. 정수



EB GUIDE에서는 서명된 32비트 정수를 지원합니다.

사용 가능한 작업은 다음과 같습니다.

- ▶ 같음(==)
- ▶ 같지 않음(!=)
- ▶ 큼(>)
- ▶ 크거나 같음(>=)
- ▶ 작음(<)
- ▶ 작거나 같음(<=)
- ▶ 더하기(+)
- ▶ 빼기(-)
- ▶ 곱하기(\*)
- ▶ 나누기(/)
- ▶ 모듈로(%)
- ▶ 지정(쓰기 가능 속성)(=)

목록에 정수 속성을 저장할 수 있습니다. 목록에 대한 자세한 내용은 [섹션 12.3.12. "목록"](#) 항목을 참조하십시오.

## 12.3.10. 메시

메시는 3D 객체의 도형을 정의합니다.

EB GUIDE 프로젝트에 메시지를 추가하려면 \$GUIDE\_PROJECT\_PATH/<project name>/resources 디렉터리에서 .ebmesh 파일을 복사하십시오.

사용 가능한 작업은 다음과 같습니다.

- ▶ 지정(쓰기 가능 속성)(=)

목록에 메시 속성을 저장할 수 있습니다. 목록에 대한 자세한 내용은 [섹션 12.3.12. "목록"](#) 항목을 참조하십시오.

## 12.3.11. 문자열

예를 들어 EB GUIDE에서는 Hello world와 같은 문자열을 지원합니다.

사용 가능한 작업은 다음과 같습니다.

- ▶ 같음(대/소문자 구분)(==)
- ▶ 같지 않음(대/소문자 구분)(!=)
- ▶ 같음(대/소문자 구분 안 함, ASCII 범위만 해당)(=Aa=)
- ▶ 큼(>)
- ▶ 크거나 같음(>=)
- ▶ 작음(<)
- ▶ 작거나 같음(<=)
- ▶ 연결(+)
- ▶ 지정(쓰기 가능 속성)(=)

목록에 문자열 속성을 저장할 수 있습니다. 목록에 대한 자세한 내용은 [섹션 12.3.12. "목록"](#) 항목을 참조하십시오.

## 12.3.12. 목록

EB GUIDE에서는 데이터 유형이 같은 값 목록을 지원합니다.

다음 목록 유형을 사용할 수 있습니다.

- ▶ 부울 목록
- ▶ 색 목록
- ▶ 부동 목록
- ▶ 글꼴 목록
- ▶ lbl 목록
- ▶ 이미지 목록
- ▶ 정수 목록
- ▶ 메시 목록
- ▶ 문자열 목록

목록에서 사용할 수 없는 유형은 다음과 같습니다.

- ▶ 목록
- ▶ 속성 참조
- ▶ 목록 요소 참조

사용 가능한 작업은 다음과 같습니다.



- ▶ 길이: (길이)
- ▶ 요소 접근자: ([])

## 12.4. EB GUIDE 스크립트

### 12.4.1. EB GUIDE 스크립트 키워드

아래에는 EB GUIDE 스크립트의 예약된 키워드 목록이 나와 있습니다. 이러한 단어를 스크립트에서 식별자로 사용하려는 경우 따옴표로 묶어야 합니다.

키워드	설명
color:	예를 들어 뒤에 {0,255,255}와 같은 색 매개변수가 옵니다.
dp:	뒤에 데이터폴 항목이 옵니다.
l:	뒤에 언어가 옵니다.
else	if 조건이 완료되었습니다. 대신 다음 블록이 실행됩니다.
ev:	뒤에 이벤트가 옵니다.
f:	뒤에 사용자 정의 함수가 옵니다.
false	부울 리터럴 값입니다.
fire	이벤트를 실행합니다.
if	뒤에 부울 식을 테스트하는 문이 옵니다. 식이 true이면 문이 실행됩니다.
in	로컬 변수 선언과 변수의 사용 범위 간 구분 기호입니다. match_event 및 let과 함께 사용됩니다.
function	함수를 선언합니다.
length	속성의 길이입니다.
let	범위에서 액세스할 수 있는 로컬 변수를 선언합니다.
list	예를 들어 정수 목록 등의 목록 유형을 선언합니다.
match_event	현재 이벤트가 필요한 이벤트에 해당하는지를 확인하고 let 등의 변수를 선언합니다.
popup_stack	동적 상태 기계의 우선순위를 정의하는 동적 상태 기계 목록입니다.
sm:	뒤에 상태 기계가 옵니다.
true	부울 리터럴 값입니다.

키워드	설명
unit	void 유형의 값입니다.
v:	뒤에 로컬 변수가 옵니다.
while	조건이 true이면 문을 반복합니다.

## 12.4.2. EB GUIDE 스크립트 연산자 우선순위

아래에는 EB GUIDE 스크립트의 연산자와 해당 우선순위 및 연결 위치의 목록이 나와 있습니다. 연산자 우선 순위는 내림차순(위에서 아래로)으로 나열됩니다.

표 12.3. EB GUIDE 스크립트 연산자 우선순위

연산자	연결 위치
(( ), { })	없음
([])	없음
(->)	왼쪽
(.)	없음
::	왼쪽
length	없음
(&)	오른쪽
(!), (-) 단항 빼기	오른쪽
(*), (/), (%)	왼쪽
(+), (-)	왼쪽
(<), (>), (<=), (>=)	왼쪽
(!=), (==), (=Aa=)	왼쪽
(&&)	왼쪽
(  )	왼쪽
(=), (+=), (-=), (=>)	오른쪽
(,)	오른쪽
(;)	왼쪽

## 12.4.3. EB GUIDE 스크립트 표준 라이브러리

다음 장에서는 모든 EB GUIDE 스크립트 함수에 대해 설명합니다.

### 12.4.3.1. EB GUIDE 스크립트 함수 A - B

#### 12.4.3.1.1. abs

이 함수는 정수 x의 절대값을 반환합니다.

표 12.4. abs의 매개변수

매개변수	유형	설명
x	정수	절대값을 반환할 숫자
<return>	정수	반환 값

#### 12.4.3.1.2. absf

이 함수는 부동 숫자 x의 절대값을 반환합니다.

표 12.5. absf의 매개변수

매개변수	유형	설명
x	float	절대값을 반환할 숫자
<return>	float	반환 값

#### 12.4.3.1.3. acosf

이 함수는 x의 아크코사인 주요 값을 반환합니다.

표 12.6. acosf의 매개변수

매개변수	유형	설명
x	float	아크코사인을 반환할 숫자
<return>	float	반환 값

#### 12.4.3.1.4. animation\_before

이 함수는 실행되는 애니메이션이 지정된 포인트를 통과했는지를 확인합니다.

표 12.7. animation\_before의 매개변수

매개변수	유형	설명
animation	GtfTypeRecord	조작할 애니메이션

매개변수	유형	설명
time	정수	포인트
<return>	boolean	true인 경우 애니메이션이 지정된 포인트를 아직 통과하지 않은 것입니다.

#### 12.4.3.1.5. animation\_beyond

이 함수는 실행되는 애니메이션이 지정된 포인트를 통과했는지를 확인합니다.

표 12.8. animation\_beyond의 매개변수

매개변수	유형	설명
animation	GtfTypeRecord	조작할 애니메이션
time	정수	포인트
<return>	boolean	true인 경우 애니메이션이 지정된 포인트를 통과한 것입니다.

#### 12.4.3.1.6. animation\_cancel

이 함수는 애니메이션을 취소하고 편집된 속성을 현재 상태로 유지합니다.

표 12.9. animation\_cancel의 매개변수

매개변수	유형	설명
animation	GtfTypeRecord	조작할 애니메이션
<return>	boolean	true인 경우 함수가 정상적으로 실행된 것입니다.

#### 12.4.3.1.7. animation\_cancel\_end

이 함수는 애니메이션을 취소하고 가능한 경우 편집된 속성을 종료 상태로 설정합니다.

표 12.10. animation\_cancel\_end의 매개변수

매개변수	유형	설명
animation	GtfTypeRecord	조작할 애니메이션
<return>	boolean	true인 경우 함수가 정상적으로 실행된 것입니다.

#### 12.4.3.1.8. animation\_cancel\_reset

이 함수는 애니메이션을 취소하고 가능한 경우 편집된 속성을 초기 상태로 다시 설정합니다.



표 12.11. animation\_cancel\_reset의 매개변수

매개변수	유형	설명
animation	GtfTypeRecord	조작할 애니메이션
<return>	boolean	true인 경우 함수가 정상적으로 실행된 것입니다.

#### 12.4.3.1.9. animation\_pause

이 함수는 애니메이션을 일시 중지합니다.

표 12.12. animation\_pause의 매개변수

매개변수	유형	설명
animation	GtfTypeRecord	조작할 애니메이션
<return>	boolean	true인 경우 함수가 정상적으로 실행된 것입니다.

#### 12.4.3.1.10. animation\_play

이 함수는 애니메이션을 시작하거나 계속합니다.

표 12.13. animation\_play의 매개변수

매개변수	유형	설명
animation	GtfTypeRecord	조작할 애니메이션
<return>	boolean	true인 경우 애니메이션이 아직 실행되고 있지 않은 것입니다.

#### 12.4.3.1.11. animation\_reverse

이 함수는 애니메이션을 역방향으로 재생합니다.

표 12.14. animation\_reverse의 매개변수

매개변수	유형	설명
animation	GtfTypeRecord	조작할 애니메이션
<return>	boolean	true인 경우 애니메이션이 아직 실행되고 있지 않은 것입니다.

#### 12.4.3.1.12. animation\_running

이 함수는 애니메이션이 현재 실행되고 있는지를 확인합니다.

표 12.15. animation\_running의 매개변수

매개변수	유형	설명
animation	GtfTypeRecord	조작할 애니메이션
<return>	boolean	true인 경우 애니메이션이 실행되고 있는 것입니다.

#### 12.4.3.1.13. animation\_set\_time

이 함수는 애니메이션의 현재 시간을 설정하며, 애니메이션을 건너뛰거나 재생하는 데 사용할 수 있습니다.

표 12.16. animation\_set\_time의 매개변수

매개변수	유형	설명
animation	GtfTypeRecord	조작할 애니메이션
time	정수	time
<return>	boolean	true인 경우 함수가 정상적으로 실행된 것입니다.

#### 12.4.3.1.14. asinf

이 함수는 x의 아크사인 주요 값을 계산합니다.

표 12.17. asinf의 매개변수

매개변수	유형	설명
x	float	아크사인을 반환할 숫자
<return>	float	반환 값

#### 12.4.3.1.15. atan2f

이 함수는 두 인수의 부호를 사용해 y/x의 아크탄젠트 주요 값을 계산하여 결과의 사분면을 결정합니다.

표 12.18. atan2f의 매개변수

매개변수	유형	설명
y	float	인수 y
x	float	인수 x
<return>	float	반환 값



#### 12.4.3.1.16. atan2i

이 함수는 두 인수의 부호를 사용해  $y/x$ 의 아크탄젠트 주요 값을 계산하여 결과의 사분면을 결정합니다.

표 12.19. atan2i의 매개변수

매개변수	유형	설명
y	정수	인수 y
x	정수	인수 x
<return>	float	반환 값

#### 12.4.3.1.17. atanf

이 함수는 x의 아크탄젠트 주요 값을 계산합니다.

표 12.20. atanf의 매개변수

매개변수	유형	설명
x	float	아크탄젠트를 반환할 숫자
<return>	float	반환 값

#### 12.4.3.1.18. bool2string

이 함수는 부울 변수를 true 또는 false 문자열로 변환합니다.

표 12.21. bool2string의 매개변수

매개변수	유형	설명
x	bool	문자열로 변환할 값
<return>	문자열	true x가 true일 경우, 그렇지 않을 경우 false

### 12.4.3.2. EB GUIDE 스크립트 함수 C - H

#### 12.4.3.2.1. ceil

이 함수는 인수보다 작지 않은 최소 정수 값을 반환합니다.

표 12.22. ceil의 매개변수

매개변수	유형	설명
value	float	반올림할 값
<return>	정수	반올림된 값

### 12.4.3.2.2. changeDynamicStateMachinePriority

이 함수는 동적 상태 머신의 우선순위를 변경합니다.

표 12.23. changeDynamicStateMachinePriority의 매개변수

매개변수	유형	설명
state		동적 상태 머신 목록이 포함된 상태
sm	정수	동적 상태 머신
priority	정수	목록에서 동적 상태 머신의 우선순위

### 12.4.3.2.3. character2unicode

이 함수는 문자열의 첫 번째 문자 유니코드 값을 반환합니다.

표 12.24. character2unicode의 매개변수

매개변수	유형	설명
str	string	입력 문자열
<return>	정수	유니코드 값으로서의 문자 오류의 경우 0

### 12.4.3.2.4. character2unicode

이 함수는 문자열의 첫 번째 문자 유니코드 값을 반환합니다.

표 12.25. character2unicode의 매개변수

매개변수	유형	설명
str	string	입력 문자열
<return>	정수	유니코드 값으로서의 문자 오류의 경우 0

#### 12.4.3.2.5. clampf

이 함수는 부동 소수점 값을 정의된 범위 [xmin, xmax]로 한정합니다. 즉, 함수는  $\max(xmin, \min(xmax, x))$ 를 계산합니다.

표 12.26. clampf의 매개변수

매개변수	유형	설명
x	float	고정할 값
xmin	float	최소 범위
xmax	float	최대 범위
<return>	float	x 값은 [xmin, xmax] 범위로 고정됩니다.

#### 12.4.3.2.6. clampi

이 함수는 정수 값을 정의된 범위 [xmin, xmax]로 고정합니다. 즉, 함수는  $\max(xmin, \min(xmax, x))$ 를 계산합니다.

표 12.27. clampi의 매개변수

매개변수	유형	설명
x	int	고정할 값
xmin	int	최소 범위
xmax	int	최대 범위
<return>	int	x 값은 [xmin, xmax] 범위로 고정됩니다.

#### 12.4.3.2.7. clearAllDynamicStateMachines

이 함수는 동적 상태 머신 목록에서 모든 동적 상태 머신을 제거합니다.

표 12.28. clearAllDynamicStateMachines의 매개변수

매개변수	유형	설명
state		동적 상태 머신 목록이 포함된 상태

#### 12.4.3.2.8. color2string

이 함수는 색을 8개 16진수 값으로 변환합니다.

표 12.29. color2string의 매개변수

매개변수	유형	설명
value	color	문자열로 변환할 색
<return>	string	#를 접두사로 사용하여 16진수 숫자 문자열로 서식이 지정된 색

**[참고]**



**서식 지정 예제**

반환되는 문자열의 형식은 #RRGGBBAA이며, 여기에는 각각 색 채널 빨강, 초록, 파랑, 알파에 해당하는 문자가 두 개씩 포함됩니다.

예를 들어 불투명 순수 빨강은 #ff0000ff로 변환되고 반투명 순수 초록은 #00ff007f로 변환됩니다.

**12.4.3.2.9. cosf**

이 함수는 x의 코사인을 반환합니다. 여기서 x는 라디안 단위로 지정됩니다.

표 12.30. cosf의 매개변수

매개변수	유형	설명
x	float	코사인을 반환할 숫자
<return>	float	반환 값

**12.4.3.2.10. deg2rad**

이 함수는 각도 단위를 도에서 라디안으로 변환합니다.

표 12.31. deg2rad의 매개변수

매개변수	유형	설명
x	float	도에서 라디안으로 변환할 각도
<return>	float	반환 값

**12.4.3.2.11. expf**

이 함수는 x의 거듭제곱으로 올림된 e의 값(자연 로그의 밑)을 반환합니다.

표 12.32. expf의 매개변수

매개변수	유형	설명
x	float	지수

매개변수	유형	설명
<return>	float	반환 값

#### 12.4.3.2.12. float2string

이 함수는 단순 부동을 문자열로 변환합니다.

표 12.33. float2string의 매개변수

매개변수	유형	설명
value	float	문자열로 변환할 값
<return>	string	문자열로 서식이 지정된 부동 값

#### 12.4.3.2.13. floor

이 함수는 매개변수 값보다 크지 않은 최대 정수 값을 반환합니다.

표 12.34. floor의 매개변수

매개변수	유형	설명
value	float	반올림할 값
<return>	정수	반올림된 값

#### 12.4.3.2.14. fmod

이 함수는 부동 소수점 나눗셈  $x/y$ 의 나머지를 계산합니다.

표 12.35. fmod의 매개변수

매개변수	유형	설명
x	float	부동 소수점 분자
y	float	부동 소수점 분모
<return>	float	나눗셈 $x/y$ 의 나머지

#### 12.4.3.2.15. focusMoveTo

이 함수는 포커스 관리자가 포커스 가능한 전용 요소로 포커스를 전달하도록 강제 지정합니다.



표 12.36. focusMoveTo의 매개변수

매개변수	유형	설명
widget	widget	위젯의 포커스가 이동됩니다.
<return>	void	

**12.4.3.2.16. focusNext**

이 함수는 포커스 관리자가 포커스 가능한 다음 요소로 포커스를 전달하도록 강제 지정합니다.

표 12.37. focusNext의 매개변수

매개변수	유형	설명
<return>	void	

**12.4.3.2.17. focusPrevious**

이 함수는 포커스 관리자가 포커스 가능한 이전 요소로 포커스를 되돌리도록 강제 지정합니다.

표 12.38. focusPrevious의 매개변수

매개변수	유형	설명
<return>	void	

**12.4.3.2.18. format\_float**

함수는 부동 값을 생성합니다.

표 12.39. format\_float의 매개변수

매개변수	유형	설명
format	string	다음과 같은 구조의 문자열입니다. %[플래그] [너비] [.정밀도] 유형 <ul style="list-style-type: none"> <li>▶ 플래그: 출력 양쪽 맞춤과 부호, 공백, 앞에 오는 0, 8진수/16진수 접두사 출력을 제어하는 하나 이상의 선택적 문자입니다.</li> <li>▶ 너비: 출력되는 문자의 최소 수를 지정하는 선택적 10진수입니다.</li> <li>▶ 정밀도: 소수점 문자 뒤의 숫자 자릿수 또는 유효 자릿수의 수를 지정하는 선택적 10진수입니다.</li> </ul>

매개변수	유형	설명
		▶ 유형: 연결된 인수가 해석되는 대상(문자, 문자열, 정수 또는 부동 숫자)을 결정하는 필수 변환 지정자 문자입니다.
useDotAsDelimiter	boolean	구분 기호 부호를 정의합니다.  사용 가능한 값은 다음과 같습니다.  ▶ true: 점을 구분 기호로 사용합니다. ▶ false: 쉼표를 구분 기호로 사용합니다.
value	float	서식을 지정할 숫자

**주의!**



**C++용 printf 사양 준수**

format 매개변수는 C++용 printf 사양에 따라 정의됩니다.

이 사양을 준수하지 않는 값을 사용하는 경우 예기치 않은 동작이 발생할 수 있습니다.

예를 들어 format\_float에 대해 사용 가능한 유형은 f, a, g 및 e이며 유형 문자를 두 개 이상 사용할 수는 없습니다.

**12.4.3.2.19. format\_int**

이 함수는 정수 값을 생성합니다.

표 12.40. format\_int의 매개변수

매개변수	유형	설명
format	string	다음과 같은 구조의 문자열입니다.  %[플래그] [너비] [.정밀도] 유형  ▶ 플래그: 출력 양쪽 맞춤과 부호, 공백, 앞에 오는 0, 8진수/16진수 접두사 출력을 제어하는 하나 이상의 선택적 문자입니다. ▶ 너비: 출력되는 문자의 최소 수를 지정하는 선택적 10진수입니다. ▶ 정밀도: 인쇄되는 숫자의 최소 수를 지정하는 선택적 10진수입니다. ▶ 유형: 연결된 인수가 해석되는 대상(문자, 문자열, 정수 또는 부동 숫자)을 결정하는 필수 변환 지정자 문자입니다.
value	int	서식을 지정할 숫자

**주의!**



**C++용 printf 사양 준수**

format 매개변수는 C++용 printf 사양에 따라 정의됩니다.

이 사양을 준수하지 않는 값을 사용하는 경우 예기치 않은 동작이 발생할 수 있습니다.

예를 들어 format\_int에 대해 사용 가능한 유형은 d, i,o, x 및 u이며 유형 문자를 두 개 이상 사용할 수는 없습니다.

**12.4.3.2.20. frac**

이 함수는 부동 소수점 값의 소수부를 계산합니다. 반환 값은 [0, 1] 간격에 있습니다. 예를 들어 함수가 매개변수 값 x=1.5 또는 x=-1.5에 대해 0.5를 반환합니다.

표 12.41. frac의 매개변수

매개변수	유형	설명
x	float	부동 소수점 값
<return>	float	부동 소수점 값의 소수부.

**12.4.3.2.21. getConfigItem**

이 함수는 데이터폴 항목을 구성 항목 값으로 채웁니다.

표 12.42. getConfigItem의 매개변수

매개변수	유형	설명
itemId	dp_id	데이터폴 ID는 구성 항목이 저장되는 곳입니다.
name	string	구성 항목 이름
<return>	부울	데이터폴 항목이 구성 항목 값으로 성공적으로 채워진 경우 true입니다.

**12.4.3.2.22. getFontAscender**

이 함수는 매개변수로 전달된 글꼴의 오름차순을 반환합니다.

표 12.43. getFontAscender의 매개변수

매개변수	유형	설명
x	font	평가될 글꼴
<return>	정수	글꼴의 오름차순



#### 12.4.3.2.23. getFontDescender

이 함수는 매개변수로 전달된 글꼴의 내림차순을 반환합니다.

표 12.44. getFontDescender의 매개변수

매개변수	유형	설명
x	font	평가될 글꼴
<return>	정수	글꼴의 내림차순

#### 12.4.3.2.24. getFontLineGap

이 함수는 매개변수로 전달된 글꼴의 줄 간격을 반환합니다.

표 12.45. getFontLineGap의 매개변수

매개변수	유형	설명
x	font	평가될 글꼴
<return>	정수	글꼴의 줄 간격

#### 12.4.3.2.25. getImageHeight

이 함수는 매개변수로 전달된 이미지의 높이(픽셀)를 반환합니다.

표 12.46. getImageHeight의 매개변수

매개변수	유형	설명
x	이미지 위젯	평가할 위젯
<return>	정수	이미지 높이(픽셀)

#### 12.4.3.2.26. getImageWidth

이 함수는 매개변수로 전달된 이미지의 너비(픽셀)를 반환합니다.

표 12.47. getImageWidth의 매개변수

매개변수	유형	설명
x	이미지 위젯	평가될 위젯
<return>	정수	이미지 너비(픽셀)



**12.4.3.2.27. getLabelTextHeight**

이 함수는 레이블 텍스트의 총 높이(픽셀)를 반환합니다. 총 높이는 다음 식을 이용해 계산됩니다.

$$\text{total\_height} = \text{line\_height} * \text{line\_count} + \text{line\_gap} * (\text{line\_count} - 1)$$

line\_gap은 글꼴 줄 간격과 여러 줄 위젯 피처의 lineGap 속성의 합으로 계산됩니다. 글꼴 줄 간격과 lineGap 속성은 모두 음수일 수 있습니다.

표 12.48. getLabelTextHeight의 매개변수

매개변수	유형	설명
widget	레이블 위젯	평가될 위젯
<return>	정수	텍스트 높이(픽셀)

**12.4.3.2.28. getLabelTextWidth**

이 함수는 레이블 텍스트 중 가장 긴 줄의 너비를 반환합니다.

표 12.49. getLabelTextWidth의 매개변수

매개변수	유형	설명
widget	레이블 위젯	평가할 위젯
<return>	정수	텍스트 중 가장 긴 줄의 너비(픽셀)

**12.4.3.2.29. getLineCount**

이 함수는 레이블 텍스트의 줄 수를 반환합니다.

표 12.50. getLineCount의 매개변수

매개변수	유형	설명
widget	레이블 위젯	평가될 위젯
<return>	정수	텍스트의 줄 수

**12.4.3.2.30. getLineHeight**

이 함수는 매개변수로 전달된 글꼴로 작성된 줄의 높이를 반환합니다.

표 12.51. getLineHeight의 매개변수

매개변수	유형	설명
x	font	평가될 글꼴

매개변수	유형	설명
<return>	정수	지정된 글꼴로 작성된 줄의 높이

#### 12.4.3.2.31. getProductString

이 함수는 EB GUIDE GTF의 제품 이름이 포함된 문자열을 반환합니다.

표 12.52. getProductString의 매개변수

매개변수	유형	설명
<return>	string	제품 이름

#### 12.4.3.2.32. getTextHeight

이 함수는 글꼴 리소스를 기준으로 하는 텍스트 높이를 반환합니다. 높이는 글꼴 오름차순 및 내림차순의 합계를 나타냅니다.

표 12.53. getTextHeight의 매개변수

매개변수	유형	설명
text	string	평가할 텍스트
font	font	평가할 글꼴
<return>	정수	텍스트의 높이 글꼴 크기가 0 또는 음수인 경우 함수에서 0을 반환합니다.

**[참고]** 이 함수는 항상 한 줄로 된 텍스트를 가정한 높이 값을 계산합니다.



#### 12.4.3.2.33. getTextLength

이 함수는 텍스트의 문자 수를 반환합니다.

표 12.54. getTextLength의 매개변수

매개변수	유형	설명
text	string	평가할 텍스트

매개변수	유형	설명
<return>	정수	텍스트의 문자 수

**[참고]**



**이스케이프 시퀀스**

EB GUIDE 스크립트는 \n 등의 이스케이프 시퀀스를 해결하지 않고 모든 문자를 셉니다. 예를 들어, Label\n 텍스트의 경우 getTextLength 함수는 7을 반환합니다.

**12.4.3.2.34. getWidth**

이 함수는 글꼴 리소스를 기준으로 하는 텍스트 너비를 반환합니다.

표 12.55. getWidth의 매개변수

매개변수	유형	설명
text	string	평가할 텍스트
font	font	평가할 글꼴
<return>	정수	텍스트의 너비 글꼴 크기가 0 또는 음수인 경우 함수에서 0을 반환합니다.

**[참고]**



이 함수는 항상 한 줄로 된 텍스트를 가정한 너비 값을 계산합니다.

**12.4.3.2.35. getVersionString**

이 함수는 EB GUIDE GTF의 버전 번호가 포함된 문자열을 반환합니다.

표 12.56. getVersionString의 매개변수

매개변수	유형	설명
<return>	string	버전 문자열

**12.4.3.2.36. has\_list\_window**

이 함수는 데이터폴 항목의 유형 목록 대해 인덱스가 유효한지를 확인합니다. 창 모드 목록의 경우에는 인덱스가 하나 이상의 창 내에 있는지도 확인합니다.

표 12.57. has\_list\_window의 매개변수

매개변수	유형	설명
itemId	dp_id	데이터폴 항목의 유형 목록에 대한 ID
index	정수	데이터폴 항목 내의 인덱스
<return>	부울	true인 경우 데이터폴 항목 내의 인덱스가 올바르며 하나 이상의 창 내에 있는 것입니다.

#### 12.4.3.2.37. hsba2color

이 함수는 HSB/HSV 색을 GTF 색으로 변환합니다.

