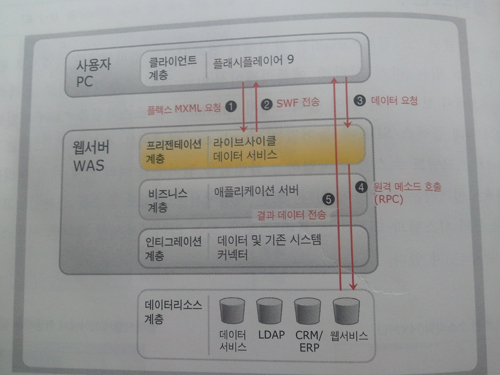
플랙스

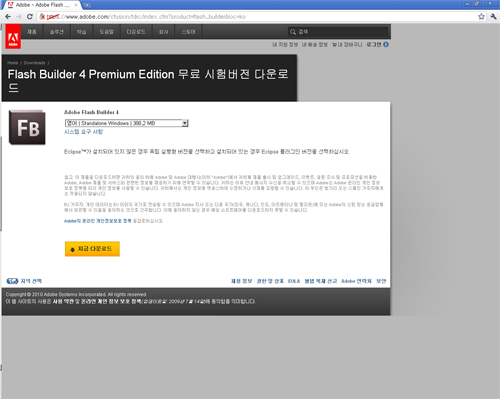
**\*.mxml -> 컴파일.(mxmlc.exe) -> swf**

**\*.mxml -> 라이브사이클 데이터서비스에서 컴파일 -> swf  
라이브사이클데이터 서비스에서 데이터를 가져오는 서비스 RPC 서비스를 통해 데이터가져오고 결과만사용자에게 던진다**



독립형 FB4(Flash Builder 4)를 사용하지 않고 굳이 Eclipse에 FB4 plugin을 설치하는 이유는 독립형 FB4는 기능의 제약점이 있기 때문이다. 물론 FB4로도 충분히 개발이 가능하지만 Eclipse와 연동하는 것은 이 편이 실무에 더욱 적합하다. 그 이유는 hika님의 글 [Flex Builder는 반드시 Plug in 으로 설치](http://www.diebuster.com/?p=394)글을 읽어보면 알 수 있다.   
  
하지만 이러한 설치법은 처음 해보는 사람에게는 단순하지 않다. 한방에 설치하는 독립형 FB4에 비해 이것저것 고려할 것이 많은 것 같은 생각이 들기 때문이다. 이 글을 쓰는 목적은 FB4 plugin을 설치를 하는데 있어서 다소 어려움을 쉽게 극복하기 위한 방법을 소개하기 위함이다.   
  
설치하는 대략적인 순서는 hika님이 매우 잘 설명해 주었다.   
[Flex Builder는 반드시 Plug in 으로 설치](http://www.diebuster.com/?p=394)  
  
그러나 더욱 자세한 방법은 Chocodonut님의 글대로 하면 되겠다.  
[Flex Builder 3 Professional Eclipse plug-in 설치](http://chocodonut.tistory.com/106)  
  
FB4도 위 글과 거의 동일한 방법으로 설치하지만 아래와 같은 몇가지 체크할 사항이 있다. (단, 본인은 Window XP환경에서 테스트 했음을 언급한다. Mac이나 Linux의 경우 아래에서 지적한 문제가 있을지는 테스트해보지 않았다.)  
 **1. Eclipse는 Ganymede를 설치한다.**  
먼저, Eclipse버전은 Ganymede RC2 버전을 이용하기 바란다. 최신 버전인 Galileo는 제대로 설치되지 않는다.  
[http://www.eclipse.org/downloads/](http://www.eclipse.org/downloads/%20)  
에서 화면 오른쪽에 Older Versions를 선택한 뒤 Eclipse Ganymede SR2 Packages로 들어간다. 여기서 Eclipse IDE for Java EE Developers를 다운로드 받으면 되겠다. 아래 링크로 직접가도 되겠다.  
<http://www.eclipse.org/downloads/packages/release/ganymede/sr2>  
  
다운받은뒤 설치를 원하는 곳에 압축을 풀면 설치가 완료이다. 설치완료후 반드시 한번이상 실행해준다.  
  
만약 Galileo버전을 사용하고 싶다면 굳이 말리지 않는다. ^^  
다음 글을 참고한다.  
<http://blog.naver.com/joycestudy/100090900907>  
  
  
**2. Eclipse실행시 JVM terminated. Exit code=-1 에러가 발생하는 경우**  
Eclipse 설치폴더에 eclipse.ini를 열어 다음과 같이 수정한다. 

-showsplash  
org.eclipse.platform  
--launcher.XXMaxPermSize  
256M  
**-vm  
C:\Program Files\Java\jdk1.6.0\_07\bin\javaw.exe**  
-Dosgi.requiredJavaVersion=1.5  
**-vmargs**  
-Xms40m  
-Xmx512m

원래 eclipse.ini에서 수정된 부분은 -vmargs의 위치가 -Xms40m 바로 위로 이동된 것이며 추가된 부분은 -vm 부분이 들어간 것이다. 이때 javaw.exe 경로는 각자 jdk또는 jre 설치 경로를 따라 설정하면 되겠다. 이렇게 해도 같은 에러가 난다면 검색을 통해 다른 방법을 찾길 바란다.  
  
