

TM671 강력한 mapp View 화면

I 버전 정보

버전	날짜	수정내역	번역	검수
1.0	2018.01.03	첫번째 버전 TM671TRE.433-ENG (V2.0.0.3)	Eun Im	Eun Im
		4.2.3 바인딩 제거 4.3.1 Page editor 와 Area editor 가 개발중인 관계로 매뉴얼 타입핑 추가		
		전반적인 help 주소 변경		

Table 1: Versions

선행 및 필요 조건

교육 자료	TM611 - Working with mapp View	
소프트웨어	TM611 에서 작업한 프로젝트 파일 필요	
	Automation Studio 4.3.3	
	Automation Runtime 4.33	
	Technology Package – mappView 5.0	
하드웨어	ARsim X20 CPU with ETAL611.1T10-1 https://www.br-automation.com/en/academy/eta-system/	

Ⅱ 목차

1 소개	1
1.1 학습목표	1
1.2 안전 표시와 무호	1
2 다이어르그아메시지 바스(Dialog and mossage boyes)	2
	ב
2.1 디어글포크과 베까지 국은 개효	ב כ
2.2 디어글포크 ㄱ드 제ㄱ피 르기	ב כ
2.2.1 레이아 ((ayou)) 제국	0
2.2.3 다이얼로그 박스(dialog box) 제작	5
2.2.4 HMI 어플리케이션 (.vis)에 다이얼로그 박스 레퍼런스 잡기	6
2.2.5 다이얼로그 박스 열기용 이벤트 바인딩(Event binding)	6
2.2.6 다이얼로그 박스 닫기용 이벤트 바인딩(Event binding)	8
2.3 메시지 박스 개요	9
2.3.1 메시지 박스 타입	9
2.3.2 메지지 약스 디던 값	10
2.5.5 이글디게이신 경국 2.4 메시지 반스를 역고 화이 평가	10
2.4 메시지 박스를 열기 위한 버튼 위젯	10
2.4.2 이벤트 바인딩 생성(event binding)	11
3 역할과 권리 위젯 사용하기	. 14
3.1 역할과 권리 위젯 개요	. 14
3.2 역할 기반 위젯의 가시성과 유효성	. 14
3.2.1 "permissionView" 속성에 역할 구성하기	14
3.2.2 역할-기반 위젯 구동	16
4 HMI 어플리케이션에서 애니메이션	18
4.1 간단한 위젯의 가시성 제어	. 18
4.1.1 Visible 속성에 바인딩	19
4.2 다수 프로세스 값을 사용하여 가시성 제어하기	. 20
4.2.1 익스프레션 타입(expression type) 구성	21
4.2.2 익스프레션(expression) 구성	22
4.2.3 익스프레션의 피연산자를 OPC UA 변수에 바인딩하기	23
4.2.4 가시성 옥성에 익스프레션 결과 바인당하기	23
4.3 페이지의 매경 이미지	. 24
4.5.1 MainFage의 매장 아마지 데피먼스 설정 시시 의제이 새사 변경	20
4.4 기 것과 국장 단장	. 29
4.4.2 스타일을 이벤트 바인딩으로 변경하기	
4.5 컨테이너 위젯에 상대적이고 절대적인 위치	. 32
4.5.1 그룹박스(GroupBox)추가 및 구성	32
4.6 브라우저 윈도우 사이즈에 따른 HMI 어플리케이션 스케일링	. 34
5 다른 변수 타입을 포함한 테스크	35
5.1 다른 위젯과 정보를 교환	. 36

5.1.3 콘텐츠 보더를 통한 위젯 정보 교환	39
5.1.4 Session variable	39
5.2 사용자 변경 및 페이지 전환	41
5.3 시스템 변수(system variables) 사용하기	43
6 텍스트 시스템(text system) 연습	45
6.1 색인 테스트 출력하기	45
6.1.1 텍스트 파일에 텍스트 리스트 구성하기	46
6.1.2 Snippet 생성하기	46
6.1.3 Snippet 을 위해 session variable 바인딩 - snippet 을 TextOutput 위젯에 설정	47
7 XML로 콘텐츠 엔지니어링(Content engineering in XML)	49
7 1 위젠 추가하기	49
7.2 Container 위제 초가하기	50
	00
8 먹티-큭라이어트 먹티-사용자	51
	OT
8.1.1 트저 OPC UN 변스에 대하 크라이어든 조소 바이다	54
8.1.2 =	56
8 1 3 로그아운 커메드 수신을 위한 session variable 바이팅	57
0 001	58
υ μη	50
Automation Acadamy 에너 제고치는 거	50
	59

1 소개

mapp View 는 표준 컴포넌트를 활용한 쉬운 구성과 다양한 조합을 포함한다. 교육 자료의 예제는 mapp View 에 기본 지식을 심도있게 디자인 했을 때를 고려하였다. 추가적인 정보, 구성 옵션과 다양한 위젯 구성은 새로운 HMI 어플리케이션 디자인 진입을 가능하게 한다.



1.1 학습목표

교육자료는 mapp view 의 기본 지식 확장하고 강화하는데 예제를 사용할 것이다.

- 다양한 경우에 이벤트와 액션(event and action)을 어떻게 사용하는지 배울 것입니다.
- Automation Studio 의 역할과 권리 시스템(roles and rights system)운영과 가시성을 위한 위젯을 사용할 것이다.
- 에니메이션 위젯을 위한 절차와 옵션을 배울 것입니다.
- 해당 테스크를 위한 mapp View HMI 어플리케이션에서 다른 변수 타입으로 확장하는 것을 배울 것이다.
- 텍스트 시스템(text system)과 관련된 추가 예제를 실습할 것이다.
- XML 파일에서 콘텐츠 엔지니어링(content engineering) 개요를 배울 것이다.
- 다수의 HMI 어플리케이션 실행 개요를 배울 것이다.

1.2 안전 표시와 부호

별도 명시가 없는 한 "TM210 - Working with Automation Studio"에 열거 된 안전 표시와 부호가 적용된다.

2 다이얼로그와 메시지 박스(Dialog and message boxes)

다이얼로그 박스와 메시지 박스는 HMI 어플리케이션의 요소이자 페이지 위에 출력될 수 있다.

2.1 다이얼로그와 메시지 박스 개요

다이얼로그 박스는 페이지와 유사하게, HMI 어플리케이션 개발자가 제작하는 창(window)이며, 영역(area)에서 콘텐츠(content)를 레퍼런스 잡고 레이아웃(layout)을 이용하여 디자인한다.

다이얼로그 박스의 콘텐츠에 모든 위젯을 배치하거나 구성할 수 있다.



그림 1 다이얼로그 박스 예시

다이얼로그 박스의 어플리케이션 영역

다이얼로그 박스는 사용자의 입력을 수집하여 출력한다. 모덜 다이얼로그 박스(Modal dialog box)는 다이얼로그 박스가 표시 되는 동안 나머지 사용자 인터페이스를 잠근다. 모덜리스 다이얼로그 박스(Modeless dialog box)는 지속적인 상호작용을 허용한다.

2.2 다이얼로그 박스 제작과 열기

예제의 목표는 HMI 어플리케이션에 사용자 입증을 위한 다이얼로그 박스를 만드는 것이다. 다이얼로그 박스를 위해서, 영역(area)을 가진 레이아웃(layout)이 해당 레이아웃 사이즈의 콘텐츠(content)에 할당되어있다.

예제: 다이얼로그 박스 제작

아래 설명을 따라하여, 로그인 위젯과 다이얼로그 박스를 숨기는 버튼 위젯을 다이얼로그 박스에 추가한다. 다이얼로그 박스는 헤더에 이미지를 클릭하면 열린다.



6) 다이얼로그 박스 닫기용 이벤트 바인딩(Event binding)



Visualization / mapp View / Engineering / Dialog boxes

2.2.1 레이아웃(layout) 제작

레이아웃은 로지컬 뷰(Logical view)에서 추가한다. 레이아웃 ID 는 "DialogLayout"이며 XML 편집기에서 width="450", height="300" 로 구성한다.

Property name	Value	
ld	DialogLayout	
Height	300	
Widh	450	

표 1 레이아웃 파일의 속성

레이아웃과 같은 사이즈로 영역(area)를 구성한다.

ld	Height	Width	Left	Тор
AreaDialog	300	450	0	0

표 2 영역(area)의 속성

Logical View 👻 🕈 🗙	Layouts::DialogLayout.layout [XML File]* ×
Image: Sector State Image: Sector Sta	<pre></pre>

그림 3 로지컬 뷰에 layout 파일을 추가하고 XML 파일 수정하기

마지막 결과

```
<?xml version="1.0" encoding="utf-8"?>
<Idef:Layout id="DialogLayout" height="300" width="450" xmlns:Idef="<u>http://www.br-automation.com/iat2015/layoutDefinition/v2</u>">
<Areas>
<Areas>
<Area id="AreaDialog" height="300" width="450" left="0" top="0" />
</Areas>
</Idef:Layout>
```

2.2.2 콘텐츠(content) 제작

"Dialog boxes / AreaConents"에 콘텐츠(content)를 추가한다. 로지컬 뷰에 콘텐츠 파일이 추가된 후, 이름을 "ContentLogin"으로 변경한다.



그림 4 Content 파일 추가

콘텐츠 수정

콘텐츠는 visual editor 에서 편집할 수 있다. 속성 창에서, 고유의 이름, 폭(width), 높이(height)를 입력한다.

Common		
Name	ContentLogin	
Property		
height	300	
width	450	

그림 5 ContentLogin 속성

로그인 위젯(login widget)과 버튼 위젯(button widget)을 콘텐츠에 배치한다. 다이얼로그 박스는 "ButtonClose"로 명명된 버튼을 선택하면 닫힐 것이다.

Properties - ButtonClose		🗕 🕯 🗙
ButtonClose - widgets.bre	ase.Button	*
21 21 I		
opacity rotation shadow > style > text textAlign textColor Appearance.Font Behavior Appearance.Font Behavior Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Appearance.Font Ap	100% Odeg none default: Close: center gba(100, 100, 100, 1) ButtonClose widgets.brease.Button 245; 178 100; 30	
	 Properties - ButtonClose ButtonClose - widgets.bree Departing the state of the stat	 Properties - ButtonClose ButtonClose - widgets.brease.Button apacity 100% opacity 100% rotation 0deg shadow none style default text text Close: textAlign center textAlign center textColor rgba(100, 100, 100, 10, 10, 10, 10, 10, 10, 1

그림 6 콘텐츠에 배치된 위젯

2.2.3 다이얼로그 박스(dialog box) 제작



다이얼로그 박스는 Object Catalog 에서 "Dialogs" 패키지에 있는 "Dialog"를 드레그-앤드-드롭 이나 더블 클릭하여 추가한다. "Dialogs" 패키지는 mappView viualization package 의 일부이다.

다음 단계는 다이얼로그 박스 이름을 "LoginDialog"라 명명하고 레이아웃(layout)과 콘텐츠(content)를 레퍼런스 잡는다.



다음 단계는 페이지 구성과 구별 할 수 없다. 다이얼로그 박스는 "Dialogs" 패키지에서 관리된다.

그림 7 dialog box 추가

다이얼로그의 고유한 식별을 위해서, 고유의 "ID"를 명시해야한다. 이전에 정의한 레이아웃의 ID 는 "DialogLayout"이다. 다음 영역과 콘텐츠는 "LoginDialog"에서 사용될 디자인이다.