표 12.58. hsba2color의 매개변수

매개변수	유형	설명
hue	정수	0~360 사이의 색 값(도)
saturation	정수	채도(%)
brightness	정수	밝기(%)
alpha	정수	0(완전 투명)~255(불투명) 사이의 알파 값
<return>	color	알파 값이 적용된 결과 GTF 색

#### 12.4.3.3. EB GUIDE 스크립트 함수 I - R

##### 12.4.3.3.1. int2float

이 함수는 부동 소수점 값으로 변환된 정수 값을 반환합니다.

표 12.59. int2float의 매개변수

매개변수	유형	설명
value	정수	부동으로 변환할 값
<return>	float	부동으로 변환된 정수 값

##### 12.4.3.3.2. int2string

이 함수는 단순 정수를 문자열로 변환합니다.



표 12.60. int2string의 매개변수

매개변수	유형	설명
value	정수	문자열로 변환할 값
<return>	string	문자열로 변환된 정수 값(10진수 표기법 사용)

#### 12.4.3.3.3. isDynamicStateMachineActive

이 함수는 동적 상태 머신 목록이 포함된 상태가 활성화인지 확인합니다.

표 12.61. isDynamicStateMachineActive의 매개변수

매개변수	유형	설명
state		동적 상태 머신 목록이 포함된 상태
sm	정수	동적 상태 머신

#### 12.4.3.3.4. isWidgetOnActiveStatemachine

이 함수는 위젯이 활성 상태의 상태 머신에 속해 있는지 확인합니다.

표 12.62. isWidgetOnActiveStatemachine의 매개변수

매개변수	유형	설명
widget	widget	평가될 위젯
<return>	부울	위젯이 활성 상태의 상태 머신에 속해 있다면 true

#### 12.4.3.3.5. language

이 함수는 모든 데이터폴 항목의 언어를 전환합니다. 이 작업은 비동기 방식으로 수행됩니다.

표 12.63. language의 매개변수

매개변수	유형	설명
language	languageType	전환 대상 언어 예: f:language(!:German)
<return>	void	

#### 12.4.3.3.6. lerp

이 함수는  $(1-s) * x + s * y$  식을 이용해 x 및 y 두 값의 선형 보간을 계산합니다.

표 12.64. lerp의 매개변수

매개변수	유형	설명
x	float	첫 번째 값
y	float	두 번째 값
s	float	x 및 y의 값 사이를 선형 보간하는 값
<return>	float	선형 보간 $(1-s) * x + s * y$ 를 반환합니다.

#### 12.4.3.3.7. localtime\_day

이 함수는 시스템 시간 값에서 로컬 시간의 날짜([1:31])를 추출합니다.

표 12.65. localtime\_day의 매개변수

매개변수	유형	설명
time	정수	시스템 시간에서 반환하는 타임스탬프
<return>	정수	추출된 요일

#### 12.4.3.3.8. localtime\_hour

이 함수는 시스템 시간 값의 로컬 시간에서 시를 추출합니다.

표 12.66. localtime\_hour의 매개변수

매개변수	유형	설명
time	정수	시스템 시간에서 반환하는 타임스탬프
<return>	정수	추출된 시

#### 12.4.3.3.9. localtime\_minute

이 함수는 시스템 시간 값의 로컬 시간에서 분을 추출합니다.

표 12.67. localtime\_minute의 매개변수

매개변수	유형	설명
time	정수	시스템 시간에서 반환하는 타임스탬프
<return>	정수	추출된 분

#### 12.4.3.3.10. localtime\_month

이 함수는 시스템 시간 값의 로컬 시간에서 월([0:11])을 추출합니다.

표 12.68. localtime\_month의 매개변수

매개변수	유형	설명
time	정수	시스템 시간에서 반환하는 타임스탬프
<return>	정수	추출된 월

#### 12.4.3.3.11. localtime\_second

이 함수는 시스템 시간 값의 로컬 시간에서 초를 추출합니다.

표 12.69. localtime\_second의 매개변수

매개변수	유형	설명
time	정수	시스템 시간에서 반환하는 타임스탬프
<return>	정수	추출된 초

#### 12.4.3.3.12. localtime\_weekday

이 함수는 시스템 시간 값의 로컬 시간에서 요일([0:6])을 추출합니다. 0은 일요일입니다.

표 12.70. localtime\_weekday의 매개변수

매개변수	유형	설명
time	정수	시스템 시간에서 반환하는 타임스탬프
<return>	정수	추출된 요일

#### 12.4.3.3.13. localtime\_year

이 함수는 시스템 시간 값의 로컬 시간에서 연도를 추출합니다.

표 12.71. localtime\_year의 매개변수

매개변수	유형	설명
time	정수	시스템 시간에서 반환하는 타임스탬프
<return>	정수	추출된 연도



#### 12.4.3.3.14. log10f

이 함수는  $x$ 의 밑이 10인 로그를 반환합니다.

표 12.72. log10f의 매개변수

매개변수	유형	설명
$x$	float	인수
<return>	float	반환 값

#### 12.4.3.3.15. logf

이 함수는  $x$ 의 자연 로그를 반환합니다.

표 12.73. logf의 매개변수

매개변수	유형	설명
$x$	float	인수
<return>	float	반환 값

#### 12.4.3.3.16. maxf

이 함수는 두 부동 소수점 값의 최댓값을 계산합니다.

표 12.74. maxf의 매개변수

매개변수	유형	설명
$x$	float	첫 번째 값
$y$	float	두 번째 값
<return>	float	$x$ 및 $y$ 의 최댓값

#### 12.4.3.3.17. maxi

이 함수는 두 정수 값의 최댓값을 계산합니다.

표 12.75. maxi의 매개변수

매개변수	유형	설명
$x$	int	첫 번째 값
$y$	int	두 번째 값

매개변수	유형	설명
<return>	int	x 및 y의 최댓값

#### 12.4.3.3.18. minf

이 함수는 두 부동 소수점 값의 최솟값을 계산합니다.

표 12.76. minf의 매개변수

매개변수	유형	설명
x	float	첫 번째 값
y	float	두 번째 값
<return>	float	x 및 y의 최솟값

#### 12.4.3.3.19. mini

이 함수는 두 정수 값의 최솟값을 계산합니다.

표 12.77. mini의 매개변수

매개변수	유형	설명
x	int	첫 번째 값
y	int	두 번째 값
<return>	int	x 및 y의 최솟값

#### 12.4.3.3.20. nearbyint

이 함수는 가장 인접한 정수로 반올림합니다.

표 12.78. nearbyint의 매개변수

매개변수	유형	설명
value	float	반올림할 값
<return>	정수	반올림된 값

#### 12.4.3.3.21. popDynamicStateMachine

이 함수는 우선순위 대기열 맨 위의 동적 상태 머신을 제거합니다.

표 12.79. popDynamicStateMachine의 매개변수

매개변수	유형	설명
state		동적 상태 기계 목록이 포함된 상태
sm	정수	동적 상태 기계

#### 12.4.3.3.22. powf

이 함수는  $y$ 의 거듭제곱으로 올림된  $x$ 의 값을 반환합니다.

표 12.80. powf의 매개변수

매개변수	유형	설명
$x$	float	인수 $x$
$y$	float	인수 $y$
<return>	float	반환 값

#### 12.4.3.3.23. pushDynamicStateMachine

이 함수는 우선순위 대기열에 동적 상태 머신을 삽입합니다.

표 12.81. pushDynamicStateMachine의 매개변수

매개변수	유형	설명
state		동적 상태 머신 목록이 포함된 상태
sm	정수	동적 상태 머신
priority	정수	목록에서 동적 상태 머신의 우선순위

#### 12.4.3.3.24. rad2deg

이 함수는 각도 단위를 라디안에서 도로 변환합니다.

표 12.82. rad2deg의 매개변수

매개변수	유형	설명
$x$	float	인수
<return>	float	반환 값



**12.4.3.3.25. rand**

이 함수는 0에서  $2^{31}-1$  사이의 임의 값을 가져옵니다.

표 12.83. rand의 매개변수

매개변수	유형	설명
<return>	정수	0에서 $2^{31}-1$ 사이의 임의 수

**12.4.3.3.26. rgba2color**

이 함수는 RGB 색 공간에서 GTF 색으로 변환합니다.

표 12.84. rgba2color의 매개변수

매개변수	유형	설명
red	정수	빨간색 좌표(0~255)
green	정수	녹색 좌표(0~255)
blue	정수	파란색 좌표(0~255)
alpha	정수	0(완전 투명)~255(불투명) 사이의 알파 값
<return>	color	RGB 색 공간에서 GTF 색으로 변환된 색(알파 값 적용)

**12.4.3.3.27. round**

이 함수는 가장 인접한 정수로 반올림하되 두 정수의 정확히 중간에 해당하는 숫자는 0에서 더 먼 숫자로 반올림합니다.

표 12.85. round의 매개변수

매개변수	유형	설명
value	float	반올림할 값
<return>	정수	반올림된 값

**12.4.3.4. EB GUIDE 스크립트 함수 S - W**

**12.4.3.4.1. saturate**

이 함수는 부동 소수점 값을 [0, 1] 범위로 고정합니다. 즉, 이 함수는  $\max(0, \min(1, x))$ 를 계산하고 clampf(0, 1, x)의 속기 표기로 작동합니다.

표 12.86. saturate의 매개변수

매개변수	유형	설명
x	float	고정할 값
<return>	float	x 값은 [0, 1] 범위로 고정됩니다.

#### 12.4.3.4.2. seed\_rand

이 함수는 난수 생성기의 초기값을 설정합니다.

표 12.87. seed\_rand의 매개변수

매개변수	유형	설명
seed	정수	난수 생성기의 초기값으로 사용할 값
<return>	void	

#### 12.4.3.4.3. shutdown

이 함수는 프레임워크에 프로그램 종료를 요청합니다.

#### 12.4.3.4.4. sinf

이 함수는 x의 사인을 반환합니다. 여기서 x는 라디안 단위로 지정됩니다.

표 12.88. sinf의 매개변수

매개변수	유형	설명
x	float	인수
<return>	float	반환 값

#### 12.4.3.4.5. skin

이 함수는 모든 데이터폴 항목의 스킨을 전환합니다. 이 작업은 비동기 방식으로 수행됩니다.

표 12.89. skin의 매개변수

매개변수	유형	설명
skin	skinType	전환 대상 스킨(예: f:skin(s:Standard))



매개변수	유형	설명
<return>	void	

**12.4.3.4.6. smoothstep**

이 함수가 [xmin, xmax] 범위에 있을 경우 매끄러운 허마이트 보간법  $3z^2 - 2z^3$  with  $z = (x - xmin) / (xmax - xmin)$ 을, 그렇지 않을 경우 0을 계산합니다. 이 함수는 [0,1] 간격의 값을 반환합니다.

표 12.90. smoothstep의 매개변수

매개변수	유형	설명
xmin	float	xmin 값
xmax	float	xmax 값
x	float	보간될 값
<return>	float	허마이트 보간 $3z^2 - 2z^3$ with $z = (x-xmin) / (xmax-xmin)$ 을 반환합니다.

**12.4.3.4.7. sqrtf**

이 함수는 x의 음수가 아닌 제곱근을 반환합니다.

표 12.91. sqrtf의 매개변수

매개변수	유형	설명
x	float	인수
<return>	float	반환 값

**12.4.3.4.8. string2float**

이 함수는 문자열의 첫 요소를 부동으로 변환합니다.

문자열의 첫 요소에 필요한 형식은 다음과 같습니다.

1. 선행 공백(선택 사항)
2. 더하기('+') 또는 빼기('-') 기호(선택 사항)
3. 다음 항목 중 하나
  - ▶ 10진수
  - ▶ 16진수



- ▶ 무한
- ▶ NAN(숫자가 아님)

표 12.92. string2float의 매개변수

매개변수	유형	설명
str	string	문자열 값
<return>	float	반환 값

#### 12.4.3.4.9. string2int

이 함수는 문자열의 첫 요소를 정수로 변환합니다. 입력이 범위를 초과하는 경우 결과는 2147483647~-2147483648 범위로 잘립니다. 문자열이 숫자로 시작하지 않으면 0이 반환됩니다.

표 12.93. string2int의 매개변수

매개변수	유형	설명
str	string	문자열 값
<return>	정수	반환 값

#### 12.4.3.4.10. string2string

이 함수는 문자열의 서식을 지정합니다.

표 12.94. string2string의 매개변수

매개변수	유형	설명
str	string	서식을 지정할 문자열
len	정수	문자열의 최대 길이
<return>	string	언어 문자열

#### 12.4.3.4.11. substring

이 함수는 문자열의 하위 문자열 복사본을 만듭니다. 음수 끝 인덱스가 지원됩니다.

예:

- ▶ substring("abc", 0, -1) abc를 반환합니다.
- ▶ substring("abc", 0, -2) ab를 반환합니다.
- ▶ substring ("abcd", 1, 3) bc를 반환합니다.

표 12.95. substring의 매개변수

매개변수	유형	설명
str	string	입력 문자열
startIndex	정수	결과 문자열의 첫 번째 문자 인덱스
endIndex	정수	결과의 요소가 아닌 첫 번째 문자 인덱스
<return>	string	언어 문자열

#### 12.4.3.4.12. system\_time

이 함수는 현재 시스템 시간(초)을 가져옵니다. 결과는 localtime\_\* 함수로 전달됩니다.

표 12.96. system\_time의 매개변수

매개변수	유형	설명
<return>	정수	시스템 시간(초)

#### 12.4.3.4.13. system\_time\_ms

이 함수는 현재 시스템 시간(밀리초)을 가져옵니다.

표 12.97. system\_time\_ms의 매개변수

매개변수	유형	설명
<return>	정수	시스템 시간(밀리초)

#### 12.4.3.4.14. tanf

이 함수는 x의 탄젠트를 반환합니다. 여기서 x는 라디안 단위로 지정됩니다.

표 12.98. tanf의 매개변수

매개변수	유형	설명
x	float	인수
<return>	float	반환 값

#### 12.4.3.4.15. trace\_dp

이 함수는 추적 로그와 연결 로그에 데이터폴 항목에 대한 디버깅 정보를 씁니다.



표 12.99. trace\_dp의 매개변수

매개변수	유형	설명
itemId	dp_id	디버그 정보를 추적할 항목의 데이터폴 ID
<return>	void	

#### 12.4.3.4.16. trace\_string

이 함수는 추적 로그와 연결 로그에 문자열을 씁니다.

표 12.100. trace\_string의 매개변수

매개변수	유형	설명
str	string	추적할 텍스트
<return>	void	

#### 12.4.3.4.17. transformToScreenX

이 함수는 위젯 및 로컬 좌표를 가져온 다음 화면 기준 월드 좌표계에서 x 위치를 반환합니다.

표 12.101. transformToScreenX의 매개변수

매개변수	유형	설명
widget	widget	좌표의 기준이 되는 위젯
localX	정수	로컬 좌표의 x 위치
localY	정수	로컬 좌표의 y 위치
<return>	정수	화면 좌표의 x 위치

#### 12.4.3.4.18. transformToScreenY

이 함수는 위젯 및 로컬 좌표를 가져온 다음 화면 기준 월드 좌표계 내 위치의 Y 위치를 반환합니다.

표 12.102. transformToScreenY의 매개변수

매개변수	유형	설명
widget	widget	좌표의 기준이 되는 위젯
localX	정수	로컬 좌표의 x 위치
localY	정수	로컬 좌표의 y 위치



매개변수	유형	설명
<return>	정수	화면 좌표의 y 위치

**12.4.3.4.19. transformToWidgetX**

이 함수는 터치 반응에 대해 제공된 화면 좌표 및 위젯을 가져온 다음 위젯 기준 로컬 좌표계에서 x 위치를 반환합니다.

표 12.103. transformToWidgetX의 매개변수

매개변수	유형	설명
widget	widget	좌표의 기준이 되는 위젯
screenX	정수	화면 좌표의 x 위치
screenY	정수	화면 좌표의 y 위치
<return>	정수	로컬 좌표의 x 위치

**12.4.3.4.20. transformToWidgetY**

이 함수는 터치 반응에 대해 제공된 화면 좌표 및 위젯을 가져온 다음 위젯 기준 로컬 좌표계에서 y 위치를 반환합니다.

표 12.104. transformToWidgetY의 매개변수

매개변수	유형	설명
widget	widget	좌표의 기준이 되는 위젯
screenX	정수	화면 좌표의 x 위치
screenY	정수	화면 좌표의 y 위치
<return>	정수	로컬 좌표의 y 위치

**12.4.3.4.21. trunc**

이 함수는 항상 0에 가장 인접한 정수 값으로 반올림합니다.

표 12.105. trunc의 매개변수

매개변수	유형	설명
value	float	반올림할 값
<return>	정수	반올림된 값

### 12.4.3.4.22. widgetGetChildCount

이 함수는 지정된 위젯의 하위 위젯 수를 가져옵니다.

표 12.106. widgetGetChildCount의 매개변수

매개변수	유형	설명
widget	widget	하위 위젯의 수를 가져올 위젯
<return>	정수	하위 위젯의 수

## 12.5. 이벤트

표 12.107. 이벤트의 속성

속성 이름	설명
Name	이벤트의 이름입니다.
Event ID	EB GUIDE TF에서 이벤트를 보내고 받는 데 사용하는 숫자 값입니다.
Event group	이벤트 그룹의 이름입니다.  이벤트 그룹에는 EB GUIDE TF에서 이벤트를 보내고 받는 데 사용하는 ID가 있습니다.

### 12.5.1. 핵심 이벤트를 위한 십진 코드

표 12.108. 숫자 패드 키의 십진 코드

숫자 패드 키	십진 코드
0	5
1	6
2	7
3	8
4	9
5	10
6	11
7	12



숫자 패드 키	십진 코드
8	13
9	14

표 12.109. 기능 키의 십진 코드

기능 키	십진 코드
F1	19
F2	20
F3	21
F4	22
F5	23
F6	24
F7	25
F8	26
F9	27
F10	28
F11	29
F12	30

표 12.110. ASCII 키의 십진 코드

ASCII 키	십진 코드
공간	32
a	97
b	98
c	99
d	100
e	101
f	102
g	103
h	104
i	105
j	106
k	107

ASCII 키	십진 코드
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

## 12.6. model.json 구성 파일

model.json은 EB GUIDE TF 구성 파일이며 단일 EB GUIDE 모델과 관련된 구성 항목을 포함합니다.

model.json 파일은 내보낸 EB GUIDE 모델의 일부입니다.

다음 테이블은 모든 기본 구성 매개변수에 대해 문서로 사용됩니다.

**[참고]**



**JSON 객체 표기법**

EB GUIDE Studio에서 model.json을 구성할 경우 JSON 객체 표기법을 사용하십시오.

예는 [섹션 12.6.1, “EB GUIDE Studio의 예제 model.json”](#)를 참조하십시오.

JSON 형식에 대한 자세한 내용은 <http://www.json.org>를 참조하십시오.

표 12.111. 일반

구성 항목	유형	설명	기본값
gtf.eventsystem.maxQueue	정수	이벤트 대기열의 최대 값	0



구성 항목	유형	설명	기본값
gtf.model.traces	부울	f:trace_string 스크립트 기능 추적 가능	true
gtf.model.identifier	string	EB GUIDE 모델의 고유한 식별자(EB GUIDE Studio 프로젝 트의 UUID와 동일)	비어 있음
gtf.model.identifier.short	정수	EB GUIDE 모델의 축 약 식별자	0xdeadbeaf

표 12.112. 파일 및 경로

구성 항목	유형	설명	기본값
gtf.model.path	string	EB GUIDE 모델로의 경로	None
gtf.model.config	string	EB GUIDE 모델 구성 으로의 전체 경로	<gtf.model.path>/ model.json
gtf.datapool.descriptionFile	string	데이터풀 설명 파일의 이름	datapool.gtf
gtf.model.files.sm	string	상태 기계 설명 파일의 이름	model.bin
gtf.model.files.rm	string	리소스 설명 파일의 이 름	resources.bin
gtf.model.files.views	string	보기 설명 파일의 이름	views.bin
gtf.model.files.types	string	유형 설명 파일의 이름	types.bin
gtf.model.pluginstoload	문자열 목 록	로드할 EB GUIDE 모 델 플러그인의 이름	비어 있는 문자열 목록
gtf.eventsystem.mapFile	string	이벤트 시스템 매핑 파 일의 이름	eventMap.gtf

gtf.model.coreNames 옵션은 구성된 모든 코어의 이름을 포함하는 문자열 목록입니다. 다음 테이블에는 모든 코어의 구성 항목이 포함되어 있습니다.

표 12.113. 코어

구성 항목	유형	설명	기본값
gtf.model.cores.<corename>.ownThread	부울	코어에서 실행할 자체 스레드 사용 여부 지정	false
gtf.model.cores.<corename>.id	정수	코어 컨텍스트 식별자 지정	0

gtf.model.sceneNames 옵션은 구성된 모든 장면을 포함하는 문자열 목록입니다. 모든 장면에 다음 테이블의 구성 항목이 있습니다.

표 12.114. 장면

구성 항목	유형	설명	기본값
gtf.model.scenes.<scenename>.visible	부울	장면의 가시성 결정	true
gtf.model.scenes.<scenename>.width	정수	장면의 너비	800
gtf.model.scenes.<scenename>.height	정수	장면의 높이	480
gtf.model.scenes.<scenename>.x	정수	장면 시작 포인트의 좌표	0
gtf.model.scenes.<scenename>.y	정수	장면 시작 포인트의 좌표	0
gtf.model.scenes.<scenename>.projectName	string	작업 중인 프로젝트의 이름	
gtf.model.scenes.<scenename>.-windowCaption	string	표시된 창 이름 텍스트	
gtf.model.scenes.<scenename>.sceneId	정수	장면의 식별자	0
gtf.model.scenes.<scenename>.maxFPS	정수	FPS(초당 프레임) 단위의 다시 그리기 속도입니다. 다시 그리기 속도를 무제한으로 지정하려면 0으로 설정합니다.	60
gtf.model.scenes.<scenename>.hwLayerId	정수	코어 컨텍스트 식별자 지정	0
gtf.model.scenes.<scenename>.colorMode	정수	색상 모드 지정: ▶ 1: 32비트 (RGBA8888) ▶ 2: 16비트 (RGB565) ▶ 3: 24비트 (RGB888) ▶ 4: 32비트 sRGB ▶ 5: 32비트 sRGB(에 물레이트됨)	1
gtf.model.scenes.<scenename>.multisampling	정수	장면의 멀티샘플링 지정	0



구성 항목	유형	설명	기본값
		<ul style="list-style-type: none"> <li>▶ 0: 멀티샘플링 없음</li> <li>▶ 1: 2배 멀티샘플링</li> <li>▶ 2: 4배 멀티샘플링</li> </ul>	
gtf.model.scenes.<scenename>.-enableRemoteFramebuffer	부울	true인 경우 시뮬레이션 창으로의 화면 외부 버퍼 전송이 활성화됩니다.	false
gtf.model.scenes.<scenename>.-showWindowFrame	부울	렌더러 창 프레임을 표시할지 결정	true
gtf.model.scenes.<scenename>.showWindow	부울	true인 경우 Windows 기반 시스템에서 시뮬레이션용 추가 창이 열립니다.	true
gtf.model.scenes.<scenename>.disableVsync	부울	true인 경우 렌더러의 수직 동기화가 비활성화됩니다.	false
gtf.model.scenes.<scenename>.showFPS	정수	사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ 0: FPS 표시 안 함</li> <li>▶ 1: 화면에 FPS 표시</li> <li>▶ 2: 콘솔에 FPS 표시</li> <li>▶ 3: 화면과 콘솔에 FPS 표시</li> </ul>	0
gtf.model.scenes.<scenename>.renderer	string	사용할 렌더러의 이름: DirectXRenderer OpenGLRenderer 또는 OpenGL3Renderer	

표 12.115. 렌더링 일반

구성 항목	유형	설명	기본값
gtf.model.fontCache.width	정수	글꼴 캐시 atlas 질감의 너비	512





구성 항목	유형	설명	기본값
gtf.model.fontCache.height	정수	글꼴 캐시 atlas 질감의 높이	512
gtf.model.fontCache.age	정수	글꼴 캐시의 새로 고침 작업이 완료되어야 하는 최대 허용 기간	100
gtf.model.traversalStackSize	정수	렌더러 순회 스택 크기 (바이트)	32768

다음 테이블의 구성 항목은 서로 속해있습니다. 즉, 렌더러는 동일한 수의 항목이 세 목록 모두에 있음을 예상합니다. 한 목록에 인덱스가 있는 항목은 다른 목록에 같은 인덱스가 있는 항목에 속합니다.

표 12.116. 렌더러 표시 확장

구성 항목	유형	설명	기본값
gtf.model.displayId	정수 목록	장면의 식별자	
gtf.model.maxCacheSize	정수 목록	장면에 대한 최대 질감 캐시	
gtf.model.driverName	문자열 목록	OS 장면의 특정 드라이버 이름(예: /dev/fb0)	

다음 테이블의 구성 항목은 TextEngine 구성 요소를 구성하는 데 사용됩니다. TextEngine은 FreeType 서드파티 라이브러리를 기반으로 합니다. 다음 매개변수는 FreeType 구현으로 전달됩니다. FreeType에 대한 자세한 내용은 [https://www.freetype.org/freetype2/docs/reference/ft2-cache\\_subsystem.html](https://www.freetype.org/freetype2/docs/reference/ft2-cache_subsystem.html)을 참조하십시오.

글꼴 크기를 처리하는 EB GUIDE TF 방식으로 인해 ft\_size 객체가 ft\_face 객체에서 별도로 캐시되지 않습니다. max\_sizes의 값은 대상 플랫폼의 하드웨어에 의해 제한될 수 있다는 점을 고려하십시오.

표 12.117. TextEngine 구성 항목

구성 항목	유형	설명	기본값
gtf.model.textengine.replacementGlyph	정수	현재 글꼴에서 전용 글꼴의 문자를 찾을 수 없는 경우 사용되어야 하는 유니코드 문자입니다.	0xffffd
gtf.model.textengine.maxFaces	정수	캐싱된 서체의 최대 개수	0
gtf.model.textengine.maxSizes	정수	캐싱된 글꼴 크기의 최대 개수	0
gtf.model.textengine.maxBytes	정수	캐시에 사용될 수 있는 메모리의 최대량(바이트)	0

구성 항목	유형	설명	기본값
gtf.model.textengine.enablePlainFileStream	부울	글꼴 액세스 구성을 결정합니다. true인 경우 일반 파일 I/O 액세스가 사용됩니다. false인 경우 ROM 매핑된 파일 액세스가 사용됩니다.	false

**[참고]**

**비트맵 글꼴의 구성 항목**



.fnt 비트맵 글꼴의 경우 replacementGlyph 구성 항목만 사용할 수 있습니다. 비트맵 글꼴의 [테이블 12.117, "TextEngine 구성 항목"](#)에서는 다른 구성 항목을 사용할 수 없습니다.

**[참고]**

**ROM 매핑된 파일 방식과 일반 파일 I/O 방식 비교**



ROM 매핑된 파일 방식은 일반적으로 더 높은 성능을 제공합니다. 그러나 QNX와 같은 일부 시스템에서는 일반 파일 I/O 방식보다 더 많은 메모리를 사용합니다. 일반 파일 I/O 방식은 일반적으로 ROM 매핑된 파일 방식보다 더 적은 메모리를 사용합니다. 그러나 성능은 저하될 수 있습니다.

