

Flex 3: фрагменти коду

Уклав [Виспянський Ігор](#), virua [at] ukr [dot] net.

Як додати власні елементи у контекстне меню?

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application name="Application_contextMenu"
  xmlns:mx="http://www.adobe.com/2006/mxml" initialize="init();">

  <mx:Script>
  <![CDATA[
    private function init():void {
      // перший елемент меню
      var customMenuItem1: ContextMenuItem =
        new ContextMenuItem("Назва 1", false, false);

      // ... ще один елемент меню
      var customMenuItem2: ContextMenuItem =
        new ContextMenuItem("Назва 2", false, false);

      var contextMenuCustomItems:Array =
        application.contextMenu.customItems;

      contextMenuCustomItems.push(customMenuItem1);
      contextMenuCustomItems.push(customMenuItem2);
    }
  ]]>
</mx:Script>

  <mx:Label text="Клацніть правою кнопкою миші, щоб побачити
    контекстне меню." />
</mx:Application>
```

Actionscript Cookie Util

Див. також: [Actionscript Cookie Util](#)

```
import de.aggro.utils.CookieUtil;

// задаємо ім'я для куки "mycookie" і значення "mycookie value"
// з "часом життя" 30 днів
CookieUtil.setCookie("mycookie", "mycookie value", 30);

// отримуємо куки і простежуємо його значення
trace(CookieUtil.getCookie("mycookie"));
```

```
// видаляємо куки з комп'ютера користувача
CookieUtil.deleteCookie("mycookie");
```

AS3: Як отримати куки (cookie)?

```
public var cookieStr:String;
public var cookieHeader:URLRequestHeader;
...
ExternalInterface.call('eval', 'window.cookieStr = function ()
{
    return document.cookie
};
');

cookieStr = ExternalInterface.call('cookieStr');
cookieHeader = new URLRequestHeader("Cookie", cookieStr);
```

Як використовувати showEffect і hideEffect?

```
<mx:Fade id="fadeOut" alphaFrom="1.0" alphaTo="0.0"/>
<mx:Fade id="fadeIn" alphaFrom="0.0" alphaTo="1.0"/>
...
<mx:Box id="innerBox" width="80%" height="20"
    showEffect="{fadeIn}" hideEffect="{fadeOut}"/>
```

Налаштування ефектів за допомогою ActionScript

Наступний приклад показує, як ви можете задати ефекти для Image, використовуючи ActionScript для налаштування showEffect і hideEffect.

```
<mx:Fade id="fade" />
<mx:Rotate id="rotate" angleFrom="-180" angleTo="0" />
<mx:Image id="img" source="@Embed('flash_icon.jpg')" />
...
<mx:Script>
<![CDATA[
    private function init():void {
        img.setStyle("showEffect", rotate);
        img.setStyle("hideEffect", fade);
    }
]]>
</mx:Script>
```

Приклад BindingUtils

```
public var myArrayCollection:ArrayCollection = new ArrayCollection();

<mx:SomeComponent id="myComponent"
    dataProvider={myArrayCollection} ... />
```

```
BindingUtils.bindProperty(myComponent, "dataProvider",
    this, "myArrayCollection");
```

Створення курсору у вигляді руки для компонента Flex

```
var label:Label = new Label();
label.buttonMode = true;
label.useHandCursor = true;
label.mouseChildren = false;
```

Свій Preloader (або як змінити надпис при завантаженні додатку)

Див. також: [Showing the download progress of an application](#)

Прописуємо наш preloader:

```
<mx:Application ... preloader="Preloader">
```

Вміст файлу **Preloader.as**:

```
package
{
    import flash.geom.Rectangle;
    import mx.graphics.RoundedRectangle;
    import mx.preloaders.DownloadProgressBar;

    public class Preloader extends DownloadProgressBar
    {
        public function Preloader()
        {
            super();
            // Set the downloading and initializing label text
            // to a blank string.
            downloadingLabel = "Завантаження...";
            initializingLabel = "Ініціалізація...";
        }