Туре	baseContentRefld	areaRefld
Content	ContentLogin	AreaDialog

표 3 dialog box 의 속성

다음 XML 코드는 "LoginDialog.dialog" 파일에 기입될 내용이다:

xml version="1.0" encoding="utf-8"?
<pre><ddef:dialog id="LoginDialog" layoutrefid="DialogLayout" xmlns:ddef="<u>http://www.br-</u></pre></td></tr><tr><td>automation.com/iat2015/dialogDefinition/v2"></ddef:dialog></pre>
<assignments></assignments>
<pre><assignment arearefid="AreaDialog" basecontentrefid="ContentLogin" type="Content"></assignment></pre>

2.2.4 HMI 어플리케이션 (.vis)에 다이얼로그 박스 레퍼런스 잡기

컨피규레이션 뷰(Configuration View)에 있는 HMI 어플리케이션(.vis)에 다이얼로그 박스를 레퍼런스 잡고 HMI 어플리케이션의 일부로써 전달한다.

HMI 어플리케이션의 일부인 다이얼로그 박스는 <Dialogs>에서 정의한다.

다음은 예제에서 visualization object (.vis)에서 필요한 관련 콘텐츠를 보여준다.

```
<Dialogs>
<Dialog refld="LoginDialog" />
</Dialogs>
```

2.2.5 다이얼로그 박스 열기용 이벤트 바인딩(Event binding)

새로운 이벤트 바인딩 파일(event binding file)을 생성하고 visualization object (.vis) / <EventBindingsSets>에 레퍼런스 잡는다.

이벤트 바인딩 파일에 새로운 <EventBinding>을 생성한다.

소스는, ContentTop 에 있는 "Image" 위젯을 클릭 이벤트로 다이얼로그 박스가 나타난다.

콘텐츠(content)에 있는 "Image" 위젯의 widgetRefld 를 모른다면, 해당 콘텐츠를 열어서 위젯을 클릭하여 확인하라.

"Image" 위젯을 클릭하여 발생하는 이벤트는 아래와 같다:

<Source xsi:type="widgets.brease.lmage.Event" contentRefId="ContentTop" widgetRefId="Image1" event="Click" />

이벤트가 발생한 뒤, 추가 액션이 정의되야한다. 사용 가능한 속성들은 Automation Studio help 에 설명되어 있다.

다이얼로그 박스를 여는 구체적인 행동은 아래와 같다:

```
<Target xsi:type="clientSystem.Action">
<Method xsi:type="clientSystem.Action.OpenDialog" dialogId="LoginDialog" />
</Target>
```

다이얼로그 박스를 여는 전체적인 이벤트 바인딩은 아래와 같다:



예상되는 결과

Visualization training ×					
$\leftrightarrow \rightarrow c$	Collocalhost:81/index.html?visuId=trai	ning			
			🗾 Logged in as:	Anonymous	
				Logout	
Main				Service	
Service	Set to default	Button?		Alarm	
Alarm		La MANATAR.			
		Please Login	×		
	Numeric output				
	35.0	Username	ig unit		
	35.0°C	Password	init		
		Login			
183		Close			
1	معاملين المسابق مستعد والمستعد	the second and the se	and a second a second s	and the second state of th	

그림 8 브라우저를 통해 HMI 어플리케이션에서 사용자 이미지를 클릭하여 다이얼로그 박스를 띄웠다.

2.2.6 다이얼로그 박스 닫기용 이벤트 바인딩(Event binding)

이벤트 바인딩 파일에 새로운 <EventBinding>을 생성한다.

소스는, ContentLogin 에 있는 "ButtonClose" 버튼 위젯을 클릭 이벤트를 다이얼로그 박스를 닫는데 사용한다. "Button" 위젯을 클릭하여 발생하는 이벤트는 아래와 같다:

<Source xsi:type="widgets.brease.Button.Event" contentRefId="ContentLogin" widgetRefId="ButtonClose" event="Click" />

이벤트를 정의한 이후에, 추가 액션이 정의되어야 한다. 선택할 수 있는 파라미터는 Automation Studio help 에 나와 있다.

?

Visualization / MappView / Engineering / Events and actions / Action / Client system actions / CloseDialog

다이얼로그 박스를 닫는 액션은 아래와 같다.:

<Target xsi:type="clientSystem.Action"> <Method xsi:type="clientSystem.Action.CloseDialog" dialogId="LoginDialog" /> </Target>

다이얼로그 박스를 닫는 전체 액션은 아래와 같다:

<EventBinding> <Source xsi:type="widgets.brease.Button.Event" contentRefId="ContentLogin" widgetRefId="ButtonClose" event="Click" /> <EventHandler> <Action> <Target xsi:type="clientSystem.Action"> <Method xsi:type="clientSystem.Action.CloseDialog" dialogId="LoginDialog" /> </Target> </Action> </EventHandler> </EventBinding>

2.3 메시지 박스 개요

메시지 박스는 페이지를 통해 보인다. 사용자는 메시지 박스를 반드시 확인해야하며, 선택된 항목에 따라 추가적인 액션이 실행된다.

Reset SetTemperature
Are you sure?
OK Cancel
그림 9 MessageBox 예제
메시지 박스는 콘텐츠 영역, 헤더와 버튼 영역으로 구성되어 있다.
콘텐츠 영역은 보이는 메시지를 출력하고 옵션으로 이미지도 출력할 수 있다.
텍스트는 헤더에 출력할 수 있다.
버튼 영역은 한 개 또는 그 이상의 버튼을 포함하며 메시지 박스 타입에 따라 결정된다.
Header text Reset SetTemperature
Message text
Button types> OK Cancel

그림 10 message box 의 영역(area)

2.3.1 메시지 박스 타입

메시지 박스를 부를때, 어떤 버튼이 표시될지 유형이 시작된다. 기대되는 이벤트 결과 확정에 따라 타입이 선택된다.

메시지 박스 타입 (Message box type)	설명
AbortRetryIgnore	"Abort", "Repeat", "Ignore" 표시
OK	"OK" 버튼 표시
OKCancel	"OK", "Abort" 버튼 표시
RetryCancel	"Repeat", "Abort" 버튼 표시
YesNo	"Yes", "No" 버튼 표시
YesNoCancel	"Yes", "No", "Abort" 버튼 표시

표 4 메시지 박스 타입

2.3.2 메시지 박스 리턴 값

특정 메시지 박스 타입이 보이면, HMI 어플리케이션 조작자는 확정 이후를 기대할 것이다. 선택한 버튼에 따라서, 결과는 이벤트 바인딩(event binding)을 통해 평가 될 수 있고 희망하는 액션이 구성된다.

여기, 모든 버튼에 ID 가 할당되었으며 이를 통해 상태를 평가할 수 있다.

Button ID	Value
Yes	1
No	2
OK	4
Cancel	8
Abort	16
Retry	32
Ignore	64

표 5 Message Box 버튼 ID

2.3.3 어플리케이션 영역

메시지 박스는 사용자가 해당 입력을 전달했을때 출력된다. 메시지 박스는 확인이 필요하며, 사용자가 선택한 것에 따라서 프로세스 시퀀스에 영향을 준다. 다이얼로그 박스와 달리, 메시지 박스 구조는 사용자가 변경 할 수 없다. 메시지 박스는 시스템 컴포넌트이다.

2.4 메시지 박스를 열고 확인 평가

예제의 목표는 "OKCancle" 메시지 박스를 구성하고 특정 이벤트에서 메시지 박스가 열린다. "OK" 버튼을 누르면 정의된 값으로 OPC UA 변수가 변경된다.

메시지 박스는 이벤트 바인딩(event binding)으로 구성된다.

예제: "OKCancel" 메시지 박스 구성하기

다음은 "OK" 버튼을 클릭하여 OPC UA 변수 "SetTemperature"를 특정 값으로 변경하는 메시지 박스 구성 방법에 대해 설명한다. "Cancel" 버튼을 클릭하면 메시지 박스는 추가 액션 없이 닫힌다.

다음 단계가 필요하다:

- 1) ContentMainPage 에서 메시지 박스를 열기 위한 버튼 사용
- 2) 이벤트 바인딩 생성(event binding)
- 3) 메시지 박스 콜하기
- 4) OPC UA 액션을 사용하여 "OK" 버튼 클릭시 결과 구성하기



Visualization / mapp View / Engineering / Events and actions / Action / Client system actions / ShowMessageBox

2.4.1 메시지 박스를 열기 위한 버튼 위젯

"Button" 위젯은 ContentMainPage 에 위치하며 이는 메시지 박스를 열기 위해 사용된다. 버튼 이름은 "ButtonMsgBox"라 명명한다. 게다가, 버튼의 텍스트 라벨(text)을 정의한다.

ContentMainPage.content [Co	ntent Editor] ×	*	Properties - ButtonMs	gBox	▼ ₽ >
문 팬 릐 프 속 아 🗳 💁	🍋 🔩 🛞 🏉		ButtonMsgBox - widge	ets.brease.Button	+
		*	21		
Set	to default	C MsgBox C	mouseDownBackG mouseDownBorderd mouseDownBorderd b mouseDownImage mouseDownImage	roundGradi none Color rgba(51, 153, 255, 1) Widt 1 px 1px 1px Color rgba(0, 0, 0, 1)	
Numerie	authau é	Numeria input	mouseDownShadov mouseDownText mouseDownTextCo	w none ; olor rgba(255, 255, 255, 1) 100%	
Numeric	output		rotation shadow	0deg none Operate 1:	
	0.0	0.0	text textAlign textCaler	MsgBox; center	
	0.0	0.0	Appearance.Font Behavior	rgba(u, u, u, i)	
			Name Type	ButtonMsgBox widgets.brease.Button	=
			 Position Size Layout Margin 	62: 495 100: 30	

그림 11 메시지 박스를 열기 위한 Button

2.4.2 이벤트 바인딩 생성(event binding)

사용가능한 이벤트 바인딩 파일(event binding)에 새 <EventBinding>을 생성한다.

소스에, ContentMainPage 에 있는 "ButtonMsgBox"버튼 위젯의 클릭 이벤트에 따라 메시지 박스가 열리게 코딩한다.

"Button" 위젯 클릭에 따른 시작은 아래와 같다:

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

이벤트 정의 후, 추가 액션을 정의한다. 선택할 수 있는 파라미터는 Automation Studio help 에서 참고 할 수 있다.



Visualization / mapp View / Engineering / Events and actions / Action / Client system actions / ShowMessageBox

"OKCancel" 타입에 메시지 박스를 열기 위한 액션은 다음과 같다:

<Target xsi:type="clientSystem.Action"> <Method xsi:type="clientSystem.Action.ShowMessageBox" type="OKCancel" message="Are you sure?" header="Reset SetTemperature" icon="Warning"/> </Target> •

메시지 박스(message=) 와 텍스트 해더(header=)는 텍스트 시트템에 로컬 텍스트 ID 를 명시하여 레퍼런스 잡을 수 있다. (예: \$IAT/MessageText)

다음 단계는 버튼을 클릭 후 기대되는 액션 결과를 구성한다. OPC UA 변수 "::AsGlobalPV:SetTemperature"를 "30"으로 설정한다.

