



TM611 Mapp View 시작하기

I 버전 정보

버전	날짜	수정내역	번역	검수
1.0	2017.2.21	첫번째 버전 TM611TRE.425-ENG_Working with mapp View_V4250		임은
2.0	2017.12.12	TM611TRE.433-ENG_Working with mapp View_V5.0 4.2.4 항목 남겨둠.		임은

Table 1: Versions

선행 및 필요 조건

교육 자료	TM210 – Automation Studio 시작하기		
소프트웨어	Automation Studio 4.3.3		
	Automation Runtime 4.33		
	Technology package – mapp View 5.0		
하드웨어	ARsim X20 CPU 와 ETAL611.1T10-1 https://www.br-automation.com/en/academy/eta-system/		

목차

1 소개	6
1.1 학습 목표	7
1.2 안전 표시와 부호	7
2 mapp View 개념	8
2.1 설치: mapp View Technology Package	9
2.2 라이선싱	9
2.3 HMI 어플리케이션의 구조	10
3 페이지 작성과 내비게이션(navigation)	11
3.1 개요 – 페이지 작성	11
3.1.1 레이아웃(Layout)	12
3.1.2 영역(Area)	12
3.1.3 콘텐츠(Content)	13
3.1.4 페이지(Page)	13
3.2 단계별 페이지 작성	14
3.2.1 mapViewGettingStarted 프로젝트	15
3.2.2 로지컬 뷰에 패키지 추가하기	15
3.2.3 레이아웃(layout) 작성	15
3.2.4 영역(area) 작성	16
3.2.5 콘텐츠(Content) 작성	17
3.2.6 페이지(Pages) 생성	18
3.2.7 컨피규레이션 뷰(Configuration View)에 파일 추가하기	20
3.2.8 예제: 브라우저에서 HMI 어플리케이션 열기	22
3.3 내비게이션 개요	23
3.3.1 수동 내비게이션	23
3.3.2 자동 내비게이션	24
3.4 수동 내비게이션 이용하기	25
3.4.1 ServicePage 및 AlarmPage 작성하기	25
3.4.2 NavigationBar 위젯	26
3.4.3 NavigationButton 위젯	26
3.4.4 .vis file 파일에 ServicePage 및 AlarmPage 입력하기	27
3.4.5 수정사항 전송하기	28
3.5 자동 내비게이션 이용하기	28
3.5.1 컨피규레이션 뷰(Configuration View) 에 내비게이션 오브젝트	29
3.5.2 내비게이션 위젯을 추가하고 설정하기	30
3.5.3 .vis 파일에서 내비게이션 참조하기	30
4 시각적 외관 – 스타일링(Styling)	32
4.1 스타일링 개요	32
4.1.1 스타일 가능한 속성(Styleable property)	32
4.1.2 스타일(Style)	32
4.1.3 테마(Theme)	32
4.2 스타일(styles) 이용하기	33
4.2.1 스타일링 가능한 속성(Styleable property)	33
4.2.2 B&R 테마와 스타일	34
4.2.3 디폴트 스타일 변경하기	36
4.2.4 (선택)스타일을 갖는 테마 작성하기	38

5 데이터 바인딩(Data binding)	42
5.1 데이터 바인딩(Data binding) 개요	42
5.1.1 OPC UA	42
5.1.2 바인딩(Binding)	43
5.2 데이터 바인딩 이용하기	44
5.2.1 HMI 어플리케이션에서 데이터 표시하기	44
5.2.2 설정 값(Setpoint) 입력	48
5.2.3 값 범위 규정하기	50
6 미디어 파일(Media files)	52
6.1 개요 – 미디어 파일	52
6.2 미디어 파일 추가하기	52
6.2.1 이미지 삽입하기	52
6.2.2 내비게이션 버튼(NavigationButton) 위젯의 이미지	54
7 사용자 역할 시스템(User role system)	57
7.1 개요 – 사용자 역할 시스템	57
7.2 사용자 역할 시스템 이용하기	58
7.2.1 OPC UA 노드에 쓰기 접근 제한하기	58
7.2.2 값 범위 제한하기	62
7.2.3 역할을 변경하고 추가하기	63
8 현지화(Localization)	64
8.1 개요 – 텍스트 시스템(Text system)	64
8.1.1 식별자(Identifier)	64
8.1.2 프로젝트의 언어 설정하기	65
8.1.3 타겟 시스템의 언어 설정	65
8.1.4 mapp View 에서 텍스트 시스템(Text system)	66
8.2 텍스트 시스템 이용하기	67
8.2.1 수동 내비게이션의 텍스트	67
8.2.2 자동 내비게이션의 텍스트	71
8.2.3 로그아웃 버튼(LogoutButton)과 라벨 로그인 정보(LabelLoginInfo) 위젯의 텍스트	72
8.3 개요 – 단위 시스템(Unit system)	73
8.4 단위 시스템 이용하기	74
8.4.1 값과 단위를 함께 출력하기와 측정 시스템 전환하기	74
9 이벤트(events)와 액션(actions)	78
9.1 개요 – 이벤트(Event)	78
9.2 개요 – 액션(Action)	79
9.2.1 OPC UA 액션(OPC UA actions)	79
9.2.2 위젯 액션(Widget actions)	79
9.3 이벤트와 액션 이용하기	80
9.3.1 값 설정하기	80
9.3.2 위젯 표시와 감추기	83
10 운영 시스템에 적용하는 방법	86
10.1 제어기에 mapp View 서버 운영하기	86
10.2 T50 에서 mapp View 클라이언트 운영하기	87
10.2.1 Automation Studio 에서 구성	87
10.2.2 터미널로 구성 전송	88
10.2.3 Context caching 으로 성능 향상	90

11 요약	91
Automation Academy 에서 제공하는 것	92

1 소개

태블릿, 스마트폰 등과 같은 스마트 장치는 궁극적 사용성을 갖춘 강력한 기술의 완벽한 예시로 간주된다. 당연히 산업 기계와 시스템 작업자들과 그러한 기계 제조사들은 그들이 매일 사용하는 기계와 상호작용하는 것을 원한다.

B&R 은 mapp View 을 사용하여 B&R 자동화 어플리케이션의 HMI 시스템을 개발할 수 있는 웹 기술(web technology)을 제공한다. 어플리케이션 엔지니어들은 강력하고 직관적인 HMI 솔루션을 작성하기 위해 mapp View 를 이용할 수 있다. 여기에서 이용되는 웹 기술은 mapp View 로 요약된다. 광범위한 기술 분야를 학습할 필요는 없다. HMI 개발자는 전적으로 주어진 작업에 대한 해결책을 작성하는데 집중할 수 있다.



그림 1 mapp View 화면작화 오브젝트

mapp View 는 B&R 의 Automation Studio 엔지니어링 환경에 완전히 통합되어 있다. mapp View 는 웹 프로그래밍 언어 지식을 요구하지 않는 세계 최초의 웹 기반 HMI 솔루션이다.

1.1 학습 목표

이 교육 자료는 여러분이 요구하는 기능을 Automation Studio 에서 어떻게 구성하고 이용할 수 있는지 학습에 도움이 되는 전형적인 HMI 작업을 예를 들어 설명한다.

- mapp View 의 기본적 개념을 학습하고 이러한 개념의 속성을 보다 잘 이해하게 될 것이다.
- 새로운 mapp View Visualization object 을 Automation Studio 프로젝트에 추가하고 그것을 설정할 수 있을 것이다.
- mapp View HMI 어플리케이션을 위한 콘텐츠(contents)를 개발하고 그 콘텐츠를 HMI 페이지에 추가하기 위한 레이아웃 기능을 이용할 수 있을 것이다.
- mapp View HMI 어플리케이션 콘텐츠를 관리하기 위해 mapp View HMI 위젯을 이용할 수 있을 것이다.
- mapp View HMI 어플리케이션의 다양한 페이지 사이에 내비게이션¹을 설정하는 여러 가지 방법을 익힐 것이다.
- 자동화 어플리케이션의 프로세스 데이터를 mapp View HMI 어플리케이션에 포함시킬 수 있을 것이다.
- mapp View HMI 어플리케이션 콘텐츠의 시각적 디자인을 조직화를 위해 이미지를 이용할 수 있을 것이다.
- 로그인한 사용자에 따라 mapp View HMI 어플리케이션의 속성 설정을 위해, 어떻게 사용자 역할 시스템을 이용하는지 배울 것이다.
- 텍스트 시스템이 작동하는 방법을 배우고 언어 전환을 설정을 배울 것이다.
- 유닛 시스템(unit system)에 친숙해지고 단위 변환 설정 방법을 배울 것이다.
- 이벤트가 발생할 때 특정한 액션을 설정하는 방법을 배울 것이다.

1.2 안전 표시와 부호

별도 명시가 없는 한 “TM210 – Automation Studio 시작하기”에 열거된 안전 표시와 부호가 적용된다.

¹ (역주) 페이지 전환하는 기능을 “내비게이션”이라고 칭한다.

2 mapp View 개념

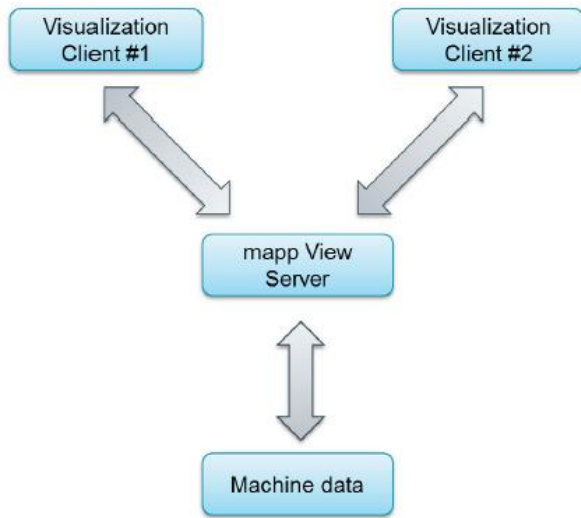


그림 2 mapp View 구조

mapp View 는 HMI 어플리케이션을 위해 통합된 시스템이다. 기존 자동화 소프트웨어에 통합된 시스템은 기계를 위한 HMI 인터페이스를 설계하는데 이용된다.

mapp View 는 모듈형 구조를 갖는다. 기계와 기기의 기술적 프로세스 감시와 운전을 허용한다. mapp View 구조는 한 개 이상의 HMI 클라이언트에 연결된 HMI 서버로 구성되는 클라이언트-서버 시스템으로써 고안되었다. HMI 서버는 (기계나 시스템 로직 내의) 자동화 어플리케이션으로부터 분리되어 있다.

mapp View 의 모듈 구조는 콘텐츠와 레이아웃을 별도로 편집할 수 있다는 사실이 강조된다. 이는 HMI 요소의 재사용성을 높이고 개발 시간을 단축한다.

mapp View 는 멀티-클라이언트 및 멀티-유저 시스템으로 개발되었다. 다양한 사용자들이 각각 다른 시각화 클라이언트로부터 맞춤형된 콘텐츠를 볼 수 있다. mapp View 는 여러분이 자동화 어플리케이션에 프로그래밍을 할 필요 없이 역할-기반 콘텐츠를 볼 수 있도록 한다. 맞춤형된 HMI 콘텐츠는 상호 독립적인 각각 다른 HMI 클라이언트에서 동시에 표시될 수 있다.

mapp View 는 웹 기술을 기반으로 하지만, mapp View HMI 어플리케이션 개발자는 직접 이 기술을 다뤄야 할 필요가 없다. mapp View 는 웹 기술의 복잡성을 단순화하고 HMI 개발자들에게 인간-기계 인터페이스 디자인에 집중할 수 있는 자유를 제공해준다. HMI 요소들은 (위젯, widget) 용이하게 HMI 콘텐츠를 디자인할 수 있도록 드래그 앤 드롭(drag-and-drop)을 통해 이용된다.



그림 3 mapp View - 재사용 가능한 요소들을 통한 용이한 설정

2.1 설치: mapp View Technology Package

“mapp View Technology Package” 설치에 요구되는 최소 버전은 Automation Studio 4.3.3 이다. mapp View Technology Package 는 B&R 웹사이트에서 다운로드 받거나 업그레이드 대화 상자(Upgrade)를 이용하여 설치할 수 있다.

상단 메뉴 Tools > Upgrades...

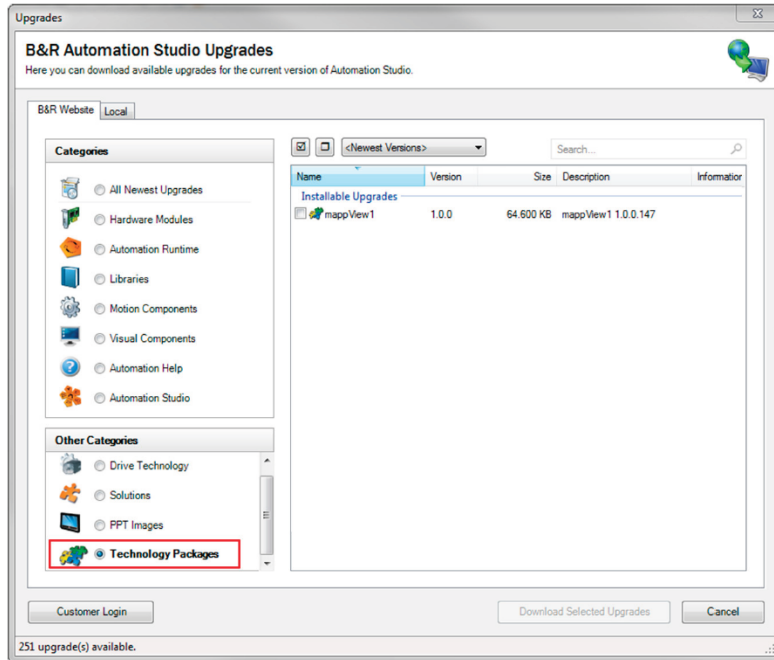


그림 4 Automation Studio 업그레이드 대화 상자 – Technology Package

Technology Package 가 설치되는 동안, mapp View 관련 도움말(Automation studio help)도 함께 설치된다.

2.2 라이선싱

mapp View HMI 어플리케이션은 ARsim 에서 라이선스나 제약 없이 이용할 수 있다.

ARemb 및 ARwin 에서 mapp View 를 운영하려면, 기본형 라이선스가 (모델 번호: 1TGMPVIEW.00-01) 요구된다. 기본형 라이선스는 mapp View 를 하나 이용하고 하나의 클라이언트가 연결할 수 있도록 한다. 다수의 클라이언트를 mapp View 서버에 연결할 때는 추가적인 클라이언트 라이선스가 (모델 번호: 1TGMPCLIENT.10-01) 요구된다.

필요한 클라이언트 라이선스 수량은 (실제로 연결되는 클라이언트의 수가 아니라) mapp View 설정에서 허용하는 클라이언트 총 수에 의해 결정된다.



Technology Guard 가 설치되어 있지 않으면, 로거(logger entry)에 기록된다.

“No valid license for mapp View found. The mapp View server configuration allows more connections(10) than the number of available client license(4)!”

(번역) mapp View 에 대한 유효한 라이선스가 발견되지 않음. mapp View 서버 설정에서 이용 가능한 클라이언트 라이선스의 수 (4) 보다 많은 연결을 (10) 허용되었다!

2.3 HMI 어플리케이션의 구조

mapp View HMI 어플리케이션 요소는 자동화 어플리케이션과 마찬가지로 Automation Studio 에서 설정되고 관리된다.

HMI 페이지, 텍스트 및 이미지 파일은 로지컬 뷰(Logical View)에서 관리된다. 로지컬 뷰에 있는 한 개 이상의 HMI 어플리케이션은 컨피규레이션 뷰(Configuration View)에 활성화 된 설정(active configuration)에 추가할 수 있다. 이 방법으로 HMI 요소에 대한 소스 파일과 기계 설정에 필요한 HMI 사이가 명확히 구분된다.

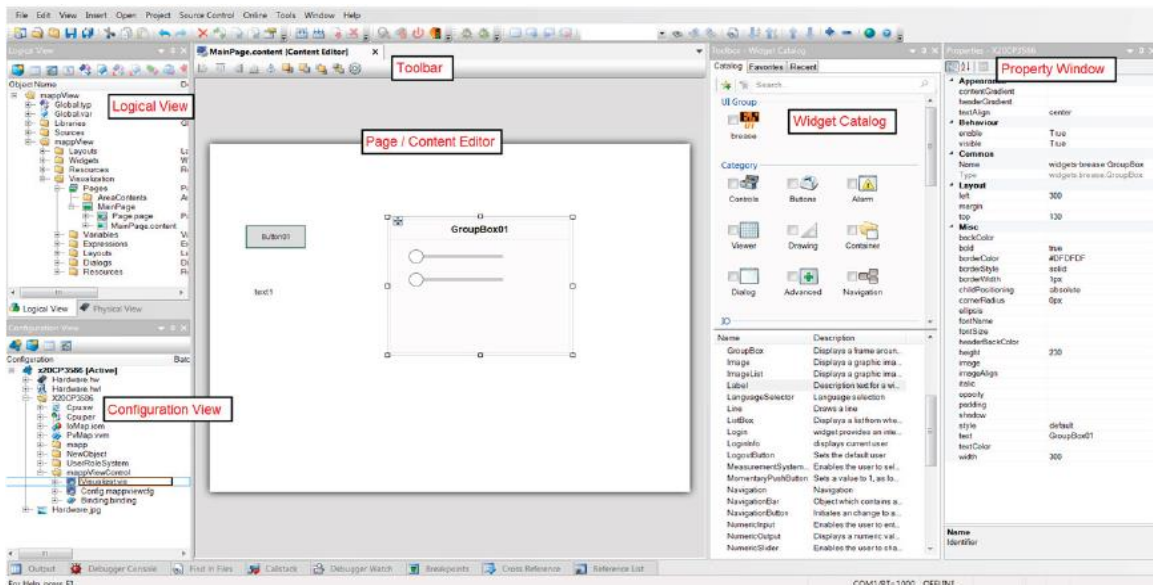


그림 5 Automation Studio 에서 HMI 어플리케이션 구성

로지컬 뷰(Logical View)

로지컬 뷰에서, HMI 어플리케이션 요소들은 (페이지, 콘텐츠 일부, 텍스트, 미디어 파일 등) mapp View 패키지에서 관리된다.

컨피규레이션 뷰(Configuration View)

컨피규레이션 뷰는 한 개 이상의 HMI 어플리케이션이 설정되고 관리되는 곳이다.

콘텐츠 편집기(Content Editor)

콘텐츠 편집기는 HMI 콘텐츠를 (콘텐츠의 일부) 작성하는 데 이용되는 Visual editor 이다. HMI 콘텐츠는 텍스트 편집기에서도 (XML 편집기) 처리될 수 있다.

툴바(Toolbar)

Visual editor 의 툴바는 디자인 단계에서 위젯을 사용하여 작업하기 위한 툴을 제공한다.

위젯 카탈로그(Widget Catalog)

콘텐츠의 일부가 (그래픽 방식 또는 XML 형식) 열리면, 카탈로그로부터 위젯을 추가하고 설정할 수 있다.

속성 창(Properties Window)

HMI 어플리케이션 요소들은 속성 창에서 설정된다. (예를 들어 위젯의) 속성에 따라, 편집을 위해 각각 다른 대화 상자가 이용된다.

3 페이지 작성과 내비게이션(navigation)

mapp View HMI 어플리케이션은 보통 여러 개의 HMI 페이지로 구성된다. HMI 클라이언트에서, 기계 작업자는 한 번에 한 개의 HMI 페이지 콘텐츠를 본다. 다른 페이지의 콘텐츠를 보려면, HMI 어플리케이션에서 페이지 전환이 필요하다.

HMI 어플리케이션 내비게이션은 기계 작업자가 HMI 어플리케이션 페이지 사이를 전환하는 방법을 결정한다.

이 장에서는 페이지 작성 방법과 페이지 사이에 내비게이션 기능을 설명한다.

3.1 개요 – 페이지 작성

디자인 목적을 위해, mapp View HMI 어플리케이션 페이지는 영역(area)로 구분된다. 이 영역의 콘텐츠(content)는 상호 독립적으로 디자인될 수 있다.

레이아웃(Layout)은 HMI 어플리케이션 페이지가 구성되는 방법을 정의한다. 레이아웃은 독립 구조이고 HMI 어플리케이션 페이지(Page)에 정의된 영역들로 구성된다.

HMI 어플리케이션 페이지를 디자인 하려면, 먼저 페이지에 영역이 구성되는 방법을 정의할 필요가 있다. 이것은 요구하는 크기와 요구하는 영역의 수를 포함하는 레이아웃을 선택함으로써 이루어진다. 특정한 HMI 어플리케이션 페이지에, 정의된 콘텐츠를 각 영역에 할당할 수 있다.

그럼 HMI 어플리케이션 페이지 구조는 실질적 예시의 형태로 HMI 어플리케이션 페이지를 디자인하는 데 이용되는 요소들을 보여준다.

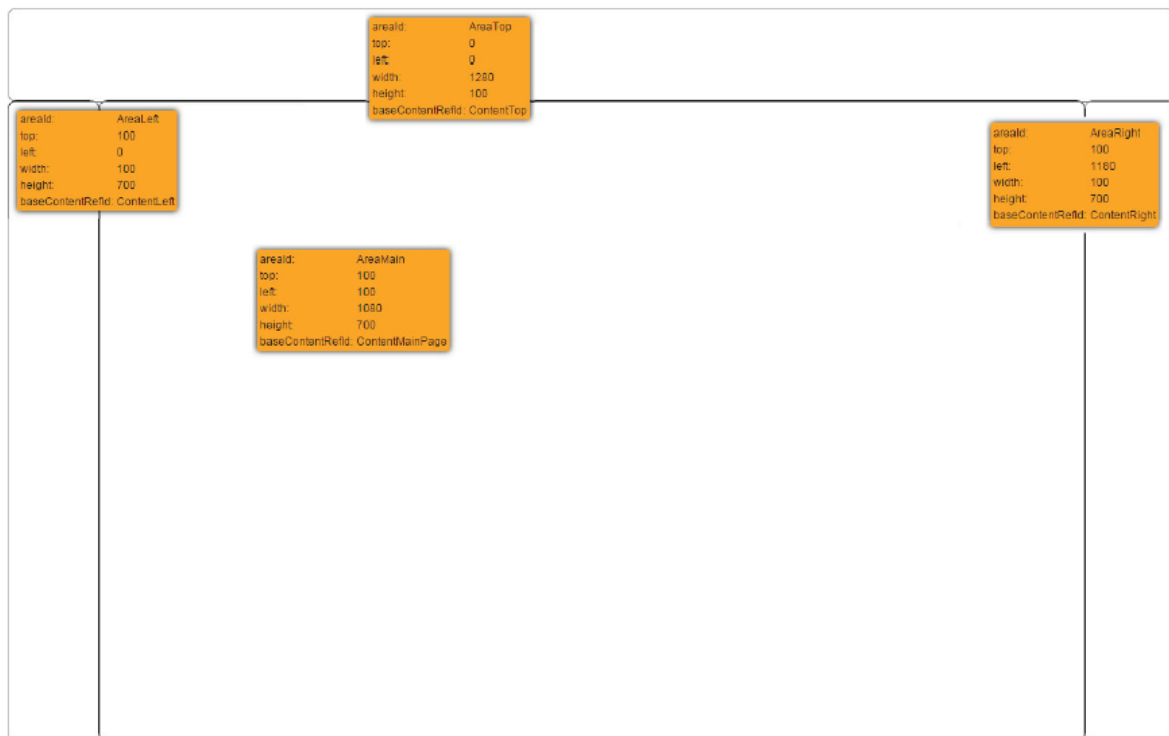


그림 6 HMI 어플리케이션 페이지 구조

3.1.1 레이아웃(Layout)

레이아웃은 다수에 영역이 구성되는 공간을 정의한다. 레이아웃에 의해 구성된 공간은 파라미터 “너비(width)” 및 “높이(height)”를 이용하여 정의된다. 명확한 식별을 위해, 각 레이아웃은 고유한 “ID”를 갖는다. 그림 “레이아웃 오버뷰”는 레이아웃 개요를 보여준다.

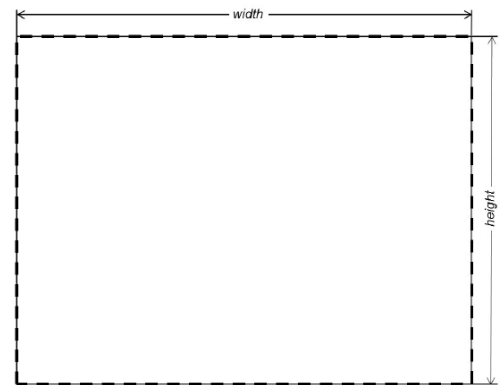


그림 7 레이아웃 오버뷰

3.1.2 영역(Area)

영역은 레이아웃에 정의된 일부이다.

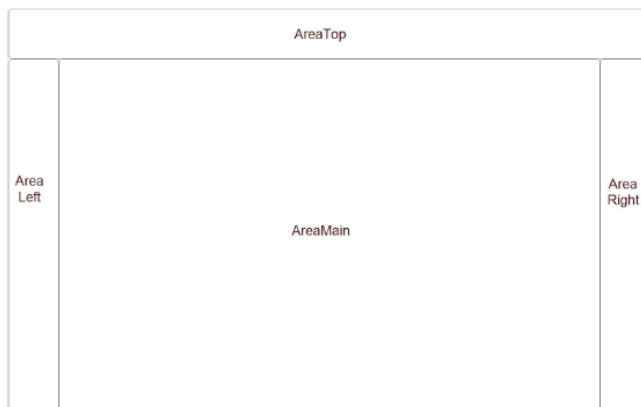


그림 8 4개 영역을 갖는 레이아웃

레이아웃 내 각 영역은 고유한 “ID”를 갖는다. 영역의 “ID”는 그 영역이 정의된 레이아웃 내에서 고유하다. 영역은 크기에 (“너비(width)” 및 “높이(height)”) 따라서 정의되며, 크기는 픽셀로 규정되고 레이아웃 내에 위치도 (“상단(top)” 및 “좌단(left)”) 또한 픽셀로 정의된다.

레이아웃 내 영역 위치에 대한 기준점은 좌측 상단 모서리이다.

3.1.3 콘텐츠(Content)

콘텐츠는 HMI 어플리케이션에서 표시될 수 있는 콘텐츠의 일부를 가리킨다. 하나의 콘텐츠는 전체적으로 고유한 “ID”와 크기를(“너비(width)” 및 “높이(height)”) 이용하여 식별된다.

하나의 콘텐츠에 위젯(Widgets)이 위치할 수 있다.

그림 “위젯을 포함하는 콘텐츠”은 다음 위젯을 포함한 콘텐츠를 보여준다:

- 버튼(Button)
- 라벨(Label)
- 수치 출력(Numeric output)
- 수치 입력(Numeric input)
- 이미지(Image)
- 측정 시스템 선택 (Measurement-SystemSelector)
- 언어 선택(LanguageSelector)

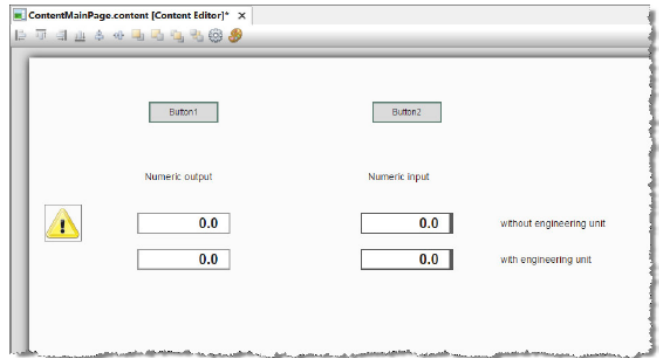


그림 9 위젯을 포함하는 콘텐츠

3.1.4 페이지(Page)

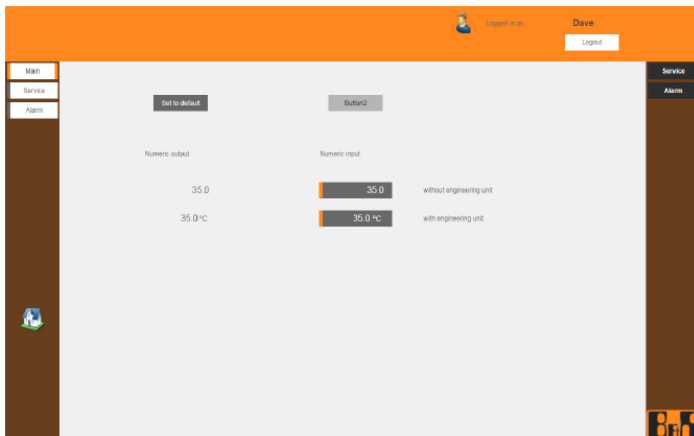


그림 10 참조된 영역, 콘텐츠 및 위젯을 갖는 페이지

그림 “참조된 영역, 콘텐츠 및 위젯을 갖는 페이지”는 샘플 페이지를 보여준다. 네 개 영역이 잘 구별되도록 각각 다른 배경 색상으로 구성하였다.

페이지는 HMI 어플리케이션 클라이언트의 가시적 공간에서 표시될 수 있는 콘텐츠를 정의한다.

페이지는 전체적으로 고유한 “ID”를 이용하여 식별된다. 페이지는 영역을 구성하기 위해서 어느 레이아웃이 사용되고 개별 영역에 어느 콘텐츠가 할당되는지를 정의한다.

3.2 단계별 페이지 작성

첫 번째 예제의 목적은 Automation Studio 프로젝트 “mappViewGettingStarted”를 기반으로 페이지 한장을 HMI 어플리케이션에 제작하는 것이다. 이 프로젝트에는 자동화 어플리케이션과 요구되는 이미지가 포함되어 있다.

페이지는 4 개의 영역을 갖는 한 개의 레이아웃을 사용할 것이며, 각 영역에는 특정한 콘텐츠가 할당된다. 4 개의 콘텐츠를 보다 잘 식별하기 위해 각각 다른 배경 색상이 사용된다.

그림 참조된 레이아웃, 영역 및 콘텐츠를 갖는 메인 페이지(MainPage)은 디자인된 페이지 구조의 개요와 사용되는 요소들(페이지, 레이아웃, 영역, 콘텐츠 부분)의 “ID”를 보여준다.



그림 11 참조된 레이아웃, 영역 및 콘텐츠를 갖는 메인 페이지(MainPage)

예제: 첫번째 HMI 어플리케이션 페이지 작성하기

다음 장에서, 페이지 한 개 있는 mapp View HMI 어플리케이션을 작성한다. 반드시 다음 단계들을 수행해야 한다.

- 1) mappViewGettingStarted 프로젝트 열기
- 2) 로지컬 뷰(Logical View)에서 패키지 추가
- 3) 레이아웃 작성
- 4) 콘텐츠 작성
- 5) 페이지 작성
- 6) 컨피규레이션 뷰(Configuration View)에 파일 추가
- 7) mapp View 서버(.mappviewcfg) 설정
- 8) Visualization object(.vis file) 설정
- 9) 프로젝트를 구축하고 전송하기
- 10) HMI 어플리케이션을 브라우저에 표시하기

3.2.1 mappViewGettingStarted 프로젝트

mappViewGettingStarted 프로젝트가 압축 파일(.zip)로 제공된다. 이 프로젝트는 mapp View HMI 어플리케이션을 작성을 위한 기반이다.

예제: mappViewGettingStarted 프로젝트 열기

mappViewGettingStarted 프로젝트 열기

3.2.2 로지컬 뷰에 패키지 추가하기

패키지 추가

mappViewGettingStarted 프로젝트가 열리면, 로지컬 뷰에 mapp View 패키지(mappView)와 시각화 패키지(visualization package)를 추가한다. 시각화 패키지는 mapp View 패키지 아래에 추가한다.

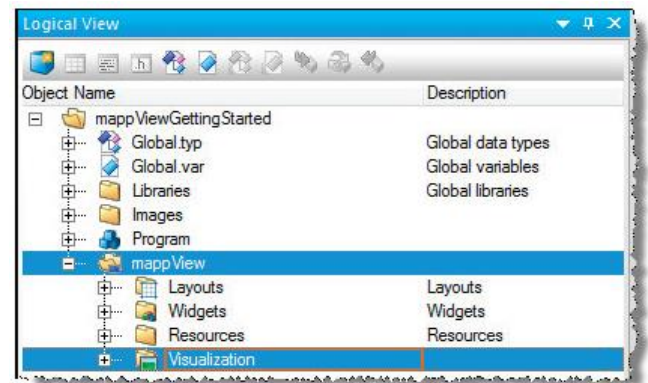


그림 12 mapp View 패키지와 visualization 패키지가 추가되었다.

예제: 패키지 추가하기

- 1) mapp View 패키지(mapp view package) 추가
- 2) visualization 패키지(visualization package) 추가

?

Visualization / mapp View / Engineering / HMI organization / Logical View

- mappView package
- The mappView visualization package

3.2.3 레이아웃(layout) 작성

레이아웃 추가

모든 것을 영역(area)으로 나누기 위해서는 레이아웃이 요구된다. 레이아웃은 로지컬 뷰에 추가된다.

로지컬 뷰에서 레이아웃 패키지가 선택되면, Object Catalog 에서 새로운 레이아웃을 추가할 수 있다. 그 다음에는 추가된 레이아웃 파일 이름을 “MyLayout”으로 바꾼다.

Property name	Value
id	"MyLayout"
height	"800"
width	"1280"

표 1 레이아웃 파일 속성

“MyLayout.layout” 파일을 더블 클릭하면 Automation Studio 작업 공간에 XML 편집기가 열린다.