  
**3. Flash Builder 4 plug in을 설치시 반드시 Eclipse 설치경로를 지정해줄것!**  
FB4 plug in을 설치하면 중간에 Eclipse설치 경로를 지정하는 순서가 있다. 이 부분을 잊지 말고 Eclipse 경로를 지정하기 바란다.  
<https://www.adobe.com/cfusion/tdrc/index.cfm?product=flash_builder&loc=ko>  
  
**4. FB4를 실행할때는 Eclise를 실행하거나 FB4에 설치된 Run FB Plug-in with alternate Eclipse를 실행한다.**  
FB4 plug-in에서 Run FB plug-in 을 실행하면 eclipse-host-distro의 Eclipse가 실행되므로 이것을 사용하지 말자.   
  
**5. 바로가기를 FB4 아이콘으로 바꾸기**일반적으로 Eclipse를 실행하기 위해 바탕화면에 바로가기를 만들거나 빠른실행아이콘으로 등록해서 사용할 것이다. 하지만 이 바로가기는 기본 아이콘이 이클립스이다. 그러므로 기존에 사용하는 이클립스 아이콘이 있다면 무엇이 Flash Builder이고 무엇이 Eclipse인지 잘 분간이 안갈터...   
  
바로가기의 아이콘을 Flash Builder 아이콘으로 바꿔보자.  
1. Flash Builder Plug-in이 설치된 폴더로 가서 eclipse-host-distro 폴더에 가면 Eclipse.ico 파일이 있다. 이것을 복사해서 Eclipse 설치 폴더에 복사한다.   
2. 다음으로 바탕화면 또는 빠른실행아이콘을 선택해 마우스 오른쪽 버튼을 눌러 속성을 선택한다.   
3. 아이콘 변경 버튼을 눌러 Eclipse 설치 폴더에 복사한 Eclipse.ico로 바꿔준다.   
  
이제 근사하게 Flash Builder 아이콘을 가진 바로가기를 만들 수 있게 되었다.  
  
  
**6. 이전 워크스페이스를 열때 만약 Run, Debug 버튼이 사라진다면!**  
다음 글을 참고한다.  
[Flash Builder/Flex Builder의 툴바에서 Run, Debug Launch 버튼이 없어진 경우 해결법](http://blog.jidolstar.com/608)  
  
  
**7. 이전 워크스페이스의 SDK경로는 따로 빼자!**  
Flash Builder의 SDK가 아닌 기존에 원래 사용하던 SDK를 참조할 필요가 있는 경우가 있다. 이런 문제를 해결하기 위한 방법은 여러가지가 있겠지만 본인은 C나 D 드라이브로 SDK폴더를 따로 만들어 FB가 이 SDK를 참조하도록 하고 있다.   
  
다음과 같이 진행한다.  
  
1. C나 D 드라이브등에 사용할 SDK를 복사한다.(새 FB의 설치경로에 sdks에 있는 파일을 복사할 수 있다.)  
2. FB를 실행한다.  
2. 메뉴에서 Window > Preference를 선택한다.  
3. 창이 뜨면 좌측에 Flash Builder를 열어 Installed Flex SDKs를 선택한다.   
4. 우측에 Flex 3.4, Flex 4.0이 이전 FB버전의 경로를 참고하고 있을 것이다. 이를 C나 D드라이브에 복사한 SDK를 참조하도록 한다.  
  
  
**8. 기본 플러그인 설치는 기본!**  
독립형 FB4에는 설치되지 않는 플러그인이 Eclipse에는 언제든지 설치가 가능하다. 주로 협업이 이뤄지므로 SVN등을 설치해야하고 다양한 배포 경로를 확보하기 위해 ANT관련 설정이 필요할 것이다. C/C++개발을 위한 CDT, PHP개발을 위한 PDT등 플러그인을 제한없이 설치할 수 있으니 Eclipse의 FB4 plugin을 설치하는 편이 훨씬 득이라는 것을 느낄 수 있을 것이다.   
  
다음글들을 참고하면 좋겠다.  
[Flex Builder 에서 Subversion(SVN) 사용하기](http://blog.jidolstar.com/78)  
[Flex Builder 3에서 ANT 사용하기](http://blog.jidolstar.com/505)  
[Maven 시작하기](http://kjunine.net/search/Maven)  
  
  
**정리하며**  
사실 위에 제시한 사항들은 개인적으로 사용하는 방법이다. 꼭 그렇게 따라할 필요 없으며 필요시 참조하면 무난할 것이라 판단한다.   
  
실무에 종업하시는 분들은 왠만하면 FB4 플러그인을 사용하고 있다. hika님의 블로그에 실무를 통해 자주 사용하는 설정정보를 정리해두었으니 참고 바란다.   
  