이를 위해, <ResultHeadler>와 <Result> 구성요소는 모든 버튼에 사용가능하며 각<Action>을 사용하여 구성할 수 있다. <ResultHeadler>에 button 의 id 가 4 인, "OK" 버튼이 반응할 때의 상태 반환 값을 넣는다.

```
<Result>

<ResultHandler condition="result = 4">

<Action>

<Target xsi:type="opcUa.NodeAction"

refId="::AsGlobalPV:SetTemperature" >

<Method xsi:type="opcUa.NodeAction.SetValueNumber" value="30" />

</Target>

</Action>

</ResultHandler>

</Result>
```

메시지 박스를 열기 위한 전체 이벤트 바인딩은 다음과 같다:

```
<EventBindina>
<Source xsi:type="widgets.brease.Button.Event"
  contentRefId="contentMainPage" widgetRefId="ButtonMsgBox" event="Click" />
 <EventHandler>
  <Action>
       <Target xsi:type="clientSystem.Action">
         <Method xsi:type="clientSystem.Action.ShowMessageBox"</pre>
              type="OKCancel"
              message="Are you sure?"
              header="Reset SetTemperature"
              icon="Warning"/>
       </Target>
       <Result>
        <ResultHandler condition="result = 4">
           <Action>
                <Target xsi:type="opcUa.NodeAction"
               refld="::AsGlobalPV:SetTemperature" >
                      <Method xsi:type="opcUa.NodeAction.SetValueNumber"
                 value="30" />
                </Taraet>
              </Action>
        </ResultHandler>
       </Result>
   </Action>
 </EventHandler>
</EventBinding>
```

예상되는 결과

"MsgBox" 버튼을 눌렀을 때, 메시지 박스가 열린다.

BAR Visualization tra	aining ×				
← → C 01	ocalhost:81/index.html?visuId=	training			
			Logge	e in as: Anonymous	
				Logout	
Main					Service
Service Alarm	Set to default	MsgBox			Alarm
	Name and Address of Add	N work inst			
	Numeric Output	wurnend input			
	35.0	Reset SetTemperature	it		
	35.0°C	Are you sure?			
		OK Cancel			
<u>8</u>					

그림 12 전경에 배치된 메시지 박스

OK 버튼을 눌렀을때, 무엇이 기대되는가? OPC UA 변수 "SetTemperature" 값이 30 으로 설정된다.

*Anonymous" 사용자로 로그인할 경우, OPC UA 변수 접근 권한이 이 경우에 할당되어 있지 않다. 로그인한 사용자가 OPC UA 변수 "SetTemperature" 쓰기 권한을 가지고 있다면, 그 값이 쓰일 것이다.

Automation Runtime logger 에 해당 에러, "BadUserAccessDenied"가 입력된다.

2 😵 Error 2016-12-14 10:35:33,890... -1061091064 B&R mapp View Server OPC UA variable "::AsGlobalPV:SetTemperature" write failed. Status: BadUserAccessDenied 그림 13 접근 권한이 거부 할 때, AR logger

3 역할과 권리 위젯 사용하기

3.1 역할과 권리 위젯 개요

가시성과 유효성을 제어하기 위한 두가지 옵션이 있다(위젯 상태 활성화), 여기에는 프로세스 종속 기능과 역할 기반 기능이 구분된다.

예를 들어서, OPC UA 변수를 통해 위젯 그룹 또는 위젯의 가시성과 유효성을 제어하는게 필요하다면, 위젯의 "visible" 또는 "enable" 속성을 바인딩이 필요하다. 이 교육자료에는 관련 기능을 연습할 예제가 제공된다.

사전에 TM611 에서 보았듯이, OPC UA 변수 접근 권한은 각 역할별로 제한된다. 입력-의존 위젯은 사용 가능 상태에 영향을 미친다.

게다가, 위젯의 가시성 또는 유효성은 로그인한 사용자에 따라서 역할 기능이 제한된다.

operties - SetTemperature			×-
Name	Value	Value	Value Value
🗉 👔 SetTemperature			
	True		
- Audit Events	Off		
Authorization			
Inherit From Parent	Inherit None		
erit Rights / Roles	Everyone	Operator	Service
- Visible			
Browse	\checkmark	\checkmark	\checkmark
- Subscribe	\checkmark	\checkmark	\checkmark
- Read			
Write			

그림 14 "Everyone" 역할시 OPC UA 변수 쓰기 권한 제한

Visualization / Mapp View / Guildlines / FAQ / Widget applications / Restricting visibility and operation

3.2 역할 기반 위젯의 가시성과 유효성

3.2.1 "permissionView" 속성에 역할 구성하기

서비스 페이지(ServicePage)로 페이지 전환을 지원하는 NavigationButton 위젯의 콘텐츠 보기를 허용하는 모든 역할은 "permissionView" 속성에 기입되어야 한다.

다음 문법에 따라 역할을 입력하면, 구성이 실행된다:

['Role'[,'Role*']]

연습: "ContentLeft"에 있는 NavigationButton 위젯의 가시성 접근 권한 구성하기

예제의 목표는 ServicePage 로 변환 할 때, "Everyone" 역할시 NavigationButton 위젯의 가시성을 제한한다.

ServicePage 로 변환은 HMI 어플리케이션에서 "Operator" 와 "Service" 역할의 사용자가 할당되었을 때만 가능하다. 이는 사용자가 로그아웃하면, 네이게이션 버튼은 보이지 않는다.

여기에서 다음 단계가 필요하다:

 NavigationButton 위젯에 "permission View" 속성에 역할 추가하기 역할이 입력되어 있지 않다면, 모든 역할에 가시성이 주어진다. 즉, 제한이 없다.



그림 15 permissionView 속성으로 가시성 제어 구성



예상되는 결과

Ki Visualization training ×	
← → C Olocalhost:81/index.html?visuId=training	
Logged in as: Anonymous Loggout Loggout	
Main Alarm Set to default MsgBox	Service Alarm
CA Visualization training ×	
← → C O localhost:81/index.html?visuId=training	
Logged in as: Dave	
Main	Service
Service Y Alarm MsgBox	Alarm

그림 16 "Service"와 "Operator" 역할에 사용자가 할당되었을때 네비게이션버튼이 보인다.

3.2.2 역할-기반 위젯 구동

서비스 페이지(ServicePage)로 페이지 전환을 지원하는 NavigationButton 위젯의 "permissionOperte" 속성에 시스템 구동을 허용하는 역할을 모두 기입한다.

다음 문법에 따라 역할을 입력하면, 구성이 실행된다.:

['Role'[,'Role*']]

예제: "ContentLeft"에 있는 NavigationButton 위젯의 구동 접근 권한 구성하기

예제의 목표는 ServicePage 로 변환할때 "Operator" 역할이 NavigationButton 위젯 구동 권한을 가지면서 가시성을 갖는다.

지난 예제에서, NavigationButton 위젯의 가시성은 이미 제한되었다. 구동을 위한 역할을 "permission-Operte" 속성에 명시하라.

2↓ 🖾	
Accessibility	
permissionOperate	['Service']
permissionView	['Operator', 'Service']
Appearance	
Appearance.Font	
Behavior	
Common	
Name	NavigationButton2
Туре	widgets.brease.NavigationButton
Data	
pageld	ServicePage
value	false;
ayout	

그림 17 permissionOperate 에 접근 권한 구성

예상되는 결과



그림 18 "Operate" 역할 사용자가 로그인하자 NavigationButton 은 보이지만 사용할 수 없다.

4 HMI 어플리케이션에서 애니메이션

구성 도중 콘텐츠 편집기에서 확인할 수 있는 색상 체계와 모양을 가진 정적 스크린 요소 구성을 통해 매력적인 HMI 어플리케이션 프로그램 요소를 많이 만들 수 있습니다.

게다가, 위젯 또는 위젯 그룹의 가시성을 프로세스 상태 함수로 변경하거나 작업 중 위젯의 시작적 모양 변경이 가능합니다.

추가적인 예제를 통해, HMI 어플리케이션에서 특정 사이즈를 다른 사이즈 디바이스를 위해 조정하고, 어떻게 페이지의 배경 이미지 구성을 할 수 있는지 배울 것이다.

4.1 간단한 위젯의 가시성 제어

다음 예제는 프로세스 상태, 존재하는 OPC UA 변수에 따라서 위젯의 가시성이 어떻게 보이는지 확인한다.



Visualization / mapp View / Guildlines / FAQ / Widget applications / Restricting visibility and operation

예제: 위젯의 가시성 상태 변경

Bool 타입 OPC UA 변수를 사용하여 위젯의 가시성을 제어할 것이다.

이 예제에서는 다음 단계가 필요하다:

- 1) 서브 프로그램 "Program" Init 파트에, "BOOL" 타입인 "State 1" 변수 추가
- 2) OPC UA 변수 활성화 설정
- 3) "ContentMainPage" 콘텐츠에 ToggleButton 위젯을 추가하고 OPC UA 변수 바인딩
- 4) NumericOutput 위젯의 가시성 속성에 OPC UA 변수 바인딩



1~3 단계는 학습자가 직접 구현한다. OPC UA 변수의 모든 역할에 대해 쓰기 권한이 가능하도록 하십시오.

4.1.1 Visible 속성에 바인딩

OPC UA 변수 "State 1"를 바인딩하기 위해서 NumericOutput 위젯의 visual 속성에 설정한다.

초기 출력동안 위젯 가시성의 점멸(블링킹)을 피하기 위해서, 디폴트 값(default)은 "false"로 설정한다.



예상되는 결과

"시뮬레이션"에서 ToggleButton 변환은 "ture" 또는 "false"이다. 이 상태는 NumericOutput 위젯의 가시성에 영향을 준다.



그림 20 ToggleButton 을 누르기와 해제에 따른 위젯의 가시성

4.2 다수 프로세스 값을 사용하여 가시성 제어하기

존재하는 OPC UA 변수를 사용하여 3 가지 프로세스 상태에 따라서 HMI 어플리케이션에 위젯 가시성을 어떻게 제어하는지 배울 것이다. 이를 위해서, 제어 프로그램에서 계산이 필요하지 않다.

HMI 어플리케이션에서 사용되는 계산 결과는 익스프레션(expression)을 이용하여 구현한다. 익스프레션(expression)은 계산된 이후에 결과 값을 도출한다.



Result = InA (BOOL) AND InB (BOOL) AND (InC (INT) > 30)



Visualization / mapp View / Engineering / Variables and data / Expressions

예제: 익스프레션(expression)을 생성하고 그 결과를 가시성(visible) 속성에 연결하기

다음 에제는 익스프레션(expression)을 구성하고 그 결과를 NumericOutput 위젯의 visible 속성에 연결하는 방법을 설명한다.

이를 위해 다음 단계가 필요하다:

- 1) 서브프로그램 "Progrm"에, "BOOL" 타입 변수 "State1", "State2" 와 "REAL" 타입 변수 "State3" 추가
- 2) 세개 변수를 OPC UA 변수로 활성화
- 3) "ContentMainPage" 콘텐츠에 Togglebutton 위젯을 2개 추가하고 각각 OPC UA 변수 "State1", "State2"와 바인딩
- 4) "ContentMainPage" 콘텐츠에 BasicSlider 위젯을 추가하고 OPC UA 변수 "State3" 바인딩
- 5) 로지컬 뷰에 익스프레션(expression)과 익스프레션타입(ExpressionType)을 추가하고 구성하기
- 6) OPC UA 변수를 익스프레션(expression)의 피연산자 함수(operand) 바인딩
- 7) NumericInput 위젯의 visible 속성에 익스프레션(expression)의 결과 바인딩



1~4 단계는 학습자가 직접 구현한다. OPC UA 변수의 모든 역할에 대해 쓰기 권한이 가능하도록 하십시오.

