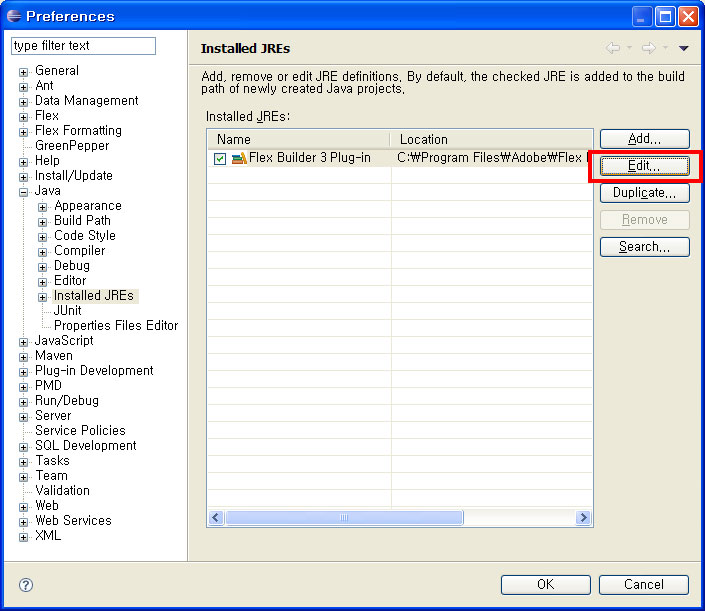
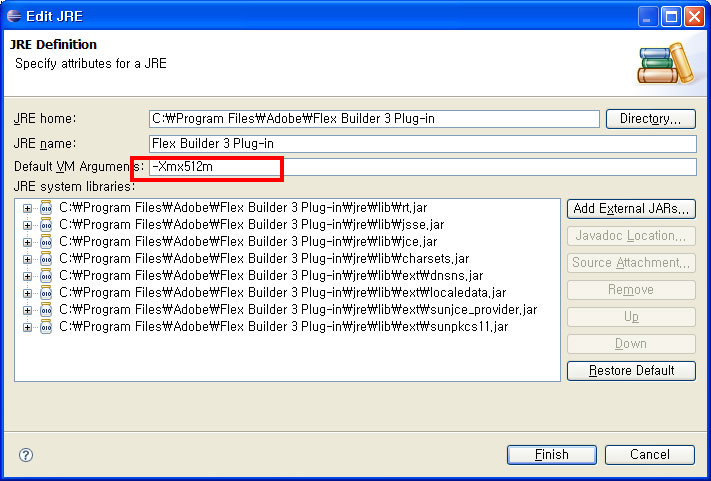
플랙스팁

[**FLEX로 개발하고 ANT빌드할 때 Out of Memory가 발생한다면..**](http://hans.thoth.kr/4381259)**Flex**[EDIT](http://hans.thoth.kr/?mid=blog&document_srl=4381259&act=dispTextyleToolPostManageWrite)

by **Choi Han Sik**, 2010.07.28 15:28, [0 Comments](http://hans.thoth.kr/?mid=blog&document_srl=4381259#comment), [0 Trackbacks](http://hans.thoth.kr/?mid=blog&document_srl=4381259#trackback)

출처 : <http://lovedev.tistory.com/475>  
  
가끔 플렉스를 ant로 컴파일 하다보면..   
  
Error: Java heap space java.lang.OutOfMemory  
Error: Java heap space  
  
에러가 발생하면서.. 컴파일이 안되는 경우가 있죠..  
며칠 전  직장동료가 이걸로 고생을 하더군요.  
  
그 친구의 해결책은 mxmlc에 fork="true"를 설정하고, memorySize를 잡아 주는 방식이었는데...그 때 그냥 지나친 문제가..  
드뎌 제게 닥치고 말았습니다. build.xml에 fork같은 구문을 넣기는 싫고, 로컬과 서버 환경에 같은 build.xml이 존재했으면 해서  
다음과 같은 조치를 했습니다.  
  
그 때 잘 안살펴봤어.. 승길아 미안해~ ^^; 용서해~

[](http://cfile5.uf.tistory.com/original/203BFF1B4A6076C729DD84)

[](http://cfile2.uf.tistory.com/original/121BC81B4A6076C75A6DE7)

**jre 즉 jvm이 실행될 때  -Xmx512m 파라미터를 전달해서 메모리를 올리시면 됩니다.**  
  
512는 512M겠죠.. **-Xmx256m**이면 256M가 되는 겁니다.. ^

이벤트 발생 디스페치이벤트

바인더블 함수로.

<?xml version="1.0"?>

<!-- binding/ASFunction.mxml -->

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

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

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

<s:layout>

<s:VerticalLayout/>

</s:layout>

<fx:Script>

<![CDATA[

import flash.events.Event;

// Define a function that gets invoked

// in response to the myFlagChanged event.

[Bindable(event="myFlagChanged")]

private function isEnabled():String {

if (myFlag)

return 'true';

else

return 'false';

}

private var \_myFlag:Boolean = false;

// Define a setter method that dispatches the

// myFlagChanged event to trigger the data binding.

public function set myFlag(value:Boolean):void {

\_myFlag = value;

dispatchEvent(new Event("myFlagChanged"));

}

public function get myFlag():Boolean {

return \_myFlag;

}

]]>

</fx:Script>

<!-- Use the function as the source of a data binding expression. -->

<s:TextArea id="myTA" text="{isEnabled()}"/>

<!-- Modify the property, causing the setter method to

dispatch the myFlagChanged event to trigger data binding. -->

<s:Button label="Clear MyFlag" click="myFlag=false;"/>

<s:Button label="Set MyFlag" click="myFlag=true;"/>

</s:Application>

The executing SWF file for the previous example is shown below:

<http://help.adobe.com/en_US/flex/using/WS2db454920e96a9e51e63e3d11c0bf64c3d-7ff3.html>

<http://help.adobe.com/en_US/flex/using/WS2db454920e96a9e51e63e3d11c0bf69084-7cc5.html>

# **Data binding in Flex**

by [PETER DEHAAN](http://blog.flexexamples.com/author/admin/) on OCTOBER 1, 2007 · [17 COMMENTS](http://blog.flexexamples.com/2007/10/01/data-binding-in-flex/#comments)

The following examples show a few different ways to bind data in Flex.

Full code after the jump.

The following example shows how you can use the <mx:Binding /> tag to bind values between two controls:

[View MXML](http://blog.flexexamples.com/wp-content/uploads/Binding_test/bin/srcview/source/main.mxml.html)

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

<!-- http://blog.flexexamples.com/2007/10/01/data-binding-in-flex/ -->

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

layout="vertical"

verticalAlign="middle"

backgroundColor="white">

<mx:Binding source="textInputSrc.text"

destination="textInputDst.text" />

<mx:Form>

<mx:FormItem label="source:">

<mx:TextInput id="textInputSrc" />

</mx:FormItem>

<mx:FormItem label="destination:">

<mx:TextInput id="textInputDst"

width="{textInputSrc.width}" />

</mx:FormItem>

</mx:Form>

</mx:Application>

[View source](http://blog.flexexamples.com/wp-content/uploads/Binding_test/bin/srcview/index.html) is enabled in the following example.

The following example shows how you can use the static BindingUtils.bindProperty()method to bind values between two controls:

[View MXML](http://blog.flexexamples.com/wp-content/uploads/Binding_test/bin/srcview/source/main2.mxml.html)

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

<!-- http://blog.flexexamples.com/2007/10/01/data-binding-in-flex/ -->

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

layout="vertical"

verticalAlign="middle"

backgroundColor="white"

creationComplete="init();">

<mx:Script>

<![CDATA[

import mx.binding.utils.BindingUtils;

private function init():void {

BindingUtils.bindProperty(textInputDst, "text", textInputSrc, "text");

}

]]>

</mx:Script>

<mx:Form>

<mx:FormItem label="source:">

<mx:TextInput id="textInputSrc" />

</mx:FormItem>

<mx:FormItem label="destination:">

<mx:TextInput id="textInputDst"

width="{textInputSrc.width}" />

</mx:FormItem>

</mx:Form>

</mx:Application>

[View source](http://blog.flexexamples.com/wp-content/uploads/Binding_test/bin/srcview/index.html) is enabled in the following example.

나인패스.