[플래시를 위한 이클립스 기본 설정](http://www.diebuster.com/?p=1059)  
  
다음 글들은 이외에 개발에 큰 도움이 되는 글이니 참고 바랍니다.  
[플래시 어플리케이션 개발하기](http://www.diebuster.com/?p=951)  
[플래시 프로젝트 생성하기](http://www.diebuster.com/?p=957)  
  
  
**다운로드**Flash Builder 및 갖가지 Adobe 제품군을 다운로드 받으려면 다음 링크를 참고하세요.  
<http://www.adoberia.co.kr/pds/down.html?src=text&kw=000026>  
  
글쓴이 : 지돌스타(<http://blog.jidolstar.com/607>)

## [Flex설치하기](http://stella.thoth.kr/3284075) FLEX

2010.06.04 08:10 [Edit](http://stella.thoth.kr/?mid=blog&document_srl=3284075&act=dispTextyleToolPostManageWrite)

. Adobe Flex Builder 3 설치   
  - Flex를 개발시 기본적으로 필요한 도구이다.  
  - <http://www.adobe.com/cfusion/entitlement/index.cfm?e=flex3email>   
  - 다운로드 받은 프로그램을 설치한다.   
    중간에 Optional Installations에 JSEclipse도 선택하여 설치해준다.  
  - c:/Program Files/Adobe/Flex Builder 3에서 FlexBuilder.exe를 실행하면 된다.

2. Flash Player Debug 버전 설치  
  - Debug 버전을 설치하면 프로그램 코딩시 디버깅이 가능하다.   
  - <http://www.adobe.com/support/flashplayer/downloads.html>  
  - Adobe Flash Player 10 ? Debugger Versions 에서 ActiveX용 Flash Player 10을 다운로드 받는다.  
  - 다운로드 받은 것을 설치한다.  
  - 만약 FireFox 브라우져에도 설치할려면 Netscape용 Debug 버전을 설치하면 되겠다.

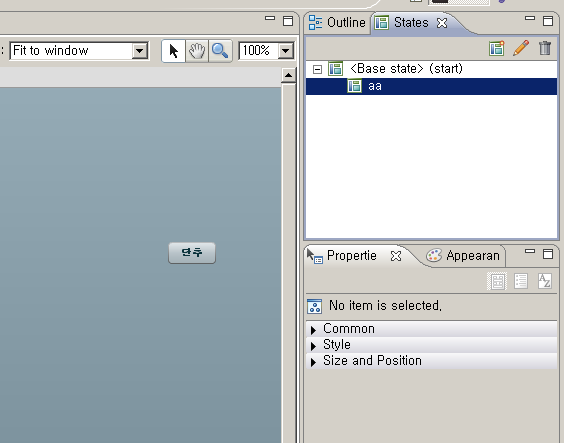
3. Adobe AIR 1.1 Runtime 설치   
  - AIR 프로그램을 OS에 설치할 수 있게끔하는 플러그인이다.  
  - <http://get.adobe.com/air/?loc=kr>

4. Flex SDK 3.x.x (옵션)  
  - Flex Builder를 설치하면 자동으로 Flex SDK 3.0.0이 설치된다.  
    하지만 몇가지 기술적, 보안적 이유로 업데이트를 권장한다.  
  - <http://opensource.adobe.com/wiki/display/flexsdk/Download+Flex+3>  
  - 3.x.x Stable Builds 버전을 다운로드 받는다.  
  - c:/Program Files/Adobe/Flex Builder 3/sdks 에 다운로드한 것을 압축하여 풀어준다. 폴더는 3.x.x로 한다.  
  - Flex Builder를 실행후 메뉴에 Windows->Preferences를 선택하면 창이 뜬다.  
  - 창이 뜨면 왼쪽 Flex->Installed Flex SDKs를 선택한다.  
  - 선택후 우측 Add버튼을 이용해서  
    c:/Program Files/Adobe/Flex Builder 3/sdks/3.x.x 를 location으로 잡고 name은 Flex SDK 3.x.x 로 하고 OK  
  - Flex 3.x.x check  박스 선택하고 OK!

**[출처]** flex환경설정

설치-> 자바깔고->이클립스 헬리우스깔고 -> 빌더깔고 끝!

플랙스 State



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

<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute" minWidth="955" minHeight="600">

<mx:states>

<mx:State name="aa">

<mx:AddChild position="lastChild">

<mx:Button label="단추" x="600" y="155"/>

</mx:AddChild>

<mx:SetProperty name="layout" value="absolute"/>

<mx:SetProperty target="{datechooser1}" name="x" value="275"/>

<mx:SetProperty target="{datechooser1}" name="y" value="10"/>

<mx:RemoveChild target="{checkbox1}"/>

</mx:State>

</mx:states>

<mx:DateChooser width="399" height="376" id="datechooser1" x="107" y="0"/>

<mx:CheckBox x="215" y="252" label="CheckBox" id="checkbox1"/>

</mx:Application>

State 이동

<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute"

initialize="**trace**(**'My Flex initialize!'**)">

<mx:states>

<mx:State name="image">

<mx:RemoveChild target="{panel1}"/>

<mx:AddChild position="lastChild">

<mx:RichTextEditor x="87" y="28">

</mx:RichTextEditor>

</mx:AddChild>

</mx:State>

</mx:states>

<mx:Script>

<![CDATA[

**import** mx.controls.Alert;

**private** **function** showMsg(input:String):**void** {

**mx.controls.Alert.show(input);**

**currentState = 'image';**

}

]]>

</mx:Script>

<mx:Panel layout="absolute" left="10" bottom="10" right="10" top="10" title="My Flex" id="panel1">