표에 파라미터를 반드시 XML 파일에 입력해야 한다.

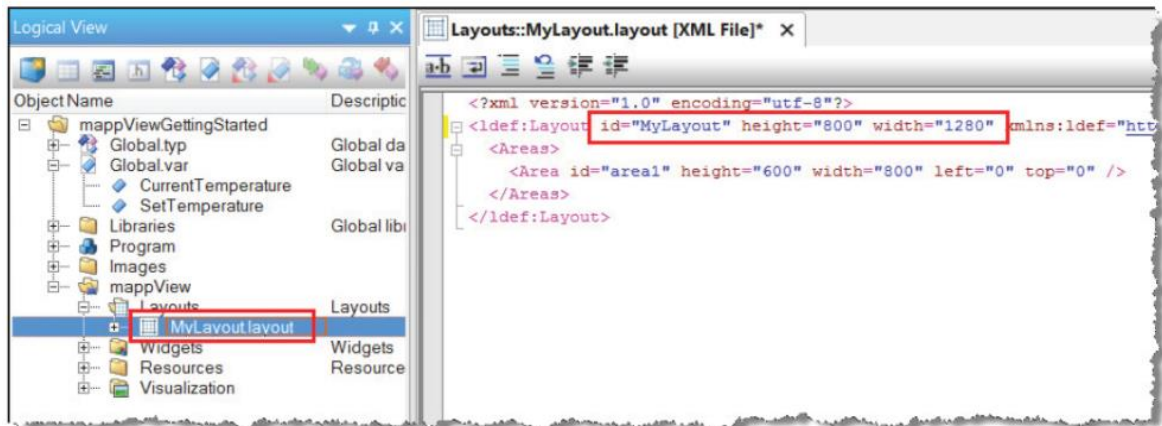


그림 13 MyLayout 파일이 로지컬 뷰에 추가되고, XML 에디터 파일



HMI application / mapp View / Engineering / Layout and areas / Adding a layout

3.2.4 영역(area) 작성

페이지는 레이아웃에 영역들을 정의함으로써 각각 다른 구역으로 나누어진다.

예제: 영역 정의 (XML)

“MyLayout.layout”에는 4 개 영역으로 구성될 것이며 파라미터들은 반드시 아래 표에 규정된 값으로 설정되어야 한다.

ID	Height	Width	Left	Top
AreaMain	700	1080	100	100
AreaTop	100	1280	0	0
AreaLeft	700	100	0	100
AreaRight	700	100	1180	100

표 2 레이아웃에 사용되는 영역에 대한 속성

최종 결과

표시된 그림은 “MyLayout”의 4 개 영역에 대한 정의를 보여준다.

```
<?xml version="1.0" encoding="utf-8"?>
<ldf:Layout id="MyLayout" height="800" width="1280"
xmlns:ldf="http://www.br-automation.com/iat2015/layoutDefinition/v2">
  <Areas>
    <Area id="AreaMain" height="700" width="1080" left="100" top="100" />
    <Area id="AreaTop" height="100" width="1280" left="0" top="0" />
    <Area id="AreaLeft" height="700" width="100" left="0" top="100" />
    <Area id="AreaRight" height="700" width="100" left="1180" top="100" />
  </Areas>
</ldf:Layout>
```

3.2.5 콘텐츠(Content) 작성

콘텐츠 추가하기

“AreaContents” 폴더에 콘텐츠 3 개를 추가한다. “AreaContents” 폴더는 다수 페이지에서 사용될 수 있는 다양한 콘텐츠를 저장하는데 이용된다.

“AreaContents” 폴더를 선택하고 콘텐츠 파일 3 개를 추가한다.

로지컬 뷰에 콘텐츠 파일을 추가한 후에는, 반드시 “ContentTop”, “ContentLeft” 및 “ContentRight”으로 명칭을 변경해야 한다.

추가된 콘텐츠 파일은 추가된 콘텐츠 파일 3 개를 (고유한 파일 이름과 함께) 보여준다.

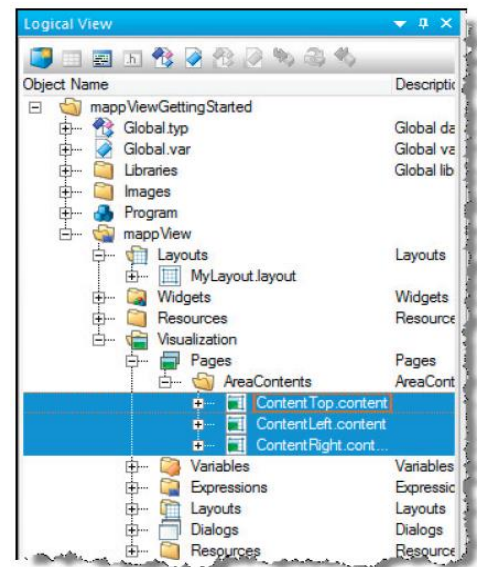


그림 14 추가된 콘텐츠 파일

콘텐츠 편집하기

콘텐츠 한개를 열고 Visual editor 또는 텍스트 편집기(text editor)로 열고 편집할 수 있다.

다음 장에서는 Visual editor 사용 절차에 대해 설명한다.

XML 편집기를 사용하여 콘텐츠를 편집하는 절차는 도움말 시스템에서 확인할 수 있다.

Visual editor

콘텐츠 파일을 더블 클릭하면 콘텐츠 편집기와 속성(Properties) 창이 열린다. 속성 창은 콘텐츠 속성을 (“Name”, “width”, “height”) 편집하는 데 이용될 수 있다.

속성 창에는 콘텐츠의 고유한 “Name”, “width” 및 “height”를 입력한다.

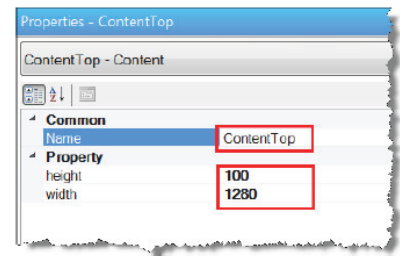


그림 15 ContentTop 속성 - Visual editor



로지컬 뷰에 추가되는 콘텐츠 파일 이름을 고유한 “이름”으로 명명 할 것을 권장한다.

예제: 모든 콘텐츠 편집하기

콘텐츠의 파라미터 name, height 및 width 는 반드시 다음과 같이 정의되어야 한다.

Name	Height (px)	Width (px)
ContentTop	100	1280
ContentLeft	700	100
ContentRight	700	100

표 3 콘텐츠의 상단, 좌단 및 우단 속성



[HMI application / mapp View / Engineering / HMI page content / Adding content](#)

3.2.6 페이지(Pages) 생성

드래그 앤 드롭(drag-and-drop) 또는 더블 클릭을 이용하여 Object Catalog 에서 기존 “Pages” 패키지에 페이지 오브젝트를 추가한다. “Pages” 패키지는 mapp View 시각화 패키지의 일부이다.

다음 단계는 영역과 콘텐츠를 참조할 수 있는 “MainPage”를 작성하는 것이다.

“Pages” 패키지를 선택하면 Object Catalog 의 페이지를 로지컬 뷰에 추가하고 이름을 변경할 수 있다.

MainPage 패키지 아래에 “ContentMainPage” 콘텐츠를 추가하고, 더블 클릭하여 아래와 같이 설정한다.

Property name	Value
Name	"ContentMainPage"
Height	"700"
Width	"1080"

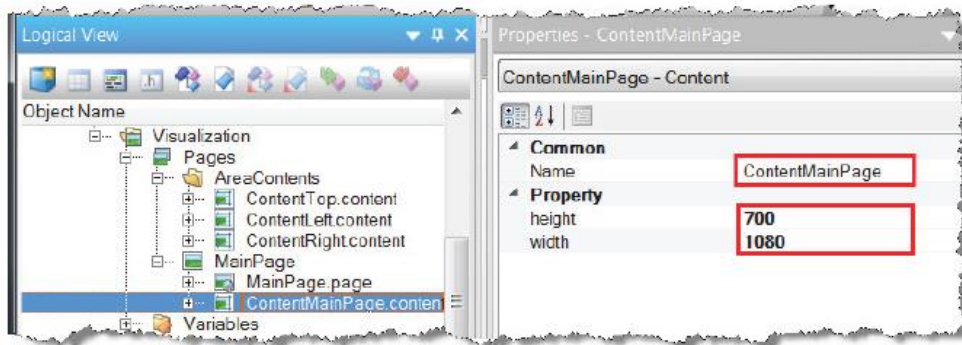
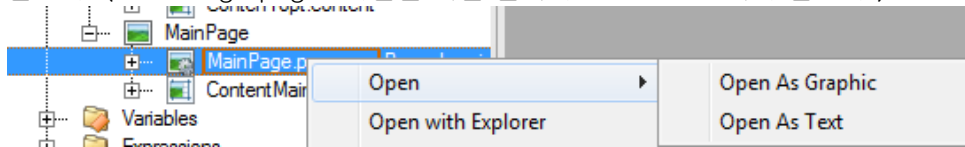


그림 16 속성창에서 ContentMainPage

“MainPage.page” 파일 선택 후 마우스 우클릭, Open / Open AS Graphic 을 누르면 visual editor 가 열린다. (“MainPage.page” 파일을 더블 클릭하면 XML 편집기가 열린다.)



고유의 페이지를 식별하기 위해서, 반드시 글로벌하게 고유한 “pageId”를 규정해야 한다.
“MainPage”에 사전 정의한 레이아웃을 이용하기 위해서, 속성(Property) / “layoutId”에 레이아웃 “MyLayout”을 선택한다.

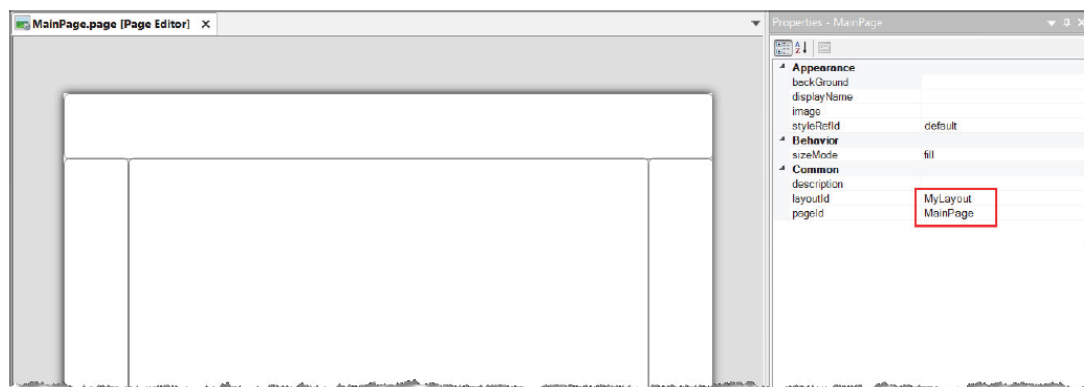


그림 17 레이아웃 할당을 위한 페이지의 visual editor

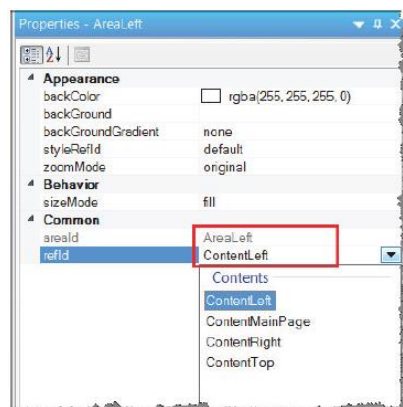


그림 18 영역을 선택했을 때 콘텐츠를 할당할 수 있다.

Visual editor 에서 영역을 선택했을 때, 영역의 속성이 출력된다. 해당 콘텐츠 할당은 여기서 수행한다.

런타임 중 각 콘텐츠가 명확히 인식될 수 있도록, 각 콘텐츠의 고유 배경 색상을 (backColor) 정의한다. 색상 코드(color code)는 속성 /“backColor” 에 rgba 코드로 규정된다.

refId	areald	backColor
ContentMainPage	AreaMain	rgba(240,240,240,1)
ContentTop	AreaTop	rgba(255,133,0,1)
ContentLeft	AreaLeft	rgba(102,53,0,1)
ContentRight	AreaRight	rgba(102,53,0,1)

표 4 Main Page

	HMI application / mappView / Engineering / HMI pages / Adding HMI pages
---	---

3.2.7 컨피규레이션 뷰(Configuration View)에 파일 추가하기

“MainPage” 작성 및 설정 후, 반드시 mapp View 서버와 HMI 어플리케이션을 설정해야 한다. 이것은 Object Catalog로부터 컨피규레이션 뷰(Configuration View) / mapp View 노드 아래에 “mapp View 설정” (“Config.mappviewcfg”) 및 “Visualization” (“Visualizat.vis”)를 추가함으로써 이루어진다.

mapp View visualization object (.vis)

mapp View visualization object (.vis)는 로지컬 뷰와 컨피규레이션 뷰의 시각화 컴포넌트 정의이고 클라이언트에서 페이지 표시와 관련이 있다. 각 설정은 1~n 개의 visualization object (.vis)를 포함할 수 있다.

.vis 파일에 입력된 모든 정의와 파라미터에 관한 정보는 Automation Studio 도움말에서 찾아볼 수 있다.

	HMI application / mappView / Engineering / HMI organization / Configuration View / mapp View visualization object
---	---

mapp View server configuration (.mappviewcfg)

정적 설정은 mapp View 서버에 대한 다양한 세팅을 설정할 수 있다.

서버에 대한 정확한 설정은 Automation Studio 도움말에서 찾아볼 수 있다.

	HMI application / mappView / Engineering / HMI organization / Configuration View / mapp View configuration
---	--

예제: 서버 설정과 Visualization object 추가하기

Configuration View 에 반드시 아래 파일들이 추가되어야 한다.

- 1) mapp View 설정 (“Config.mappviewcfg”)
- 2) Visualization object (“Visualizat.vis”)

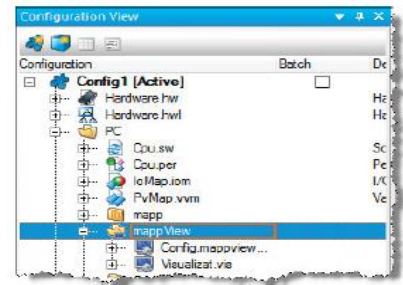


그림 19 Configuration View 에 Configuration
과 Visualization object 추가하기

Config.mappviewcfg 설정하기

.mappviewcfg 의 디폴트 세팅을 변경 없이 이용할 수 있다.

Visualizat.vis 설정하기

“Visualization ID”요소는 HMI 어플리케이션이 클라이언트로부터 시작되는 방법을 결정한다.

<StartPage>요소는 HMI 어플리케이션이 클라이언트에 전달될 때 어느 페이지가 표시되는지를 정의한다.

<Pages> 요소 목록에는 HMI 어플리케이션의 일부인 모든 페이지가 포함된다.

이미지는 isualization object (.vis)를 사용하여 실행하는데 관련된 콘텐츠들을 보여준다.

```
<?xml version="1.0" encoding="utf-8"?>
<vdef:Visualization id="Training"
xmlns:vdef="http://www.br-automation.com/iat2015/visualizationDefinition/v2">
  <StartPage pageRefId="MainPage" />
  <Pages>
    <Page refId="MainPage"/>
  </Pages>
```

3.2.8 예제: 브라우저에서 HMI 어플리케이션 열기

첫 번째 페이지("MainPage")가 작성되고 Visualization object (.vis)에서 참조된 후, 프로젝트를 컴파일하고 ARsim 에 전송할 수 있다.

HMI 어플리케이션은 다음 URL 을 이용하여 표시할 수 있다:

<http://127.0.0.1:81/index.html?visuid=Training>

예제: 프로젝트 컴파일하기와 브라우저에 표시하기

- 1) 프로젝트를 컴파일하고 전송하기
- 2) 브라우저를 열고 URL 을 입력하기

예상되는 결과

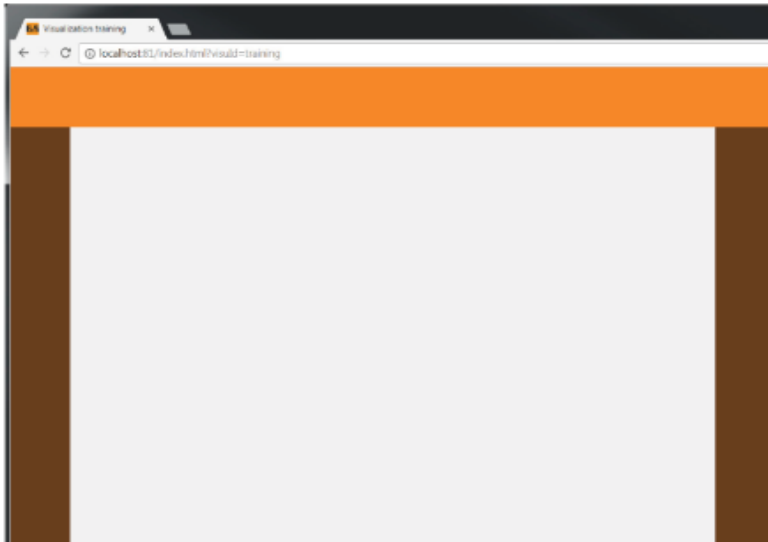


그림 20 영역 4 개와 레이아웃 1 개를 포함하는 HMI 어플리케이션이 브라우저에 표시됨.



[HMI application / mapp View / Guides / Getting started / Testing the HMI application](#)

3.3 내비게이션 개요

HMI 어플리케이션은 한 개 이상의 페이지로 구성된다. 페이지에 접근하려면 내비게이션 시스템이 필요하다.

이 시스템은 HMI 어플리케이션에서 페이지 사이를 이동하는 방법을 결정한다.

수동 내비게이션 스키마 예시는 3 개 페이지 (“MainPage”, ServicePage”, “AlarmPage”)와 이들 사이에 이동 옵션을 보여준다.

예시에서, 한 페이지에서 다른 페이지로 이동하는 것이 가능하다.

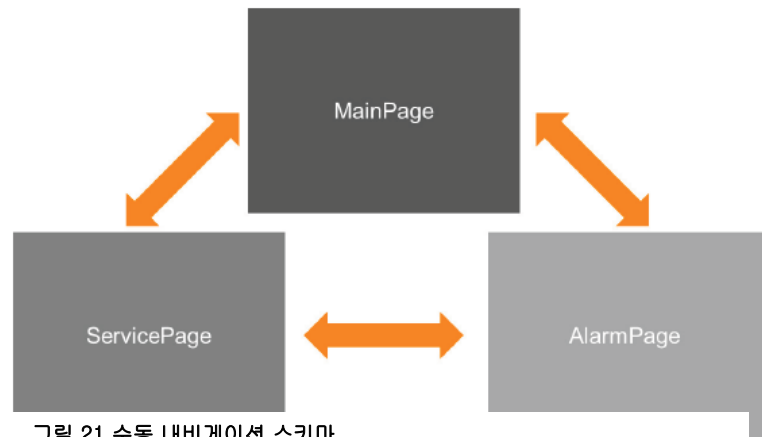


그림 21 수동 내비게이션 스키마

HMI 어플리케이션의 내비게이션에는 두 가지 옵션이 있다(수동과 자동 내비게이션). 두 가지 유형의 내비게이션을 조합할 수 있다.

3.3.1 수동 내비게이션

수동 내비게이션은 개별 “NavigationButton” 위젯을 이용하여 구현한다. 이 경우, 각 페이지에서 현재 페이지로 오고 갈 수 있는 “NavigationButton” 위젯이 배치되고 설정되어야 한다.

그림 수동 내비게이션 스키마는 각 페이지에 배치되고 설정된 “NavigationButton” 위젯 3 개를 보여준다. 개별 “NavigationButton” 위젯은 버튼을 누를 때 열려야 하는 페이지의 ID 를 함께 설정한다.

수동 내비게이션은 동일한 “NavigationButton” 위젯을 모든 페이지에서 이용할 수 있는 상황에 적합하다. 이 경우에, 이들은 한 개 콘텐츠 안에 배치되고 그 곳에서 설정한다. 그러면 이 내비게이션 콘텐츠는 각 페이지에서 재사용될 (참조될) 수 있다.

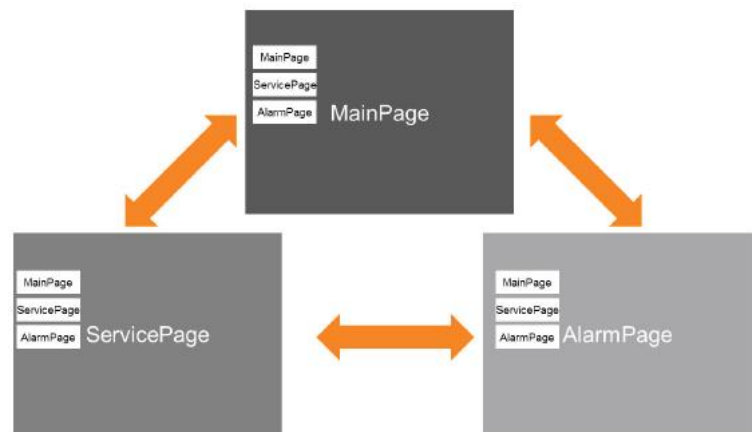


그림 22 수동 내비게이션 스키마

수동 내비게이션의 한 가지 단점은 이 방식이 환경에 독립적이라는 것이다. “NavigationButton” 위젯의 배치와 설정은 모든 페이지에서 동일하다. 만약 “NavigationButton” 위젯의 배치와 설정이 페이지에

따라 달라진다면, 여러분은 내비게이션을 위해 각각 다른 콘텐츠를 작성하고 해당 페이지에서 개별적으로 참조해야 할 것이다.

다수의 페이지가 있는 HMI 어플리케이션의 경우에는, 많은 작업이 요구 될 수 있다. 당장에는 광범위한 HMI 어플리케이션의 내비게이션 구조를 개발하는데 많은 작업이 요구되지만, 수동 내비게이션의 유지보수는 매우 용이하다. mapp View 에서 수동 내비게이션의 대안은 자동 내비게이션이다.



HMI application / mapp View / Engineering / HMI organization / Configuration View / Navigation

3.3.2 자동 내비게이션

자동 내비게이션은 내비게이션 구조 정의에 의존한다. 각각 다른 내비게이션 경로, 즉 어느 페이지에서 어느 페이지로 이동할 수 있는지에 대한 경로를 정의하는 데 이용된다. mapp View 는 런타임 중에 각 페이지에 대응되는 “NavigationButton” 위젯을 배치하는 것을 담당한다.

각 페이지는 현재 페이지의 내비게이션에 정의된 목표 페이지만 표시한다. “NavigationButton” 위젯 표시는 환경에 의존적이다.

자동 내비게이션 경로는 3 개 페이지 “Main-Page”, “ServicePage”, “AlarmPage”에 대한 자동 내비게이션을 보여준다. 모든 페이지로부터 모든 페이지로 이동이 가능하지만 각 페이지는 환경에 의존적인 내비게이션 목표만 보여준다.

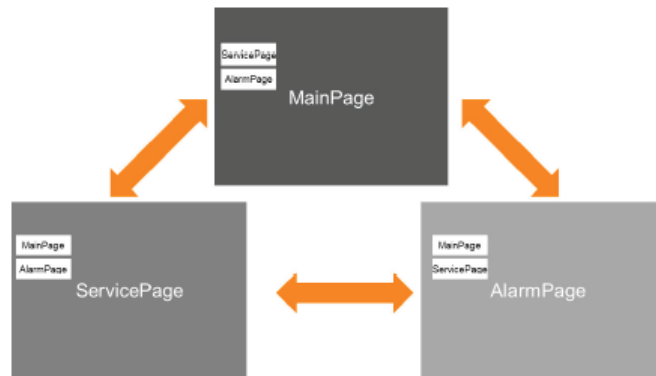


그림 23 자동 내비게이션 경로



HMI application / mapp View / Engineering / HMI organization / Configuration View / Navigation

3.4 수동 내비게이션 이용하기

예제의 목적은 다른 페이지 이동에 이용될 수 있는 수동 내비게이션 시스템을 작성하는 것이다. 수동 내비게이션을 이용하려면, 반드시 페이지 2 장을 추가 작성해야 한다 (“ServicePage”, “AlarmPage”). 수동 내비게이션 시스템은 Visual editor 에서 콘텐츠 “ContentLeft”에 위치하고 설정한다.

예제: 수동 내비게이션 설정하기

수동 내비게이션 설정을 위해 다음 장을 이용한다.
반드시 다음 단계를 수행해야 한다.

- 1) ServicePage 및 AlarmPage 작성
- 2) NavigationBar 위젯을 추가하고 설정
- 3) NavigationButton 위젯을 추가하고 설정
- 4) .vis file 에서 ServicePage 및 AlarmPage 를 참조
- 5) 프로젝트를 컴파일/전송하고 브라우저에 표시

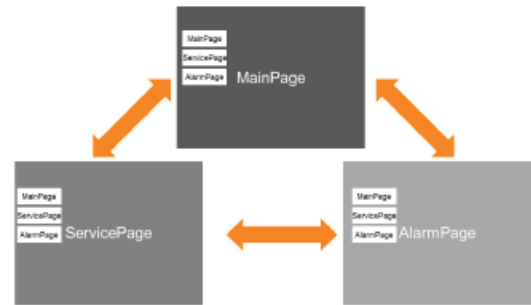


그림 24 수동 내비게이션 경로

3.4.1 ServicePage 및 AlarmPage 작성하기

수동 내비게이션을 표시하기 위해서는, 반드시 각각 새로운 콘텐츠를 갖는 페이지 2 장을 (“ContentServicePage” 및 “ContentAlarmPage”) 추가로 작성해야 한다. 페이지 2 장은 “MainPage”와 동일한 레이아웃(“MyLayout”) 과 참조 콘텐츠 “ContentServicePage”, “ContentAlarmPage”를 갖는다.

예제: ServicePage 및 AlarmPage 작성하기

페이지 작성에 관한 정보는, “단계별 페이지 작성” 을 참조한다.

- 1) ContentServicePage 를 갖는 ServicePage 를 작성
- 2) ContentAlarmPage 를 갖는 AlarmPage 를 작성

페이지마다 다른 외관을 제공하기 위해, 각각의 주 콘텐츠를 할당할 때 개별적으로 속성 “backColor”를 설정한다.

Name	ServicePage	AlarmPage
ContentServicePage/AreaMain	X	
ContentAlarmPage/AreaMain		X
ContentTop/AreaTop	X	X
ContentLeft/AreaLeft	X	X
ContentRight/AreaRight	X	X

표 5 콘텐츠와 영역 할당

3.4.2 NavigationBar 위젯

“NavigationBar” 위젯(widget)은 “NavigationButton” 위젯을 위한 컨테이너로 이용된다. “NavigationButton” 위젯이 내비게이션 바(navigation bar)에 추가될 때, 이들은 이미 추가된 다른 “NavigationButton” 위젯들과 자동적으로 함께 자리를 잡거나 정렬 될 수 있다. 사용자는 개별 “NavigationButton” 위젯을 수동으로 정렬시켜야 하는 노력을 절약 할 수 있다.

NavigationBar 위젯

“ServicePage”, “AlarmPage” 작성 후, “NavigationBar” 위젯을 “ContentLeft”에 추가하고 아래와 같이 설정한다:

Property name	Value
Layout/Size	100:300
Behavior / Child positioning	relative
Common/Name	NavigationBar1

표 6 NavigationBar 위젯 속성

드래그 앤 드롭(drag-and-drop) 기능을 이용하여 위젯 카탈로그로부터 “NavigationBar” 위젯을 “ContentLeft”에 추가한다.

컨테이너의 크기를 “Layout/Size” = 100:300 으로 설정한다.

“NavigationButton” 위젯들이 상대적으로 상호 정렬되도록, “Behavior/child positioning” 속성을 “relative”으로 설정한다.

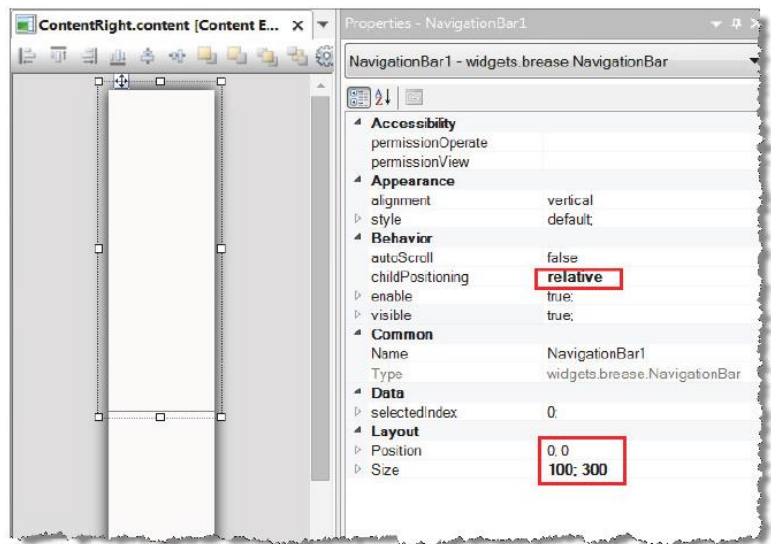


그림 25 속성창에서 NavigationBar 위젯

3.4.3 NavigationButton 위젯

“NavigationButton” 위젯은 특정한 페이지로 내비게이션을 한다.

“NavigationButton” 위젯 추가와 설정

“NavigationBar” 위젯을 추가 및 설정한 후, “MainPage”에 대한 “NavigationButton” 위젯을 추가하고 설정한다.

“NavigationButton” 위젯은 드래그 앤 드롭 기능으로 위젯 카탈로그에서 “NavigationBar” 위젯에 추가된다. “NavigationButton”에 표시되는 텍스트는 속성은 “Appearance/text”에서 규정된다. “NavigationButton” 위젯의 고유 명칭은 속성 “Common/Name”에 정의되고, “NavigationButton” 위젯을 클릭할 때 이동되는 페이지는 속성은 “Data / pageId”에서 설정한다.

NavigationButton의 속성 창은 “MainPage”로 이동을 위한 “NavigationButton” 위젯 값을 보여준다.

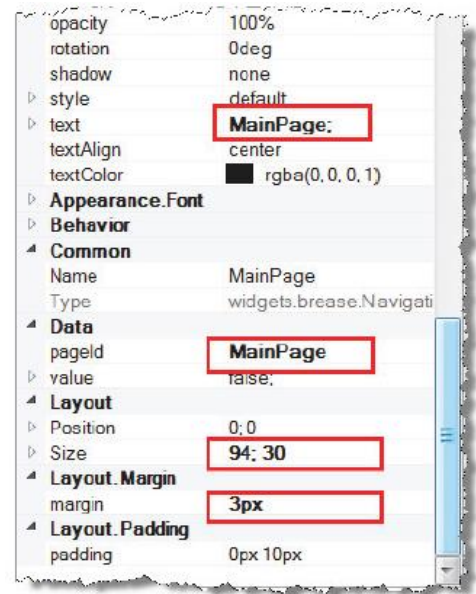


그림 26 NavigationButton의 속성 창(Properties window)

예제: ServicePage, AlarmPage에 내비게이션 버튼 추가하기

“ServicePage” 및 “AlarmPage”로 이동하려면 다음 파라미터를 이용하는 절차를 반드시 수행해야 한다.

NavigationButton name	Appearance / text	Common/Name	Data / pagelId
MainPage	MainPage	MainPageButton	MainPage
ServicePage	ServicePage	ServicePageButton	ServicePage
AlarmPage	AlarmPage	AlarmPageButton	AlarmPage

표 7 내비게이션 버튼의 속성

3.4.4 .vis file 파일에 ServicePage 및 AlarmPage 입력하기

새로 작성된 페이지가 HMI 어플리케이션에서 반영되기 위해서는, 반드시 Visualization object(.vis)에 페이지가 참조되어야 한다.

“ServicePage” 및 “AlarmPage”는 반드시 아래 이미지에 표시된 바와 같이 요소 <Pages>에 입력되어야 한다.

```
<StartPage pageRefId="MainPage" />
<Pages>
  <Page refId="MainPage"/>
  <Page refId="ServicePage"/>
  <Page refId="AlarmPage"/>
</Pages>
```

3.4.5 수정사항 전송하기

예제: 프로젝트를 컴파일하고 브라우저에서 화면보기

모든 “NavigationBar” 위젯들이 추가되고 편집된 후, 프로젝트를 컴파일하고 ARsim 에 전송할 수 있다.

HMI 어플리케이션을 아래 URL 을 이용하여 표시할 수 있다:

<http://127.0.0.1:81/index.html?visuid=Training>

예상되는 결과

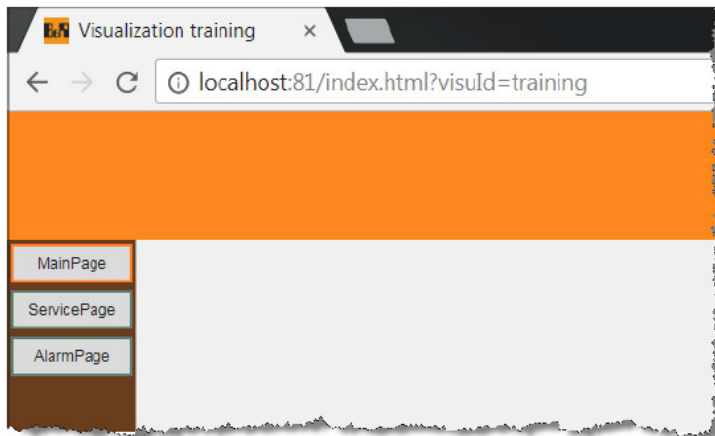


그림 27 수동 내비게이션을 포함하는 HMI 어플리케이션

3.5 자동 내비게이션 이용하기

이 예제의 목표는 페이지 이동에 이용될 수 있는 자동 내비게이션 시스템을 작성하는 것이다.

