

[IDE 트랙 - 고급] 화면을 멋지게 만들기

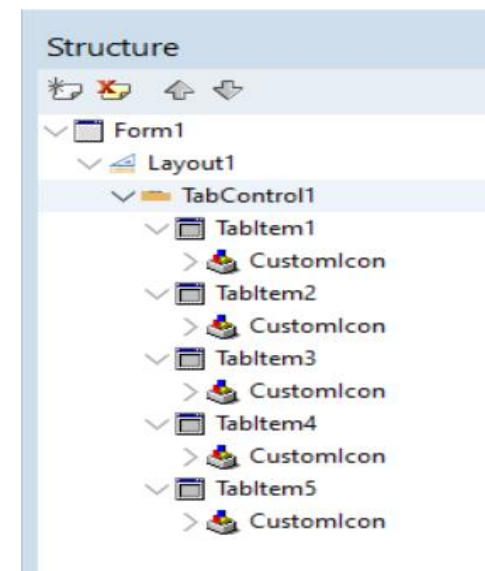
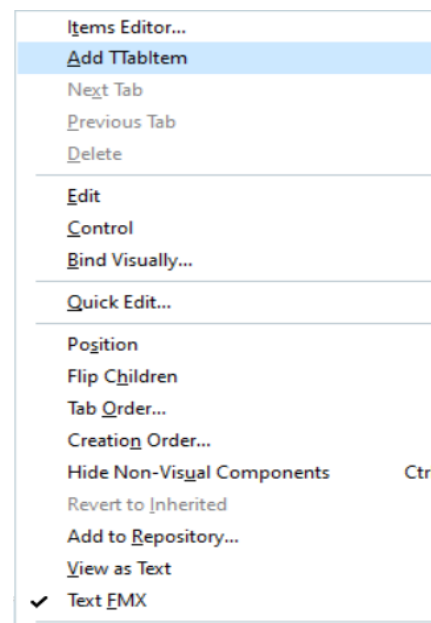
이번에는 파이어몽키를 사용하여 VCL에서 사용했던 사용했던 컴포넌트 및 좀 더 멋진 UI를 만들기 위한 다양한 컴포넌트, 제스처, 효과, 애니메이션등을 사용해 보는 예제를 작성해 보겠다.



[IDE 트랙 - 고급] 화면을 멋지게 만들기

[따라하기]

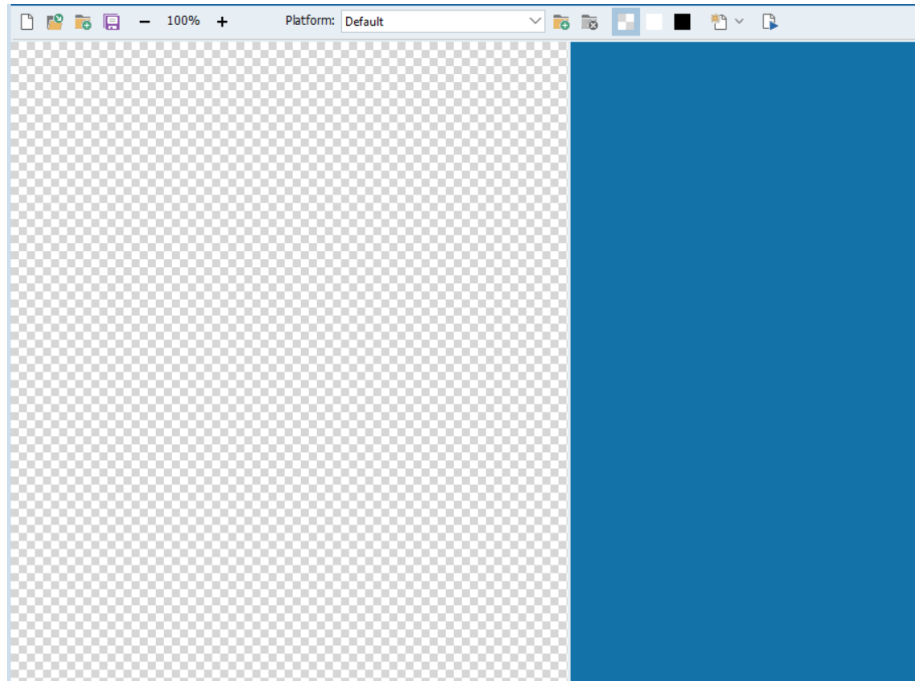
- File > New > Multi Device Application을 선택하여 프로젝트를 생성합니다. (uBasicControl.pas PBasicControl.dpr)
- 폼 위에 TLayout 컴포넌트를 내려놓고 Align속성을 Client로 지정한다.
 - Tlayout 컴포넌트는 FMX에서 컨테이너로 많이 사용됨 물론 Tpanel도 사용 가능
- Tlayout 컴포넌트위에 TTabControl 컴포넌트를 내려놓는다.
- TabControl 컴포넌트에서 오른쪽 마우스를 클릭하여 'Add TabItem'을 5개 추가한다.