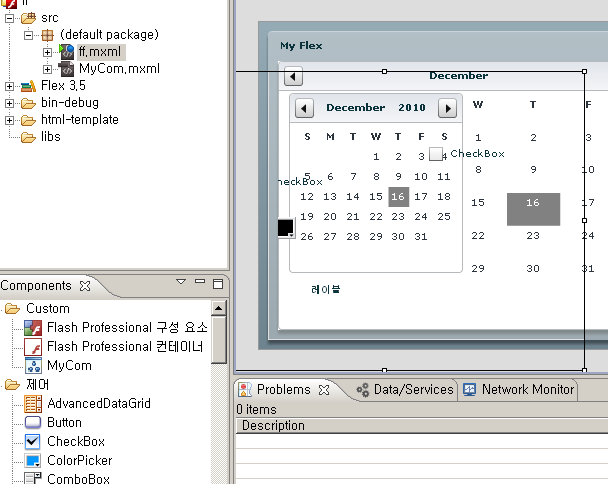
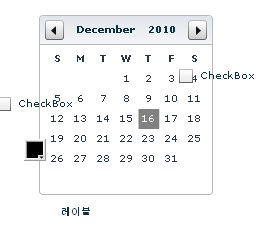
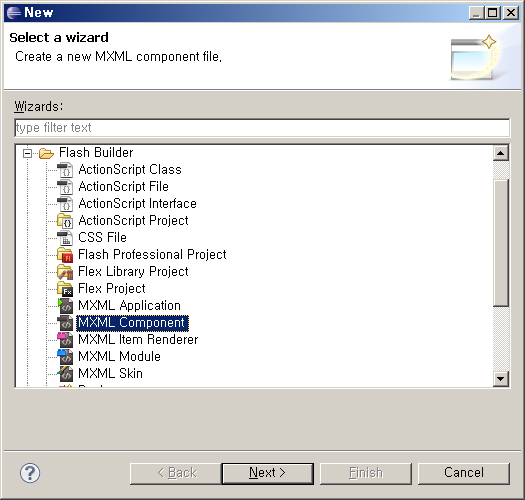
<mx:DateChooser width="100%" height="100%" change="showMsg(**'My Flex!'**)">

</mx:DateChooser>

</mx:Panel>

</mx:Application>

커스텀 컴포넌트 만들기



Mxml 기본

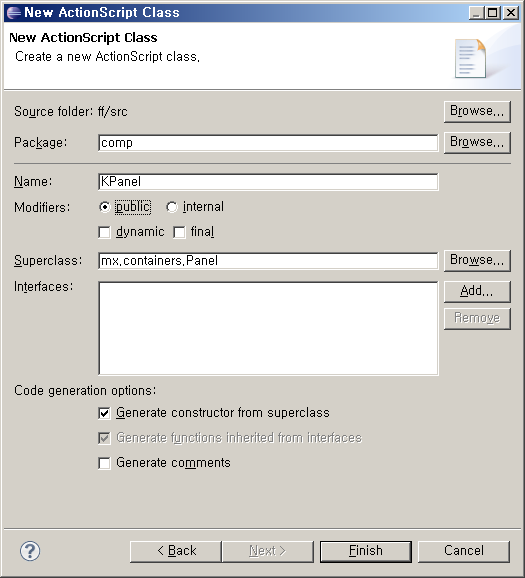
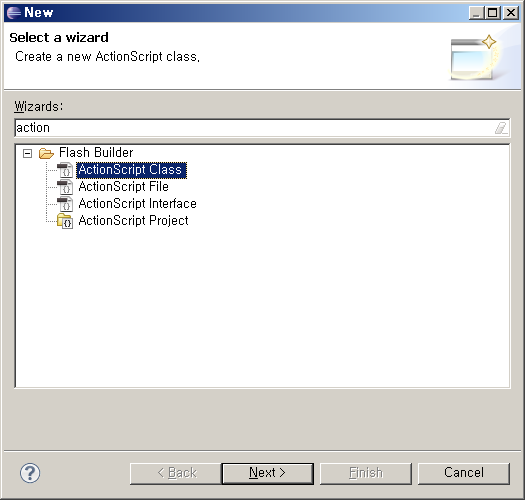
컴포넌트 mxml은