자동 내비게이션 옵션을 표시하는 데 이용되는 “Navigation” 위젯을 Visual editor 를 이용하여 “ContentRight”에 추가한다.

수동 내비게이션과는 달리, “NavigationBar” 위젯은 현재 페이지를 표시하지 않는다. 이것은 “내비게이션 오브젝트(navigation object)”를 설정하여 구현된다.

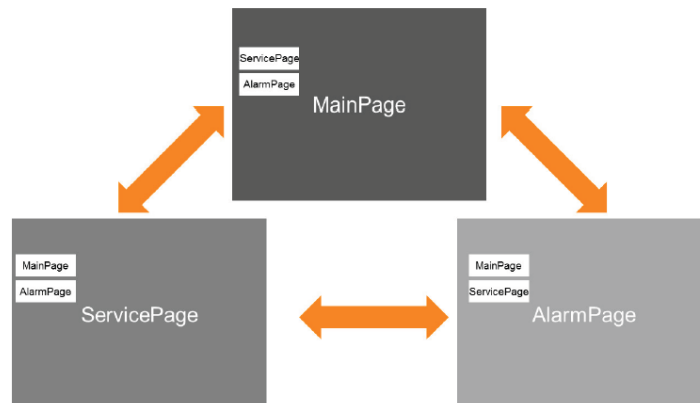


그림 28 자동 내비게이션 경로

예제: 자동 내비게이션 설정하기

자동 내비게이션은 아래 장의 도움을 받아 설정한다.

- 1) 내비게이션 오브젝트를 컨피규레이션 뷰(Configuration View)에 추가하고 내비게이션 경로를 설정
- 2) “Navigation” 위젯을 “ContentRight”에 추가하고 설정
- 3) .vis 파일에 내비게이션 ID 를 입력
- 4) 프로젝트를 컴파일/전송하고 브라우저에 표시

3.5.1 컨피규레이션 뷰(Configuration View) 에 내비게이션 오브젝트

자동 설정을 이용하면, 내비게이션 경로는 컨피규레이션 뷰에 내비게이션 오브젝트(navigation object)에 정의된다. “NavigationButton” 위젯은 런타임 중에 자동으로 “내비게이션(Navigation)” 위젯에 표시된다.

Object Catalog 에서 내비게이션 오브젝트를 “mapp View” 패키지에 추가해야 한다.

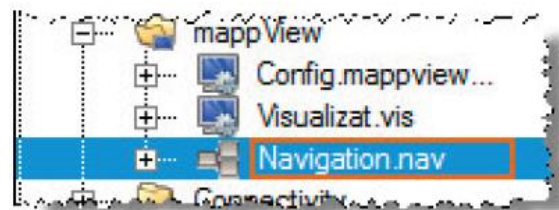


그림 29 추가된 내비게이션 오브젝트(navigation object)

각 내비게이션 오브젝트는 고유 “ID”와 함께 어느 소스 페이지로부터 (NavigationPath refId=”) 어느 목표 페이지로 (Destination refId=”) 내비게이션이 가능한지를 정의하는 내비게이션 경로가 반드시 포함되어야 한다. 속성 “index”는 런타임 중에 “NavigationButton” 위젯이 “Navigation” 위젯에 배치되는 순서를 규정한다.

내비게이션 오브젝트 설정하기

첫 단계는 고유한 <Navigation ID> “AutoNavigation”을 할당하는 것이다.

아래와 같이 내비게이션 경로를 정의한다.

```
<?xml version="1.0" encoding="utf-8"?>
<ndef:Navigation id="AutoNavigation" xmlns:ndef="http://www.brautomation.
com/iat2015/navigationDefinition/v2">
  <NavigationPaths>
    <NavigationPath refId="MainPage">
      <Destination refId="ServicePage" index="0" />
      <Destination refId="AlarmPage" index="1" />
    </NavigationPath>

    <NavigationPath refId="ServicePage">
      <Destination refId="MainPage" index="0" />
      <Destination refId="AlarmPage" index="1" />
    </NavigationPath>
  </NavigationPaths>
</ndef:Navigation>
```

```
<NavigationPath refId="AlarmPage">
    <Destination refId="MainPage" index="0" />
    <Destination refId="ServicePage" index="1" />
</NavigationPath>

</NavigationPaths>
</ndef:Navigation>
```

3.5.2 내비게이션 위젯을 추가하고 설정하기

“Navigation” 위젯은 정의된 내비게이션 경로를 기반으로 자동으로 내비게이션 버튼을 표시한다.

내비게이션(Navigation) 위젯 추가하기

드래그 앤 드롭을 이용하여 툴 박스로부터 “Navigation” 위젯을 추가하고 다음과 같이 설정한다.

먼저 “Layout/Size”에 “Navigation” 위젯 크기를 정의한다. “NavigationButton” 위젯이 잘리지 않고 올바르게 표시되려면, “NavigationButton” 위젯의 디폴트 값을 반드시 “Layout.Size/buttonWidth”에서 변경해야 한다.

위젯의 고유 명칭은 “Common / Name”에 입력한다.

올바른 내비게이션 경로가 사용되도록, “Data/navRefId”에 내비게이션 오브젝트의 참조 ID 를 (“AutoNavigation”) 규정한다.

Property name	Value
Data/navRefId	AutoNavigation
Layout/Size	100;250
buttonMargin	3
Layout.Size / buttonWidth	94

표 8 AutoNavigationWidget의 속성

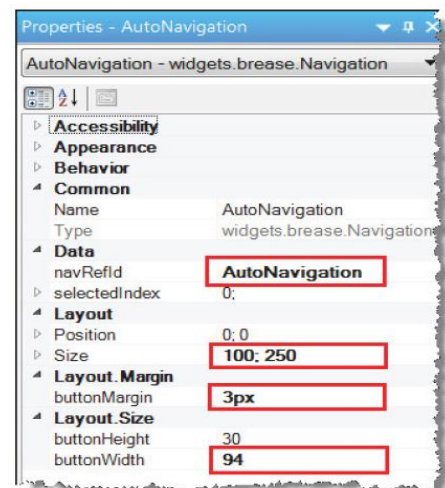


그림 30 내비게이션 위젯의 속성 창

3.5.3 .vis 파일에서 내비게이션 참조하기

내비게이션 오브젝트가 HMI 어플리케이션에 반영되려면, 반드시 .vis 파일에서 “Navigation ID”가 참조되어야 한다.

XML 에서 속성 <Navigations>의 코멘트를 해제하고 속성“refId”에 대한 “AutoNavigation”을 입력한다.

```
<Navigations>
  <Navigation refId="AutoNavigation" />
</Navigations>
```

예상되는 결과

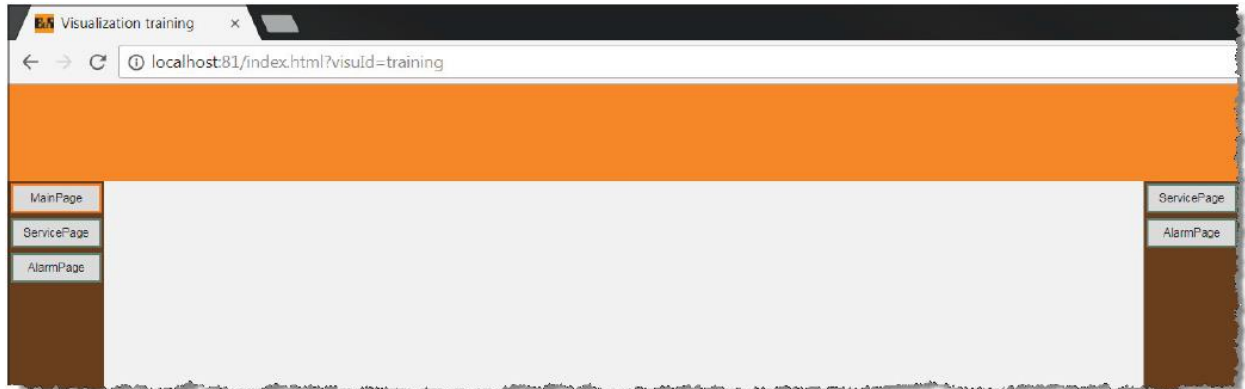


그림 31 수동과 자동 내비게이션을 포함하는 HMI 어플리케이션



내비게이션 예시에서, 페이지 “ID”는 “NavigationBar” 위젯에 대한 라벨로서 표시되었다. 맞춤화 된 라벨을 만드는 방법은 “현지화 (Localization)” 에 설명되어 있다.



mapp View 서버는 제어기에 직접 전송할 수 있다. 시뮬레이션 모드가 비활성화되고 프로젝트가 전송한다. 그리고 나서 HMI 디바이스는 제어기의 IP 주소를 통해 접근한다 (예: <http://10.43.15.46:81/index.html?visuid=Training>).

보기:

- [“제어기에 mapp View 서버 운영하기”](#)

4 시각적 외관 – 스타일링(Styling)

스타일링은 HMI 어플리케이션이 실제 기능에는 영향을 미치지 않고 HMI 어플리케이션 외관을 디자인 할 수 있는 기능이다..

4.1 스타일링 개요

전체적으로 HMI 어플리케이션이 겉으로 보이는 모습과 개별 요소들의 외관은 스타일링 가능한 속성(properties), 스타일(styles) 및 테마(Theme)를 이용하여 결정된다.

4.1.1 스타일 가능한 속성(Styleable property)

스타일링 가능한 속성은 외관에 영향을 미치는 HMI 요소의 단일 속성(예, 위젯이나 페이지)을 가리킨다. 스타일링 가능한 속성의 예는 위젯의 배경 색상이나 경계선 너비이다.

4.1.2 스타일(Style)

스타일(Style)은 HMI 요소 유형의 스타일링 가능한 모든 속성을 함께 집단화한다. HMI 요소의 각 유형에 여러 개의 스타일이 있을 수 있다. HMI 요소 유형으로써 버튼 위젯을 살펴보면, 버튼 위젯 유형에 대한 스타일 (버튼 스타일)에는 버튼 각각에 스타일 가능한 속성의 구체적인 값이 포함된다. 버튼의 스타일링 가능한 속성을 특정한 값으로 설정함으로써 버튼 위젯 유형에 대한 몇 가지 스타일이 가능하다. 예를 들어, 한개의 버튼 스타일 배경 색상은 녹색일 수 있고; 또 다른 예는 청색으로 설정할 수 있다. 스타일은 프로젝트 개발 중에 각각 다른 HMI 요소 인스턴스에 할당된다. 하나의 스타일에 HMI 요소의 인스턴스가 할당되면(예, Button01), 모든 인스턴스의 스타일링 가능한 속성은 할당된 스타일 값으로 설정된다.

4.1.3 테마(Theme)

테마는 각각 다른 HMI 요소 유형에 대한 스타일 그룹이다. 테마에는 모든 HMI 요소 유형에 대해 배경 색상을 청색의 각각 다른 색조(shade)로 설정하는 스타일들이 포함될 수 있다. 또 다른 테마에는 모든 HMI 요소 유형에 대해 배경 색상을 녹색의 다른 색조로 설정하는 스타일이 포함될 수 있다. 이러한 방식으로 청색과 녹색을 주제로 디자인 할 수 있다. 테마 스타일의 스타일링 가능한 속성 값들은 그들이 서로 잘 어울리도록 만든다. 테마에 대한 또 다른 사용 사례에는 자연 광선이 역할을 하는 HMI 어플리케이션에서 주간/야간 표시 전환과 관련된다. 야간 스타일은 어두운 배경 색상과 밝은 문자 폰트의 위젯을 표시하고, 주간 스타일에는 밝은 배경 색상과 어두운 문자 폰트를 포함할 수 있다.



그림 32 다른 테마를 갖는 HMI 어플리케이션 표시하기

4.2 스타일(styles) 이용하기

이 예제의 목적은 “버튼(Button)” 위젯의 맞춤형 스타일을 만들고 테마를 이용하여 수동 내비게이션을 위한 “NavigationButton” 위젯의 시각적 외관을 변경하는 것이다.

4.2.1 스타일링 가능한 속성(Styleable property)

스타일링 가능한 속성의 부분 집합은 각 위젯 유형을 위한 정의이다. 위젯 인스턴스 외관에 영향을 미치는 속성은 위젯 유형마다 다르다.



HMI application / mapp View / Widgets / Buttons / Navigation Button

내비게이션 버튼(NavigationButton) 위젯 스타일링

다음 예제의 목표는 “ContentLeft”에 위치한 수동 내비게이션을 위한 “NavigationButton” 위젯 외관을 변경하는 것이다.

속성(Properties) 창에서, “MainPage” “NavigationButton” 위젯에 배경 색상을 열은 적색으로, “ServicePage”와 “AlarmPage”의 “NavigationButton” 위젯은 열은 녹색으로 변경한다.

예제: “MainPage” 버튼에 스타일링 적용하기

“MainPage”라 불리는 “NavigationButton” 위젯 선택 후, 속성 창 / “Appearance/backColor” 에서 열은 적색 (rgba(255, 255, 192, 1))을 선택한다.

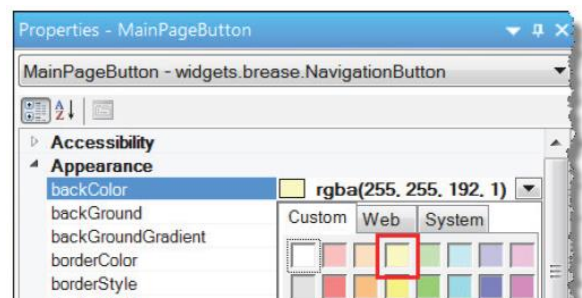


그림 33 속성 창의 배경 색상(backColor)

예제: ServicePage 와 AlarmPage 에 “NavigationButtons” 위젯에 스타일링 적용하기

“ServicePage”와 “AlarmPage”에 “NavigationButtons” 위젯에 동일한 절차를 반복한다. 열은 녹색에 대한 색상 값 (rgba(192, 255, 192, 1))을 사용한다.

예상되는 결과

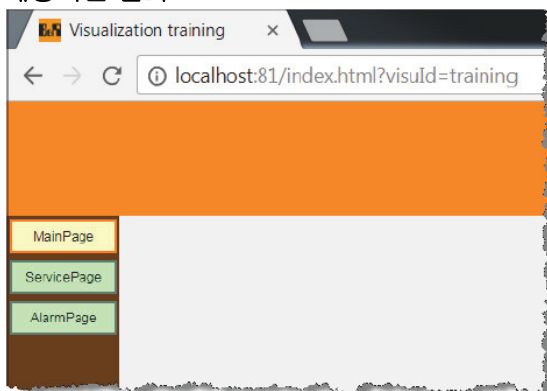


그림 34 내비게이션 버튼에 변경된 배경 색상(backColor)이 적용된 HMI 어플리케이션

이 HMI 어플리케이션에서는, 단 세 개의 “내비게이션 버튼” 위젯만 외관이 변경되었다. 실제 HMI 어플리케이션에서는, 외관 조정이 필요한 수 많은 위젯이 있는 것으로 가정해야 한다. 만약 엔지니어가 각 위젯 인스턴스의 스타일링 가능한 속성을 이용하여 작업하고자 한다면, 해야 할 작업과 오류 가능성이 불합리할 정도로 많을 것이다.

수 많은 위젯 인스턴스의 외관을 변경하려면, 스타일을 이용하는 더 좋은 방법이 있다.

4.2.2 B&R 테마와 스타일

B&R은 HMI 어플리케이션에 매력적인 외관을 신속하고 용이하게 제공할 수 있는 테마를 제공한다. 테마는 스타일 명칭을 규정함으로써 사용될 수 있는 모든 위젯 유형에 미리 정의된 맞춤형 스타일을 포함한다.

예제의 목적은 제공된 B&R 테마 패키지(B&R theme package)를 이용하고 수동과 자동 내비게이션의 “Button” 및 “NavigationBar” 위젯의 스타일을 포함시키는 것이다.

다음 장에서는 스타일과 테마를 작성하는 방법을 보다 상세하게 설명한다.

B&R 테마 추가하기

첫 단계는 Object Catalog 에서 B&R 테마를 로지컬 뷰, “Resources/Themes”에 추가하는 것이다.

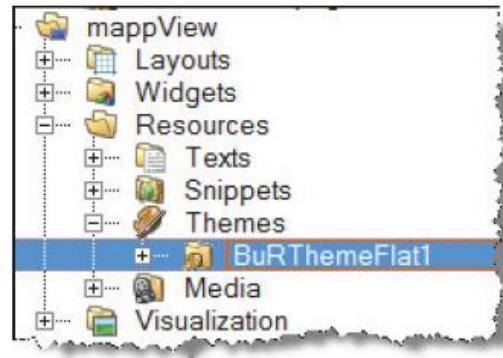


그림 35 로지컬 뷰 / B&R 테마 패키지



HMI application / mapp View / Themes and styles / Adding a Theme

버튼(Button) 위젯 추가하기

“Button” 위젯 두 개를 콘텐츠 “ContentMainPage”의 왼쪽과 오른쪽에 추가한다.

미리 정의된 스타일 “Command1”는 좌측 “Button”에, 미리 정의된 스타일 “Operate1”는 우측 “Button”에 할당되어야 한다.

예제: ContentMainPage 에 “Button” 위젯 두 개 추가하기

“button” 위젯 두 개를 “ContentMainPage”의 왼쪽과 오른쪽에 추가한다.

버튼 위젯에 스타일 할당하기

B&R 테마에 미리 정의된 스타일을 “button” 위젯의 속성창에 할당할 수 있다.

해당 “button” 위젯을 선택한 후, 스타일 “Command1”을 좌측 버튼에, “Operate1”을 우측 버튼에 할당한다.

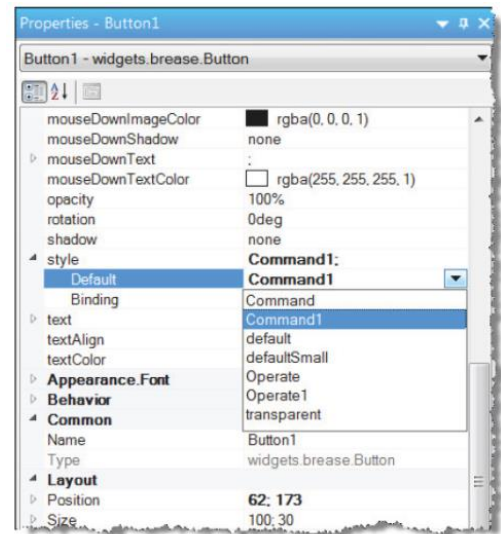


그림 36 속성 창에 서 스타일 선택하기

예제: 버튼 위젯에 스타일 할당하기
스타일 “Command1”을 좌측 버튼에
스타일 “Operate1”을 우측 버튼에
할당한다.

예상되는 결과:



그림 37 Visual editor 에 스타일 할당하기

디폴트 스타일(Default style)

각 위젯은 “디폴트(default)”라 불리는 스타일이 테마에 포함되어 있을 때는 언제나 자동적으로 디폴트 스타일을 사용한다.

디폴트 스타일은 각 위젯에 대한 B&R 테마 패키지에 정의되어 있기 때문에, 사용자가 스타일 속성을 변경하지 않는 한 각 위젯 인스턴스에 자동으로 이용된다. 이러한 방식으로, 스타일은 수동과 자동 내비게이션에 대한 “NavigationButton” 위젯에 할당될 필요가 없다. “스타일링 가능한 속성”에서 스타일링된 내비게이션 버튼 외관은 B&R 테마가 추가될 때 자동적으로 변경된다.

.vis 파일에 B&R 테마 입력하기

HMI 어플리케이션에서 B&R 테마를 이용하려면, 반드시 컨피규레이션 뷰 / .vis 파일에 참조되어 있어야 한다.

```
<StartTheme themeRefId="BuRThemeFlat1" />
<Themes>
  <ThemerefId=" BuRThemeFlat1"/>
</Themes>
```

예상되는 결과

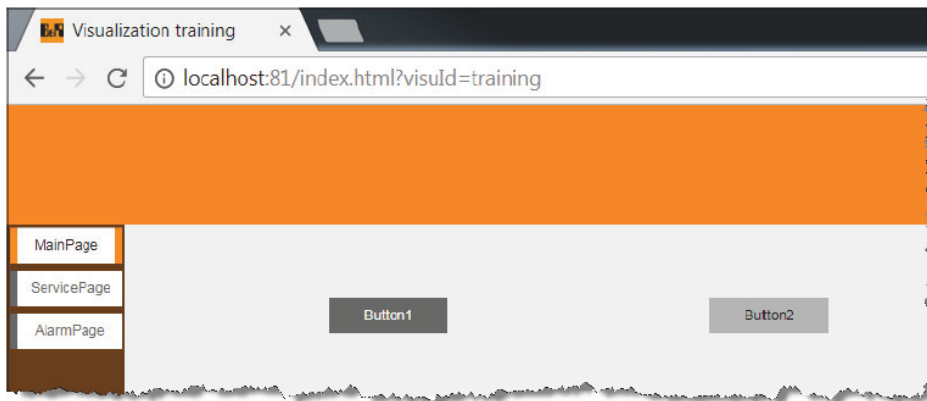


그림 38 B&R 테마를 포함하는 HMI 어플리케이션

4.2.3 디폴트 스타일 변경하기

이번에는 NavigationButton 위젯의 디폴트 스타일을 수정한다. 위젯 폭에 맞춰 글자가 출력되도록 설정한다.

예제: NavigationButton 위젯의 디폴트 스타일

예제의 목표는 B&R theme 에 “NavigationButton” 위젯의 디폴트 스타일을 변경하는 것이다. “NavigationButton” 위젯의 텍스트에 충분한 공간이 있는 방식으로 추가된다.

절차는 다음과 같다:

- 1) 로지컬 뷰 / B&R theme 열기
- 2) 스타일 파일, Default.styles 열기
- 3) 시작 <Style id="default" xsi:type="widgets.brease.NavigationButton"...> 위치 찾기
- 4) 스타일 속성 padding = "0px 5px"
- 5) 프로젝트 컴파일/전송하고 HMI 어플리케이션 표시

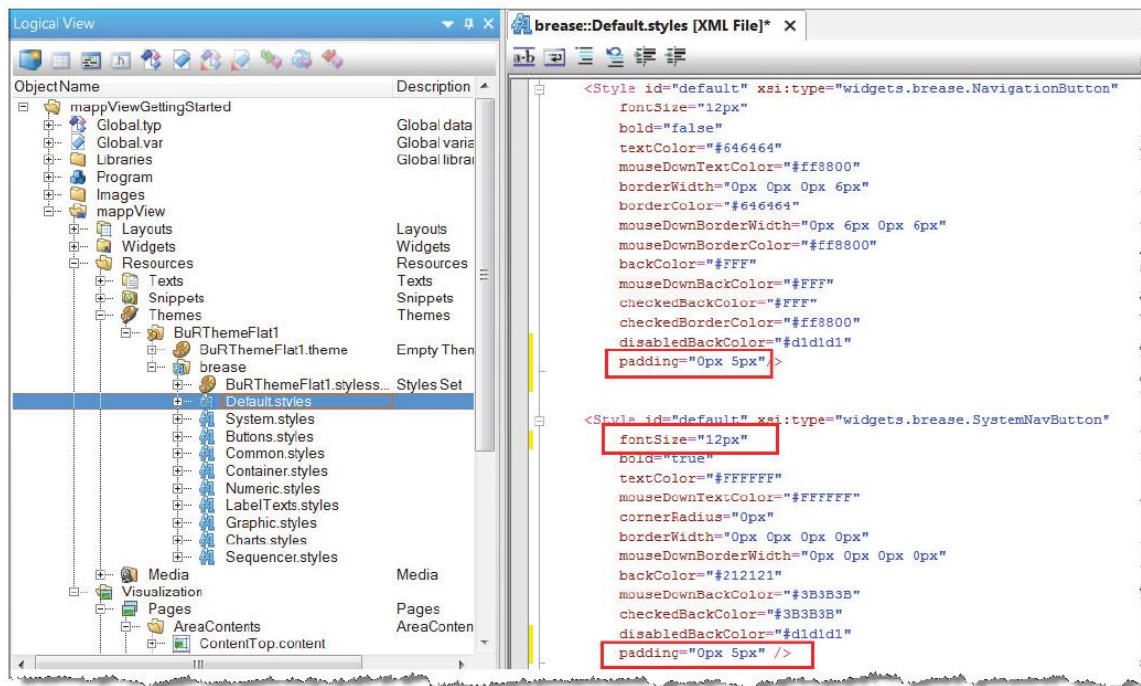


그림 39 NavigationButton 위젯의 디폴트 스타일 조정



스타일이 자동 네비게이션에서 사용되고 있다면, 위젯 타입 “widgets.brease.SystemNavButton” 을 반드시 조정해야한다.

예상되는 결과

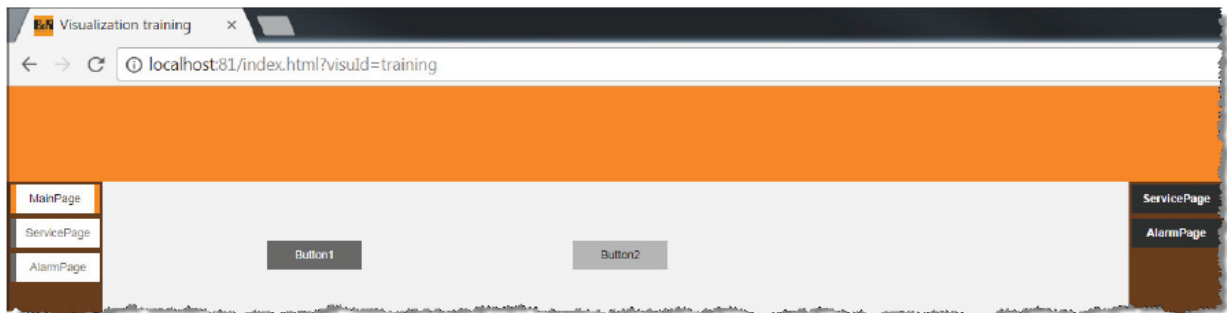


그림 40 디폴트 스타일이 변경된 NavigationButton 위젯

4.2.4 (선택)스타일을 갖는 테마 작성하기

이번 장에서는 앞서 작성된 “button” 위젯에 사용하기 위한 스타일과 맞춤형 테마를 작성 방법을 설명한다.

예제: 스타일을 포함하는 테마 작성하기

이 예제의 목표는 기존 “button” 위젯을 위하여 테마 한 개와 스타일 두 개를 작성하는 것이다. 절차는 다음과 같다:

- 1) 새로운 테마 패키지 추가
- 2) 태블릿, 스마트폰 등과 같은 스마트 장치는 궁극적 사용성을 갖는 강력한 기술의 완벽한 예시로 간주함
- 3) 스타일 파일(style file) 추가
- 4) 버튼 위젯을 위한 스타일을 작성하고 할당
- 5) Visualization.vis 파일에 테마 입력
- 6) 프로젝트를 컴파일/전송하고 HMI 어플리케이션 표시



Visualization / mapp View / Engineering / themes and styles / Creating a theme

새로운 주제 : 패키지 추가하기

첫 단계는 Object Catalog 를 통해 “Empty Theme Package”를 추가하는 것이다. 이 패키지는 기반 스타일 클래스를 포함하여 스타일 집합(StyleSet) 파일을 구축하는 데 이용된다.

Empty Theme Package 는 로지컬 뷰에서 “theme” 패키지를 선택한 후 Object Catalog 부터 추가할 수 있다. 테마 패키지는 .theme 파일을 포함한다.

명확한 구조를 유지하기 위해, 테마 패키지와 포함된 .theme 파일의 명칭을 각각 “MyThemePackage” 및 “MyTheme”으로 변경한다.

사용될 스타일 집합(StyleSet) 파일은 . theme 파일에서 사용되고 참조된다. 또한 Visualization object 에서 테마를 참조하기 위해 사용되는 고유한 ID 는 . theme 파일에 규정되어 있다.

. theme 파일을 더블 클릭하면 XML 편집기에서 파일이 열린다. 그러면 다음 단계에서 설명되는 고유한 < theme ID=“MyTheme”> 및 <StyleSet refId=“MyStyleSet”>를 입력할 수 있다.

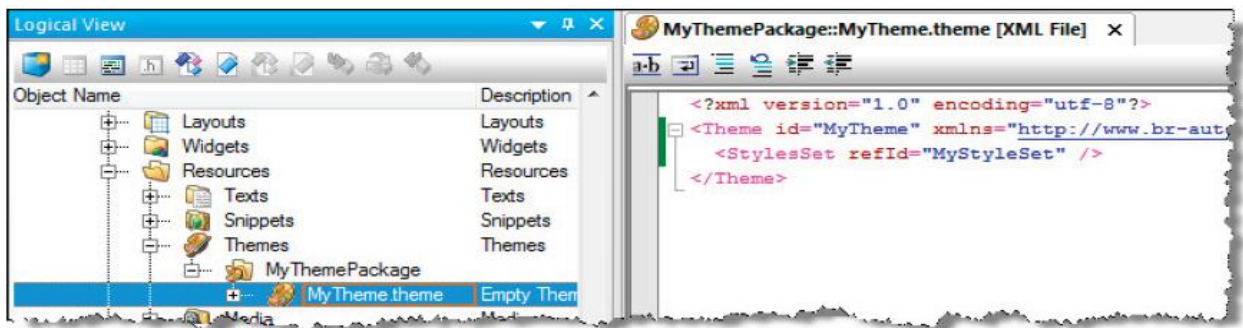


그림 41 테마 패키지가 추가되고 ID 가 할당된 상태



Visualization / mapp View / Engineering / themes and styles / Creating a theme

스타일 집합(StyleSet) 패키지

Object Catalog 를 통해서 스타일 집합 패키지(StyleSet Package)를 “MythemePackage” 패키지에 추가할 수 있다. 스타일 집합 패키지가 추가될 때, .styleset 파일도 함께 만들어진다. 스타일 집합 패키지는 스타일 파일을 구성하는 데 이용된다.

. styleset 파일을 포함하는 스타일 집합 패키지가 추가되고 로지컬 뷰에서 MyStyleSet 로 명칭을 변경한 후, 더블 클릭하여 XML 편집기를 열 수 있다.

먼저 입력해야 할 것은 .테마 파일에 할당된 고유한 <StyleSet id=“MyStyleSet”>이다.

이때 다음 단계에서 참조되는 스타일의 <Style refId>가 참조된다.

기존 “버튼” 위젯에 대해 스타일 2 가지를 작성한 후, “MyButtonStyle”은 “refId”로 할당된다.

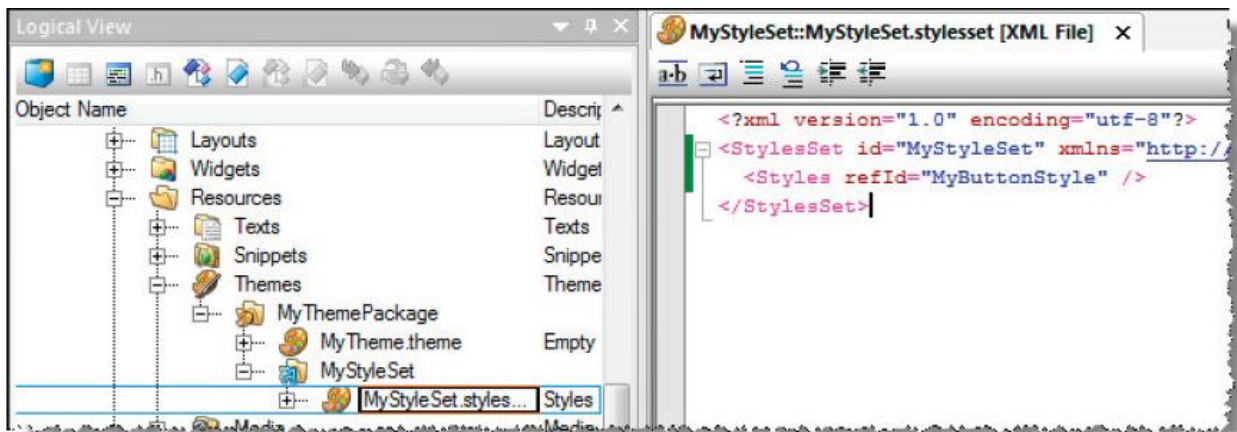


그림 42 StyleSet Package 가 추가되고 ID 가 할당된 상태



Visualization / mapp View / Engineering / themes and styles / Creating a theme

스타일 파일(Style file)

로지컬 뷰에서 MyStyleSet 패키지를 선택한 후, Object Catalog 에서 스타일 파일을 추가할 수 있다. 그 다음, 스타일 집합 파일에 규정된 바와 같이 “MyButtonStyle”로 이름을 바꿀 수 있다.

고유 ID 를 입력한다: <Style id=“MyButtonStyle”>.