4.2.1 익스프레션 타입(expression type) 구성



익스프레션(expression)은 두 단계로 구성된다. 첫 단계는 익스프레션 인스턴스의 템플렛으로 익스프레션 타입을 생성한다.

Object Catalog 에서 "Expression Type" 파일을 선택해서 비주얼라이제이션 패키지(visualization package) / Expressions 노드에 추가한다.

그림 21 로지컬 뷰에 익스프레션타입과 익스프레션 파일

파일을 연 이후에, ExpressionTypeSet ID 에 고유한 이름을 기입해야하며, 여기서는 "myExpressionTypeSet"이라 명명한다.



```
<?xml version="1.0" encoding="utf-8"?>
<ExpressionTypeSet id="myExpressionTypeSet"
xmlns="<u>http://www.br-automation.com/iat2016/expressionType/v1</u>">
<ExpressionType>
```

</ExpressionTypes> </ExpressionTypeSet>

다음 단계는 Expression Type 구성을 위해 이름은 "myCalculation", 데이터 타입은 "BOOL"로 구성한다.

<ExpressionType name="myCalculation" datatype="BOOL">

</ExpressionType>

Expression Type 안에 Operand 와 Operation 을 구성한다.

```
<Operands>

<Operand name="State1" datatype="BOOL" />

<Operand name="State2" datatype="BOOL" />

<Operand name="State3" datatype="ANY_REAL" />

</Operands>

<Operation>

(State3 > 30 AND State1) AND State2

</Operation>
```

Operation 은 어떻게 operands 가 연계될지 정의한다. 이 계산 결과는 명시된 Expression Type 타입에 따라서 "ture" 또는 "false"가 반환된다. Expression Type 의 최종 구성은 아래와 같다:

```
<?xml version="1.0" encoding="utf-8"?>
<ExpressionTypeSet id="myExpressionTypeSet"
 xmlns="http://www.br-automation.com/iat2016/expressionType/v1">
 <ExpressionTvpes>
  <ExpressionType name="myCalculation" datatype="BOOL">
              <Operands>
                  <Operand name="State1" datatype="BOOL" />
                     <Operand name="State2" datatype="BOOL" />
                     <Operand name="State3" datatype="ANY_REAL" />
              </Operands>
              <Operation>
                     (State3 & gt; 30 AND State1) AND State2
              </Operation>
  </ExpressionType>
 </ExpressionTypes>
</ExpressionTypeSet>
```

4.2.2 익스프레션(expression) 구성

두번째 단계는 익스프레션(expression)의 인스턴스를 구성하는 것이다.

Object Catalog 에서 "Expression" 파일을 선택해서 비주얼라이제이션 패키지(visualization package) / Expressions 노드에 추가한다.

파일을 연 이후에, ExpressionsSet ID 에 고유한 이름을 기입해야하며 여기서는 "myExpressionSet"이라 명명한다.

```
<?xml version="1.0" encoding="utf-8"?>
<ExpressionsSet id="myExpressionSet"
xmlns="<u>http://www.br-automation.com/iat2015/expression/engineering/v3</u>"
xmlns:xsi="<u>http://www.w3.org/2001/XMLSchema-instance</u>">
<Expressions>
```

</Expressions> </ExpressionsSet>

HMI 어플리케이션(.vis) 의 <ExpressionsSet> 요소에 레퍼런스로 ID 를 기입한다.

```
<ExpressionsSets>
<ExpressionsSet refld="myExpressionSet"/>
</ExpressionsSets>
```

완료된 후, <Expressions> 아래에 expression 자신을 추가한다.

```
<Expression xsi:type="content" id="myExprResult"
contentRefId="contentMainPage" type="myCalculation"/>
```

최종 완료된 expression 은 다음과 같다:

<?xml version="1.0" encoding="utf-8"?> <ExpressionsSet id="myExpressionSet" xmlns="<u>http://www.br-automation.com/iat2015/expression/engineering/v3</u>" xmlns:xsi="<u>http://www.w3.org/2001/XMLSchema-instance</u>"> <ExpressionS: <Expressions> <Expressions> </Expressions> </Expressions> </ExpressionsSet>

4.2.3 익스프레션의 피연산자를 OPC UA 변수에 바인딩하기

요구되는 OPC UA 변수 "State1", "State2", "State3"는 해당되는 피연산자에 연결되며, 익스프레션(expression)은 운영 도중 값을 도출할 수 있다. 바인딩은 바인딩 파일(binding file)에서 수동으로 작성한다. OPC UA 변수(source)는 익스프레션의 피연산자와 "oneWay" 바이딩으로 연결된다.

<Binding mode="oneWay"> <Source xsi:type="opcUa" refId="::Program:State1" attribute="value" /> <Target xsi:type="expression" refId="myCalculation" attribute="State1" /> </Binding>

익스프레션 ID, "myExprResult"는 <Target>에 레퍼런스되며, 피연산자 이름은 attribute 에 연결된다. 완료된 피연산자 바인딩은 다음와 같다:

<Binding mode="oneWay"> <Source xsi:type="opcUa" refld="::Program:State1" attribute="value" /> <Target xsi:type="expression" refld="myExprResult" attribute="State1" /> </Binding> <Binding mode="oneWay"> <Source xsi:type="opcUa" refld="::Program:State2" attribute="value" /> <Target xsi:type="expression" refld="myExprResult" attribute="State2" /> </Binding> <Binding mode="oneWay"> <Source xsi:type="expression" refld="::Program:State3" attribute="value" /> <Target xsi:type="opcUa" refld="::Program:State3" attribute="value" /> <Target xsi:type="opcUa" refld="::Program:State3" attribute="value" /> <Target xsi:type="expression" refld="myExprResult" attribute="state3" /> </Binding>

이 마지막 단계에서, 계산된 결과 값은 위젯의 가시성 속성(visible property)와 연결해야한다.

4.2.4 가시성 속성에 익스프레션 결과 바인딩하기

익스프레션의 결과는 Bool 타입으로 반환된다. "myExprResult" ID 의 익스프레션으로 구성된, 결과 값은 위젯의 가시성 속성(visible property)에 연결될 수 있다. 바인딩은 바인딩 파일에 수동으로 기입할 수 있다. Soruce 의 타입에는 "expression", attribute 에 "result"를 입력한다.

```
<Binding mode="oneWay">
<Source xsi:type="expression" refId="myExprResult" attribute="result" />
<Target xsi:type="brease" contentRefId="ContentMainPage"
widgetRefId="NumericOutput2" attribute="visible" />
</Binding>
```

예상되는 결과

세 개의 피연산자 값을 통해 TRUE 또는 FALSE 가 반환된다. 그 결과는 위젯의 가시성 제어에 사용되었다.



그림 22 expression 결과에 다른 NumericOutput 위젯의 가시성

4.3 페이지의 배경 이미지

이전 예제에서, 페이지의 콘텐츠에서 해당 영역의 배경 색을 정의했다.

Appearance	
backColor	gba(255, 133, 0, 1)
backGroundGradient	none
styleRefid	default
zoomMode	original
Behavior	
sizeMode	611
Common	
areald	AreaTop
refld	ContentIop .

예제: 메인페이지(Main Page)에 배경 이미지 출력

예제의 목표는 영역 배경 색상 대신하여 전체 페이지에 이미지를 출력하는 것이다. 다음 장에서 페이지에 이미지 구성이 사용될 것이다.

이를 위한 필요한 단계는 다음과 같다:

- 1) 이미지를 선택하고 페이지 비율(page ratio)을 4:3 설정
- 2) Media 패키지에 이미지 저장
- 3) MainPage 의 .page 파일의 이미지를 레퍼런스 잡기
- 4) 영역(area)의 배경 색 변경

1~2 단계는 학습자가 직접 구현한다. 랜덤 이미지를 사용한다.

4.3.1 MainPage 의 배경 이미지 레퍼런스 설정

미디어 패키징 이미지를 추가하고, 페이지에 레퍼런스 잡는다. 레퍼런스 속성은 back-Ground="Media/Images/Background.png" 이다.



그림 24 페이지 편집기에서 페이지를 위한 배경 명시

프로젝트 전송 이후에, 영역(area)의 배경색 속성으로 이미지가 보이지 않는다. 영역 속성 backColor="transparent"로 속성을 변경해야 운영 중 배경 이미지가 보인다.

배경 이미지 색은 영역(area)의 "backColor" 속성을 단축키 "Reset"를 눌러 초기화 할 수 있다.

			Properties - AreaLeft	
₩ 2↓ 1				
Appearance			Appearance	
backColor	Dent	a(102, 53, 0, 1)	backColor	rgba(255, 255, 255, 0)
backGround	Keset		backGround	
backGroundGradient	none		backGroundGradient	none
styleRefld	defau	ilt	styleRefld	default
zoomMode	origin	alaterian	A monthly and	

그림 25 단축 매뉴 / Reset 를 사용한 배경 색 초기화

재 설정 이후에, 영역의 테마에서 사전에 정의된 디폴트 스타일이 적용된다.

디폴트 스타일은 그래이 배경 색이다: <Style id="default" xsi:type="system.brease.Area" backColor="#e6e6e6" />

Transparent 배경을 사용하기 위해서, 각

Transparent 스타일이 영역에 적용되었다.

<Style id="transparent" xsi:type="system.brease.Area" backColor="transparent" />

Appearance	
backColor	rgba(255, 255, 255, 0)
backGround	
backGroundGradient	none
styleRefld	transparent
zoomMode	original

그림 26 area 에 적용된 "transparent" 스타일

MainPage에서 배경 이미지가 적용되었다.

예상되는 결과

영역의 "styleRefId" 속성에 transparent 가 정의되어야 한다.



그림 27 스타일 변경





4.4 위젯의 색상 변경

이 예제에서, 프로세스 시퀀스의 상태 변화 반응을 위해서 위젯의 비주얼 디자인을 변경한다. HMI 어플리케이션 사용자가(예: 프로세스 시퀀스 중 에러 발생) 디스플레이 위젯의 변화를 알아차리게 만든다.

Set to default	MsgBox
----------------	--------

그림 28 시각적으로 표시된 프로세스 시퀀스의 상태

•

TM611 에서 미리 살펴보았듯이, 비주얼 디자인은 스타일 레퍼런스(style reference) 또는 스타일 속성(styleable properties)에서 실행된다.

MainPage 에서 ToggleButton 으로 값이 변화되는 OPC UA 변수 "::Program:State2"로 프로세스 상태를 대변한다. TRUE 및 FALSE 상태에 대한 시각적 모양을 나타내는 두가지 스타일이 필요하다.

4.4.1 "TRUE" 상태를 위한 스타일 정의

스타일 "Operate1"은 "ButtonMsgBox"라 명명된 메시지 박스의 운영을 위한 버튼 위젯에 해당한다.

새로운 스타일을 만들고 외각선은 빨간색(red border color)으로 프레임 폭을 정의한다.

예제: 에러 상태를 위한 버튼 스타일

예제의 목표는 "Button" 위젯을 위한 "Error" 이름에 스타일을 새로 만드는 것이다. 스타일 "Operate1"이 BuRTheme1 에 추가되어 있으며, 이는 "Button.stypes" 파일에 포함되어 있다. 이 스타일을 복사 및 붙여넣기 한 이후에, 필요한 스타일 속성을 변경한다.