이미지 동적 늘리고 줄이기

Button

{

color:#FFF;

disabledColor: #777777;

textRollOverColor:#FFF;

text-selected-color:#A42E00;

/\* textSelectedColor: #FFBD2F; \*/

font-size:12;

font-weight:normal;

paddingTop:2;

upSkin: Embed(source="images/Button\_upSkin1.png",scaleGridLeft="4",scaleGridTop="4",scaleGridRight="67",scaleGridBottom="21");

overSkin: Embed(source="images/Button\_overSkin.png",scaleGridLeft="4",scaleGridTop="4",scaleGridRight="73",scaleGridBottom="27");

downSkin: Embed(source="images/Button\_downSkin.png",scaleGridLeft="4",scaleGridTop="4",scaleGridRight="73",scaleGridBottom="27");

disabledSkin: Embed(source="images/Button\_disabledSkin.png",scaleGridLeft="4",scaleGridTop="4",scaleGridRight="73",scaleGridBottom="27");

selectedUpSkin: Embed(source="images/Button\_downSkin.png",scaleGridLeft="4",scaleGridTop="4",scaleGridRight="73",scaleGridBottom="27");

selectedOverSkin: Embed(source="images/Button\_overSkin.png",scaleGridLeft="4",scaleGridTop="4",scaleGridRight="73",scaleGridBottom="27");

selectedDownSkin: Embed(source="images/Button\_upSkin1.png",scaleGridLeft="4",scaleGridTop="4",scaleGridRight="67",scaleGridBottom="21");

/\* disabledSkin: Embed(source="images/Button\_disabledSkin.png",

scaleGridLeft="4", scaleGridTop="4", scaleGridRight="63", scaleGridBottom="17");

downSkin: Embed(source="images/Button\_downSkin.png",

scaleGridLeft="4", scaleGridTop="4", scaleGridRight="63", scaleGridBottom="17");

overSkin: Embed(source="images/Button\_overSkin.png",

scaleGridLeft="4", scaleGridTop="4", scaleGridRight="63", scaleGridBottom="17");

upSkin: Embed(source="images/Button\_upSkin.png",

scaleGridLeft="4", scaleGridTop="4", scaleGridRight="63", scaleGridBottom="17");

color: #320202;

textRollOverColor: #FFFFFF;

textSelectedColor: #FD03D1;

selectedUpSkin: Embed(source="images/Button\_overSkin.png",

scaleGridLeft="4", scaleGridTop="4", scaleGridRight="63", scaleGridBottom="17");

selectedOverSkin: Embed(source="images/Button\_overSkin.png",

scaleGridLeft="4", scaleGridTop="4", scaleGridRight="63", scaleGridBottom="17");

selectedDisabledSkin: Embed(source="images/Button\_disabledSkin.png",

scaleGridLeft="4", scaleGridTop="4", scaleGridRight="63", scaleGridBottom="17");

selectedDownSkin: Embed(source="images/Button\_upSkin.png",

scaleGridLeft="4", scaleGridTop="4", scaleGridRight="63", scaleGridBottom="17");

disabledColor: #6D7070;

font-weight: normal;

font-size: 12;

\*/

}

### <http://cookbooks.adobe.com/post_Detecting_the_network_connection_type_with_Flex_4_-18791.html>

### [Detecting the network connection type with Flex 4.5](http://www.multimediacollege.be/2011/04/detecting-the-network-connection-type-with-flex-4-5/)

[< back to overview](http://www.multimediacollege.be/blog)

As you may already be aware of, Flex 4.5 is the next version of Flex that is also optimised for mobile development. That means that you can create compelling Rich Internet Applications with the Flex framework and deploy them on mobile Android devices, as well as the BlackBerry PlayBook (and iOS devices in the near future).

**One of the great features of mobile RIAs is the fact that you can connect your application to all kinds of different server technologies, such as PHP, Java, ColdFusion, .NET, … in exactly the same way as you would in your desktop AIR application.** You can even work with the DataService component to use real-time data connections over the RTMP channel. Just think about real-time stock updates, video streaming, trader applications on mobile, collaboration across multiple devices … The sky really is the limit. Or is it?

I think all of this always sounds very nice, but **people tend to forget one important thing when working on a mobile device and that is that if you’re not on a WiFi connection, your data contract can become very expensive when you use the sever connections**. However, in Flex 4.5 there is a way to actually check whether you are working on your data contract or on a WiFi connection.

In the application **you can access all of the device interfaces and check to see whether they are active or not**. The only thing you need to know is which interface to look for. In the code below, you can see how I check for the “WiFi” and “mobile” interfaces. Based on what I find, I just check whether it is active. Remember, some people may have the possibility to work via a data contract, but didn’t sign up for it, so finding the “mobile” interface isn’t enough.

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

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

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

title="Connection Test"

creationComplete="initView()">

<fx:Script>

<![CDATA[

private function initView():void {

var interfaces:Vector.<NetworkInterface> = NetworkInfo.networkInfo.findInterfaces();

for(var i:uint = 0; i < interfaces.length; i++) {

if(interfaces[i].name.toLowerCase() == "wifi" && interfaces[i].active) {

lbl.text = "WiFi connection enabled";

break;

} else if(interfaces[i].name.toLowerCase() == "mobile" && interfaces[i].active) {

lbl.text = "Mobile data connection enabled";

break;

}

}

}

]]>

</fx:Script>

<s:Label id="lbl" horizontalCenter="0" verticalCenter="0"/>

</s:View>

In this example I’m simply setting the text property of a label, but you can also use this in a real world application to determine the update rate for your server data, for example.

There is one thing you should not forget to do when creating such an application. **You have to set the proper permissions in the <applicationName>-app.xml file, in the android section.** That section should contain the ACCESS\_NETWORK\_STATE and theACCESS\_WIFI\_STATE permissions in order for it to work. If you don’t include this, you will get an empty Vector returned from the findInterfacesmethod.

**<android>**

**<manifestAdditions>**<![CDATA[

<manifest>

<!-- See the Adobe AIR documentation for more information about setting Google Android permissions -->

<uses-permission android:name="android.permission.INTERNET"/>

<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE"/>

<uses-permission android:name="android.permission.ACCESS\_WIFI\_STATE"/>

</manifest>

]]>**</manifestAdditions>**

**</android>**

## [Adobe AIR 2.0 Overview #1 - 개발환경 구축방법 및 새로운 기능 소개](http://blog.jidolstar.com/629)

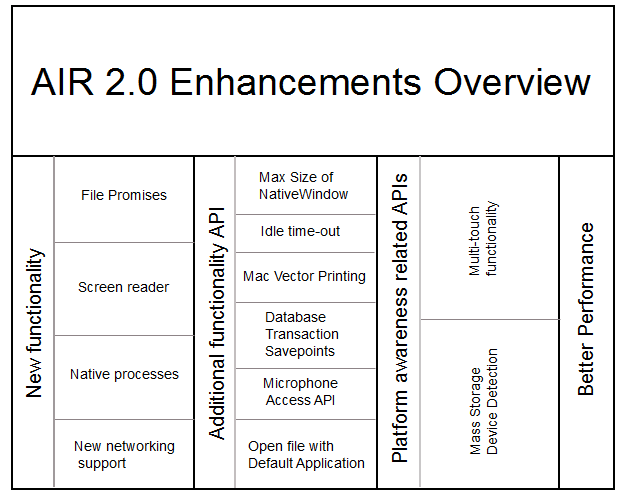
2009/12/08 18:14

[9 Comments](http://blog.jidolstar.com/629#commentlist)

[](http://cfile4.uf.tistory.com/original/160994164B1E184D0E51B1)

지난 2009년 11월 어도비(Adobe)에서는 AIR 2.0 Beta를 새롭게 발표하면서 퍼포먼스 향상과 더 많은 OS의 자원을 사용할 수 있는 API 기능을 추가해서 발표했다. Flash Player 10.1 Prerelease와 함께 발표된 Adobe AIR 2.0의 새로운 기능 및 추가/개선 사항에 대해서 정리해보았다.   
  
참고로 이 글은 Elad Elrom의 [AIR 2 Enhancements Complete Overview](http://www.insideria.com/2009/10/air-2-enhancements-complete-ov.html) 와 [AIR 2 Rerelease Note](http://labs.adobe.com/wiki/index.php/AIR_2:Release_Notes)를 참고해서 정리한 것이다. 

1. **새로운 기능**(이번 페이지에서 다룸)  
   - File Promises  
   - Screen reader  
   - Native Processes  
   - New networking support  
   - Global Error Handling  
   - Packaging an AIR application in a native installer
2. **기존 API에 대한 추가된 기능**(다음 편에 다룸)  
   - Max Size of NativeWindow  
   - Idle time-out  
   - Mac Vector Printing  
   - Database Transaction Savepoints  
   - Microphone Access API  
   - Open file with Default Application
3. **플랫폼 인식 관련 API** (다음 편에 다룸)  
   - Multi-touch functionality  
   - Mass Storage Device Detection
4. **퍼포먼스 향상**(다음 편에 다룸)

[](http://cfile7.uf.tistory.com/original/17012E264B1CB4870796A9)

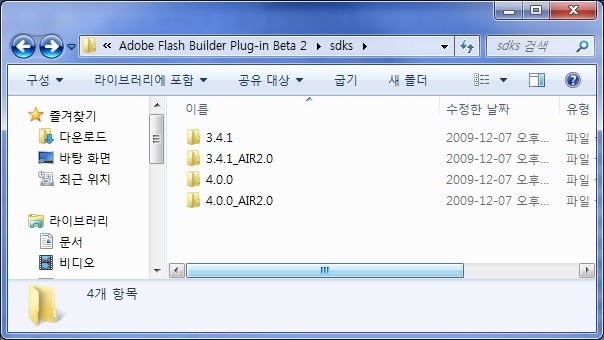
출처 : http://tinyurl.com/ydxevnu

**Flash Builder 4 beta 2에서 Adobe AIR 2.0로 개발 환경 만들기**  
여기에서 보여지는 예제는 모두 Flash Builder 4 beta 2에 기본으로 설치된 Flex 4 SDK를 이용한다. 하지만 AIR 2.0로 테스트해보기 위해서는 AIR 2.0 SDK를 따로 받아 다음과 같은 과정이 필요하다.   
  
**Windows 환경**이라면 다음과 같이 하면 되겠다. 

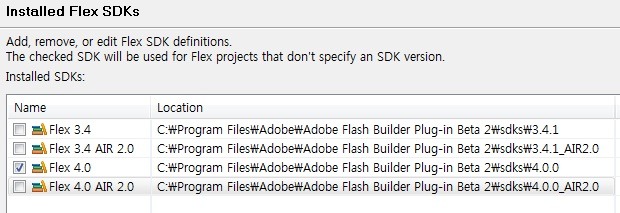
 만약 Flash Builder를 설치 안했다면 다음 링크를 통해 받는다.