옵션 gtf.model.touchDevicesNames는 구성된 모든 터치 기기의 이름을 포함하는 문자열 목록입니다. 모든 터치 기기에 대해 다음 테이블에 나열된 구성 항목을 사용할 수 있습니다.

표 12.118. 터치 기기

구성 항목	유형	설명	기본값
gtf.model.touchDevices.<deviceName>.-touchscreenType	정수	터치 기기 유형 정의: <ul style="list-style-type: none"> <li>▶ 0: Galaxy</li> <li>▶ 1: imx WVGA</li> <li>▶ 2: Mouse</li> <li>▶ 3: General</li> <li>▶ 4: Lilliput_889GL</li> <li>▶ 5: GeneralMulti-touch</li> <li>▶ 6: Lilliput with automatic calibration</li> <li>▶ 7: GenericTouch-Configuration</li> </ul>	3



구성 항목	유형	설명	기본값
gtf.model.touchDevices.<deviceName>.-displayManagerId	정수	기기가 유효한 장면 ID 지정	0
gtf.model.touchDevices.<deviceName>.-touchId	정수	기기 ID 지정	0
gtf.model.touchDevices.<deviceName>.-minimalDistanceToMove	정수	터치 위치 변경에 반응하는 임계값	0
gtf.model.touchDevices.<deviceName>.-touchMoveRepeatTimeout	정수	터치 위치 변경 알림 간 지연	0
gtf.model.touchDevices.<deviceName>.width	정수	터치할 수 있는 기기 영역의 너비	0
gtf.model.touchDevices.<deviceName>.height	정수	터치할 수 있는 기기 영역의 높이	0
gtf.model.touchDevices.<deviceName>.x_high	정수	터치할 수 있는 기기 영역의 최대 수평 해상도 확장	0
gtf.model.touchDevices.<deviceName>.y_high	정수	터치할 수 있는 기기 영역의 최대 수직 해상도 확장	0
gtf.model.touchDevices.<deviceName>.x_low	정수	터치할 수 있는 기기 영역의 최소 수평 해상도 확장	0
gtf.model.touchDevices.<deviceName>.y_low	정수	터치할 수 있는 기기 영역의 최소 수직 해상도 확장	0
gtf.model.touchDevices.<deviceName>.-devicePath	string	터치에 사용된 드라이버 이름(예: /dev/input0)	

다음 테이블의 구성 항목은 렌더러에서 바이너리 셰이더 캐시를 구성하는 데 사용됩니다.

표 12.119. 셰이더

구성 항목	유형	설명	기본값
gtf.model.binShadersLocation	string	바이너리 셰이더가 있어야 하는 GTF 바이너리 디렉터리 아래의 폴더 이름입니다. 절대 경로가 될 수도 있습니다.	

구성 항목	유형	설명	기본값
gtf.model.readBinShaders	부울	이전 실행에 작성된 바이너리 셰이더가 디스크에서 읽혀야 하는지 여부를 결정합니다.	false
gtf.model.writeBinShaders	부울	나중의 실행 시 재사용될 디스크에 바이너리 셰이더를 써야 하는지 결정합니다.	false

### 12.6.1. EB GUIDE Studio의 예제 model.json



**예 12.1.**  
EB GUIDE Studio의 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": "DirectXRenderer",
        "context": 0
    }
},
"identifier": "179c9d9b-be5e-4419-b603-01b31d152089",
"pluginstoload": [
    "GtfGui",
    "GtfGuiDirectX11"
]
}
}
}
}

```

## 12.7. OpenGL ES 확장

OpenGL 및 OpenGL3 렌더러는 아래 열거된 몇몇 OpenGL ES 확장을 이용합니다.

확장을 사용할 수 없는 경우 렌더링 기능에 제한이 발생합니다.

**[참고]**



**아래 표의 여러 확장**

OR와 함께 열거된 여러 확장: 열거된 확장 중 하나를 사용할 수 없을 경우 제한이 적용됩니다.

AND와 함께 열거된 여러 확장: 열거된 모든 확장을 사용할 수 없을 경우 제한이 적용됩니다.

표 12.120. OpenGL 확장

OpenGL 확장	OpenGL 렌더러에서 사용할 수 없을 경우 제한	OpenGL3 렌더러에서 사용할 수 없을 경우 제한
GL_OES_get_program_binary	바이너리 셰이더 캐시 지원 안 됨	없음
GL_EXT_texture_compression_s3tc AND GL_EXT_texture_compression_dxt1	DXT1 질감 압축 지원 안 됨	DXT1 질감 압축 지원 안 됨
GL_EXT_texture_compression_s3tc	DXT3 및 DXT5 질감 압축 지원 안 됨	DXT3 및 DXT5 질감 압축 지원 안 됨



OpenGL 확장	OpenGL 렌더러에서 사용할 수 없을 경우 제한	OpenGL3 렌더러에서 사용할 수 없을 경우 제한
GL_IMG_texture_compression_pvrtpc	PVRTC 질감 압축 지원 안 됨	PVRTC 질감 압축 지원 안 됨
GL_IMG_texture_compression_pvrtpc2	PVRTC2 질감 압축 지원 안 됨	PVRTC2 질감 압축 지원 안 됨
GL_KHR_texture_compression_astc_ldr	ASTC 질감 압축 지원 안 됨	ASTC 질감 압축 지원 안 됨
GL_OES_depth32 AND GL_OES_depth24	<ul style="list-style-type: none"> <li>▶ 심도가 덜한 정밀도로 렌더링된 장면 그래프</li> <li>▶ 위젯의 수가 1600만 대신 65543로 제한됨</li> <li>▶ EB GUIDE SDK 인터페이스 프레임버퍼는 심도가 16비트 이상인 정밀도로 생성될 수 없음</li> </ul>	없음
GL_EXT_shader_texture_lod OR GL_EXT_texture_rg OR GL_OES_texture_float OR GL_OES_texture_half_float	이미지 기반 조명 지원 안 됨	없음
GL_EXT_multisampled_render_to_texture AND GL_IMG_multisampled_render_to_texture AND (GL_NV_framebuffer_blitORGL_NV_framebuffer_multisample)	멀티샘플링은 장면 그래프 및 EB GUIDE SDK 오프스크린 프레임버퍼에 대해 지원 안 됨	없음



OpenGL 확장	OpenGL 렌더러에서 사용할 수 없을 경우 제한	OpenGL3 렌더러에서 사용할 수 없을 경우 제한
AND  (GL_ANGLE_framebuffer_blitORGL_ANGLE_framebuffer_multisample)		
GL_OES_texture_float	장면 그래프에 대한 톤 매핑 지원 안 됨	없음
GL_EXT_texture_border_clamp  AND  GL_OES_texture_border_clamp  AND  GL_NV_texture_border_clamp	알파 마스크 위젯 외부 영역에 시각적 오류 발생 가능	알파 마스크 위젯 외부 영역에 시각적 오류 발생 가능
GL_OES_element_index_uint	기능 제한은 없으나 성능 저하 가능성 있음	없음
GL_OES_packed_depth_stencil	기능 제한은 없으나 성능 저하 가능성 있음	없음
GL_OES_vertex_half_float	기능 제한은 없으나 성능 감소 및 그래픽 메모리 소비량 증대 가능성 있음	없음
GL_OES_vertex_type_10_10_10_2	기능 제한은 없으나 성능 감소 및 그래픽 메모리 소비량 증대 가능성 있음	없음
GL_EXT_sRGB	기능 제한은 없으나 sRGB 색상 모드 또는 sRGB 질감에서 장면 그래프 성능이 감소될 가능성 있음	없음

## 12.8. platform.json 구성 파일

platform.json은 EB GUIDE TF 구성 파일로 일반 및 플랫폼에 따라 달라지는 항목을 포함합니다.

platform.json 파일은 내보낸 EB GUIDE 모델의 일부입니다.

다음 테이블은 모든 기본 구성 매개변수에 대해 문서로 사용됩니다.

**[참고]**



**JSON 객체 표기법**

EB GUIDE Studio내에서 platform.json을 구성할 경우 JSON 객체 표기법을 사용하십시오.

예는 [섹션 12.8.1, “EB GUIDE Studio의 예제 platform.json”](#)를 참조하십시오.

JSON 형식에 대한 자세한 내용은 <http://www.json.org>를 참조하십시오.

표 12.121. 플랫폼 구성

구성 항목	유형	설명	기본값
gtf.servicemapper.port	정수	서버서비스의 연결 포트(예: EB GUIDE Monitor)	60000
gtf.core.pluginstoload	문자열 목록	코어 플러그인 목록을 업로드해야 함(이진수 폴더 또는 절대 경로과 관련됨)	None
gtf.launcher.editmode	boolean	EB GUIDE Studio에서 EB GUIDE TF를 실행할지 결정합니다. 읽기 전용 항목입니다.	false
gtf.platform.config	string	platform.json 파일에 대한 전체 경로입니다. 읽기 전용 항목입니다.	<model_folder>/platform.json
gtf.framework.path	string	GtfStartup 실행 파일에 대한 경로입니다. 읽기 전용 항목입니다.	<binary_folder>
gtf.diagnostic.memory.interval	정수	메모리 진단의 시간 간격을 지정합니다. 값이 0이면 진단이 비활성화됩니다.	0
gtf.ipc.role	string	IPC 노드의 역할입니다. 가능한 값은 server 또는 client입니다.	server
gtf.ipc.discovery.network	string	서버-클라이언트 검색 메커니즘에 사용되는 IPv4 네트워크 주소입니다. 직접 연결의 경우 서버의 네트워크 주소를 나타냅니다.	255.255.255.255





구성 항목	유형	설명	기본값
gtf.ipc.discovery.port	정수	서버-클라이언트 검색 메커니즘에 사용되는 네트워크 포트입니다. 직접 연결의 경우 서버 구성의 gtf-servicemapper.port 항목과 동일해야 합니다.	4711
gtf.ipc.datapool.config	string	IPC 통신의 일부여야 하는 데이터풀 항목을 포함하는 구성 파일입니다.	ipc_datapool.gtf
gtf.ipc.discovery.mode	string	서버와 클라이언트를 연결하는 데 사용되는 검색 모드입니다. 가능한 옵션은 다음과 같습니다. "broadcast", "multicast" 및 "direct".	broadcast
gtf.ipc.client.timeout	정수	클라이언트와 서버 간 연결에 대한 재시도 기간이며 밀리초로 표현됩니다.	5000

### 12.8.1. EB GUIDE Studio의 예제 platform.json



**예 12.2.**  
EB GUIDE Studio의 platform.json

```
{
  "gtf": {
    "core": {
      "pluginstoload": [
        "GtfService",
        "GtfRuntime"
      ]
    },
    "servicemapper": {
      "port": 60000
    }
  }
}
```

```
}

```

## 12.9. 장면

표 12.122. 장면의 속성

속성 이름	설명
height	햅틱 상태 머신의 보기가 타깃 기기에서 렌더링되는 영역의 높이입니다.
width	햅틱 상태 머신의 보기가 타깃 기기에서 렌더링되는 영역의 너비입니다.
x	햅틱 상태 머신의 보기가 타깃 기기에서 렌더링되는 영역의 x 오프셋입니다.
y	햅틱 상태 머신의 보기가 타깃 기기에서 렌더링되는 영역의 y 오프셋입니다.
visible	true인 경우 상태 머신과 해당 하위 위젯이 표시됩니다.
projectName	EB GUIDE 프로젝트의 이름입니다.
windowCaption	창 프레임에 표시되는 텍스트입니다.
sceneID	예를 들어 입력 처리 등에 사용할 수 있는 고유한 장면 식별자입니다.
maxFPS	FPS(초당 프레임) 단위의 다시 그리기 속도입니다. 다시 그리기 속도를 무제한으로 지정하려면 0으로 설정합니다.
hwLayerID	현재 상태 머신에 매핑되는 타깃 기기 디스플레이의 하드웨어 계층 ID입니다.
colorMode	사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ 32-bit (=1): RGBA8888</li> <li>▶ 16-bit (=2): RGB565</li> <li>▶ 24-bit (=3): RGB888</li> <li>▶ 32-bit sRGB (=4):</li> </ul> 이 값은 GPU 하드웨어 지원을 사용합니다. 이미지 위젯이나 <b>확산 질감</b> 위젯 피처에 대한 sRGB 지원을 이용하고 싶은 경우 이 값을 사용하십시오. <ul style="list-style-type: none"> <li>▶ 32-bit sRGB (Emulated) (=5):</li> </ul> 32-bit sRGB가 올바른 결과를 제공하지 않는 경우에만 이 값을 사용하십시오.



속성 이름	설명
multisampling	<p>사용 가능한 값은 다음과 같습니다.</p> <ul style="list-style-type: none"> <li>▶ Off (= 0): 멀티샘플링 없음</li> <li>▶ 2x (=1): 2배 멀티샘플링</li> <li>▶ 4x (=2): 4배 멀티샘플링</li> </ul> <p><a href="#">멀티샘플링 설정</a>도 참조하십시오.</p>
enableRemoteFramebuffer	true인 경우 시뮬레이션 창으로의 화면 외부 버퍼 전송이 활성화됩니다.
showWindowFrame	true인 경우 시뮬레이션 창에 프레임이 표시됩니다. 프레임을 통해 창을 잡아서 이동할 수 있습니다.
showWindow	true인 경우 Windows 기반 시스템에서 시뮬레이션용 추가 창이 열립니다.
disableVSync	true인 경우 렌더러의 가상 동기화가 비활성화됩니다.
showFPS	<p>사용 가능한 값은 다음과 같습니다.</p> <ul style="list-style-type: none"> <li>▶ Off (=0): FPS 표시 안 함</li> <li>▶ On screen (=1): 화면에 FPS 표시</li> <li>▶ Console (=2): 콘솔에 FPS 표시</li> <li>▶ Console &amp; on screen (=3): 화면과 콘솔에 FPS 표시</li> </ul>
Renderer	<p>장면의 렌더러를 정의합니다.</p> <p>사용 가능한 값은 다음과 같습니다.</p> <ul style="list-style-type: none"> <li>▶ DirectXRenderer</li> <li>▶ OpenGLRenderer</li> <li>▶ OpenGL3Renderer</li> </ul>

**[작은 정보]**

**멀티샘플링 설정**



멀티샘플링 해상도가 높을수록 렌더링 결과 품질이 좋습니다. 그렇지만 멀티샘플링은 특히 타깃 기기에서 렌더링 성능을 저하시킵니다. 고해상도의 작은 화면에서는 멀티샘플링의 영향이 거의 없습니다.

멀티샘플링 없이 시작해보고 성능이 좋으면 2배 또는 4배로 멀티샘플링을 설정합니다. 높은 배수의 멀티샘플링에 큰 차이가 없다면 낮은 배수로 설정하여 사용합니다.

**[작은 정보]**

**멀티샘플링 설정은 하드웨어 종속적입니다.**



필요한 멀티샘플링 설정을 하드웨어 측에서 이용할 수 없는 경우 이에 대한 정보가 로그 파일에 제공됩니다.

**[참고]**

**장면 구성에서 sceneID 사용**



장면 구성에서 동일한 sceneID를 사용할 경우 여러 상태 머신이 동시에 입력을 처리하기 위해 반응합니다.

이를 방지하고 단 하나의 상태 머신만 입력 처리에 반응하기를 원한다면 장면 구성에서 각 상태 머신에 서로 다른 sceneID 값을 할당합니다.

## 12.10. EB GUIDE GTF에서 지원하는 터치 스크린 유형

지원되는 유형은 타깃 기기에 따라 달라집니다.

표 12.123. EB GUIDE GTF에서 지원하는 터치 스크린 유형

값	설명	플랫폼
0	Galaxy	Linux
1	IMX WVGA	Linux
2	마우스 인터페이스에 연결된 터치 스크린	모두
3	일반 플랫폼별 터치 스크린 인터페이스	모두
4	Lilliput 889GL	QNX
5	일반 플랫폼별 멀티터치 터치 스크린 인터페이스	Linux

## 12.11. tracing.json 구성 파일

tracing.json은 EB GUIDE TF 구성 파일로 Tracing 종속 항목을 포함합니다.

tracing.json 파일은 내보낸 EB GUIDE 모델이나 표준 제공에 속하지 않습니다. 따라서 PC에 tracing.json 파일을 생성해야 합니다.

다음 테이블은 모든 기본 구성 매개변수에 대해 문서로 사용됩니다.

**[참고]**

**JSON 객체 표기법**



예는 [섹션 12.11.1, "예제 tracing.json"](#)를 참조하십시오.

JSON 형식에 대한 자세한 내용은 <http://www.json.org>를 참조하십시오.

표 12.124. 구성 항목 추적

항목	유형	설명	기본값
gtf.tracing.Plugins	문자열 목록	출력 플러그인 추적	빈 목록, 통합 출력 플러그인은 Android 및 Android용 LogCat 외의 모든 플랫폼용 StdErr/StdErr/StdOut과 같이 사용됩니다.
gtf.tracing.-ChannelSeverities	문자열 목록	지정된 채널에 대한 사용자 정의 심각도 설정	
gtf.tracing.DefaultSeverity	string	기본 추적 심각도	Notice
gtf.tracing.output.path	부울	추적 출력 파일 log.txt이 내보내진 위치	tracing.json이 저장된 것과 동일한 디렉터리
gtf.tracing.output.-enabled	부울	추적 출력 파일 내보내기 활성화	true
gtf.tracing.gtfoutput.severity	string	GtfFileOutput 플러그인의 심각도	
gtf.tracing.-gtfkerneloutput.severity	string	GtfKernelEventOutput 플러그인의 심각도	
gtf.tracing.-gtfvsdebugoutput.severity	string	GtfVsDebugOutput 플러그인의 심각도	
gtf.tracing.stdoutoutput.severity	string	StdErrOutput의 심각도	



항목	유형	설명	기본값
gtf.tracing.logcatoutput.-severity	string	LogCatOutput의 심각도	
gtf.tracing.output.-printChannelName	부울	추적 메시지 내부에 채널 이름을 출력하기 위한 플래그입니다. 구성 항목 이름의 output은 추적 출력 이름(예: gtfoutput, gtfkerneleventoutput, stdout, gtfvsdebugoutput)으로 교체되어야 합니다.	true
gtf.tracing.output.-printTimestamp	부울	추적 메시지 내부에 타임스탬프를 출력하기 위한 플래그입니다. 구성 항목 이름의 output은 추적 출력 이름(예: gtfoutput, gtfkerneleventoutput, stdout, gtfvsdebugoutput)으로 교체되어야 합니다.	true
gtf.tracing.output.-printSeverity	부울	추적 메시지 내부에 심각도를 출력하기 위한 플래그입니다. 구성 항목 이름의 output은 추적 출력 이름(예: gtfoutput, gtfkerneleventoutput, stdout, gtfvsdebugoutput)으로 교체되어야 합니다.	true
gtf.tracing.-PeriodicOutputFlush	정수	주기적인 출력 플러시의 값(밀리초)입니다.  -1 플러시가 전혀 시행되지 않았음을 의미합니다.	-1

## 12.11.1. 예제 tracing.json



**예 12.3.**  
**tracing.json**

```
{
  "gtf":
  {
    "tracing":
    {
      "Plugins": ["GtfFileoutput"],
      "ChannelSeverities":
      ["GTF_Launcher", "Warning",
      "GTF_PluginLoader", "Notice"],
      "DefaultSeverity": "Info"
    }
  }
}
```

## 12.12. 위젯

### 12.12.1. 보기

표 12.125. 보기의 속성

속성 이름	설명
name	위젯의 이름입니다.
height	위젯의 높이(픽셀)입니다.
width	위젯의 너비(픽셀)입니다.
visible	true인 경우 위젯과 해당 하위 위젯이 표시됩니다.
x	위젯의 x 좌표입니다.
y	위젯의 y 좌표입니다.

보기 상태와 보기 템플릿은 관점 전환 애니메이션의 추가 속성을 가집니다. 관점 전환 애니메이션은 진입 애니메이션, 종료 애니메이션, 변경 애니메이션, 애니메이션 팝업 켜기, 애니메이션 팝업 끄기에 적용됩니다.

표 12.126. 관점 전환 애니메이션의 속성

속성 이름	설명
enabled	애니메이션이 실행되는지 여부를 정의합니다.



속성 이름	설명
repeat	반복 횟수입니다. 값이 0이면 애니메이션이 무한 반복됩니다.
alternating	true인 경우 애니메이션이 반복적으로 앞뒤로/양방향으로 실행됩니다. false인 경우 애니메이션이 한 방향/단일 방향으로만 반복적으로 실행됩니다. 반복 횟수는 repeat 속성에서 정의됩니다.
scale	애니메이션 시간에 곱하는 계수입니다.
onPlay	애니메이션을 시작하거나 계속하면 실행되는 반응입니다. 매개변수: 시작 시간 및 재생 방향입니다(true인 경우 정방향, false인 경우 역방향).
onPause	애니메이션을 일시 중지하면 실행되는 반응입니다. 매개변수: 현재 애니메이션 시간입니다.
onTerminate	애니메이션이 완료되면 실행되는 반응입니다. 첫 번째 매개변수: 애니메이션 시간입니다. 두 번째 매개변수: 종료 이유입니다. 다음과 같이 인코딩됩니다. ▶ 0: 애니메이션이 완료되었습니다. ▶ 1: 애니메이션이 취소되었습니다(f:animation_cancel에 의해 트리거됨). ▶ 2: 보기 전이로 인해 위젯이 소멸되었습니다. ▶ 3: 애니메이션이 마지막 단계로 이동되었습니다(f:animation_cancel_end에 의해 트리거됨). ▶ 4: 애니메이션이 첫 단계로 이동했다가 취소되었습니다(f:animation_cancel_reset에 의해 트리거됨).

### 12.12.2. 기본 위젯

기본 위젯은 다음과 같이 8가지입니다.

- ▶ 알파 마스크
- ▶ 애니메이션
- ▶ 컨테이너
- ▶ 타원
- ▶ 이미지
- ▶ 인스턴스화기
- ▶ 레이블
- ▶ 사각형

다음 섹션 목록에서는 기본 위젯의 속성을 소개합니다.



**[참고]**



**고유 이름**

상위 위젯이 같은 두 위젯에는 고유 이름을 사용합니다.

**[참고]**



**음수**

height 및 width 속성에 음수를 입력하지 마십시오. EB GUIDE Studio는 음수를 0으로 처리하므로 해당 위젯이 표시되지 않습니다.

### 12.12.2.1. 알파 마스크

알파 마스크는 이미지를 통해 그 하위 위젯의 알파 채널(즉, 투명도)을 제어하는 컨테이너 위젯입니다.

표 12.127. 알파 마스크의 속성

속성 이름	설명
visible	true인 경우 위젯과 해당 하위 위젯이 표시됩니다.
width	위젯의 너비(픽셀)입니다.
height	위젯의 높이(픽셀)입니다.
x	상위 위젯을 기준으로 한 위젯의 x 좌표입니다.
y	상위 위젯을 기준으로 한 위젯의 y 좌표입니다.
enabled	true인 경우 알파 마스크가 하위 위젯에 적용됩니다.
image	알파 채널, 즉 하위 위젯의 투명도를 제어하는 이미지입니다.
horizontalAlign	위젯 경계 내의 이미지 파일 가로 맞춤입니다.
verticalAlign	위젯 경계 내의 이미지 파일 세로 맞춤입니다.
scaleMode	이미지의 크기 조정 모드입니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ original size (=0)</li> <li>▶ fit to size (=1)</li> <li>▶ keep aspect ratio (=2)</li> </ul>

**[참고]**



**알파 마스크에 지원되는 이미지 파일 유형**

사용 가능한 이미지 형식은 렌더러 구현에 따라 달라집니다. DirectX 11 및 OpenGL ES 버전 2.0 이상에서는 .png 파일과 .jpg 파일을 지원합니다. RGB 이미지는 알파 마스크로 사용되기 전에 흑백 이미지로 변환됩니다. 흑백 이미지는 그대로 사용됩니다. 이미지의 알파 채널은 무시됩니다.

나인패치 이미지로는 알파 마스크를 사용할 수 없습니다.

**12.12.2.2. 애니메이션**

애니메이션은 보기와 함께 위젯의 이동을 정의합니다. 애니메이션의 모양을 정의하려면 **애니메이션** 편집자에 커브를 추가하십시오.

표 12.128. 애니메이션의 속성

속성 이름	설명
enabled	애니메이션이 실행되는지 여부를 정의합니다.
repeat	반복 횟수입니다. 값이 0이면 애니메이션이 무한 반복됩니다.
alternating	true인 경우 애니메이션이 반복적으로 앞뒤로/양방향으로 실행됩니다. false인 경우 애니메이션이 한 방향/단일 방향으로만 반복적으로 실행됩니다. 반복 횟수는 repeat 속성에서 정의됩니다.
scale	애니메이션 시간에 곱하는 계수입니다.
onPlay	애니메이션을 시작하거나 계속하면 실행되는 반응입니다. 매개변수: 시작 시간 및 재생 방향입니다(true인 경우 정방향, false인 경우 역방향).
onPause	애니메이션을 일시 중지하면 실행되는 반응입니다. 매개변수: 현재 애니메이션 시간입니다.
onTerminate	애니메이션이 완료되면 실행되는 반응입니다. 첫 번째 매개변수: 애니메이션 시간입니다. 두 번째 매개변수: 종료 이유입니다. 다음과 같이 인코딩됩니다. ▶ 0: 애니메이션이 완료되었습니다. ▶ 1: 애니메이션이 취소되었습니다(f:animation_cancel에 의해 트리거됨). ▶ 2: 보기 전이로 인해 위젯이 소멸되었습니다. ▶ 3: 애니메이션이 마지막 단계로 이동되었습니다(f:animation_cancel_end에 의해 트리거됨). ▶ 4: 애니메이션이 첫 단계로 이동했다가 취소되었습니다(f:animation_cancel_reset에 의해 트리거됨).



**12.12.2.2.1. 상수 커브**

상수 커브는 정의된 연기 시간 이후 타깃 값을 설정합니다. 정수, 부울, 부동 및 색 유형에 대해 상수 커브를 사용할 수 있습니다.

표 12.129. 상수 커브의 속성

속성 이름	설명
enabled	애니메이션이 실행되는지 여부를 정의합니다.
delay	애니메이션 시작을 기준으로 하는 연기 시간(ms)입니다.
duration	커브 세그먼트의 기간(ms)입니다.
repeat	반복 횟수입니다. 값이 0이면 애니메이션이 무한 반복됩니다.
alternating	애니메이션이 반복적으로 실행되는지 여부를 정의합니다.
relative	초기 값에 업데이트 값이 적용되는지 여부를 정의합니다.
value	결과 상수 값입니다.
target	결과 값이 지정되는 대상 속성입니다.

**12.12.2.2.2. 빠른 시작 커브**

빠른 시작 커브는 시작 부분에서는 빠르게 증가하지만 끝 부분으로 갈수록 증가 속도가 일정하게 느려지는 값을 주기적으로 설정합니다. 정수, 부동 및 색 유형에 대해 빠른 시작 커브를 사용할 수 있습니다.

표 12.130. 빠른 시작 커브의 속성

속성 이름	설명
enabled	애니메이션이 실행되는지 여부를 정의합니다.
delay	애니메이션 시작을 기준으로 하는 연기 시간(ms)입니다.
duration	커브 세그먼트의 기간(ms)입니다.
repeat	반복 횟수입니다. 값이 0이면 애니메이션이 무한 반복됩니다.
alternating	애니메이션이 반복적으로 실행되는지 여부를 정의합니다.
relative	초기 값에 업데이트 값이 적용되는지 여부를 정의합니다.
start	초기 값입니다.
end	최종 값입니다.
target	결과 값이 지정되는 대상 속성입니다.

**12.12.2.2.3. 느린 시작 커브**

느린 시작 커브는 시작 부분에서는 느리게 증가하지만 끝 부분으로 갈수록 증가 속도가 일정하게 높아지는 값을 주기적으로 설정합니다. 정수, 부동 및 색 유형에 대해 느린 시작 커브를 사용할 수 있습니다.



표 12.131. 느린 시작 커브의 속성

속성 이름	설명
enabled	애니메이션이 실행되는지 여부를 정의합니다.
delay	애니메이션 시작을 기준으로 하는 연기 시간(ms)입니다.
duration	커브 세그먼트의 기간(ms)입니다.
repeat	반복 횟수입니다. 값이 0이면 애니메이션이 무한 반복됩니다.
alternating	애니메이션이 반복적으로 실행되는지 여부를 정의합니다.
relative	초기 값에 업데이트 값이 적용되는지 여부를 정의합니다.
start	초기 값입니다.
end	최종 값입니다.
target	결과 값이 지정되는 대상 속성입니다.

**12.12.2.2.4. 정방향 커브**

정방향 커브는 정방향 함수 커브를 사용하여 값을 주기적으로 설정합니다. 정수, 부동 및 색 유형에 대해 정방향 커브를 사용할 수 있습니다.

표 12.132. 정방향 커브의 속성

속성 이름	설명
enabled	애니메이션이 실행되는지 여부를 정의합니다.
delay	애니메이션 시작을 기준으로 하는 연기 시간(ms)입니다.
duration	커브 세그먼트의 기간(ms)입니다.
repeat	반복 횟수입니다. 값이 0이면 애니메이션이 무한 반복됩니다.
alternating	애니메이션이 반복적으로 실행되는지 여부를 정의합니다.
relative	초기 값에 업데이트 값이 적용되는지 여부를 정의합니다.
acceleration	커브의 가속입니다.
velocity	결과를 계산하는 속도입니다.
constant	결과를 계산하는 상수 값입니다.
target	결과 값이 지정되는 대상 속성입니다.

**12.12.2.2.5. 공동 커브**

공동 커브는 공동 함수 커브를 사용하여 값을 주기적으로 설정합니다. 정수, 부동 및 색 유형에 대해 공동 커브를 사용할 수 있습니다.

표 12.133. 공동 커브의 속성

속성 이름	설명
enabled	애니메이션이 실행되는지 여부를 정의합니다.
delay	애니메이션 시작을 기준으로 하는 연기 시간(ms)입니다.
duration	커브 세그먼트의 기간(ms)입니다.
repeat	반복 횟수입니다. 값이 0이면 애니메이션이 무한 반복됩니다.
alternating	애니메이션이 반복적으로 실행되는지 여부를 정의합니다.
relative	초기 값에 업데이트 값이 적용되는지 여부를 정의합니다.
amplitude	공동 커브의 진폭입니다.
constant	결과를 계산하는 상수 값입니다.
frequency	커브 주파수(Hz)입니다.
phase	각도 위상 변환(라디안)입니다.
target	결과 값이 지정되는 대상 속성입니다.

#### 12.12.2.2.6. 스크립트 커브

스크립트 커브는 EB GUIDE 스크립트에서 설명하는 커브를 사용하여 값을 설정합니다. 정수, 부울, 부동 및 색 유형에 대해 스크립트 커브를 사용할 수 있습니다.

표 12.134. 스크립트 커브의 속성

속성 이름	설명
enabled	애니메이션이 실행되는지 여부를 정의합니다.
delay	애니메이션 시작을 기준으로 하는 연기 시간(ms)입니다.
duration	커브 세그먼트의 기간(ms)입니다.
repeat	반복 횟수입니다. 값이 0이면 애니메이션이 무한 반복됩니다.
alternating	애니메이션이 반복적으로 실행되는지 여부를 정의합니다.
relative	초기 값에 업데이트 값이 적용되는지 여부를 정의합니다.
curve	그 결과인 커브 함수는 아래 예를 참조하십시오.
target	결과 값이 지정되는 대상 속성입니다.



#### 예 12.4. Sinus 스크립트 커브의 예

부동 유형의 Sinus 스크립트 커브를 생성하려면 다음 EB GUIDE 스크립트를 입력합니다.