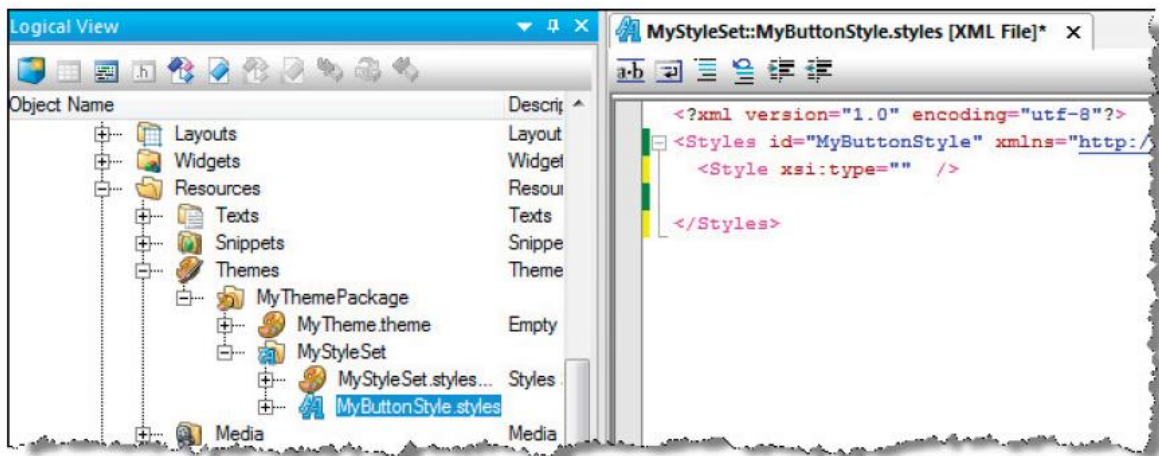


그림 43 스타일 파일이 추가되고 ID 가 할당된 상태



Visualization / mapp View / Engineering / themes and styles / Creating a theme



위젯에 속하는 모든 스타일 가능한 속성에 관한 정보는 Automation Studio 도움말에서 확인할 수 있다.

버튼 위젯에 대한 스타일을 작성하고 스타일을 위젯 인스턴스에 할당하기

다음 단계는 “Button” 위젯을 위한 2 가지 스타일을 (스타일 “Command2” 및 스타일 “Operate2”) 작성하는 것이다.

이 스타일은 아래와 같은 스타일링 가능한 속성을 설정한다.

Style “ID”	sxi:type=	backColor=	textColor=	border-Style=	border-Width=	cornerRadius=
“Command2”	“widgets.brease.Button”	“#FF0000”	“#000000”	“solid”	“4px”	“10px”
“Operate2”	“widgets.brease.Button”	“#00FF00”	“#000000”	“solid”	“4px”	“10px”

표 9 버튼 스타일 “Command2” 및 “Operate2”의 속성

예상되는 결과:

```
<?xml version="1.0" encoding="utf-8"?>
<Styles id="MyButtonStyle"
xmlns="http://www.br-automation.com/iat2015/styles/engineering/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Style xsi:type="widgets.brease.Button" id="Command2"
    backColor="#FF0000"
    textColor="#000000"
    borderStyle="solid"
    borderWidth="4px"
    cornerRadius="10px"
  />
```

```

<Style xsi:type="widgets.brease.Button" id="Operate2"
    backColor="#00FF00"
    textColor="#000000"
    borderStyle="solid"
    borderWidth="4px"
    cornerRadius="10px"
/>
</Styles>

```

.vis 파일에 테마입력하기

새로 작성된 테마가 런타임중에 사용하려면, 반드시 컨피규레이션 뷰 / .vis 파일에 참조되어야 한다.

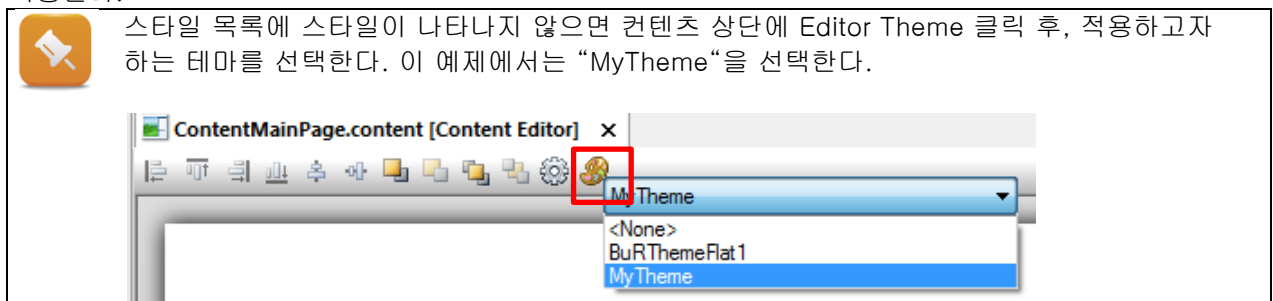
```

<Start theme theme RefId="Mytheme" />
< themes>
    < theme refId=" BuRThemeFlat1"/>
    < theme refId="Mytheme"/>
</ themes>

```

메인페이지 버튼에 스타일 적용하기

메인페이지에 좌측에 있는 버튼 style 에는 “Command2”를 우측에 있는 버튼 style 에는 “Operate2”를 적용한다.



프로젝트를 컴파일,전송하고 HMI 어플리케이션 표시하기

프로젝트를 컴파일하고 ARsim 에 전송할 수 있다.

예상되는 결과

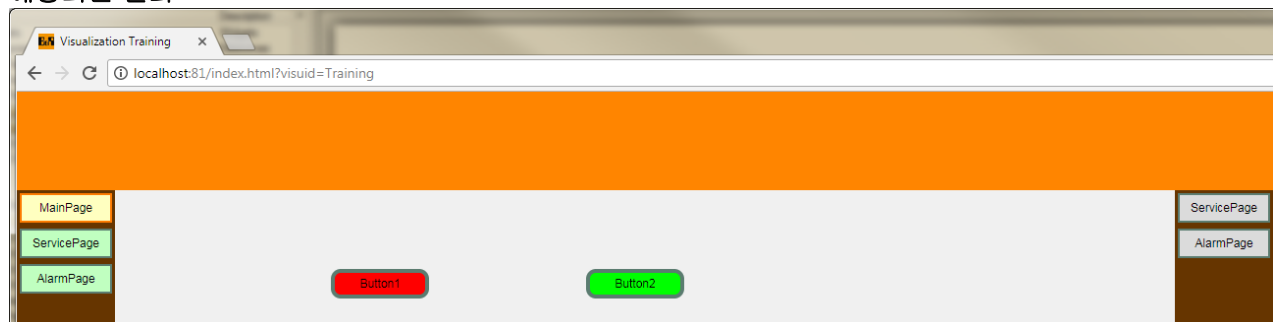


그림 44 테마에 포함되어 있는 버튼 스타일을 적용한 HMI 어플리케이션

5 데이터 바인딩(Data binding)

5.1 데이터 바인딩(Data binding) 개요

5.1.1 OPC UA

OPC Unified Architecture (OPC UA)는 산업 자동화와 그 밖의 산업 분야에서 안정되고 신뢰성 있는 데이터 교환을 보장하기 위한 상호운용성 표준이다. OPC UA 는 독립적 플랫폼이며 각각 다른 공급자의 장치 사이에서 빈틈없이 데이터를 교환하는 데 이용될 수 있다.

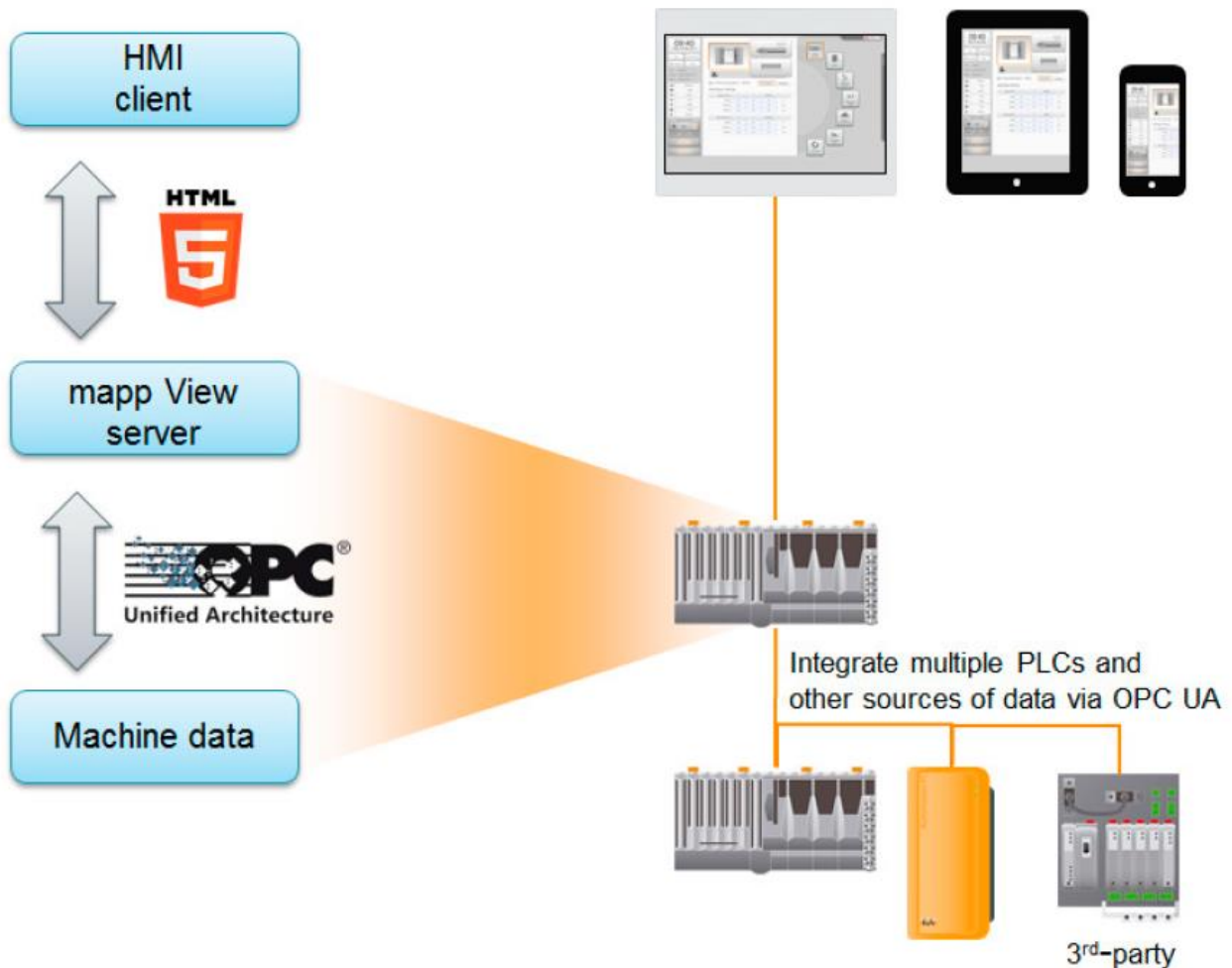


그림 45 OPC UA 구조

OPC UA 노드(OPC UA node)

자동화 어플리케이션의 프로세스 변수를 OPC UA 시스템을 통해 이용할 수 있다면, OPC UA 노드로써 OPC UA 클라이언트로부터 읽고 쓰기가 가능하다. 엔지니어링 단위(engineering unit)나 EU 범위(EU range)와 같은 속성을 OPC UA 노드에 추가할 수 있다.

엔지니어링 단위(Engineering unit)

OPC UA 노드에 대한 엔지니어링 단위(engineering unit)는 값을 해석할 때 사용되는 물리적 단위를 규정한다. Automation Studio 는 7 개 물리량에 대한 1,400 가지의 엔지니어링 단위를 제공하며, 여기에는 길이, 질량, 시간, 전류, 온도, 물질의 양, 광도와 함께 유도된 물리량이 (예, 속도, 힘, 압력, 가속도) 포함된다.

EU 범위(EU range)

EU 범위 (엔지니어링 단위 범위)는 OPC UA 노드 값의 유효한 범위를 결정한다. EU 범위는 하한과 상한 값에 의해 정의된다.

OPC UA 디폴트 뷰(OPC UA default view)

OPC UA 디폴트 뷰는 OPC UA 서버에 의해 B&R 컨트롤러에서 OPC UA 클라이언트에게 제공되는 자동화 어플리케이션의 모든 프로세스 변수를 포함한다.

5.1.2 바인딩(Binding)

HMI 어플리케이션은 사람과 기계 사이의 상호작용을 용이하게 한다. HMI 어플리케이션은 반드시 자동화 어플리케이션의 데이터 (현재의 센서 값, 현재의 기계 상태 등)를 표시할 수 있어야 한다.

또한 (예를 들어, 새로운 설정 값을 입력하기 위해) 기계 작업자가 자동화 어플리케이션에 개입하는 것도 가능해야 한다. 이러한 이유로, HMI 어플리케이션의 요소들 (예, 위젯 인스턴스)은 HMI 어플리케이션과 자동화 어플리케이션 사이에서 데이터가 교환될 수 있도록, 반드시 자동화 어플리케이션의 요소들과 연결되어야 한다.

mapp View 에서 이러한 연결은 바인딩(binding) (데이터 바인딩)이라 알려진 기능을 이용하여 구현된다. mapp View HMI 어플리케이션의 바인딩은 어느 데이터 소스가 (자동화 어플리케이션의 OPC UA 노드) 어느 HMI 요소 (예, 위젯 인스턴스), 바인딩 유형 및 바인딩 모드와 (데이터가 교환되는 방식) 연결되는지를 정의한다.

mapp View 는 다음과 같은 바인딩 유형을 구별한다:

- 단일 값 바인딩에 대한 “값 바인딩(Value binding)”
- 단위와 한도를 갖는 바인딩 OPC UA 노드를 위한 “노드 바인딩(Node binding)”
- 배열에 대한 “배열 바인딩(Array binding)”
- 목록에서 변수를 선택하기 위한 “목록 바인딩(List binding)”
- 바인딩 구조에 대한 “복합 바인딩(Complex binding)”

이 세미나에는 “값 바인딩(Value binding)” 및 “노드 바인딩(Node binding)” 유형만 포함된다. 다른 모든 유형들은 mapp View 고급 과정에 포함된다.

바인딩 모드(Binding modes)

바인딩 모드를 규정하는 것은 데이터가 흐르는 방향을 정의하는 것이다.

바인딩 모드 “oneWay” (읽기 전용(Read only))

“oneWay” 바인딩 모드는 소스에 대한 읽기 접근에 이용된다.

예시: OPC UA 노드와 출력 위젯 사이의 바인딩 (예, NumericOutput 위젯)

바인딩 모드 “twoWay” (읽기/쓰기(Read/Write))

“twoWay” 바인딩 모드는 소스에 대한 읽기와 쓰기 접근에 이용된다.

예시: OPC UA 노드와 입력 위젯 사이의 바인딩 (예, NumericInput 위젯)

바인딩 모드 “oneWayToSource” (초기 읽기/쓰기(Init Read/Write))

“oneWayToSource” 바인딩 모드는 소스에 대한 쓰기 접근에 이용된다.

예시: OPC UA 노드와 PushButton 위젯 사이의 바인딩.

5.2 데이터 바인딩 이용하기

다음 예제의 목표는 다양한 바인딩 유형과 바인딩 모드를 이용하여 HMI 어플리케이션에서 자동화 어플리케이션의 프로세스 변수 값을 표시하는 것이다.

5.2.1 HMI 어플리케이션에서 데이터 표시하기

이 예제의 목표는 HMI 어플리케이션에서 “현재 온도(Current temperature)” OPC UA 노드의 값을 표시하는 것이다.

예제: NumericOutput 위젯을 이용하여 현재 온도(CurrentTemperature) 변수 값 표시하기

이 목표를 달성하기 위해, 값을 표시하는 데 “NumericOutput” 위젯과 “읽기 전용(Read only)” 바인딩 모드가 이용된다.

여기에서 반드시 다음 단계들이 수행되어야 한다:

- 1) 프로그램과 변수
- 2) OPC UA 서버 활성화
- 3) OPC UA 디폴트 뷰를 추가하고 OPC UA 노드 활성화
- 4) Object Catalog로부터 바인딩 파일 추가
- 5) 수치 출력(NumericOutput) 위젯을 추가하고 OPC UA 노드 할당
- 6) 바인딩 집합(BindingSet) ID를 정의하고 .vis 파일에 입력
- 7) 프로젝트를 컴파일 및 전송하고, HMI 어플리케이션에 표시

프로그램과 변수

“NumericOutput” 위젯에 값을 표시하기 위한 프로그램과 글로벌 변수 (CurrentTemperature)는 이미 존재한다.

다른 프로그램이나 추가적인 변수를 만들 필요는 없다.

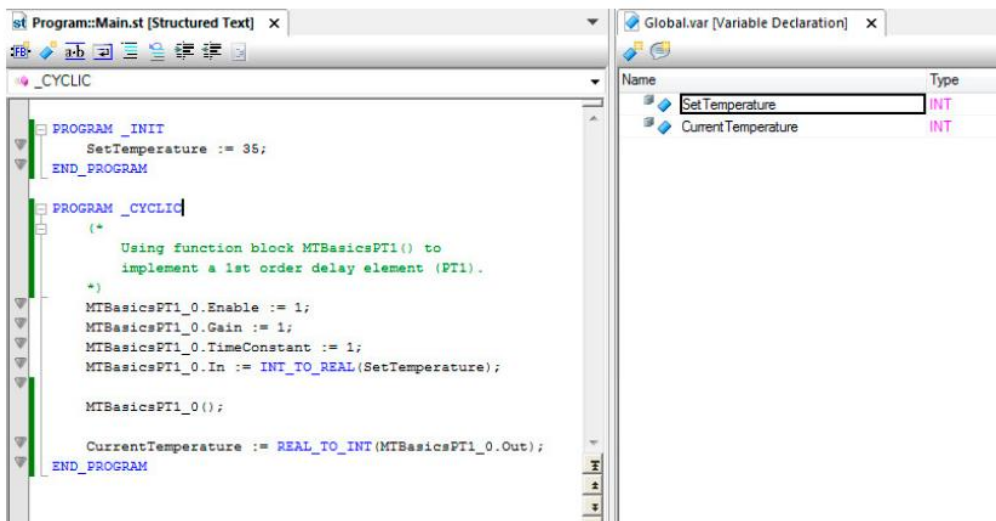


그림 46 글로벌 변수 CurrentTemperature

OPC UA 서버 활성화

자동화 어플리케이션과 HMI
어플리케이션 사이에서 통신이
이루어지려면, 반드시 OPC UA 서버가
활성화되어야 한다.
OPC UA 서버는 OPC UA 시스템 노드의
Physical View / CPU 설정에서
활성화한다.



Communication / OPC UA /
Configuration in AS / Activate
OPC UA system

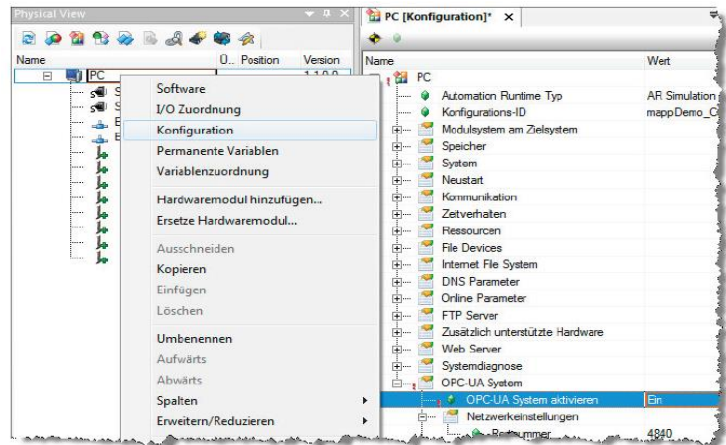


그림 47 OPC UA 서버 활성화

OPC UA default view 추가하고 OPC UA 노드 활성화하기

만들어진 변수(process variables)를
사용하려면, 반드시 OPC UA 노드에서
선언되어야 한다. 이를 수행하기
위해서는 Object Catalog 에서 드래그 앤
드롭을 이용하여 OPC UA default view 를
Configuration View, “Connectivi-
ty/OpCUA” 노드에 추가한다.



Communication / OPC UA /
Configuration in AS / OPC UA
default view configuration /
OPC UA default view editor

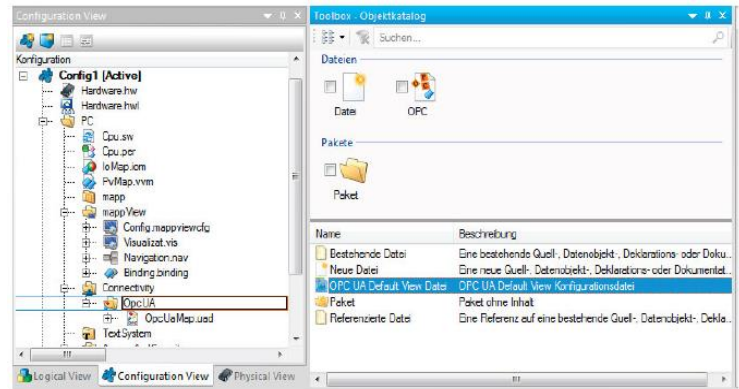


그림 48 Object Catalog로부터 OPC UA 디폴트 뷰로 추가한다.

그러면 글로벌 변수를 OPC UA 노드로
활성화 시킬 수 있다.

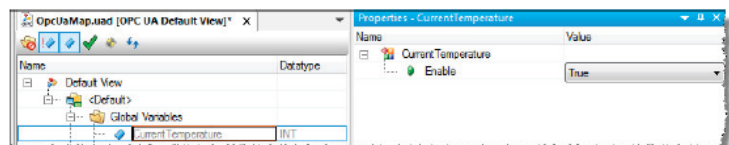


그림 49 OPC UA 디폴트 뷰에서 글로벌 변수 활성화하기

Object Catalog 에서 바인딩 파일을 추가하고 ID 설정하기

“변수 선택 대화 상자(Select Variable dialog box)”를 이용하여 작성된 모든 바인딩은 시스템에 의해
자동적으로 바인딩 파일에 입력된다.

바인딩 파일은 mapp View 노드를 선택함으로써 Object Catalog로부터 추가될 수 있다.

시스템의 바인딩에 어느 바인딩 파일을 입력해야 하는지를 알 수 있도록, 반드시 바인딩 파일에 고유한 “ID”가 정의되어야 한다.

“MyBinding”을 고유 “ID”로 입력하도록 한다.

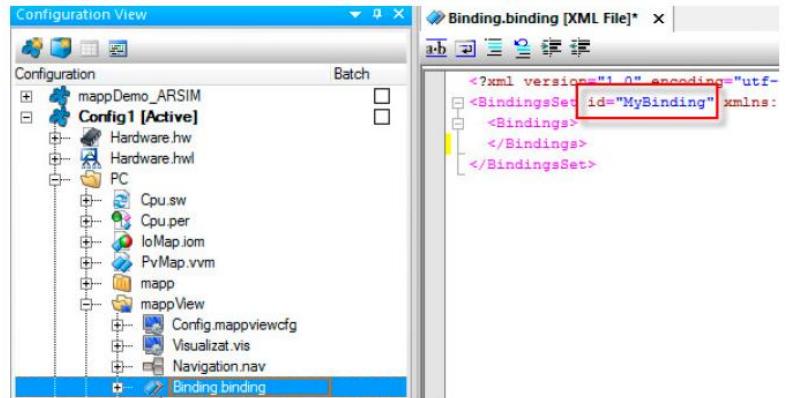


그림 50 바인딩 파일이 추가되고 ID 가 할당된 상태

수치 출력(NumericOutput) 위젯을 추가하고 OPC UA 노드에 할당하기

OPC UA 노드 값이 HMI 어플리케이션에 표시되려면, 반드시 “NumericOutput” 위젯이 “MainPage” 콘텐츠에 추가되어야 한다.

수치 값이 단위나 한도 없이 필요한 경우에만 바인딩이 이용된다. 데이터 전달에는 바인딩된 변수 값만 관련된다.

“ContentMainPage”를 더블 클릭하면 Visual editor 가 열리고 Object Catalog 로부터 “NumericOutput” 위젯을 배치할 수 있다.

“NumericOutput” 위젯을 선택함으로써, “Select Variable dialog box”가 “Data/Value/Binding” 의 속성 창에 열릴 수 있다. 그러면 OPC UA 노드가 대화 상자에서 선택된다.

이것은 “NumericOutput” 위젯이기 때문에, “읽기 전용(Read-only)” 바인딩 모드가 이용된다. “MyBinding”이 바인딩 집합으로 선택된 것은 작성된 바인딩이 여기에 저장되기 때문이다.

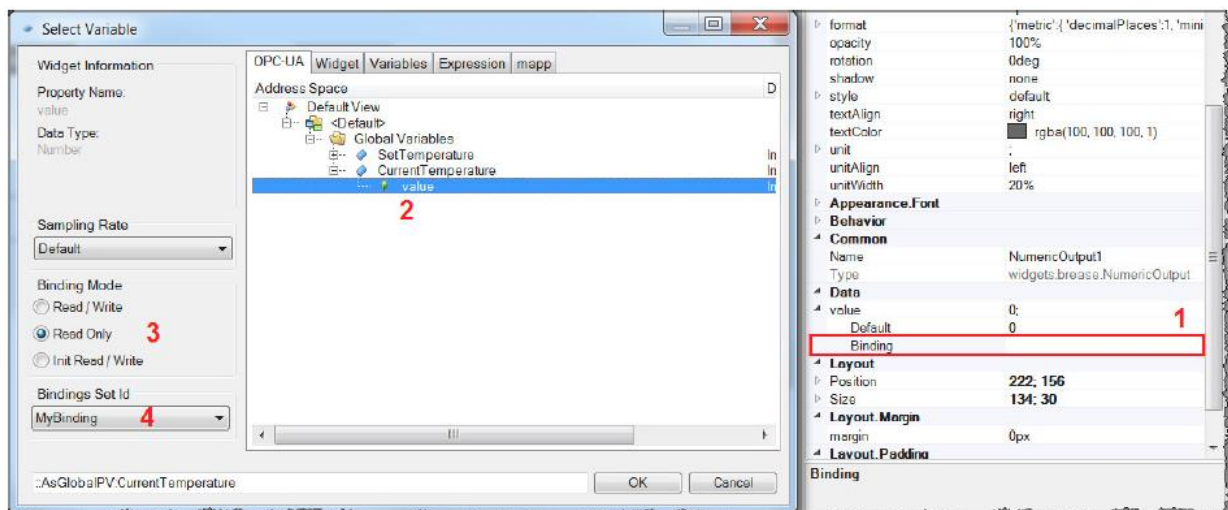


그림 51 바인딩 대화 상자를 열고 속성을 선택하기

.vis 파일에서 바인딩 파일 참조하기

작성된 바인딩 파일이 반영되도록 하려면, 반드시 파일에 대한 고유 “ID”를 (“MyBinding”) .vis 파일에 입력해야 한다.

< BindingSets>에, BindingSet 섹션을 코멘트하고 고유 “ID”를 (“MyBinding”) 입력한다.

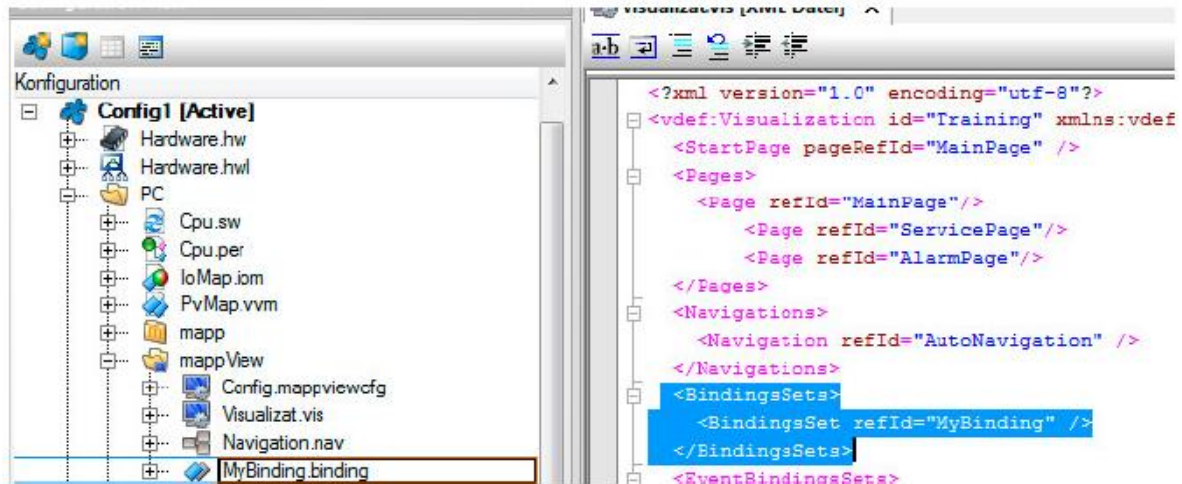


그림 52 .vis 파일에서 바인딩 파일 참조하기

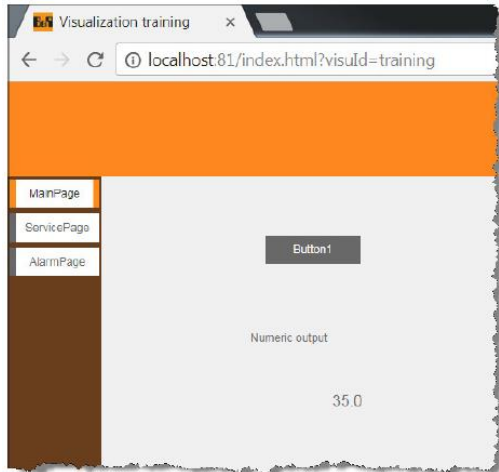
예상되는 결과:

그림 53 NumericOutput 위젯에 값 표시된 어플리케이션

5.2.2 설정 값(Setpoint) 입력

이 예제의 목표는 사용자가 HMI 어플리케이션에 입력할 때 OPC UA 노드의 설정 값 (온도 설정(SetTemperature))을 변경하는 것이다.

예제: NumericInput 위젯을 이용하여 설정 값 변경하기

이 목표를 달성하기 위해 numericInput 위젯을 “MainPage” 콘텐츠에 배치하고 바인딩을 통해 자동화 어플리케이션의 대응되는 OPC UA 노드에 연결하는 것이다. HMI 어플리케이션에서 변경된 값이 쓰이도록 하려면, TwoWay (“읽기/쓰기”) 바인딩 모드가 선택되어야 한다. 여기에서 반드시 다음 단계를 수행해야 한다.

- 1) 프로그램과 프로세스 변수
- 2) 변수를 OPC UA 노드 활성화
- 3) NumericInput 위젯을 추가하고 OPC UA 노드 할당
- 4) 프로젝트를 컴파일/전송하고, HMI 어플리케이션에 표시

프로그램과 변수

“NumericOutput” 위젯에 값을 표시하기 위한 프로그램과 글로벌 변수(CurrentTemperature, SetTemperature)는 이미 존재한다.

다른 프로그램이나 추가적인 변수를 만들 필요가 없다.

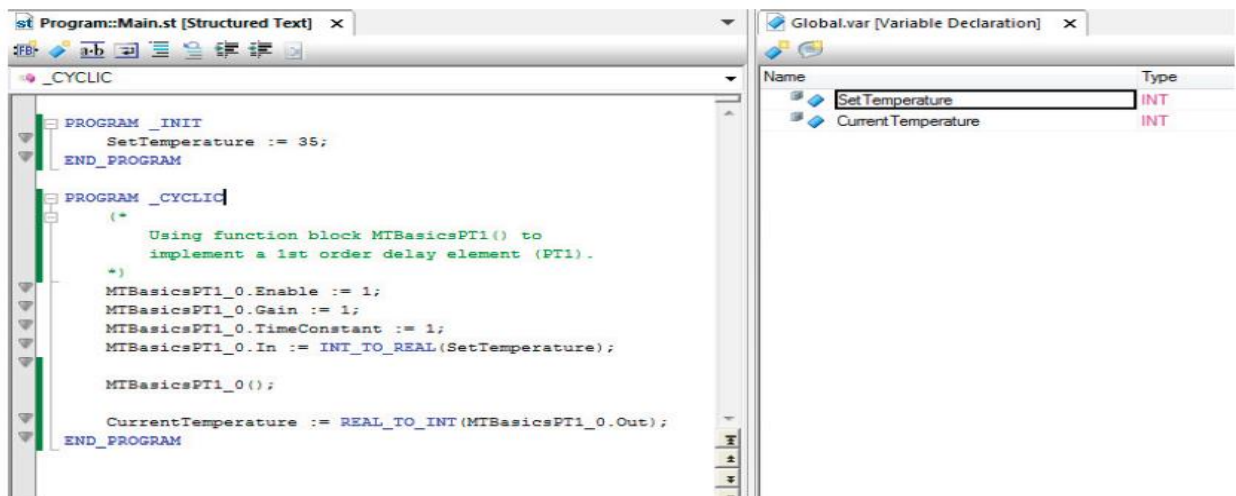


그림 54 글로벌 변수 CurrentTemperature 및 SetTemperature

예제: OPC UA 노드로 변수 활성화하기

OPC UA 서버는 이미 활성화 되어 있고 OPC UA default view 는 단 한 번만 추가될 수 있기 때문에, 변수 (SetTemperature)는 OPC UA 디폴트 뷰에서 즉시 활성화 될 수 있다.