<http://www.adoberia.co.kr/pds/down.html?src=text&kw=000026>

1. [AIR 2.0 SDK와 Runtime을 다운로드](http://labs.adobe.com/downloads/air2.html) 받는다.
2. 다운받은 Runtime을 실행해 설치한다.
3. Flash builder가 설치된 sdks/4.0.0과 sdks/3.4.1 폴더를 복사해서 sdks 폴더안에 붙인뒤 4.0.0\_AIR2.0, 3.4.1\_AIR2.0을 만든다. 이렇게 되면 sdks 폴더안에는 3.4.1, 4.0.0, 3.4.1\_AIR2.0, 4.0.0\_AIR2.0 이름을 가진 4개의 SDK가 만들어진다.  
   본인의 경우는 C:/Program Files/Adobe/Adobe Flash Builder Plug-in Beta 2/sdks 내에 있다.
4. AIR 2.0 SDK는 압축을 풀고 그안에 있는 내용을 방금 만든 4.0.0\_AIR2.0, 3.4.1\_AIR2.0 폴더에 복사한다.

[](http://cfile7.uf.tistory.com/original/1263D8154B1CB2772FEE17)

1. Flash Builder를 실행한다.
2. 메뉴에서 Window > Preference 로 들어간다.
3. 창이 뜨면 왼쪽 메뉴에서 Flash builder > Installed Flex SDKs를 선택한다.
4. 우측에 Add버튼을 눌러 위에서 새로 만든 AIR 2.0을 위한 SDKs들을 선택한다.    
   Flex SDK Name은 각각 Flex 3.4 AIR 2.0, Flex 4.0 AIR 2.0 등으로 이름을 바꿔도 된다.  
   본인의 경우 아래 경로가 되겠다.  
   C:/Program Files/Adobe/Adobe Flash Builder Plug-in Beta 2/sdks/3.4.1\_AIR2.0/  
   C:/Program Files/Adobe/Adobe Flash Builder Plug-in Beta 2/sdks/4.0.0\_AIR2.0/  
   아래처럼 Flex 4.0으로 체크되어 있다면 AIR 2.0기반의 Flex 4.0 SDK를 디폴트로 바꾸자.

[](http://cfile2.uf.tistory.com/original/203BBA244B1CB39C99F080)

1. 이제 모든 개발환경이 완료되었다. 자신의 프로젝트 생성시에 원하는 SDK를 선택하면 되겠다.  물론 아래에서 소개하는 예시를 따르려면 Flex 3든 Flex 4든 AIR 2.0 SDK를 선택해야겠다.

**Mac 환경**이라면 다음과 같이 하면 되겠다.   
  
[Mac에서 AIR 2.0 Beta 개발환경 만들기 - Flash Builder 4 Beta 2 기반](http://blog.jidolstar.com/626)  
  
참고로 Flash CS4나 CS3, 드림위버나 기타 다른 IDE에서 개발하시는 분들에게는 본인의 경험이 없어 정보를 줄 수 없다.  
    
**1. Adobe AIR 2.0 새로운 기능**  
**1.1 IPv6 지원**   
AIR 2.0은 모든 Network API에 대해서 IPv6를 지원하게 되었다. 현재 인터넷에서 주로 사용하고 있는 IP 주소의 버전은 IPv4로서 무려 20년이 넘도록 사용하고 있다. IPv4는 32비트 주소체계를 가지며 4,294,967,296개의 제한된 갯수의 주소를 가진다. IPv6는 이 부족함을 해소하기 위해 만들어진 IP 주소 체계이다. IPv6는 128비트 주소체계를 가진다. IPv6가 일반적으로 사용되고 있는 것은 아니지만  가까운 미래에 IPv6가 범용적으로 도입되어 네트워크 IP 주소체계가 더욱 확장될 수 있으므로 미리 지원될 수 있도록 한 것으로 여겨진다. IPv6에 대한 자세한 내용은 아래 링크를 참고한다.

* [(Adobe 문서)IPv6 addresses](http://tinyurl.com/ylzjgup)
* <http://ko.wikipedia.org/wiki/IPv6>

**1.2  UDP(User Datagram Protocol) 소켓, Server 소켓, TLS/SSL 소켓 지원**AIR 2.0에서는 더욱 다양한 소켓 통신 관련 클래스들이 추가되었다. 이전버전까지 반쪽짜리 였다면 이제 어느정도 구색을 갖춘듯 하다.   
  
AIR 2.0에서 지원하는 네트워크에 관련된 내용은 다음 링크를 참고한다.

* [Networking](http://tinyurl.com/y8fq3a3)

**서버 소켓 : ServerSocket 클래스**기존 AIR 버전에서는 반쪽 소켓통신만 지원되었다. 즉, 클라이언트는 만들 수 있었지만 서버측을 만들지 못했다. 이번 AIR 2.0에서 서버 소켓이 지원됨에 따라 이를 이용한 다양한 네트워크 애플리케이션을 만들 수 있을 것으로 보인다.   
  
서버 소켓은 TCP(Transmissin Control Protocal)을 기반으로 한다. TCP는 연결지향 통신 규약으로 인터넷 상의 컴퓨터들 사이에 데이터를 통신하기 위해 IP와 함께 사용된다. TCP는 데이터의 통신 중간에 데이터가 유실되더라도 신뢰도를  체크해서 완벽한 데이터를 통신할 수 있는 신뢰형 기반의 데이터 통신시 사용된다. 일반적인 웹브라우져를 이용한 통신데이터는 모두 TCP를 근간으로 한다.

* [TCP에 대해](http://tinyurl.com/ybngwtz)

AIR 2.0에서 지원하는 서버 Socket는 ServerSocket 클래스에서 관장한다.

* [(Adobe 문서)Server sockets](http://tinyurl.com/y9o6f56)
* [ServerSocket 클래스](http://tinyurl.com/ydc7d2z)
* [머드초보님의 ServerSocket을 이용한 간단한 채팅서버 만들기](http://mudchobo.tomeii.com/tt/454)
* [Creating a socket server in Adobe AIR 2](http://tinyurl.com/ye4wgr4)
* [A Demonstration of the ServerSocket API in AIR 2](http://blogs.adobe.com/cantrell/archives/2009/11/server_socket_demo_in_air_2.html)

**UDP 소켓 : DatagramSocket 클래스**  
UDP는 User Datagram Protocal의 약어로 IP를 사용해 네트워크내에서 컴퓨터들간에 데이타 통신하기 위해 제한된 서비스만 제공하는 통신 규칙이다. TCP의 대안이기도 하며 IP와 함께 사용하면 UDP/IP라고 표현하기도 한다. UDP는 정확한 데이터를 전송할 의무가 주어지지 않은 비연결지향성을 가져 데이터의 흐름에 신뢰를 할 수 없는 것이 특징이다. TCP처럼 통신을 하기전 상대방을 확인하는 절차가 없기 때문에 데이터 정확도는 떨어질 수 있지만 빠른 데이타 전송에 유리하다. 그래서 음악 및 비디오 실시간 스트리밍 서비스에 잘 사용하는 프로토콜이다.

* [UDP 소켓 프로그래밍](http://tinyurl.com/yjxbcz5)

AIR 2.0에서는 UDP 통신을 DatagramSocket 클래스와 DatagramSocketDataEvent 클래스를 지원하고 있다. 더욱 자세한 내용은 아래 링크를 참고한다.

* [(Adobe 문서)UDP sockets (AIR)](http://tinyurl.com/y9b2ana)
* [DatagramSocket 클래스](http://tinyurl.com/yduygdo)

**Secure 클라이언트 소켓 : SecureSocket 클래스**  
소켓서버에 접속하기 위해 SSLv4(Secure Socket Layer version 4) 또는 TLSv1(Transport Layer Security version 1) 기반으로 하는 프로토콜을 이용하여 SecureSocket 클래스로 접속할 수 있다. Secure 소켓을 이용하면 암호화된 데이터 전송으로 서버인증(server authentication), 데이터 무결성(data integrity), 메시지 기밀성(message confidentiality)등의 장점을 가진다.

* [(Adobe 문서) Secure client sockets(AIR)](http://tinyurl.com/ygazvly)
* [SecureSocket 클래스](http://tinyurl.com/ykawcsr)
* [A Demonstration of Encrypted Socket Support in AIR 2](http://blogs.adobe.com/cantrell/archives/2009/11/a_demonstration_of_encrypted_sockets.html)

**1.3 네트워크 정보**  
AIR 2.0의 API로 네트워크 정보를 얻어낼 수 있는 클래스인 NetworkInfo가 추가되었다. 이 클래스를 이용하면 사용자의 기기의 Mac Address, IP정보 등의 네트워크 정보를 얻어올 수 있다.

* [(Adobe 문서)Network Interfaces](http://tinyurl.com/yzjf8u7)
* [NetworkInfo 클래스](http://tinyurl.com/yzpowq4)
* [NetworkInterface 클래스](http://tinyurl.com/yjw729s)
* [InterfaceAddress 클래스](http://tinyurl.com/yktrcx9)

Adobe 문서에서 보여지는 예제에서는 var networkInfo:NetworkInfo = new NetworkInfo() 처럼 생성해서 사용하도록 되어 있는데 잘못된 것이다. var networkInfo:NetworkInfo = NetworkInfo.networkInfo; 처럼 static 속성으로 접근해야한다.   
그리고 NetworkInterface에 networkInterfaceDefault 속성 또한 존재하지 않는다.