```
function(v:diff::int, v:t_anim::int)
{
    dp:xFloat+=0.2
    f:floor(dp:xFloat*dp:xFloat)
}
```

### 12.12.2.2.7. 선형 커브

선형 커브는 선형 진행 커브를 사용하여 값을 주기적으로 설정합니다. 정수, 부동 및 색 유형에 대해 선형 커브를 사용할 수 있습니다.

표 12.135. 선형 커브의 속성

속성 이름	설명
enabled	애니메이션이 실행되는지 여부를 정의합니다.
delay	애니메이션 시작을 기준으로 하는 연기 시간(ms)입니다.
duration	커브 세그먼트의 기간(ms)입니다.
repeat	반복 횟수입니다. 값이 0이면 애니메이션이 무한 반복됩니다.
alternating	애니메이션이 반복적으로 실행되는지 여부를 정의합니다.
relative	초기 값에 업데이트 값이 적용되는지 여부를 정의합니다.
velocity	결과를 계산하는 속도입니다.
constant	결과를 계산하는 상수 값입니다.
target	결과 값이 지정되는 대상 속성입니다.

### 12.12.2.2.8. 선형 보간 커브

선형 보간 커브 위젯은 선형 보간 커브를 사용하여 값을 주기적으로 설정합니다. 정수, 부동 및 색 유형에 대해 선형 보간 커브를 사용할 수 있습니다.

**[참고]**



**선형 키값 보간 커브**

3D 그래픽 파일을 가져오는 동안 가져온 3D 장면애 애니메이션이 있는 경우, 선형 키값 보간 정수 커브 및 선형 키값 보간 부동 커브가 생성됩니다. 이러한 커브의 근본적인 키값 쌍은 EB GUIDE Studio에서 수정할 수 없습니다.

표 12.136. 선형 보간 커브의 속성

속성 이름	설명
enabled	애니메이션이 실행되는지 여부를 정의합니다.

속성 이름	설명
delay	애니메이션 시작을 기준으로 하는 연기 시간(ms)입니다.
duration	커브 세그먼트의 기간(ms)입니다.
repeat	반복 횟수입니다. 값이 0이면 애니메이션이 무한 반복됩니다.
alternating	애니메이션이 반복적으로 실행되는지 여부를 정의합니다.
relative	초기 값에 업데이트 값이 적용되는지 여부를 정의합니다.
target	결과 값이 지정되는 대상 속성입니다.

### 12.12.2.3. 컨테이너

컨테이너는 여러 위젯을 하위 위젯으로 포함함으로써 위젯을 그룹화합니다.

표 12.137. 컨테이너의 속성

속성 이름	설명
height	위젯의 높이(픽셀)입니다.
width	위젯의 너비(픽셀)입니다.
visible	true인 경우 위젯과 해당 하위 위젯이 표시됩니다.
x	상위 위젯을 기준으로 한 위젯의 x 좌표입니다.
y	상위 위젯을 기준으로 한 위젯의 y 좌표입니다.

### 12.12.2.4. 타원

타원은 위젯의 치수와 좌표를 사용하여 색이 지정된 타원을 보기에 그립니다. 위젯으로 부채꼴이나 원호 모양을 그릴 수도 있습니다.

표 12.138. 타원의 속성

속성 이름	설명
height	위젯의 높이(픽셀)입니다.
width	위젯의 너비(픽셀)입니다.
visible	true인 경우 위젯과 해당 하위 위젯이 표시됩니다.
x	상위 위젯을 기준으로 한 위젯의 x 좌표입니다.
y	상위 위젯을 기준으로 한 위젯의 y 좌표입니다.
fillColor	타원에 채워지는 색입니다.

속성 이름	설명
arcWidth	타원 호의 너비
centralAngle	타원 부채꼴을 정의하는 각도(도)
sectorRotation	타원 부채꼴의 회전을 설명하는 각도(도)

### 12.12.2.5. 이미지

이미지는 보기에 그림을 배치합니다.

표 12.139. 이미지의 속성

속성 이름	설명
height	위젯의 높이(픽셀)입니다.
width	위젯의 너비(픽셀)입니다.
visible	true인 경우 위젯과 해당 하위 위젯이 표시됩니다.
x	상위 위젯을 기준으로 한 위젯의 x 좌표입니다.
y	상위 위젯을 기준으로 한 위젯의 y 좌표입니다.
image	위젯에 표시되는 이미지입니다.
sRGB	이 속성이 활성화된 경우 image에서 선택된 이미지가 sRGB 색 공간을 이용하여 렌더링됩니다.  sRGB 기능을 사용하려면 구성 > colorMode 속성에 대한 <b>프로필</b> 아래의 프로젝트 센터에서 32-bit sRGB (=4) 또는 32-bit sRGB (Emulated) (=5)를 선택하십시오.
horizontalAlign	위젯 경계 내의 이미지 파일 가로 맞춤입니다.
verticalAlign	위젯 경계 내의 이미지 파일 세로 맞춤입니다.

**[참고]**

**지원되는 이미지 파일 유형**



사용 가능한 이미지 형식은 렌더러 구현에 따라 달라집니다. DirectX 11 및 OpenGL ES 버전 2.0 이상에서는 .png 파일과 .jpg 파일을 지원합니다.

### 12.12.2.6. 인스턴스화기

인스턴스화기는 런타임 중에 위젯 인스턴스를 만듭니다. 인스턴스화기를 사용하여 동적 또는 정적 콘텐츠가 있는 목록이나 테이블을 모델링할 수 있습니다. 인스턴스화기의 하위 위젯은 런타임 중에 작성되는 목록이나 테이블의 선 템플릿으로 사용됩니다. 인스턴스화기는 기본적으로 처음 줄 템플릿만 인스턴스화합니다.



표 12.140. 인스턴스화기의 속성

속성 이름	설명
height	위젯의 높이(픽셀)입니다.
width	위젯의 너비(픽셀)입니다.
visible	true인 경우 위젯과 해당 하위 위젯이 표시됩니다.
x	상위 위젯을 기준으로 한 위젯의 x 좌표입니다.
y	상위 위젯을 기준으로 한 위젯의 y 좌표입니다.
numItems	인스턴스화되는 하위 위젯의 수입니다. numItems가 0이면 하위 위젯이 생성되지 않습니다.
lineMapping	어떤 하위 위젯이 각 줄에 대한 줄 템플릿인지 즉, 인스턴스화 순서를 정의합니다.

### 12.12.2.7. 레이블

레이블은 보기에 텍스트를 배치합니다.

표 12.141. 레이블의 속성

속성 이름	설명
height	위젯의 높이(픽셀)입니다.
width	위젯의 너비(픽셀)입니다.
visible	true인 경우 위젯과 해당 하위 위젯이 표시됩니다.
x	상위 위젯을 기준으로 한 위젯의 x 좌표입니다.
y	상위 위젯을 기준으로 한 위젯의 y 좌표입니다.
text	레이블에 표시되는 텍스트입니다. 위젯 영역에 맞지 않는 텍스트는 기본적으로 끝부분이 잘립니다.
textColor	텍스트가 표시되는 색입니다.
font	텍스트가 표시되는 글꼴입니다.
horizontalAlign	레이블 경계 내의 텍스트 가로 맞춤입니다.
verticalAlign	레이블 경계 내의 텍스트 세로 맞춤입니다.

### 12.12.2.8. 사각형

사각형은 위젯의 치수와 좌표를 사용하여 색이 지정된 사각형을 보기에 그립니다.



표 12.142. 사각형의 속성

속성 이름	설명
height	위젯의 높이(픽셀)입니다.
width	위젯의 너비(픽셀)입니다.
visible	true인 경우 위젯과 해당 하위 위젯이 표시됩니다.
x	상위 위젯을 기준으로 한 위젯의 x 좌표입니다.
y	상위 위젯을 기준으로 한 위젯의 y 좌표입니다.
fillColor	사각형에 채워지는 색입니다.

### 12.12.3. 애니메이션

다음 섹션에서는 **애니메이션** 범주의 위젯 속성을 소개합니다.

#### 12.12.3.1. 애니메이션

애니메이션은 그 상위 위젯에 영향을 줍니다. 애니메이션을 사용하려면 하위 위젯으로 커브가 하나 이상 필요 합니다.

표 12.143. 애니메이션의 속성

속성 이름	설명
name	애니메이션의 이름입니다.
alternating	애니메이션이 반복적으로 실행되는지 여부를 정의합니다.
repeat	반복 횟수입니다. 값이 0이면 애니메이션이 무한 반복됩니다.
enabled	애니메이션이 실행되는지 여부를 정의합니다.
scale	애니메이션 시간에 곱하는 계수입니다.
onPause	애니메이션을 일시 중지하면 실행되는 반응입니다. 매개변수: 현재 애니메이션 시간입니다.
onPlay	애니메이션을 시작하거나 계속하면 실행되는 반응입니다. 매개변수: 시작 시간 및 재생 방향입니다(true인 경우 정방향, false인 경우 역방향).
onTerminate	애니메이션이 완료되면 실행되는 반응입니다. 첫 번째 매개변수: 애니메이션 시간입니다. 두 번째 매개변수: 종료 이유입니다. 다음과 같이 인코딩됩니다. <ul style="list-style-type: none"> <li>▶ 0: 애니메이션이 완료되었습니다.</li> <li>▶ 1: 애니메이션이 취소되었습니다(에 의해 트리거됨). f:animation_cancel</li> </ul>



속성 이름	설명
	<ul style="list-style-type: none"> <li>▶ 2: 보기 전이로 인해 위젯이 소멸되었습니다.</li> <li>▶ 3: 애니메이션이 마지막 단계로 이동되었습니다(에 의해 트리거됨). f:animation_cancel_end</li> <li>▶ 4: 애니메이션이 첫 단계로 이동했다가 취소되었습니다(에 의해 트리거됨). f:animation_cancel_reset</li> </ul>

### 12.12.3.2. 상수 커브

상수 커브는 애니메이션의 하위 위젯입니다. 상수 커브는 정의된 연기 시간 이후 타깃 값을 설정합니다. 정수, 부울, 부동 및 색 유형에 대해 상수 커브를 사용할 수 있습니다.

표 12.144. 상수 커브의 속성

속성 이름	설명
name	커브의 이름입니다.
delay	애니메이션 시작을 기준으로 하는 연기 시간(ms)입니다.
duration	커브 세그먼트의 기간(ms)입니다.
enabled	애니메이션이 실행되는지 여부를 정의합니다.
alternating	애니메이션이 반복적으로 실행되는지 여부를 정의합니다.
relative	초기 값에 업데이트 값이 적용되는지 여부를 정의합니다.
repeat	반복 횟수입니다.
target	결과 값이 지정되는 타깃 속성입니다.
value	결과 상수 값입니다.

### 12.12.3.3. 빠른 시작 커브

빠른 시작 커브는 애니메이션의 하위 위젯입니다. 빠른 시작 커브는 시작 부분에서는 빠르게 증가하지만 끝 부분으로 갈수록 증가 속도가 일정하게 느려지는 값을 주기적으로 설정합니다. 정수, 부동 및 색 유형에 대해 빠른 시작 커브를 사용할 수 있습니다.

표 12.145. 빠른 시작 커브의 속성

속성 이름	설명
name	커브의 이름입니다.
delay	애니메이션 시작을 기준으로 하는 연기 시간(ms)입니다.

속성 이름	설명
duration	커브 세그먼트의 기간(ms)입니다.
enabled	애니메이션이 실행되는지 여부를 정의합니다.
alternating	애니메이션이 반복적으로 실행되는지 여부를 정의합니다.
relative	초기 값에 업데이트 값이 적용되는지 여부를 정의합니다.
repeat	반복 횟수입니다.
target	결과 값이 지정되는 타겟 속성입니다.
start	초기 값입니다.
end	최종 값입니다.

#### 12.12.3.4. 느린 시작 커브

느린 시작 커브는 애니메이션의 하위 위젯입니다. 느린 시작 커브는 시작 부분에서는 느리게 증가하지만 끝 부분으로 갈수록 증가 속도가 일정하게 높아지는 값을 주기적으로 설정합니다. 정수, 부동 및 색 유형에 대해 느린 시작 커브를 사용할 수 있습니다.

표 12.146. 느린 시작 커브의 속성

속성 이름	설명
name	커브의 이름입니다.
delay	애니메이션 시작을 기준으로 하는 연기 시간(ms)입니다.
duration	커브 세그먼트의 기간(ms)입니다.
enabled	애니메이션이 실행되는지 여부를 정의합니다.
alternating	애니메이션이 반복적으로 실행되는지 여부를 정의합니다.
relative	초기 값에 업데이트 값이 적용되는지 여부를 정의합니다.
repeat	반복 횟수입니다.
target	결과 값이 지정되는 타겟 속성입니다.
start	초기 값입니다.
end	최종 값입니다.

#### 12.12.3.5. 정방향 커브

정방향 커브는 애니메이션의 하위 위젯입니다. 정방향 커브는 정방향 함수 커브를 사용하여 값을 주기적으로 설정합니다. 정수, 부동 및 색 유형에 대해 정방향 커브를 사용할 수 있습니다.

표 12.147. 정방향 커브의 속성

속성 이름	설명
name	커브의 이름입니다.
delay	애니메이션 시작을 기준으로 하는 연기 시간(ms)입니다.
duration	커브 세그먼트의 기간(ms)입니다.
enabled	애니메이션이 실행되는지 여부를 정의합니다.
alternating	애니메이션이 반복적으로 실행되는지 여부를 정의합니다.
relative	초기 값에 업데이트 값이 적용되는지 여부를 정의합니다.
repeat	반복 횟수입니다.
target	결과 값이 지정되는 타겟 속성입니다.
velocity	결과를 계산하는 속도입니다.
acceleration	커브의 가속입니다.
constant	결과를 계산하는 상수 값입니다.

### 12.12.3.6. 공동 커브

공동 커브는 애니메이션의 하위 위젯입니다. 공동 커브는 공동 함수 커브를 사용하여 값을 주기적으로 설정합니다. 정수, 부동 및 색 유형에 대해 공동 커브를 사용할 수 있습니다.

표 12.148. 공동 커브의 속성

속성 이름	설명
name	커브의 이름입니다.
delay	애니메이션 시작을 기준으로 하는 연기 시간(ms)입니다.
duration	커브 세그먼트의 기간(ms)입니다.
enabled	애니메이션이 실행되는지 여부를 정의합니다.
alternating	애니메이션이 반복적으로 실행되는지 여부를 정의합니다.
relative	초기 값에 업데이트 값이 적용되는지 여부를 정의합니다.
repeat	반복 횟수입니다.
target	결과 값이 지정되는 타겟 속성입니다.
amplitude	공동 커브의 진폭입니다.
constant	결과를 계산하는 상수 값입니다.
phase	각도 위상 변환(라디안)입니다.
frequency	커브 주파수(Hz)입니다.

### 12.12.3.7. 스크립트 커브

스크립트 커브는 애니메이션의 하위 위젯입니다. 스크립트 커브는 EB GUIDE 스크립트에서 설명하는 커브를 사용하여 값을 설정합니다. 정수, 부울, 부동 및 색 유형에 대해 스크립트 커브를 사용할 수 있습니다.

표 12.149. 스크립트 커브의 속성

속성 이름	설명
name	커브의 이름입니다.
delay	애니메이션 시작을 기준으로 하는 연기 시간(ms)입니다.
duration	커브 세그먼트의 기간(ms)입니다.
enabled	애니메이션이 실행되는지 여부를 정의합니다.
alternating	애니메이션이 반복적으로 실행되는지 여부를 정의합니다.
relative	초기 값에 업데이트 값이 적용되는지 여부를 정의합니다.
repeat	반복 횟수입니다.
target	결과 값이 지정되는 타겟 속성입니다.
curve	결과 커브 함수입니다.

### 12.12.3.8. 선형 커브

선형 커브는 애니메이션의 하위 위젯입니다. 선형 커브는 선형 진행 커브를 사용하여 값을 주기적으로 설정합니다. 정수, 부동 및 색 유형에 대해 선형 커브를 사용할 수 있습니다.

표 12.150. 선형 커브의 속성

속성 이름	설명
name	커브의 이름입니다.
delay	애니메이션 시작을 기준으로 하는 연기 시간(ms)입니다.
duration	커브 세그먼트의 기간(ms)입니다.
enabled	애니메이션이 실행되는지 여부를 정의합니다.
alternating	애니메이션이 반복적으로 실행되는지 여부를 정의합니다.
relative	초기 값에 업데이트 값이 적용되는지 여부를 정의합니다.
repeat	반복 횟수입니다.
target	결과 값이 지정되는 타겟 속성입니다.
velocity	결과를 계산하는 속도입니다.

### 12.12.3.9. 선형 보간 커브

선형 보간 커브는 애니메이션의 하위 위젯입니다. 선형 보간 커브 위젯은 선형 보간 커브를 사용하여 값을 주기적으로 설정합니다. 정수, 부동 및 색 유형에 대해 선형 보간 커브를 사용할 수 있습니다.

### [참고]



#### 선형 키값 보간 커브

3D 그래픽 파일을 가져오는 동안 가져온 3D 장면애 애니메이션이 있는 경우, 선형 키값 보간 정수 커브 및 선형 키값 보간 부동 커브가 생성됩니다. 이러한 커브의 근본적인 키값 쌍은 EB GUIDE Studio에서 수정할 수 없습니다.

표 12.151. 선형 보간 커브의 속성

속성 이름	설명
name	커브의 이름입니다.
delay	애니메이션 시작을 기준으로 하는 연기 시간(ms)입니다.
duration	커브 세그먼트의 기간(ms)입니다.
enabled	애니메이션이 실행되는지 여부를 정의합니다.
alternating	애니메이션이 반복적으로 실행되는지 여부를 정의합니다.
relative	초기 값에 업데이트 값이 적용되는지 여부를 정의합니다.
repeat	반복 횟수입니다.
target	결과 값이 지정되는 타겟 속성입니다.
start	초기 값입니다.
end	최종 값입니다.

## 12.12.4. 3D 위젯

### 12.12.4.1. 주변 조명

주변 조명은 장면을 균일하게 비춰주는 조명입니다. 주변 조명은 재료 위젯의 ambient 색 속성에 영향을 미칩니다.

표 12.152. 주변 조명의 속성

속성 이름	설명
enabled	true인 경우 위젯이 활성화됩니다.
color	조명 색
intensity	조명의 강도입니다. 값이 0.0이면 주변 조명이 없는 것입니다.

### 12.12.4.2. 카메라

카메라는 특정 관점에서 장면의 보기를 정의합니다. 다른 관점에서 장면을 표시하려면 여러 카메라를 사용합니다.

표 12.153. 카메라의 속성

속성 이름	설명
enabled	true인 경우 위젯이 활성화됩니다.
nearPlane	장면이 표시되는 보기 방향의 카메라에서 가장 가까운 거리
farPlane	장면이 표시되는 보기 방향의 카메라에서 가장 먼 거리
fieldOfView	카메라의 세로 보기 각도(도)
projectionType	카메라의 투영 유형을 정의합니다. 개체는 perspective (=0) 또는 orthographic (=1) 투영 중 하나로 렌더링됩니다.  참고: 투영 유형이 정사영인 경우 fieldOfView 각도를 사용하여 보기 볼륨이 계산됩니다.

### 12.12.4.3. 방향 조명

하나의 방향 조명은 한 방향에서 장면을 비춥니다.

표 12.154. 방향 조명의 속성

속성 이름	설명
enabled	true인 경우 위젯이 활성화됩니다.
color	조명의 색상
intensity	조명의 강도입니다. 값이 0.0이면 방향 조명이 없는 것입니다.

### 12.12.4.4. 이미지 기반 조명

이미지 기반 조명은 .pfm 파일에 저장된 실시간 조명 정보로 장면을 미추는 조명입니다. .pfm 파일은 .ebibl 파일을 생성하는 IBLGenerator에 대한 입력 데이터로 사용됩니다.

표 12.155. 이미지 기반 조명의 속성

속성 이름	설명
enabled	true인 경우 위젯이 활성화됩니다.
ibl	수동으로 생성된 .ebibl IBL 파일입니다.
intensity	조명의 강도입니다. 값이 0.0이면 이미지 기반 조명이 없는 것입니다.



### 12.12.4.5. 재료

재료는 Phong 반사 모델을 이용해 메시 표면의 시각적 모양을 정의합니다.

표 12.156. 재료의 속성

속성 이름	설명
ambient	주변 조명에서 비추는 경우 객체가 반영하는 색상입니다. 주변 조명이 상위 장면 그래프에 추가되지 않았다면 이 속성은 아무 영향도 미치지 않습니다.
diffuse	단일한 백색광을 비추는 경우 객체가 모든 방향에서 고르게 반영하는 색상
emissive	자체적으로 비추는 객체의 색상
shininess	광도 요소
specular	빛나는 표면을 반영하는 객체의 색상
opacity	불투명 값 0.3과 같이 0과 1 사이의 값만 유효합니다.

### 12.12.4.6. 메시

메시는 3D 객체의 도형을 정의합니다.

표 12.157. 메시의 속성

속성 이름	설명
visible	true인 경우 위젯과 해당 하위 위젯이 표시됩니다.
mesh	자동으로 만들어진 **.ebmesh 메시 파일
culling	메시에서 추린 삼각형이 없는지(0), 전면 삼각형만 있는지(1) 또는 후면 삼각형만 있는지(2) 여부를 정의합니다.

### 12.12.4.7. PBR GGX 재료

PBR GGX 재료는 물리적으로 올바른 Cook-Torrance 모델을 이용하여 메시 표면의 시각적 모양을 정의합니다.

표 12.158. PBR GGX 재료의 속성

속성 이름	설명
ambient	주변 조명에서 비추는 경우 객체가 반영하는 색상
diffuse	단일한 백색광을 비추는 경우 객체가 모든 방향에서 고르게 반영하는 색상
emissive	자체적으로 비추는 객체의 색상

속성 이름	설명
specular	빛나는 표면을 반영하는 객체의 색상
metallic	금속성 표면 품질에 대한 값 이 값은 확산과 정반사성 기여 사이를 보간합니다. 0.3과 같이 0과 1 사이의 값만 유효합니다.
roughness	거친 표면 품질에 대한 값 이 값은 표면의 미세 구조를 제어합니다. 0.3과 같이 0과 1 사이의 값만 유효합니다.
opacity	불투명 값 0.3과 같이 0과 1 사이의 값만 유효합니다.

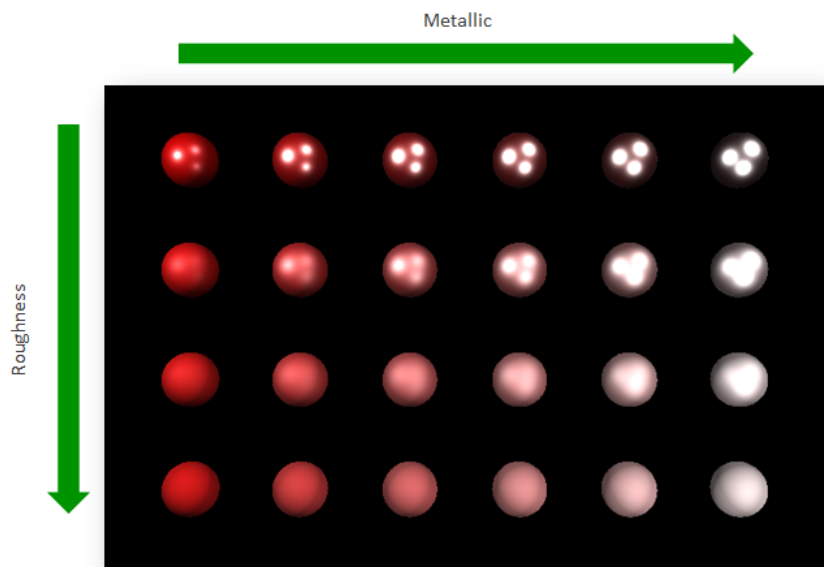


그림 12.1. 물리적 기반 재료의 예

#### 12.12.4.8. PBR Phong 재료

PBR Phong 재료는 물리적으로 올바른 Phong 반사 모델을 이용하여 메시 표면의 시각적 모양을 정의합니다.

표 12.159. PBR Phong 재료의 속성

속성 이름	설명
ambient	주변 조명에서 비추는 경우 객체가 반영하는 색상

속성 이름	설명
diffuse	단일한 백색광을 비추는 경우 객체가 모든 방향에서 고르게 반영하는 색상
emissive	자체적으로 비추는 객체의 색상
shininess	광도 요소
specular	빛나는 표면을 반영하는 객체의 색상
metallic	금속성 표면 품질에 대한 값  이 값은 확산과 정반사성 기여 사이를 보간합니다.  0.3과 같이 0과 1 사이의 값만 유효합니다.
opacity	불투명 값  0.3과 같이 0과 1 사이의 값만 유효합니다.

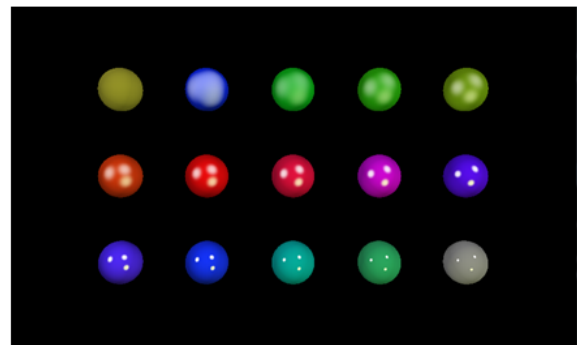
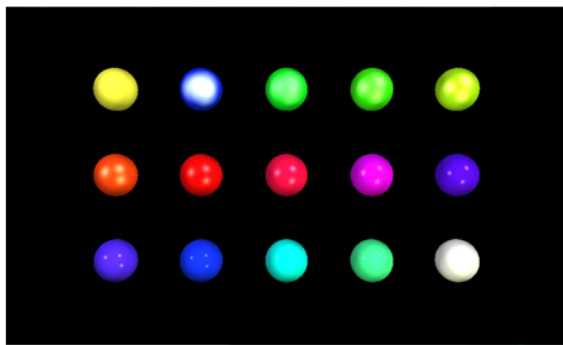


그림 12.2. 비정규화된 재료(왼쪽) 및 정규화된 재료(오른쪽)의 예

### 12.12.4.9. 포인트 조명

포인트 조명은 백열 전구처럼 모든 방향에서 빛을 발산하는 장면에 조명을 추가합니다.

표 12.160. 포인트 조명의 속성

속성 이름	설명
enabled	true인 경우 위젯이 활성화됩니다.
color	조명의 색상
intensity	조명의 강도입니다. 값이 0.0이면 포인트 조명이 없는 것입니다.
attenuationConstant	거리가 증가함에 따라 조명이 약해지는 상수 계수
attenuationLinear	거리가 증가함에 따라 조명이 약해지는 선형 계수
attenuationQuadratic	거리가 증가함에 따라 조명이 약해지는 정방향 계수

### 12.12.4.10. 장면 그래프

장면 그래프를 통해 보기에 3D 객체를 배치합니다.

표 12.161. 장면 그래프의 속성

속성 이름	설명
visible	true인 경우 위젯과 해당 하위 위젯이 표시됩니다.
width	위젯의 너비(픽셀)입니다.
height	위젯의 높이(픽셀)입니다.
x	상위 위젯을 기준으로 한 위젯의 x 좌표입니다.
y	상위 위젯을 기준으로 한 위젯의 y 좌표입니다.

### 12.12.4.11. 장면 그래프 노드

장면 그래프 노드는 하위 노드이며 장면 그래프 또는 다른 장면 그래프 노드에 추가됩니다. 장면 그래프 노드를 사용하여 변환 속성이 있는 3D 장면에 3D 위젯을 배치할 수 있습니다. 장면 그래픽 노드에 다음 3D 위젯을 추가할 수 있습니다.

- ▶ 카메라
- ▶ 방향 조명
- ▶ 이미지 기반 조명
- ▶ 메시
- ▶ 포인트 조명
- ▶ 스포트 조명

표 12.162. 장면 그래프 노드의 속성

속성 이름	설명
visible	true인 경우 위젯과 해당 하위 위젯이 표시됩니다.
rotationX	x축 기준 회전
rotationY	y축 기준 회전
rotationZ	z축 기준 회전
scalingX	x축을 따라 크기 조정
scalingY	y축을 따라 크기 조정
scalingZ	z축을 따라 크기 조정
translationX	x축을 따라 변환
translationY	y축을 따라 변환

속성 이름	설명
translationZ	z축을 따라 변환

### 12.12.4.12. 스포트 조명

스포트 조명은 영향권에 대한 조명을 제한하는 조명을 추가합니다.

표 12.163. 스포트 조명의 속성

속성 이름	설명
enabled	true인 경우 위젯이 활성화됩니다.
color	조명의 색상
intensity	조명의 강도입니다. 값이 0.0이면 스포트 조명이 없는 것입니다.
attenuationConstant	거리가 증가함에 따라 조명이 약해지는 상수 계수
attenuationLinear	거리가 증가함에 따라 조명이 약해지는 선형 계수
attenuationQuadratic	거리가 증가함에 따라 조명이 약해지는 정방향 계수
coneAngleInner	조명의 내부 원뿔각(°)
coneAngleOuter	조명의 외부 원뿔각(°)

## 12.13. 위젯 피처

아래 목록에는 구현되는 모든 위젯 피처에 대한 설명과 EB GUIDE 모델에서 해당 위젯 피처를 사용하는 방법에 대한 간략한 설명이 나와 있습니다.