        // Override the function that gets the border rectangle so that
        // the area dedicated to the text label is removed.
        override protected function get borderRect():RoundedRectangle
        {
            // flex default settings: RoundedRectangle(0, 0, 182, 60,
            4);
            return new RoundedRectangle(0, 20, 182, 40, 4);
        }
    }
}
```

FileReference: дозволити завантажувати лише файли певного типу

```

var filters:Array = [];
filters.push(new FileFilter("Малюнки", "*.jpg;*.jpeg;*.gif;*.png"));

fileRef = new FileReference();
fileRef.addEventListener(Event.SELECT, fileRef_select);
fileRef.browse(filters);

```

Як розгорнути Flex 3 додаток на весь екран (full-screen)?

Кроки для створення перемикача повний/нормальний екран:

1) Прописуємо в тілі теґу <mx:Application>:

```

<mx:Script>
    <![CDATA[
        import flash.display.StageDisplayState;

        private function toggleFullScreen():void{

            if(fullScrnChecker.selected){
                stage.displayState =
                    StageDisplayState.FULL_SCREEN;
            } else {
                stage.displayState = StageDisplayState.NORMAL;
            }
        }
    ]]>
</mx:Script>

```

2) Задаємо перемикач:

```

<mx:CheckBox id="fullScrnChecker" label="Розгорнути на весь екран"
    click="toggleFullScreen()" />

```

3) Залишилось в файлі html-template/index.template.html додати властивість allowFullScreen в JavaScript код:

```

else if (hasRequestedVersion) {
    AC_FL_RunContent(
        ...
        "allowFullScreen","true",
        ...
    );
}

```

А також додати

```

<param name="allowFullScreen" value="true"/>

```

в теґ <object> і allowFullScreen="true" в теґ <embed> для сумісності.

Додавання іконки до кнопки

```
<mx:Button icon="@Embed('/bin/assets/images/icons/newnote.png')"  
    click="takeNotes()" tooltip="Take notes"/>
```

ActionScript: зміна іконок для кнопок при старті системи

```
// Set the icon of a Flex button in runtime  
  
// embed your icons  
[Embed(source='images/play.png')]  
[Bindable]  
public var imagePlay:Class;  
  
[Embed(source='images/pause.png')]  
[Bindable]  
public var imagePause:Class;  
  
// using one button to toggle play and pause of video  
private function playpause():void {  
    if (seismicVideo.playing){  
        seismicVideo.pause();  
        btn_play.setStyle("icon", imagePlay);  
    }else{  
        seismicVideo.play();  
        btn_play.setStyle("icon", imagePause);  
    }  
}
```

Passing Additional Parameters To Event Listener

```
myComponent.addEventListener(MouseEvent.CLICK,  
    function(e:MouseEvent):void {  
        var parm1:String = "First Parameter";  
        var parm2:String = "Second Parameter";  
        myClickHandler(e, parm1, parm2);  
    });  
  
function myClickHandler(e, str1, str2):void {  
    trace("Parameter 1 :" + str1);  
    trace("Parameter 2 :" + str2);  
}
```

Як в циклі пройтись по елементах <mx:Form>?

```
var theForm:Form = userInfoWindow["userInfoForm"];
```

```

for each (var Obj:Object in theForm.getChildren()){
    if (Obj is FormItem) {
        // Let's assume you only have a single child in one FormItem
        // and always one child for simplicity
        // (Obj as FormItem).getChildAt(0).text = 'xxx';

        Alert.show("" + (Obj as FormItem).getChildAt(0));

        // field = (Obj as FormItem).getChildAt(0) as TextInput;
        // field.text = 'xxx';
    }
}

```

Фонове зображення, що повторюється

```

.navPrimary {
    background-image: Embed("images/nav-background.png");
    background-size: "100%";
}

```

Як вивести на екран вміст ArrayCollection?