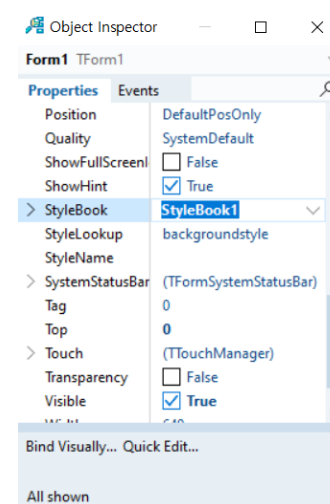
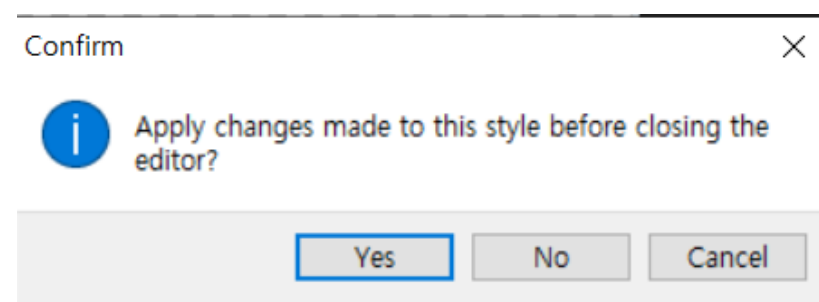
[IDE 트랙 - 고급] 화면을 멋지게 만들기

[따라하기]

- TStyleBook 컴포넌트를 내려놓고 더블 클릭하면 스타일 에디터가 열린다. 열기 버튼을 눌러 원하는 스타일을 선택한다.
- C:\Users\Public\Documents\Embarcadero\Studio\21.0\Styles 이 기본 폴더이다.



- 스타일 에디터 창을 닫으면 아래와 같은 대화상자가 표시된다. Ok를 눌러 스타일을 적용한다. 폼의 StyleBook 속성에 StyleBook1을 연결한다.



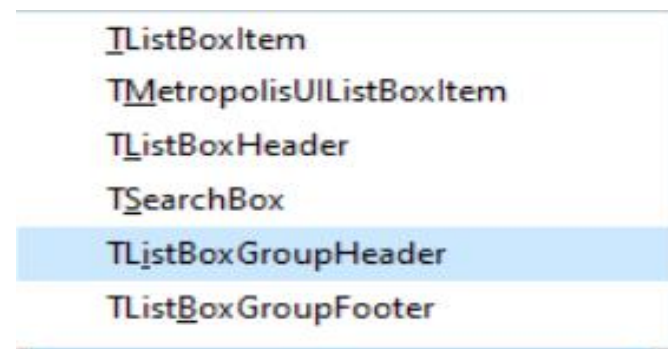
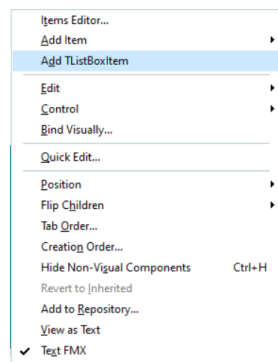
[IDE 트랙 - 고급] 화면을 멋지게 만들기

[따라하기]

- 아래와 같이 스타일이 적용된 것을 확인 한다.