01.package {

02.import flash.display.Sprite;

03.import flash.net.InterfaceAddress;

04.import flash.net.NetworkInfo;

05.import flash.net.NetworkInterface;

06.

07.public class NetworkInformationExample extends Sprite

08.{

09.public function NetworkInformationExample()

10.{

11.var networkInfo:NetworkInfo.<NetworkInterface> = NetworkInfo.networkInfo;

12.var interfaces:Vector.<NetworkInterface> = networkInfo.findInterfaces();

13.

14.if( interfaces != null )

15.{

16.trace( "Interface count: " + interfaces.length );

17.for each ( var interfaceObj:NetworkInterface in interfaces )

18.{

19.trace( "\nname: "             + interfaceObj.name );

20.trace( "display name: "     + interfaceObj.displayName );

21.trace( "mtu: "                 + interfaceObj.mtu );

22.trace( "active?: "             + interfaceObj.active );

23.//trace( "default?: "         + interfaceObj.networkInterfaceDefault );

24.trace( "parent interface: " + interfaceObj.parent );

25.trace( "hardware address: " + interfaceObj.hardwareAddress );

26.if( interfaceObj.subInterfaces != null )

27.{

28.trace( "# subinterfaces: " + interfaceObj.subInterfaces.length );

29.}

30.trace("# addresses: "     + interfaceObj.addresses.length );

31.for each ( var address:InterfaceAddress ininterfaceObj.addresses )

32.{

33.trace( "  type: "           + address.ipVersion );

34.trace( "  address: "         + address.address );

35.trace( "  broadcast: "         + address.broadcast );

36.trace( "  prefix length: "     + address.prefixLength );

37.}

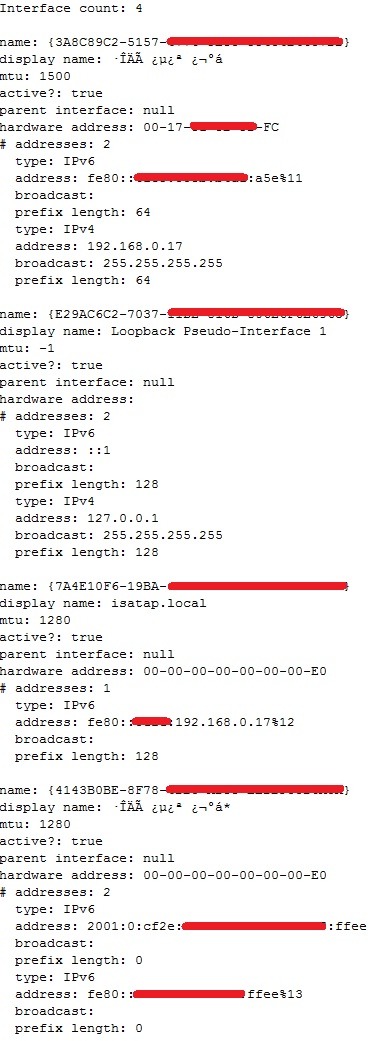
38.}

39.}

40.}

41.}

42.}

[](http://cfile6.uf.tistory.com/original/1327610D4B1CE99016ADA4)

위 프로그램은 NetworkInfo를 사용하는 아주 간단한 예시이다. Flash Builder 4 에서 이 예시를 테스트해볼때는 AIR 프로젝트 생성시 마지막에 mxml대신 as로 확장자를 두고 생성해서 테스트 하면 되겠다.  
  
  
**1.4 DNS lookup**  
AIR 2.0에서부터 DNSResolver 클래스를 제공하여 DNS(Domain Name System) 정보를 찾을 수 있게 되었다. 정보를 찾게되면 DNSResolverEvent가 발생한다.   
  
DNS는 각 도메인 이름(가령 examples.com)을 IP주소로 가리킨다. 일반적으로 사람은 111.123.431.11와 같은 IP주소보다 examples.com처럼 도메인 이름과 같은 단어조합을 더욱 인지하기 쉽다. 이러한 이유로 실제로는 도메인 이름을 사용하면 DNS를 걸쳐 이에 해당하는 IP주소로 연결시켜 관련 서버로 접속하게 한다. AIR 2.0에서 제공하는 DNSResolver 클래스를 이용하면 이러한 관계를 조사해볼 수 있게 된다.   
  
IP주소는 IPv4(32비트) 또는 IPv6(128비트)를 사용할 수 있다.   
  
DNSResolver 클래스의 lookup() 함수를 통해 host이름과 레코드 형태(record type)을 넘겨주면 DNSResolverEvent.LOOK\_UP 이벤트를 발생하여 관련 레코드 타입에 대한 정보를 반환해준다. 

* [DNSResolver 클래스](http://tinyurl.com/yjzpl59)
* [(레코드 타입)ARecord](http://tinyurl.com/ykatxu8)
* [(레코드 타입)AAARecord](http://tinyurl.com/yhtsdax)
* [(레코드 타입)SRVRecord](http://tinyurl.com/ykkpf4d)
* [(레코드 타입)MXRecord](http://tinyurl.com/yjwat9u)
* [(레코드 타입)PTRRecod](http://tinyurl.com/yzcnpvk)
* [(Adobe 문서)DNS 레코드](http://tinyurl.com/ykseg3g)

 다음은 예제이다.

01.package

02.{

03.import flash.desktop.NativeApplication;

04.import flash.display.Sprite;

05.import flash.events.DNSResolverEvent;

06.import flash.events.ErrorEvent;

07.import flash.net.dns.AAAARecord;

08.import flash.net.dns.ARecord;

09.import flash.net.dns.DNSResolver;

10.import flash.net.dns.MXRecord;

11.import flash.net.dns.PTRRecord;

12.import flash.net.dns.SRVRecord;

13.import flash.utils.getQualifiedClassName;

14.

15.public class DNSResolverExample extends Sprite

16.{

17.private var lookupCount:int = 0;

18.public function DNSResolverExample()

19.{

20.//Create the resolver object

21.var resolver:DNSResolver = new DNSResolver();

22.resolver.addEventListener( DNSResolverEvent.LOOKUP, lookupComplete );

23.resolver.addEventListener( ErrorEvent.ERROR, lookupError );

24.

25.//Look up records

26.resolver.lookup( "yahoo.com", ARecord );

27.resolver.lookup( "yahoo.com", AAAARecord );

28.resolver.lookup( "yahoo.com", MXRecord );

29.resolver.lookup( "209.191.93.53", PTRRecord );

30.resolver.lookup( "\_sip.\_tcp.yahoo.com.", SRVRecord );

31.}

32.

33.private function lookupComplete( event:DNSResolverEvent ):void

34.{

35.trace("-------------------------------------");

36.trace( "Query string: " + event.host );

37.trace( "Record type: " +  flash.utils.getQualifiedClassName( event.resourceRecords[0] ) +

38.", count: " + event.resourceRecords.length );

39.for each( var record:\* in event.resourceRecords )

40.{

41.if( record is ARecord ) trace( record.name + " : " + record.address );

42.if( record is AAAARecord ) trace( record.name + " : " + record.address );

43.if( record is MXRecord ) trace( record.name + " : " + record.exchange + ", " + record.preference );

44.if( record is PTRRecord ) trace( record.name + " : " + record.ptrdName );

45.if( record is SRVRecord ) trace( record.name + " : " + record.target + ", " + record.port +

46.", " + record.priority + ", " + record.weight );

47.}

48.if( ++lookupCount == 5 )

49.{

50.NativeApplication.nativeApplication.exit();

51.}

52.}

53.

54.private function lookupError( error:ErrorEvent ):void

55.{

56.trace("-------------------------------------");

57.trace("Error: " + error.text );

58.if( ++lookupCount == 5 )

59.{

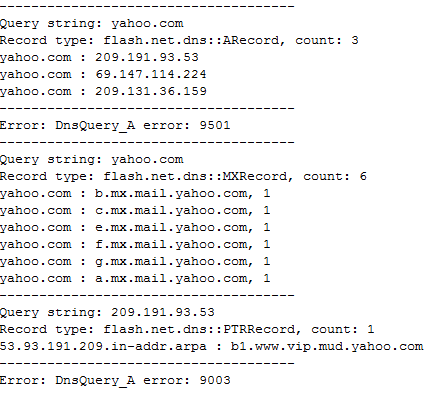
60.NativeApplication.nativeApplication.exit();

61.}

62.}

63.}

64.}

[](http://cfile7.uf.tistory.com/original/1805FF0C4B1CEE71AFBE1E)

**1.5 스크린 리더 지원(Screen Reader Support)**  
AIR 2.0은 스크린 리더를 지원하기 위한 Accessibility 클래스가 추가되었다. 스크린 리더는 시각 장애 사용자들을 위한 보조 기술로서 화면 내용을 오디오 버전으로 제공하는 것이다. 이 클래스는 현재 Windows 환경에서만 제공된다. 이것이 지원되는지 확인하려면 Capabilities.hasAccessibility 정적속성이 true이어야 한다. 버튼, 무비클립, 텍스트필드와 같은 객체에 액세스 가능한 속성을 얻고 설정하기 위해 DisplayObject.accessibilityProperties 속성을 사용하며 이 값이 바뀌었다는 것을 알려주기 위해 Accessibility.updateProperties() 정적메소드를 이용한다.