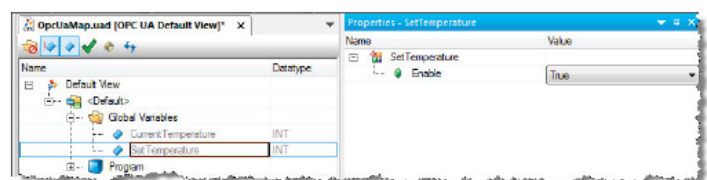


그림 55 OPC UA default view 에서 SetpTemperature 변수 활성화하기

- 1) OPC UA node 노드로써 변수를 활성화한다.

수치 입력(numeric input)을 추가하고 변수 할당하기

모든 바인딩은 동일한 파일에 저장되기 때문에, 다른 바인딩 파일을 만들 필요가 없다. 그러므로 이제 “NumericInput” 위젯을 “ContentMainPage”에 추가할 수 있다.

“NumericInput” 위젯과 상호작용할 필요가 있고 OPC UA 노드 값은 사용자에게 의해 설정 가능해야 하기 때문에, “읽기/쓰기”는 반드시 바인딩 모드로써 선택되어야 한다. 이로써 사용자가 값을 변경하는 것이 가능하다.

예제: 수치 입력(NumericInput) 위젯을 추가하고 설정하기

- 1) NumericInput 위젯을 ContentMainPage 에 추가
- 2) NumericInput 을 온도 설정(SetTemperature) OPC UA 노드에 할당
- 3) “MyBinding”을 바인딩 집합 ID 로써 선택
- 4) Read/Write 를 바인딩 모드로 사용

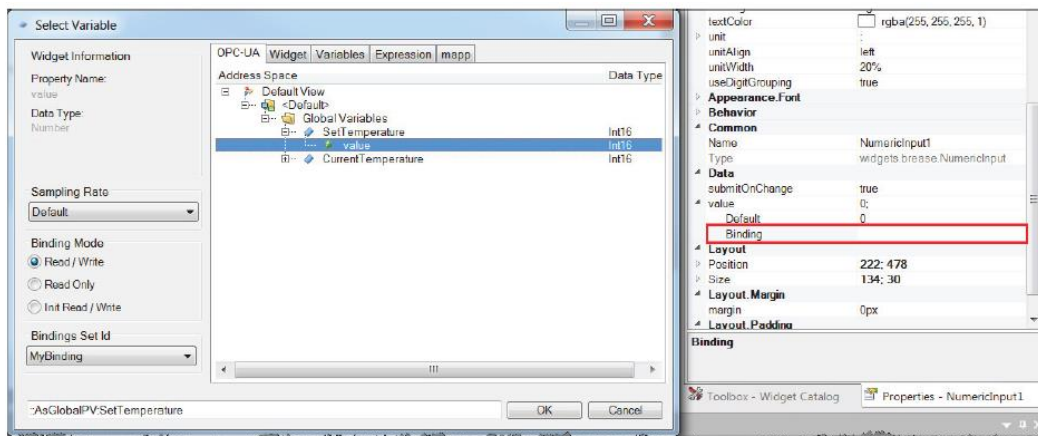


그림 56 바인딩 대화 상자에서, numeric input 을 SetTemperature 의 OPC UA 노드에 연결하기

예상되는 결과

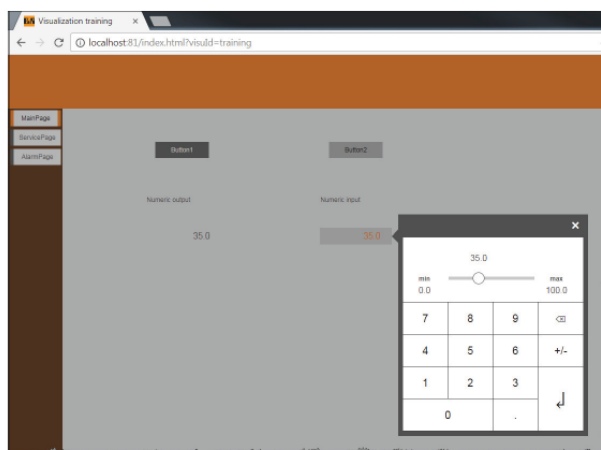


그림 57 Numericinput 위젯을 사용하여 브라우저에 HMI 어플리케이션 표시하기

5.2.3 값 범위 규정하기

지금까지는 온도 설정(SetTemperature)을 어느 값으로든 설정할 수 있다. 이것은 언제나 유용한 것은 아니다. 사용자가 임의로 혹은 허용 불가능한 값을 입력하는 것을 방지하기 위해, OPC UA 노드에 대한 EU 범위 (하한/상한 값)를 정의할 수 있다. 그러므로 EU 범위를 정의하는 것은 사용자가 허용 할 수 없는 값을 입력하는 것을 불가능하게 한다.

예제: 수치 입력(NumericInput) 범위를 제한하기

이 예제의 목표는 사용자가 허용할 수 없는 값을 입력하지 못하도록 하는 것이다.

이 목표를 달성하기 위해, OPC UA 노드에 한도가 할당된다.

반드시 다음 단계를 수행해야 한다:

- 1) OPC UA 노드에 대한 하한/상한 값 정의
- 2) 바인딩을 “노드 바인딩(NodeBinding)”으로 변경
- 3) 프로젝트를 컴파일하고, 전송하고, HMI 어플리케이션에 표시

OPC UA 노드에 대한 하한/상한 값 정의하기

OPC UA default view 를 더블 클릭하고 속성 창을 업데이트하기 위해 OPC UA 노드 (SetTemperature)를 선택한다.

그 다음 “EU 범위(EU range)” 노드에 노드의 하한과 상한 값을 설정할 수 있다.

EU range	Value
Low	25
High	50

표 10 SetTemperature OPC UA 노드에 대한 EU 범위 속성

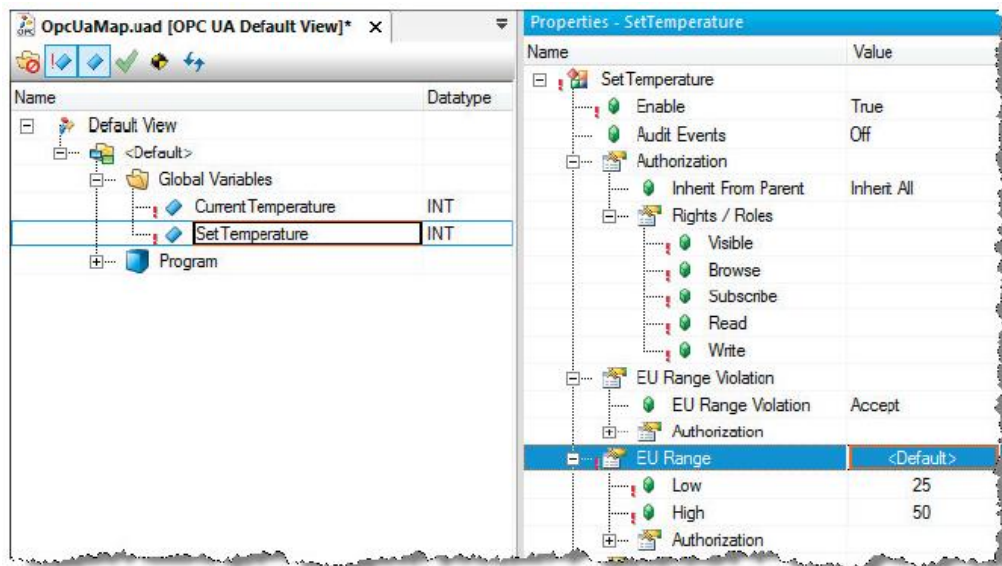


그림 58 SetTemperature OPC UA 노드에 대한 EU 범위 정의하기



Communication / OPC UA / Configuration in AS / OPC UA Default view configuration / OPC UA Default view editor / Properties of OPC UA tags / Range of values / Role-based range of values

바인딩을 노드 바인딩(Node-Binding)으로 변경하기

OPC UA 노드의 EU 범위를 보장하기 위해, 반드시 바인딩을 값 바인딩으로부터 노드 바인딩으로 변경해야 한다.

노드 바인딩은 전체 OPC UA 노드 (“SetTemperature”)를 선택함으로써 설정된다.

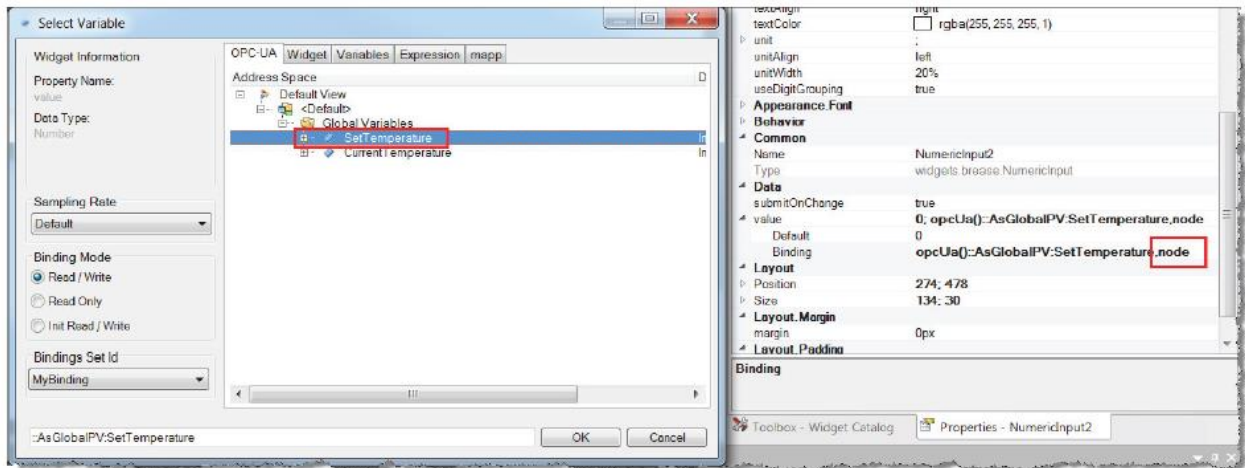


그림 59 NumericInput 을 SetTemperature OPC UA 노드에 할당하기

이제 사용자는 EU 범위에 정의된 것보다 더 크거나 작은 값을 입력할 수 없게 되었다.

예상되는 결과

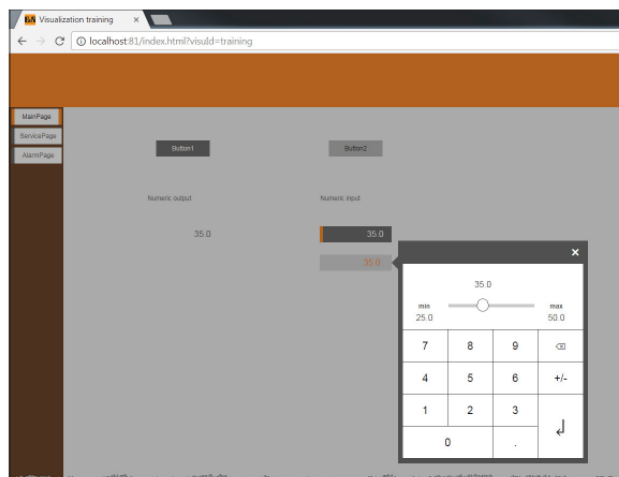


그림 60 SetTemperature OPC UA 노드에 대한 정의된 EU 범위를 갖는 HMI 어플리케이션을 브라우저에 표시하기

6 미디어 파일(Media files)

6.1 개요 – 미디어 파일

HMI 어플리케이션의 외관을 향상시키기 위해 mapp View 에 미디어 파일을 추가할 수 있다.

6.2 미디어 파일 추가하기

6.2.1 이미지 삽입하기

새로운 mapp View 프로젝트에는 미디어 패키지가 자동적으로 제공되고 HMI 어플리케이션에 이용되는 추가적인 이미지 파일을 논리적으로 관리하는데 사용된다. 이것은 패키지를 이용하여 구성될 수 있다.

.png, .jpg, .svg 등과 같은 모든 일반적 이미지 포맷을 이용할 수 있다.

이 폴더 내의 모든 파일은 타겟 시스템으로 전송된다.

참조 이미지 경로는 언제나 “Media” 패키지로 시작되고, 이미지 또는 하위 패키지의 파일 명칭과 이미지 파일 이름이 뒤따른다.

미디어 패키지에 직접 저장된 이미지 “test.png”에 대한 예시:

URL = “Media/Test.png”

미디어 폴더의 “SmallPictures” 하위 패키지에 저장된 이미지 “Test.png”에 대한 예시:

URL = “Media/SmallPictures/Test.png”



HMI application / mapp View / Engineering / HMI organization / Logical View / mapp View
Media package

예제: 이미지 삽입하기

이 예제의 목표는 모든 페이지 상에 회사 로고를 표시하는 것이다.

회사 로고는 “이미지(Image)” 위젯을 이용하여 “ContentRight”의 우측 하단에 추가해야 한다.

이를 위해 다음 단계들이 필요하다:

- 1) 미디어 폴더에 이미지 저장
- 2) 이미지 위젯을 추가하고 필요에 따라 배치
- 3) 위젯에 이미지를 설정
- 4) 프로젝트를 컴파일/전송하고, HMI 어플리케이션에 표시

미디어 폴더에 이미지 저장하기

드래그 앤 드롭을 이용하여, 이미지를 손쉽게 로지컬 뷰의 미디어 패키지에 추가할 수 있다. 준비된 프로젝트에서, B&R 로고는 이미 mapp View 패키지 외부의 이미지 패키지에 저장되어 있고 이 패키지를 mapp View 미디어 패키지로 이동시키기만 하면 된다.

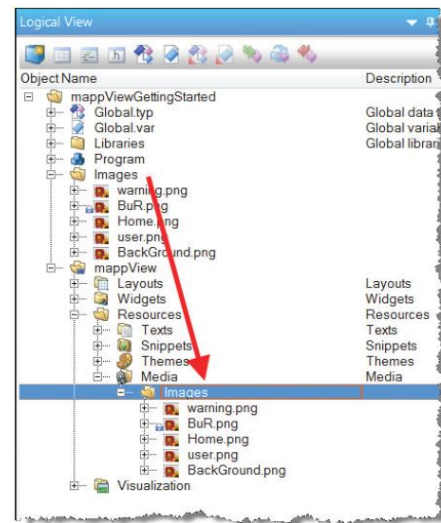


그림 61 미디어 패키지에 이미지 저장하기

이미지 위젯을 추가, 배치, 설정하기

콘텐츠를 선택한 후, Object Catalog로부터 이미지 위젯을 추가할 수 있다.

그 다음 속성 창에서 위젯의 크기와 위치를 정의한다.

“Layout/..”에서 위젯 값을 아래와 같이 설정할 수 있다.

Name	Value(px)
Position “top”	640
Position “left”	0
Size “width”	100
Size “height”	60

표 11 이미지 위젯 속성

위젯에서 이미지 설정하기

위젯에 추가된 B&R 로고를 참조하려면, 반드시 “Appearance/image”에 이미지의 위치와 명칭을 입력해야 한다.

이미지는 “Media” 패키지에 직접 저장되었고 “BuR.png”라고 명명되었기 때문에, 경로는 다음과 같다.

“Media/Images/BuR.png”

“이미지(Image)”의 크기는 이미 조정되었고 디폴트 속성 세팅은 “size-Mode”=“contain”이기 때문에, “이미지” 위젯은 이미지에 의해 완전히 채워진다. 다른 “크기 모드(sizeMode)” 속성에 관한 정보는 도움말 시스템에서 찾아볼 수 있다.

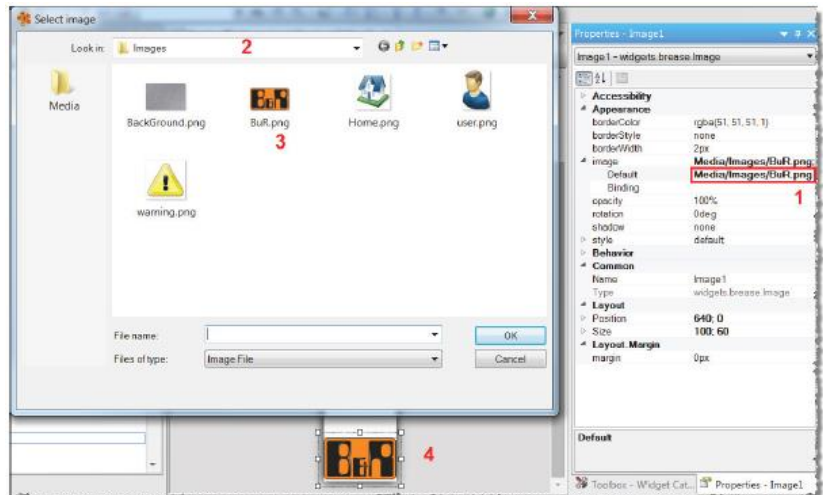


그림 62 이미지 위젯 속성



예상되는 결과:

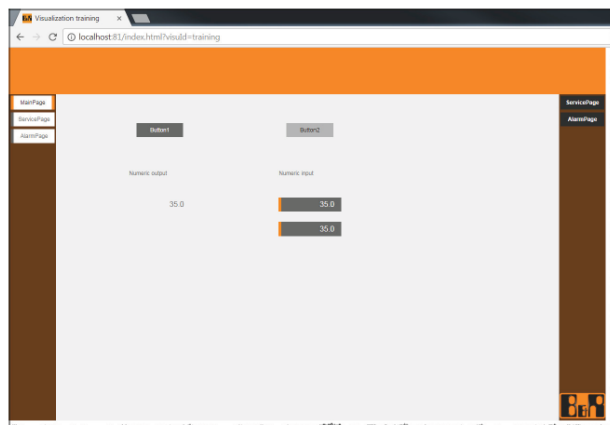


그림 63 브라우저에 B&R 로고가 추가된 HMI 어플리케이션 표시하기

6.2.2 내비게이션 버튼(NavigationButton) 위젯의 이미지

이미지는 “Button” 위젯에도 표시될 수 있다. “MainPage” 에서 “NavigationButton”의 내비게이션을 위한 “Home” 아이콘을 구현하기 위해서 이러한 기능이 다음 예제에서 이용된다.

예제: 내비게이션 버튼(NavigationButton)에 홈(Home) 이미지 설정하기

이 예제의 목표는 “MainPage”에서 “Navigation”을 위한 정상적인 “NavigationButton” 위젯을 사용하는 대신에 “NavigationButton”에 아이콘을 구현하는 것이다.

“NavigationButton”을 각 페이지에 개별적으로 배치하지 않도록 “NavigationButton”을 ContentLeft 에 배치하고, 그것을 각 페이지에서 참조한다.

반드시 다음 단계를 수행해야 한다:

- 1) NavigationButton 위젯을 추가하고 설정하기
- 2) 내비게이션 버튼 상에 아이콘 설정
- 3) 프로젝트를 컴파일, 전송하고, HMI 어플리케이션에 표시

내비게이션 버튼(NavigationButton) 위젯을 추가하고 설정하기

먼저, Object Catalog로부터 내비게이션 버튼을 ContentLeft의 하단에 드래그 앤 드롭으로 추가한다. 다음에는 “NavigationButton” 위젯의 크기를 변경하고 “NavigationButton”을 콘텐츠의 하단 중간에 배치한다.

이미지를 “NavigationButton” 위젯에 배치해야 하기 때문에, 디폴트 텍스트는 삭제할 수 있다.

“NavigationButton” 위젯의 “Data/pageld” 속성에는 클릭할 때 이동해야 하는 페이지로 MainPage를 선택한다.

Property name	Value
Layout/size	48;48
Position/top/left	452;26
Data/pageld	MainPage
Appearance/text	
Appearance/style	Transparent

표 12 내비게이션 버튼 위젯 속성

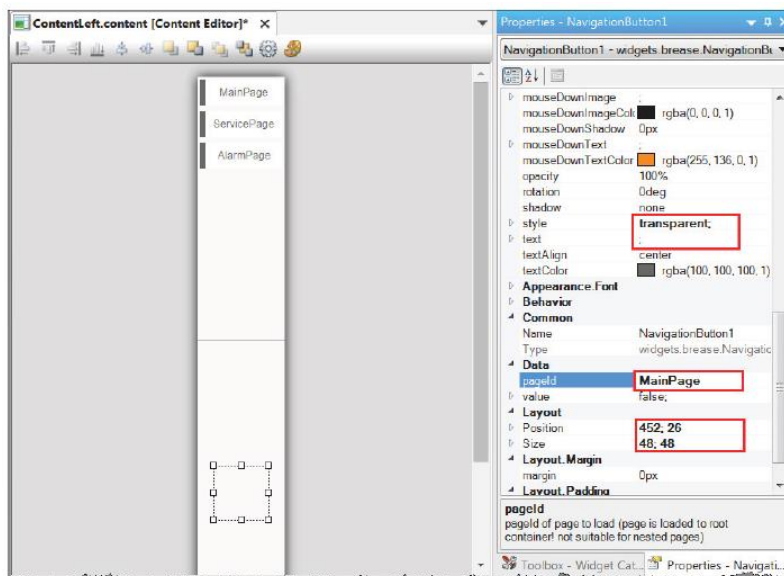


그림 64 추가된 위치에 배정된 내비게이션 버튼

내비게이션 버튼(NavigationButton) 위젯에서 이미지 참조하기

“NavigationButton”에 사용되는 이미지는 미디어(Media) 패키지에 저장되어 있다. 이전 예제를 통해서 드래그-앤드-드롭으로 미디어 폴더에 이미지가 추가되어 있다.

“Image” 위젯의 경우, “NavigationButton”은 이미지 경로를 규정하는 데 이용되는 “image” 속성이 있다.

“ImageAlign” 속성은 추가된 이미지가 올바르게 배치되어 있도록 설정하는데 이용된다.

이미지는 클리핑 후에, 즉 격리되어 있을 때 더 잘 보이고 배경 색상과 프레임이 없기 때문에, “backgroundColor” 및 “borderStyle” 속성을 변경한다.

다음 속성을 정의해야 한다:

Property name	Value
Appearance/image	Media/Image/Home.png

표 13 내비게이션 버튼 위젯 속성

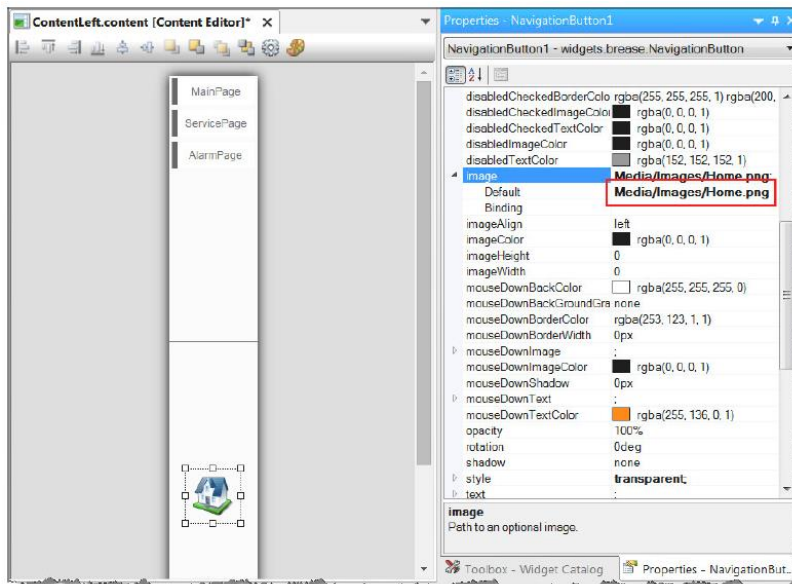


그림 65 내비게이션 버튼 상에서 홈(Home) 아이콘 참조하기

예상되는 결과

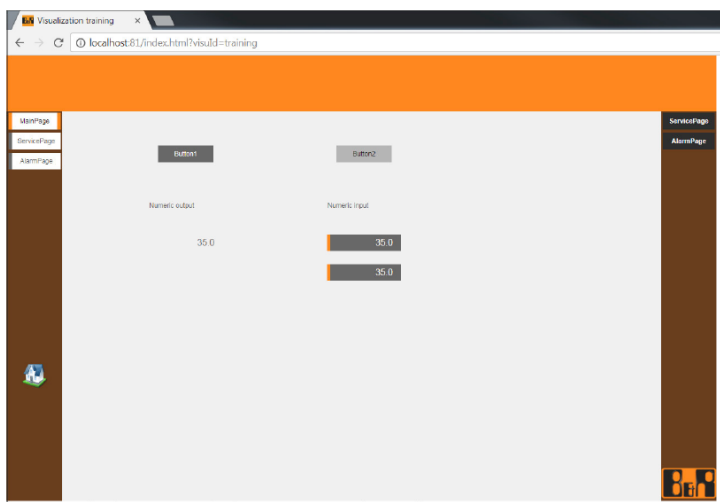


그림 66 Main Page 에 내비게이션 버튼으로 홈 이미지를 갖는 HMI 어플리케이션을 브라우저에 표시하기



터미널(Terminal)에서 화면 어플리케이션을 시작하는 것은 좋은 기회이다. 예를 들어서, 이를 완료하고 Power Panel T50 을 하드웨어 구성에 추가한다. 구성은 USB 플래쉬 드라이브를 사용해서 터미널에 전송한다. 그리고 나서 화면 어플리케이션은 터미널에서 작동할 것이다.

참조:

- [“T50 에서 mapp View 클라이언트 운영하기”](#)

7 사용자 역할 시스템(User role system)

자동화 소프트웨어는 mapp View 에서 이용하는 사용자 역할 시스템을 제공한다. 이 시스템은 ANSI 표준 359-2004 에 정의된 바와 같은 역할 기반 접근 제어(role-based access control, RBAC)를 구현한다.

7.1 개요 – 사용자 역할 시스템

역할 기반 접근 제어는 사용자 역할 및 권한을 다룬다. 권한은 역할에 부여되고 역할은 사용자와 관련된다. 사용자는 동시에 여러 역할을 가질 수 있다. 사용자에게는, 실제 상황에서 복잡하고 오류 발생이 쉽다는 것이 입증되었기 때문에, 직접 역할이 부여되지 않는다.

시스템에서 사용자는 성과 이름 등으로 식별되는 “진짜” 사람이다. 사용자는 또한 시스템에서 인증(authentication)에 사용되는 정보를 통합한다. 이 인증으로, “진짜” 사람은 그 사람이라고 주장하는 사람임을 시스템에 입증할 수 있다. 가장 일반적으로 이용되는 인증 방식은 사용자의 고유 식별과 (사용자 ID) “진짜” 사람과 시스템 만이 알고 있는 비밀 암호로 구성된다.

역할은 사용자가 특정 임무를 수행하기 위해 시스템과 상호작용하는 방법을 기술한다. 역할의 예시에는: 관리자, 서비스 엔지니어, 기계 작업자가 될 수 있다. 다양한 임무를 완수하기 위해서는 일반적으로 각각 다른 허가가 요구되고, 이를 다루기 위해 권한과 역할이 부여된다. 만약 “진짜” 사람의 임무가 달라진다면, 해당되는 사용자를 이 사람이 새로운 임무를 완수하는 데 필요한 권한을 갖도록 하는 새로운 역할로 변경하기만 하면 된다.

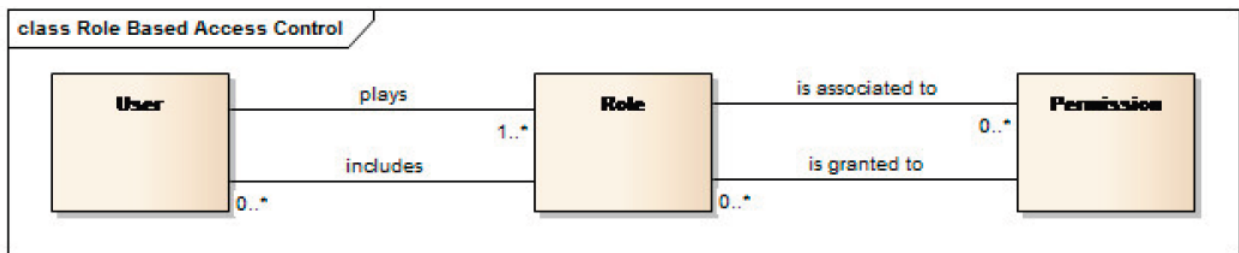


그림 67 사용자 역할 시스템의 상호 관계



[Programming / Access & Security / User role system / General](#)

7.2 사용자 역할 시스템 이용하기

7.2.1 OPC UA 노드에 쓰기 접근 제한하기

역할에 부여될 수 있는 한 가지 권한은 자동화 어플리케이션에서 변수 값을 변경하는 권한 (쓰기 접근)이다. 이 경우에, 변수를 변경하는 권한은 반드시 역할에 부여되어야 한다. 이것은 OPC UA 설정에서 이루어진다.

사용자가 OPC UA 노드 값을 변경할 수 있으려면, 반드시 쓰기 접근이 있는 역할이 부여되어야 한다. 쓰기 접근이 부여된 사용자만 값을 변경하도록 허용된다.

예제: 사용자에게 쓰기 접근 제한

이 예제의 목표는 미승인 사용자의 SetTemperature 설정 값에 대한 쓰기 접근을 금지하기 위한 것이다. 이에 대한 조건은 사용자 역할 시스템을 이용하여 만든다. 또한 HMI 어플리케이션에 인증을 추가하는 것도 가능하다.

이를 위해 다음 단계들이 필요하다:

- 1) “Operator” 역할과 “Service” 역할 추가
- 2) 새로운 사용자를 추가하고 역할 부여
- 3) OPC UA default view 에서 쓰기 접근 정의
- 4) Login, LogoutButton 및 LoginInfo 위젯 배치
- 5) 프로젝트를 컴파일, 전송하고 HMI 어플리케이션 표시

새로운 역할 추가하기



Programming / Access & Security / User role system / Configuration in Automation Studio / Editing roles in the user role system

“Operator” 역할과 “Service” 역할 추가하기

Automation Studio 에서, “Administrators” 및 “Everyone” 역할은 이미 새 프로젝트에 존재한다. 기존 역할에 “Operator”과 “Service” 역할을 추가한다. Configuration view 에서 Role.role 파일을 더블 클릭하면, 역할 관리하는 편집기가 열린다.

두 역할을 추가하고 “Operator”과 “Service”라 명명한다. 부여된 “ID”는 변동 없이 유지된다.

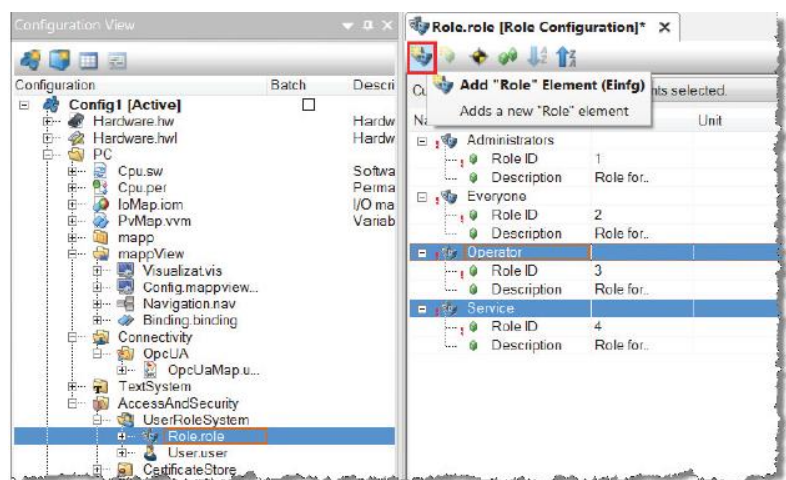


그림 68 Operator 와 Service 역할 추가하기

새로운 사용자를 추가하고 역할 부여하기

Automation Studio 에서 “Anonymous” 사용자는 이미 새 프로젝트에 존재한다.

새 역할을 작성한 후, 두 개의 새로운 사용자를 만들고 역할을 부여한다.

User.user 파일을 더블 클릭하면 사용자가 관리되는 편집기가 열린다.

그 다음 사용자 “John”과 “Dave”를 만든다. 사용자 “John”에게는 “Operator” 역할을 부여하고 사용자 “Dave”에게는 “Service” 역할을 부여한다. 각 이름은 암호로 이용된다.

Username	Password	Role
John	John	Operator
Dave	Dave	Service

표 14 사용자와 부여된 역할 개요

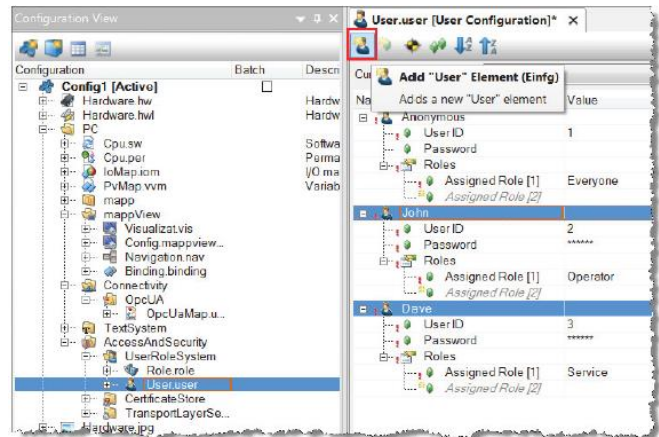


그림 69 사용자 “John”과 “Dave” 추가하고 설정하기



Programming / Access & Security / User role system / Configuration in Automation Studio / Editing roles in the user role system

OPC UA default view 에서 쓰기 접근 정의하기

사용자를 만들고 역할이 부여된 후, OPC UA Default View 에 역할에 대한 권한을 정의한다.

“Default View” 노드를 선택하면 디폴트 허가가 설정되고, 그 아래 변수에 대한 모든 허가가 도출된다.