```

var names:ArrayCollection = new ArrayCollection (
    [
        "Ihor Vispyanskiy", "Vasiliy Beregivskiy", "Ivan
        Verchin", "Bogdan Detsik", "Andriy Doronin"
    ]
);

Alert.show( names.source.join(', ') );

```

Результат:

```

Ihor Vispyanskiy, Vasiliy Beregivskiy, Ivan Verchin, Bogdan Detsik,
Andriy Doronin

```

Як показати busy-курсор за допомогою CursorManager?

Див. також: [Making a Cooler Cursor in Flex](#)

```

import mx.managers.CursorManager;

// показуємо "зайнятий" курсор (busy cursor):
CursorManager.setBusyCursor();

// ховаємо "зайнятий" курсор:
CursorManager.removeBusyCursor();

```

Як змусити браузер не кешувати файли?

Ми хочемо кожного разу для свого Flex-додатку отримувати з сервера свіжу версію певного файлу. Але браузер кешує всі GET-запити і при повторному зверненні ми отримаємо файл не із сервера, а з кешу, тобто стару версію файлу.

Скажімо, ви завантажуєте XML-файл:

```
var service:HTTPService = new HTTPService();
service.url = 'http://сайт.com/xmlData.xml';
service.send();
```

Якщо в файл xmlData.xml були внесені зміни, то до тих пір, поки не буде очищено кеш браузера, їх не буде "видно".

Рішення просте.

Примітка. Ті, хто програмував на JavaScript і використовував AJAX підхід, має бути знайомий з даним трюком.

Підкоректуємо адресу в HTTPService:

```
service.url = 'http://сайт.com/xmlData.xml?r=' + Math.random();
```

Тобто до URL додаємо GET-параметр, який буде кожного разу іншим. Тим самим ми матимемо кожного разу іншу адресу, що дозволить завантажувати свіжі дані.

Як відомо, `Math.random()` поверне деяке випадкове число. Можна замість `Math.random()` використовувати `new Date().getMilliseconds()`.

Детальний опис помилки при її виникненні в HTTPService

```
private function httpService_fault(evt:FaultEvent):void {
    var title:String = evt.type + " (" + evt.fault.faultCode + ")";
    var text:String = evt.fault.faultString;
    alert = Alert.show(text, title);
    xmlListColl.removeAll();
}
```

Використання параметрів URL в HTTPService

```
<mx:Script>
    <![CDATA[
        var myURL:String="http://www.bpurcell.org/blog/rss.cfm";
        var mode:String="short";
        var mode2:String="cat";
        var catid="14";
    ]]>
</mx:Script>
```

```
<!-- If you set resultFormat to Object it turns the
```

```

returning data into an object -->

<mx:HTTPService url="{myURL}" id="myHTTPData"
    method="GET" resultFormat="object">
    <mx:request>
        <mode>{mode}</mode>
        <mode2>{mode2}</mode2>
        <catid>{catid}</catid>
    </mx:request>
</mx:HTTPService>

```

Передача параметрів

```

var http:HTTPService = new HTTPService();

// create parameters object
var params:Object = { abc: "123", def: "456" };

// call the HTTP Service's send() with a parameters object
http.send( params );

```

Доступ до даних, що знаходяться поза доменом (crossdomain.xml)

Щоб ви могли звертатися до файлу *.xml, який розташований на іншому сайті, вам слід створити міждоменний файл crossdomain.xml, що забезпечує необмежений доступ до ресурсів.

Взірці файлу crossdomain.xml:

- <http://greenlike.com/crossdomain.xml>
- <http://www.macromedia.com/crossdomain.xml>
- <http://www.yahoo.com/crossdomain.xml>

Removing folder icons from the Flex Tree control

```

<mx:Style>
Tree {
    folderClosedIcon: ClassReference(null);
    folderOpenIcon: ClassReference(null);
}
</mx:Style>

```

Removing leaf icons from the Flex Tree control

```

<mx:Style>
Tree {
    defaultLeafIcon: ClassReference(null);
}
</mx:Style>

```

Показуємо кількістю нащадків в кожній гілці компонента Tree

```
<mx:Tree id="tree" dataProvider="{dp}"
  labelFunction="tree_labelFunc" />

private function tree_labelFunc(item:Object):String {
  var suffix:String = "";
  if (tree.dataDescriptor.isBranch(item)) {
    suffix = " (" + item.children().length() + ")";
  }
  return item.@label + suffix;
}
```

"Зебра" для елементів компонента Tree

```
<mx:Tree id="mainMenuTree" ...
  alternatingItemColors="[#FFFFFF, #F3F3F3]" />
```

Посилання на членів батьківського компонента

Посилання на членів батьківського компонента може бути досягнуто за рахунок використання об'єкту **outerDocument** (зовнішній документ).