- TabItem1 위에 TListBox 컴포넌트를 내려놓고 오른쪽 마우스를 클릭하여 TListBoxGroupHeader를 추가하고 3개의 TListItem를 추가한다.



[따라하기]

- 아래와 같이 컴포넌트의 속성들을 설정한다.

상위 컴포넌트	컴포넌트	속성	속성값/설명
TTabItem1	ListBox1	StyleLookUp	transparentlistboxstyle
		GroupingKind	Grouped
ListBox1	ListBoxGroupHeader1	Text	텔파이정규과정
ListBox1	ListBoxItem1	ItemData.Text	기본교육
		ItemData.detail	3일
		ItemData.Accessory	aCheckmark
		ItemData.bitmap	원하는 이미지를 연결
		StyleLookUp	listboxitembottomdetail

- 같은 방식으로 다음과 같이 나머지 2개의 ListBoxItem의 속성도 지정한다.



[IDE 트랙 - 고급] 화면을 멋지게 만들기

[따라하기]

- TabItem1 위에 TListBox 컴포넌트를 더 내려놓고 5개의 TListBoxItem을 추가한다.

- 아래와 같이 컴포넌트의 속성들을 설정한다.

상위 컴포넌트	컴포넌트	속성	속성값/설명
ListBoxItem4	Edit1	KeyBoardType	Alphabet
ListBoxItem5	DateEdit1	Date DateFormatKind Format 등을 확인	
ListBoxItem6	ComboEdit1	Items ItemIndex KeyBoardType	'A' 'B' 'C' 입력 0 Alphabet
ListBoxItem7	TimeEdit1	Time Format등을 확인	
ListBoxItem8	ColorComboBox1	Color	원하는 색상으로 지정

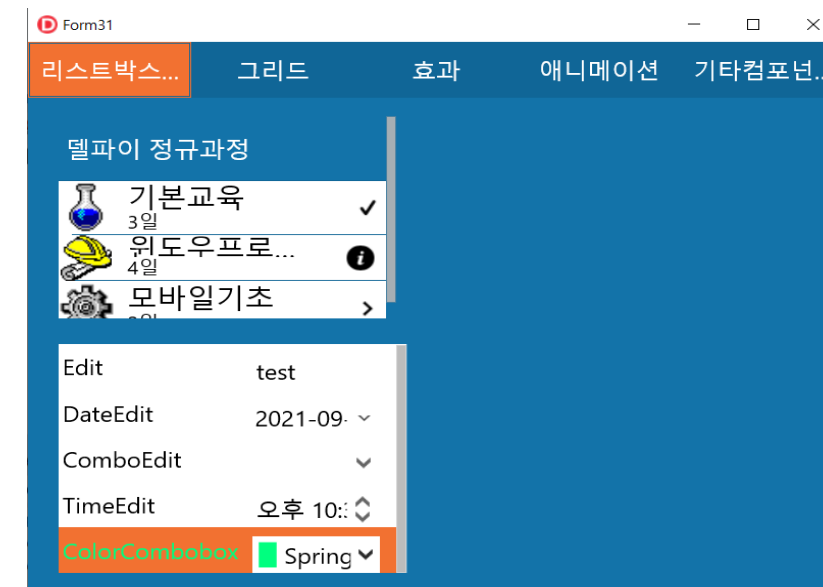
- ColorComboBox1의 OnChange 이벤트 핸들러에서 선택한 색상으로 해당 ListBoxItem.Testetting.FontColor 를 변경해보자.
 - ListBoxItem의 StyleSetting.FontColor를 체크하지 않는다.(스타일이 적용된 컴포넌트는 색상 변경이 안됨)

[IDE 트랙 - 고급] 화면을 멋지게 만들기

[따라하기]

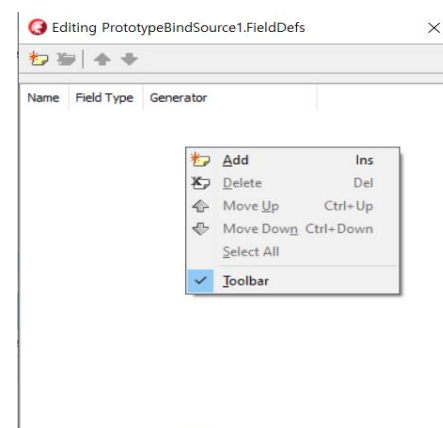
```
procedure TForm1.ColorComboBox1Change(Sender: TObject);
begin
    listBoxItem8.TextSettings.Fontcolor := ColorCombobox1.Color;
end;
```

- TListView 컴포넌트를 오른쪽에 내려 놓고 다음과 같이 속성을 설정한다.