* [[IT용어 아하!] 스크린 리더](http://tinyurl.com/yffnd7o)
* [Accessibility 클래스](http://tinyurl.com/yjwj93f)
* [[열이아빠] RIA가 웹을 불필요한 제약으로 만들고 있나요 - 플렉스에도 접근성을 적용할 수 있나요. 참고](http://koko8829.tistory.com/503)

이 기능을 사용하기 위해 프로젝트의 컴파일 옵션을 추가해야한다. Flash builder 4에서 해당 프로젝트를 선택후 메뉴의 Project > Properties를 선택한다. 새창이 나오면 Flex Compiler를 선택후 컴파일 옵션에 Generate accessible SWF file을 선택후 OK 버튼을 누른다. 

001.package {

002.import flash.accessibility.Accessibility;

003.import flash.desktop.NativeApplication;

004.import flash.display.Sprite;

005.import flash.display.StageAlign;

006.import flash.display.StageScaleMode;

007.import flash.utils.setTimeout;

008.

009.public class AccessibilityExample extends Sprite {

010.public static const BUTTON\_WIDTH:uint = 90;

011.public static const BUTTON\_HEIGHT:uint = 20;

012.

013.private var gutter:uint = 5;

014.private var menuLabels:Array = new Array("PROJECTS", "PORTFOLIO","CONTACT");

015.private var menuDescriptions:Array = new Array("Learn more about our projects"

016., "See our portfolio"

017., "Get in touch with our team");

018.

019.public function AccessibilityExample() {

020.configureAssets();

021.NativeApplication.nativeApplication.autoExit = true;

022.stage.align = StageAlign.TOP\_LEFT;

023.stage.scaleMode = StageScaleMode.NO\_SCALE;

024.stage.nativeWindow.activate();

025.setTimeout(updateAccessibility, 2000);

026.}

027.

028.private function updateAccessibility():void {

029.trace("Accessibility.active: " + Accessibility.active);

030.if(Accessibility.active) {

031.Accessibility.updateProperties();

032.}

033.}

034.

035.private function configureAssets():void {

036.var child:CustomAccessibleButton;

037.for(var i:uint; i < menuLabels.length; i++) {

038.child = new CustomAccessibleButton();

039.child.y = (numChildren \* (BUTTON\_HEIGHT + gutter));

040.child.setLabel(menuLabels[i]);

041.child.setDescription(menuDescriptions[i]);

042.addChild(child);

043.}

044.}

045.}

046.}

047.

048.import flash.accessibility.AccessibilityProperties;

049.import flash.display.Shape;

050.import flash.display.SimpleButton;

051.import flash.display.Sprite;

052.import flash.events.Event;

053.import flash.text.TextFormat;

054.import flash.text.TextFormatAlign;

055.import flash.text.TextField;

056.

057.class CustomAccessibleButton extends Sprite {

058.private var button:SimpleButton;

059.private var label:TextField;

060.private var description:String;

061.private var \_name:String;

062.

063.public function CustomAccessibleButton(\_width:uint = 0, \_height:uint = 0) {

064.\_width = (\_width == 0) ? AccessibilityExample.BUTTON\_WIDTH : \_width;

065.\_height = (\_height == 0) ? AccessibilityExample.BUTTON\_HEIGHT : \_height;

066.

067.button = buildButton(\_width, \_height);

068.label = buildLabel(\_width, \_height);

069.

070.addEventListener(Event.ADDED, addedHandler);

071.}

072.

073.private function addedHandler(event:Event):void {

074.trace("addedHandler: " + this.\_name);

075.var accessProps:AccessibilityProperties = new AccessibilityProperties();

076.accessProps.name = this.\_name;

077.accessProps.description = description;

078.accessibilityProperties = accessProps;

079.removeEventListener(Event.ADDED, addedHandler);

080.}

081.

082.private function buildButton(\_width:uint, \_height:uint):SimpleButton {

083.var child:SimpleButton = new CustomSimpleButton(\_width, \_height);

084.addChild(child);

085.return child;

086.}

087.

088.private function buildLabel(\_width:uint, \_height:uint):TextField {

089.var format:TextFormat = new TextFormat();

090.format.font = "Verdana";

091.format.size = 11;

092.format.color = 0xFFFFFF;

093.format.align = TextFormatAlign.CENTER;

094.format.bold = true;

095.

096.var child:TextField = new TextField();

097.child.y = 1;

098.child.width = \_width;

099.child.height = \_height;

100.child.selectable = false;

101.child.defaultTextFormat = format;

102.child.mouseEnabled = false;

103.

104.addChild(child);

105.return child;

106.}

107.

108.public function setLabel(text:String):void {

109.label.text = text;

110.this.\_name = text;

111.}

112.

113.public function setDescription(text:String):void {

114.description = text;

115.}

116.}

117.

118.class CustomSimpleButton extends SimpleButton {

119.private var upColor:uint = 0xFFCC00;

120.private var overColor:uint = 0xCCFF00;

121.private var downColor:uint = 0x00CCFF;

122.

123.public function CustomSimpleButton(\_width:uint, \_height:uint) {

124.downState = new ButtonDisplayState(downColor, \_width, \_height);

125.overState = new ButtonDisplayState(overColor, \_width, \_height);

126.upState = new ButtonDisplayState(upColor, \_width, \_height);

127.hitTestState = new ButtonDisplayState(upColor, \_width, \_height);

128.useHandCursor = true;

129.}

130.}

131.

132.class ButtonDisplayState extends Shape {

133.private var bgColor:uint;

134.private var \_width:uint;

135.private var \_height:uint;

136.

137.public function ButtonDisplayState(bgColor:uint, \_width:uint, \_height:uint) {

138.this.bgColor = bgColor;

139.this.\_width = \_width;

140.this.\_height = \_height;

141.draw();

142.}

143.

144.private function draw():void {

145.graphics.beginFill(bgColor);

146.graphics.drawRect(0, 0, \_width, \_height);

147.graphics.endFill();

148.}

149.}

**1.6 네이티브 프로세스의 직접 실행과 상호 작용( Launching and Interacting with Native Processes )**  
네이티브 프로세스에 대해서 본인도 처음 접하는 용어라 해석하기 어려웠는데, 이해한바 각 운영체제에 실행될 수 있는 코드라고 생각하면 될 것 같다. 가령, 윈도우에서는 exe, dll, ocx등이겠고 맥이라면 dmg, app 등이 아닐까 생각한다.  
  
AIR 2.0부터 기존의 애플리케이션을 실행해 AIR 애플리케이션과 표준입출력을 통한 상호작용할 수 있는 기능이 추가되었다. 가령, C언어로 만든 프로그램을 자신의 AIR 애플리케이션과 함께 배포했다고 하자. AIR 애플리케이션이 이 C언어로 만든 프로그램을 실행하고 표준입출력을 이용해 통신이 가능해진다. 또는 이미 설치된 Editplus에 File을 넘겨서 실행하고 종료시점도 알 수 있다.   
  
자세한 내용은 다음을 참고한다.

* [(Adobe 문서) Communicating with native processes in AIR](http://tinyurl.com/yjyhx8k)
* [NativeProcess 클래스](http://tinyurl.com/yfo4jy2)
* [NativeProcessStartupInfo 클래스](http://tinyurl.com/yfmsktf)
* [A Demonstration of the NativeProcess APIs in AIR 2](http://blogs.adobe.com/cantrell/archives/2009/11/demo_of_nativeprocess_apis.html)
* [(찬익)Adobe AIR 2.0의 NativeProcess와 Native Installer](http://blog.chanik.com/45)

이 기능을 수행하려면 반드시 AIR 디스크립터 파일에 다음 코드를 꼭 넣어야 한다. 

1.<supportedProfiles>extendedDesktop</supportedProfiles>

아래는 Editplus를 이용해 Text파일을 열고 Editplus가 닫힐 때 감지하는 간단한 Flex 4를 이용한 AIR 2.0 애플리케이션 코드이다. NativeProcess와 NativeProcessStartupInfo 클래스의 용도를 잘 살펴보자.

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

02.<s:WindowedApplication xmlns:fx="<http://ns.adobe.com/mxml/2009>"

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

04.xmlns:mx="<library://ns.adobe.com/flex/halo>">

05.<fx:Script>

06.<![CDATA[

07.import flash.events.NativeProcessExitEvent;

08.

09.public function executeNativeProcess():void

10.{

11.var executable:File = new File("C:/Program Files/EditPlus 3/editplus.exe"); //c:/Windows/System32/notepad.exe

12.var workingDirectory:File = new File("c:/");

13.

14.var nativeProcess:NativeProcess = new NativeProcess();

15.

16.if (NativeProcess.isSupported)

17.{

18.trace("Native Process Supported");

19.}

20.

21.var nativeProcessStartupInfo:NativeProcessStartupInfo = new NativeProcessStartupInfo();

22.nativeProcessStartupInfo.executable = executable;

23.nativeProcessStartupInfo.workingDirectory = workingDirectory;

24.

25.var args:Vector.<String> = new Vector.<String>();

26.args.push("c:/LAN.log"); // open file that was given with the executable application

27.nativeProcessStartupInfo.arguments = args;

28.

29.nativeProcess.addEventListener( NativeProcessExitEvent.EXIT, onExitError );

30.