"Button.styles" 파일에 다음을 수행하라:

- 1) 버튼 스타일 "Operate1" 복사
- 2) 새로운 라인 추가, ID 를 "Error"로 변경
- 3) borderWidth 와 mouseBownBorderWidth 를 "2px 2px 2px 2px"로 변경
- 4) borderColor 와 mouseBownBorderColor 를 추가하고 빨간 프레임을 위하여 "#FF0000"로 설정

완료된 "Error" 버튼 스타일:

<style <="" id="Error" th="" xsi:type="widgets.brease.Button"></style>
--

4.4.2 스타일을 이벤트 바인딩으로 변경하기

이번 단계에서는 OPC UA 변수 "::Program:State2" 상태를 이용하여 스타일 "Error" 또는 "Operate1"로 표현한다.

<Source xsi:type="opcUa.Event" refld="::Program:State2" event="ValueChanged" />

변수가 "FALSE" 상태일때는 스타일 "Operate1"으로 보이며, "TRUE" 상태일때는 스타일 "Error"로 표시된다. "TRUE" 상태는 <EventHandler>에 추가 상태 condition="newValue"를 추가하여 평가하고 "FALSE" 상태는 <EventHandler>에 추가 상태 condition="NOT newValue"를 추가하여 구별한다.

위젯 액션 "SetStyle"로 스타일을 변경한다.

```
<EventHandler condition="newValue">

<Action>

<Target xsi:type="widgets.brease.Button.Action"

contentRefId="ContentMainPage" widgetRefId="ButtonMsgBox">

<Method xsi:type="widgets.brease.Button.Action.SetStyle"

value="Error" />

</Target>

</Action>

</EventHandler>
```

위젯 스타일 변경을 위한 완료된 이벤트 바인딩은 다음과 같다:

```
<EventBinding>
 <Source xsi:type="opcUa.Event" refId="::Program:State2"</pre>
         event="ValueChanged" />
   <EventHandler condition="newValue">
       <Action>
         <Target xsi:type="widgets.brease.Button.Action"
           contentRefId="ContentMainPage" widgetRefId="ButtonMsgBox">
              <<u>Method xsi:type="widgets.brease.Button.Action.SetStyle"</u>
              value="Error" />
         </Target>
       </Action>
   </EventHandler>
   <EventHandler condition="NOT newValue">
       <Action>
         <Target xsi:type="widgets.brease.Button.Action"
           contentRefId="ContentMainPage" widgetRefId="ButtonMsgBox">
               <Method xsi:type="widgets.brease.Button.Action.SetStyle"</pre>
            value="Operate1" />
        </Target>
       </Action>
   </EventHandler>
</EventBinding>
```

예상되는 결과

Ki Visualization training ×	
← → C ① localhost:81/index.ht	ml?visuld=training
Main	
Alarm	to default MegBox
Numeric o	utput Numeric input
	35.0
	35.0 °C
	/
5	itate1 State2

그림 29 스타일 변경

이벤트 바인딩에 OPC UA 변수 사용시, 클라이언트가 재시작할때 ValueCahnged event 는 한번만 실행되며 더 이상 존재하지 않음에 유의하라. 이 경우 위젯 스타일이 변경된다.

이 솔루션으로, OPC UA 변수 (source)는 섹션 변수(session variable) (target)에 연결할 수 있다. 이벤트 바인딩에서 섹션 변수의 값 변경 이벤트에는 항상 반응이 있다. 왜냐하면 바인딩에 의해 계속 업데이트되기 때문이다.

<u>"Seesion variables</u>" 참조

4.5 컨테이너 위젯에 상대적이고 절대적인 위치

디폴트로, 위젯은 절대 상단/ 왼쪽 위치로 콘텐츠가 표시된다. TM 611 에서 학습했듯이 ContentLeft 에 NavigationButton 위젯은 NaviagationBar 위젯에서 상대 위치로 배치되었다. Container 위젯이 "child-Positioning" 속성을 제공한다면, 여기에 배치되는 Child 위젯 배치방법을 상대적(relative)이거나 절대적인(absolute) 방법을 선택할 수 있다. 예제의 목표는 GroupBox 에 행 단위로 임의로 위젯을 배치한다. 이를 위해서, transparent GroupBox 를 라인에 추가하고 가시성(visible) 속성을 바인딩 설정한다.

예제: GroupBox 의 라인 숨기기

비주얼 속성 바인딩을 이용하여 라인을 보이고 숨기는 기능을 GroupBox 에서 구성하는 방법을 배울 것이다. 이 예제를 위하여 특정 기계 구성이 필요하다.

이를 위해 다음 단계가 필요하다:

- 1) ContentServicePage 콘텐츠에 relative 위치로 GroupBox 추가
- 2) 첫번째 GroupBox(=line)에 투명한 하위(Child) GroupBox 추가
- 3) 각 그룹박스(GroupBox)에 위젯 추가
- 4) 첫 하위(Child) 그룹박스(GroupBox)의 시각화 속성(visible)에 OPC UA 변수 "::Program:State2" 바인딩

4.5.1 그룹박스(GroupBox)추가 및 구성

" ContentServicePage"에 그룹박스(GroupBox)를 추가하고 "childPositioning"속성을 "relative"으로 설정한다.

Property name	Value
childPositioning	relative
Position	25;480
Size	580;270

표 6 GroupBox 의 속성



그림 30 relative positioning 으로 설정된 부모 GroupBox

GroupBox 에 GroupBox 를 3 개 추가하고 사이즈는 parent 의 폭으로 조절한다. default text 를 제거하여 그룹박스(GroupBox)의 헤더를 숨기고 referencing style="transparent"로 설정하여 경계선과 배경색을 생략한다.

3개 그룹박스 위젯은 다음 속성처럼 설정한다:

Property name	Value
Style	Transparent
Text	
Position	0;0
Size	550;55
Margin	10рх

표 7 GroupBox 위젯의 속성

부모 그룹박스에 자식 그룹박스를 추가하면 자동으로 top/left="0"으로 설정된다. 라벨(Label), 버튼(button), 수치입력(NumericInput) 위젯을 각 자식 그룹박스에 absolute 포지션으로 추가한다.

Usemame	.	GroupBox1	
Password	Line1	Button 1	0.0
Login		-	
	Line2	Button2	0.0
Deutsch	Line3	Button3	0.0
Deutsch	D		

그림 31 완전히 궝된 그룹박스 - 멀티 선택 출력

나중에, 첫번째 자식 그룹박스의 가시성 속성(visible)을 OPC UA 변수 "::Program:State2"와 바인딩한다.

예상되는 결과

MainPage 에 OPC UA 변수 "::Program:State2"와 연계된 ToggleButton 을 값을 변경한다면 첫번째 자식 그룹박스가 보이거나/숨겨진다. 나머지 그룹박스 위젯은 문서에 빈 공간을 채우며 상대적으로 배치된다.

⇒ G [① localhost81/inde	x.html?visuId=training				
				Logged in as	Anonymous Logout	
Main		~				Main
Alarm	Usemame			GroupBox1		Alam
	Password		Line2	Button2	0.0	
		Logim				
			Line3	Button3	0.0	
	English					
	Matric					

그림 32 첫번째 자식 그룹박스를 숨긴 결과

4.6 브라우저 윈도우 사이즈에 따른 HMI 어플리케이션 스케일링

HMI 어플리케이션은 디스플레이 해상도를 생성한다. 페이지의 사이즈는 페이지가 레퍼런스 된 레이아웃을 통해 결정된다. 같은 HMI 어플리케이션을 다른 사이즈 디스플레이(예: HD Ready 와 Full HD)에 출력하고자 한다면, 자동 줌(automatic zoom) 구성으로 설정할 수 있다.

서로 다른 대각선 사이즈의 디스플레이 단말 장치에 HMI 어플리케이션 구동을 보장하려고 한다. 게다가, 디스플레이 단말 장치의 영상비¹는 반드시 동일해야 한다. (예: 4:3 또는 16:9)

예제: 브라우저 윈도우 사이즈에 따른 HMI 어플리케이션 자동 조절

<Cnfigurations> 요소에서 HMI 어플리케이션의 줌 설정을 구성할 수 있다. 자동 줌은 key="zoom", value="true"로 입력하여 활성화시킨다.

<Configuration key="zoom" value="true" />

줌 설정은 HMI 어플리케이션이 제공되었을 때 브라우저 윈도우 사이즈와 관련하여 조정된다. 사이즈를 다이나믹하게 변경하면 페이지를 다시 그리며; 몇몇 위젯은 사이즈 변화에 반응하지 않는다. 그러나 초기 크기 또는 입력 필드를 유지한다.

예상되는 결과

브라우저 윈도우나 HMI 어플리케이션을 다시 시작하면 크기가 변경되어 HMI 어플리케이션이 그에 따라 확장된다. 콘텐츠 조각 크기가 조정되고 최대 사이즈로 보인다.

¹ (역자) aspect ratio, 화상의 가로 세로 비

5 다른 변수 타입을 포함한 테스크

이전 예제에서, 우리는 HMI 어플리케이션에서 변수 사용을 위해 다음 타입을 이용하였다.

Variable type	Scope
OPC UA variables	Controller program
Expressions	Visualization

표 8 사전에 사용된 변수 타입 개요

콘텐츠 조각을 위해서 위젯 인스턴스에 값 바인딩(value binding), 노드 바인딩(node binding)으로 변수가 연결되어 있다. 이 장에서는 해당 연습을 위해 새로운 테스크에 변수와 바인딩 타입을 추가할 것이다.

Variable type	Scope
Session variables	Visualization

표 9 추가적인 변수 타입



Visualization / mappView / Engineering / Variables and data

Session variables

Session variables 은 Session 범위에서 사용되는 변수이다. 다른 말로, 이 값은 특정 클라이언트-서버 연결에서 적용된다. 각 클라이언트를 위해 HMI 어플리케이션의 상태를 저장하는 것이 가능하다. Session 은 value binding 을 이용하여 값을 쓰거나 읽을 수 있다.

리스트 바인딩(List binding)

리스트 바인딩을 위해서, <Source>와 <Target> 사이의 바이딩을 실행해야하는 선택기를 사용하여 변수 목록에서 요소를 선택한다.



5.1 다른 위젯과 정보를 교환

위젯 속성은 Brease-Brease 바인딩을 사용하여 콘텐츠와 연결될 수 있다.

이를 수행 할 때, <Source>와 <Target>의 변수 유형은 xsi:type="brease"로 정의된다. Brease-Brease 바인딩 범위는 같은 콘텐츠로 제한된다.



그림 34 같은 콘텐츠 사이에 정보 교환

네비게이션을 위한 상태, 값 변화는 저장되지 않기 때문에, Content A 의 상태 변화는 Content B 가 같은 시간에 페이지를 출력하고 있을때만 효과가 있다.

Content B 를 포함하지 않는 페이지가 탐색한후, Content A 를 위한 상태 변화는 바인딩에 의해 가능하지 않으며(source 와 target 이 동시에 활성화되지 않는다) 그러므로 잃어버릴 것이다. Seesion variable 은 위젯의 상태와 값 변화를 캐싱(casing)하기 위해 사용된다. 해당 콘텐츠(Content)가 보이지 않을 경우 리턴 값이 반환된다.



그림 35 콘텐츠 보더를 통해 위젯 정보 교환

5.1.1 동일한 콘텐츠에서 위젯 정보 교환

예제의 목표는 동일한 콘텐츠에 있는 위젯에 OPC UA 변수를 사용하지 않고도 콘텐츠에 다른 위젯 값을 표시하는 것이다.