컴포넌트	속성	값
ListView1	ItemAppearance	ImageListItemBottomDetail

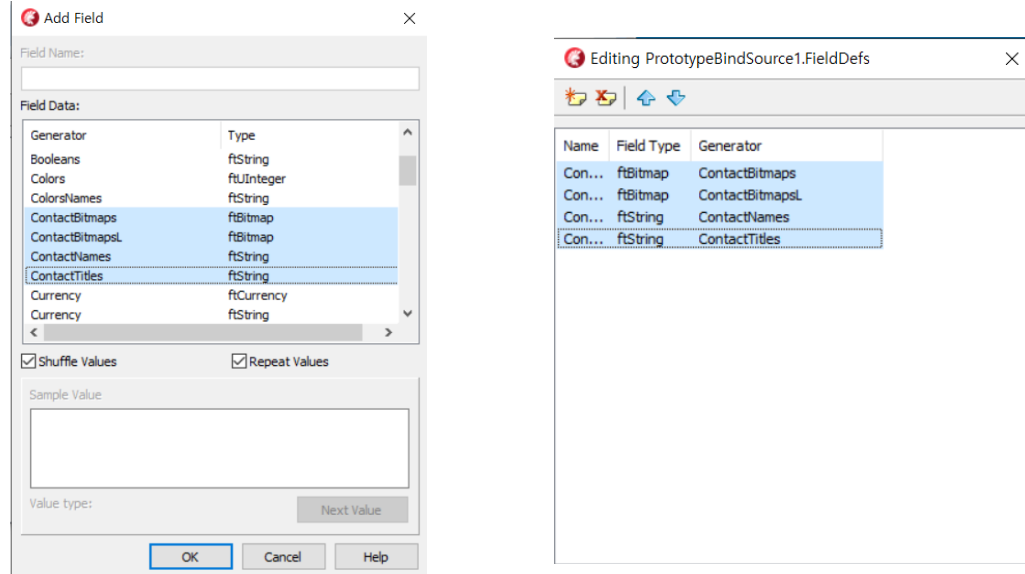
- TPrototypeBindSource 컴포넌트를 내려 놓고 더블 클릭하여 필드 에디터를 표시되면 다시 팝업메뉴에서 Add를 선택한다.



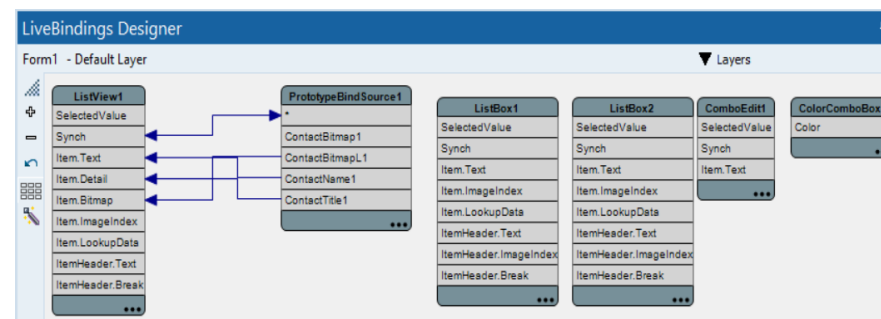
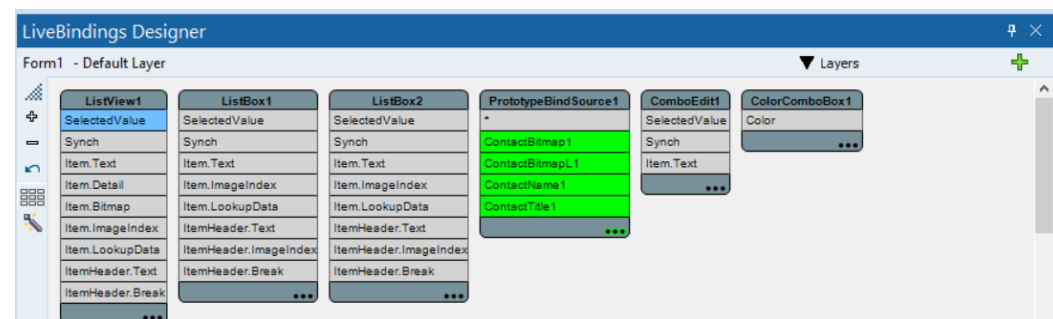
[IDE 트랙 - 고급] 화면을 멋지게 만들기