### 12.13.1. 일반

#### 12.13.1.1. 하위 가시성 선택

**하위 가시성 선택** 위젯 피처는 하위 위젯의 표시 여부를 처리합니다. 한 번에 하나의 하위 위젯 콘텐츠만 표시됩니다.

표 12.164. 하위 가시성 선택 위젯 피처의 속성

속성 이름	설명
containerIndex	상위 위젯의 하위 위젯 인덱스입니다.



속성 이름	설명
containerMapping	<p>매핑이 설정된 경우에는 컨테이너의 각 하위 항목이 containerMapping의 해당 값을 기준으로 다시 주소 지정됩니다.</p> <p>매핑이 설정 또는 정의되지 않았거나 길이가 컨테이너의 하위 위젯 수와 일치하지 않으면 매핑이 사용되지 않습니다. 대신 위젯 트리의 위젯 순서를 인덱스로 사용합니다. 최상위 하위 항목의 인덱스는 0이고 그 다음 항목의 인덱스는 1인 식으로 인덱스가 결정됩니다.</p>

### 12.13.1.2. 활성화됨

활성화됨 위젯 피처는 위젯에 enabled 속성을 추가합니다.

표 12.165. 활성화됨 위젯 피처의 속성

속성 이름	설명
enabled	true인 경우 위젯이 터치 및 누르기 입력에 반응합니다.

### 12.13.1.3. 포커스됨

포커스됨 위젯 피처는 위젯에 입력 포커스를 적용할 수 있도록 설정합니다.

표 12.166. 포커스됨 위젯 피처의 속성

속성 이름	설명
focusable	<p>위젯이 포커스를 받는지 여부를 정의합니다. 사용 가능한 값은 다음과 같습니다.</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	true인 경우 위젯에 포커스가 적용된 것입니다.

### 12.13.1.4. 여러 줄

여러 줄 위젯 피처를 사용하면 줄 바꿈을 사용할 수 있습니다.

제한:

- ▶ 여러 줄 위젯 피처는 레이블 위젯에만 사용할 수 있습니다.

표 12.167. 여러 줄 위젯 피처의 속성

속성 이름	설명
lineGap	줄 사이의 간격 크기입니다. 값이 음수이면 간격이 작아지고 양수이면 간격이 커 집니다.  line gap이 너무 작으면(큰 음수값) 더 영향을 미치지 않으며 텍스트가 하나의 줄 에 렌더링됩니다. 예를 들어 글꼴 스타일이 PT_Sans_Narrow, 크기가 30으로 설정 되고 line gap이 -50으로 지정되어 있는 경우가 그 예입니다.
maxLineCount	표시되는 줄의 최대 수입니다. 값이 0이면 줄 수에 제한이 없는 것입니다.

**[작은 정보]**

**사용된 줄 수**



스크립트 함수인 `getLineCount`를 사용하여 텍스트의 줄 수를 확인할 수 있습니다.

자세한 내용은 [섹션 12.4.3.2.29, "getLineCount"](#) 항목을 참조하십시오.

**[참고]**

**문자 바꾸기**



'\n\n' 시퀀스는 '\w'로 바뀝니다. '\n'n' 시퀀스는 '\wn'으로 바뀝니다.

레이블 크기가 증가하여 한 줄에 텍스트를 모두 표시할 수 있는 경우 '\wn'은 ''으로 바뀝니다.

**12.13.1.5. 누름**

누름 위젯 피처는 위젯을 누를 수 있음을 정의합니다.

제한:

- ▶ 누름 위젯 피처를 추가하면 **포커스됨** 위젯 피처가 자동으로 추가됩니다.

표 12.168. 누름 위젯 피처의 속성

속성 이름	설명
pressed	true인 경우 위젯에 포커스가 적용되어 있는 동안 키를 누른 것입니다.

터치함 위젯 피처와 **터치 누름** 위젯 피처를 함께 사용하면 푸시 버튼을 모델링할 수 있습니다.

**12.13.1.6. 선택됨**

**선택됨** 위젯 피처는 위젯에 `selected` 속성을 추가합니다. 이 기능은 대개 응용 프로그램 또는 HMI 모델러에 의 해 설정되며, 프레임워크의 다른 구성 요소를 통해서 변경되지 않습니다.



표 12.169. **선택됨** 위젯 피처의 속성

속성 이름	설명
selected	true인 경우 위젯이 선택됩니다.

### 12.13.1.7. 선택 그룹

**선택 그룹** 위젯 피처는 라디오 버튼 목록을 모델링하는 데 사용됩니다. 목록에서 모든 라디오 버튼에는 **선택 그룹** 위젯 피처와 고유한 버튼 ID가 있습니다.

buttonValue 속성에는 데이터풀 항목을 사용합니다. 라디오 버튼 배열의 모든 위젯에 데이터풀 항목을 지정합니다.

buttonValue 속성을 설정하는 응용 프로그램을 통해 버튼 그룹 내에서 위젯을 선택하고 선택을 취소할 수 있습니다. 터치 또는 키 입력을 통해 변경을 트리거할 뿐 아니라 버튼 값을 설정하는 조건을 추가하여 변경을 트리거할 수도 있습니다.

제한:

- ▶ **선택 그룹** 위젯 피처를 추가하면 **선택됨** 위젯 피처가 자동으로 추가됩니다.

표 12.170. **선택 그룹** 위젯 피처의 속성

속성 이름	설명
buttonId	버튼 그룹 내의 버튼을 식별하는 ID입니다.
buttonValue	버튼의 현재 값입니다. 이 값이 buttonId와 일치하면 버튼이 선택됩니다.
selected	buttonID와 buttonValue가 동일한지를 평가합니다. true인 경우 버튼이 선택됩니다.

### 12.13.1.8. 스피닝

**스피닝** 위젯 피처는 위젯을 회전식 버튼으로 설정합니다. **스피닝** 위젯 피처가 포함된 위젯은 내부 값을 변경하여 증가 및 감소 이벤트에 반응합니다. **스피닝** 위젯 피처를 사용하면 크기 조정, 진행 표시줄 또는 미리보기 값이 포함된 위젯을 만들 수 있습니다.

표 12.171. **스피닝** 위젯 피처의 속성

속성 이름	설명
currentValue	현재 회전 값입니다.
maxValue	currentValue 속성의 최대값입니다.
minValue	currentValue 속성의 최소값입니다.
incValueTrigger	true인 경우 currentValue 속성이 1 증가합니다.





속성 이름	설명
incValueReaction	currentValue 속성 증가에 대한 반응입니다.
decValueTrigger	true인 경우 현재 값이 1 감소합니다.
decValueReaction	currentValue 속성 감소에 대한 반응입니다.
steps	currentValue 속성 증가 또는 감소를 계산할 단계의 수입니다.
valueWrapAround	사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ true: minValue 또는 maxValue가 초과되더라도 역 경계에서 currentValue 속성이 계속 적용됩니다.</li> <li>▶ false: minValue 또는 maxValue가 초과되면 currentValue 속성이 증가/감소하지 않습니다.</li> </ul>

### 12.13.1.9. 텍스트 잘림

텍스트 잘림 위젯 피처는 text 속성의 내용이 위젯 영역에 맞지 않으면 해당 텍스트를 자릅니다. 위젯 피처를 사용하면 기본 설정 trailing이 아닌 다른 잘림을 사용할 수 있습니다.

제한:

- ▶ 텍스트 잘림 위젯 피처는 레이블 위젯에만 사용할 수 있습니다.

표 12.172. 텍스트 잘림 위젯 피처의 속성

속성 이름	설명
truncationPolicy	한 줄로 된 텍스트의 경우 truncationPolicy 속성은 잘림 위치를 정의합니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ leading (=0): 시작 부분에서 텍스트가 바뀝니다.</li> <li>▶ trailing (=1): 끝 부분에서 텍스트가 바뀝니다.</li> </ul> 여러 줄로 된 텍스트의 경우 truncationPolicy 속성은 텍스트가 바뀌는 위치를 정의합니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ leading (=0): 시작 요소의 줄이 바뀌고 처음으로 표시되는 줄의 텍스트가 시작 요소에서 잘립니다.</li> <li>▶ trailing (=1) 끝 요소의 줄이 바뀌고 마지막으로 표시되는 줄의 텍스트가 끝 요소에서 잘립니다.</li> </ul>
truncationSymbol	바뀐 텍스트 요소 대신 표시되는 문자열

### 12.13.1.10. 터치함

**터치함** 위젯 피처는 위젯이 터치 입력에 반응할 수 있도록 설정합니다.

표 12.173. 터치함 위젯 피처의 속성

속성 이름	설명
touchable	true인 경우 위젯이 터치 입력에 반응합니다.
touched	true인 경우 현재 위젯을 터치한 것입니다.
touchPolicy	위젯 경계를 교차하는 터치 및 이동을 처리할 방법을 정의합니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ Press then react (=0): 먼저 누르면 위젯이 반응합니다. 이동 및 놓기 알림은 위젯 영역 내에서만 활성화됩니다.</li> <li>▶ Press and grab (=1): 접촉부를 눌러서 잡습니다. 접촉부는 위젯 영역 외부로 이동하더라도 잡은 상태로 유지됩니다.</li> <li>▶ Press then react on contact (=3): 접촉부가 위젯 경계 외부에서 누른 상태로 설정되더라도 후속 이동 및 놓기 이벤트가 위젯으로 전달됩니다.</li> </ul>
touchBehavior	터치 평가를 정의합니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ Whole area (=0): 렌더러는 터치한 위젯을 식별하기 위해 위젯의 자르기 사각형을 평가합니다.</li> <li>▶ Visible pixels (=1): 렌더러는 터치한 위젯을 식별하기 위해 터치한 픽셀이 속하는 위젯을 평가합니다.</li> </ul> 알파 투명도가 적용된 이미지 내의 투명 픽셀 또는 O, A 등의 문자 내에 있는 픽셀은 터치할 수 없습니다.  Visible pixels 값은 레이블에 영향을 주지 않습니다.

**터치함** 위젯 피처와 **누름** 위젯 피처를 함께 사용하면 푸시 버튼을 모델링할 수 있습니다.

**[작은 정보] 성능 관련 권장 사항:**



프로젝트에서 성능이 중요한 문제인 경우에는 touchBehavior 속성을 Whole area (=0)으로 설정하십시오. EB GUIDE GTF는 Visible pixels (=1)보다 Whole area (=0)가 더 빠릅니다.

## 12.13.2. 효과

### 12.13.2.1. 테두리

**테두리** 위젯 피처는 위젯에 구성 가능한 테두리를 추가합니다. 테두리는 위젯 경계에서 시작되며 위젯 내에 배치됩니다.

제한:

- ▶ 위젯 피처를 사각형에 사용할 수 있어야 합니다.

표 12.174. 테두리 위젯 피처의 속성

속성 이름	설명
borderThickness	테두리의 두께(픽셀)입니다.
borderColor	테두리를 렌더링하는 데 사용되는 색입니다.
borderStyle	테두리를 렌더링하는 데 사용되는 스타일입니다.

### 12.13.2.2. 천연색

**천연색** 위젯 피처는 위젯과 위젯 하위 트리에 색을 적용합니다. 또한 알파 값이 불투명이 아닌 경우에는 투명도에도 영향을 줍니다.



**예 12.5. 천연색 위젯 피처의 사용법**

RGBA 구성 요소 값이 0.0에서 1.0 사이인 모든 색에 대해 **천연색** 위젯 피처의 알고리즘은 위젯의 현재 색 값에 colorationColor 속성 값을 곱합니다. 이 곱하기는 픽셀 단위로 전체 구성 요소에 대해 수행됩니다.

예를 들어 반투명 회색에 불투명 파랑을 적용하면 다음과 같이 반투명 진한 파랑이 생성됩니다.

$$(0.5, 0.5, 0.5, 0.5) * (0.0, 0.0, 1.0, 1.0) = (0.0, 0.0, 0.5, 0.5)$$

표 12.175. 천연색 위젯 피처의 속성

속성 이름	설명
colorationEnabled	true인 경우 천연색이 사용됩니다.
colorationColor	천연색에 사용되는 색입니다.

### 12.13.2.3. 스트로크

**스트로크** 위젯 피처는 구성 가능한 텍스트 윤곽선, 즉 레이블 테두리를 활성화합니다.

제한:

- ▶ 위젯 피처를 레이블에 사용할 수 있습니다.



표 12.176. 스트로크 위젯 피처의 속성

속성 이름	설명
strokeEnabled	true인 경우 스트로크가 사용됩니다.
strokeThickness	윤곽선의 두께(픽셀)입니다.
strokeColor	윤곽선을 렌더링하는 데 사용되는 색입니다.

### 12.13.3. 포커스

포커스 위젯 피처 범주는 포커스 관리와 관련된 위젯 피처를 제공합니다.

#### 12.13.3.1. 자동 포커스

자동 포커스 위젯 피처를 사용하여 포커스되는 하위 위젯의 순서를 미리 정의합니다. 자동 포커스 위젯 피처는 위젯 하위 트리에서 focusable 속성이 포함된 하위 위젯을 확인합니다.

레이아웃 내의 위젯 순서를 사용하여 포커스 순서를 계산합니다. 레이아웃 방향에 따라 알고리즘이 왼쪽 위 또는 오른쪽 위 모서리에서 시작됩니다.

제한:

- ▶ 위젯 피처 **자동 포커스**는 자동으로 **포커스됨** 위젯 피처를 추가합니다.

표 12.177. 자동 포커스 위젯 피처의 속성

속성 이름	설명
focusNext	포커스 인덱스가 증가하는 조건입니다.
focusPrev	포커스 인덱스가 감소하는 조건입니다.
focusFlow	포커스의 동작은 계층 내에서 변경됩니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ stop at hierarchy (=0)</li> <li>▶ wrap within hierarchy level (=1)</li> <li>▶ step up in hierarchy (=2)</li> </ul>
focusedIndex	현재 포커스된 하위 위젯의 인덱스(포커스 가능한 n 번째 하위 위젯)입니다.
initFocus	이 인덱스에 따라 초기화 시 포커스되는 하위 위젯이 정의됩니다. 위젯이 포커스 가능하지 않은 경우에는 포커스 가능한 다음 하위 위젯이 사용됩니다.

#### 12.13.3.2. 사용자 정의 포커스

**사용자 정의 포커스** 위젯 피처를 사용하면 위젯에 대해 추가 포커스 기능을 활성화할 수 있습니다. 이 기능을 사용하는 위젯은 위젯 하위 트리에 대한 로컬 포커스 계층을 관리합니다.

제한:

- ▶ 위젯 피처 **사용자 정의 포커스**는 자동으로 **포커스됨** 위젯 피처를 추가합니다.

표 12.178. 사용자 정의 포커스 위젯 피처의 속성

속성 이름	설명
focusNext	다음 하위 위젯에 포커스를 지정하는 트리거입니다.
focusOrder	focusOrder 속성을 사용하면 포커스 지정 시 하위 위젯을 건너뛴 수 있습니다. 하위 위젯의 ID는 하위 트리 내의 위젯 위치에 해당합니다. 포커스 가능하지 않은 하위 위젯은 기본적으로 건너뛴니다. 하위 위젯이 포커스되는 순서입니다. <ul style="list-style-type: none"> <li>▶ 정의함: 사용자 정의 위젯 순서가 사용됩니다.</li> <li>▶ 정의하지 않음: 대신 기본 위젯 순서가 사용됩니다.</li> </ul> 각 하위 위젯에는 <b>포커스됨</b> 위젯 피처가 필요합니다. 위젯 피처가 없으면 포커스 처리 시 위젯이 무시됩니다. 예를 들어 focusOrder=1 0 2는 두 번째 위젯, 첫 번째 위젯, 세 번째 위젯이 순서대로 포커스됨을 의미합니다.
focusPrev	이전 하위 위젯에 포커스를 지정하는 트리거입니다.
focusFlow	포커스의 동작은 계층 내에서 변경됩니다. 사용 가능한 값은 다음과 같습니다. <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	인덱스를 통해 focusOrder 목록 내 하위 위젯의 위치가 정의됩니다. 위젯이 포커스 가능하지 않은 경우에는 목록에서 다음 하위 위젯이 사용됩니다.
initFocus	초기화 시 포커스되는 하위 위젯의 인덱스입니다.

## 12.13.4. 제스처

### 12.13.4.1. 굿기 제스처

표면 위에서 접촉부를 빠르게 문지르는 동작

제한:

- ▶ **굿기 제스처** 위젯 피처를 추가하면 **제스처** 및 **터치됨** 위젯 피처가 자동으로 추가됩니다.

표 12.179. 굿기 제스처 위젯 피처의 속성

속성 이름	설명
onGestureFlick	제스처가 인식되면 트리거되는 반응입니다. 반응 인수: ▶ speed: 굿기 제스처의 상대 속도 밀리초당 픽셀 수를 flickMinLength/flickMaxTime으로 나눈 속도 ▶ directionX: 제스처 방향 벡터의 x 요소 ▶ directionY: 제스처 방향 벡터의 y 요소
flickMaxTime	제스처가 굿기 제스처로 인식되도록 하기 위해 접촉 위치를 유지해야 하는 최대 시간(밀리초)입니다.
flickMinLength	제스처가 굿기 제스처로 인식되도록 하기 위해 접촉부가 표면 위에서 이동해야 하는 최소 거리(픽셀)입니다.

### 12.13.4.2. 누르기 제스처

이동하지 않는 누르기 제스처

제한:

- ▶ 누르기 제스처 위젯 피처를 추가하면 제스처 및 터치됨 위젯 피처가 자동으로 추가됩니다.
- ▶ 누르기 제스처 위젯 피처는 터치 잡기 농침 위젯 피처를 트리거하지 않습니다.

표 12.180. 누르기 제스처 위젯 피처의 속성

속성 이름	설명
onGestureHold	제스처가 인식되면 트리거되는 반응입니다. 반응은 접촉부당 한 번만 트리거됩니다. holdDuration 만료 시 접촉부가 초기 터치 위치 주위의 작은 경계 사각형 내에 있어야 합니다. 반응 인수: ▶ x: 접촉부 위치의 x 좌표 ▶ y: 접촉부 위치의 y 좌표
holdDuration	제스처가 누르기 제스처로 인식되도록 하기 위해 접촉 위치를 유지해야 하는 최소 시간(밀리초)입니다.

### 12.13.4.3. 길게 누르기 제스처

이동하지 않는 길게 누르기 제스처

제한:

- ▶ 길게 누르기 제스처 위젯 피처를 추가하면 제스처 및 터치됨 위젯 피처가 자동으로 추가됩니다.
- ▶ 길게 누르기 제스처 위젯 피처는 터치 잡기 놓침 위젯 피처를 트리거하지 않습니다.

표 12.181. 길게 누르기 제스처 위젯 피처의 속성

속성 이름	설명
onGestureLongHold	제스처가 인식되면 트리거되는 반응입니다. 반응은 접촉부당 한 번만 트리거됩니다. longHoldDuration 만료 시 접촉부가 초기 터치 위치 주위의 작은 경계 사각형 내에 있어야 합니다.  반응 인수: <ul style="list-style-type: none"> <li>▶ x: 접촉부 위치의 x 좌표</li> <li>▶ y: 접촉부 위치의 y 좌표</li> </ul>
longHoldDuration	제스처가 길게 누르기 제스처로 인식되도록 하기 위해 접촉 위치를 유지해야 하는 최소 시간(밀리초)입니다.

12.13.4.4. 경로 제스처

접촉부 하나로 그리는 모양이 알려진 도형 집합과 일치하는지를 확인하는 동작

제한:

- ▶ 경로 제스처 위젯 피처를 추가하면 제스처 및 터치됨 위젯 피처가 자동으로 추가됩니다.

표 12.182. 경로 제스처 위젯 피처의 속성

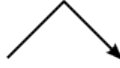
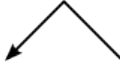




속성 이름	설명
onPath	입력한 도형이 일치하면 트리거되는 반응입니다. onPathStart가 이미 트리거된 경우에만 반응이 트리거됩니다. 반응 인수: <ul style="list-style-type: none"> <li>▶ gestureId: 일치한 경로의 ID입니다.</li> </ul>
onPathStart	접촉부가 최소 상자(pathMinXBox, pathMinYBox)를 벗어나 이동하면 트리거되는 반응입니다.
onPathNotRecognized	입력한 도형이 일치하지 않으면 트리거되는 반응입니다. onPathStart가 이미 트리거된 경우에만 반응이 트리거됩니다.
pathMinXBox	경로 제스처 인식기가 입력 고려를 시작하도록 하려면 접촉부가 이동해야 하는 최소 거리(픽셀)의 x 좌표입니다.

속성 이름	설명
pathMinYBox	경로 제스처 인식기가 입력 고려를 시작하도록 하려면 접촉부가 이동해야 하는 최소 거리(픽셀)의 y 좌표입니다.


### 12.13.4.4.1. 제스처 ID

제스처 식별자는 경로 제스처 인식기의 구성에 따라 달라집니다. 다음 표에는 EB GUIDE에 포함된 예제 구성이 나와 있습니다.

표 12.183. EB GUIDE에 포함된 경로 제스처 샘플 구성

ID	도형	설명
0		왼쪽에서 오른쪽 방향 지붕 모양
1		오른쪽에서 왼쪽 방향 지붕 모양
2		왼쪽에서 오른쪽 방향 가로줄
3		오른쪽에서 왼쪽 방향 가로줄
4		확인 표시
5		왼쪽에서 오른쪽 방향 파도 모양



ID	도형	설명
6		오른쪽에서 왼쪽 방향 파도 모양

### 12.13.4.5. 모으기 제스처

두 접촉부를 서로 가까이 또는 멀리 움직이는 동작

제한:

- ▶ **모으기 제스처** 위젯 피처를 추가하면 **제스처** 및 **터치됨** 위젯 피처가 자동으로 추가됩니다.

표 12.184. 모으기 제스처 위젯 피처의 속성

속성 이름	설명
onGesturePinchStart	제스처 시작이 인식되면 트리거되는 반응입니다. 반응 인수: <ul style="list-style-type: none"> <li>▶ ratio: 초기 접촉부 거리에 대한 현재 접촉부 거리의 비율입니다.</li> <li>▶ centerX: 두 접촉부 사이의 현재 중심 포인트 X 좌표입니다.</li> <li>▶ centerY: 두 접촉부 사이의 현재 중심 포인트 Y 좌표입니다.</li> </ul>
onGesturePinchUpdate	모으기 속도 또는 중심 포인트가 변경되면 트리거되는 반응입니다. 반응 인수: <ul style="list-style-type: none"> <li>▶ ratio: 초기 접촉부 거리에 대한 현재 접촉부 거리의 비율입니다.</li> <li>▶ centerX: 두 접촉부 사이의 현재 중심 포인트 X 좌표입니다.</li> <li>▶ centerY: 두 접촉부 사이의 현재 중심 포인트 Y 좌표입니다.</li> </ul>
onGesturePinchEnd	제스처가 완료되면 트리거되는 반응입니다. 반응 인수: <ul style="list-style-type: none"> <li>▶ ratio: 초기 접촉부 거리에 대한 현재 접촉부 거리의 비율입니다.</li> <li>▶ centerX: 두 접촉부 사이의 현재 중심 포인트 X 좌표입니다.</li> <li>▶ centerY: 두 접촉부 사이의 현재 중심 포인트 Y 좌표입니다.</li> </ul>
pinchThreshold	제스처가 인식되도록 하기 위해 각 접촉부를 초기 위치에서 이동해야 하는 최소 거리(픽셀)입니다.

### 12.13.4.6. 회전 제스처

두 접촉부를 원 모양으로 이동하는 동작

제한:

- ▶ **회전 제스처** 위젯 피처를 추가하면 **제스처** 및 **터치됨** 위젯 피처가 자동으로 추가됩니다.

표 12.185. 회전 제스처 위젯 피처의 속성

속성 이름	설명
onGestureRotateStart	제스처 시작이 인식되면 트리거되는 반응입니다.
onGestureRotateUpdate	인식되는 각도 또는 중심 포인트가 변경되면 트리거되는 반응입니다.
onGestureRotateEnd	제스처가 완료되면 트리거되는 반응입니다.
rotateThreshold	제스처 시작이 인식되도록 하기 위해 각 접촉부를 초기 위치에서 이동해야 하는 최소 거리(픽셀)입니다.

onGestureRotateEnd, onGestureRotateStart, 및 onGestureRotateUpdate의 반응 인수:

- ▶ angle: 회전에 포함되는 두 접촉부의 초기 위치로 지정되는 선과 두 접촉부의 현재 위치로 지정되는 선 사이의 각도입니다. 각도는 시계 반대 방향으로 측정됩니다.
- ▶ centerX: 두 접촉부 사이의 현재 중심 포인트 X 좌표입니다.
- ▶ centerY: 두 접촉부 사이의 현재 중심 포인트 Y 좌표입니다.

## 12.13.5. 입력 처리

### 12.13.5.1. 제스처

**제스처** 위젯 피처는 위젯이 터치 제스처에 반응할 수 있도록 설정합니다.

제한:

- ▶ **제스처** 위젯 피처를 추가하면 **포커스됨** 위젯 피처가 자동으로 추가됩니다.
- ▶ **제스처** 위젯 피처에 추가 속성은 없습니다.

### 12.13.5.2. 키 누름

**키 누름** 위젯 피처는 위젯이 키 누르기에 반응할 수 있도록 설정합니다.

제한:

- ▶ **키 누름** 위젯 피처를 추가하면 **누름** 및 **포커스됨** 위젯 피처가 자동으로 추가됩니다.

표 12.186. 키 누름 위젯 피처의 속성

속성 이름	설명
keyPressed	키를 누를 때의 위젯 반응입니다.  반응 인수: ▶ keyId: 처리된 키의 ID입니다.

### 12.13.5.3. 키 놓음

키 놓음 위젯 피처는 위젯이 키 놓기에 반응할 수 있도록 설정합니다.

제한:

- ▶ 키 놓음 위젯 피처를 추가하면 누름 및 포커스됨 위젯 피처가 자동으로 추가됩니다.

표 12.187. 키 놓음 위젯 피처의 속성

속성 이름	설명