각 역할에 대한 허가는 다양한 역할을 추가함으로써 정의될 수 있다. OPC UA 노드에 대한 쓰기 접근이 “Service” 역할이 할당된 사용자에게만 주어짐을 보장하기 위해, “쓰기(Write)”는 “Service” 역할에만 설정한다.

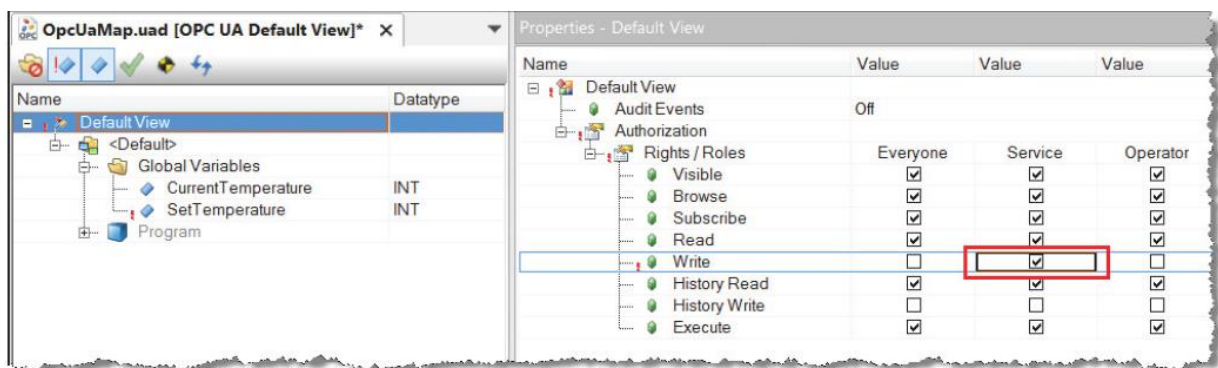


그림 70 OPC UA default view 에서 역할에 따른 쓰기 접근 정의하기



쓰기 접근 권한을 부여하기 전에, OPC UA 쓰기 접근 권한이 몇가지 또는 대부분의 OPC UA 변수에 제한되어 있는지 확인이 필요하다. 그렇지 않으면 쓰기 접근은 OPC UA 변수 스스로 제한을 받는다.

이 문서에서, 쓰기 접근은 오직 OPC UA 변수 “SetTemperature”만 제한되었다. 즉 노드 “Default View”는 각 노드에 대한 쓰기 접근이 사용되는 위치이다.

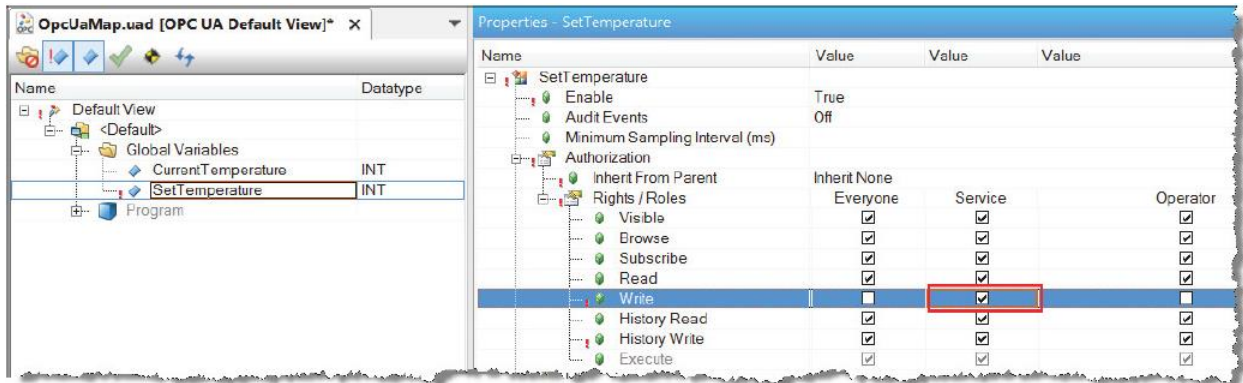


그림 71 “SetTemperature” OPC UA 변수의 쓰기 접근 권한은 Service 역할에만 부여되었다.

로그인(Login), 로그아웃 버튼(LogoutButton) 및 로그인 정보(LoginInfo) 위젯 배치하기

각 사용자에게 설정된 허가가 발효되려면, 사용자는 시스템에서 인증을 받아야 한다.

로그인 및 로그아웃을 위한 위젯 뿐만 아니라 상세한 로그인 정보를 표시하기 위한 위젯도 있다.



사용자가 인증되지 않으면, 시스템은 “Anonymous” 사용자가 시스템에 접근하는 것으로 생각한다. 시스템의 미인증 사용 중에는 “Anonymous” 사용자에게 부여된 모든 역할에 주어진 권한이 유효하다. 디폴트로, “Anonymous” 사용자에게는 “Everyone” 역할이 부여된다.

- 로그인 위젯(Login widget)

“Login” 위젯은 위젯 카탈로그로부터 드래그 앤 드롭에 의해 서비스 페이지의 콘텐츠에 배치될 수 있다.

- 로그인 정보 위젯(LoginInfo widget)

현재 어느 사용자가 로그인했는지 알아보기 위해, “LoginInfo” 위젯을 “ContentTop”에 배치해야 한다.

- 로그아웃 버튼 위젯(LogoutButton widget)

사용자가 다시 로그아웃하기 위해, “ContentTop”에 “logout button”을 배치해야 한다.

- 라벨과 이미지 위젯(Label and Image widget)

HMI 어플리케이션을 더 명확하고 잘 보이게 하기 위해, 사용자 아이콘을 갖춘 “Image” 위젯과 “Logged in as:”라는 텍스트를 갖는 “label” 위젯을 “ContentTop”에 배치한다. 사용자 아이콘의 이미지는 Media 폴더에 존재한다.



로그인한 사용자가 LogoutButton 을 이용하여 로그아웃하면, “Anonymous” 사용자에게 부여된 권한이 다시 유효해진다.

속성 “Layout.Padding/padding”을 0px 으로 설정한다.

Widget	Content	Top	Left	Width	Height
Login	Service page	40	50	300	160
Login info	Content Top	15	1030	230	30
Logout Button	Content Top	50	1030	100	30
Label	Content Top	15	880	140	30
Image	Content Top	11	830	30	38

표 15 로그인 정보에 대한 위젯, 콘텐츠 및 속성

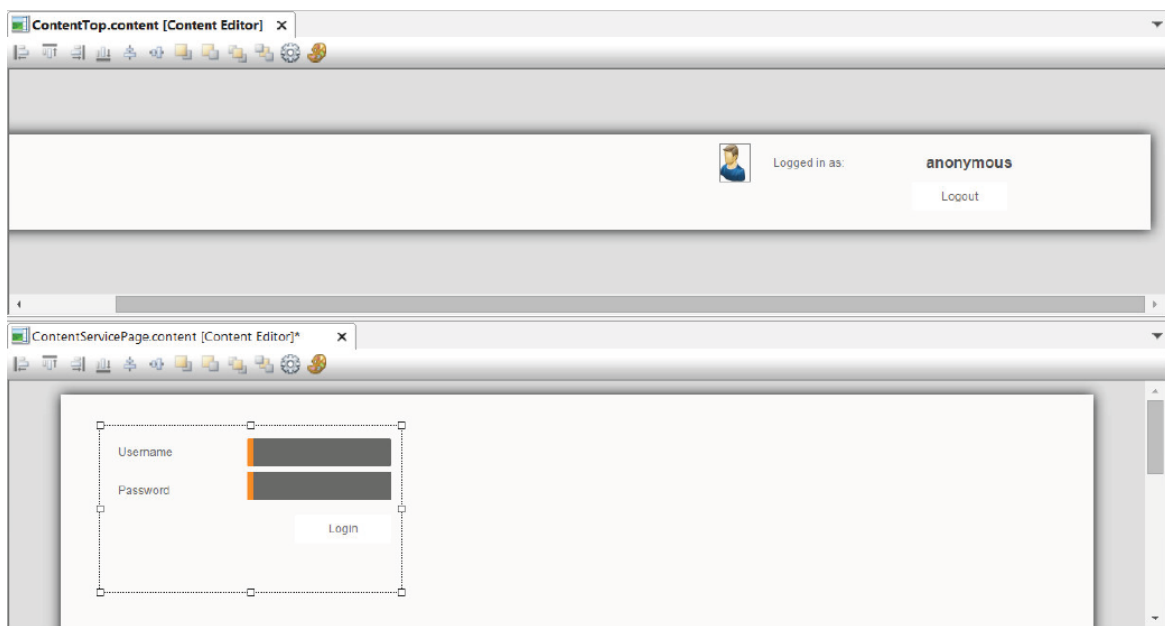


그림 72 ContentServicePage 및 ContentTop 에 배치된 Login, Logout, Label 과 image 위젯



“Layout.padding/padding” 속성이 0px 로 설정되면, 위젯의 텍스트와 프레임 사이의 거리가 달라진다.

예상되는 결과

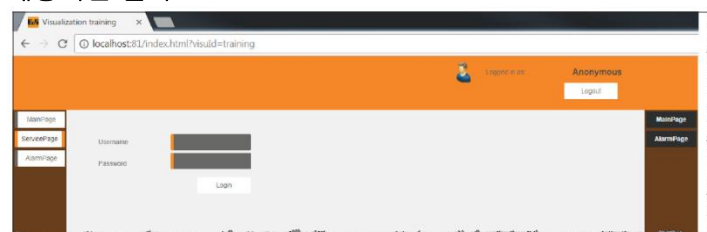


그림 73 login 위젯을 포함하는 HMI 어플리케이션

사용자 “Dave”로 로그인한 후에, 항목은 MainPage 에 NumericInput 위젯 입력에 영향을 줄 수 있다. 다른 사용자들을 위해서, OPC UA 노드에 쓰기 접근 부족으로 입력은 자동으로 비활성화된다.

7.2.2 값 범위 제한하기

EU 범위 OPC UA 노드(EU range OPC UA node)

OPC UA 노드는 입력할 때 값 범위 (EU 범위)를 제한하는 것을 허용한다. 사용자 역할 시스템을 이용하여, EU 범위는 다양한 역할에 대해 개별적으로 정의할 수 있다.

예제: “Service” 역할에 따른 값 범위 확장하기

예제의 목표는 새로운 “Service” 역할에 대한 OPC UA 노드의 EU 범위를 확장하는 것이다. 이 목표를 달성하기 위해, 새로운 역할과 새로운 사용자를 만들고 OPC UA 노드의 EU 범위를 반드시 정의해야 한다.

다음 단계를 반드시 수행해야 한다:

- 1) “Service” 역할에 따른 EU 범위 추가하기
- 2) 프로젝트를 컴파일, 전송하고, HMI 어플리케이션에 표시하기

“Service” 역할에 대한 EU 범위 확장하기

역할 설정을 통해, OPC UA 디폴트 뷰에서 각 역할에 대한 값 범위를 개별적으로 정의할 수 있다. 이전에는, OPC UA 노드에 대한 “디폴트” EU 범위가 이용되었다. EU 범위 범주에 역할을 추가함으로써, 값 범위를 확장하거나 제한할 수 있다.

“EU 범위(EU range)”에 “Service” 역할을 추가한다. 값 범위는 “Default” 값으로부터 도출하여 하한(Low)=20” 및 “상한(High)=60”을 기반으로 한다.

Role	EU range LOW	EU range HIGH
Service	20	60

표 16 “Service”역할에 확장된 EU 범위

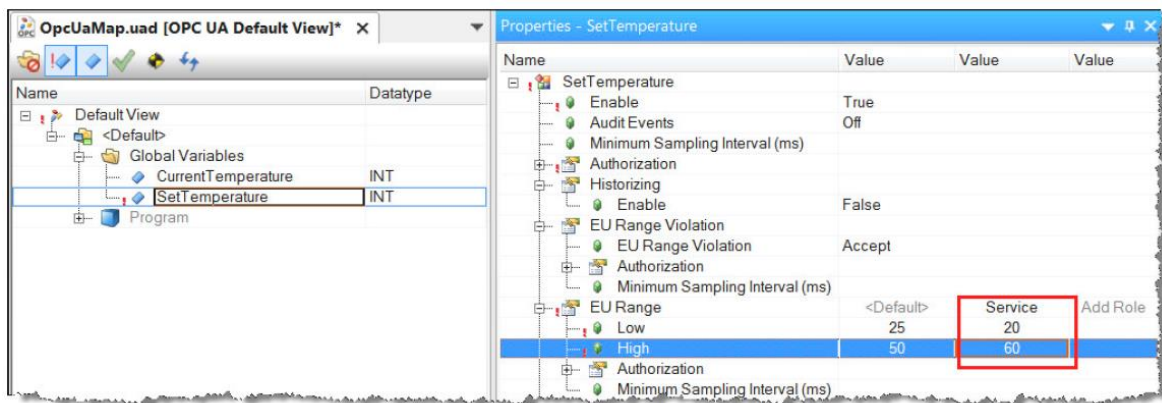


그림 74 “Service” 역할에 확장된 EU 범위

맞춤화된 EU 범위는 OPC UA 노드에 쓰기 접근을 가진 다른 역할에 대해서는 정의되지 않았기 때문에, 디폴트 값이 다른 역할에 유효하다.

EU 범위 위반 응답은 여러분이 EU 범위를 벗어나는 값을 입력할 때 일어나는 행동이다.

다음과 같은 EU 범위 위반 응답이 존재한다.:

EU 범위 위반 (EU range violation)	설명
Accept	값이 EU 범위를 벗어나도 값 입력이 수용된다.
Reject	입력된 값이 수용되지 않는다.
Clamp	입력된 값이 상한/하한에서 잘립니다.

표 17 개요 - EU 범위 위반

프로젝트를 컴파일하고, 다양한 사용자 이름으로 로그인하고 각각의 허가를 설정함으로써 HMI 어플리케이션을 시험할 수 있다 (읽기/쓰기 접근 + 디폴트/ 확장된 값 범위).

예상되는 결과

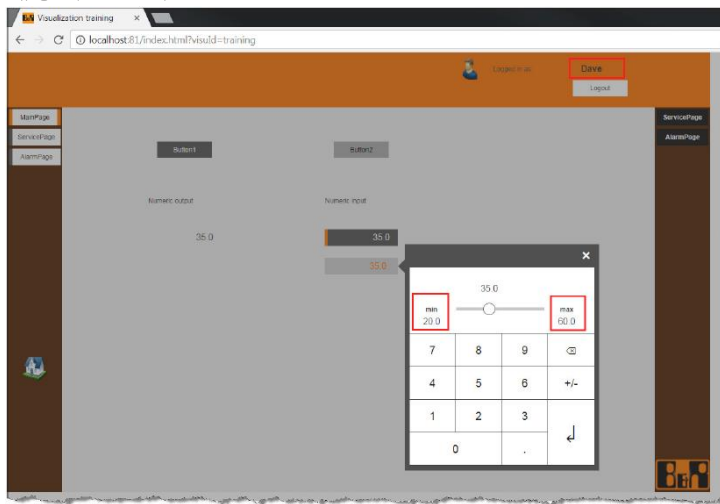


그림 75 Dave 사용자로 로그인하고 수치 입력을 선택한 상태에서 HMI 어플리케이션

7.2.3 역할을 변경하고 추가하기

기계 사용자가 변경되고, 새로운 사용자가 추가되고, 다른 사용자가 이동되는 등, 기계 수명 주기 동안 지속적으로 역할 변경이 이루어진다.

다양한 사용자에게 대한 모든 허가는 정의된 역할에 좌우되기 때문에, 기존 사용자는 적절한 역할 활동만 부여하면 된다.

예제: 사용자와 권한 변경하기

예제에서는, “Supervisor” 역할을 갖는 사용자 “Mike”가 다른 회사로 이직한다. “Supervisor” 역할은 사용자 “John”에게 인수된다. 새로운 직원 “Patrick”이 채용되고 “Operator” 역할이 부여된다. 사용자 Dave 는 “Service” 역할을 유지한다.

다음 예제의 목표는 기술된 바와 같이 사용자 변경을 수행하는 것이다.

- 1) 사용자 “Mike”를 제거하기
- 2) 사용자 “John”에게 새로운 “Supervisor” 역할 부여하기
- 3) 암호가 “Patrick”인 새로운 사용자 “Patrick”을 만들고 “Operator” 역할 부여하기

8 현지화(Localization)

mapp View 에서, 현지화(localization)는 HMI 어플리케이션 내용을 현지 언어와 현지 문화 기준에 적용시키는 것을 의미한다. mapp View 는 HMI 어플리케이션의 현지화를 위해 실행 시간에서 텍스트와 단위 적용을 허용한다.

텍스트 시스템(text system)은 텍스트 현지화를 위해, 단위 시스템(unit system)은 단위를 조정하는데 이용된다. 텍스트 시스템과 단위 시스템은 mapp View 에 통합된 것이 아니다. 이 시스템의 기능은 mapp View 에서 이용되기 때문에, 텍스트 시스템과 단위 시스템의 이용에 대한 소개는 이 교육 매뉴얼의 일부이다.

8.1 개요 – 텍스트 시스템(Text system)

Automation Studio 는 텍스트 현지화를 위해 텍스트 시스템을 제공한다. 기반구조 기능은 mapp View 의 통합된 일부는 아니다. 텍스트 시스템은 다중 언어로 된 텍스트를 이용할 수 있도록 하고 아래 표에 보인 바와 같이 2 차원적인 표로 생각할 수 있다.

한 차원은 고유 식별자에 의해 대표되는 의미론적 의미를 열거하고, 다른 차원은 의미 정보의 텍스트 기반 표현을 갖는 모든 언어에 의해 형성된다. 이 표의 각 행(行)은 해당 언어의 텍스트를 갖는 의미론적 의미에 대응된다. 이 표의 각 열은 다양한 의미론적 의미를 갖는 특정한 언어의 텍스트를 대표한다. 그러므로 표의 각각의 셀은 이 표의 열(列)이 대표하는 언어의 의미론적 의미에 대한 (첫 번째 열의 식별자) 텍스트이다.

식별자(Identifier)	de	en	fr	es
Text_Cancel	Abbrechen	Cancel	Annuler	Cancelar
Text_Yes	Ja	Yes	Oui	Si
Text_No	Nein	No	Non	No

표 18 현지화된 텍스트의 도표 표현의 예

표의 예시는 “취소”, “예” (승인) 및 “아니오” (거부)의 의미론적 의미에 대한 행을 보여준다. 이들 의미론적 의미는 식별자 “Text_Cancel”, “Text_Yes” 및 “Text_No”를 이용하여 명확히 정해진다.

의미론적 의미 “취소”의 경우, 독일어 (“Abbrechen”), 영어 (“Cancel”), 프랑스어 (“Annuler”) 및 스페인어 (“Cancelar”) 텍스트가 있다.

텍스트 서술이 현지화된 텍스트가 있는 시스템에서 사용될 때, 이 텍스트 서술의 의미론적 의미에 대한 식별자가 반드시 사용되어야 한다. 시스템이 “독일어” 언어 콘텍스트에서 사용되는 경우에, 시스템은 독일어 언어 텍스트를 갖는 의미론적 의미에 대한 식별자를 대체한다. 여기에서는 표의 “de” 열(列)로부터의 텍스트가 식별자에 대해 사용된다. 동일한 시스템이 “영어” 언어 콘텍스트에서 사용된다면, 시스템은 영어 언어 텍스트를 갖는 의미론적 의미에 대해 식별자를 (표의 “en” 열) 대체한다.

8.1.1 식별자(Identifier)

식별자는 각각 다른 언어로 표현될 수 있는 언어론적 서술이다. 자동화 시스템에는 많은 양의 의미론적 서술이 포함될 수 있기 때문에, 동일한 식별자가 각각 다른 의미론적 서술에 부여되는 것을 방지하기 위해 이들을 구조화할 필요가 있다. 의미론적 서술은 서로 독립적인 다른 사람들에 의해 정의될 수 있기 때문에 이것은 특별히 중요하다.

예를 들어, 자동화 어플리케이션의 개발자는 로거(logger) 입력에 사용하기 위한 의미론적 서술을 정의하고, 이와 독립적으로 HMI 어플리케이션의 개발자는 HMI 요소에 사용되는 의미론적 서술을 정의할 수 있다.

텍스트 시스템은 네임 스페이스(namespace)에서 의미론적 서술에 대한 식별자를 구성하는 기능을 제공한다. 의미론적 서술에 대한 식별자는 몇 개의 부분으로 구성된다:

Identifier = Namespace+NamespaceSeparator+Text_ID

네임 스페이스는 또다른 네임 스페이스의 일부일 수 있다. 그러므로 네임 스페이스는 계층 구조로 구성될 수 있다. 네임 스페이스(namespace)에 대해서:

Namespace = NamespaceName [+ NamespaceSeparator+Namespace]

식별자는 또한 완전히 갖춰진 텍스트 ID(Text_ID)로도 불린다. Text_ID 및 NamespaceName 의 규칙은 도움말에 상세히 기술되어 있다. 네임 스페이스 분리자(NamespaceSeparator)는 단일 문자로써, 이 또한 도움말에 설명되어 있다.

몇 개의 식별자 예시는 아래와 같다 (완전히 갖춰진 Text_IDs).

- 1) Texts/AppEvents/Internal/FatalError
- 2) Texts/Alarms/Alarm1
- 3) Texts/Program/Alarms/AlarmID1

모든 예시에 대해, 네임 스페이스 분리자는 문자 “/”이다. 예시에서 Text_ID 는 "Fatal Error", "Alarm1", "AlarmID1"이다. 첫번째 예시에는 “Texts”라 불리는 네임 스페이스가 포함되고, 여기에는 “AppEvents”라고 불리는 다른 네임 스페이스이 포함되며, 이 네임 스페이스는 또 다른 네임 스페이스 “Internal”을 포함한다.

8.1.2 프로젝트의 언어 설정하기

Automation Studio 프로젝트에서는, 의미론적 서술을 위해 어떤 언어를 사용할 것인지 정의해야 한다. 프로젝트 언어가 몇 개인지도 반드시 규정되어야 한다.

제공하는 프로젝트 언어를 규정하는 것 이외에도, 또한 “설계 언어(design language)”를 반드시 정의해야 한다. 설계 언어는 설정 중에 Automation Studio 에서 텍스트를 표시할 때 사용되는 언어를 정의한다.

8.1.3 타겟 시스템의 언어 설정

Automation Studio 프로젝트에 설정된 언어는, 각 프로젝트 구성을 위해 정의된 특별한 설정을 Automation Studio 프로젝트에서 타겟 시스템으로 전송해야 언어 설정이 적용가능하다.

“시스템 언어(system language)”와 “대체 언어(fallback language)”도 정의된다. “시스템 언어”는 디폴트로 텍스트를 표시하는 데 사용되는 언어를 결정한다. “대체 언어”는 “시스템 언어”의 텍스트를 이용할 수 없는 경우에 텍스트를 표시하는 언어를 정의한다.

8.1.4 mapp View 에서 텍스트 시스템(Text system)

mapp View 에서는, 텍스트 시스템으로 현지화된 텍스트를 이용할 수 있다. mapp View 에서 여러분의 텍스트를 만들기 위해 텍스트 파일을 이용할 수 있다. 텍스트 파일을 이용하여 여러분의 텍스트가 시스템에 입력된다면, 그 텍스트는 텍스트 시스템에 의해 이용되도록 제공될 수 있다.

여러분의 현지화된 텍스트와 텍스트 시스템은 mapp View HMI 어플리케이션이 다른 언어로 표시되는 것을 허용하는 기반이다. mapp View 는 HMI 어플리케이션을 위해 언어를 전환할 수 있는 기능을 제공한다.

8.2 텍스트 시스템 이용하기

현지화된 텍스트는 mapp View HMI 어플리케이션의 여러 위치에서 이용될 수 있다. 다음 예제에서는 몇 개 예시로 이에 대한 시범을 보인다.

8.2.1 수동 내비게이션의 텍스트

이 예제의 목표는 독일어와 영어로 수동 내비게이션(manual navigation) “NavigationButton” 위젯을 설정하고 실행 중 언어를 전환하는 것이다.

예제: 수동 내비게이션의 텍스트 현지화하기

이 목표를 달성하기 위해, 반드시 다음 단계를 수행해야 한다.

- 1) 프로젝트 언어 “독일어(de)”와 “영어(en)”를 설정
- 2) “독일어(de)”와 “영어(en)” 텍스트 작성
- 3) 위젯에 현지화된 텍스트 설정
- 4) 텍스트 설정(textconfig) 파일 설정
- 5) 언어 셀렉터(Language Selector) 위젯 추가
- 6) 프로젝트를 컴파일, 전송하고, 브라우저에 표시

프로젝트 언어 “독일어(de)”와 “영어(en)” 설정하기

프로젝트에 사용되는 언어를 정의하기 위해, 로지컬 뷰 어느 위치에서든 Object Catalog로부터 언어 설정 파일인 프로젝트 언어(Project.language)를 반드시 추가해야 한다.



[Programming / Text system / Managing project languages](#)

먼저, 드래그 앤 드롭으로 Object Catalog에서 로지컬 뷰로 반드시 프로젝트 언어(Project.language) 파일을 추가해야 한다.

파일을 더블 클릭하면 표 편집기가 열린다. 언어(.language) 파일을 삽입하면 영어(“en”), 독일어(“de”) 및 프랑스어(“fr”) 언어는 이미 존재한다. 프랑스어는 우리 예시에 요구되지 않으므로 제거할 수 있다.

“설계 언어(design language)”는 디폴트에 의해 설정된 바와 같이 “English(영어)”로 유지한다.

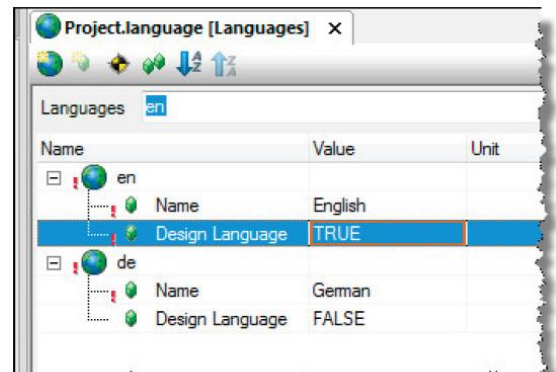


그림 76 DE 및 EN 언어가 정의된 프로젝트 언어 편집기

“독일어”(“de”)와 “영어”(“en”)로 텍스트 작성하기

언어 “독일어”(“de”)와 “영어”(“en”)가 규정된 후, “mapp View LocalizableText.tmx”파일은 Object Catalog 에서 “mappView/Resources/Texts” 노드로 추가하고 “VisualizationTexts”로 이름을 변경한다.



네임 스페이스(Namespace) “IAT”는 mapp View HMI 어플리케이션에서 사용하도록 텍스트 파일에 반드시 입력되어야 한다.

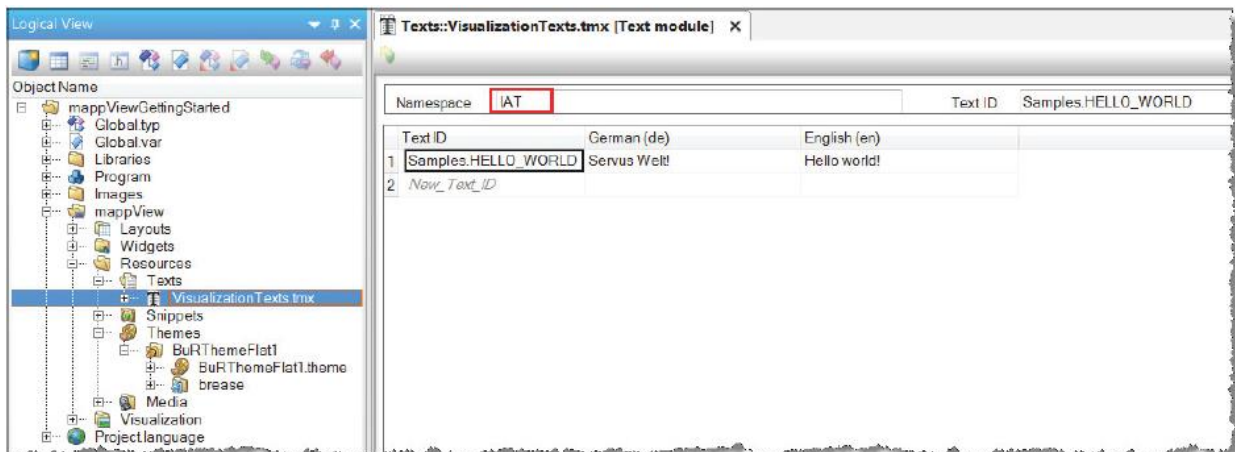


그림 77 현지화 가능한 테스트 파일에 정의된 네임 스페이스 IAT

다음에는 수동 “NavigationButton” 위젯에 대한 텍스트를 고유 “ID”와 함께 영어와 독일어로 표 편집기에 작성한다.

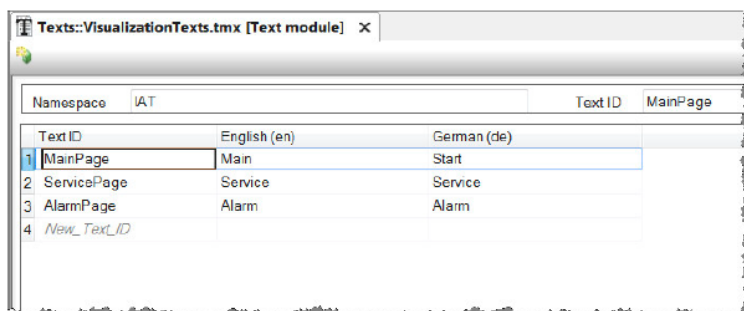


그림 78 정의된 텍스트를 갖춘 현지화 가능한 텍스트 파일

현지화된 텍스트 설정하기

모든 텍스트가 독일어와 영어로 만들어지면, 수동 내비게이션 “NavigationButton” 위젯에서 참조될 수 있다.

이를 위해 Visual editor 에서 “ContentLeft”를 열고 “NavigationButton”을 선택한다.

두 가지 변형을 속성 창을 이용하여 텍스트 위젯에 할당 할 수 있다:

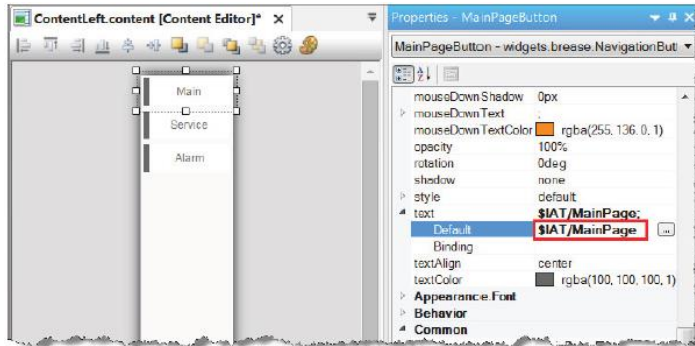


그림 79 접두어 "\$IAT/"를 갖는 현지화된 텍스트를 수동 내비게이션 버튼 위젯에 할당하기

- 1) 접두어로서 문자 "\$IAT/"를 텍스트 ID 를 직접 입력할 수 있다.

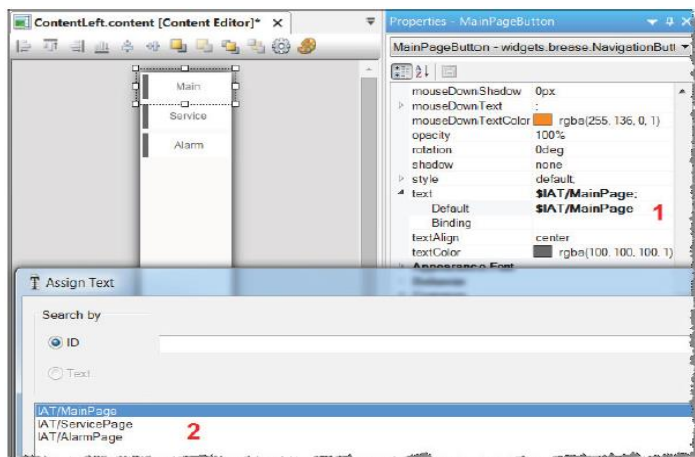


그림 80 "텍스트 할당(Assign Text)" 대화 상자를 사용하여 현지화된 텍스트를 수동 내비게이션 버튼에 할당하기

- 2) text ID 는 “텍스트 할당(Assign Text)” 대화 상자를 이용하여 찾을 수 있다.

텍스트 설정(textconfig) 파일 설정하기

작성된 텍스트가 타겟 시스템에 전송되려면, 반드시 Object Catalog 의 “TC.textconfig” 파일을 컨피규레이션 뷰 / “텍스트 시스템(Text System)” 패키지에 추가해야 한다.

그러면 “시스템 언어”와 “대체 언어”가 “TC.textconfig” 파일에 정의된다. 또한 어느 언어가 타겟 시스템에 전송되는지도 (Target Language 1, 기타 등등) 정의해야 한다.

텍스트 파일이 타겟 시스템에서 이용될 수 있으려면, 작성된 텍스트를 포함하는 TMX 파일이 반드시 “TMX files for target”에 선택되어야 한다.