[따라하기]

- 아래와 같이 ContactName, ContactTitle, ContactBitmaps, ContactBitmapL 등을 선택하여 추가한다.



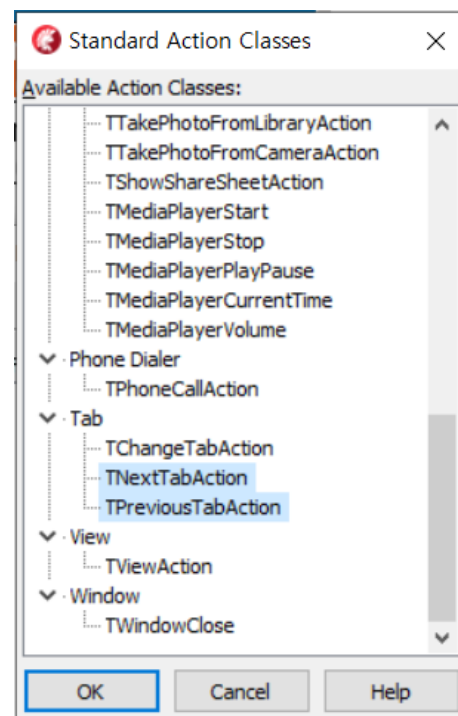
- 라이브 바인딩을 이용하여 아래와 같이 연결하여 ListView에 표시한다.(Sync도 연결)



[IDE 트랙 - 고급] 화면을 멋지게 만들기

[따라하기]

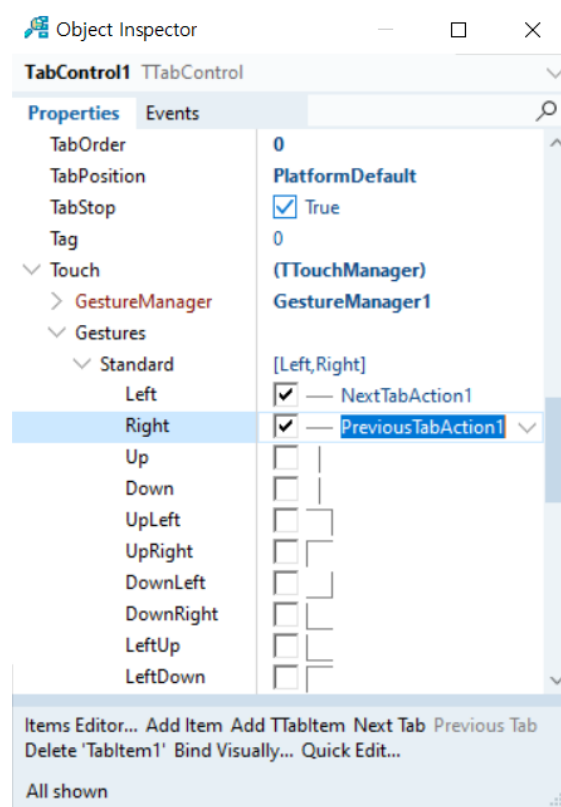
- 제스처를 사용하여 탭 간 이동을 하도록 구현하자.
1. TActionList 컴포넌트를 내려놓고 더블 클릭하고, 오른쪽 마우스를 클릭하여 Standard Action..을 선택한 후 NextTabAction, PreviousTabAction 2개를 추가한다.