31.try {

32.nativeProcess.start(nativeProcessStartupInfo);

33.} catch (error:IllegalOperationError) {

34.trace("Illegal Operation: "+error.toString());

35.} catch (error:ArgumentError) {

36.trace("Argument Error: "+error.toString());

37.} catch (error:Error) {

38.trace ("Error: "+error.toString());

39.}

40.

41.if (nativeProcess.running)

42.{

43.trace ("Native Process Support");

44.}

45.}

46.

47.public function onExitError(event:NativeProcessExitEvent):void

48.{

49.trace( "Native Process Exit code: "+event.exitCode );

50.}

51.

52.]]>

53.</fx:Script>

54.

55.<s:Button id="button"

56.label="Open File foobar.txt with text editor"

57.click="executeNativeProcess();"

58.width="250" />

59.

60.</s:WindowedApplication>

Mac인 경우 Mac에 설치되어 기본으로 설치되어 있는 텍스트 에디터를 실행할 수도 있다. 이것도 별다른 설정없이 같은 방법으로 실행할 수 있다. new File()에는 "/Applications/TextEdit/TextEdit.app/Contents/MacOS/TextEdit"와 "/" 로 대체하고 args.push()부분은 foobar.txt가 데스크톱에 있을때 "/User/사용자계정/Desktop/foobar.txt"로 대체해서 실행하면 된다.   
  
NativeProcess 클래스는 단순히 실행과 종료만 담당하지 않는다. AIR 애플리케이션과 Native 실행 소스간에 통신도 가능하다. 다음 링크는 HTML/Javascript, Flex, Flash 개발자들을 위한 예제이다. AIR코드 뿐 아니라 C코드도 있으니 꼭 다운받아 실행해보길 바란다.

* [Interacting with a native process(HTML/Javascript)](http://www.adobe.com/devnet/air/ajax/quickstart/interacting_with_native_process.html)
* [Interacting with a native process(Flex 3)](http://www.adobe.com/devnet/air/flex/quickstart/interacting_with_native_process.html)
* [Interacting with a native process(Flash CS5)](http://www.adobe.com/devnet/air/flash/quickstart/interacting_with_native_process.html)

다음과 같이 Flex 4 코드로 변경해봤다.

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

02.<s:WindowedApplication xmlns:fx="<http://ns.adobe.com/mxml/2009>"

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

04.xmlns:mx="<library://ns.adobe.com/flex/halo>"

05.width="350" height="150"

06.applicationComplete="windowedapplication1\_applicationCompleteHandler(event)">

07.<fx:Script>

08.<![CDATA[

09.import flash.events.ProgressEvent;

10.import flash.system.Capabilities;

11.import flash.utils.IDataInput;

12.

13.import mx.controls.Alert;

14.import mx.events.FlexEvent;

15.

16.private var process:NativeProcess;

17.

18.private function windowedapplication1\_applicationCompleteHandler(event:FlexEvent):void {

19.//Native Process가 지원되는지 확인.

20.//지원되려면 반드시 디스크립터에  <supportedProfiles>extendedDesktop</supportedProfiles>을 추가해야한다.

21.if( NativeProcess.isSupported ) {

22.launchEchoTest();

23.} else {

24.Alert.show("NativeProcess not supported.","Error");

25.}

26.}

27.private function launchEchoTest():void {

28.//실행할 Native 애플리케이션 선택

29.var file:File = File.applicationDirectory;

30.file = file.resolvePath("NativeApps");

31.if( Capabilities.os.toLocaleLowerCase().indexOf("win") > -1 ) {

32.file = file.resolvePath("Windows/bin/echoTestWin.exe");

33.} else {

34.file = file.resolvePath("Mac/bin/echoTestMac");

35.}

36.

37.//Native Process를 진행하기 위한 정보 설정

38.var info:NativeProcessStartupInfo = new NativeProcessStartupInfo();

39.info.executable = file;

40.

41.//Native Process생성

42.process = new NativeProcess();

43.

44.//보내고 받을때 발생하는 이벤트에 대한 처리

45.process.addEventListener(ProgressEvent.STANDARD\_OUTPUT\_DATA, function(event:ProgressEvent):void {

46.var date:Date = new Date();

47.var bytes:IDataInput = process.standardOutput;

48.textReceived.text = bytes.readUTFBytes( bytes.bytesAvailable );

49.launchEchoTest();

50.lbDateStamp.text = date.toString();

51.} );

52.process.addEventListener(ProgressEvent.STANDARD\_INPUT\_PROGRESS, function(event:ProgressEvent):void {

53.process.closeInput();

54.} );

55.//Native Process 시작

56.process.start( info );

57.}

58.private function writeData():void {

59.process.standardInput.writeUTFBytes( textToSend.text + "\n" );

60.}

61.]]>

62.</fx:Script>

63.<fx:Declarations>

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

65.</fx:Declarations>

66.<s:VGroup paddingLeft="10" paddingRight="10" paddingTop="10"paddingBottom="10">

67.<mx:Form>

68.<mx:FormItem label="Text to send" direction="horizontal">

69.<s:TextInput id="textToSend"/>

70.<s:Button label="writeData()" click="writeData()"/>

71.</mx:FormItem>

72.<mx:FormItem label="Text received">

73.<s:TextInput id="textReceived" editable="false"/>

74.</mx:FormItem>

75.</mx:Form>

76.<s:Label id="lbDateStamp"/>

77.</s:VGroup>

78.</s:WindowedApplication>

아래 코드는 AIR 애플리케이션과 통신할 C코드(Mac용)이다.

01.#include <stdio.h>

02.#include <stdlib.h>

03.#include <sys/time.h>

04.#include <sys/types.h>

05.#include <fcntl.h>

06.#include <unistd.h>

07.

08.#define BUFFER\_SIZE 8192

09.

10.int main(int argc, char\*\* argv)

11.{

12.char buf[BUFFER\_SIZE];

13.int  cnt;

14.

15.while ( !feof(stdin) )

16.{

17.cnt = read(STDIN\_FILENO, buf, sizeof buf);

18.if (-1 == cnt)

19.{

20.perror("read");

21.exit(1);

22.}

23.if (0 == cnt)

24.{

25.// eof reached...

26.exit(0);

27.}

28.

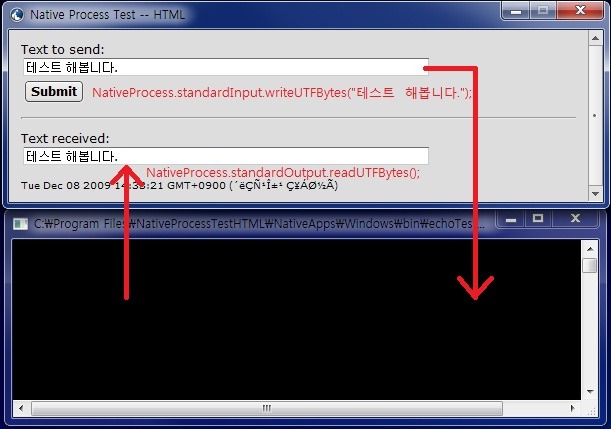
29.write(STDOUT\_FILENO, buf, cnt );

30.}

31.

32.return 0;

33.}

[](http://cfile7.uf.tistory.com/original/144038254B1DE9019D7995)

NativeProcess의 standardInput과 standardOutput 속성은 각각 IDataOutput, IDataInput을 입력과 출력값으로 받고 있다. 그러므로 이 속성을 이용해 표준입출력(standard input/output)을 이용해 서로 통신하게 된다.   
  
매우 다양한 형태로 응용이 가능할 것이라 판단한다.  
  
  
**1.7 File Promise 지원**  
  
AIR 2.0부터 FilePromise 클래스가 새로 추가되어 드래그&드롭 형태로 리모트 파일을 AIR 애플리케이션 외부로 다운로드 받을 수 있게 되었다. URLFilePromise 클래스를 이용해 이 작업을 할 수 있다.   
  
이 기능은 Windows와 Mac에서만 지원되며(Clipboard.supportsFilePromise로 확인) 반드시 AIR 애플리케이션 외부로 드롭하는 경우만 해당한다. 일반적인 클립보드 복사&붙이기 기능으로는 할 수 없다.

* URLFilePromise 클래스 : <http://tinyurl.com/yg5pe32>
* Dropping file promises : <http://tinyurl.com/yj8psb6>
* [A Screencast Explaining and Demoing File Promises](http://blogs.adobe.com/cantrell/archives/2009/10/a_screencast_showing_file_promises.html)

다음 코드는 예시이다.

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

02.<s:WindowedApplication xmlns:fx="<http://ns.adobe.com/mxml/2009>"

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

04.xmlns:mx="<library://ns.adobe.com/flex/halo>"

05.applicationComplete="windowedapplication1\_applicationCompleteHandler(event)">

06.<fx:Script>

07.<![CDATA[

08.import mx.controls.Alert;

09.import mx.events.DragEvent;

10.import mx.events.FlexEvent;

11.

12.private var clipboard:Clipboard;

13.private function windowedapplication1\_applicationCompleteHandler(event:FlexEvent):void {

14.clipboard = new Clipboard();

15.if( !clipboard.supportsFilePromise ) {

16.Alert.show( "File Promise를 지원하지 않습니다." );

17.}

18.}

19.private function onDragOut(event:DragEvent):void {

20.trace( event.target, event.type );

21.var items:Array = fileData.selectedItems;

22.var promises:Array = new Array();

23.for each (var item:Object in items) {

24.var filePromise:URLFilePromise = new URLFilePromise();

25.var request:URLRequest = new URLRequest(item.url);

26.filePromise.request = request;

27.filePromise.relativePath = item.name;

28.promises.push(filePromise);

29.}

30.clipboard.setData(ClipboardFormats.FILE\_PROMISE\_LIST\_FORMAT, promises);

31.NativeDragManager.doDrag(fileData, clipboard);

32.}

33.private function onDragOutComplete(event:NativeDragEvent):void {

34.trace( "Drag out complete");

35.clipboard = new Clipboard();

36.}

37.]]>

38.</fx:Script>

39.