keyShortReleased	키를 놓을 때의 위젯 반응입니다.  반응 인수: ▶ keyId: 처리된 키의 ID입니다.

### 12.13.5.4. 키 상태 변경됨

키 상태 변경됨 위젯 피처는 위젯이 키 누르거나 놓기에 반응할 수 있도록 설정합니다. 이 기능은 짧게 누르기, 길게 누르기, 매우 길게 누르기, 연속 누르기 등의 키 입력에 대한 반응을 정의합니다.

제한:

- ▶ 키 상태 변경됨 위젯 피처를 추가하면 누름 및 포커스됨 위젯 피처가 자동으로 추가됩니다.

표 12.188. 키 상태 변경됨 위젯 피처의 속성

속성 이름	설명
keyStatusChanged	키를 누르거나 놓을 때의 위젯 반응입니다.  반응 인수: ▶ keyId: 처리된 키의 ID입니다. ▶ status: 상태 변경의 숫자 ID입니다.

### 12.13.5.5. 키 유니코드

키 유니코드 위젯 피처는 위젯이 유니코드 키 입력에 반응할 수 있도록 설정합니다.

제한:

- ▶ 키 유니코드 위젯 피처를 추가하면 누름 및 포커스됨 위젯 피처가 자동으로 추가됩니다.

표 12.189. 키 유니코드 위젯 피처의 속성

속성 이름	설명
keyUnicode	유니코드 키 입력에 대한 위젯의 반응입니다.  반응 인수: ▶ keyId: 처리된 키의 ID입니다.

### 12.13.5.6. 안으로 이동

안으로 이동 위젯 피처를 사용하면 위젯이 해당 경계 내로의 이동에 반응하도록 설정할 수 있습니다.

제한:

- ▶ 안으로 이동 위젯 피처를 추가하면 터치됨 위젯 피처가 자동으로 추가됩니다.

표 12.190. 안으로 이동 위젯 피처의 속성

속성 이름	설명
moveIn	경계 내로의 이동에 대한 위젯의 반응입니다.  반응 인수: ▶ touchId: 사용자가 클릭했거나 놓은 터치스크린의 ID입니다. ▶ x: x 좌표 ▶ y: y 좌표 ▶ fingerId: 위젯에서 이동하는 접촉부의 ID입니다.

### 12.13.5.7. 밖으로 이동

밖으로 이동 위젯 피처를 사용하면 위젯이 해당 경계 외부로의 이동에 반응하도록 설정할 수 있습니다.

제한:

- ▶ 밖으로 이동 위젯 피처를 추가하면 터치됨 위젯 피처가 자동으로 추가됩니다.

표 12.191. 밖으로 이동 위젯 피처의 속성

속성 이름	설명
moveOut	경계 외부로의 이동에 대한 위젯의 반응입니다.  반응 인수: <ul style="list-style-type: none"> <li>▶ touchId: 사용자가 클릭했거나 놓은 터치스크린의 ID입니다.</li> <li>▶ x: x 좌표</li> <li>▶ y: y 좌표</li> <li>▶ fingerId: 위젯에서 이동하는 접촉부의 ID입니다.</li> </ul>

### 12.13.5.8. 위로 이동

위로 이동 위젯 피처를 사용하면 위젯이 해당 경계 내에서의 이동에 반응하도록 설정할 수 있습니다.

제한:

- ▶ 위로 이동 위젯 피처를 추가하면 터치됨 위젯 피처가 자동으로 추가됩니다.

표 12.192. 위로 이동 위젯 피처의 속성

속성 이름	설명
moveOver	경계 내에서의 이동에 대한 위젯의 반응입니다.  반응 인수: <ul style="list-style-type: none"> <li>▶ touchId: 사용자가 클릭했거나 놓은 터치스크린의 ID입니다.</li> <li>▶ x: x 좌표</li> <li>▶ y: y 좌표</li> <li>▶ fingerId: 위젯에서 이동하는 접촉부의 ID입니다.</li> </ul>

### 12.13.5.9. 이동 가능

이동 가능 위젯 피처를 사용하면 위젯을 터치로 이동하도록 설정할 수 있습니다.

제한:

- ▶ 이동 가능 위젯 피처를 추가하면 터치됨 및 터치 이동됨 위젯 피처가 자동으로 추가됩니다.

표 12.193. 이동 가능 위젯 피처의 속성

속성 이름	설명
moveDirection	위젯이 이동하는 방향입니다. 사용 가능한 값은 다음과 같습니다.



속성 이름	설명
	<ul style="list-style-type: none"> <li>▶ horizontal (=0)</li> <li>▶ vertical (=1)</li> <li>▶ free (=2)</li> </ul>

### 12.13.5.10. 회전식

회전식 위젯 피처는 위젯이 회전에 반응할 수 있도록 설정합니다.

제한:

- ▶ 회전식 위젯 피처를 추가하면 **포커스됨** 위젯 피처가 자동으로 추가됩니다.

표 12.194. 회전식 위젯 피처의 속성

속성 이름	설명
rotaryReaction	회전할 때의 위젯 반응입니다. true인 경우 위젯이 들어오는 회전식 이벤트에 반응합니다.  반응 인수: <ul style="list-style-type: none"> <li>▶ rotaryId: 정수 ID</li> <li>▶ increment: 들어오는 이벤트가 전송될 때 회전식 입력이 이동하는 단위 수</li> </ul>

### 12.13.5.11. 터치 잡기 놓침

터치 잡기 놓침 위젯 피처를 사용하면 위젯이 터치 접촉부 놓침에 반응하도록 설정할 수 있습니다.

접촉부는 제스처의 일부분일 때나 놓지 않고 터치 화면을 벗어날 때 사라질 수 있습니다. 이러한 경우에는 touchShortReleased 반응이 실행되지 않습니다.

제한:

- ▶ 터치 잡기 놓침 위젯 피처를 추가하면 **터치됨** 위젯 피처가 자동으로 추가됩니다.
- ▶ 터치 잡기 놓침을 추가할 경우 터치됨 위젯 피처의 touchPolicy 드롭다운 메뉴 상자에서 Press and grab을 선택합니다.

터치 잡기 놓침은 다른 터치 정책과 함께 적용되지 않습니다.

표 12.195. 터치 잡기 놓침 위젯 피처의 속성

속성 이름	설명
onTouchGrabLost	터치 접촉부를 놓쳤을 때의 위젯 반응입니다.

속성 이름	설명
	반응 인수: <ul style="list-style-type: none"> <li>▶ touchId: 사용자가 클릭했거나 놓은 터치스크린의 ID입니다.</li> <li>▶ x: x 좌표</li> <li>▶ y: y 좌표</li> <li>▶ fingerId: 위젯에서 이동하는 접촉부의 ID입니다.</li> </ul>

### 12.13.5.12. 터치 이동

**터치 이동** 위젯 피처는 위젯이 터치하여 이동하기에 반응할 수 있도록 설정합니다.

제한:

- ▶ **터치 이동** 위젯 피처를 추가하면 **터치됨** 위젯 피처가 자동으로 추가됩니다.

표 12.196. 터치 이동 위젯 피처의 속성

속성 이름	설명
touchMoved	터치 및 이동할 때의 위젯 반응입니다.  반응 인수: <ul style="list-style-type: none"> <li>▶ touchId: 사용자가 클릭했거나 놓은 터치스크린의 ID입니다.</li> <li>▶ x: x 좌표</li> <li>▶ y: y 좌표</li> <li>▶ fingerId: 위젯에서 이동하는 접촉부의 ID입니다.</li> </ul>

### 12.13.5.13. 터치 누름

**터치 누름** 위젯 피처는 위젯이 누르기에 반응할 수 있도록 설정합니다.

제한:

- ▶ **터치 누름** 위젯 피처를 추가하면 **터치됨** 위젯 피처가 자동으로 추가됩니다.

표 12.197. 터치 누름 위젯 피처의 속성

속성 이름	설명
touchPressed	누를 때의 위젯 반응입니다.  반응 인수:

속성 이름	설명
	<ul style="list-style-type: none"> <li>▶ touchId: 사용자가 클릭했거나 놓은 터치스크린의 ID입니다.</li> <li>▶ x: x 좌표</li> <li>▶ y: y 좌표</li> <li>▶ fingerId: 위젯에서 이동하는 접촉부의 ID입니다.</li> </ul>

### 12.13.5.14. 터치 놓음

터치 놓음 위젯 피처는 위젯이 놓기에 반응할 수 있도록 설정합니다.

제한:

- ▶ 터치 놓음 위젯 피처를 추가하면 **터치됨** 위젯 피처가 자동으로 추가됩니다.

표 12.198. 터치 놓음 위젯 피처의 속성

속성 이름	설명
touchShortReleased	놓을 때의 위젯 반응입니다.  반응 인수: <ul style="list-style-type: none"> <li>▶ touchId: 사용자가 클릭했거나 놓은 터치스크린의 ID입니다.</li> <li>▶ x: x 좌표</li> <li>▶ y: y 좌표</li> <li>▶ fingerId: 위젯에서 이동하는 접촉부의 ID입니다.</li> </ul>

### 12.13.5.15. 터치 상태 변경됨

터치 상태 변경됨 위젯 피처는 위젯이 터치 상태 변경에 반응할 수 있도록 설정합니다.

제한:

- ▶ 터치 상태 변경됨 위젯 피처를 추가하면 **터치됨** 위젯 피처가 자동으로 추가됩니다.

표 12.199. 터치 상태 변경됨 위젯 피처의 속성

속성 이름	설명
touchStatusChanged	터치 상태 변경에 대한 위젯의 반응입니다.  반응 인수: <ul style="list-style-type: none"> <li>▶ touchId: 사용자가 클릭했거나 놓은 터치스크린의 ID입니다.</li> </ul>





속성 이름	설명
	<ul style="list-style-type: none"> <li>▶ x: x 좌표</li> <li>▶ y: y 좌표</li> <li>▶ touchStatus: 터치 유형의 ID</li> </ul> <p>사용 가능한 값은 다음과 같습니다.</p> <ul style="list-style-type: none"> <li>▶ 0: 새 접촉부</li> <li>▶ 1: 터치 누르기</li> <li>▶ 2: 터치 이동</li> <li>▶ 3: 터치 놓음</li> <li>▶ 4: 터치 없이 이동</li> <li>▶ 5: 터치 없음</li> </ul> <ul style="list-style-type: none"> <li>▶ fingerId: 위젯에서 이동하는 접촉부의 ID입니다.</li> </ul>

## 12.13.6. 레이아웃

### 12.13.6.1. 절대 레이아웃

상위 위젯의 **절대 레이아웃** 위젯 피처는 하위 위젯의 위치 및 크기를 정의합니다. 표시되지 않는 하위 위젯은 무시됩니다. 추가된 위젯 피처 속성은 정수 목록으로 구성됩니다. 각 목록 요소는 하나의 하위 위젯에 매핑됩니다.

제한:

- ▶ **절대 레이아웃** 위젯 피처에는 다음 위젯 피처가 제외됩니다.
  - ▶ 상자 레이아웃
  - ▶ 흐름 레이아웃
  - ▶ 눈금 레이아웃
  - ▶ 목록 레이아웃

표 12.200. 절대 레이아웃 위젯 피처의 속성

속성 이름	설명
itemLeftOffset	하위 위젯에 대해 왼쪽 테두리로부터의 오프셋을 저장하는 정수 목록입니다. 각 목록 요소는 하위 위젯에 매핑됩니다.
itemTopOffset	하위 위젯에 대해 위쪽 테두리로부터의 오프셋을 저장하는 정수 목록입니다. 각 목록 요소는 하위 위젯에 매핑됩니다.

속성 이름	설명
itemRightOffset	하위 위젯에 대해 오른쪽 테두리로부터의 오프셋을 저장하는 정수 목록입니다. 각 목록 요소는 하위 위젯에 매핑됩니다.
itemBottomOffset	하위 위젯에 대해 아래쪽 테두리로부터의 오프셋을 저장하는 정수 목록입니다. 각 목록 요소는 하위 위젯에 매핑됩니다.

### 12.13.6.2. 상자 레이아웃

**상자 레이아웃** 위젯 피처는 각 하위 위젯의 위치 및 크기를 정의합니다.

하위 위젯의 위치 및 크기 속성은 상위 위젯에 의해 설정됩니다. 표시되지 않는 하위 위젯은 계산에서 무시됩니다.

제한:

- ▶ **상자 레이아웃** 위젯 피처에는 다음 위젯 피처가 제외됩니다.
  - ▶ **절대 레이아웃**
  - ▶ **흐름 레이아웃**
  - ▶ **눈금 레이아웃**
  - ▶ **목록 레이아웃**

표 12.201. **상자 레이아웃** 위젯 피처의 속성

속성 이름	설명
gap	레이아웃 방향에 따라 두 하위 위젯 간의 공간입니다.
layoutDirection	목록 요소(하위 위젯)가 배치되는 방향입니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ horizontal (=0)</li> <li>▶ vertical (=1)</li> </ul>

### 12.13.6.3. 흐름 레이아웃

**흐름 레이아웃** 위젯 피처는 각 하위 위젯의 위치 및 크기를 정의합니다.

하위 위젯의 위치 및 크기 속성은 상위 위젯에 의해 설정됩니다. 표시되지 않는 하위 위젯은 계산에서 무시됩니다.

제한:

- ▶ **흐름 레이아웃** 위젯 피처에는 다음 위젯 피처가 제외됩니다.
  - ▶ **절대 레이아웃**

- ▶ 상자 레이아웃
- ▶ 눈금 레이아웃
- ▶ 목록 레이아웃

표 12.202. 흐름 레이아웃 위젯 피처의 속성

속성 이름	설명
horizontalGap	두 하위 위젯 간의 가로 공간입니다.
verticalGap	두 하위 위젯 간의 세로 공간입니다.
layoutDirection	목록 요소(하위 위젯)가 배치되는 방향입니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ horizontal (=0)</li> <li>▶ vertical (=1)</li> </ul>
horizontalChildAlign	하위 위젯의 가로 맞춤입니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ leading (=0): 하위 위젯이 가운데에 배치됩니다.</li> <li>▶ center (=1): 하위 위젯이 위쪽에 배치됩니다.</li> <li>▶ trailing (=2): 하위 위젯이 아래쪽에 배치됩니다.</li> </ul>
verticalChildAlign	하위 위젯의 세로 맞춤입니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ center (=0): 하위 위젯이 가운데에 배치됩니다.</li> <li>▶ top (=1): 하위 위젯이 위쪽에 배치됩니다.</li> <li>▶ bottom (=2): 하위 위젯이 아래쪽에 배치됩니다.</li> </ul>

#### 12.13.6.4. 눈금 레이아웃

눈금 레이아웃 위젯 피처는 각 하위 위젯의 위치 및 크기를 정의합니다.

하위 위젯의 위치 및 크기 속성은 상위 위젯에 의해 설정됩니다. 표시되지 않는 하위 위젯은 계산에서 무시됩니다.

제한:

- ▶ 눈금 레이아웃 위젯 피처에는 다음 위젯 피처가 제외됩니다.
  - ▶ 절대 레이아웃
  - ▶ 상자 레이아웃
  - ▶ 흐름 레이아웃
  - ▶ 목록 레이아웃

표 12.203. 눈금 레이아웃 위젯 피처의 속성

속성 이름	설명
horizontalGap	두 하위 위젯 간의 가로 공간입니다.
verticalGap	두 하위 위젯 간의 세로 공간입니다.
numRows	행의 수를 정의합니다.
numColumns	열의 수를 정의합니다.

### 12.13.6.5. 레이아웃 여백

레이아웃 여백 위젯 피처는 **흐름 레이아웃**, **절대 레이아웃**, **상자 레이아웃** 또는 **눈금 레이아웃** 위젯 피처를 사용하는 위젯에 구성 가능한 여백을 추가합니다.

표 12.204. 레이아웃 여백 위젯 피처의 속성

속성 이름	설명
leftMargin	왼쪽 테두리의 여백입니다.
topMargin	위쪽 테두리의 여백입니다.
rightMargin	오른쪽 테두리의 여백입니다.
bottomMargin	아래쪽 테두리의 여백입니다.

### 12.13.6.6. 목록 레이아웃

**목록 레이아웃** 위젯 피처는 각 하위 위젯의 위치 및 크기를 픽셀로 정의합니다.

하위 위젯의 위치 속성과 **목록 인덱스** 위젯 피처의 listIndex 속성은 상위 위젯에 의해 설정됩니다.

하위 위젯을 만들 때 인스턴스화기와 함께 사용하면 가장 효율적입니다.

**목록 인덱스** 위젯 피처에 대한 자세한 내용은 [섹션 12.13.7.2. "목록 인덱스"](#) 항목을 참조하십시오.

제한:

- ▶ **목록 레이아웃** 위젯 피처는 인스턴스화기와 함께 사용합니다.
- ▶ **목록 레이아웃** 위젯 피처에는 다음 위젯 피처가 제외됩니다.
  - ▶ **절대 레이아웃**
  - ▶ **상자 레이아웃**
  - ▶ **흐름 레이아웃**
  - ▶ **눈금 레이아웃**

표 12.205. 목록 레이아웃 위젯 피처의 속성

속성 이름	설명
layoutDirection	목록 요소(하위 위젯)가 배치되는 방향입니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ horizontal (=0)</li> <li>▶ vertical (=1)</li> </ul>
scrollOffset	목록을 스크롤할 픽셀 수입니다.
scrollOffsetRebase	scrollOffsetRebase 속성이 변경되면 현재 scrollOffset이 scrollIndex로 변환됩니다. 나머지 오프셋은 scrollOffset 속성에 기록됩니다.
firstListIndex	첫 번째로 표시되는 목록 요소의 목록 인덱스는 위젯 피처에 의해 정의됩니다.
scrollIndex	scrollOffset 속성이 적용되는 기본 목록 인덱스입니다. 스크롤은 scrollIndex 속성에 지정된 목록 요소에서 시작됩니다.
scrollValue	현재 스크롤 값(픽셀)입니다.
scrollValueMax	목록 끝 부분에 매핑되는 최대 스크롤 값(픽셀)입니다.
scrollValueMin	목록 시작 부분에 매핑되는 최소 스크롤 값(픽셀)입니다.
bounceValue	scrollOffset 속성을 적용한 결과 위치가 올바른 스크롤 범위 내로 지정되는 경우 bounceValue 속성은 0이 됩니다. 스크롤 위치가 목록 시작 부분을 넘어가면 값은 양수가 되고, 스크롤 위치가 목록 끝을 넘어가면 값은 음수가 됩니다. bounceValue를 scrollOffset에 추가하면 스크롤 위치가 다시 범위 내로 설정됩니다.
bounceValueMax	scrollOffset이 유효한 스크롤 범위 외부로 이동할 수 있는 최대값입니다. 사용자가 이보다 멀리 스크롤하려고 하면 scrollOffset은 잘립니다.
segments	가로 레이아웃 방향의 경우 행의 수입니다. 세로 레이아웃 방향의 경우 열의 수입니다.
listLength	목록 요소의 수입니다.
wrapAround	사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ true: scrollValueMin 또는 scrollValueMax가 초과되더라도 역 경계에서 scrollValue 속성이 계속 적용됩니다.</li> <li>▶ false: scrollValueMin 또는 scrollValueMax가 초과되면 scrollValue 속성이 증가/감소하지 않습니다.</li> </ul>

### 12.13.6.7. 크기 조정 모드

크기 조정 모드 위젯 피처는 해당 크기가 위젯의 크기와 다른 경우 이미지가 표시되는 방법을 정의합니다.

제한:

- ▶ **크기 조정 모드** 위젯 피처는 위젯 이미지에만 사용할 수 있습니다.

표 12.206. 크기 조정 모드 위젯 피처의 속성

속성 이름	설명
scaleMode	이미지의 크기 조정 모드입니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ 0 = original size</li> <li>▶ 1 = fit to size</li> <li>▶ 2 = keep aspect ratio</li> </ul>

## 12.13.7. 목록 관리

### 12.13.7.1. 줄 인덱스

**줄 인덱스** 위젯 피처는 목록 또는 테이블의 각 줄의 고유한 위치를 지정합니다.

제한:

- ▶ **줄 인덱스** 위젯 피처는 인스턴스화기와 함께 사용합니다.

표 12.207. 줄 인덱스 위젯 피처의 속성

속성 이름	설명
lineIndex	테이블 내 현재 줄의 인덱스입니다.

### 12.13.7.2. 목록 인덱스

**목록 인덱스** 위젯 피처는 목록의 위젯 위치를 지정합니다.

제한:

- ▶ **목록 인덱스** 위젯 피처와 **목록 레이아웃** 위젯 피처를 함께 사용합니다.

표 12.208. 목록 인덱스 위젯 피처의 속성

속성 이름	설명
listIndex	목록 내 현재 위젯의 인덱스입니다.



### 12.13.7.3. 템플릿 인덱스

템플릿 인덱스 위젯 피처는 사용된 줄 템플릿의 고유한 위치를 지정합니다.

제한:

- ▶ **템플릿 인덱스** 위젯 피처는 인스턴스화기와 함께 사용합니다.

표 12.209. 템플릿 인덱스 위젯 피처의 속성

속성 이름	설명
lineTemplateIndex	사용되는 줄 템플릿의 인덱스입니다.

### 12.13.7.4. 뷰포트

뷰포트 위젯 피처는 위젯 테두리에서 너무 큰 요소를 자릅니다.

제한:

- ▶ **뷰포트** 위젯 피처는 컨테이너 또는 목록과 함께 사용합니다.
- ▶ **뷰포트** 위젯 피처는 다음 모델 요소에 적용됩니다.
  - ▶ **뷰포트**에 추가한 위젯의 하위 위젯은 위젯 치수 내부에 포함되도록 잘립니다.
  - ▶ **뷰포트**에 추가한 위젯은 상위 보기 치수 내부에 포함되도록 잘립니다.

표 12.210. 뷰포트 위젯 피처의 속성

속성 이름	설명
xOffset	하위 위젯에서 그린 영역 내에 표시되는 잘림의 가로 오프셋입니다.
yOffset	하위 위젯에서 그린 영역 내에 표시되는 잘림의 세로 오프셋입니다.

## 12.13.8. 3D

**3D** 범주의 위젯 피처는 3D 위젯에서만 사용할 수 있습니다.

### 12.13.8.1. 카메라 뷰포트

**카메라 뷰포트** 위젯 피처는 장면 그래프 내에서 카메라의 그리기 영역을 정의합니다.

제한:

- ▶ **카메라 뷰포트** 위젯 피처는 카메라에 사용할 수 있습니다.

표 12.211. 카메라 뷰포트 위젯 피처의 속성

속성 이름	설명
viewportX	장면 그래프 내의 x 기준 뷰포트
viewportY	장면 그래프 내의 y 기준 뷰포트
viewportWidth	뷰포트 너비(픽셀)
viewportHeight	뷰포트 높이(픽셀)

### 12.13.8.2. 주변 질감

**주변 질감** 위젯 피처는 확장된 구성 값을 재료에 추가합니다.

제한:

- ▶ **주변 질감** 위젯 피처는 재료, PBR Phong 재료 및 PBR GGX 재료에 대해 사용할 수 있습니다.

표 12.212. 주변 질감 위젯 피처의 속성

속성 이름	설명
ambientTexture	질감의 파일 이름
ambientTextureAddressModeU	u 방향을 따른 질감의 주소 모드. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ repeat (=0): 질감 경계 외부에서 액세스할 때 질감이 반복됩니다. 랩 또는 타일이라고 합니다.</li> <li>▶ clamp (=1): 질감 경계 외부에서 액세스할 때 질감 가장자리의 픽셀이 사용됩니다.</li> </ul>
ambientTextureAddressModeV	v 방향을 따른 질감의 주소 모드입니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ repeat (=0): 질감 경계 외부에서 액세스할 때 질감이 반복됩니다. 랩 또는 타일이라고 합니다.</li> <li>▶ clamp (=1): 질감 경계 외부에서 액세스할 때 질감 가장자리의 픽셀이 사용됩니다.</li> </ul>
ambientFilterMode	질감의 필터링 모드입니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ point (=0): 질감이 전혀 부드럽지 않습니다. 질감이 최소화될 때 가장 저렴하지만 울퉁불퉁한 인공물이 쉽게 발생할 수 있습니다.</li> </ul>





속성 이름	설명
	<ul style="list-style-type: none"> <li>▶ linear (=1): 이선형 필터링이라고도 합니다. 울퉁불퉁한 인공물을 줄이기 위해 최소화할 때 질감을 부드럽게 합니다.</li> <li>▶ trilinear (=2): 가장 비싸지만 선형 필터링보다 더 나은 결과물을 산출합니다.</li> </ul>

### 12.13.8.3. 확산 질감

확산 질감 위젯 피처는 확장된 구성 값을 재료에 추가합니다.

제한:

- ▶ 확산 질감 위젯 피처는 재료, PBR Phong 재료 및 PBR GGX 재료에 대해 사용할 수 있습니다.

표 12.213. 확산 질감 위젯 피처의 속성

속성 이름	설명
diffuseTexture	질감의 파일 이름
diffuseTextureAddressModeU	u 방향을 따른 질감의 주소 모드. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ repeat (=0): 질감 경계 외부에서 액세스할 때 질감이 반복됩니다. 랩 또는 타일이라고 합니다.</li> <li>▶ clamp (=1): 질감 경계 외부에서 액세스할 때 질감 가장자리의 픽셀이 사용됩니다.</li> </ul>
diffuseTextureAddressModeV	v 방향을 따른 질감의 주소 모드입니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ repeat (=0): 질감 경계 외부에서 액세스할 때 질감이 반복됩니다. 랩 또는 타일이라고 합니다.</li> <li>▶ clamp (=1): 질감 경계 외부에서 액세스할 때 질감 가장자리의 픽셀이 사용됩니다.</li> </ul>