2. 각 Next, Previous Action의 TabControl 속성을 TabControl1으로 지정한다. (즉 어떤 TabControl을 앞,뒤로 이동 할 것인지 설정)
3. TGestureManager 컴포넌트를 내려 놓는다. (31개의 스탠다드 제스처를 관리)
4. 제스처를 제어하고 싶은 컴포넌트 여기서는 TabControl1의 Touch 속성에 GestureManager1을 연결하고 아래와 같이 속성을 지정한다.

[IDE 트랙 - 고급] 화면을 멋지게 만들기

[따라하기]

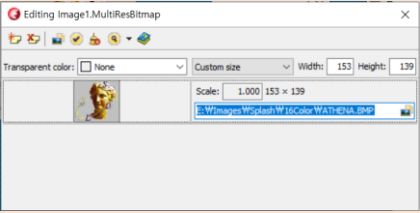


5. 실행하여(윈도우 플랫폼에서) 탭 간 이동하는지 확인한다.

[IDE 트랙 - 고급] 화면을 멋지게 만들기

[따라하기]

- 이번에는 다양한 효과(Effect)를 적용해 보도록 하겠습니다.
(C:\Users\Public\Documents\Embarcadero\Studio\21.0\Samples\Object Pascal\Multi-Device Samples\User Interface\ShaderFilters
폴더에는 모든 효과가 적용된 예제가 있다.)
- 1. 효과 탭에 Tedit 컴포넌트를 최소 10개 정도 내려놓고 Text는 'Hello' 로 지정한다. (복사해서 사용)
- 2. 이미지 탭에는 Timage, TimageControl 컴포넌트를 내려 놓습니다.

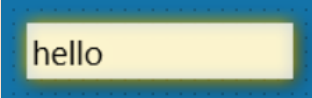
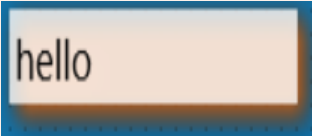

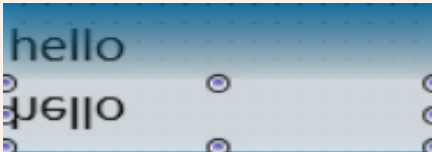


컴포넌트	속성	값
TImage	MultiResBitmap	
TImageControl	Bitmap	원하는 이미지를 연결한다.

- 3. 툴 팔레트 Effect를 검색해 보면 61개의 Effect 컴포넌트를 확인 할 수 있다.

[IDE 트랙 - 고급] 화면을 멋지게 만들기

[따라하기]

4. 이 중에 몇가지 효과를 에디터와 이미지에 적용해 보겠다. * 교육중에는 더 많은 효과를 적용해 보겠다.

상위 컴포넌트	효과 컴포넌트	속성	속성값/설명	결과
Edit1	TGlowEffect	GlowColor Oppacity Trigger	원하는 색상으로 지정 1 IsMouseOver=true	
Edit2	TShadowEffect	Direction Distance ShadowColor	45 5 원하는 색상으로 지정	
Edit3	TBlurEffect	SoftNess Trigger	0.5 IsMouseOver=true	
Edit4	TReflectionEffect	Length Opacity	1 1	
ImageControl1	TRippleEffect	Amplitude Frequency AspectRatio	0.3 90 2	
Image1	TTilerEffect	HorizontalTileCount VerticalTileCount	5 3	

[IDE 트랙 - 고급] 화면을 멋지게 만들기

[따라하기]

• 이번에는 다양한 애니메이션을 적용해 보자.

1. 애니메이션 컴포넌트의 종류

● 시작 값에서 종료 값까지 변환

이름	내용
TFloatAnimation	위치(X, Y, Z), 회전, 투명도 등의 실수 속성 값으로 변환
TRectAnimation	TBound 속성의 네 개 끝점(end point) 위치 변환
TColorAnimation	색상 값(TAlphaColor형)에 따라 변환
TGradientAnimation	그래데이션을 정의하는 각 점의 색상 값에 따라 변환
TBitmapAnimation	투명도를 증가시켜 시작 이미지를 다른 이미지로 전환

● 두 값이 아니라 일련 값에 따라 변환

이름	내용
TFloatKeyAnimation	실수 목록에 따라 전환
TColorKeyAnimation	색상 목록에 따라 전환
TPathAnimation	오브젝트의 2D 위치 경로에 따라 전환

● 목록에 따라 변환

이름	내용
TBitmapListAnimation	모든 이미지가 가로로 결합되어 하나의 비트 맵으로 되어 일정한 간격의 슬라이드 쇼처럼 변환

2. 애니메이션 탭에 아래와 같이 컴포넌트를 내려 놓고 속성값을 지정한다. 이 방법은 속성에서 직접 연결해서 사용하는 방법이다.