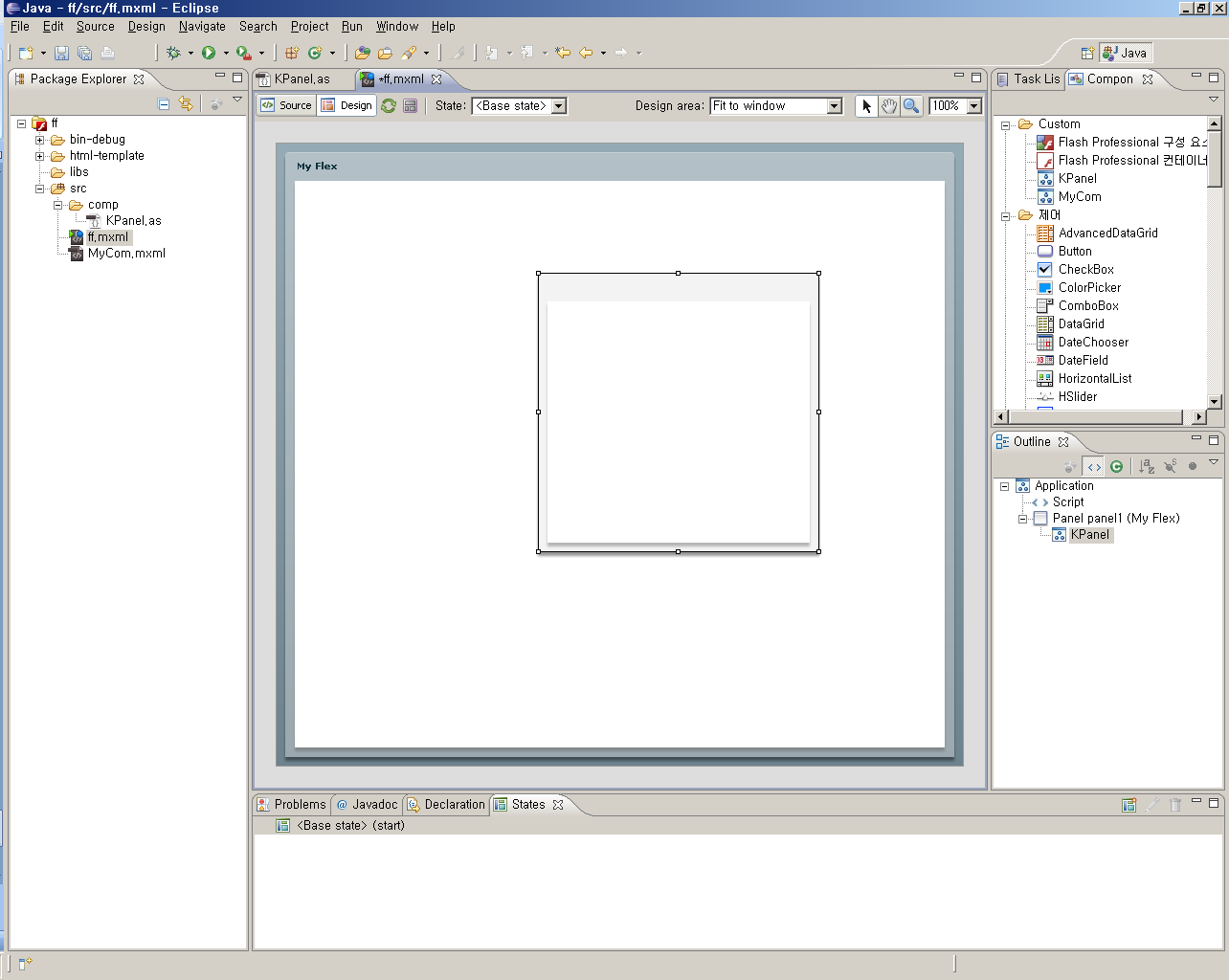
D:\Program Files\Adobe\Adobe Flash Builder 4 Plug-in\sdks\3.5.0\frameworks\projects\framework\src\mx\controls

에

AS 와 매칭된다

커스텀 컴포넌트 만들기.





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

<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute"

initialize="**trace**(**'My Flex initialize!'**)" xmlns:ns1="\*" xmlns:comp="comp.\*">

<mx:Script>

<![CDATA[

**import** mx.controls.Alert;

**private** **function** showMsg(input:String):**void** {

mx.controls.Alert.show(input);

currentState = **'image'**;

}

]]>

</mx:Script>

<mx:Panel layout="absolute" left="10" bottom="10" right="10" top="10" title="My Flex" id="panel1">

<comp:KPanel x="254" y="96" width="294" height="292">

</comp:KPanel>

</mx:Panel>

</mx:Application>

컴포넌트는 화면에 보여질 비주얼 컴포넌트와

넌비주얼 컴포넌트로 두가지종류로 구분된다.(안쪽에서 프로세싱하는놈들)

**Button / LinkButton /속성 properties!**

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

<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="vertical"

fontSize="11" backgroundGradientColors="[#f7f794, #adb2ce]">

<mx:Script>

<![CDATA[

**private** **function** showMsg(input:String):**void** {

mx.controls.Alert.show(input);