Programming / Text system / Text system configuration

완전한 정의는 다음과 같다.:

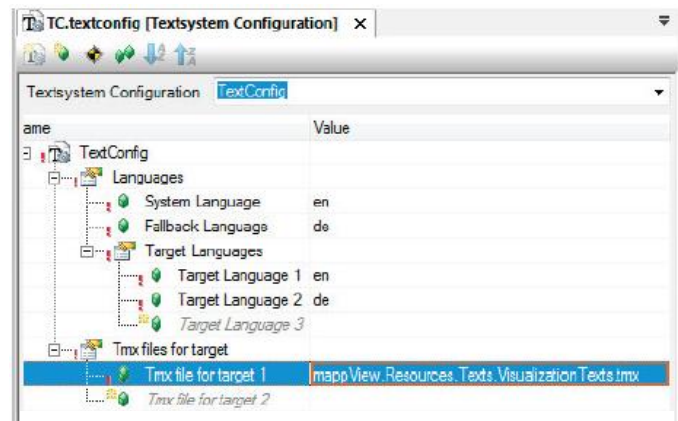


그림 81 설정된 TextConfig 파일

LanguageSelector 위젯 추가하기
HMI 어플리케이션에서 언어를 변경할 수 있도록, 반드시 Object Catalog 에서 LanguageSelector 위젯을 “ServicePage” 콘텐츠에 추가해야 한다.

추가된 LanguageSelector 는 드롭 다운으로 언어를 선택할 수 있다.

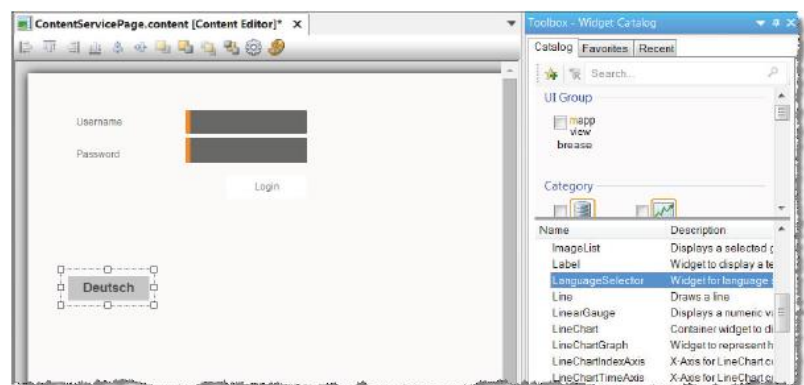


그림 82 Service Page 에 추가된 Language Selector

예상되는 결과:

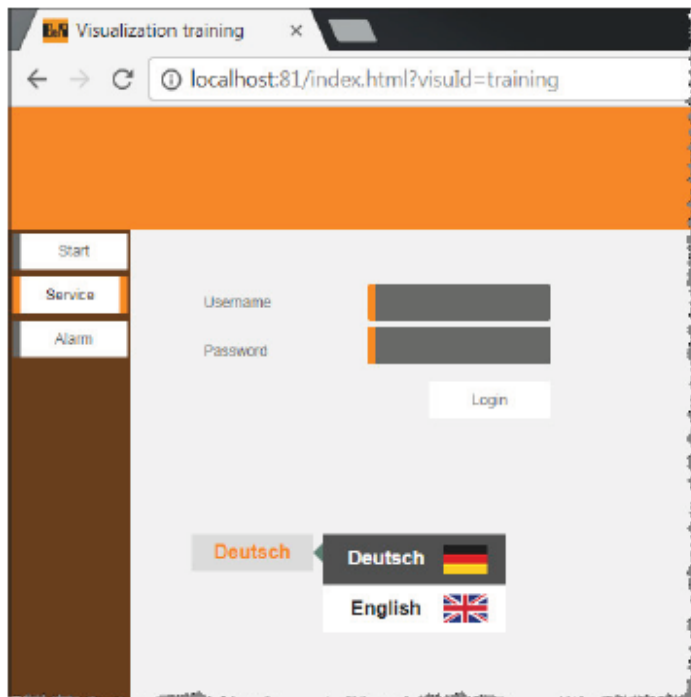


그림 83 수동 내비게이션 버튼에 현지화된 텍스트와 언어 선택기를 포함하는 HMI 어플리케이션

8.2.2 자동 내비게이션의 텍스트

이 예제의 목표는 독일어와 영어로 자동 “NavigationBar” 위젯의 텍스트를 설정하고 실행 중 언어를 전환하는 것이다. 자동 내비게이션의 “NavigationBar” 위젯은 링크된 페이지에 설정된 텍스트를 이용한다.

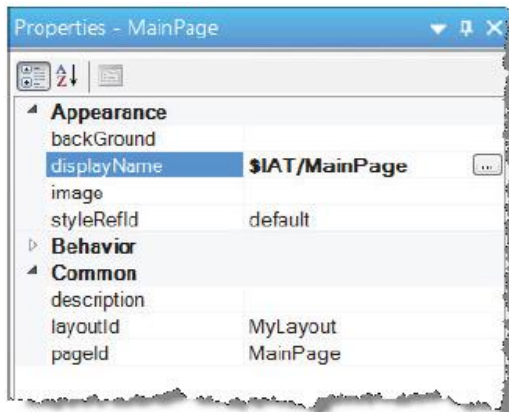
예제: 자동 내비게이션의 텍스트 현지화하기

이 목표를 달성하기 위해, 반드시 다음 단계들을 수행해야 한다:

- 1) 해당 페이지 상에 TextID 를 설정
- 2) 프로젝트를 컴파일, 전송하고, 브라우저에 표시

해당 페이지에 텍스트 ID(text ID) 설정하기

프로젝트 언어 “독일어(de)”와 “영어(en)” 이고 고유 텍스트 ID 가 “de”와 “en”로 된 텍스트가 이미 작성되었고 수동 내비게이션 버튼에 대해 설정되어 있기 때문에, 이 예제에서 이들 단계는 더 이상 필요하지 않다.

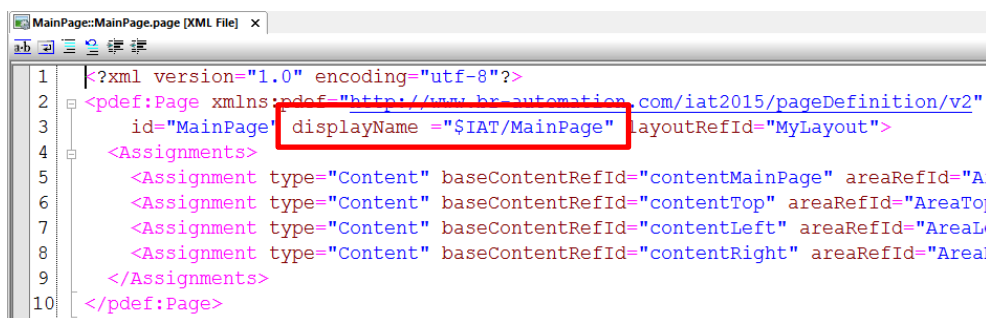


자동 내비게이션 버튼의 텍스트를 현지화하고 그에 따라 텍스트를 HMI 어플리케이션에 표시하려면, 해당 페이지에 대한 “디스플레이 명칭(displayName)” 속성이 반드시 .page 파일에 입력되어야 한다.

“MainPage.page” 파일을 더블 클릭하면 페이지에 대한 XML 편집기가 열린다.

그러면 반드시 “displayName=” 속성을 페이지에 설정해야 한다. “displayName” 속성에 “\$IAT/”로 시작되는 관련 텍스트 “ID”를 규정하면 “NavigationBar” 위젯에 텍스트 참조가 가능하다.

그림 84 자동 내비게이션 버튼에 현지화된 텍스트 설정하기



예제: 서비스 페이지(ServicePage)와 알람 페이지(AlarmPage)에 텍스트 ID 입력하기

“서비스 페이지(ServicePage)”와 “알람 페이지(AlarmPage)”에 설명된 절차를 반복한다.

Page	displayName=
ServicePage	“\$IAT/ServicePage”
AlarmPage	“\$IAT/AlarmPage”

표 19 자동 내비게이션에 텍스트 ID 부여하기

예상되는 결과:

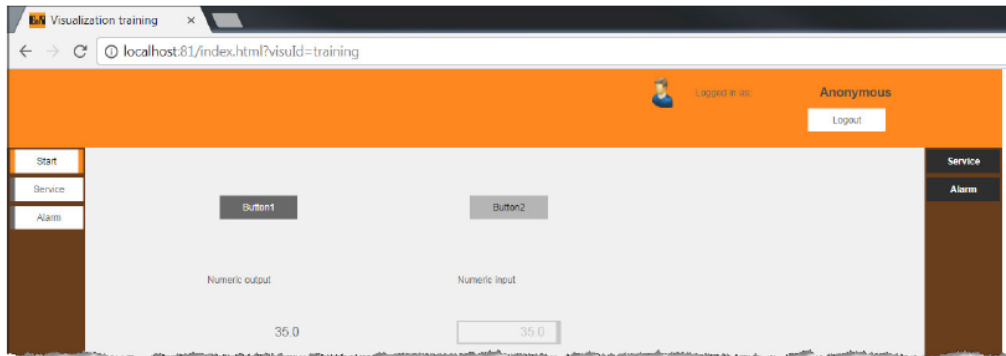


그림 85 독일어로 전환된 상태로 브라우저에 표시된 HMI 어플리케이션

8.2.3 로그아웃 버튼(LogoutButton)과 라벨 로그인 정보(LabelLoginInfo) 위젯의 텍스트

이 예제의 목표는 로그인한 사용자에게 관한 정보를 갖는 “Label” 위젯의 텍스트와 “LogoutButton” 위젯의 텍스트를 현지화하는 것이다.

예제: 위젯 텍스트 현지화하기

로그인 한 사용자 정보가 있는 “Label” 위젯의 텍스트와 “logout button”의 텍스트를 현지화하고 “Language Selector”를 이용하여 독일어와 영어를 전환한다.

예상되는 결과:

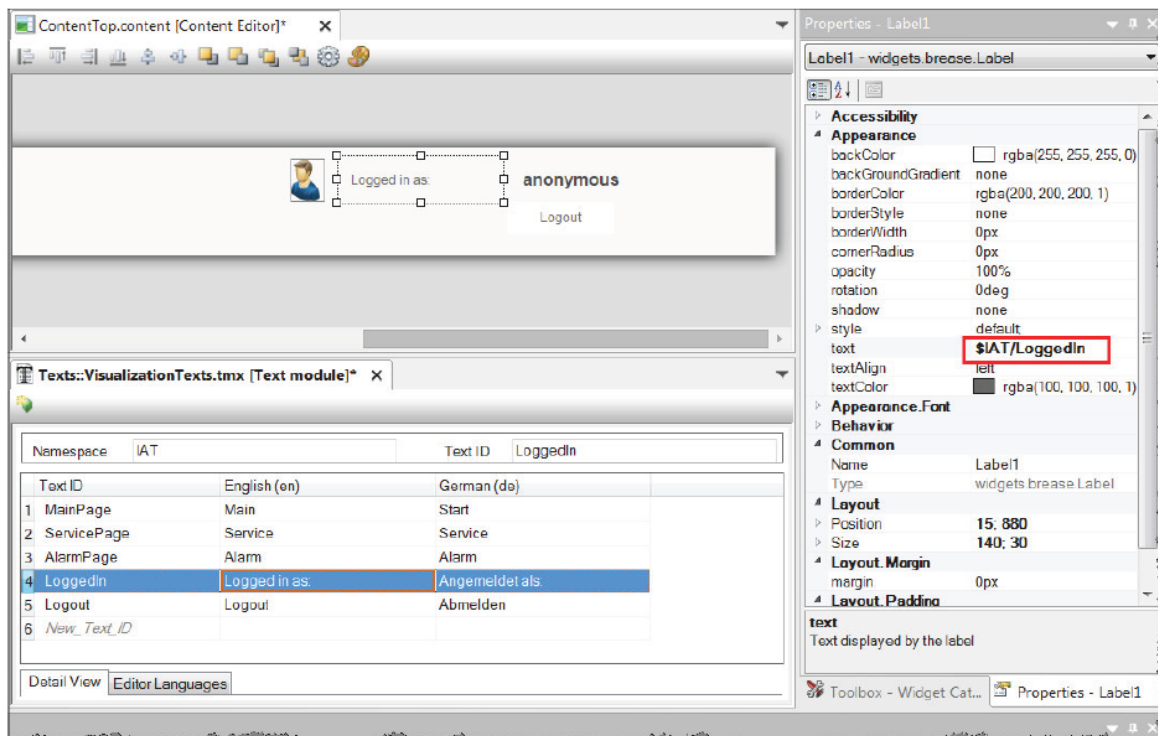


그림 86 라벨과 로그아웃 버튼의 현지화 된 텍스트

8.3 개요 – 단위 시스템(Unit system)

Automation Studio 는 자동 단위 변환 기능과 통합 단위 시스템을 사용자에게 제공한다. 단위 현지화를 위해, 사용자는 “metric(미터법)”, “imperial(영국식)” 및 “imperial-us(영미식)” 측정 시스템과 1400 개 이상의 단위를 제공한다.

OPC UA 노드를 이용하면, EU 범위에 추가하여 엔지니어링 단위를 설정할 수 있다. OPC UA 노드에 대한 엔지니어링 단위는 그 값을 해석할 때 사용되는 물리적 단위를 규정한다.

위젯을 사용하여, 결합된 OPC UA 노드에 값을 표시하기 위해 사용되는 단위는 선택된 측정 시스템에 따라 정의될 수 있다.

단위가 자동화 어플리케이션에서 준비한 값에 사용 된 단위와 값을 표시하는데 사용해야하는 단위가 알려져 있는 경우, 시스템은 자동적으로 그 값을 원하는 단위로 변환한다.

모든 단위와 각 단위에 대해 “unit” 속성에 입력되어야 하는 공통 규격 (예, 섭씨 온도 = CEL)은 Automation Studio 도움말에 기술되어 있다.



[Programming / Unit system / Available standard units](#)

8.4 단위 시스템 이용하기

8.4.1 값과 단위를 함께 출력하기와 측정 시스템 전환하기

HMI 어플리케이션에서 자동화 어플리케이션 값의 단위를 표시하려면, 수치 값뿐만 아니라 단위를 바인딩해야 한다.

예제: 수치 출력(NumericOutput) 위젯에 단위 “섭씨(degrees Celsius)” 표시하기

이 예제의 목표는 OPC UA 노드 “CurrentTemperature”와 “SetTemperture”에 섭씨 온도(°C)를 정의하고, 이를 “NumericOutput”, “NumericInput” 위젯에 표시하고, °C 와 °F 사이를 전환하기 위해 “MeasurementSystemSelector”를 이용할 수 있도록 하는 것이다.

여기에서 반드시 다음 단계들을 수행해야 한다:

- 1) OPC UA 노드 엔지니어링 단위를 “섭씨 온도(degrees Celsius)”로 설정
- 2) 수치 출력(NumericOutput)을 추가하고 노드 바인딩 선택
- 3) 단위 전환 설정
- 4) “측정 시스템 셀렉터(MeasurementSystemSelector)” 위젯 추가
- 5) 설정 온도(SetTemperature)와 수치 입력(NumericInput) 과정을 반복
- 6) 프로젝트를 컴파일, 전송하고, HMI 어플리케이션에 표시

OPC UA 노드 엔지니어링(OPC UA node Engineering)에 “섭씨 온도(degrees Celsius)” 단위 덧붙이기
OPC UA 디폴트 뷰에서, OPC UA 노드가 활성화된 후 노드에 단위를 설정할 수 있다.

Object Catalog 에서 드래그-앤드-드롭으로 속성 창으로 “섭씨 온도(degree Celsius)”를 추가하므로써, 선택된 “엔지니어링 단위(Engineering Unit)” 노드가 OPC UA 디폴트 뷰의 속성 창에 추가된다.

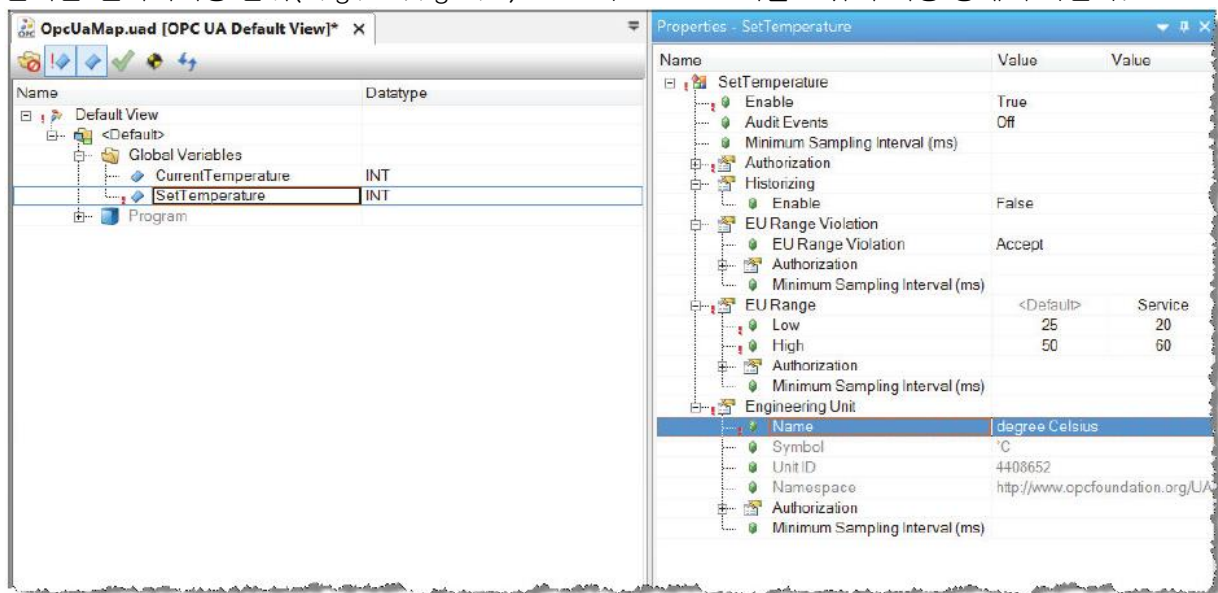


그림 87 OPC UA 노드 현재 온도(CurrentTemperature)에 섭씨 온도(degrees Celsius) 단위 덧붙이기



HMI application / mapp View / Guides / Getting started / Connecting widgets to data / Displaying values and units

수치 출력(NumericOutput) 위젯 추가하기

다른 “NumericOutput” 위젯을 “메인 페이지(MainPage)” 콘텐츠에 추가하고 크기를 증가시킨 후, “NumericOutput” 위젯에 적절한 단위가 표시되도록 반드시 값 바인딩 대신에 노드 바인딩을 선택한다.

절차:

- 수치 출력을 추가, 크기를 변경하고 선택
- 변수 선택(Select Variable) 대화 상자를 열기

OPC UA 노드를 선택하고 (값 이 아니라) 노드 바인딩을 설정한다.

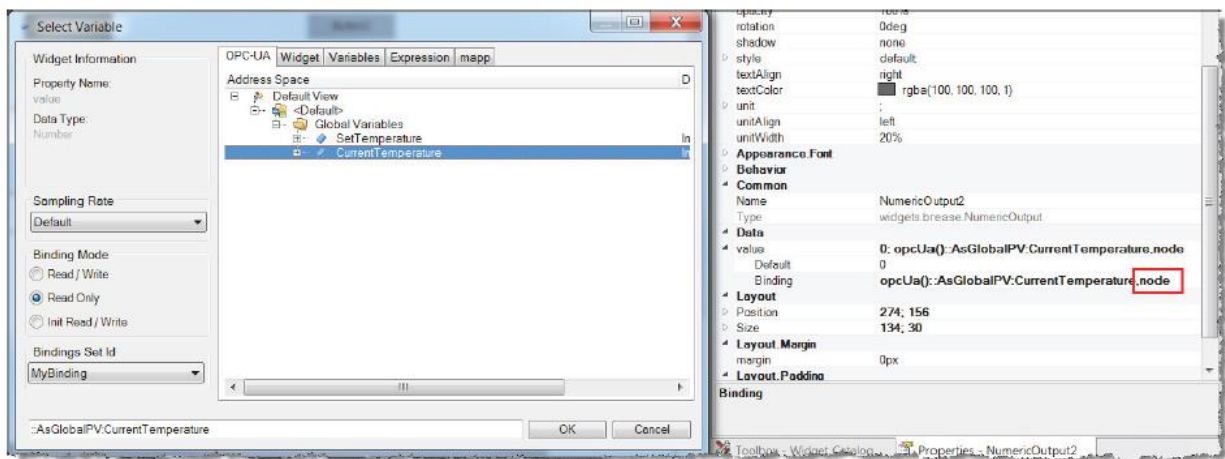


그림 88 실제 온도(ActualTemperature) 노드의 바인딩 대화 상자에서 노드 바인딩 선택하기

단위 전환 설정하기

NumericOutput 위젯에 단위가 표시되려면, 반드시 해당 “단위(unit)” 속성도 설정되어야 한다. 표시되는 단위는 반드시 각 측정 시스템에 대해 ("metric", "imperial", "imperial-us") 규정되어야 한다. 단위는 공통 규격(common code)을 이용하여 규정된다.

지원되는 단위의 공통 규격의 정보는 Automation Studio 도움말에서 제공한다. 자동화 어플리케이션에서 제공하는 단위에서 표시되는 단위로 값 변환은 자동으로 수행된다.

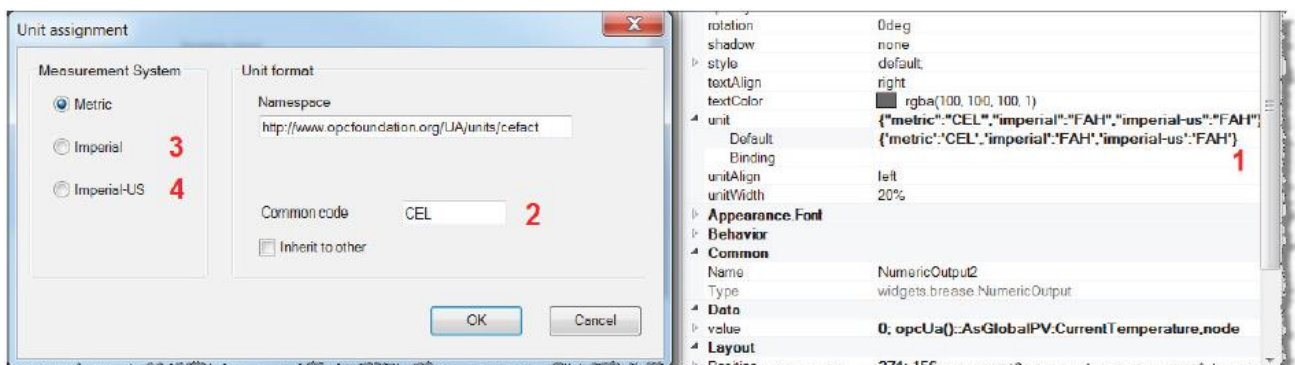


그림 89 단위 속성 정의하기

단위 속성은 언제나 반드시 다음과 같이 규정되어야 한다.:

```
{'metric':'<CommonCode>','imperial':'<CommonCode>','imperial-us':'<CommonCode>'}
```

예시에서 사용된 단위 문자열은 다음과 같다:

```
{'metric':'CEL','imperial':'FAH','imperial-us':'FAH'}
```

측정 시스템 선택기(MeasurementSystemSelector) 위젯 추가하기

OPC UA 노드 단위가 설정되고 측정 시스템을 위해 표시되는 단위가 정의된 후, 측정 시스템의 전환 가능성을 반드시 고려해야 한다. 단위를 전환하려면, Object Catalog의 “MeasurementSystemSelector” 위젯을 “메인 페이지(MainPage)” 콘텐츠에 배치한다.

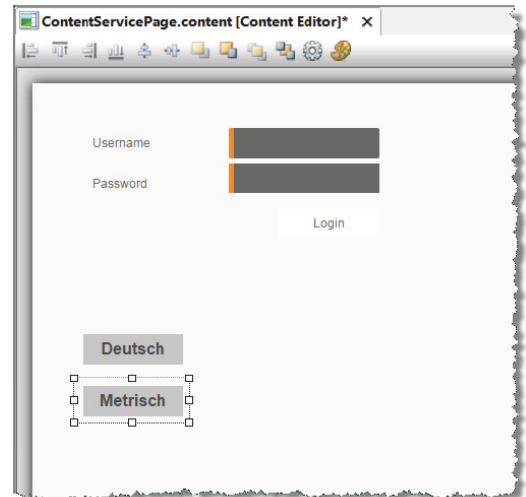


그림 90 메인페이지(MainPage)에 측정 선택기(MeasurementSelector) 배치하기

예제: 수치 입력(NumericInput) 위젯의 단위로 설정 온도(SetTemperature) 노드 표시하기

프로젝트를 컴파일하기 전에, OPC UA 노드 “설정 온도(SetTemperature)” 또한 °F 로 전환될 수 있도록 다른 “NumericInput” 위젯에서 단위 °C 로 표시될 수 있어야 한다.

반드시 다음 작업이 완료되어야 한다:

- 1) OPC UA 노드 (SetTemperature) 단위를 “섭씨 온도(degrees Celsius)”로 설정
- 2) “NumericInput” 위젯을 추가하고 노드 바인딩(node binding) 선택
- 3) 단위 전환 설정
- 4) 프로젝트를 컴파일하고 브라우저에 표시

예상되는 결과

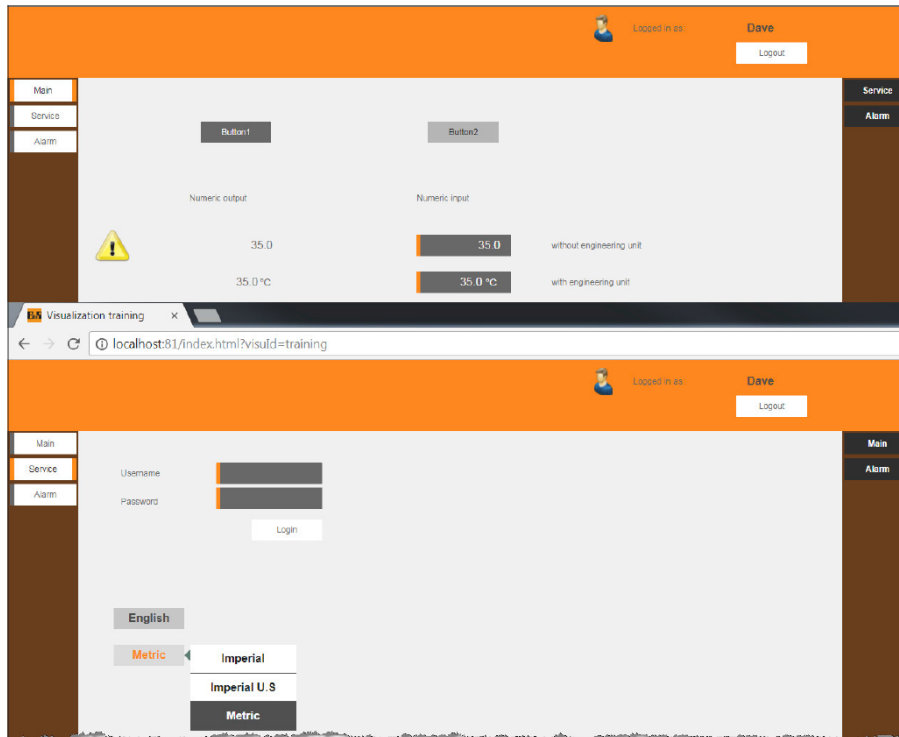


그림 91 대응되는 단위가 브라우저에 표시된 HMI 어플리케이션

9 이벤트(events)와 액션(actions)

HMI 어플리케이션 행동을 설정하기 위해서 우리는 이미 데이터 바인딩(data binding)과 내비게이션에 친숙하다. 내비게이션은 사용자가 HMI 어플리케이션에서 다양한 페이지를 이동하는 방법을 결정한다. 데이터 바인딩은 자동화 어플리케이션과 HMI 어플리케이션 사이에서 데이터가 어느 방향으로 교환되는지를 결정한다.

일부 HMI 어플리케이션에 대해서는, 이 옵션들이 충분하지 않을 수 있다. mapp View 는 HMI 어플리케이션의 행동을 개별적으로 설정하기 위해 이벤트와 액션을 제공한다.

이벤트는 응답을 촉발하는 데 이용될 수 있는 사건이다. 액션은 이벤트에 의해 촉발된 응답이다. 이벤트와 그 뒤를 이은 액션 조합은 이벤트 처리(event handling)라고 불린다.

9.1 개요 – 이벤트(Event)

이벤트는 응답을 촉발하기 위해 이용될 수 있는 사건이다. mapp View 는 HMI 어플리케이션 개발자에게 여러 가지 유형의 이벤트를 제공한다.

OPC UA 이벤트(OPC UA events)

mapp View 는 자동화 어플리케이션에서 값 변경에 관한 정보를 제공하는 이벤트를 정의한다. 이러한 목적을 위해, mapp View 는 “ValueChanged”라는 이름에 이벤트 유형 “opcUa.Event”를 제공한다.

위젯 이벤트(Widget events)

위젯 이벤트는 특정한 위젯 인스턴스의 사건 정보를 제공한다. 각 다른 유형의 위젯은 각 다른 사건에 관한 정보를 이벤트 형태로 제공한다. 예시는 “Button” 위젯의 “클릭(Click)” 이벤트이다.

세션 이벤트(Session events)

mapp View 는 클라이언트 세션의 사건을 검출하기 위한 이벤트를 정의한다. 세션 이벤트는 mapp View 고급 과정 교육에서 다룬다.

9.2 개요 – 액션(Action)

액션(action)은 이벤트에 의해 촉발된 응답이다.

액션은 제공자에 의해 함께 그룹화된다. mapp View 에는 다음과 같은 액션 제공자가 존재한다.

- OPC UA
- 세션(Session)
- 클라이언트(Client)
- 위젯(Widgets)

각 제공자는 한 개 또는 그 이상의 액션 그룹을 정의한다..

제공자 “OPC UA”는 그룹 “opcUa.NodeAction”을 제공하고, 제공자 “위젯(Widgets)”은 그룹 “widgets.browse.<WidgetType>”을 제공한다.

제공자 “세션(Session)” 및 “클라이언트(Client)”는 mapp View 고급 과정 교육의 주제이다. 이 교육 자료에서는, 제공자 “OPC UA” 및 “위젯(Widgets)”만 다룬다..

각 그룹은 한 개 이상의 액션을 정의한다..



[HMI application / mapp View / Engineering / Events and actions / Action](#)

9.2.1 OPC UA 액션(OPC UA actions)

OPC UA 를 위해, OPC UA 노드에 영향을 미치는 액션을 이용할 수 있다.



[HMI application / mapp View / Engineering / Events and actions / Action / INIT NC Actions / OPC UA actions](#)

9.2.2 위젯 액션(Widget actions)

위젯을 위해, 위젯 인스턴스에 영향을 미치는 액션을 이용할 수 있다.

각 위젯 유형에 특정한 액션이 있다. 각 위젯 유형에 사용 가능한 액션을 알아보려면, 위젯에 대한 문서를 참조하도록 한다.



[HMI application / mapp View / Engineering / Events and actions / Action / INIT NC Actions / Widget actions](#)

9.3 이벤트와 액션 이용하기

9.3.1 값 설정하기

이 예제의 목표는 “NumericInput” 위젯에 표시되는 “설정 온도(SetTemperature)” OPC UA 노드의 설정 값을 버튼을 클릭시 디폴트 값 (35)으로 재설정하는 것이다.

예제: OPC UA 노드 설정 값을 버튼 클릭으로 재설정하기

이 목표를 달성하기 위해, 적절한 이벤트를 이용하고 이벤트에 응답하는 적합한 액션을 설정한다. 반드시 다음 단계들이 수행되어야 한다:

- 1) 새로운 버튼(Button) 위젯을 추가하고 설정
- 2) 이벤트 바인딩(EventBinding) 파일을 추가하고 액션과 이벤트 정의
- 3) .vis 파일에서 이벤트 바인딩 파일 참조
- 4) 프로젝트를 컴파일, 전송하고, HMI 어플리케이션에 표시



HMI application / mapp View / Engineering / Events and action / Event

버튼 설정하기

먼저 드래그 앤 드롭에 의해 새로운 “Button” 위젯을 Object Catalog 에서 “메인 페이지(MainPage)”로 추가하면, 버튼 위젯에 대한 공통 명칭(CommonName)을 용이한 식별을 위해 “SetToDefault”으로 변경한다. 그 다음에는 버튼 텍스트가 독일어와 영어로 표시될 수 있도록 현지화한다.

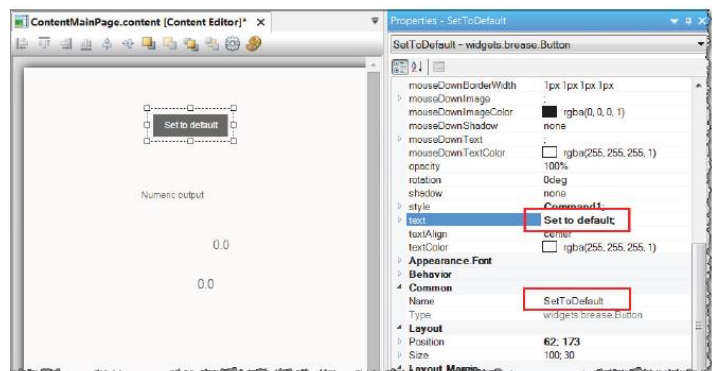


그림 92 버튼 설정하기

이벤트 바인딩(EventBinding) 파일을 추가하고 액션과 이벤트 정의하기

이벤트와 이를 뒤따르는 액션은 이벤트 바인딩 파일에 정의된다. Automation Studio 도움말에서, 다양한 문자열과 해당 위젯에 사용되는 모든 가능한 파라미터 (예, 이벤트 유형 “Click”)가 정의된다.