40.<fx:Declarations>

41.<mx:ArrayCollection id="arrColl">

42.<mx:source>

43.<fx:Array>

44.<fx:Object name="rhall.jpg"url="<http://a1.twimg.com/profile_images/57117466/robert_m_hall_bio_photo_big_normal.jpg>"/>

45.<fx:Object name="bobjim.jpg"url="<http://a1.twimg.com/profile_images/51723308/ryancampbell3_normal.jpg>"/>

46.<fx:Object name="jenschr.jpg"url="<http://a1.twimg.com/profile_images/43222252/jenschr_mugshot3_normal.jpg>"/>

47.<fx:Object name="adamflater.jpg"url="<http://a1.twimg.com/profile_images/21503622/Photo_8_normal.jpg>"/>

48.<fx:Object name="reboog711.jpg"url="<http://a1.twimg.com/profile_images/16984682/DSCF0044_normal.jpg>"/>

49.</fx:Array>

50.</mx:source>

51.</mx:ArrayCollection>

52.</fx:Declarations>

53.

54.<mx:DataGrid id="fileData" dragEnabled="true"

55.dataProvider="{arrColl}"

56.dragStart="onDragOut(event)"

57.nativeDragComplete="onDragOutComplete(event)" width="100%">

58.<mx:columns>

59.<mx:DataGridColumn dataField="name" width="100"/>

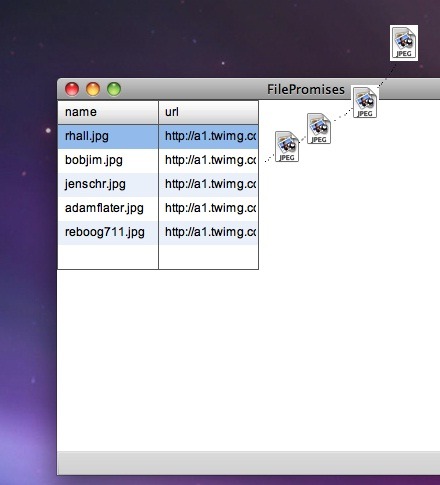
60.<mx:DataGridColumn dataField="url"/>

61.</mx:columns>

62.</mx:DataGrid>

63.

64.</s:WindowedApplication>

[](http://cfile22.uf.tistory.com/original/120E25284B1DFA4B47003B)

**1.8 Global Error Handling**  
  
AIR 애플리케이션이 런타임 도중에 에러가 발생할 때 국부적 Error 처리는 이미 가능했다. 하지만 개발자가 애플리케이션의 어느 부분에서 Error가 발생할지 또는 문제가 발생할지 찾기 힘든 것이 현실이다. 국부적으로 일어나는 에러를 처지하지 못하는 경우 애플리케이션이 동작중에 어떤 부분에서 에러가 발생하든지 대처할 수 있도록 Global Error를 처리할 수 있는 기능이 AIR 2.0과 Flash Player 10.1에 추가가 되었다. 다음 글을 참고하자.

* [Global Error Handling in AIR 2.0 and Flash 10.1](http://blogs.adobe.com/cantrell/archives/2009/10/global_error_handling_in_air_20.html)
* [UncaughtErrorEvent 클래스](http://help.adobe.com/en_US/FlashPlatform/beta/reference/actionscript/3/flash/events/UncaughtErrorEvent.html)
* [Loader.uncaughtErrorEvent](http://help.adobe.com/en_US/FlashPlatform/beta/reference/actionscript/3/flash/display/Loader.html#uncaughtErrorEvents)
* [LoaderInfo.uncaughtErrorEvent](http://help.adobe.com/en_US/FlashPlatform/beta/reference/actionscript/3/flash/display/LoaderInfo.html#uncaughtErrorEvents)

Global Error를 처리하는데 UncaughtErrorEvent 클래스를 이용한다. 이름에서도 볼 수 있듯이 기존의 try..catch문으로 처리하지 못한 Error가 발생할 때 발생함을 알 수 있다.   
  
발생하는 UncaughtErrorEvent를 처리하기 위해 LoaderInfo 객체의 uncaughtErrorEvents 속성에 이벤트 처리를 위한 함수를 등록하면 된다.

loaderInfo.uncaughtErrorEvents.addEventListener( UncaughtErrorEvent.UNCAUGHT\_ERROR, handler );

단, 로드된 SWF 객체의 경우에는 Loader의 uncaughtErrorEvents 속성에 이벤트 처리 함수를 등록한다.

loader.uncaughtErrorEvents.addEventListener( UncaughtErrorEvent.UNCAUGHT\_ERROR, handler );

하나의 SWF가 다른 SWF에 로드되는 경우 UncaughtErrorEvent 이벤트 전파에 대한 처리가 필요하다.  A.swf가 B.swf를 Loader를 이용해 로드했다면 다음과 같은 순서로 LoaderInfo와 Loader 객체에 UncaughtErrorEvent 가 전파된다.

1. (Capture phase) A.swf의 LoaderInfo  
2. (Capture phase) A.swf내의 Loader   
3. (Target phase) B.swf의 LoaderInfo  
4. (Bubble phase) A.swf의 Loader  
5. (Bubble phase) A.swf의 LoaderInfo

Loader객체의 uncaughtErrorEvents 속성은 결코 이벤트 전파시 target phase가 발생하지 않음에 주목하자. 그리고 이렇게 이벤트가 전파되는데는 디스플레이 리스트에 등록되어 있어야만 한다. 즉 addChild()된 객체어야한 한다. 중간에 이벤트 전파를 중단하려면 stopPropagation()이나 stopImmediatePropagation() 메소드를 이용하면 되겠다. 또한 ADL 기반나 디버그용 Flash Player에서 구동 경우 에러표시 다이얼로그가 뜨는 것을 방지하기 위해 UncaughtErrorEvent 이벤트 발생시 preventDefault() 메소드를 사용해 안뜨도록 할 수 있다.  
  
아래는 AIR 2.0 예제이다.

01.package {

02.import flash.desktop.\*;

03.import flash.display.\*;

04.import flash.events.\*;

05.public class UncaughtErrorEventExample extends Sprite {

06.public function UncaughtErrorEventExample()

07.{

08.var options:NativeWindowInitOptions = new NativeWindowInitOptions();

09.options.type = NativeWindowType.UTILITY;

10.

11.var window:NativeWindow = new NativeWindow(options);

12.window.width = 200;

13.window.height = 200;

14.window.title = "UncaughtErrorEventExample";

15.

16.window.stage.scaleMode = StageScaleMode.NO\_SCALE;

17.window.stage.align = StageAlign.TOP\_LEFT;

18.

19.window.activate();

20.window.addEventListener(Event.CLOSING, function($event:Event):void {

21.NativeApplication.nativeApplication.exit();

22.});

23.

24.var btn:Sprite = new Sprite();

25.btn.useHandCursor = true;

26.btn.buttonMode = true;

27.btn.graphics.clear();

28.btn.graphics.beginFill(0xFFCC00);

29.btn.graphics.drawRect(0, 0, 100, 50);

30.btn.graphics.endFill();

31.btn.addEventListener(MouseEvent.CLICK, clickHandler);

32.window.stage.addChild(btn);

33.

34.loaderInfo.uncaughtErrorEvents.addEventListener(UncaughtErrorEvent.UNCAUGHT\_ERROR, uncaughtErrorHandler);

35.}

36.private function uncaughtErrorHandler(e:UncaughtErrorEvent):void {

37.if (e.error is Error) {

38.var error:Error = e.error as Error;

39.// do something with the error

40.trace(error.errorID, error.name, error.message);

41.} else if (e.error is ErrorEvent) {

42.var errorEvent:ErrorEvent = e.error as ErrorEvent;

43.// do something with the error

44.trace(errorEvent.errorID);

45.} else {

46.// a non-Error, non-ErrorEvent type was thrown and uncaught

47.trace( " a non-Error, non-ErrorEvent type was thrown and uncaught " );

48.}

49.e.preventDefault();

50.}

51.private function clickHandler(event:MouseEvent):void

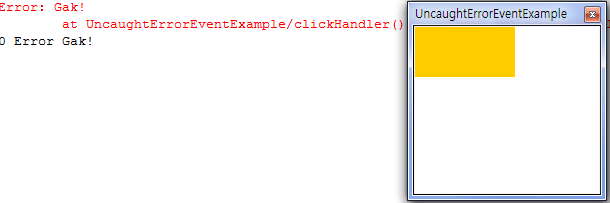
52.{

53.throw new Error("Gak!");

54.}

55.}

56.}

[](http://cfile24.uf.tistory.com/original/110D82244B1F44E129B790)

위 예제에서 볼 수 있듯이 AIR 애플리케이션의 노란 부분을 클릭하면 강제로 에러를 발생시킨다. 이때 이 에러를 처리하기 위한 try...catch문이 없으므로 UncaughtErrorEvent가 발생하고 그것을 처리하고 있다. 마지막에 e.preventDefault()를 둬서 에러표시 다이얼로그가 뜨는 것을 방지한다.   
  