예제: RadialGage 위젯에 BasicSlider 위젯의 값 출력하기

다음 장은 RadialGage 에 BasicSlider 의 포인터를 감시하여 출력시키는 것이다. 이 프로세스에서, "value" 속성 연결은 직접 위젯에 연결한다.

이를 위해 다음 스텝이 필요하다:

- 1) 새로운 "TestPage"를 만들고 자동 네비게이션 확장
- 2) BasicSlider 위젯과 RadialGage 위젯을 ContentTestPage 에 추가
- 3) RadialGage 위젯의 값(value) 속성을 BasicSlider 위젯의 값(value) 속성과 연결



1~2 단계는 학습자가 직접 구현한다.

Copyright © B&R - Subject to change without notice TM671 - 강력한 mapp View 화면.docx 새 페이지 생성 방법:

- 1) 로지컬 뷰에 존재하는 페이지를 복사한다.
- 아웃풋 윈도우에 보이는 에러메시지는 Page ID 또는 content ID 가 이미 사용되고 있기 때문에 나타난다.
- 2) Page 패키지 이름을 "TestPage"로 설정한다.
- 3) 새로 추가된 페이지의 .content 파일을 마우스 우클릭 후 "Open as test"를 클릭하여 연다.
- 4) ID 를 "ContentTestPage"로 변경한다.
- 5) .page 파일을 열어서 page ID = "TestPage"로 변경한다.
- 6) 새롭게 생성한 text ID (예: \$IAT/TestPage)를 "displayName"속성에 기입한다.
- 7) AreaMain 의 <Assignment> 항목에 baseContentRefld="ContentTestPage"로 변경한다.
- 8) 새로운 페이지 ID 를 HMI 어플리케이션(.vis) 내 <Pages> 요소에 기입한다.
- 9) 텍스트 ID 및 현지화 가능 텍스트로 기존 텍스트 파일(.tmx) 수정
- 10) 자동 네비게이션(.nav)에 새로운 페이지를 추가한다.

예를 들어서 MainPage 에서도 새로운 페이지를 접근할 수 있도록 설정한다.

이러한 과정이 완료되면, 당신은 새로운 페이지를 접근할 수 있다. 이제 BasicSlider 위젯과 Radial-Gauge 위젯을 그래픽 콘텐츠 편집기에 추가하자.



5.1.2 Brease-Brease 바인딩

RadialGuage 위젯의 "value" 속성 바인딩은 BasciSlider 위젯의 값 변화(source)에 따른 RadialGauge 위젯의 값 변화(target)를 나타낼 것이다.



그림 36 RadialGuage 위젯의 값 속성에 바인딩 설정

콘텐츠에 있는 모든 위젯은 변수 선택 다이얼로그 박스의 "Widget" 탭에서 바인딩할 수 있는 속성을 볼 수 있다. 위젯의 값 속성을 선택하여 바인딩하므로써, 값이 보인다. 읽기 접근만 발생하기 때문에, 바인딩 모드는 "read only"를 선택한다.

 Select Variable 	
Widget Information	OPC-UA Widget Variables Expression mapp
Property Name: value	Bindable Content Widget Properties
Data Type: Number	EasicSlider1
	i international and a state of the state of
Binding Mode	
Read / Write	
Read Only	
Init Read / Write	
Bindings Set Id	
MyBinding -	۲
BasicSlider1.value	OK Cancel

그림 37 같은 콘텐츠에 위치한 위젯 값 바인딩

예상되는 결과

위젯 사이에 바인딩은 바인딩 파일에 다음과 같이 입력된다:

```
<Binding mode="oneWay">
<Source xsi:type="brease" contentRefId="ContentTestPage"
widgetRefId="BasicSlider1" attribute="value" />
<Target xsi:type="brease" contentRefId="ContentTestPage"
widgetRefId="RadialGauge1" attribute="value" />
</Binding>
```

TestPage 에서 슬라이더 포인터(slider) 조작은 RadialGauge 포인터를 이동시킨다.



그림 38 슬라이더 값 변화에 다른 RadialGauge 화살표 포인터의 이동

5.1.3 콘텐츠 보더를 통한 위젯 정보 교환

이 예제의 목표는 다른 콘텐츠 위젯에 OPC UA 변수를 사용하지 않고 콘텐츠 위젯 값을 보여주는 것이다.

예제: NumericOutput 위젯에 BasicSlider 위젯 값 출력하기

다음 장에서, "ContnetTestPage" 콘텐츠에 BasicSlider 위젯 값을 "ContentTop" 콘텐츠에 NumericOutput 위젯에 출력하는 것이다. "value" 속성은 session variable 을 이용하여 연결된다.

이를 위해 필요한 과정은 다음과 같다:

- 1) 비주얼라이제이션 패키지(Visualization package)에 session variable 파일 추가
- 2) "ANY_REAL"데이터 타입에 "SliderValue" session variable 정의
- 3) ContentTestPage / BasicSlider 위젯 값 속성에 session variable 바인딩
- 4) ContentTop / NumericOutput 위젯 값 속성에 session variable 바인딩

5.1.4 Session variable

Session variable 은 로지컬 뷰에서 비주얼라이제이션 패키지 내 '.svar' 파일에서 관리된다.

```
? Visualization / mapp View / Engineering / Varilabe and Data / Session Variable
```



그림 39 로지컬 뷰에 seesion variable 파일

파일을 열고, 유일한 VariableSet ID 을 위해 "myVariables"라 명명한다. 이 ID 는 visualization object(.vis) 내 <VariableSets> 요소에 레퍼런스 시켜야 한다.

```
<VariablesSets>
<VariablesSet refld="MyVariables" />
</VariablesSets>
```

"ANY_REAL"데이터 타입에 "SliderValue" 변수는 <Variables>에 기입한다. session variable 의 초기 값은 "0"이다.

```
<Variable name="SliderValue" xsi:type="ANY_REAL" value="0" />
```

완료된 session variable 파일은 다음과 같다:

```
<?xml version="1.0" encoding="utf-8"?>
<VariablesSet id="MyVariables"
xmlns=http://www.br-automation.com/iat2015/session/engineering/v1
xmlns:xsi="<u>http://www.w3.org/2001/XMLSchema-instance</u>">
<Variables:
<Variables>
<Variables>
</Variables>
</Variables>
</VariablesSet>
```

다음 단계는 ContentTestPage 에 BasicSlider 를 추가하는 것이다. "value" 속성에서 session variable 바인딩을 수행한다. session variable 은 "Variables" 텝을 누르면 선택할 수 있다.

Widget Information	OPC-UA Widget Variables Expression mapp	
Property Name:	Name	Data Type
value	B Session Variables	
Data Type: Number	Biographic Struct Variables (:SYSTEM)	AINT
Binding Mode		
Read Only		
Init Read / Write		
Bindings Set Id		
MyBinding	•	

그림 40 session variable 선택하기

ContentTop 에 있는 NumericOutput 위젯의 value 속성에 같은 session variable 을 연결한다. 바인딩 파일에는 다음과 같이 저장된다.

```
<Binding mode="twoWay">
<Source xsi:type="session" refld="SliderValue" attribute="value" />
<Target xsi:type="brease" contentRefld="ContentTestPage"
widgetRefld="BasicSlider2" attribute="value" />
</Binding>
<Binding mode="oneWay">
<Source xsi:type="session" refld="SliderValue" attribute="value" />
<Target xsi:type="brease" contentRefld="ContentTop"
widgetRefld="NumericOutput1" attribute="value" />
</Binding>
```

예상되는 결과

TestPage 에 슬라이더를 이동시키면 헤더 값이 출력된다. HMI 어플리케이션에서 다른 페이지로 이동하더라도 session variable 값은 남아 있다.



그림 41 session variable 을 통하여 다른 콘텐츠간에 바인딩 연결

5.2 사용자 변경 및 페이지 전환

이 예제는, 생산 교대 끝에서, 활성 사용자가 HMI 어플리케이션에서 로그아웃 아이콘이나 로그아웃 버튼을 클릭하고 로그아웃한다. 로그아웃 동안, HMI 어플리케이션은 다시 로그인하기 위해서 MainPage 로 전환된다.

예제: 이미지를 클릭하여 로그아웃하고 MainPage 로 페이지 전환하기

로그아웃 구성하는 방법과 특정 페이지로 이동하는 방법을 안내한다.

이를 위해 다음 과정이 필요하다:

- 1) NavigiationRight 콘텐츠에서 B&R 심볼을 클릭하여 로그아웃 수행
- 2) 로그아웃 이후에, 자동으로 MainPage 로 전환



이 예제는 mapp View 도움말 문서를 이용해서 학습자가 직접 구현해본다.



Visualization / mapp View / Guidelines / FAQ / Event and action applications / Chaning page on logout

예상되는 결과



그림 42 MainPage 이동과 로그아웃

5.3 시스템 변수(system variables) 사용하기

시스템 변수(system variables)는 런타임시 mapp View 서버에 의해 값이 채워지는 미리 정의된 영구적인 변수이다.



Visualization / mapp View / Engineering / Variables and data / Session variables / System variables

시스템 변수는 제어 프로그램에서 연결되어 있는 모든 클라이언트에 클라이언트 정보를 평가하기 위해 사용된다. 이 예제의 목표는 mapp View 도움말 문서를 이용하여 클라이언트 정보를 평가하는 것이다. 시스템 변수를 OPC UA 구조체 배열에 바인딩하므로써 모든 클라이언트를 위한 적절한 사용 권한 정보는 프로그램에서 이용가능하다.

예제: OPC UA 구조체 배열에 시스템 변수 연결하기

모든 클라이언트 정보를 어떻게 수집하고 OPC UA 구조체 배열에 연결하는지 설명한다. 이를 위해 다음 과정이 필요하다:

- 1) 전송되어야 할 변수를 위한 구조체 타입 생성
- 2) 위 구조 타입으로 길이가 [0..2]인 전역 변수 "ClientInfo" 생성
- 3) "ClientInfo"변수를 프로그램에서 사용
- 4) 새 바인딩 파일(binding file) 추가
- 5) Bining ID 를 정의하고 .vis 파일에 레퍼런스 잡기
- 6) 도움말에 리스트 바인딩(list binding) 복사



이 예제는 mapp view 도움말 문서을 참조하여 학습자가 직접 구현한다. 구현방법이 자세하게 적혀있다. 이 예제는 "a list binding of client information via OPC UA "장 까지만 수행하면 된다.

리스트 바인딩이 예제에서 사용되었다. 다음 XML 코드는 시스템 변수를 위해 발췌하였다:

```
<Binding mode="oneWayToSource">
<Source xsi:type="listElement">
<Selector xsi:type="session" refld="::SYSTEM:clientInfo.slotId"
attribute="value" />
<be:List xsi:type="be:opcUa" attribute="value">
<be:List xsi:type="be:opcUa" attribute="value">
<bt:Element index="0" refld="::AsGlobalPV:ClientInfo[0].userId" />
<bt:Element index="1" refld="::AsGlobalPV:ClientInfo[1].userId" />
<bt:Element index="2" refld="::AsGlobalPV:ClientInfo[2].userId" />
</be:List>
</source>
<Target xsi:type="session" refld="::SYSTEM:clientInfo.userId"
attribute="value" />
</Binding>
```

다음 이미지는 <Target>과 <Source> 요소, OPC UA 변수 리스트 사이에 "oneWayToSource" 효과를 보여준다. Mapp View 서버를 통해서 클라이언트가 처음 로그인하면 서버에 의해 slotId="0"이다. 그러므로 이 포인터는="::AsGlobalPV:ClientInfo[0].userId"에 해당한다. 그 결과, 사용자 이름은 구조체 배열의 첫번째 요소 [0]에 해당되는 변수의 <Target>에 정의된 시스템 변수에 의해 기록된다.