HMI application / mapp View / Widgets

컨피규레이션 뷰에서 mapp View 노드를 선택한 후, Object Catalog 에서 “이벤트 바인딩 파일(event binding file)”을 추가할 수 있다. 파일을 더블 클릭하면 XML 편집기가 열리고 첫 단계로 고유 “ID”를 (=“EventBinding”) 입력할 수 있다.

다음 단계에서, “버튼(Button)” 위젯을 클릭할 때 촉발되는 이벤트를 소스로 정의한다.

먼저, 각 위젯 속성에서 볼 수 있는 이벤트 유형이 정의되며, 이것은 다음과 같이 구성된다:

```
widgets."Namespace"."WidgetTyp".Event
```

그러면 어느 위젯 인스턴스 (CommonName)가 참조되는지와 그것이 위치하는 콘텐츠를 규정할 필요가 있다.

위젯 인스턴스(widget instance)는 widgetRefId="" 속성을 이용하여 규정된다.

콘텐츠(content)는 contentRefId="" 속성을 이용하여 규정된다.

마지막으로 응답을 촉발할 수 있는 이벤트(event)는 event="" 속성을 이용하여 규정된다.

“Button” 위젯을 클릭할 때의 예시는 아래와 같다.

```
<Source xsi:type="widgets.brease.Button.Event"
  contentRefId="ContentMainPage" widgetRefId="SetToDefault"
  event="Click" />
```

이벤트가 정의된 후, 반드시 뒤따르는 액션을 정의해야 한다. Automation 도움말에는 선택할 수 있는 파라미터가 정의되어 있다.



HMI application / mapp View / Engineering / Events and action / Actions / OPC UA actions

액션의 입력은 “opcUaNodeAction” 속성과 변경되어야 하는 글로벌 변수 (SetTemperature)로 구성된다.

정의는 아래와 같다.:

```
<Action>
<Target xsi:type="opcUa.NodeAction" refId="::AsGlobalPV:SetTemperature">
```

로컬과 글로벌 변수에 대한 refId 설정에 관한 정보는 “OPC UA 변수 다루기(Addressing OPC UA variable)”의 도움말에서 찾아볼 수 있다.

마지막 단계에서, 실행되어야 하는 액션 요소는 <Method>에 정의된다.

문자열은 가능한 액션과 (NodeAction) 설정되어야 하는 값으로 구성되고, 아래와 같다.

```
<Method xsi:type="opcUa.NodeAction.SetValueNumber" value="35" />
```

전체 이벤트 바인딩은 아래와 같다.:

```
<?xml version="1.0" encoding="UTF-8"?>
<EventBindingSet id="EventBinding"
xmlns="http://www.br-automation.com/iat2014/eventbinding/v2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Bindings>
    <EventBinding>
      <Source xsi:type="widgets.brease.Button.Event"
        contentRefId="ContentMainPage" widgetRefId="SetToDefault"
        event="Click" />
    <EventHandler>
      <Action>
        <Target xsi:type="opcUa.NodeAction"
          refId="::AsGlobalPV:SetTemperature ">
        <Method xsi:type="opcUa.NodeAction.SetValueNumber" value="35" />
        </Target>
      </Action>
    </EventHandler>
  </EventBinding>
</Bindings>
</EventBindingSet>
```

.vis 파일에서 이벤트 바인딩 참조하기

이벤트와 액션이 정의된 후, 이벤트 바인딩도 반영되도록, 입력된 EventBindingSet Id(=EventBinding)를 반드시 .vis 파일에서 참조해야 한다.

.vis 파일을 더블 클릭하면 바인딩 요소의 코멘트 아웃과 “ID” 입력이 허용된다.

다양한 역할로 HMI 어플리케이션 테스트하기

수치 입력(numeric input) 위젯의 값은 정의된 파라미터 (OPC UA 디폴트 뷰)와 로그인한 역할/사용자에 대한 허가가 충족되는 경우에만 설정된다.

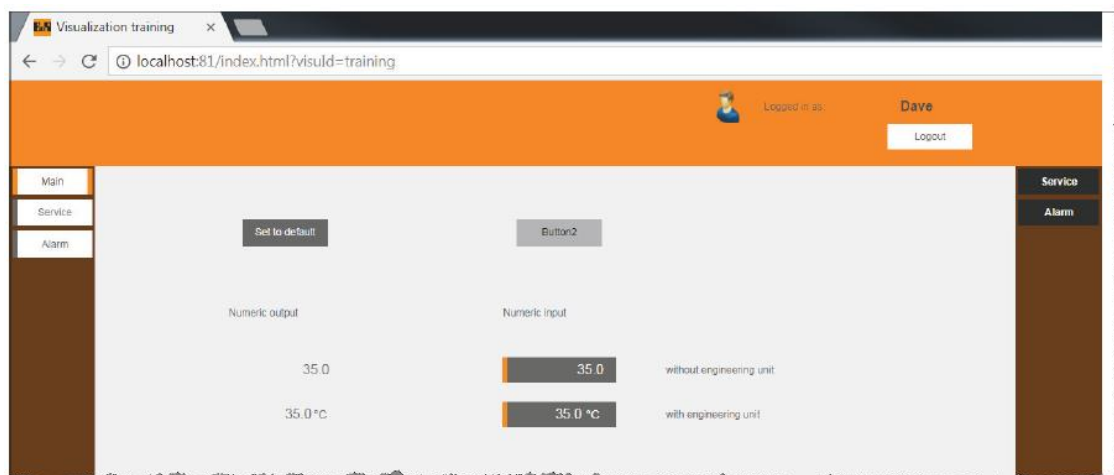


그림 93 사용자 Dave 로 로그인한 상태에서 HMI 어플리케이션

9.3.2 위젯 표시와 감추기

이 예제의 목표는 “CurrentTemperature” OPC UA 노드가 특정 값에 도달했을 때 (이벤트) “MainPage” 콘텐츠에 이미지를 표시하는 것 (액션)이다.

예제: 특정한 값에 도달할 때 이미지를 표시하고 숨기기

이 목표를 달성하기 위해 메인 페이지(MainPage) 콘텐츠에 이미지를 추가하고 가시성(visibility)을 “false”로 설정한다. 그 다음, 값에 도달할 때 이벤트와 이미지 표시를 정의하는 액션을 이벤트 바인딩(EventBinding)에 정의한다.

여기에서 반드시 다음 단계들을 수행해야 한다:

- 1) 이미지를 추가하고 설정
- 2) 이벤트 바인딩(event binding)에 이미지 표시를 위한 이벤트와 액션 정의
- 3) 이벤트 바인딩에서 이미지를 숨기기 위한 이벤트와 액션 정의
- 4) 프로젝트를 컴파일, 전송하고, HMI 어플리케이션에 표시

이미지 추가하고 설정하기

“Image” 위젯을 Object Catalog 에서 드래그 앤 드롭을 통해 “메인 페이지(MainPage)” 콘텐츠에 배치한다. 그 다음에는 이미 저장된 “warning” 이미지를 이미지 위젯에서 참조한다. 이벤트에 의해 촉발될 때만 이미지가 보이도록 하기 위해 “가시성(visible)” 속성을 반드시 “false”로 설정한다. 위젯에 대한 공통 명칭(CommonName)은 반드시 용이한 식별을 위해 “ImageWarning”로 변경한다.

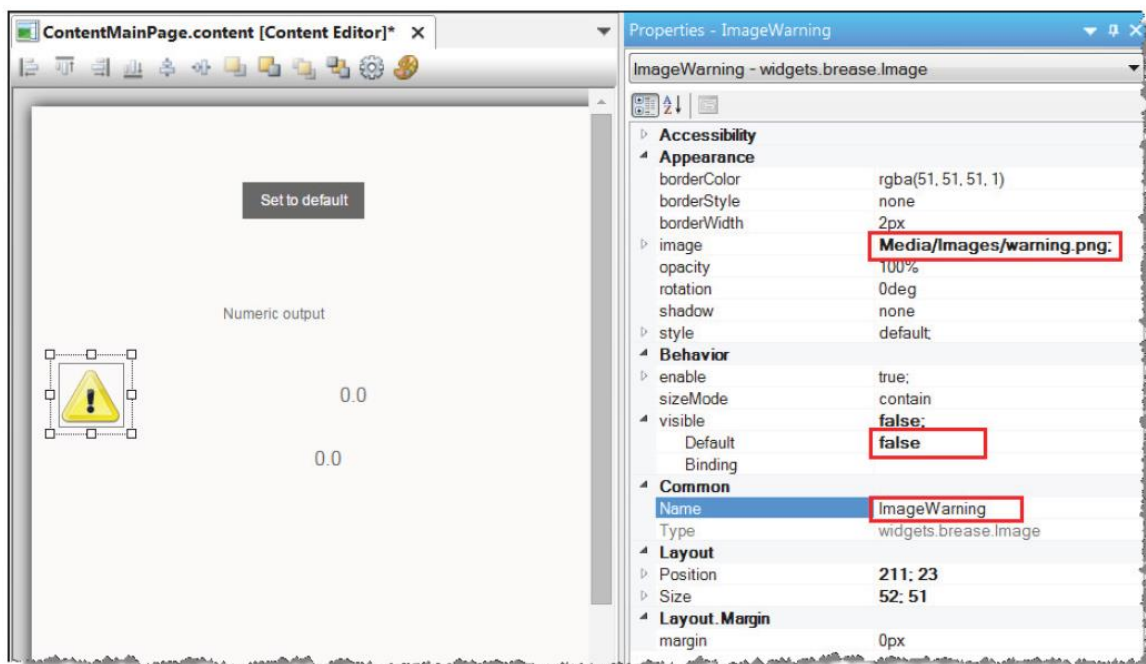


그림 94 Image 위젯 속성

이미지 표시를 위한 이벤트와 액션 정의하기

이벤트와 이를 뒤따르는 액션은 이벤트 바인딩 파일에 정의된다.

이벤트 바인딩(EventBinding) 파일은 “값 설정하기” 예제에서 추가되었고 .vis 파일에서 참조되었다.

먼저 “CurrentTemperature” OPC UA 노드 값이 “45”에 도달할 때의 이벤트를 정의한다. 이 이벤트는 “OPC UA”에 대한 것이다. 그러므로 이벤트 유형은 “opcUA.Event”이다. 이벤트 바인딩(EventBinding) 요소 < Source>에 대한 입력은 아래와 같다.:

```
<Source xsi:type="opcUa.Event"
refId="::AsGlobalPV:CurrentTemperature" event="ValueChanged"/>
```

문자열 요소(String) <(Source)>에서, 대응되는 OPC UA 노드와 “ValueChanged event”의 “refId”는 xsi:type 에 추가하여 규정한다.

이벤트 핸들러(EventHandler)는 이벤트에 대한 응답을 규정한다. 이벤트 핸들러(EventHandler)에 대해, 설정된 액션이 실행되기 위해 반드시 충족되어야 하는 조건을 정의할 수 있다.

```
<EventHandler condition="newValue > 45">
```

이제 이벤트에 대한 응답으로써 액션을 정의한다.

요소는 액션이 적용되는 오브젝트를 정의한다. 우리의 예시에서, 액션은 콘텐츠에 배치된 위젯에 적용된다.

```
<Target xsi:type="widgets.brease.Image.Action"
widgetRefId="ImageWarning" contentRefId="ContentMainPage">
```

이미지의 “가시성(visible)” 속성을 “false”에서 “true”으로 변경하려면, xsi:type 이외에도 “Method” 요소의 “value” 속성을 “true”로 설정한다.

```
<Method xsi:type="widgets.brease.Image.Action.SetVisible"
value="true" />
```

완성된 이벤트 바인딩(EventBinding)은 아래와 같다:

```
<EventBinding>
  <Source xsi:type="opcUa.Event"
    refId="::AsGlobalPV:CurrentTemperature" event="ValueChanged"/>
  <EventHandler condition="newValue >= 45">
    <Action>
      <Target xsi:type="widgets.brease.Image.Action"
        contentRefId="ContentMainPage" widgetRefId="ImageWarning">
        <Method xsi:type="widgets.brease.Image.Action.SetVisible"
          value="true" />
      </Target>
    </Action>
  </EventHandler>
</EventBinding>
```

이미지를 숨기기 위한 이벤트와 액션

값이 “45” 미만으로 떨어질 때 “warning” 이미지를 다시 숨기려면, 또 다른 이벤트와 그에 따르는 액션을 반드시 정의해야 한다.

이미지를 숨기기 위해, 이미지를 표시하기 위한 전체 이벤트 바인딩(EventBinding)을 복사하고, 조건 파라미터를 “>”에서 “<”로 변경하고, “SetVisible” 액션의 “value”를 “false”으로 변경한다.

최종 결과:

```
<EventBinding>
  <Source xsi:type="opcUa.Event"
    refId="::AsGlobalPV:CurrentTemperature" event="ValueChanged"/>
  <EventHandler condition="newValue &gt;= 45">
    <Action>
      <Target xsi:type="widgets.brease.Image.Action"
        contentRefId="ContentMainPage" widgetRefId="ImageWarning">
        <Method xsi:type="widgets.brease.Image.Action.SetVisible"
          value="true" />
      </Target>
    </Action>
  </EventHandler>

  <EventHandler condition="newValue &lt; 45">
    <Action>
      <Target xsi:type="widgets.brease.Image.Action"
        contentRefId="ContentMainPage" widgetRefId="ImageWarning">
        <Method xsi:type="widgets.brease.Image.Action.SetVisible"
          value="false" />
      </Target>
    </Action>
  </EventHandler>
</EventBinding>
```

예상되는 결과:

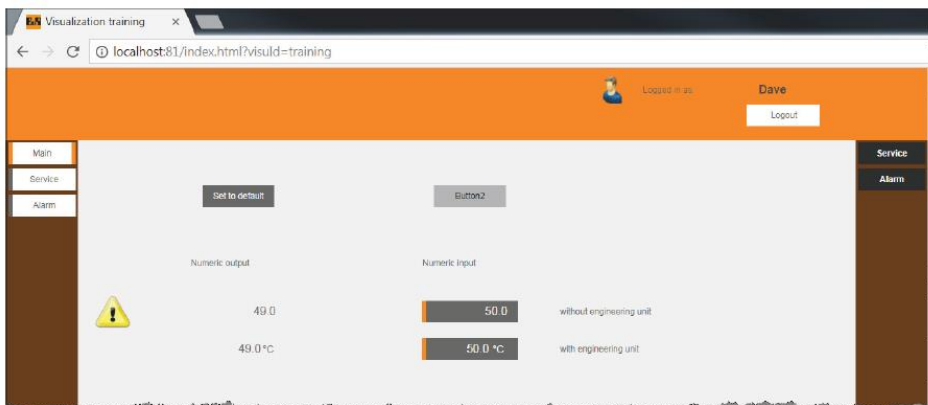


그림 95 이미지가 표시된 상태의 브라우저의 HMI 어플리케이션

10 운영 시스템에 적용하는 방법

이번 장에서는 어떻게 제어기에 mapp View 서버를 설치하고 터미널(terminal)에 mapp View 클라이언트를 설치하는지에 대한 자세한 방법에 대해 설명한다.

10.1 제어기에 mapp View 서버 운영하기

시뮬레이션 모드 종료

제어기에서 mappView 서버(mapp View server)를 시작하기 위한 첫번째 단계로 Automation Studio 의 시뮬레이션 모드를 종료해야 한다.

CPU 시뮬레이션이 운영중일 때, Automation Studio 상태 바는 아래와 같이 나타낸다.

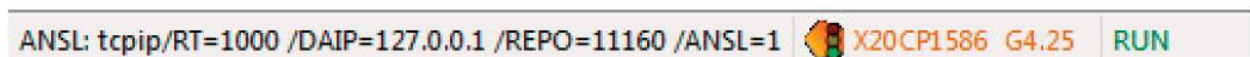
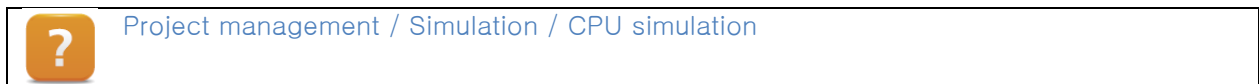


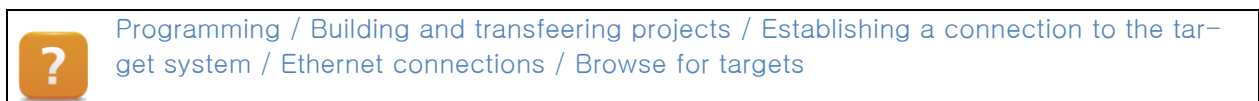
그림 96 Automation Studio 상태 바 - 시뮬레이션 구동중



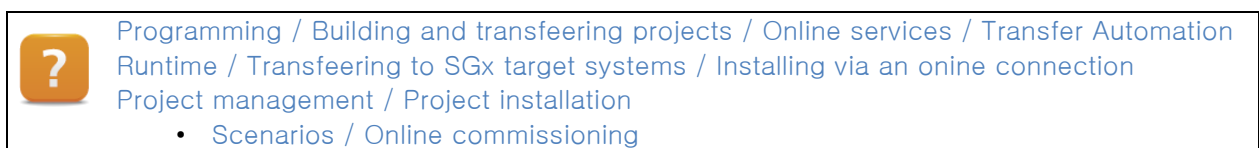
프로젝트 설치

프로젝트를 컴파일하고 제어기에 전송한다.

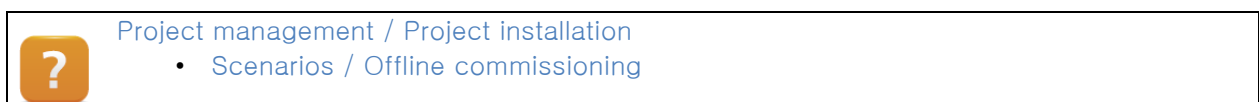
제어기와 연결은 Automation Studio, “Browse for target system”을 이용하여 구성한다. 이 기능은 네트워크에 존재하는 B&R 제어기를 검색한다. 제어기 연결 설정은 검색 다이얼로그 박스에서 임시적으로 변경할 수 있다.



연결이 설정되면 Automation Runtime 을 전송하거나 온라인 설치를 수행 할 수 있다.



오프라인 설치를 위해, Automation Studio 는 설치 미디어(Compact Flash 또는 CFast 카드)를 생성한다. 설치 미디어를 타겟 시스템에 삽입한다. 전원이 켜지면, Automation Runtime 과 어플리케이션이 시작한다.



Automation Runtime 과 어플리케이션 소프트웨어는 프로젝트 설치 패키지(project installation package)에 포함되어 있고 USB 플래쉬 드라이버, CompactFlash 카드 또는 DHCP 서버를 통해 타겟 시스템에 전송한다.



Project management / Project installation

- Scenarios / Offline commissioning

Real-time operating system / Target systems / SG4 / Automation Runtime remote install

Programming / Libraries / Configuration, system information, runtime control / ArProject

화면 어플리케이션을 브라우저에 표시하기

연결 테스트를 위해서, 크롬 브라우저(Chrome)에 입력했던 이전 주소를 URL 에 입력한다. 여기에서는 아래에 있는 제어기 IP 주소를 사용한다:

예: `http://10.43.15.46:81/index.html?isuld=Training`

예제: X20 CPU 에서 mapp View 서버 운영하기

예제의 목적은 제어기에서 mapp View 서버를 운영하는 것이다.

- 1) 프로젝트 시뮬레이션 모드 종료
- 2) 프로젝트 빌드
- 3) 제어기에 프로젝트 전송
- 4) 화면 어플리케이션을 브라우저에 표시하기

10.2 T50 에서 mapp View 클라이언트 운영하기

Power Panel T50 은 서비스 매뉴 항목(Service pages) 이나 Automation Studio 프로젝트에서 구성할 수 있다.

서비스 매뉴 항목을 통한 구성

서비스 매뉴 항목을 통해서 구성을 정의할때, 필요한 모든 설정은 터미널로 직접 입력한다. 이 구성이 저장되고 터미널이 재시작한다.



Hardware / Power Panel / Power Panel T50 / Configuration / Service pages

10.2.1 Automation Studio 에서 구성

터미널 구성을 정의할 때, 먼저 하드웨어 카테고리 툴박스를 이용하여 제어기의 이더넷 인터페이스에 터미널을 추가해야한다.

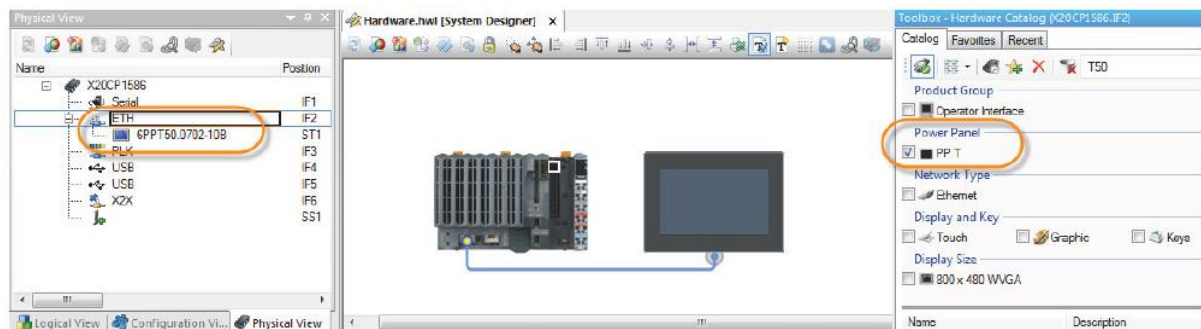


그림 97 Physical View 와 System Designer 에서 Power Panel T50

두번째 단계는 디바이스 설정이다. 각 장치의 설정 매뉴얼에 모든 파라미터가 설명되어있다. 예제 구성은 아래와 같다:

Name	Value	Description
6PPT50.0702-10B		
Operating system	PPT50 OS	
Startup		
Start mode	Web	
Network		
Mode	get IP address from DHCP server	
Hostname		
DNS parameters		
Time		
Screen		
Audio		
Gesture		
VNC		
Web		
Active web server	10.43.15.46:81/index.html?isuld=Training	IP address or hostname of server
Web server list		Selectable servers
Enable virtual keyboard	off	Virtual keyboard appears when clicking on a text input in web
Storage		
Update		
Security		

그림 98 mapp View 서버 연결을 위해 필요한 설정

예제 구성에서, 아래 파라미터를 설정하였다.

파라미터	값	설명
Start mode	Web	켜진후에, 디바이스가 즉시 설정된 웹 서버 연결 구성을 시도한다.
Network	해당 네트워크 인프라에 따른 설정	
Active web server	10.43.15.46:81/index.html?isuld=Training	
Enable virtual keyboard	Off	Mapp View 가상 키보드 보이기

표 20 터미널을 위한 예제 구성

	Hardware / Power Panel / Power Panel T50 / Configuration / Service pages	
	• Service page "Startup"	
	• Service page "Network"	
	• Service page "Web"	

10.2.2 터미널로 구성 전송

터미널 구성은 Automation Studio 에서 직접 제어기로 전송한다. Automation Studio 에서, 구성은 프로젝트 설치 패키지(Project installation package)로 만든 USB 플래쉬 드라이버로 복사하거나 직접 제어기에서 터미널로 불러들여진다.

프로젝트 설치 패키지(Project installation package) 생성

프로젝트 설치 패키지는 Project 메뉴에서 만든다. 다이얼로그 박스에서, 프로젝트 설치 패키지에 해당되는 디바이스를 선택하고 생성한다. 여기에서 터미널을 선택하였다.

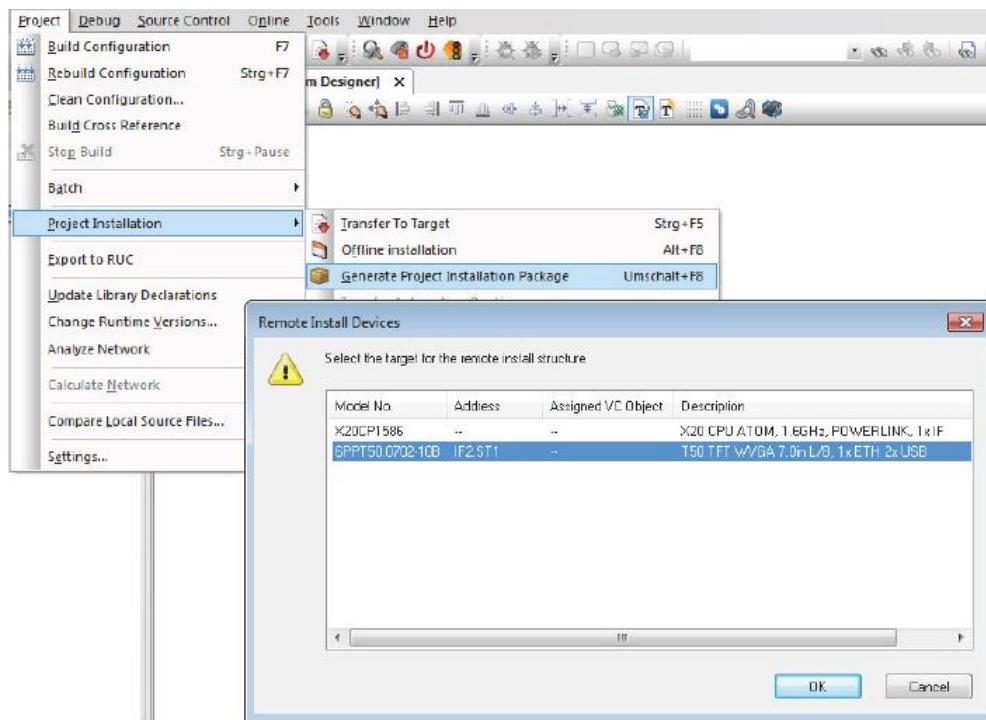


그림 99 터미널을 위한 프로젝트 설치 패키지 생성

터미널에서 구성 불러오기

터미널의 서비스 메뉴에서 USB 플래쉬 드라이버에서 프로젝트 설치 패키지를 삽입하거나 제어기에서 직접 불러온다.

USB 플래쉬 드라이버가 터미널에 연결되면, “Update” 서비스 페이지에서 “Load settings from USB”를 선택하여 구성을 불러온다. 완료 후에는 터미널이 재시작되며, 구성이 업데이트되고 디바이스는 구성된 모드로 시작한다.



Hardware / Power Panel / Power Panel T50 / Configuration / Service pages

- Service page “Update”
- Service page “Backup & reset”

예제: Power Panel T50 에서 mapp View 클라이언트 운영하기

mapp View 화면 어플리케이션을 접근하기 위하여 터미널을 설정할 것이다.

- 1) Physical View 또는 System Designer 에 터미널 추가하기
- 2) 디바이스 구성하기 (예제 구성 참조)
- 3) 프로젝트 설치 패키지(project installation package) 생성
- 4) 터미널 구성 업데이트
- 5) 화면 어플리케이션 테스트

10.2.3 Content caching 으로 성능 향상

터미널에 페이지를 요청할때, 각 페이지는 mapp View 서버로부터 배달되고 mapp View 클라이언트에서 구성된다. 불러오는 시간은 페이지 복잡성과 관련있다.

Mapp View 서버는 “content caching” 기능을 지원한다. 이미 불러온 페이지들은 클라이언트에 다시 보낼 필요가 없다. 이미 불러온 콘텐츠(Content)들은 클라이언트에서 메모리를 유지하고 있고 요청시 재 사용될 수 있다. 페이지 변환시 현저하게 성능이 향상된다.

Content caching 설정은 고급 mapp View 서버 설정에서 이루어진다.

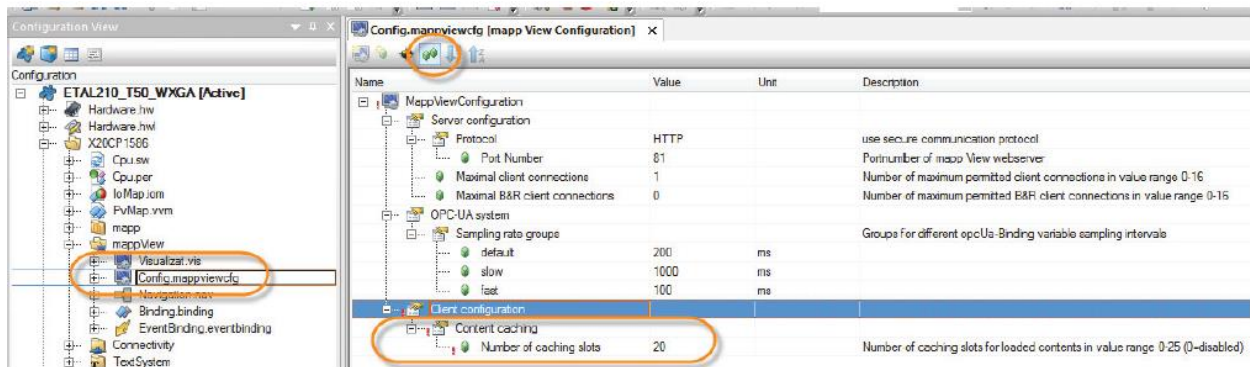


그림 100 mapp view 구성 - content caching 활성화



HMI application / mapp View / Engineering / HMI organization / configuration View / mapp View configuration

- Client configuration / Caching strategy

11 요약

이 교육을 이수하면, 다수의 HMI 페이지가 있는 mapp View HMI 어플리케이션을 작성할 수 있다. HMI 페이지의 내비게이션은 수동과 자동으로 설정할 수 있다. 데이터 입력뿐만 아니라 자동화 어플리케이션의 값 표시도 가능하다. 이렇게 사용할 때, 자동화 어플리케이션에 정의된 값 범위 한도를 HMI 어플리케이션에서 자동적으로 점검할 수 있다. 특정한 역할에 대해 상이하게 정의된 값 범위 한도 점검도 가능하다. HMI 어플리케이션의 특정 역할에 따른 접근 제한 및 HMI 사용자 인증을 위해 Automation Studio의 사용자 역할 시스템을 이용할 수 있다. 개별적인 요구에 따라 HMI 어플리케이션의 시각적 외관을 적용시키기 위해서, 테마(theme)와 스타일(style)을 이용하여 위젯과 미디어를 수정할 수 있다. 텍스트와 단위는 현지화 기능(localization)을 통해 언어 및 문화적 조건에 적용시킬 수 있다. 이 교육 자료에서는 이벤트에 대한 응답으로 액션 정의를 다루었다.



그림 101 mapp View 안경

세미자 참여자들은 이제 mapp View 를 이용하여 간단한 HMI 어플리케이션을 작성할 수 있다. 더 복잡한 HMI 어플리케이션은 “mapp View 고급과정(mapp View Advanced)” 교육 자료에서 다룬다.

Automation Academy 에서 제공하는 것

우리는 고객뿐만 아니라 직원을 대상으로 한 교육 과정을 제공합니다.

Automation Academy 에서, 당신은 필요로 하는 능력을 즉시 향상시킬 수 있습니다.

자동화 엔지니어링 분야에서 필요로 하는 지식증진을 위해 세미나가 준비되어 있습니다. 한번 이수하면, 당신은 B&R 기술을 이용하여 능률적인 자동화 솔루션을 개발하는 위치에 있을 것입니다.

이를 통해 귀하와 귀사는 끊임없이 변화하는 시장 수요에 보다 빠르게 대응할 수 있게 됨으로써 결정적인 경쟁 우위를 확보 할 수 있습니다.



세미나



품질 및 관련성은 세미나의 필수 구성 요소입니다. 특정 세미나의 페이스는 엄격하게 코스 참가자가 직면한 요구 사항과 경험에 근거합니다. 그룹 스터디와 자율 학습에 조합은 학습 경험을 극대화하는데 필요한 높은 수준의 유연성을 제공합니다. 각 세미나는 숙련된 경험이 풍부한 강사 중 한 명이 진행합니다.

교육 자료(Training module)

교육 자료는 세미나뿐만 아니라 자율 학습을 위한 기초를 제공합니다. 컴팩트 모듈은 일관된 교육 개념에 의존합니다. 상황식 구조는 복잡하고 상호 연관된 주제를 효율적이고 효과적으로 배울 수 있습니다. 광범위한 도움말 시스템이 가장 좋은 보완책입니다. 교육 자료는 다운받을 수 있으며 인쇄된 버전으로 주문할 수 있습니다.

카테고리 주제:

- 제어 기술(Control technology)
- 모션 제어(Motion control)
- 세이프티 기술(Safety technology)
- 화면작화(HMI)
- 프로세스 컨트롤(Process control)
- 진단 및 서비스(Diagnostics and service)
- 파워링크와 오픈세이프티 (POWERLINK and openSAFETY)

ETA 시스템(ETA system)



ETA 시스템(ETA system)은 훈련, 교육 및 실험실에서 사용하기 위해 실제와 같은 구조를 제공합니다. 두가지 이상의 다른 기구 구조가 선택될 수 있습니다. ETA light system 은 높은 자유도, 공간 절약 및 연구소 작업에 적합합니다. ETA standard system 은 튼튼한 기구 구조와 사전에 와이어링된 센서와 액추에이터를 포함합니다.

더 알아보기!

추가적인 교육이 필요하시나요? B&R Automation Academy 가 제공하는 것에 흥미가 있으신가요? 맞게 찾아오셨습니다.

상세한 정보는 아래 링크에서 확인하실 수 있습니다:

www.br-autoation.com/academy