  
**1.9 Packaging an AIR application in a native installer**  
  
AIR 2.0가 생소하게 다가오는 이유중에 하나는 바로 AIR 2.0을 설치할 때 과정이다. 뱃지(Badge)기능을 이용해 웹페이지에서 AIR 런타임 및 해당 AIR 애플리케이션을 다운로드 받을 수 있지만 그 과정이 일반 사용자에게 생소하게 다가올 수 있다. AIR 2.0에서는 이러한 생소함을 덜어주기 위해 각 OS별로 설치패키지를 따로 구성할 수 있도록 지원하고 있다. 가령, Windows는 exe, Mac은 dmg으로 설치 패키지를 구성할 수 있다는 말이다.   
  
AIR 2.0의 ADT의 옵션설정으로 설치 패키지를 만들 수 있다. 다음 링크를 참고한다.

<http://blog.jidolstar.com/629>

**14SEPT/10**[**0**](http://www.flex-tutorial.fr/2010/09/14/air-android-detecter-les-interfaces-reseau-wifi-avec-networkinfo-et-networkinterface/#comments)

## [AIR pour Android – Détecter les interfaces réseau (Wifi) avec NetworkInfo et NetworkInterface](http://www.flex-tutorial.fr/2010/09/14/air-android-detecter-les-interfaces-reseau-wifi-avec-networkinfo-et-networkinterface/)

Voici une fonctionnalité qui n'était pas prévue dans Air 2.5 il y a quelques mois et qui finalement est au rendez-vous. En effet, depuis les derniers builds, on a accès à la classe NetworkInfo, même quand on compile pour une application mobile. Cette classe est déjà présente dans Air 2 et permet de **lister les interfaces réseau** de la machine sur laquelle est lancée l'application.

Sur un mobile, on trouve de nombreuses interface réseau non-actives (préfixées par "mobile\_") mais on trouve aussi des informations plus intéressantes comme **la présence de l'interface Wi-fi**. Vous pouvez ensuite savoir si cette connexion est active et si oui, **récupérer son adresse IP et son adresse MAC**.

Attention, avoir une interface Wi-fi activée et des adresses IP/MAC **ne signifie pas que l'utilisateur a accès au Internet par le Wi-fi**. Vous verrez quand dans mon exemple, je me connecte au réseau Free Wifi du voisin. Je suis connecté à ce réseau mais pourtant, je ne reçoit par internet par ce biais (c'est un hotspot, il faut se logger). Je ne sais pas s'il on peut vraiment savoir si la connexion internet provient de la 3G ou du Wifi, vous pouvez simplement savoir si le Wifi est actif.

Pour lister les interfaces, il faut interroger la méthode "findInterfaces()" sur la propriété static "networkInfo" de la classe NetworkInfo:

var networkInfo:NetworkInfo = NetworkInfo.networkInfo;

var networkInterfaces:Vector.<NetworkInterface> = networkInfo.findInterfaces();

### Documentation

<http://help.adobe.com/en_US/FlashPlatform/beta/reference/actionscript/3/flash/net/NetworkInterface.html>

<http://help.adobe.com/en_US/FlashPlatform/beta/reference/actionscript/3/flash/net/NetworkInfo.html>

### Permissions Android

Pour pouvoir accéder à ces informations, il vous faut ajouter des permissions Android dans le fichier de description:

<android>

<manifestAdditions>

<manifest>

<data>

<![CDATA[

<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />

<uses-permission android:name="android.permission.ACCESS\_WIFI\_STATE" />

]]>

</data>

</manifest>

</manifestAdditions>

</android>

### Vidéo de démonstration

Dans cette vidéo, je lance une petite application Air qui va lister les interfaces réseaux et j'affiche un petit descriptif. Dans un premier temps, le Wifi est désactivé donc la propriété "active" de la NetworkInterface" est à false. Ensuite, j'active le Wifi sur mon portable pour montrer la mise à jour de cette valeur.  


### Code de démonstration

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

<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" paddingLeft="0" paddingRight="0"

paddingTop="0" paddingBottom="0"

applicationComplete="application1\_applicationCompleteHandler(event)">

<mx:Style>

global {

font-size: 24;

}

</mx:Style>

<mx:Script>

<![CDATA[

import mx.events.FlexEvent;

/\*\*

\*

\*/

protected function application1\_applicationCompleteHandler(event:FlexEvent):void {

if (NetworkInfo.isSupported) {

findInterfacesButton.enabled = true;

} else {

showInfo("NetworkInfo non disponible");

}

}

public function onFindInterfaces():void {

var networkInfo:NetworkInfo = NetworkInfo.networkInfo;

var networkInterfaces:Vector.<NetworkInterface> = networkInfo.findInterfaces();

if (!networkInterfaces) {

showInfo("Aucune interface détectée");

return;

}

ta.text = "";

for each (var networkInterface:NetworkInterface in networkInterfaces) {

var name:String = networkInterface.name;

var isActive:Boolean = networkInterface.active;

var displayName:String = networkInterface.displayName;

var addresses:Vector.<InterfaceAddress> = networkInterface.addresses;

var hardwareAddress:String = networkInterface.hardwareAddress;

ta.text += "Name: " + name + "\n";

ta.text += "isActive: " + isActive + "\n";

ta.text += "Display Name: " + displayName + "\n";

if (addresses.length > 0) {

ta.text += "Adresses:\n";

for each (var address:InterfaceAddress in addresses) {

ta.text += " Adresse: " + address.address + "\n";

}

}

ta.text += "Adresse MAC: " + hardwareAddress + "\n";

ta.text += "----------------------\n";

ta.text += "----------------------\n";

}

}

private function showInfo(text:String):void {

log.text += text + "\n";

}

]]>

</mx:Script>

<mx:ApplicationControlBar left="0" right="0" bottom="0" height="80" paddingTop="8" paddingLeft="8"

paddingBottom="8" paddingRight="8" horizontalGap="8" verticalGap="8"

dock="true">

<mx:Box width="100%">

<mx:HBox>

<mx:Button id="findInterfacesButton" width="100%" height="100%" label="Find Interfaces"

click="onFindInterfaces()" enabled="false" />

</mx:HBox>

<mx:Label id="log" fontSize="18" />

</mx:Box>

</mx:ApplicationControlBar>

<mx:TextArea id="ta" backgroundColor="white" width="100%" height="100%" selectable="false" />

</mx:Application>

[Télécharger le projet au format FXP](http://www.flex-tutorial.fr/wp-content/uploads/fxp/NetworkInfoList.fxp)

[Télécharger l'application au format APK](http://www.flex-tutorial.fr/wp-content/uploads/apk/NetworkInfoList.apk)

Remplis sous: [Adobe Air](http://www.flex-tutorial.fr/category/air/), [Android](http://www.flex-tutorial.fr/category/android/), [Exemple](http://www.flex-tutorial.fr/category/exemple/) || Taggé comme: [ACCESS\_NETWORK\_STATE](http://www.flex-tutorial.fr/tag/access_network_state/), [ACCESS\_WIFI\_STATE](http://www.flex-tutorial.fr/tag/access_wifi_state/),[findInterfaces](http://www.flex-tutorial.fr/tag/findinterfaces/), [Mac](http://www.flex-tutorial.fr/tag/mac/), [NetworkInfo](http://www.flex-tutorial.fr/tag/networkinfo/), [NetworkInterface](http://www.flex-tutorial.fr/tag/networkinterface/), [Wifi](http://www.flex-tutorial.fr/tag/wifi/)[Laisser un commentaire](http://www.flex-tutorial.fr/2010/09/14/air-android-detecter-les-interfaces-reseau-wifi-avec-networkinfo-et-networkinterface/#respond)

## Articles similaires

* [AIR Mobile – Détecter la présence d'une connexion Wifi ou 3G](http://www.flex-tutorial.fr/2011/05/12/air-mobile-detecter-la-presence-dune-connexion-wifi-ou-3g/)
* [AIR pour Android – Sortie d'une nouvelle pre-release avec support des composants Flex 4 et Camera](http://www.flex-tutorial.fr/2010/07/21/air-android-sortie-dune-nouvelle-pre-release-avec-support-des-composants-flex-4-et-camera/)
* [AIR pour Android – Debug sur mobile FroYo / emulateur avec Flash Builder 4](http://www.flex-tutorial.fr/2010/08/05/air-android-debug-sur-mobile-froyo-emulateur-avec-flash-builder-4/)
* [AIR pour Android – Ouvrir une application depuis une URI avec arguments](http://www.flex-tutorial.fr/2010/09/12/air-android-ouvrir-une-application-depuis-une-uri-avec-arguments/)
* [Flex ActionScript – Test des performances de l'ActionScript 3](http://www.flex-tutorial.fr/2008/12/27/flex-actionscript-test-des-performances-de-lactionscript-3/)

[Commentaires (0)Trackbacks (0)](http://www.flex-tutorial.fr/2010/09/14/air-android-detecter-les-interfaces-reseau-wifi-avec-networkinfo-et-networkinterface/)[(Souscrire aux commentaires de cet article)](http://www.flex-tutorial.fr/2010/09/14/air-android-detecter-les-interfaces-reseau-wifi-avec-networkinfo-et-networkinterface/feed/)

Aucun commentaire pour l'instant

<http://www.flex-tutorial.fr/2010/09/14/air-android-detecter-les-interfaces-reseau-wifi-avec-networkinfo-et-networkinterface/>