diffuseFilterMode	질감의 필터링 모드입니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ point (=0): 질감이 전혀 부드럽지 않습니다. 질감이 최소화될 때 가장 저렴하지만 울퉁불퉁한 인공물이 쉽게 발생할 수 있습니다.</li> <li>▶ linear (=1): 이선형 필터링이라고도 합니다. 울퉁불퉁한 인공물을 줄이기 위해 최소화할 때 질감을 부드럽게 합니다.</li> <li>▶ trilinear (=2): 가장 비싸지만 선형 필터링보다 더 나은 결과물을 산출합니다.</li> </ul>
diffuseSRGB	이 속성이 활성화된 경우 diffuseTexture에서 선택된 질감이 sRGB 색 공간을 이용하여 렌더링됩니다.



속성 이름	설명
	sRGB 기능을 사용하려면 구성 > colorMode 속성에 대한 <b>프로필</b> 아래의 프로젝트 센터에서 32-bit sRGB (=4) 또는 32-bit sRGB (Emulated) (=5)를 선택하십시오.

#### 12.13.8.4. 방사 질감

방사 질감 위젯 피처는 확장된 구성 값을 재료에 추가합니다.

제한:

- ▶ **방사 질감** 위젯 피처는 재료, PBR Phong 재료 및 PBR GGX 재료에 대해 사용할 수 있습니다.

표 12.214. 방사 질감 위젯 피처의 속성

속성 이름	설명
emissiveTexture	질감의 파일 이름
emissiveTextureAddressModeU	u 방향을 따른 질감의 주소 모드. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ repeat (=0): 질감 경계 외부에서 액세스할 때 질감이 반복됩니다. 랩 또는 타일이라고 합니다.</li> <li>▶ clamp (=1): 질감 경계 외부에서 액세스할 때 질감 가장자리의 픽셀이 사용됩니다.</li> </ul>
emissiveTextureAddressModeV	v 방향을 따른 질감의 주소 모드입니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ repeat (=0): 질감 경계 외부에서 액세스할 때 질감이 반복됩니다. 랩 또는 타일이라고 합니다.</li> <li>▶ clamp (=1): 질감 경계 외부에서 액세스할 때 질감 가장자리의 픽셀이 사용됩니다.</li> </ul>
emissiveFilterMode	질감의 필터링 모드입니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ point (=0): 질감이 전혀 부드럽지 않습니다. 질감이 최소화될 때 가장 저렴하지만 울퉁불퉁한 인공물이 쉽게 발생할 수 있습니다.</li> <li>▶ linear (=1): 이선형 필터링이라고도 합니다. 울퉁불퉁한 인공물을 줄이기 위해 최소화할 때 질감을 부드럽게 합니다.</li> <li>▶ trilinear (=2): 가장 비싸지만 선형 필터링보다 더 나은 결과물을 산출합니다.</li> </ul>

#### 12.13.8.5. 조명 지도 질감

조명 지도 질감 위젯 피처는 확장된 구성 값을 재료에 추가합니다.

제한:

- ▶ 조명 지도 질감 위젯 피처는 재료, PBR Phong 재료 및 PBR GGX 재료에 대해 사용할 수 있습니다.

표 12.215. 조명 지도 질감 위젯 피처의 속성

속성 이름	설명
lightMapTexture	질감의 파일 이름
lightMapTextureAddressModeU	u 방향을 따른 질감의 주소 모드. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ repeat (=0): 질감 경계 외부에서 액세스할 때 질감이 반복됩니다. 랩 또는 타일이라고 합니다.</li> <li>▶ clamp (=1): 질감 경계 외부에서 액세스할 때 질감 가장자리의 픽셀이 사용됩니다.</li> </ul>
lightMapTextureAddressModeV	v 방향을 따른 질감의 주소 모드입니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ repeat (=0): 질감 경계 외부에서 액세스할 때 질감이 반복됩니다. 랩 또는 타일이라고 합니다.</li> <li>▶ clamp (=1): 질감 경계 외부에서 액세스할 때 질감 가장자리의 픽셀이 사용됩니다.</li> </ul>
lightMapFilterMode	질감의 필터링 모드입니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ point (=0): 질감이 전혀 부드럽지 않습니다. 질감이 최소화될 때 가장 저렴하지만 울퉁불퉁한 인공물이 쉽게 발생할 수 있습니다.</li> <li>▶ linear (=1): 이선형 필터링이라고도 합니다. 울퉁불퉁한 인공물을 줄이기 위해 최소화할 때 질감을 부드럽게 합니다.</li> <li>▶ trilinear (=2): 가장 비싸지만 선형 필터링보다 더 나은 결과물을 산출합니다.</li> </ul>

### 12.13.8.6. 금속 질감

금속 위젯 피처는 확장된 구성 값을 재료에 추가합니다. 질감은 PBR GGX 재료 및 PBR Phong 재료 위젯의 금속 매개변수를 제어합니다.

제한:

- ▶ 금속 질감 위젯 피처는 재료, PBR GGX 재료 및 PBR Phong 재료에 대해 사용할 수 있습니다.
- ▶ 금속 질감은 흑백 이미지입니다. RGB 색상 이미지의 경우 빨간색 채널만 사용됩니다.
- ▶ 금속 질감이 활성화 상태인 경우 metallic 속성이 무시됩니다.

표 12.216. 금속 질감 위젯 피처의 속성

속성 이름	설명
metallicTexture	질감의 파일 이름
metallicMinFactor	질감 값을 보간하기 위한 플롯으로서의 최소 금속 매개변수
metallicMaxFactor	질감 값을 보간하기 위한 플롯으로서의 최대 금속 매개변수
metallicTextureAddressModeU	u 방향을 따른 질감의 주소 모드. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ repeat (=0): 질감 경계 외부에서 액세스할 때 질감이 반복됩니다. 랩 또는 타일이라고 합니다.</li> <li>▶ clamp (=1): 질감 경계 외부에서 액세스할 때 질감 가장자리의 픽셀이 사용됩니다.</li> </ul>
metallicTextureAddressModeV	v 방향을 따른 질감의 주소 모드입니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ repeat (=0): 질감 경계 외부에서 액세스할 때 질감이 반복됩니다. 랩 또는 타일이라고 합니다.</li> <li>▶ clamp (=1): 질감 경계 외부에서 액세스할 때 질감 가장자리의 픽셀이 사용됩니다.</li> </ul>
metallicFilterMode	질감의 필터링 모드입니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ point (=0): 질감이 전혀 부드럽지 않습니다. 질감이 최소화될 때 가장 저렴하지만 울퉁불퉁한 인공물이 쉽게 발생할 수 있습니다.</li> <li>▶ linear (=1): 이선형 필터링이라고도 합니다. 울퉁불퉁한 인공물을 줄이기 위해 최소화할 때 질감을 부드럽게 합니다.</li> <li>▶ trilinear (=2): 가장 비싸지만 선형 필터링보다 더 나은 결과물을 산출합니다.</li> </ul>

### 12.13.8.7. 일반 지도 질감

일반 지도 질감 위젯 피처는 확장된 구성 값을 재료에 추가합니다.

제한:

- ▶ 일반 지도 질감 위젯 피처는 재료, PBR Phong 재료 및 PBR GGX 재료에 대해 사용할 수 있습니다.

표 12.217. 일반 지도 질감 위젯 피처의 속성

속성 이름	설명
normalMapTexture	질감의 파일 이름



속성 이름	설명
normalMapTextureAddressModeU	u 방향을 따른 질감의 주소 모드. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ repeat (=0): 질감 경계 외부에서 액세스할 때 질감이 반복됩니다. 랩 또는 타일이라고 합니다.</li> <li>▶ clamp (=1): 질감 경계 외부에서 액세스할 때 질감 가장자리의 픽셀이 사용됩니다.</li> </ul>
normalMapTextureAddressModeV	v 방향을 따른 질감의 주소 모드입니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ repeat (=0): 질감 경계 외부에서 액세스할 때 질감이 반복됩니다. 랩 또는 타일이라고 합니다.</li> <li>▶ clamp (=1): 질감 경계 외부에서 액세스할 때 질감 가장자리의 픽셀이 사용됩니다.</li> </ul>
normalMapFilterMode	질감의 필터링 모드입니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ point (=0): 질감이 전혀 부드럽지 않습니다. 질감이 최소화될 때 가장 저렴하지만 울퉁불퉁한 인공물이 쉽게 발생할 수 있습니다.</li> <li>▶ linear (=1): 이선형 필터링이라고도 합니다. 울퉁불퉁한 인공물을 줄이기 위해 최소화할 때 질감을 부드럽게 합니다.</li> <li>▶ trilinear (=2): 가장 비싸지만 선형 필터링보다 더 나은 결과물을 산출합니다.</li> </ul>

### 12.13.8.8. 불투명 질감

**불투명 질감** 위젯 피처는 확장된 구성 값을 재료에 추가합니다.

제한:

- ▶ **불투명 질감** 위젯 피처는 재료, PBR Phong 재료 및 PBR GGX 재료에 대해 사용할 수 있습니다.

표 12.218. 불투명 질감 위젯 피처의 속성

속성 이름	설명
opaqueTexture	질감의 파일 이름
opaqueTextureAddressModeU	u 방향을 따른 질감의 주소 모드. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ repeat (=0): 질감 경계 외부에서 액세스할 때 질감이 반복됩니다. 랩 또는 타일이라고 합니다.</li> <li>▶ clamp (=1): 질감 경계 외부에서 액세스할 때 질감 가장자리의 픽셀이 사용됩니다.</li> </ul>



속성 이름	설명
opaqueTextureAddressModeV	v 방향을 따른 질감의 주소 모드입니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ repeat (=0): 질감 경계 외부에서 액세스할 때 질감이 반복됩니다. 랩 또는 타일이라고 합니다.</li> <li>▶ clamp (=1): 질감 경계 외부에서 액세스할 때 질감 가장자리의 픽셀이 사용됩니다.</li> </ul>
opaqueFilterMode	질감의 필터 모드입니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ point (=0): 질감이 전혀 부드럽지 않습니다. 질감이 최소화될 때 가장 저렴하지만 울퉁불퉁한 인공물이 쉽게 발생할 수 있습니다.</li> <li>▶ linear (=1): 이선형 필터링이라고도 합니다. 울퉁불퉁한 인공물을 줄이기 위해 최소화할 때 질감을 부드럽게 합니다.</li> <li>▶ trilinear (=2): 가장 비싸지만 선형 필터링보다 더 나은 결과물을 산출합니다.</li> </ul>

### 12.13.8.9. 반사 질감

반사 질감 위젯 피처는 확장된 구성 값을 재료에 추가합니다.

제한:

- ▶ 반사 질감 위젯 피처는 재료, PBR Phong 재료 및 PBR GGX 재료에 대해 사용할 수 있습니다.

표 12.219. 반사 질감 위젯 피처의 속성

속성 이름	설명
reflectionTopTexture	질감의 파일 이름
reflectionBottomTexture	질감의 파일 이름
reflectionLeftTexture	질감의 파일 이름
reflectionRightTexture	질감의 파일 이름
reflectionFrontTexture	질감의 파일 이름
reflectionBackTexture	질감의 파일 이름
reflectionFilterMode	질감의 필터링 모드입니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ point (=0): 질감이 전혀 부드럽지 않습니다. 질감이 최소화될 때 가장 저렴하지만 울퉁불퉁한 인공물이 쉽게 발생할 수 있습니다.</li> <li>▶ linear (=1): 이선형 필터링이라고도 합니다. 울퉁불퉁한 인공물을 줄이기 위해 최소화할 때 질감을 부드럽게 합니다.</li> </ul>

속성 이름	설명
	▶ trilinear (=2): 가장 비싸지만 선형 필터링보다 더 나은 결과물을 산출합니다.

**[참고]**



**반사 질감 위젯 피쳐**

EB GUIDE Studio는 다음과 같은 모든 속성에 대해 이미지 파일이 선택된 경우에만 **반사 질감** 위젯 피쳐를 표시합니다.

- ▶ reflectionTopTexture
- ▶ reflectionBottomTexture
- ▶ reflectionLeftTexture
- ▶ reflectionRightTexture
- ▶ reflectionFrontTexture
- ▶ reflectionBackTexture

이미지 파일은 모두 같은 크기여야 합니다.

**12.13.8.10. 거친 질감**

**거친 질감** 위젯 피쳐는 확장된 구성 값을 재료에 추가합니다. 질감은 PBR GGX 재료 위젯의 거칠기 매개변수를 제어합니다.

제한:

- ▶ **거칠기 질감** 위젯 피쳐는 PBR GGX 재료에 사용할 수 있습니다.
- ▶ **거친 질감**은 흑백 이미지입니다. RGB 색상 이미지의 경우 빨간색 채널만 사용됩니다.
- ▶ **거칠기 질감**이 활성화 상태인 경우 roughness 속성이 무시됩니다.

표 12.220. 거칠기 질감 위젯 피쳐의 속성

속성 이름	설명
roughnessTexture	질감의 파일 이름
roughnessMinFactor	질감 값을 보간하기 위한 플롯으로서의 최소 거칠기 매개변수
roughnessMaxFactor	질감 값을 보간하기 위한 플롯으로서의 최대 거칠기 매개변수
roughnessTextureAddressModeU	u 방향을 따른 질감의 주소 모드. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ repeat (=0): 질감 경계 외부에서 액세스할 때 질감이 반복됩니다. 랩 또는 타일이라고 합니다.</li> <li>▶ clamp (=1): 질감 경계 외부에서 액세스할 때 질감 가장자리의 픽셀이 사용됩니다.</li> </ul>

속성 이름	설명
roughnessTextureAddressModeV	v 방향을 따른 질감의 주소 모드입니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ repeat (=0): 질감 경계 외부에서 액세스할 때 질감이 반복됩니다. 랩 또는 타일이라고 합니다.</li> <li>▶ clamp (=1): 질감 경계 외부에서 액세스할 때 질감 가장자리의 픽셀이 사용됩니다.</li> </ul>
roughnessFilterMode	질감의 필터링 모드입니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ point (=0): 질감이 전혀 부드럽지 않습니다. 질감이 최소화될 때 가장 저렴하지만 울퉁불퉁한 인공물이 쉽게 발생할 수 있습니다.</li> <li>▶ linear (=1): 이선형 필터링이라고도 합니다. 울퉁불퉁한 인공물을 줄이기 위해 최소화할 때 질감을 부드럽게 합니다.</li> <li>▶ trilinear (=2): 가장 비싸지만 선형 필터링보다 더 나은 결과물을 산출합니다.</li> </ul>

### 12.13.8.11. 광택 질감

**광택 질감** 위젯 피처는 확장된 구성 값을 재료에 추가합니다. 질감은 질감 값을 스칼라 shininess 속성과 곱해 광택 강도를 조정합니다.

제한:

- ▶ **광택 질감** 위젯 피처는 재료 및 PBR Phong 재료에 대해 사용할 수 있습니다.
- ▶ **광택 질감**은 흑백 이미지입니다. RGB 색상 이미지의 경우 빨간색 채널만 사용됩니다.

표 12.221. 광택 질감 위젯 피처의 속성

속성 이름	설명
shininessTexture	질감의 파일 이름
shininessMinFactor	질감 값을 보간하기 위한 플롯으로서의 최소 광택 매개변수
shininessMaxFactor	질감 값을 보간하기 위한 플롯으로서의 최대 광택 매개변수
shininessTextureAddressModeU	u 방향을 따른 질감의 주소 모드. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ repeat (=0): 질감 경계 외부에서 액세스할 때 질감이 반복됩니다. 랩 또는 타일이라고 합니다.</li> <li>▶ clamp (=1): 질감 경계 외부에서 액세스할 때 질감 가장자리의 픽셀이 사용됩니다.</li> </ul>



속성 이름	설명
shininessTextureAddressModeV	v 방향을 따른 질감의 주소 모드입니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ repeat (=0): 질감 경계 외부에서 액세스할 때 질감이 반복됩니다. 랩 또는 타일이라고 합니다.</li> <li>▶ clamp (=1): 질감 경계 외부에서 액세스할 때 질감 가장자리의 픽셀이 사용됩니다.</li> </ul>
shininessFilterMode	질감의 필터링 모드입니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ point (=0): 질감이 전혀 부드럽지 않습니다. 질감이 최소화될 때 가장 저렴하지만 울퉁불퉁한 인공물이 쉽게 발생할 수 있습니다.</li> <li>▶ linear (=1): 이선형 필터링이라고도 합니다. 울퉁불퉁한 인공물을 줄이기 위해 최소화할 때 질감을 부드럽게 합니다.</li> <li>▶ trilinear (=2): 가장 비싸지만 선형 필터링보다 더 나은 결과물을 산출합니다.</li> </ul>

### 12.13.8.12. 거울 질감

거울 질감 위젯 피처는 확장된 구성 값을 재료에 추가합니다.

제한:

- ▶ 거울 질감 위젯 피처는 재료, PBR Phong 재료 및 PBR GGX 재료에 대해 사용할 수 있습니다.

표 12.222. 거울 질감 위젯 피처의 속성

속성 이름	설명
specularTexture	질감의 파일 이름
specularTextureAddressModeU	u 방향을 따른 질감의 주소 모드. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ repeat (=0): 질감 경계 외부에서 액세스할 때 질감이 반복됩니다. 랩 또는 타일이라고 합니다.</li> <li>▶ clamp (=1): 질감 경계 외부에서 액세스할 때 질감 가장자리의 픽셀이 사용됩니다.</li> </ul>
specularTextureAddressModeV	v 방향을 따른 질감의 주소 모드입니다. 사용 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>▶ repeat (=0): 질감 경계 외부에서 액세스할 때 질감이 반복됩니다. 랩 또는 타일이라고 합니다.</li> </ul>

속성 이름	설명
	<ul style="list-style-type: none"> <li>▶ clamp (=1): 질감 경계 외부에서 액세스할 때 질감 가장자리의 픽셀이 사용됩니다.</li> </ul>
specularFilterMode	<p>질감의 필터링 모드입니다. 사용 가능한 값은 다음과 같습니다.</p> <ul style="list-style-type: none"> <li>▶ point (=0): 질감이 전혀 부드럽지 않습니다. 질감이 최소화될 때 가장 저렴하지만 울퉁불퉁한 인공물이 쉽게 발생할 수 있습니다.</li> <li>▶ linear (=1): 이선형 필터링이라고도 합니다. 울퉁불퉁한 인공물을 줄이기 위해 최소화할 때 질감을 부드럽게 합니다.</li> <li>▶ trilinear (=2): 가장 비싸지만 선형 필터링보다 더 나은 결과물을 산출합니다.</li> </ul>

### 12.13.8.13. 톤 매핑

톤 매핑 위젯 피처는 톤 매핑, 즉 장면 그래프의 제한된 범위에 휘도 값을 매핑하는 기술을 활성화합니다.

제한:

- ▶ 톤 매핑 위젯 피처는 장면 그래프에 대해 사용할 수 있습니다.

톤 매핑 위젯 피처는 Erik Reinhard 등이 설명했던 전역 톤 매핑 연산자를 구현한다는 점에 유의하십시오.<sup>1</sup>

표 12.223. 톤 매핑 위젯 피처의 속성

속성 이름	설명
pureWhiteLuminance	순수 백색으로 매핑된 가장 작은 휘도 값입니다. 0보다 크거나 같은 값만 유효합니다.

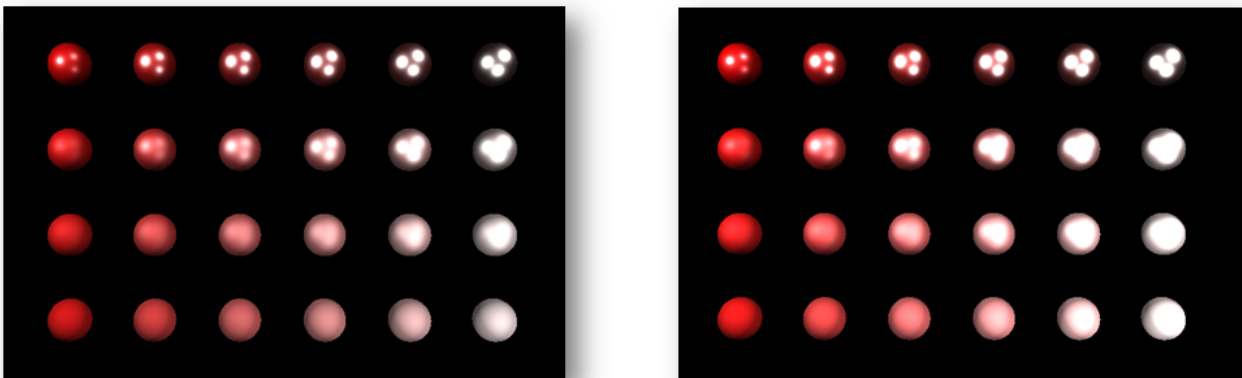


그림 12.3. 톤 매핑이 없는 이미지(왼쪽)와 톤 매핑이 있는 이미지(오른쪽)의 예

<sup>1</sup>디지털 이미지를 위한 사진 톤 재현 Reinhard, Erik 등. "Proceedings of the 29th annual conference on Computer graphics and interactive techniques" 2002년, 267~276페이지

## 12.13.9. 변환

범주 **변환**의 위젯 피처에서 위젯의 위치, 형식 및 크기를 수정합니다.

변환이 실행되는 순서는 위젯 트리에서의 순서와 같습니다. 여러 변환을 같은 위젯 트리 계층 수준에서 위젯 하나에 적용하는 경우의 순서는 다음과 같습니다.

1. 변환
2. 전단
3. 크기 조정
4. z축 기준 회전
5. y축 기준 회전
6. x축 기준 회전

### 12.13.9.1. 피벗

**피벗** 위젯 피처는 위젯에 적용되는 변환의 피벗 포인트를 정의합니다. 피벗 포인트가 구성되어 있지 않으면 (0, 0, 0.0)에 기본 피벗 포인트가 정의됩니다.

제한:

- ▶ **피벗** 위젯 피처를 추가하면 **회전**, **크기 조정** 및 **전단** 위젯 피처가 자동으로 추가됩니다.

표 12.224. **피벗** 위젯 피처의 속성

속성 이름	설명
pivotX	상위 위젯을 기준으로 한 x축의 피벗 포인트입니다.
pivotY	상위 위젯을 기준으로 한 y축의 피벗 포인트입니다.
pivotZ	상위 위젯을 기준으로 한 z축의 피벗 포인트입니다(위젯이 장면 그래프인 경우).

### 12.13.9.2. 회전

**회전** 위젯 피처는 위젯과 해당 하위 트리를 회전하는 데 사용됩니다.

표 12.225. **회전** 위젯 피처의 속성

속성 이름	설명
rotationEnabled	회전 사용 여부를 정의합니다.
rotationAngleX	x축의 회전 각도입니다. 장면 그래프에만 영향을 주는 속성입니다.

속성 이름	설명
rotationAngleY	y축의 회전 각도입니다. 장면 그래프에만 영향을 주는 속성입니다.
rotationAngleZ	z축의 회전 각도입니다.

### 12.13.9.3. 크기 조정

크기 조정 위젯 피처는 위젯과 해당 하위 트리의 크기를 조정하는 데 사용됩니다.

표 12.226. 크기 조정 위젯 피처의 속성

속성 이름	설명
scalingEnabled	크기 조정 사용 여부를 정의합니다.
scalingX	x축 크기 조정(비율)입니다.
scalingY	y축 크기 조정(비율)입니다.
scalingZ	z축 크기 조정입니다(위젯이 장면 그래프인 경우).

### 12.13.9.4. 전단

전단 위젯 피처는 위젯 하위 트리의 위젯을 비트는 데 사용됩니다.

표 12.227. 전단 위젯 피처의 속성

속성 이름	설명
shearingEnabled	전단 사용 여부를 정의합니다.
shearingXbyY	y축을 기준으로 하는 x축의 전단 정도입니다.
shearingXbyZ	z축을 기준으로 하는 x축의 전단 정도입니다(위젯이 장면 그래프인 경우).
shearingYbyX	x축을 기준으로 하는 y축의 전단 정도입니다.
shearingYbyZ	z축을 기준으로 하는 y축의 전단 정도입니다(위젯이 장면 그래프인 경우).
shearingZbyX	x축을 기준으로 하는 z축의 전단 정도입니다(위젯이 장면 그래프인 경우).
shearingZbyY	y축을 기준으로 하는 z축의 전단 정도입니다(위젯이 장면 그래프인 경우).

### 12.13.9.5. 변환

변환 위젯 피처는 위젯과 해당 하위 트리를 변환하는 데 사용됩니다. 이 기능은 x, y, z 방향으로 위젯을 이동합니다.

표 12.228. 변환 위젯 피처의 속성

속성 이름	설명
translationEnabled	변환 사용 여부를 정의합니다.
translationX	x축의 변환입니다.
translationY	y축의 변환입니다.
translationZ	z축의 변환입니다(위젯이 장면 그래프인 경우).

# 13. EB GUIDE Studio의 설치

## 13.1. 배경 정보

### 13.1.1. 제한

[참고]



**호환성**

EB GUIDE product line 6은 이전의 주 버전과 호환되지 않습니다.