그림 43 User ID 를 위한 리스트 바인딩 스키마

예상되는 결과

와치 윈도우(watch window)를 사용하여 제어 프로그램에서 클라이언트 정보를 관찰할 수 있다.

Name	Туре	Scope	Force	Value
ClientInfo	ClientinfoType[0.2]	globa		
	ClientInfoType			
- 🖉 userid	WSTRING[80]			"Dave"
- 🔷 isValid	BOOL			TRUE
- / ipAddress	WSTRING[15]			*127.0.0.1*
- languageld	WSTRING[2]			"en"
- browserResolution	WSTRING[32]			"1344x840"
L ↓ slotid	SINT			0
¢	ClientInfoType			
- 🔷 userld	WSTRING[80]			***
–	BOOL			FALSE
- / ipAddress	WSTRING[15]			
- 🔷 languageld	WSTRING[2]			
- browserResolution	WSTRING[32]			
L 🤣 slotld	SINT			0
E	ClientInfoType			
- 🧇 userld	WSTRING[80]			
🗕 🔷 isValid	BOOL			FALSE
- ipAddress	WSTRING[15]			***
- 🔷 languageld	WSTRING[2]			
- browserResolution	WSTRING[32]			***
L 🧇 slotid	SINT			0

그림 44 Watch window 에서 연결된 클라이언트 정보

6 텍스트 시스템(text system) 연습

교육자료 TM611, TM671 예제에서, 텍스트 시스템(text system)은 선택된 언어에 따라서 위젯에 지역화 언어를 나타내기 위해 여러번 사용되었다.

기계 작동 중에 스니펫(Snippets)은 형식화 된 값 중에서 텍스트의 내용을 동적으로 표현하는데 사용된다.



6.1 색인 테스트 출력하기

예제의 목표는 session variable 값이 변화될 때 텍스트 리스트로부터 특정 텍스트를 출력하는 것이다. 텍스트는 텍스트 파일(.tmx)에 생성한다.



그림 45 색인 텍스트 출력을 위한 스키마

예제: 색인 텍스트 출력하기

Session variable "SelectText" 값이 변화된 결과에 따라서 TextOutput 위제에 보여지는 텍스트 리스트를 특정 텍스트로 출력되도록 하는 것이다.

이를 위해 다음 과정이 필요하다:

- 1) "ANY_INT" 타입에 session variable "SelectText"를 만든다.
- 2) ContentTextPage 에 3개 RaidoButton 을 포함하는 RadioButtonGroup 을 구성한다.
- 3) RadioButtonGroup 의 "selectedIndex" 속성에 session variable 을 바인딩한다.
- 4) 로지컬 뷰에 새 텍스트 파일(.tmx)을 추가한다.
- 5) 텍스트 시스템 구성(text system configuration)에서 새 텍스트 파일을 레퍼런스 설정한다.
- 6) 3개 Text ID에 해당되는 각 언어를 기입한다.
- 7) Snippet 파일을 추가하고 Index Text snippet 을 생성한다.
- 8) HMI application (.vis)에 snippet ID 를 레퍼런스 설정한다.
- 9) Snippet 에 session variable "SelectText"를 바인딩한다.
- 10) ContentTestPage 에 TextOutput 위젯에 snippet 을 바인딩한다.



1~5 단계는 학습자가 직접 구현한다.

6.1.1 텍스트 파일에 텍스트 리스트 구성하기

새 텍스트 파일에 3개 text ID 를 추가한다.

Text ID 는 텍스트로 입력되고 값은 마침표로 구분된다. 마침표 앞에 있는 텍스트는 그룹 색인 된 텍스트에 대해 동일해야 한다. 마침표 뒤에 셀렉터가 나타내는 값이 추가된다.

9	
Namespace IAT	
TextID	English (en)
1 IndexText0	Fill level is too low
2 IndexText1	Fill level ok
3 IndexText2	Fill level is too high
4 New_Text_ID_1	

그림 46 색인 텍스트를 위해 Text ID 와 연결된 숫자



그림 47 SelectText 값이 2 일 경우에 출력되는 텍스트

6.1.2 Snippet 생성하기



Snippet 파일은 Object Catalog 에서 visualization packge 또는 mapp View 의 Resources/Snippets 노드에 추가한다. 파일을 열고, 고유한 SnippetSet ID 를 위해 "mySnippets"라 기입한다. 이 ID 는 visualization object(.vis)의<SnippetsSets> 요소에 레퍼런스 설정을 해야한다.

<SnippetsSets> <SnippetsSet refld="MySnippets"/> </SnippetsSets>

type="IndexText" 이고 Snippet id="TextSnippet" 인 snippet 을 추가한다.

<Snippet id="TextSnippet" xsi:type="session" type="IndexText" formatItem="IndexText.{1}" />

"formatItem" 속성에서, 사전에 생성된 text ID 가 전송되며, 그에 따라 자리 표시자{1}이 사용된다. 운영 중에 snippet 을 해결할 때, 현재 값이 ID 에 적용되며 해당 text system 을 불러온다.

Snippet 파일의 최종 결과물은 다음과 같다:

```
<?xml version="1.0" encoding="utf-8"?>

<SnippetsSet id="MySnippets"

xmlns=http://www.br-automation.com/iat2015/snippet/engineering/v3

xmlns:xsi="<u>http://www.w3.org/2001/XMLSchema-instance</u>">

<Snippet/engineering/v3

xmlns:xsi="<u>http://www.w3.org/2001/XMLSchema-instance</u>">

<Snippets>

<Snippets>

<Snippet id="TextSnippet" xsi:type="session" type="IndexText"

formatItem="IndexText.{1}" />

</Snippets>

</SnippetsSet>
```

6.1.3 Snippet 을 위해 session variable 바인딩 - snippet 을 TextOutput 위젯에 설정

session variable "SelectText"는 snippet 에 연결이 되고 snippet 는 운영 중 값을 받아드린다. 이는 바인딩 파일(binding file)에서 작성한다.

최종 완료된 바인딩 파일은 다음과 같다:

session variable 에 RadioButtonGroup 바인딩

```
<Binding mode="twoWay">
<Source xsi:type="session" refld="SelectText" attribute="value" />
<Target xsi:type="brease" contentRefld="ContentTestPage" widgetRefld="RadioButtonGroup1"
attribute="selectedIndex" />
</Binding>
```

Inedx Text 에 session variable 바인딩

```
<Binding mode="oneWay">
<Source xsi:type="session" refld="SelectText" attribute="value" />
<Target xsi:type="snippet" refld="TextSnippet" attribute="value" />
</Binding>
```

```
TextOutput 위젯에 해결 된 snippet 텍스트 바인딩
```

```
<Binding mode="oneWay">
<Source xsi:type="snippet" refld="TextSnippet" attribute="value" />
<Target xsi:type="brease" contentRefId="ContentTestPage" widgetRefId="TextOutput3"
attribute="value" />
</Binding>
```

예상되는 결과



그림 49 텍스트 시스템에서 값 변화에 따른 글자 출력

추가적인 예제: 메시지 박스를 위한 지역화 텍스트를 포함하는 출력 값

Visualization / Mapp View / Guidelines / FAQ / Event and action applications / Message-Box : Displaying a numeric value in the message

7 XML 로 콘텐츠 엔지니어링(Content engineering in XML)

콘텐츠는 비주얼 편집기에서 미리 구성되었다. XML 을 편집하고자 하는 모든 사람은, 다음 연습에서 XML 로 내용을 구성할 때 고려해야 할 상항을 보여준다.



XML 편집기와 비주얼 편집기를 동시에 콘텐츠를 열 수 없다. 또한, 고려해야할 사항은 비주얼 편집기에서 내용을 편집한 후에 XML(들여 쓰기 등)에서 수행되는 형식이 손실된다는 것이다.

XML 편집기는 Content file 을 선택하고 마우스 우클릭 후 "Open / Open as text"를 이용하여 열 수 있다. 다음 예제는, "ContentAlarmPage" 콘텐츠를 텍스트 편집기로 연다. 가독성을 위해, XML 스키마의 모든 항목이 표시되지는 않았다:

<?xml version="1.0" encoding="utf-8"?> <Content id="ContentAlarmPage" height="700" width="1080" > <Widgets> </Widgets>

</Content>

속성 이름(Property name)	설명
Content ID	로지컬 뷰에 모든 콘텐츠 파일을 구별하는 고유 ID
Height	콘텐츠의 높이, pixel 기준
Width	콘텐츠의 폭, pixel 기준

표 10 콘텐츠 파일의 기본 속성 개요

콘텐츠가 열리면, <Widgets> 요소에 추가 할 수 있는 모든 위젯은 위젯 카탈로그(Widget Catalog)에 표시된다.

7.1 위젯 추가하기

위젯은 <Widgets> 요소에 드레그-앤드-드롭을 통해 추가한다. 위젯 카탈로그에서 위젯을 드롭 할때, 최소 요구되는 위젯 속성은 다음과 같다:

<Widget xsi:type="widgets.brease.Label" id="Label1" height="30" width="80" left="" top="" text="Label1" zIndex="0" />

Label 위젯 예제를 보듯이, 위젯은 고유 ID 를 가진다. 위제의 사이즈는 "width(폭)"과 "height(높이)"로 결정된다. 위치는 사용자에 의해 "top"과 "left"를 기입하면 된다.

게다가, <Widgets> 요소에 모든 위젯이 포함되며 이는 "0"부터 시작되는 고유 zIndex 를 가져야 한다.



모든 위젯을 위하여, 각 위젯에 허용 된 추가 속성은 추가 할 수 있거나 변경 할 수 있다. 저장할때, 파일 사용이 유효하다. (대문자 사용, 등) 사용가능한 속성은 위젯 문서에서 확인할 수 있다.



Visualization / mappView / Widgets

7.2 Container 위젯 추가하기

Container 위젯 (예: GroupBox)는 자식 위젯 추가를 허락한다. Container 위젯은 <Widgets> 요소를 사용하여 라벨링한다. 간단한 위젯에 대한 모든 규칙은 위젯 자체에도 유효하지만 Container 위젯내부에 위젯의 zIndex 는 다시 "0"으로 시작할 수 있습니다.

```
<Widget xsi:type="widgets.brease.GroupBox" id="GroupBox1"
height="150" width="600" top="" left="" zIndex="0">
<Widgets>
```

<Widgets /> </Widget>

추가적인 위젯은 <Widgets> 요소 아래에 추가 할 수 있다. 연습으로, XML 편집기에 위젯을 추가하고 구성할 수 있다. 필요한 모든 속성을 지정한 후 (예: top/left), 컨텐트(content)를 저장하고 닫을 수 있다.

예상되는 결과



8 멀티-클라이언트, 멀티-사용자

다른 사용자는 개인이 원하는 대로 할 수 있는 콘텐츠(content)를 필요로 한다. mapp View 는 기계 어플리케이션에서 프로그램을 하지 않고 역할 기반(role-based)을 구현할 수 있다.