}

]]>

</mx:Script>

<mx:Panel width="100%" height="100%" layout="vertical"

horizontalAlign="center" verticalAlign="middle" label="커서 버튼"

title="Button / LinkButton">

**<mx:Button label="그냥 버튼" width="95" click="showMsg('그냥 버튼')"/>**

<!-- 토글 버튼은 누르면 그 스킨 모양이 유지 된다. -->

**<mx:Button label="토글 버튼" toggle="true" />**

<!-- Button에 buttonMode='true', useHandCursor='true'로 해주면

LinkButton처럼 커서가 생긴다-->

**<mx:Button label="커서 버튼" mouseDown="showMsg('커서 버튼')"**

**toolTip="buttonMode=\'true\' useHandCursor=\'true\'"**

**buttonMode="true" useHandCursor="true"/>**

**<mx:LinkButton label="링크 버튼" width="100"**

**click="navigateToURL(new URLRequest('http://okgosu.net'), 'quote')"/>**

</mx:Panel>

</mx:Application>

**ChckBox /속성 properties!**

<mx:CheckBox id="myChk1" labelPlacement="left" label="왼쪽 라벨 체크박스" click="myBtn.enabled=myChk1.selected"/>

<mx:CheckBox id="myChk2" labelPlacement="right" label="오른쪽 라벨 체크박스" click="myBtn.enabled=myChk2.selected"/>

**RadioButton /RadioGroup /속성 properties!**

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

<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="vertical"

fontSize="11" backgroundGradientColors="[#f7f794, #adb2ce]">

<mx:Panel title="CheckBox / RadioButton / RadioButtonGroup"

width="100%" height="100%" layout="vertical"

paddingTop="10" paddingBottom="10" paddingLeft="10" paddingRight="10">

<mx:VBox verticalAlign="middle" width="100%">

<!-- 체크박스가 선택이 되면 myBtn버튼의 활성화 속성을

똑같이 true 또는 false로 바꿔준다. -->

<mx:CheckBox id="myChk1" labelPlacement="left" label="왼쪽 라벨 체크박스" click="myBtn.enabled=myChk1.selected"/>

<mx:CheckBox id="myChk2" labelPlacement="right" label="오른쪽 라벨 체크박스" click="myBtn.enabled=myChk2.selected"/>

<mx:Button label="상태확인용버튼" id="myBtn"/>

</mx:VBox>

<mx:HBox borderStyle="solid">

<mx:Label text="과일 고르기"/>

<!-- 과일을 위한 RadioButtonGroup 생성 -->

<mx:RadioButtonGroup id="rg1"/>

<mx:RadioButton label="1.포도" groupName="rg1"/>

<mx:RadioButton label="2.메론" groupName="rg1"/>

<mx:RadioButton label="3.수박" groupName="rg1"/>

</mx:HBox>

<mx:HBox borderStyle="solid">

<mx:Label text="음료 고르기"/>

<!-- 음료를 위한 RadioButtonGroup 생성 -->

<mx:RadioButtonGroup id="rg2"/>

<mx:RadioButton label="1.쥬스" groupName="rg2"/>

<mx:RadioButton label="2.커피" groupName="rg2"/>

<mx:RadioButton label="3.콜라" groupName="rg2"/>

</mx:HBox>

</mx:Panel>

</mx:Application>

**ColorPicker /속성 properties!**

<mx:ColorPicker id="cp" showTextField="true" selectedColor="0xFFFFFF"/>

**DateChooser /속성 properties!**

<mx:DateChooser id="dc" backgroundColor="{cp.selectedColor}"

allowDisjointSelection="true" allowMultipleSelection="true"

yearSymbol="년"

dayNames="['일', '월', '화', '수', '목', '금', '토']"

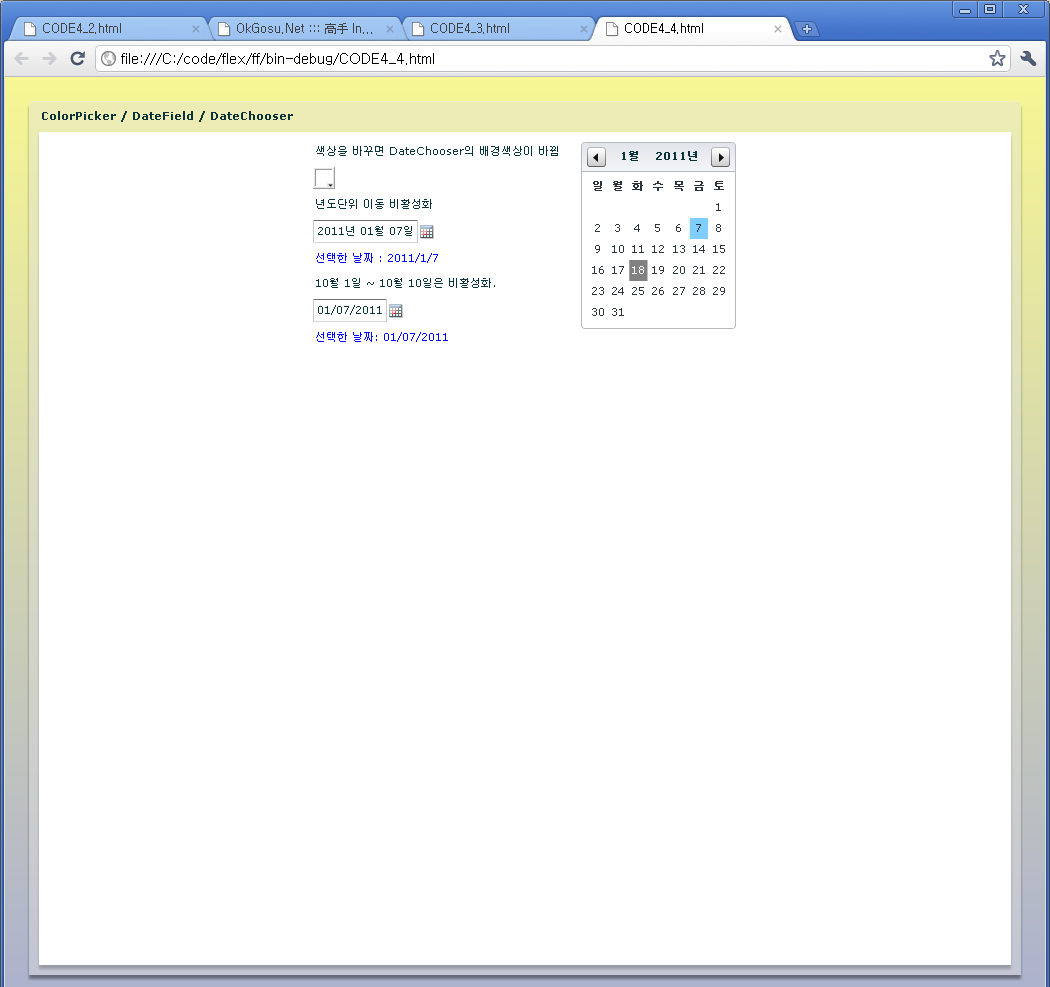
monthNames="[1,2,3,4,5,6,7,8,9,10,11,12]" monthSymbol="월"