상위 컴포넌트	컴포넌트	속성	값
	Rectangle1	Fill.Color 이속성에서 Create New Color Animation..을 선택하면 ColorAnimation 컴포넌트가 자동으로 Rectangle의 하위(자식) 개체로 생성된다.	원하는 색상 

[IDE 트랙 - 고급] 화면을 멋지게 만들기

[따라하기]

상위 컴포넌트	컴포넌트	속성	값
Rectangle1	ColorAnimation1	PropertyName Animation type Start Value Stop Value Duration Enabled Loop Interporation	Fill.Color(자동) In(디폴트) White Red 3 True True Linear(디폴트)
	Rectangle1	RotationAngle 속성을 선택 하고 Create New FloatAnimation을 선택한다.	
Rectangle1	FloatAnimation1	PropertyName Animation type Start Value Stop Value Duration Enabled Loop Interporation	RotationAngle(자동) In 0 360 3 True True Linear

[IDE 트랙 - 고급] 화면을 멋지게 만들기

[따라하기]

3. 애니메이션 탭에 TImageControl 컴포넌트를 내려놓고 Bitmap 속성에서 ‘Create New BitmapAnimation’을 선택한다.

상위 컴포넌트	컴포넌트	속성	값
ImageControl1	BitmapAniMation1	PropertyName Animation type Start Value Stop Value Duration Enabled Loop Interporation	Bitmap In 원하는 이미지 지정 원하는 이미지 지정 3 True True Linear

4. Rectagle 컴포넌트 2개, 버튼 2개를 더 추가하여 다음과 같이 배치한다.



[IDE 트랙 - 고급] 화면을 멋지게 만들기

[따라하기]

5. 메소드를 이용하여 애니메이션을 적용해본다. 첫 번째 회전 버튼의 클릭 이벤트 핸들러를 다음과 같이 구현한다.

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
    rectangle2.AnimateFloat('RotationAngle', 360, 3);  
end;
```

6. 두 번째 회전 버튼은 직접 코드로 개체를 생성하여 적용하는 코드를 작성한다.

```
procedure TForm1.Button2Click(Sender: TObject);  
var  
    ani :TFloatAnimation;  
begin  
    ani := TFloatAnimation.Create(rectangle3);  
    ani.PropertyName := 'RotationAngle';  
    ani.Duration := 3;  
    ani.StopValue := 360;  
    ani.Loop := true;  
    ani.Parent := Rectangle3;  
    ani.Start; //stop;  
end;
```


7. 애니메이션 탭에 TCheckBox 컴포넌트 2개, Tbutton을 내려놓고 다음과 같이 속성을 제어한다.

상위 컴포넌트	컴포넌트	속성	값
TabItem(애니메이션 탭)	CheckBox1	Text	Horizontal
CheckBox1	FloatAnimation2	PropertyName StartValue StopValue Duration AnimationType Interporation Trigger TriggerInverse	Position.x 32(checkBox1.position.x) 50 2 InOut Quintic IsChecked=true IsChecked=false
	CheckBox2	Text	Vertical
CheckBox2	FloatAnimation3	PropertyName StartValue StopValue Duration AnimationType Interporation Trigger TriggerInverse	Position.x Checkbox2.position.y값으로50 400 2 In Linear IsChecked=true IsChecked=false
	Button3	Text	스케일
Button3	FloatAnimation4	PropertyName StartValue StopValue Duration Trigger TriggerInverse	Scale.X 1 4 1 IsMouseOver=truels MouseOver=false
	FloatAnimation4	PropertyName StartValue StopValue Duration Trigger TriggerInverse	Scale.y 1 4 1 IsMouseOver=truels MouseOver=false