고객 맞춤형 페이지는 동시에 다른 디바이스에 볼 수 있다. 앞서 보여준대로 Session variables 를 위한 예제와 시스템 변수처럼, 같은 HMI 어플리케이션은 하나 이상의 클라이언트에서 실행된다.



모바일 장치에서 기계의 HMI 어플리케이션 실행을 추천하지 않는다. 예를 들어서 테블릿 같은 장치에서 조작은 일반적으로 특정 역할에 해당한다. (예: 서비스 기술자)

Mapp View 는 어플리케이션의 특정 페이지를 사용하여 독립형 HMI 어플리케이션(.vis) 옵션을 제공하므로 페이지의 기존 내용을 다시 사용할 수 있다.

예제는 하나의 HMI 어플리케이션에 여러명의 클라이언트가 어떻게 접속하는지 보여준다. Mapp View 서버에 동시 접속 할 수 있는 클라이언트 수는 mapp view configuration 에서 정의한다. HMI 어플리케이션이 클라이언트로 이월되면 관련이 없다.

Config.mappviewcfg [mapp View Configuration] ×						
Name	Value	Unit	Description			
🗉 🚛 MappViewConfiguration						
🖶 🚰 Server configuration						
🛱 🚰 Protocol	HTTP		use secure communication protocol			
Port Number	81					
Maximal client connections	3	and the second second				
Maximal B&Prelient connections			يري منه مين المناف المنين المنه والمراجع المناف المنتجي المناف			

그림 50 클라이언트 3 명 접근 가능

승인된 최대 클라이언트 접속 수는 클라이언트 정보와 OPC UA 구조 배열 길이는 반드시 매칭되어야 한다.

이전 예제에서, 브라우저 창에 다음과 같이 나타난다.



그림 51 브라우저에 표시된 HMI 어플리케이션

두번째 클라이언트 접근을 테스트하기 위해서 같은 URL을 2 번째로 사용하여 브라우저에서 열고자 한다면, mapp View 서버에 의해 거절될 것이며 session 은 브라우저 타입으로 "occupied"로 되어있다. 두번째 창(tab)을 브라우저에서 열거나 브라우저 인스턴스 시작과는 관련이 없다.



그림 52 같은 session 으로 브라우저 연결 시도결과인 에러 메시지

예제: 두 클라이언트 인스트스를 HMI 어플리케이션에 출력하기

두 틀라이언트 HMI 어플리케이션 테스트를 위하여, 두번째 인스터스를 열기 위한 옵션은 "incognito mode"이다. 이 경우 새로운 접근(session)은 mapp View 서버에서 생성된다. 대안으로, 다른 제조사가 만든 다른 브라우저를 같은 PC 또는 접근에 따른 다른 디바이스에서 사용될 수 있다. (예: Chrome 과 Explorer)

예상되는 결과

두 클라이언트 인스턴스는 다른 브라우저 창에서 출력되고 개별적으로 운영된다. 다른 사용자는 각각 클라이언트로 로그인할 수 있다. 심지어 언어와 측정 시스템도 클라이언트별로 다르게 선택 할 수 있다.

€ → C	ation training ×	lex.html?visuId=training				•
				Logged in ac:	Anonymous Logout	
Moin Alarm	Username			GroupBox1	Main Alarm	
	Password	 ✓ E Visualization train ← → C ① localho: 	ng × tal/index.html?visuld=training			
	English			0.0	Logged in as:	Dave Logout
	Metric	Main Bervice Alarm	Set to default	MsgBox		Service Calarma Test
6			Numeric output	Numeric input		
			35.0	35.0 35.0 °C	without engineering unit	A A
			State 1 Sta	122	Ś	Store (

그림 53 두 인스턴스를 HMI 어플리케이션에서 출력

클라이언트 정보가 맵잉되어 있는, 두 클라이언트의 값은 OPC UA 구조에 포함된다.

Name	Туре	Scope	Value	
🛛 🧇 ClientInfo	ClientInfoType[02]	globa		
🗄 🧼 ClientInfo[0]	ClientInfoType			
- 🔷 userld	WSTRING[80]		"Anonymous"	
- 🔷 isValid	BOOL		TRUE	
	WSTRING[15]		"127.0.0.1"	
- > languageld	WSTRING[2]		"en"	
- < browserResolution	WSTRING[32]		"1344x840"	
L → slotid	SINT		0	
	ClientInfoType			
⊢	WSTRING[80]		"Dave"	
→ isValid	BOOL		TRUE	
- ipAddress	WSTRING[15]		"127.0.0.1"	
- languageld	WSTRING[2]		"en"	
-> browserResolution	WSTRING[32]		"1344x840"	
L slotid	SINT		1	
E	ClientInfoType			
- 🔷 userld	WSTRING[80]			
- 🔷 isValid	BOOL		FALSE	
- < ipAddress	WSTRING[15]			
- <> languageld	WSTRING[2]			
-> browserResolution	WSTRING[32]			
L 🔷 slotld	SINT		0	

그림 54 Watch window 에 연결된 클라이언트 정보

8.1 클라이언트 - 특정 OPC UA 바인딩



그림 55 value binding 을 보여주는 다이어그램



그림 56 리스트 바인딩을 보여주는 다이어그램

위젯에 OPC UA 변수를 바인딩 할 때 또는 이벤트 바인딩에서 이벤트 트리거로 사용할 때, binding / event binding 은 HMI 어플리케이션이 전송되고 있는 모든 클라이언트에 유효하다.

다음 예제에서는 리스트 바인딩을 사용하여 멀티-클라이언트 HMI 어플리케이션에 대한 OPC UA 변수에서 클라이언트 특정 바인딩이 올바르게 구현되는 방법을 보여준다.

8.1.1 특정 OPC UA 변수에 대한 클라이언트 종속 바인딩



그림 57 다른 사용자 변환시 로그아웃이 보이는 다이어그램

예제: 다른 클라이언트 사용자 변환시 자동 로그아웃

"clinetSlot[0..2].userld" 변수가 변할 때, 다른 클라이언트에 로그아웃 커멘드가 보이도록 하는 방법을 설명한다.

다음 과정이 필요하다:

- 1) 표준 라이브러리 "AsBrStr" 추가
- 2) 사용자 변환 인식을 위한 프로그램 제작
- 3) 변수 선언
- 4) OPC UA default view 에서 "logoutCommand" 변수 활성화
- 5) "BOOL" 타입에 "LogoutCommand" session variable 추가하기
- 6) slotId 의 기능(selector)으로 session variable 을 커멘드와 연결하도록 list binding 만들기
- 7) session varialbes 값변화(ValueChanged)에 따른 로그아웃을 위한 이벤트 바인딩 생성하기



1 단계는 학습자가 직접 구현한다.

Copyright © B&R - Subject to change without notice TM671 - 강력한 mapp View 화면.docx 다음 예제는 클라이언트 한명이 로그인하면 다른 클라이언트들이 자동으로 로그아웃 되는 방법을 소개한다.

8.1.2 클라이언트에서 사용자 변환 감시를 위한 프로그래밍

다음은 로지컬 뷰 / "Program" 프로그램에서 랜덤 클라이언트의 로그인 인식을 위한 코드이다.

```
FOR i := 0 TO MAX_CLIENTS -1 DO
IF brsstrcmp(ADR(oldClientInfo[i].userId), ADR(ClientInfo[i].userId)) <> 0 THEN
FOR j := 0 TO MAX_CLIENTS -1 DO
IF i <> j THEN
IogoutCommand[i] := FALSE;
IogoutCommand[j] := TRUE;
END_IF;
END_FOR;
END_FOR;
oldClientInfo := ClientInfo;
```



다음 이미지 처럼, 프로그램을 위한 변수 선언은 변수 선언 창에 선언되어야 한다.

그림 58 로그아웃 로직을 위한 변수 선언

추가적으로, OPC UA default view 에서 IEC 변수 "logoutCommand[0..2]"를 활성화 하고 각 변수 요소를 보이도록 설정한다.

Name ∃ togoutCommand togoutCommand		Value	Value	Value	
		True			
Show array elements		True			
 Audit Events Authorization Inherit From Parent 		Off			
		Inherit All			
<u> </u>	Rights / Roles	Everyone	Operator	Serv	
	- Visible		9	2	
	Browse			\checkmark	
	- Subscribe			1	
	- Read			\checkmark	
	Write				

그림 59 OPC UA 변수 "logoutCommand" 활성화

8.1.3 로그아웃 커멘드 수신을 위한 session variable 바인딩

로직컬 뷰에 session variable 파일(.svar)에 "BOOL" 타입에 "LogoutCommand" 이름을 가진 session variable 를 만든다. 이는 로그아웃 명령을 저장하는데 사용될 것이며, 로그아웃 액션을 위한 이벤트로써 이벤트 바인딩에서 사용될 것이다.

<Variable name="LogoutCommand" xsi:type="BOOL" value="false" />

Session variable 은 리스트 바인딩(list binding)을 사용하여 존재하는 이벤트 바인딩 파일이 연결 될 것이다. System variable "::SYSTEM.clientInfo.slotId"는 선택자로 사용되며, mapp View 서버에 연결된 클라이언트는 고유 값(0-2)을 가질 것이다.

```
<Binding mode="oneWay">

<Source xsi:type="listElement">

<Selector xsi:type="session" refId="::SYSTEM:clientInfo.slotId"

attribute="value" />

<be:List xsi:type="be:opcUa" attribute="value">

<br/>
<br/>
<bt:Element index="0" refId="::userChange:logoutCommand[0]" />

<bt:Element index="1" refId="::userChange:logoutCommand[1]" />

<bt:Element index="2" refId="::userChange:logoutCommand[1]" />

<bt:Element index="2" refId="::userChange:logoutCommand[2]" />

</be:List>

</source>

<Target xsi:type="session" refId="LogoutCommand" attribute="value"/> </Binding>
```

마지막 단계에서, 새 <EventBinding> 요소를 존재하는 이벤트 바인딩 파일에 생성한다. Session variable "LogoutCommand" 값이 변화는 것은 이벤트(source)로 정의하고 클라이언트 시스템 액션(clientSystem action)의 매소드 "Logout"의 상태가 "true" 일때 동작한다.

```
<EventBinding>
<Source xsi:type="session.Event" refld="LogoutCommand"
event="ValueChanged" />
<ResultHandler condition="newValue">
<Action>
<Target xsi:type="clientSystem.Action">
<Method xsi:type="clientSystem.Action">
</Target>
</Action>
</ResultHandler>
</EventBinding>
```

```
예상되는 결과
연결된 클라이언트 중에 한 명이 로그인을 하면, 다른 클라이언트는 자동적으로 로그아웃된다.
```

9 요약

이벤트와 액션 시스템, HMI 변수는 복잡한 HMI 컨셉을 구현하기 위해 사용될 수 있다. 교육자료에서 테스트별 도움말과 예제를 통해서, 당신은 mapp View 에 대한 기본 지식을 크게 확장하고 향상할 수 있는 기회를 제공하였다. 추가 정보를 바탕으로, 구성과 다양한 위젯 조합을 활용하여 HMI 어플리케이션을 디자인 할때 새로운 컨셉으로 도전하는 것을 도울 것이다.

교육 자료의 지시 사항은 더 깊이 배울수 있도록 조언하는 역할과 mapp View HMI 어플리케이션의 확장형 개요를 제공하였다. 교육 자료에서 포함된 도움말 위치를 참조하여 구성과 관련된 추가 정보를 찾을 수 있는 위치를 안내한다.

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. 상향식 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 액츄에이터를 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