selectedDate="{**new** Date(2006, 8, 25)}"/>

**DateField /속성 properties!**

<mx:DateField id="df1" yearNavigationEnabled="true"

formatString="YYYY년 MM월 DD일" change="dateChanged(DateField(event.target).selectedDate);

dc.selectedDate=df1.selectedDate" />****

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="vertical"  fontSize="11" backgroundGradientColors="[#f7f794, #adb2ce]">  <mx:Script>  <![CDATA[  **private** **function** dateChanged(date:Date):**void** {  **if** (date == **null**)  selection.text = **"선택한 날짜 없음 "**;  **else**  selection.text = **"선택한 날짜 : "** + date.getFullYear().toString() +  **'/'** + (date.getMonth()+1).toString() + **'/'** + date.getDate();  }  ]]>  </mx:Script>  <mx:DateFormatter id="df"/>  <mx:Panel title="ColorPicker / DateField / DateChooser"  width="100%" height="100%" layout="horizontal"  paddingTop="10" paddingBottom="10" paddingLeft="10" paddingRight="10" horizontalAlign="center">  <mx:VBox width="260" horizontalAlign="left">  <mx:Label text="색상을 바꾸면 DateChooser의 배경색상이 바뀜"/>  <mx:ColorPicker id="cp" showTextField="true" selectedColor="0xFFFFFF"/>  <mx:Label text="년도단위 이동 비활성화"/>  <!-- 선택한 날짜를 YYYY년 MM월 DD일 로 되도록 하였고 날짜가 선택될 때 DateChooser의 날짜도 같이 변하도록 함-->  <mx:DateField id="df1" yearNavigationEnabled="true"  formatString="YYYY년 MM월 DD일" change="dateChanged(DateField(event.target).selectedDate);  dc.selectedDate=df1.selectedDate" />  <mx:Label id="selection" color="blue" text="Date selected:" />  <mx:Label text="10월 1일 ~ 10월 10일은 비활성화."/>  <!-- 연도 이동 버튼을 없앴고, 2006년 10월 1일부터 10일까지 비활성화 하였다. -->  <mx:DateField id="df2" yearNavigationEnabled="false"  change="dc.selectedDate=df2.selectedDate"  disabledRanges="{[{rangeStart:**new** Date(2006, 9, 1),  rangeEnd: **new** Date(2006, 9, 10)}]}" />  <mx:Label color="blue" text="선택한 날짜: {df.format(df2.selectedDate)}"/>  </mx:VBox>  <!-- 여러 개의 날짜를 선택할 수 있도록 하였고 우리나라에 맞게 연도/월/요일 표기로 설정하였다 -->  <mx:DateChooser id="dc" backgroundColor="{cp.selectedColor}"  allowDisjointSelection="true" allowMultipleSelection="true"  yearSymbol="년"  dayNames="['일', '월', '화', '수', '목', '금', '토']"  monthNames="[1,2,3,4,5,6,7,8,9,10,11,12]" monthSymbol="월"  selectedDate="{**new** Date(2006, 8, 25)}"/>  </mx:Panel>  </mx:Application> |

이벤트

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

<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute"

**initialize**="**trace**(**'My Flex initialize!'**)" xmlns:ns1="\*" xmlns:comp="comp.\*"

**creationComplete**="Alert.show(**'welcom'**)"

**initialize : 처음실행시**

**creationComplete : 컴포넌트 생성된후**

|  |  |
| --- | --- |
| 이벤트 | 상황 |
| Click | 클릭햇을 때 |
| Change | 값이 변했을대. |
| <mx:Button x="258" y="57" label="굳장" **click**="Alert.show(**'뭐밍'**)" **buttonDown**="Alert.show(**'down'**)"/> | |