```
<mx:itemRenderer>
  <mx:Component>
    <mx:Image source="{data.showRelevance + '.png'}"
      tooltip="{(data.showRelevance == 1) ? 'On' : 'Off'}"
      click="outerDocument.toggle()" />
  </mx:Component>
</mx:itemRenderer>
...
```

Виклик ActionScript 3 методу з Item Render

```
<mx:itemRenderer>
  <mx:Component>
    <mx:HBox width="100%">
      <mx:ColorPicker
        selColor="{parentDocument.determineColour(data.colour)}" />
    </mx:HBox>
  </mx:Component>
</mx:itemRenderer>
...
```

Стаття: [Calling an ActionScript3 method from a Item Render in Flex.](#)

Як програмно очистити DateField?

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
  layout="horizontal">

  <mx:Script>
    <![CDATA[
```

```

        private function clearDate():void {
            testField.selectedDate = null;
        }
    ]]>
</mx:Script>

<mx.DateField id="testField"/>
<mx.Button label="Clear" click="clearDate();"/>
</mx:Application>

```

"Обрізання" тексту в Label за допомогою truncateToFit

truncateToFit — властивість, яка відповідає за "обрізання" тексту, що не вліз.

```

<mx:Label text="The quick brown fox jumped over the lazy dog."
    truncateToFit="true"
    maxWidth="200" />

```

mx:DataGrid wordWrap Wrangling (перенесення слів)

```

<mx:DataGrid id="reservationsDataGrid" width="100%" height="100%"
    rowCount="5" variableRowHeight="true" wordWrap="true"
    dataProvider="{reservations}">

```

Flex htmlText і Word Wrap

```

<mx:Text condenseWhite="true" width="100%">
    <mx:htmlText>
        <![CDATA[ This is some<br> text with simple <b>Html Formatting</
            b>. And this is a really reallyreally long line of
            text which we would want to auto wrap nicely.]]>
    </mx:htmlText>
</Text>

```

Як змінити іконку календаря в DateField?

```

<mx:Style>
    DateField {
        // картинка
        skin: Embed("clock_add.png");
        /* обнулити дефолтні стилі default.css */
        upSkin: ClassReference(null);
        overSkin: ClassReference(null);
        downSkin: ClassReference(null);
        disabledSkin: ClassReference(null);
    }
</mx:Style>

<mx.DateField id="dateField" />

```

Округлення числа, використовуючи клас NumberFormatter

```
<mx:NumberFormatter id="numberFormatter"
    precision="2" rounding="up" />
<mx:Script>

<![CDATA[
    import mx.collections.ArrayCollection;
    import mx.formatters.NumberBaseRoundType;

    private function button_click(evt:MouseEvent):void {
        textInput.errorString = "";
        numberFormatter.format(textInput.text);
        if (numberFormatter.error) {
            textInput.errorString = numberFormatter.error;
        }

        arrColl = new ArrayCollection();

        numberFormatter.rounding = NumberBaseRoundType.UP;
        arrColl.addItem({type:numberFormatter.rounding,
            value:numberFormatter.format(textInput.text)});
    }
]]>
</mx:Script>
```

Як використовувати SWFAddress у Flex-додатку?

1. Підключаємо .js-файл:

```
<script src="/js/swfaddress.js" type="text/javascript" />
```

2. Імпортуємо:

```
import SWFAddress;
import SWFAddressEvent;
```

3. Опрацьовуємо зміни адресного рядка:

```
SWFAddress.onChange = function () {
    // виводимо поточний URL
    trace (SWFAddress.getValue());
}
```

4. В потрібному місці (наприклад, при кліку по кнопці) змінюємо адресний рядок браузера і, якщо це потрібно, заголовок вікна:

```
// адреса: http://ваш-сайт.com.ua/#/guest_book/
SWFAddress.setValue("guest_book");
```

```
// заголовок вікна браузера  
SWFAddress.setTitle("Гостьова книга");
```