[참고]



**EB GUIDE Speech Extension**

EB GUIDE Speech Extension 의 경우 구매 시에만 활성화되는 추가 기능 제품으로 사용이 허가됩니다.

[참고]



**사용자 권한**

Windows 7 또는 Windows 10 시스템에 EB GUIDE를 설치하려면 관리자 권한이 필요합니다.

### 13.1.2. 시스템 요구사항

다음 설정을 확인하십시오.

표 13.1.의 권장 설정 EB GUIDE Studio

하드웨어	CUP 속도 2GHz 이상/8GB RAM 이상의 쿼드 코어 CPU가 장착된 PC
운영 체제	Windows 7, Windows 10
화면 해상도	1920 x 1080픽셀의 2개 개별 모니터 사용
소프트웨어	Microsoft .NET Framework 4.5.1 DirectX 11

표 13.2. 의 권장 설정 EB GUIDE SDK

개발 환경(IDE)	Microsoft Visual Studio 2013 이상
파일 통합	CMake

## 13.2. EB GUIDE 다운로드 중

EB GUIDE의 커뮤니티 버전을 다운로드하려면 <https://www.elektrobit.com/ebguide/try-eb-guide/>로 이동하여 지침을 따르십시오.

EB GUIDE의 엔터프라이즈 버전을 다운로드하려면 EB Command로 이동하십시오.

### [참고]



#### 계정 활성화

제품을 주문하면 영업부에서 전자 메일을 전송합니다. 해당 전자 메일의 링크를 클릭하십시오. 그런 다음 전자 메일과 브라우저에서 지시하는 단계에 따라 계정을 만들고 로그인을 진행하십시오.

EB GUIDE product line 소프트웨어를 다운로드하는 서버는 EB Command입니다. EB Command에서 다운로드하는 방법에 대한 자세한 내용은 <https://www.elektrobit.com/support/downloading-from-eb-command/>를 참조하십시오.

## 13.3. EB GUIDE 설치하기



### EB GUIDE 설치하기

선수학습:

- 설치 파일 studio\_setup.exe를 다운로드해야 합니다.
- 운영 체제에 대한 관리자 권한이 있어야 합니다.

#### 단계 1

설치 파일 studio\_setup.exe를 더블 클릭합니다.

대화 상자가 열립니다.

#### 단계 2

예를 클릭합니다.

**설치 - EB GUIDE Studio** 대화 상자가 열립니다.

**단계 3**

라이선스 계약에 동의하고 **다음**을 클릭합니다.

**단계 4**

설치용 디렉터리를 선택합니다.

기본 설치 디렉터리는 C:/Program Files (x86)/Elektrobit/EB GUIDE <version>입니다.

**단계 5**

**다음**을 클릭합니다.

요약 대화 상자에 선택한 모든 설치 설정이 표시됩니다.

**단계 6**

표시된 설정을 사용하여 설치할 것임을 확인하려면 **설치**를 클릭합니다.

설치가 시작됩니다.

**단계 7**

설치를 종료하려면 **마침**을 클릭합니다.

EB GUIDE가 설치되었습니다.

**[작은 정보]****여러 버전 설치**

EB GUIDE 버전을 둘 이상 설치할 수 있습니다.

## 13.4. 제거 EB GUIDE



### 제거 EB GUIDE

**[참고]****영구적으로 EB GUIDE 제거하기**

이 지침에 따라 단계를 진행하면 PC에서 EB GUIDE가 영구적으로 제거됩니다.

선수학습:

- EB GUIDE 이(가) 설치되었습니다.
- 운영 체제에 대한 관리자 권한이 있어야 합니다.

**단계 1**

Windows **시작** 메뉴에서 **모든 프로그램**을 클릭합니다.





단계 2

**Elektrobit** 메뉴에서 제거할 버전을 클릭합니다.

단계 3

하위 메뉴에서 **제거**를 클릭합니다.

# 용어해설

## #

**3D 그래픽** 3D 그래픽은 3D 장면의 가상 그림입니다. 3D 장면은 3D 모델(메시 또는 도형), 재료, 광원, 카메라의 모음입니다. 재료는 색상과 질감 및 가상 조명 아래의 동작을 통해 3D 모델의 시각적 모양을 정의합니다. 카메라는 3D 장면이 찍힌 가상 그림이 있는 곳에서 보기 포인트를 제공합니다.

## A

**API** 응용 프로그래밍 인터페이스

## C

**통신 컨텍스트** 통신 컨텍스트는 통신이 수행되는 환경을 설명합니다. 고유한 숫자 ID로 각 통신 컨텍스트를 식별합니다.

## D

**데이터폴** 데이터폴은 런타임 중에 데이터폴 항목에 대한 액세스를 제공하는 EB GUIDE 모델의 데이터 캐시입니다. 응용 프로그램과 HMI 간의 데이터 교환에 사용됩니다.

**데이터폴 항목** 데이터폴 항목은 데이터를 저장하고 교환합니다. 데이터폴의 각 항목에는 통신 방향이 있습니다.

## E

**EB GUIDE GTF** EB GUIDE GTF는 EB GUIDE product line의 타깃 프레임워크이며 EB GUIDE TF의 요소입니다. EB GUIDE GTF는 대상 기기에서 EB GUIDE 모델을 실행하기 위한 런타임 환경을 나타냅니다.

**EB GUIDE GTF SDK** EB GUIDE GTF SDK는 EB GUIDE GTF에 포함된 개발 환경으로, EB GUIDE SDK의 하위 집합입니다. 또 다른 하위 집합으로는 EB GUIDE Studio SDK가 있습니다.

**EB GUIDE 모델** EB GUIDE 모델은 EB GUIDE Studio를 사용하여 만드는 HMI의 설명입니다.

**EB GUIDE product line** EB GUIDE product line는 HMI 모델을 지정하고 임베디드 환경 시스템에서 실행되는 그래픽 사용자 인터페이스로 변환하는 데 필요한 소프트웨어 라이브러리 및 도구 모음입니다.

EB GUIDE 스크립트	EB GUIDE 스크립트는 EB GUIDE product line의 스크립팅 언어입니다. EB GUIDE 스크립트를 사용하면 데이터폴, 위젯/상태 머신 등의 모델 요소 및 시스템 이벤트에 액세스할 수 있습니다.
EB GUIDE SDK	EB GUIDE SDK는 EB GUIDE의 제품 구성 요소이며 EB GUIDE product line용 소프트웨어 개발 키트입니다. 여기에는 EB GUIDE Studio SDK 및 EB GUIDE GTF SDK가 포함되어 있습니다.
EB GUIDE Studio	EB GUIDE Studio는 그래픽 사용자 인터페이스를 사용하여 HMI를 지정하고 모델링하기 위한 도구입니다.
EB GUIDE Studio SDK	EB GUIDE Studio SDK는 EB GUIDE Studio와 통신하기 위한 API(응용 프로그래밍 인터페이스)로, EB GUIDE SDK의 하위 집합입니다. 또 다른 하위 집합으로는 EB GUIDE GTF SDK가 있습니다.
EB GUIDE TF	EB GUIDE TF는 EB GUIDE product line의 런타임 환경으로, EB GUIDE GTF 및 EB GUIDE STF로 구성되며 EB GUIDE 모델을 실행하는 데 필요합니다.

## G

GL	그래픽 라이브러리
GUI	그래픽 사용자 인터페이스

## H

HMI	휴먼 기계 인터페이스
-----	-------------

## I

IBL	이미지 기반 조명
IBLGenerator	IBLGenerator는 환경 조명 정보를 처리하기 위한 도구입니다.

## L

라이브러리	라이브러리는 EB GUIDE Studio에 사용되는 리소스 집합입니다. EB GUIDE 프로젝트에 필요한 라이브러리는 프로젝트 센터에서 정의됩니다.
-------	--

## M

모델 요소	모델 요소는 상태, 위젯, 데이터폴 항목 등 EB GUIDE 모델 내의 개체입니다. 다른 살펴볼 내용 EB GUIDE 모델.
-------	---

## O

OS 운영 체제

## P

PBR 물리적 기반 렌더링

**프로필** 프로젝트 센터에서 프로파일은 사양 집합입니다. 프로파일에서는 프로젝트에 대한 라이브러리, 메시지, 장면을 정의합니다. EB GUIDE 모델을 내보내는 중에 프로파일의 데이터가 model.json 구성 파일에 기록됩니다.

**프로젝트 센터** 프로젝트 센터에는 프로파일, 언어 등 모든 프로젝트 관련 기능이 포함되어 있습니다.

**프로젝트 편집자** 프로젝트 편집자에서는 휴먼 기계 인터페이스의 모양과 동작을 모델링합니다.

## R

**리소스** 리소스는 EB GUIDE 프로젝트의 일부분인 데이터 패키지입니다. 리소스의 예로는 글꼴, 이미지, 메시 등이 있습니다. 운영 체제에 따라 리소스는 파일 등 EB GUIDE 모델 외부에 저장됩니다.

## S

**공유 라이브러리** 공유 라이브러리는 정적 라이브러리와 달리, 실행 프로그램을 준비할 때 로드할 수 있습니다. Windows 플랫폼에서 공유 라이브러리는 동적 링크 라이브러리라고 부르며 .dll 파일 확장명을 포함합니다. Unix 시스템에서 공유 라이브러리는 공유 객체라고 부르며 .so 파일 확장명을 포함합니다.

**상태** 상태는 상태 기계의 상태를 정의합니다. 상태와 상태 전이는 상태 차트에서 모델링됩니다.

**상태 머신** 상태 기계는 상태, 해당 상태 간의 전이 및 작업 집합입니다. 상태 기계는 시스템의 동적 동작을 설명합니다.

## T

**전이** 전이는 상태 간의 변경을 정의하며, 대개 이벤트에 의해 트리거됩니다.

## U

UI 사용자 인터페이스

## V

보기 보기는 프로젝트별 HMI 화면의 그래픽 표현이며, 특정 상태 기계 상태와 관련되어 있습니다. 보기는 위젯 트리로 구성됩니다.

## W

위젯 위젯은 기본 그래픽 요소입니다. 위젯은 그래픽 사용자 인터페이스와의 상호작용을 위해 사용됩니다.

# 인덱스

## Symbols

- .psd 파일 형식, 138
- 3D 객체, 35
- 3D 그래픽, 35, 62, 64, 225, 362
  - 가져오기, 225
  - 메시, 64
  - 이미지 기반 조명, 62
  - 지원되는 형식, 35, 62, 64
  - 추가, 137
- 3D 위젯, 35, 62, 64, 106
  - 참조, 311
- 가져오기
  - 언어별로 다른 텍스트, 181
- 감마 조정
  - 렌더링, 57
- 거울 질감
  - 참조, 353
- 거친 질감
  - 참조, 351
- 경로 제스처, 209
  - 참조, 327, 328
- 공동 커브
  - 참조, 300, 309
- 공유 라이브러리, 364
- 관점 전환 애니메이션, 156
- 광택 질감
  - 참조, 352
- 구성
  - 디스플레이, 179
- 구성 요소
  - 도킹, 49
  - 도킹 해제, 49
- 구성 파일, 277, 287, 293
- 글꼴, 62
  - 데이터 유형, 239
  - 비트맵 글꼴, 62, 62
  - 오픈타입 글꼴, 62
  - 트루타입 글꼴, 62
- 금속 질감
  - 참조, 347
- 긋기 제스처
  - 참조, 325
- 기록기 응용 프로그램, 40
- 기본 위젯, 105
  - 참조, 296
- 길게 누르기 제스처
  - 참조, 326
- 내보내기, 174
  - 언어별로 다른 텍스트, 180
- 내부 전이, 126
- 내비게이션 구성 요소
  - 프로젝트 편집자, 43
- 내비게이션 영역
  - 프로젝트 센터, 41
- 누르기 제스처
  - 참조, 326
- 누름
  - 참조, 319
- 눈금 레이아웃
  - 참조, 339
- 느린 시작 커브
  - 참조, 299, 308
- 데이터 유형
  - ib1, 240
  - 글꼴, 239
  - 메시, 241
  - 메시 목록, 241
  - 목록, 242
  - 문자열, 241
  - 부동, 239
  - 부울, 237
  - 부울 목록, 237
  - 색, 238
  - 이미지, 240
  - 정수, 240
  - 조건부 스크립트, 238
- 데이터폴, 52, 362
- 데이터폴 항목, 52, 165, 362
  - 가져오기, 181
  - 내보내기, 180
  - 링크, 167

- 목록, 165
- 변경, 184
- 복사, 165
- 붙여 넣기, 165
- 언어 지원, 222
- 참조, 237
- 창 모드 목록, 53
- 추가, 164
- 도구 상자
  - 프로젝트 편집자, 44
- 도구 상자 구성 요소
  - 프로젝트 편집자, 44
- 도킹
  - 구성 요소, 49
- 도킹 해제
  - 구성 요소, 49
- 독자 응용 프로그램, 40
- 동적 상태 머신
  - 추가, 113, 194
- 디스플레이
  - 구성, 179
- 라이브러리, 363
  - 추가, 177
- 레이블, 131
  - 글꼴, 131, 132
  - 참조, 305
  - 추가, 131
- 레이아웃 여백
  - 참조, 340
- 렌더러
  - 구성, 179
- 리소스, 364
  - .psd 파일 형식, 61
  - 3D 그래픽, 62, 64
  - 글꼴, 62
  - 메시, 64
  - 이미지, 63
  - 이미지 기반 조명, 62
- 리소스 관리, 61
- 링크
  - 데이터풀 항목, 168
  - 위젯 속성, 142, 144
- 멀티 터치 입력, 103
- 멀티샘플링, 292
- 메시, 64
  - 데이터 유형, 241
  - 참조, 313
- 메시 목록
  - 데이터 유형, 241
- 명령 영역
  - 프로젝트 편집자, 48
- 명령줄, 79, 174, 175, 182
- 모델 요소, 54, 363
  - 삭제, 120
- 모으기 제스처
  - 참조, 329
- 목록, 165
  - 데이터 유형, 242
  - 만들기, 212
- 목록 레이아웃
  - 참조, 340
- 목록 인덱스
  - 참조, 342
- 문자열
  - 데이터 유형, 241
- 문제 구성 요소, 172
  - 프로젝트 편집자, 49
- 바로가기
  - 사용자 인터페이스, 79
- 밖으로 이동
  - 참조, 332
- 반사 질감
  - 참조, 350
- 방사 질감
  - 참조, 346
- 방향 조명
  - 참조, 312
- 버튼
  - 사용자 인터페이스, 79
- 변경 애니메이션, 156
  - 참조, 295
- 변경 애니메이션 추가, 157
- 변환
  - 참조, 356

- 보기, 104, 365
  - 참조, 295
  - 추가, 127
- 보기 상태
  - 참조, 295
- 보기 템플릿
  - 참조, 295, 295
- 복사
  - 데이터폴 항목, 165
  - 이벤트, 162
- 복합 상태, 116
- 부동
  - 데이터 유형, 239
- 부울
  - 데이터 유형, 237
- 부울 목록
  - 데이터 유형, 238
- 불투명 질감
  - 참조, 349
- 붙여 넣기
  - 데이터폴 항목, 165
  - 이벤트, 162
- 뷰포트
  - 참조, 343
- 빠른 시작 커브
  - 참조, 299, 307
- 사각형
  - 참조, 305
- 사용자 정의 속성, 145
- 사용자 정의 포커스
  - 참조, 324
- 상수 커브
  - 참조, 299, 307
- 상자 레이아웃
  - 참조, 338
- 상태, 85, 115, 116, 198, 364
  - 보기 상태, 86
  - 복합 상태, 85
  - 선택 상태, 88
  - 이력 상태, 89
  - 전이, 120
  - 종료 작업, 119
  - 진입 작업, 118
  - 초기 상태, 86
  - 최종 상태, 87
- 상태 기계
  - UML 2.5 표기법, 100
  - UML과 비교, 100
  - 상태 기계 실행, 96
  - 상태 기계 포함, 101
  - 전이, 92
- 상태 머신, 84, 364
  - 논리적 상태 머신, 84
  - 동적 상태 머신, 84
  - 삭제, 115
  - 상태, 85
  - 상태 머신 포함, 84
  - 추가, 113
  - 해틱 상태 머신, 84
- 색
  - 데이터 유형, 238
- 선택 그룹
  - 참조, 320
- 선택 상태, 117
- 선택됨
  - 참조, 319
- 선형 보간 정수, 219
- 선형 보간 커브, 302, 310
- 선형 커브, 302, 310
- 속성 구성 요소
  - 명령 영역, 45
  - 프로젝트 편집자, 45
- 손가락 ID, 103
- 스크립트 커브, 301, 310
- 스크립팅된 값, 77, 165
- 스킨
  - 삭제, 153
  - 전환, 153
  - 지원, 83
  - 추가, 153
- 스트로크
  - 참조, 323
- 스팟 조명
  - 참조, 317



- 스피닝
  - 참조, 320
- 시뮬레이션, 174
- 아이콘
  - 사용자 인터페이스, 79
- 안으로 이동
  - 참조, 332
- 알파 마스크, 136
  - 참조, 297
- 애니메이션, 38, 106, 135, 156, 219
  - 변경 애니메이션, 156
  - 애니메이션 변경, 38
  - 종료 애니메이션, 38, 156
  - 진입 애니메이션, 38, 156
  - 참조, 298, 306
- 애니메이션 재배열, 158
- 애니메이션 팝업 끄기
  - 참조, 295
- 애니메이션 팝업 켜기
  - 참조, 295
- 언어
  - 변경, 222
- 언어별로 다른 텍스트, 222
  - 가져오기, 181
  - 내보내기, 180
- 여러 줄
  - 참조, 318
- 위로 이동
  - 참조, 333
- 위젯, 104, 365
  - 3D 위젯, 106
  - 그룹, 132
  - 기본, 106
  - 삭제, 139
  - 애니메이션, 106
  - 위치, 140
  - 추가, 128
  - 크기 조정, 141
- 위젯 속성, 107
  - EB GUIDE 스크립트, 73
  - 기본 속성, 107
  - 데이터폴 항목에 대한 링크, 144
  - 사용자 정의 속성, 107, 145
  - 위젯 속성에 대한 링크, 142
  - 위젯 템플릿, 108
  - 위젯 피쳐 속성, 107
  - 추가, 145
  - 위젯 템플릿, 108, 158, 161
  - 위젯 템플릿 인터페이스, 108
  - 위젯 피쳐, 107, 107, 109
    - 경로 제스처, 209
    - 제거, 150
    - 추가, 148
  - 응용 프로그래밍 인터페이스, 39
- 이동 가능
  - 참조, 333
- 이미지
  - 나인패치, 63
  - 데이터 유형, 240
  - 지원되는 형식, 63
  - 참조, 304
  - 추가, 129
- 이미지 기반 조명, 62, 233
  - 렌더링, 59
  - 참조, 312
- 이벤트, 54, 75
  - 복사, 162
  - 붙여 넣기, 162
  - 이벤트 ID, 54
  - 이벤트 그룹, 54
  - 참조, 275
  - 추가, 162
  - 파일, 183
- 이벤트 시스템, 54
- 인스턴스화기, 212
  - 줄 템플릿, 133, 304
  - 참조, 304
  - 추가, 133
- 일반 지도 질감
  - 참조, 348
- 자동 숨김, 49
- 자동 포커스
  - 참조, 324
- 작업

- 전이, 124
- 종료 작업, 114
- 진입 작업, 114
- 장면 구성
  - 참조, 290
- 장면 그래프, 35, 62, 64, 137, 225
  - 질감, 225
  - 참조, 316
  - 추가, 137
- 장면 그래프 노드
  - 참조, 316
- 재료
  - PBR GGX 재료, 313, 364
  - PBR Phong 재료, 314, 364
  - 참조, 313, 313, 314
- 전단
  - 참조, 356
- 전역 이름 바꾸기, 171
- 전이, 92, 120, 364
  - 내부, 126
  - 이동, 121
  - 작업, 124
  - 조건, 123
  - 추가, 120
  - 트리거, 122
- 절대 레이아웃
  - 참조, 337
- 정방향 커브
  - 참조, 300, 308
- 정수
  - 데이터 유형, 241
- 제스처, 102
  - 경로 제스처, 102
  - 비경로 제스처, 102
  - 참조, 325, 330
- 제스처 ID
  - 참조, 328
- 조건
  - 전이, 123
- 조건부 스크립트
  - 데이터 유형, 238
- 조명 지도 질감
  - 참조, 346
- 종료 애니메이션, 156
  - 참조, 295
- 종료 작업, 119
  - 상태 머신, 114
- 주변 조명
  - 참조, 311
- 주변 질감
  - 참조, 344
- 줄 인덱스
  - 참조, 342
- 진입 애니메이션, 156
  - 참조, 295
- 진입 애니메이션 추가, 156
- 진입 작업, 118
  - 상태 머신, 114
- 창 모드 목록
  - 데이터폴 항목, 53
- 천연색
  - 참조, 323
- 카메라
  - 참조, 312
- 카메라 뷰포트
  - 참조, 343
- 컨테이너
  - 참조, 303
  - 추가, 132
- 콘솔 (살펴볼 내용 명령줄)
- 콘텐츠 영역
  - 프로젝트 센터, 41
  - 프로젝트 편집자, 45
- 크기 조정
  - 참조, 356
- 크기 조정 모드
  - 참조, 341
- 키 놓음
  - 참조, 331
- 키 누름
  - 참조, 330
- 키 상태 변경됨
  - 참조, 331
- 키 유니코드

- 참조, 332
  - 타원
    - 참조, 303
  - 터치 놓음
    - 참조, 336
  - 터치 누름
    - 참조, 335
  - 터치 상태 변경됨
    - 참조, 336
  - 터치 이동
    - 참조, 335
  - 터치 입력 (살펴볼 내용 제스처)
  - 터치 잡기 놓침
    - 참조, 334
  - 터치 제스처 (살펴볼 내용 제스처)
  - 터치함
    - 참조, 321
  - 테두리
    - 참조, 322
  - 텍스트 잘림
    - 참조, 321
  - 템플릿
    - 만들기, 158
    - 사용, 160
    - 삭제, 161
  - 템플릿 인덱스
    - 참조, 343
  - 템플릿 인터페이스, 159
    - 속성 제거, 159
    - 속성 추가, 159
  - 톤 매핑
    - 참조, 354
  - 통신 컨텍스트, 40, 166, 362
  - 트리거
    - 전이, 122
  - 포인트 조명
    - 참조, 315
  - 포커스됨
    - 참조, 318
  - 프로젝트 센터, 40, 364
    - 내비게이션 영역, 41
    - 콘텐츠 영역, 41
  - 프로젝트 편집자, 42, 364
    - VTA 구성 요소, 49
    - 내비게이션 구성 요소, 43
    - 도구 상자, 44
    - 도구 상자 구성 요소, 44
    - 명령 영역, 48
    - 문제 구성 요소, 49
    - 콘텐츠 영역, 45
  - 프로필, 176, 364
    - 복제, 176
  - 피벗
    - 참조, 355
  - 하위 가시성 선택
    - 참조, 317
  - 확산 질감
    - 참조, 345
  - 활성화됨
    - 참조, 318
  - 회전
    - 참조, 355
  - 회전 제스처
    - 참조, 329
  - 회전식
    - 참조, 334
  - 효과
    - 위젯 피쳐, 322
  - 흐름 레이아웃
    - 참조, 338
- A**
- API, 362 (살펴볼 내용 응용 프로그래밍 인터페이스)
- E**
- EB GUIDE GTF, 362
  - EB GUIDE GTF SDK, 362
  - EB GUIDE Monitor, 50, 174, 183, 183, 184, 185, 189, 190, 191
    - monitor.cfg, 190
    - 감시 목록, 191
    - 감시 목록 가져오기, 191
    - 감시 목록 내보내기, 191
    - 구성, 190

- 데이터폴 구성 요소, 184
- 데이터폴 항목, 184
- 독립 실행형, 189
- 명령줄, 189
- 스크립팅 구성 요소, 185
- 이벤트, 183
- 이벤트 구성 요소, 183
- 탭, 50
- EB GUIDE product line, 362
- EB GUIDE SDK, 362
- EB GUIDE Studio, 362
- EB GUIDE Studio SDK,, 362
- EB GUIDE TF, 362
- EB GUIDE 모델, 53, 362
  - 모델 요소, 54
- EB GUIDE 스크립트, 64, 165, 362
  - if-then-else, 70
  - l 값, 69
  - r 값, 69
  - while 루프, 70
  - 네임스페이스, 65
  - 데이터폴 액세스, 72
  - 로컬 변수, 69
  - 목록, 74
  - 문자열 서식 지정, 77
  - 스크립팅된 값, 78
  - 식, 67
  - 식별자, 65
  - 외래 함수 호출, 72
  - 위젯 속성, 73
  - 유형, 66
  - 이벤트, 75
  - 자습서, 202
  - 주석, 66
  - 표준 라이브러리, 77
- EB GUIDE 프로젝트, 53
- EB GUIDE 확장, 57

## G

- GL, 363
- GUI, 363

## H

- HMI, 363

## I

- ibl
  - 데이터 유형, 240
- IBL, 363
- IBLGenerator, 363
  - 이미지 기반 조명, 59

## M

- model.json, 277

## O

- OS, 364

## P

- platform.json, 287

## T

- todo
  - EB GUIDE 스크립트, 66
- tracing.json, 293

## U

- UI, 364

## V

- VTA, 156
- VTA 구성 요소
  - 프로젝트 편집자, 49