네임스페이스

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

<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute"

initialize="**trace**(**'My Flex initialize!'**)" xmlns:ns1="\*" **xmlns:ns12="comp.\*"**

creationComplete="Alert.show(**'welcom'**)"

>

<mx:Script>

<![CDATA[

**import** mx.controls.Alert;

……

]]>

</mx:Script>

<mx:Panel layout="absolute" left="10" bottom="10" right="10" top="10" title="My Flex" id="panel1">

**<ns12:KPanel x="92.5" y="104" width="294" height="292" click="showMsg('aa');">**

**</ns12:KPanel>**

</mx:Panel>

</mx:Application>

문법 get, set 함수를 변수처럼쓴다.

<mx:Script>

<![CDATA[

**import** mx.controls.Alert;

**private** **function** showTest(a:String ,b:int){

}

**private** **var** \_aaa:int = 10;

**private** **function** **get valll():**int{

**return** \_aaa;

}

**private** **function** **set valll(a:int):void**{

\_aaa = a;

}

**private** **function** showMsg(input:String):**void** {

*// mx.controls.Alert.show(input);*

*// currentState = 'image';*

Alert.show(String(valll));

valll = 20;

Alert.show(String(valll));

*// setCurrentState('image', true);*

}

]]>

</mx:Script>

//-----------------Restart

이벤트 등록.

**addEventListener(MouseEvent.ROLL\_OVER, rl);**

addEventListener(**"aaaaaaa"**,ddd);

이벤트발생

**private** **function** rl(e:Event):**void**{

**this**.width += 10;

**dispatchEvent(new Event("aaaaaaa"));**

*/\*c ++ ;*

*if(c == 10) aaa();\*/*

}

**private** **function** ddd(e:Event):**void**{

**this**.height+= 10;

}

**상태 변경**

<local:kbutton click="change(event); **currentState='good'"** >

</local:kbutton>

**상태 설정 및 레이아웃 설정**

<s:states>

<s:State name="wow"/>

<s:State name="good"/>

</s:states>

<s:layout>

<s:VerticalLayout verticalAlign="middle" horizontalAlign="center" />

</s:layout>

**외부 as파일 불러오기**

<fx:Script source=".." >

새로운 이벤트 등록

**package**

{

**import** flash.events.Event;

**import** flash.events.MouseEvent;

**import** mx.controls.Alert;

**import** spark.components.Button;

**[Event(name="AClick", type="flash.events.Event")]**

**public** **class** abutton **extends** Button

{

**public** **function** abutton()

{

**super**();

**this**.addEventListener(MouseEvent.ROLL\_OVER, rl);

*/\* this.addEventListener("aaaaaaa", function(e:Event):void{*

*this.height += 10;*

*}\*/*

**this**.addEventListener(**"aaaaaaa"**,ddd);

}

**private** **function** ddd(e:Event):**void**{

**this**.height+= 10;

}

**private** **var** c:int = 0;

**private** **function** rl(e:Event):**void**{

**this**.width += 10;

dispatchEvent(**new** Event(**"aaaaaaa"**));

*/\*c ++ ;*

*if(c == 10) aaa();\*/*

}

**private** **function** aaa():**void**{

dispatchEvent(**new** Event(**"aClick"**));

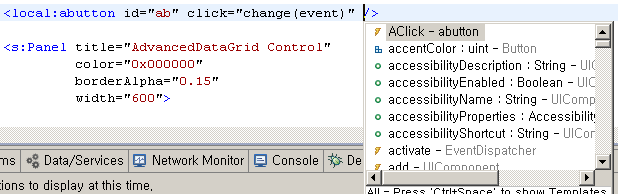
}

**private** **function** abc(e:Event):**void**{

Alert.show(**"11"**);

}

}

} 

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

<s:Button xmlns:fx="http://ns.adobe.com/mxml/2009"

xmlns:s="library://ns.adobe.com/flex/spark"

xmlns:mx="library://ns.adobe.com/flex/mx"

click="aa(event)"

>

<fx:Declarations>

<!-- Place non-visual elements (e.g., services, value objects) here -->

</fx:Declarations>

<fx:Script>

<![CDATA[

**import** mx.events.FlexEvent;

**private** **var** a:int = 0;

**private** **function** aa(e:MouseEvent):**void**{

a ++;

**this**.label = a+**""**;

}

]]>

</fx:Script>

</s:Button>

**시스템에서 발생하는 이벤트**

|  |  |
| --- | --- |
| 1. **Initialize** | **컴포넌트 초기 속성값 처리 마쳤을때** |
| 1. **creationComplete** | **컴포넌트 생성자를 호출하여 속성처리 크기 레이아웃 화면 그리기가 모드 끝났을때로 initialize이벤트보다 뒤에 발생** |
| 1. **hide** | **객체의 상태가 보이지 않는 상태로 바뀌었을때** |
| 1. **show** | **객체의 상태가 보이는 상태에서 보이지 않는 상태로 바뀌었을때** |